Making Obfuscated PUFs Secure Against Power Side-Channel Based Modeling Attacks

Trevor Kroeger*, Wei Cheng†, Sylvain Guilley‡†, Jean-Luc Danger†‡ and Naghmeh Karimi*
*CSEE Department
University of Maryland Baltimore County
Baltimore, MD 21250
firstname.lastname@umbc.edu
†LTCI, Télécom Paris
Institut Polytechnique de Paris
91 120 Palaiseau (Paris), France
firstname.lastname@telecom-paris.fr
‡Think Ahead Business Line
Secure-IC S.A.S.
35 510 Cesson-Sévigné, France
firstname.lastname@secure-ic.com

***Abstract*—To enhance the security of digital circuits, there is often a desire to *dynamically* generate, rather than *statically* store, random values used for identification and authentication purposes. Physically Unclonable Functions (PUFs) provide the means to realize this feature in an efficient and reliable way by utilizing commonly overlooked process variations that unintentionally occur during the manufacturing of integrated circuits (ICs) due to the imperfection of fabrication process. When given a challenge, PUFs produce a unique response. However, PUFs have been found to be vulnerable to modeling attacks where by using a set of collected challenge response pairs (CRPs) and training a machine learning model, the response can be predicted for unseen challenges. To combat this vulnerability, researchers have proposed techniques such as Challenge Obfuscation. However, as shown in this paper, this technique can be compromised via modeling the PUF's power side-channel. We first show the vulnerability of a state-of-the-art Challenge Obfuscated PUF (CO-PUF) against power analysis attacks by presenting our attack results on the targeted CO-PUF. Then we propose two countermeasures, as well as their hybrid version, that when applied to the CO-PUFs make them resilient against power side-channel based modeling attacks. We also provide some insights on the proper design metrics required to be taken when implementing these mitigations. Our simulation results show the high success of our attack in compromising the original Challenge Obfuscated PUFs (success rate $> 98\%$) as well as the significant improvement on resilience of the obfuscated PUFs against power side-channel based modeling when equipped with our countermeasures.**

I. Introduction

There exists a need for lightweight dynamic security mechanisms that generate random values for authenticating integrated circuits (ICs). The security primitives known as Physically Unclonable Functions (PUFs) have been developed for this purpose. PUFs create values unique to the device in which they are implemented by taking advantage of uncontrollable process variations in the manufacturing process [1]. This occurs despite the implemented designs being identical. PUFs tend to be resilient to reverse engineering attacks owing to the fact that they produce dynamic outputs that are not stored in non-volatile memories. Each PUF takes an input known as a challenge and produces an output known as a response; together these form challenge response pairs or CRPs [1].

The PUF circuits tend to be highly resilient and small in size which makes them well suited for radio-frequency identifiers (RFIDs), smart cards, and other small low-cost devices [2], [3]. PUFs are also useful in more complex systems such as Internet of Thing (IoT) devices which have strict resource constraints [4].They are used to aid in securing the widely distributed IC manufacturing supply chain by performing authentication and locking mechanisms within device such that only the intended supplier and end-user can gain access. PUFs have also taken a role in developing forthcoming technologies which is shown by their suggested use in securing autonomous vehicle communications [5] and cryptographic key generation for cryptocurrencies [6].

As PUFs find their way into everyday life, enhancing their reliability is of the utmost importance, i.e., the response to each challenge should be always the same [7]. However, due to the noise related to the change of operating conditions (e.g., temperature) the PUF response to some challenges may change. One way to improve reliability is to increase the PUFs' signal-to-noise ratio (SNR) [8], achieved via multiple measurements for the same challenge in so-called delay-PUFs.

Although deemed to be unclonable, strong PUFs (PUFs with a large set of CRPs that are mainly used for authentication purposes) and in particular, arbiter-PUFs and its derivatives, have been shown to be vulnerable to the Machine Learning (ML) based modeling attacks where the PUF behavior is modeled based on a subset of its CRPs such that the responses to the unseen CRPs become predictable [9]. A potentially more impactful type of modeling attacks includes those that perform CRP modeling in conjunction with side-channel attacks [10]–[12]. The combined side-channel modeling attack is likely more applicable due to the deployment of PUFs where the response channel is generally cut through anti-fuses after the device's enrollment by the manufacturer. This cutting makes the response traces inaccessible to an attacker, and accordingly other means are needed to attack the device [13]. Furthermore, when increasing the SNR for reliability, the device becomes more susceptible to these combined attacks [14], thus we can use SNR to evaluate the feasibility of such attack.

Several mitigation schemes have been proposed in literature to enhance the resiliency of PUFs against modeling attacks [15]. Among those, Challenge Obfuscation schemes have received a lot of attention where the input challenge is modified in an unpredictable way before the PUF is queried, so the adversary does not have access to the real CRPs [16], [17]. However, as we will show in this paper, the obfuscated PUFs may not stay resilient against power side-channel attacks.

This paper focuses on power side-channel attacks launched through ML algorithms. We show that the Challenge Obfuscation is only effective in preventing CRP based attacks but can be broken by an attack based on monitoring the power traces of the device. We propose circuit modifications that prevent these power side-channel attacks to be effective in realistic environments. All of the attacks presented are performed with *artificial noise added to the system to replicate the environment of real circuitry*. The contributions of this paper are as follows:

978-3-9819263-5-4/DATE21/©2021 EDAA

- Validate Challenge Obfuscation's effectiveness in preventing CRP modeling;
- Attack the PUF response with ML using only the part of the power traces related to the response storage;
- Show that Challenge Obfuscation is ineffective and the PUF size has no impact as the challenges are not necessary for the power trace attack;
- Propose and investigate mitigations against side-channel attacks in the presence of noise.

This paper is organized as follows: A background on the arbiter-PUF and Challenge Obfuscation technique is presented in Sec. II. An overview of PUF modeling methods along with our threat model is given in Sec. III. The proposed side-channel attack mitigations are discussed in Sec. IV. Then Sec. V and Sec. VI discuss our experimental setup and results, respectively. Sec. VII concludes the paper.

II. Background

In this paper, we focus on an emblematic type of delay-PUF, the arbiter-PUF, which is broadly studied for device authentication. This PUF creates a base for many other PUF variants such as the Obfuscated PUF techniques discussed here.

Arbiter-PUF: An arbiter-PUF is composed of a pair of delay chains and generates one response bit per challenge, in a single query [18]. In practice, this PUF operates based on the process-variation induced race between two identical paths (top and bottom paths shown in Fig. 1). The race corresponds to the difference in the delay of these two paths, and is adjudicated by the arbiter [19]. In fact, only the sign of this difference is important (not the exact amount). The sign, which is extracted by the arbiter, presents the PUF identifier (response). The arbiter can be realized by a simple S-R latch implemented through two cross-coupled NAND gates [19].

Note that a full implementation of the PUF, embedded in a chip for generating keys or authentication purposes, would contain a storage mechanism following the PUF's output. This would likely be a Flip-Flop for storing the result of the PUF before the downstream components use the response for authentication or feed it to cryptographic modules to be deployed as a key. In the following sections, we will show that the *system components create power leakages*. These leakages play an important role in the overall power consumption of the PUF, and affect the total power consumption of the chip [20].

Challenge Obfuscated PUF: As mentioned earlier, one of the most powerful techniques to prevent CRP based modeling attacks is Challenge Obfuscation. This technique takes each challenge supposed to be given to the PUF, map it to another before feeding the PUF. Thereby, the challenge that is observed externally is not the very same challenge given to the PUF internally befuddling an attacker's ML algorithm [13], [16], [21]. In this paper, we target one such state-of-the-art obfuscation scheme proposed recently in [16] and show its weakness against our tailored attack. The obfuscation technique presented in [16] is based on a Multiple Input Signature Register (MISR), which takes the input challenge (C) and a programmed nonce (α) and provides it to the structure in Fig. 2 producing a new challenge ($\hat{C}$). The registers loop back within the MISR which with the previously unknown challenge create an externally unknowable input to the PUF. As shown in Fig. 2, this structure is used on top of a standard PUF such as an arbiter-PUF. This PUF is referred to as the **CO-PUF** hereafter.

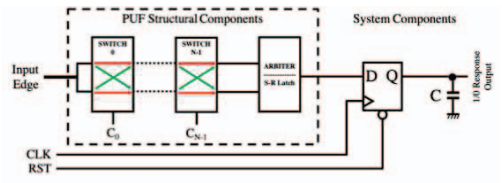

Fig. 1. Structure of an arbiter-PUF [19]. This includes both the PUF structural components as well as the system components.

Note that the CO-PUF presented in [16] includes more components than the main portion depicted in Fig. 2. Those components pertain to the reliability of the PUF and result in more current leakage to occur, and would strengthen our attack even more if considered. The CO-PUF presented in [16] includes control circuitry to manage the queries which is not implemented for this paper. The components that are not part of the main PUF structure in Fig. 2, are likely to remain static during the operation of the PUF. These components therefore would not contribute to power/current fluctuations that we are collecting. However, the noise added to the power traces in our experiments accounts for some variations in these components.

In this paper, we will show that Challenge Obfuscation does not protect the device against power side-channel attacks.

III. Modeling Methods and Threat Model

Modeling Methods: Modeling a PUF consists of characterization of the circuitry through ML techniques. Supervised ML algorithms are used for such modeling and consist of two phases, namely *training* and *evaluation* phases. In the *training* phase, the model is tuned by a set of pre-collected training data which is further used in the *evaluation* phase to predict the output for the inputs unseen during the training phase.

PUF modeling is an attack launched by an adversary in order to mimic the targeted PUF's behavior. This attack is mainly realized by deploying a set of the targeted PUF's CRPs wherein the underlying randomization of the challenge is modeled so that the corresponding response can be discerned [9], [22]. There are a few concerns on modeling PUFs through their CRPs. First, such modeling requires a large number of CRPs. Second, the PUF output is typically cut through anti-fuses

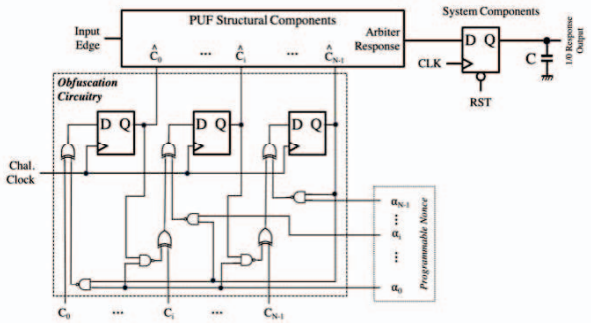

Fig. 2. Block diagram for the implemented Challenge Obfuscation technique presented in [16] and replicated for this paper to be self-contained.

following the *Enrollment phase* of the PUF [21] making such attack almost impossible. In practice, limiting access to the CRPs is performed as a countermeasure against the adversary who opts to use the CRPs to model the PUF behavior.

PUF characterization can be also performed through the modeling of side-channels. Similar to the CRP based modeling attack, in the power based modeling a collection of power traces are used to build the PUF model. The number of traces needed for this model are significantly less that the CRPs required in the CRP based modeling attacks [23]. The primary phenomenon being exploited in the power side-channel attack is the current leakage introduced by the current draw of the Flip-Flop following the PUF circuitry and this phenomenon is confirmed to be present in real silicon devices [24].

Threat Model: In this paper, we assume that an adversary has access to a device with an embedded CO-PUF, specifically the CO-PUF proposed in [16]. This actor has the ability to record the power traces of the targeted PUF during operation. Typically, the chip is modeled when it is still "open" (enrollment phase) to play with any challenges. The adversary's power side-channel based attack occurs subsequently, and allows the adversary to retrieve valuable secrets in the post-customization phase. The attacker then reintroduces the PUF into the supply chain so that it can be deployed in a critical system to exploit.

IV. Proposed Side-Channel Attack Mitigations

Side-channel attack protection for PUFs have already been researched. As an example, it is known that PUFs based on self-oscillating loops can be attacked by measuring the oscillation frequencies. This can indeed be efficiently performed by external sensing of the loop activity, and the subsequent derivation of its frequency with a spectrum analyzer. Once the frequency is known, it is easy to compare frequencies so as to derive the response bits. This is therefore also a kind of side-channel analysis. In [25], such attack is countered by a trick which consists in randomizing the oscillation order: either f_1 is followed by f_2, or vice-versa, and for the next response bit, the random variable considered is the *least significant bit* (hard to estimate accurately) of f_1. In this respect, the countermeasure is that of a *random shuffling*, as often resorted to in side-channel analysis [26]. =The advantage of this shuffling countermeasure is that the randomness cannot be predicted by the attacker, as it is not a process on its own, but the result of previous discreet realization (in that we do not expect an attacker to get that level of side-channel accuracy).

As will be shown through simulation results in Sec. VI, the CO-PUF targeted in this paper although was claimed to be resilient against modeling [16], is highly vulnerable against power side-channel based attacks. Thereby, in this paper we propose two countermeasures (discussed below) to improve the resiliency of the CO-PUF against the adversary who uses the device power traces to build the PUF behavioral model.

Dual Flip-Flop Mitigation: The arbiter included in arbiter-PUF and its obfuscated counterpart is typically implemented by an S-R Latch with 2 complementary outputs (Q and $\bar{Q}$ in Fig. 3), and feeds some sort of storage mechanism, e.g., a Flip-Flop to store the PUF response during its operation before the downstream components use this value. In fact, embedding such Flip-Flop (which is unavoidable) imposes leakage and facilitates modeling the PUF through power side-channel. To counter this leakage, we propose using a dual Flip-Flop (instead of one to store both Q and $\bar{Q}$ latch's outputs).

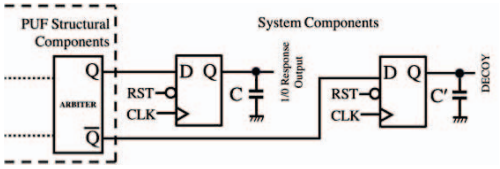

Fig. 3. Dual Flip-Flop Balancing Countermeasure.

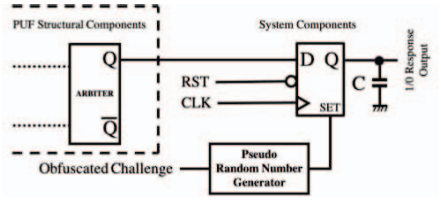

Fig. 4. Randomized Response Setting Countermeasure.

Fig. 3 shows our proposed countermeasure which will be implemented on top of the CO-PUF to make it resilient against power side-channel based modeling attacks. Attaching a similar Flip-Flop on the output of $\bar{Q}$ will mask the switching effects of the primary output Q power-wise. This mitigation is inspired by [27, § 7.3]. Note that the loading of the dual Flip-Flops (emulated by capacitors here) also needs to be matched by leveraging appropriate load for $\bar{Q}$ during implementation by taking into account the load of the circuitry derived by the Q output (namely, $C = C'$ in Fig. 3). For this countermeasure to be effective, one should ensure a good balancing between the two Flip-Flops and their loads.

Randomized Response Setting: This countermeasure, in contrast to the Dual Flip-Flop mitigation method, uses only one Flip-Flop fed by the arbiter's output. Fig. 4 shows this countermeasure. As depicted, to thwart the power side-channel based modeling attack, we initialize the sole embedded Flip-Flop with a random value before applying each challenge. This random setting serves to obfuscate the apparent leakage from the output Flip-Flop due to an unpredictable switching behavior. Note that in our implementation (Fig. 4), the Flip-Flop's set input (SET) is prior to its reset (RST), hence when both are enabled the Flip-Flop gets the value of '1', otherwise as we connected RST to '0' during initialization, the Flip-Flop output would initialize with '0'. In this method, the initialization occurs before applying each challenge to the PUF. Since the leakage is randomized, this method will increase the difficulty in modeling the PUF output. As shown, the randomization is realized via a pseudo-random number generator seeded with the least-significant bit of the obfuscated challenge (i.e., $\hat{C}$ in Fig. 2) as its value is unknown to the adversary.

V. Experimental Setup

We targeted various instances of arbiter-PUF and CO-PUF circuitries using 15,000 random challenges and recording both responses and power traces. These circuits include the 16-, 24- and 64-bit arbiter-PUF implementations and their CO-PUF counterparts as well as the 64-bit CO-PUF equipped with our two mitigations and their hybrid version (i.e., both are used).

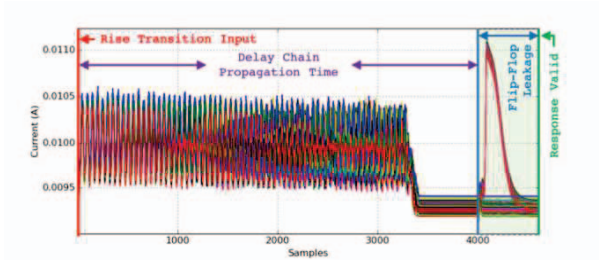

Fig. 5. Timing of the sample window used to collect the power traces of the 64-Bit CO-PUF. The leakage from the Flip-Flop is highlighted.

We leveraged Synopsys HSPICE to simulate the targeted circuitries in the transistor level using a 45 nm NANGATE technology [28]. Process variations 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.

Data Extraction: Fig. 5 shows a set of collected traces from a 64-Bit CO-PUF sampled at 1 ps. The input challenge is set prior to the start of the sampling of our power traces and sampling occurs only after the effects of setting the challenge have become fixed (this includes the registering of $\hat{C}$). The start of sampling coincides with the rising input edge starting the PUF query. Sampling continues through the propagation of the input edge through all of the switches, the arbiter, and during the registration of the response in the output Flip-Flop when the response becomes valid on its output.

Adding Noise: Noise is incurred by the technology: the PUF is embedded into a tiny chip, hence captation of compromising leakage is challenging (since micro-antennae are not off-the-shelf equipment). Moreover, PUF is likely operated in parallel with other IP blocks within the chip, which generate independent activity, thereby entailing additional "algorithmic" noise. Therefore to attempt to characterize traces that reflect the effects of silicon experiments more precisely, we added artificial noise to the power traces after simulation. A noisy trace Y is composed of the original power trace X with the addition of Gaussian noise N as below. We considered $\sigma \in$ {2.5e-4, 16e-4, 32e-4, 64e-4} in our experiments.

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

The larger the noise standard deviation σ, the smaller the SNR.

Attackers can mobilize technical skills to improve the signal despite the "baseline" level of noise. Typically, attackers can focus their probe to eliminate spurious activity, try to adapt it to the signal to be measured, and employ filtering techniques to selectively extract informative from the raw measurement traces. Regardless, the signal-to-noise ratio (SNR) is a measure of the attack feasibility, and is commonly used in side-channel analysis. SNR is the ratio of inter-variance and intra-variance [27, § 4.3.2] and can be assessed via the following equation, where $\mathcal{L}_0$ and $\mathcal{L}_1$ relate to the cases with '0' and '1' responses, respectively. A recent research targeting a real arbiter-PUF shows that a plausible SNR is 1.81 [14]. The SNR for all targeted PUFs in this paper are shown in Sec. VI.

$$\text{SNR} = \frac{\text{Var}(Signal)}{\text{Var}(Noise)} = \frac{\text{Var}([\text{Mean}(\mathcal{L}_0), \text{Mean}(\mathcal{L}_1)])}{\text{Mean}([\text{Var}(\mathcal{L}_0), \text{Var}(\mathcal{L}_1)])}. \tag{2}$$

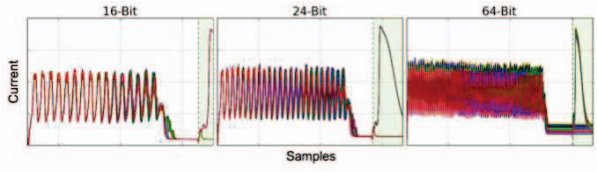

Fig. 6. 100 Power traces related to 100 queries given to CO-PUF implementations of different sizes; the leakage from flip-flop is highlighted.

Attack methodology: In this paper, the principle attack feature is the leakage from the response Flip-Flop. There are several advantages for targeting this leakage namely that the Flip-Flop is sequential, synchronized, and has considerable loading to induce higher leakage. In Fig. 5, which shows the power traces (without noise) of the CO-PUF, the leakage from the Flip-Flop can be clearly seen when each query propagates through this component. It can be clearly seen that the output is discernible for when the response is a '0' or a '1'. It is this portion of the power trace (the highlighted portion of Fig. 5) that is used for modeling the behavior of the PUF.

Support Vector Machine (SVM) algorithm is used to facilitate the modeling attacks. The attack consists of two phases: training and evaluation. In the training phase, the model is built based on a set of the PUF's power traces which is deployed on unseen power traces in the evaluation phase to test if it correctly classifies the response. In this paper, Accuracy is:

$$\text{Accuracy}(\%) = \frac{\text{Predicted Correctly}}{\text{Total Tested}} \times 100\%. \tag{3}$$

Similarly this accuracy is also used when assessing the standard modeling attack on CRPs.

VI. Experimental Results

A. Power Trace Comparison for different PUF Size

This set of results compares the leakage through the Flip-Flip of the CO-PUF with different sizes, mainly 16-, 24- and 64-bit PUFs, to determine the effectiveness of the attacks on various PUF sizes. A collection of power traces from each of these PUF sizes is shown in Fig. 6. This figure shows that the leakage from the Flip-Flop from each of the PUF sizes is exploitable by an adversary in a similar way regardless of the PUF size. This confirms the feasibility of our power side-channel based attack on CO-PUFs of any length.

B. Challenge-Response Pair Based Modeling Attacks

Since modeling attacks are normally based on CRPs, we start by presenting the results of modeling the arbiter-PUF and CO-PUF in this fashion. As shown in Tab. I, the normal arbiter-PUF is highly susceptible to CRP modeling attack with close to 100% accuracy for both the training and evaluation of the model. In contrast, the CO-PUF CRP modeling attack performs very poorly as expected, i.e., ≈50% accuracy in evaluation, ≈7% more accuracy in the training phase. These results correspond with the results presented in [16], which shows around 50% accuracy for the obfuscated PUF and confirms its resiliency against CRP based modeling attacks.

C. Power Side-Channel based Modeling Attack

The SVM algorithm was used to attack the power traces with and without post-simulation noise. The results of the attack are shown in Fig. 7. As depicted, in the absence of noise

TABLE I
THE ACCURACY OF THE CRP BASED MODELING ATTACKS (FOR ONE RESPONSE BIT) LAUNCHED ON A 64-BIT ARBITER-PUF AND 64-BIT CO-PUF FOR DIFFERENT TRAINING SET SIZE. TESTED AGAINST 5000 CRPS SHOWN AS: TEST ACCURACY (TRAINING ACCURACY)

# of Training CRPs	1000	5000	10000
Arbiter-PUF	96.88% (100%)	99.34% (99.94%)	99.24% (99.91%)
CO-PUF	51.78% (63.70%)	51.88% (56.12%)	51.10% (55.23%)

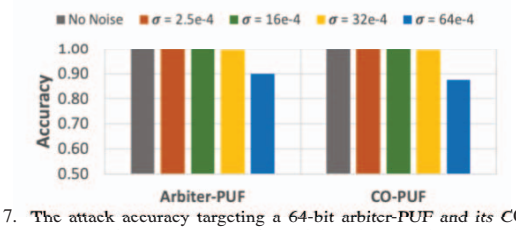

Fig. 7. The attack accuracy targeting a 64-bit arbiter-PUF and its CO-PUF counterpart using 500 power traces for training the model. The model was tested against 5000 power traces.

our attack is very effective for both the arbiter-PUF and the CO-PUF. Such vulnerability can be observed in the presence of noise as well, the accuracy $\approx$ 100% for the noise level even with σ =32e-4). By increasing the noise even further (to σ =64e-4), the attack accuracy decreases but not significantly, i.e., accuracy would be $\approx$90% for both PUFs. The training phase accuracy is not shown due to it being 100% for all levels of noise. The takeaway point from this observation is that *the CO-PUF although resilient against CRP based attacks, is not secure against power side-channel based attacks*, i.e., we can easily model this PUF. Thereby, designing appropriate countermeasures to secure CO-PUFs are unavoidable.

SNR Evaluation: As previously stated, SNR can be used as a measure to assess power side-channel based attacks' feasibility: the lower the SNR the harder the attack. Table II shows the maximal SNR during the leakage of the Flip-Flop registering the output (using Equation 2). As shown for each noise level, the SNR is very similar for the arbiter-PUF and CO-PUF. This table corroborates the previous attack results (Fig. 7), the takeaway point from these observations is that since the SNR is similar for the arbiter-PUF and the CO-PUF in the same noise level, the CO-PUF is as insecure as the arbiter-PUF against power side-channel based modeling attacks. Note that as Fig. 7 shows, our attack is successful even at much higher levels of noise (lower SNR) than what is present in a real silicon ($\approx$1.81 [14]).

TABLE II
THE MAXIMUM SNR FOR THE TRACES WHEN THE FLIP-FLOP IS QUERIED.

	σ = 2.5e-4	σ = 16e-4	σ = 32e-4	σ = 64e-4
Arbiter-PUF	12.224361	0.299846	0.079410	0.021712
CO-PUF	11.827573	0.312342	0.08023	0.022277

D. Power Side-Channel Mitigation Results

We implemented our proposed countermeasures on the CO-PUF to make this CRP based modeling-resilient PUF secure against power side-channel based attacks as well.

Dual Flip-Flop Mitigation: First we discuss the results of the Dual Flip-Flop Mitigation (Fig. 3), for which three different loading variations were realized:

- High Balanced loading on Flip-Flop outputs where C and C' have identical values of 250 fF;
- Low Balanced loading on Flip-Flop outputs where C and C' have identical values of 0 fF;
- Unbalanced loading on Flip-Flop outputs where C is 250 fF and C' is 0 fF.

Fig. 8. The attack accuracy targeting a 64-bit Dual Flip-Flop Mitigation CO-PUF for various noise levels and loading scenarios. The model was trained using 500 power traces and tested against 5000 power traces.

The outcome of the launched power side-channel based attacks is shown in Fig. 8. This mitigation does not have any meaningful effect on the attack outcome in the absence of noise. However, no-noise situation is not realistic. For the High and Low Balanced load cases, the attack accuracy fells to 96.74% and 93.46% when σ=2.5e-4, respectively. When the noise level increased to σ=16e-4 the accuracy was dramatically low, i.e., below 60%. Moreover, when the load was highly unbalanced the attack results were almost identical to the CO-PUF with no countermeasures.

The takeaway point from the Dual Flip-Flop Mitigation result is that the loads on the outputs should be balanced to enhance the resiliency. Thereby, there is a need to design the CO-PUF such that both Flip-Flop's outputs experience a close (if not exactly same) load on their outputs.

Randomized Response Setting: The outcome of the attacks when the Flip-Flop is initialized with a random value before applying each challenge is shown in Fig. 9. Comparing these results with the original CO-PUF (Fig. 7) confirms that this countermeasure degrades the attack security around 20% in the absence of noise and even further in the presence of noise.

Comparing the attack outcomes in the presence of the Dual Flip-Flop and Randomized Response mitigations show that the former outperforms in the higher noise levels and the latter is more efficient when the noise level is lower. Nevertheless the promising point is that the Randomized method is still highly resilient, as in reality there exists a parasitic capacitance tied to V_{dd} which balances the consumption between the rising and falling edges of the Flip-Flop. Hence we considered here the very worst case scenario for the protection.

Hybrid Protection: A combination of both mitigations was implemented to observe if the hybrid version provides more protection than the individual countermeasures. As the balanced load on the output Flip-Flops performed better than the unbalanced load for the Dual Flip-Flop mitigation, we

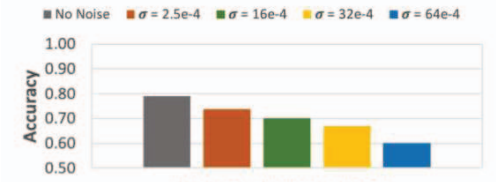

Fig. 9. The attack accuracy targeting a 64-bit CO-PUF equipped with Randomized Initialization for various noise levels. The model was trained using 500 power traces and tested against 5000 power traces.

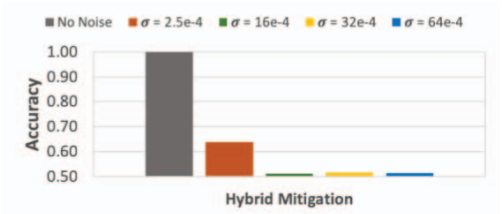

Fig. 10. The attack accuracy targeting a Hybrid mitigation 64-bit CO-PUF for various noise levels and Low Balanced loading scenario. The model was trained using 500 power traces and tested against 5000 power traces.

implement the Hybrid mitigation with a low balanced load. The attack outcome when the CO-PUF is equipped with the hybrid protection implementation is shown in Fig. 10. The noise-free instance is attackable and 100% accuracy is achieved however as previously stated it is not realistic for there to be noiseless. When the noise level increases to σ of 2.5e-4 the attack accuracy drops to 63.78%. At other noise levels of σ=16e-4, 32e-4 and 64e-4, the attack accuracy is around 50% which shows the hybrid mitigation mitigates effectively the power side-channel based modeling attacks.

The takeaway from the combined mitigation is that it is effective to introduce both mitigations so long as the countermeasures are carefully designed such that the loading on the output Flip-Flops is as balanced as possible.

SNR Evaluation: The SNR for the power traces when the CO-PUF is equipped with our proposed mitigations are shown in Table III. The values relate to the time window during which the Flip-Flop is queried. As shown, the SNR for the Unbalanced Dual Flip-Flop implementation is very similar to those without mitigation (Table II) as expected. Also the balanced Flip-Flops fall into the SNR regimes that are attackable but with increased noise the SNR plummets which is expected since the attacks are no longer successful. The SNR for the Randomized Setting mitigation SNR is rather high despite the difficulty it introduces in the attack.

TABLE III
THE MAXIMUM SNR FOR THE TRACES WHEN THE FLIP-FLOP IS QUERIED IN PRESENCE OF THE PROPOSED COUNTERMEASURES.

	σ = 2.5e-4	σ = 16e-4	σ = 32e-4	σ = 64e-4
Dual FF H. Bal. Load	0.766351	0.020974	0.005704	0.001650
Dual FF L. Bal. Load	0.431002	0.010695	0.004800	0.001697
Dual FF Unbal Load	10.760024	0.285023	0.079133	0.022343
Rand. Resp. Setting	0.420705	0.068503	0.022047	0.007547
Hybrid Mitigation	0.034492	0.001739	0.000761	0.000970

E. Discussions

As already shown in Sec. VI-A, our power side-channel based attacks are successfully irrespective to the size of the targeted PUFs. Indeed, the indispensable output Flip-Flop will leak the responses of the PUF. Moreover, our attacks are generic and applicable to any PUFs that sharing the similar structures as in this paper, e.g., XOR-PUFs [22], Multiplexer-PUF, etc. Moreover, from perspective of countermeasure, our mitigations against power side-channel based modeling attacks are also generic and independent from any specific PUF.

VII. CONCLUSION AND FUTURE DIRECTIONS

Challenge Obfuscation schemes have been proposed to make the PUFs resilient against the attacks that leverage a subset of the PUF's challenge response pairs to model the PUF behavior. In this paper, we targeted one of the state-of-the-art Challenge Obfuscation schemes implemented on top of a standard arbiter-PUF and showed that although challenge obfuscation makes a PUF secure against modeling attacks, the PUF behavior can still be revealed through its power side-channel. We proposed two countermeasures along with their hybrid version that can enhance the PUF security significantly against power side channel attacks. We will investigate our findings in PUFs realized in real silicon in near future.

REFERENCES

[1] C. Herder et al., "Physical unclonable functions and applications: A tutorial," *Proc. of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.
[2] N. Karimi et al., "Impact of aging on the reliability of delay PUFs," *JETTA*, vol. 34, no. 5, pp. 571–586, 2018.
[3] Z. Cherif et al., "An Easy-to-Design PUF based on a single oscillator: the Loop PUF," in *DSD*, September 5-8 2012.
[4] T. Idriss, H. Idriss, and M. Bayoumi, "A PUF-based paradigm for IoT security," in *IEEE 3rd World Forum on IoT*, 2016, pp. 700–705.
[5] Q. Jiang et al., "Two-Factor Authentication Protocol Using Physical Unclonable Function for IoV," in *IEEE/CIC ICCC*, 2019, pp. 195–200.
[6] A. Mars and W. Adi, "New Concept for Physically-Secured E-Coins Circulations," in *Adaptive Hardware and Systems*, 2018, pp. 333–338.
[7] Y. Gao, H. Ma, G. Li, S. Zeitouni, S. Al-Sarawi, D. Abbott, A.-R. Sadeghi, and D. Ranasinghe, "Exploiting PUF Models for Error Free Response Generation," *ArXiv*, vol. abs/1701.08241, 2017.
[8] S. Morozov et al., "An analysis of delay based PUF implementations on FPGA," in *ARC*, 2010, pp. 382–387.
[9] U. Rührmair and J. Sölter, "PUF modeling attacks: An introduction and overview," in *DATE*, 2014, pp. 1–6.
[10] A. Mahmoud et al., "Combined Modeling and Side Channel Attacks on Strong PUFs," *IACR Cryptology ePrint Archive*, vol. 2013, p. 632, 2013.
[11] U. Rührmair et al., "Efficient power and timing side channels for physical unclonable functions," in *CHES*, 2014, pp. 476–492.
[12] F. Ganji and S. Tajik, "Physically Unclonable Functions and AI: Two Decades of Marriage," *ArXiv*, vol. abs/2008.11355, 2020.
[13] X. Xi, A. Aysu, and M. Orshansky, "Fresh re-keying with strong PUFs: A new approach to side-channel security," in *HOST*, 2018, pp. 118–125.
[14] K. Fukushima 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*, 2016, pp. 201–207.
[15] A. Vijayakumar and S. Kundu, "A novel modeling attack resistant PUF design based on non-linear Voltage Transfer Characteristics," in *DATE*, 2015, pp. 653–658.
[16] S. S. Zalivaka et al., "Reliable and modeling attack resistant authentication of arbiter PUF in FPGA implementation with trinary quadruple response," *IEEE TIFS*, vol. 14, no. 4, pp. 1109–1123, 2019.
[17] Q. Wang, M. Gao, and G. Qu, "A machine learning attack resistant dual-mode PUF," in *Great Lakes Symposium on VLSI*, 2018, pp. 177–182.
[18] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *DAC*, 2007, pp. 9–14.
[19] B. Gassend et al., "Silicon physical random functions," in *CCS*, 2002, pp. 148–160.
[20] D. Merli et al., "Side-channel analysis of PUFs and fuzzy extractors," in *Trust and Trustworthy Computing*, 2011, pp. 33–47.
[21] Y. Gao et al, "Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices," in *PerCom Workshops*, 2016, pp. 1–6.
[22] U. Rührmair et al., "Modeling attacks on physical unclonable functions," in *CCS*, 2010, pp. 237–249.
[23] T. Kroeger et al, "Effect of aging on PUF modeling attacks based on power side-channel observations," in *DATE*, 2020, pp. 454–459.
[24] Y. Yu 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.
[25] L. Tebelmann et al, "Self-Secured PUF: Protecting the Loop PUF by Masking," in *COSADE*, 2020.
[26] N. Veyrat-Charvillon et al, "Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note," in *ASIACRYPT*, 2012, pp. 740–757.
[27] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2006.
[28] "Nangate 45nm open cell library," "http://www.nangate.com".