Cross-PUF Attacks on Arbiter-PUFs through their Power Side-Channel

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, CNRS, 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

2020 IEEE International Test Conference (ITC) | 978-1-7281-9113-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITC44778.2020.9325241

***Abstract*—The silicon primitives known as Physically Unclonable Functions (PUFs) are used for various security purposes including key generation, device authentication, etc. Due to the imperfections in manufacturing process, PUFs produce their unique outputs (responses) for given input signals (challenges) fed to identical circuitry designs. Although PUFs are deployed to preserve security and are assumed to be unclonable, their functionality may still be compromised by modeling attacks. However, such attacks only target one single PUF aiming at reversing its behavior (based on a subset of its challenge-response pairs), and are not useful for attacking other PUFs. Moreover a subset of the target PUF's response has to be known by the attacker. This paper moves one step forward and investigates the possibility of *Cross-PUF* attacks in which a particular PUF's power fingerprints can be used to break another PUF's security. In these *Cross-PUF* attacks, the attacker has at his disposal a reference PUF, and uses its power side-channel to train a machine learning model which can be deployed to attack other identical PUFs. The experimental results show the high success of the proposed attacks even in presence of noise and temperature differences between the target PUF and the one used to train the model. We target arbiter-PUFs but we deduce that the findings extend to all its derivatives, e.g., XOR-PUFs and Feed-Forward-PUFs.**

I. Introduction

With the increasing concern about the security of Integrated Circuits (ICs), Physically Unclonable Functions (PUFs) are broadly deployed to provide a unique signature for each IC. A PUF signature can be used for device authentication or generating secret keys in cryptographic devices. PUFs generate unique signatures despite having identical circuit designs due to the random process variations, related to a normal distribution of inadvertent technological perturbations [1]. Using a PUF avoids storing keys in digital memory, thereby enhancing the security of the systems in which they are embedded.

With the distribution of IC design and manufacturing process all over the globe, IC overproduction becomes a major threat. To address such a threat, PUFs can be used in order to perform authentication or unlocking of approved devices for regular use. PUFs are also being postulated as an authentication mechanism to prevent nefarious activities in the communication and operation of autonomous vehicles [2]. Thanks to their small size and unclonability, PUFs have found their way into the resource constrained Internet of Things (IoT) devices [3]. Due to the efficacy of using PUFs in cryptographic key generation, they are also being considered for securing cryptocurrencies [4]. PUFs are also particularly well suited for low-cost devices such as smart cards [5], [6].

A PUF's signature corresponds to its input and output pairs, so-called Challenge-Response Pairs (CRPs)—c.f. ISO/IEC 20897. For each PUF, the CRPs are registered once after the PUF fabrication and during the enrollment phase. However, when the PUF is used, corresponding to the reconstruction phase, its CRPs can be erroneous due to measurement noise. To increase the reliability of PUFs, it is necessary to have a high signal-to-noise ratio (SNR), or perform post-processing relying on error correcting codes. SNR can be improved in delay-PUFs [7] where n elements are chained, and the total delay of the chain is measured. In this paper, we focus on an emblematic type of delay-PUFs, the arbiter-PUF, which is broadly studied for device authentication.

Another problem with the PUF usage is its sensitivity to attacks. Although PUFs are deployed to preserve security and are assumed to be unclonable, even the so-called strong PUFs, such as the arbiter-PUF, may be compromised by modeling attacks [8], side-channel attacks [9], or a combination of the two [10]. In the modeling attacks an adversary collects extensive number of CRPs and uses them to predict the PUF response for other challenges based on statistical methods including Machine Learning (ML) techniques [11].

Targeting a PUF using its power traces is of great interest for ML attacks as once the chip is enrolled and the response is not accessible anymore (generally this is cut by an anti-fuse), the only *adversarial* way to observe the response is by indirect side-channel captures. Therefore, one can imagine an ML attack scenario where the attacker registers a training dataset (including power traces) during enrollment and perpetrates the attack when the PUF is in use with unseen challenges.

This paper extends the scope of PUF modeling attacks and further investigates the effectiveness of such attacks by successfully attacking one implementation of a PUF with a model created by another implementation. In other words, attacking one PUF using the power traces of a reference PUF implemented on the same wafer i.e. via similar GDSII file. We refer to these attacks as a *Cross-PUF* attacks hereafter.

The contributions of this paper are as follows:

- Successful attacks of one PUF based on a model created from a different PUF instance (i.e. *Cross-PUF*), even in the presence of temperature mismatch and noise;
- Assessment of the susceptibility of the deployed PUFs against *Cross-PUF* attacks targeting either their arbitration latch or their response bit sampling Flip-Flop.

II. PUF Modeling Schemes

Typically modeling attacks are launched by intercepting a set of CRPs of the target PUF. Using such a set, an ML model is trained such that the response of any unobserved

978-1-7281-9113-3/20/$31.00 ©2020 IEEE

challenge is predicted [8], [12]. These attacks aim to glean several responses from corresponding challenges, and attempt to characterize function F in the equation: $r = F(c)$, where r is the response and c is the challenge of the target PUF F.

Attacking a PUF through its CRPs has two issues. The first is the excessive number of CRPs required to model a PUF. The second is the accessibility to the portions of the circuit that would reveal the CRPs as those areas are rendered inaccessible after enrollment through anti-fuses [13]. With respect to the proposed *Cross-PUF* attacks, CRPs are unique for each instance of PUF realized from the same GDSII file, hence it is impossible to launch *Cross-PUF* attacks using PUFs' CRPs.

A more realistic modeling attack on PUFs is performed via monitoring the power consumption of the underlying device over time [14]–[16] where the current drawn by the PUF is monitored during the time it is queried with challenges and the related traces are recorded. These traces are correlated to the physical specification of the target PUF and can be used (instead of challenges) to train an ML model to mimic the PUF behavior. This model can then be used to infer the PUF responses.

Since the power traces reveal the underlying characteristics of the PUF's design, we use them to launch *Cross-PUF* attacks, i.e., building a model created on one PUF to attack another PUF realized from the same wafer/GDSII file.

III. Threat Model

In this paper, we consider a threat model in which the adversary does not have access to the response of the target PUF, but has at his disposal a reference PUF. In addition, he can observe the target's PUF activity via its power consumption traces. The proposed *Cross-PUF* attacks are non-invasive "profiling" attacks which allow the attacker to predict the response of the target PUF via its power consumption traces through the model trained by the reference PUF's power traces. This differs from previous research where the PUF that was used for training the model is the very targeted PUF [10], [15].

As different PUF instances behave differently, it seems paradoxical to attempt to learn from another instance. However, the motivation is that even though the responses to a given challenge are intrinsic to one PUF instance, the side-channel leakage of an instance is highly correlated to its response (at least close to the arbiter, in time). We show in this paper that this relationship does exist and can be exploited. In addition, we address a more complex threat model where the adversary is not aware of the operating temperature of the target PUF to align the temperatures of the two PUFs.

In this paper we target arbiter-PUFs. An arbiter-PUF is composed of a pair of delay chains and generates one response bit per challenge, in a single query [17]. It operates based on the process-variation induced race between the two identical paths (top and bottom paths shown in Fig. 1). The arbiter can be realized by a simple S-R latch (easy symmetric layout).

Note that a full implementation of the PUF, embedded in a chip for generating keys or authentication purposes, would contain some sort of storage mechanism following the PUF's output such as a Flip-Flop for storing the response bits before the downstream components use the PUF response. As we will show these system components create power leakages that play an important role in the overall power consumption of the PUF, and affect the total power consumption of the chip [9].

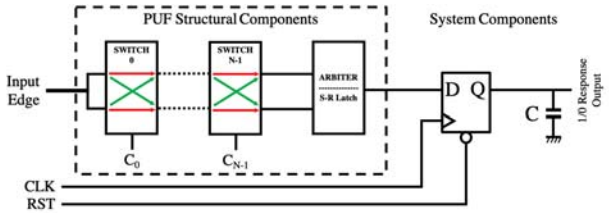


Fig. 1: Structure of an arbiter-PUF [5]. This includes the PUF structural components as well as the system components.

IV. Proposed Attack Methodology

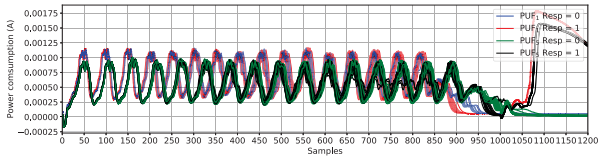
In practice, there are two components in the arbiter-PUF (Fig. 1) that can be targeted for the *Cross-PUF* attacks: the *Latch arbiter* and the *embedded Flip-Flop*. Specifically, the latch is intrinsic to the arbiter-PUF as it carries out the arbitration, while the Flip-Flop is extrinsic to the PUF as it depends on the system which uses the PUF output. Therefore, when considering the PUF as a primitive, the leakage of latch has to be evaluated, while after the PUF is embedded in a system, the leakage of the Flip-Flop has also to be assessed. In particular, one advantage of targeting the Flip-Flop is that it is sequential, synchronized and heavily loaded compared to the PUF's latch. This facilitates a side-channel attack on the Flip-Flop. Figure 2 shows the power traces when the query propagates through the deployed latch and Flip-Flop.

Targeting the Latch: At the point of time when the latch is queried, the power traces are highly distinct from each other. This point of time is crucial in distinguishing the output as it is when the delays of the related paths are compared and the PUF's response is decided for each challenge [18]. Due to the process variations, the latches being active at slightly different times. Thereby, to increase the success rate of the attacks, the traces are first shifted or aligned[1] appropriately (as shown in Fig. 2b), and then these aligned traces are targeted. Indeed, it was observed that the *Cross-PUF* attack was not successful when the power traces were not aligned (as shown in Fig. 2a).

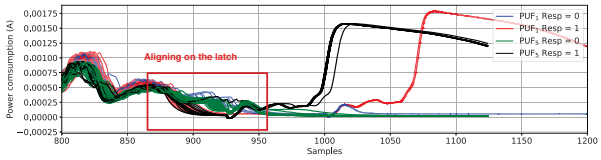
Targeting the Flip-Flop: Similar to the latch, the Flip-Flop output is also a good target for attacks. As shown in Fig. 2a, due to the load on the Flip-Flop output, the power traces related to '0' and '1' responses are clearly separated from each other in the point of time when the Flip-Flop is queried. Accordingly, the response can be determined without the use of modeling techniques in the absence of noise. Due to the clock synchronization no alignment is necessary for the attack of the Flip-Flop. We also study the leakage from the latch stage (this leakage cannot be avoided) due to the fact that the latch is a part of the PUF, which could be an IP Core that is utilized across multiple designs thus making the attack more portable.

We launch the attacks which take advantage of the ML algorithms consisting of two phases: training and evaluation. (Note that this occurs after the aforementioned alignment required for attacking the latch). In the training phase, we build the model based on the power traces of the reference PUF

[1]In this paper, the alignment means shifting all power traces of a PUF by a fixed distance, e.g., all traces of PUF_5 are shifted left by 75 samples in Fig. 2b to align with PUF_1 on the latch. The values for shifting depend on PUF instances, which can be observed easily from the traces, e.g., in Fig. 2a.



(a) Full traces of two PUFs.



(b) Zoomed in traces after alignment (by shifting) on the latch.

Fig. 2: Superimposing 50 traces of PUF_1 and PUF_5.

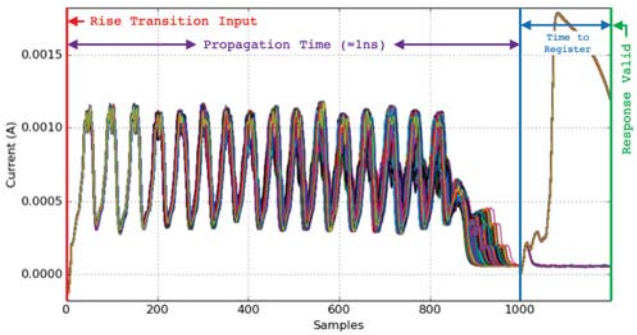


Fig. 3: Timing of the sample window used to collect the power traces of the PUFs.

and the corresponding responses. Finally, in the evaluation phase, unseen inputs (power traces in our case) are tested to investigate if the model correctly classifies the response. In this paper the Support Vector Machine (SVM) [11] scheme is used to launch the modeling attacks.

V. Experimental Setup

We implemented the targeted arbiter-PUF shown in Fig. 1 (with a load capacitor of 250 fF), at the transistor level using a 45 nm technology from the NANGATE library [19]. Using Synopsys HSPICE, we conducted five Monte-Carlo simulations, each representing one 16-stage PUF_i where $i \in \{1, 2, 3, 4, 5\}$ using a Gaussian distribution: transistor gate length L: $3\sigma = 10\%$, threshold voltage V_{TH}: $3\sigma = 30\%$, and gate-oxide thickness t_{OX}: $3\sigma = 3\%$. All five of the PUF result sets are independently identically distributed. We utilized PUF_1 as the reference PUF and the other PUFs for validation of the *Cross-PUF* attacks. To study the effect of temperature misalignments in the success of the *Cross-PUF* attacks, we also simulated PUF_1 at 60°C.

Data Extraction: The entire cycle for querying the PUF is 5 ns. The PUF is fed with a rise transition 2.5 ns after applying each challenge.The PUF's current is sampled 1000 times between the time that the PUF is fed with the rise transition and the time that the response becomes stable ($<$ 1 ns after the input edge). The Flip-Flop's clock period is 5 ns with a rising transition on its clock signal 1 ns after each transition of the PUF input, to register the response. Figure 3 shows a set of collected traces, sampled within the aforementioned window.

Adding Noise: To account for the noise effects occurring in real silicon experiments, artificial noise was added to the power traces post simulation. Four different levels of Gaussian noise N, with $\sigma \in \{2.5\text{e-}4, 16\text{e-}4, 32\text{e-}4, 64\text{e-}4\}$, were added to the original power traces X to obtain the noisy traces Y:

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

To compare the level of the noise added in our experiments with the state-of-the-art research, one can refer to [16] which targets a real arbiter-PUF using power traces. The metric used in [16] for leakage detection is the ratio of inter-variance and intra-variance. This ratio is in fact the SNR [20, § 4.3.2] which is commonly used in side-channel analysis.

Let $\mathcal{L}$ be one sample point in power traces, then all traces can be categorized into two classes $\mathcal{L}_0$ and $\mathcal{L}_1$, where the subscripts correspond to two responses of a PUF. Hence,

$$\text{SNR} = \frac{\text{Var}(Signal)}{\text{Var}(Noise)} = \frac{\text{Var}([\text{Mean}(\mathcal{L}_0), \text{Mean}(\mathcal{L}_1)])}{\text{Mean}([\text{Var}(\mathcal{L}_0), \text{Var}(\mathcal{L}_1)])}. \quad (1)$$

The maximum SNR in [16] is estimated to be 1.81 whereas in our cases, as shown in Table I, the maximum SNR of PUF_1 when targeting the latch or Flip-Flop is much lower.

TABLE I: The maximum SNR for the traces related to the PUF_1's latch and Flip-Flop.

	σ = 2.5e-4	σ = 16e-4	σ = 32e-4	σ = 64e-4
Latch	0.235314	0.009083	0.002350	0.001593
Flip-Flop	12.320019	0.308742	0.079990	0.022701

VI. Experimental Results

A. Attack Success Rate

In these experiments, we used 200 power traces to train the model and 11,000 traces to test it. The accuracy was calculated based on the correctness of response prediction.

1) Self-PUF Attacks: Here we target each PUF using its own power traces, so-called *Self-PUF* attack hereafter, and use it as a baseline for the *Cross-PUF* attack results.

Targeting the Latch: This set of results investigates the success of the modeling attacks on the target PUF when the power traces are monitored at the point of time the latch is queried.

With as low as 40 training traces the targeted PUF can be modeled with a high accuracy (≈97%) using the SVM algorithm. The modeling accuracy increases to 99% by using 200 training traces. These full *Self-PUF* attack results are shown in the bolded diagonal of Table II.

Targeting the Flip-Flop: Attacking the arbiter-PUF using its own power traces by targeting its embedded Flip-Flop result in 100% accuracy. This can be easy observed in Fig. 2a. As shown, the power traces led to responses '0' and '1' are highly distinct from each other in the point of time when the Flip-Flop is queried. Such distinction leads to 100% modeling accuracy.

The takeaway point from these observations is that an arbiter-PUF can be modeled using its power traces at the point in time when either the latch or Flip-Flop are queried. The latter, is more strong as the power traces are highly dependent to the PUF response when the embedded Flip-Flop is queried.

2) Cross-PUF Attacks: Recall that our proposed *Cross-PUF* attacks are where one PUF is used as the reference to build the model, and the other PUFs are being targeted.

Targeting the Latch: Table II demonstrates the *Cross-PUF* attack accuracy for all PUFs when targeting the latch. The

diagonal of this table shows the *Self-PUF* attacks where each PUF is attacked using its own power traces. As shown, the average accuracy of the *Self-PUF* attacks is 99.68% while the attacks accuracy is 97.30% for the *Cross-PUF* attacks when 200 traces are used for training the model. The minimum accuracy for the *Cross-PUF* attacks is ≈94.5%.

TABLE II: Accuracy of *Cross-PUF* attacks for each PUF pair.

Modeled PUF	Traces used for Training	Attacked PUF				
		PUF_1	PUF_2	PUF_3	PUF_4	PUF_5
PUF_1	200	**0.9998**	0.9605	0.9965	0.9997	0.9483
PUF_2	200	0.9454	**0.9987**	0.9776	0.9517	0.9545
PUF_3	200	0.9735	0.9997	**0.9983**	0.9775	0.9494
PUF_4	200	0.9936	0.9700	0.9815	**0.9975**	0.9470
PUF_5	200	0.9880	0.9951	0.9640	0.9855	**0.9895**

Targeting the Flip-Flop: Similar to the *Self-PUF* attacks that targeted the embedded Flip-Flop, in *Cross-PUF* attacks the '0' and '1' responses are clearly discerned from each other based on the power consumption of the Flip-Flop. This can be observed in Fig. 2a. Based on our experiments, the accuracy of such attacks is ≈100%.

The takeaway point from this set of experiments is that we can successfully launch *Cross-PUF* attacks targeting either of the arbiter latch or the embedded Flip-Flop and they can be as strong as *Self-PUF* attacks. This is a significant threat for the security of devices employing PUFs, since the adversary can model a PUF realized from the same GDSII to break the security of the target PUF.

B. Attacks Efficiency in the Presence of Noise

To show the efficacy of the proposed attacks in real silicon experiments, as discussed in Section V, Gaussian noise was added to the power traces extracted from our HSpice simulations. To launch the *Cross-PUF* attacks, on both the latch and Flip-Flop, the model was trained on PUF_1 using 2,000 traces and tested against 11,000 traces of each of the other PUFs.

Targeting the Latch: The attack results when the arbiter latch is targeted are shown in Fig. 4. This figure depicts the modeling accuracy in different levels of noise. As shown, the attacks are highly successful when the noise σ = 2.5e-4, i.e., 97% accuracy for the *Self-PUF* attacks (attacking PUF_1), and more than 92% accuracy for the *Cross-PUF* attacks. However, the attack accuracy for the other noise levels is considerably less as the power traces themselves are fully concealed by the noise, therefore the SNR is too low to successfully launch the *Self-PUF* or *Cross-PUF* attacks. The results show that in the presence of acceptable amount of noise (i.e. reasonable SNR), both *Cross-PUF* and *Self-PUF* attacks are highly accurate.

Targeting the Flip-Flop: The attack accuracies are shown in Fig. 5. As depicted, in these attacks, the accuracy decreases when increasing the noise level, even-so the attacks are still highly successful, i.e., the accuracy is ≈100% across all attacks (*Self-PUF* and *Cross-PUF* attacks) when σ = 16e-4 and for σ = 32e-4, the accuracy drops only marginally to ≈98.5%, Increasing the noise even further finally results in a dip when σ = 64e-4. However, in this level still the *Self-PUF* attacks have 91.4% accuracy and the *Cross-PUF* attacks experience between 83.7% and 89.5% accuracy.

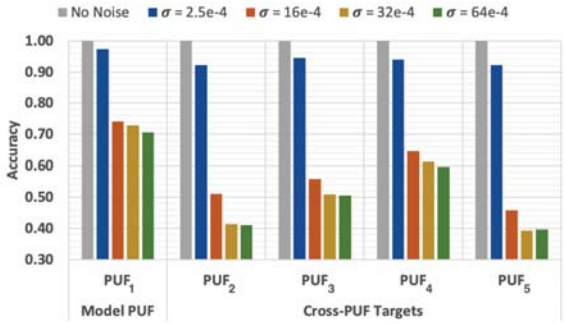


Fig. 4: *Cross-PUF* attacks targeting the arbiter latch in five PUFs in the presence of different noise levels.

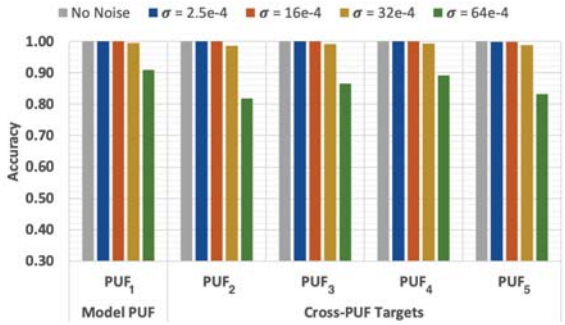


Fig. 5: *Cross-PUF* attacks targeting the embedded Flip-Flop in five PUFs in the presence of different noise levels.

The takeaway from these experiments is the high success of the *Cross-PUF* attacks even in the presence of noise (albeit with a reasonable SNR).

C. Attacks Efficiency in Case of Temperature Misalignment

In these experiments, we consider PUF_1 operating at 60°C as a reference and target PUF_1 and the other 4 PUFs operating at 80°C. The model was trained with 1,000 power-traces and tested against 11,000 power-traces. Figure 6 shows superimposed traces of PUF_1 operating at different temperatures. As expected the PUF operates faster at lower temperatures.

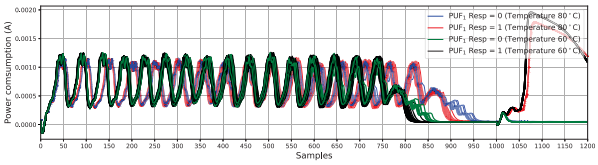


Fig. 6: Superimposing 50 traces of PUF_1 under different temperatures to observe the difference in the collected traces.

Targeting the Latch: Figure 7 shows the effect of temperature misalignments on the attacks accuracy when the arbiter latch is attacked. As shown, for the *Self-PUF* attacks, the PUF can be modeled with 100% accuracy in case of no noise. The accuracy changes to 96.11% when Gaussian noise with σ = 2.5e-4 is added artificially to the power traces and diminishes with increasing σ.Again, we want to emphasize that the noise levels with $\sigma >$ 2.5e-4 results in a very low SNR.

The results shown in Fig. 7 confirm that *Cross-PUF* attacks through latches are performed at an accuracy higher than >97% for the SVM algorithm in the case of no noise. When

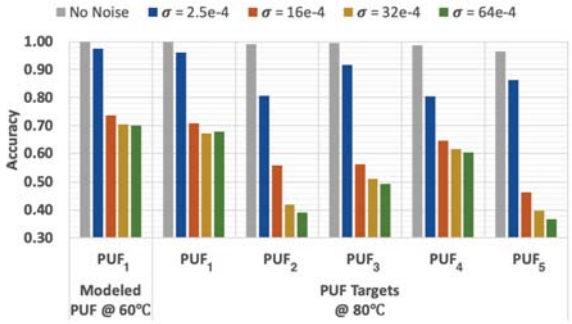


Fig. 7: The temperature misalignment modeling results for the *Self-PUF* and *Cross-PUF* attacks targeting the latch.

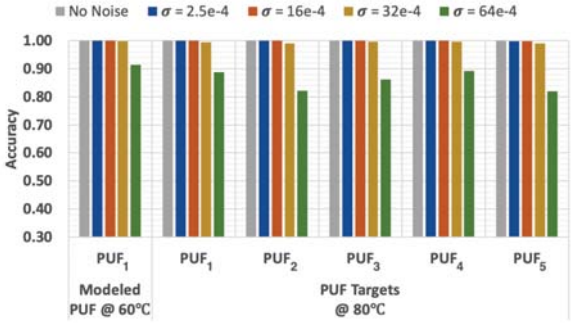


Fig. 8: The temperature misalignment modeling results for the *Self-PUF* and *Cross-PUF* attacks targeting the Flip-Flop.

noise is added with σ = 2.5e-4 the accuracy decreases to 80% for the *Cross-PUF* attacks. Note that the greater the noise level, the lower the accuracy.

Targeting the Flip-Flop: The results of both *Self-PUF* and *Cross-PUF* attacks targeting the Flip-Flop are shown in Fig. 8. As depicted, targeting the Flip-Flop results in 100% accuracy for the *Self-PUF* attacks in case of no noise up to a noise with $\sigma \leq$ 16e-4 even when there are temperature misalignments. The results are very similar for *Cross-PUF* attacks, i.e., the average accuracy of >99% when $\sigma \leq$16e-4. Both the *Self-PUF* and *Cross-PUF* attacks demonstrate more than 98.9% accuracy for $\sigma \leq$32e-4. Finally, the accuracy diminishes on average to 88% and 84% when σ = 64e-4 for the *Self-PUF* and *Cross-PUF* attacks, respectively.

The takeaway from these observations is that the *Cross-PUF* attacks are highly successful despite having a misalignment in temperature between the modeled and attacked PUFs. This observation makes the attack more realistic as the adversary may not be able to control the temperature of the target PUF.

VII. Discussion about the scope of the attack

The experimental results show that an arbiter-PUF can be attacked via profiling the traces of a reference PUF realized from the same GDSII. The simulation results clearly confirm that the Flip-Flop sampling the arbitration result is the main leakage source when compared to the arbitration latch. Note that any kind of arbiter-PUF, even those more robust against modeling attacks (through their CRPs) such as the Feed-Forward PUF or XOR-PUF can be targeted in a similar way (i.e., via *Cross-PUF* attacks). There are several reasons explaining this significant leakage at the Flip-Flop stage. First, the Flip-Flop is necessarily connected to the system bus and thus more heavily loaded. Secondly, the output is synchronized with the system clock, hence there is no need of synchronization and the peak of energy is denser. Finally, the Flip-Flop has a fixed initial state which can be forced by the reset signal. Thus the leakage is both intense and reproducible.

VIII. Conclusions

We proposed the *Cross-PUF* attacks which allow an adversary to attack an arbiter-PUF and its derivatives by using the power traces of another counterpart fabricated from the same wafer (or same GDSII), without the knowledge of CRPs. This corresponds to both enhanced modeling attacks and profiling attacks, as a reference PUF is required. Compared to the classical modeling attacks which target a single PUF and requires a subset of its CRPs, the *Cross-PUF* attacks allow the adversary to target all PUFs fabricated from the same GDSII by observing their side-channel leakage. An ML model is trained with the power traces of the reference PUF, targeting the leakages related to either the arbitration phase at the latch level, or to the sampling in a system D-Flip-Flop. We showed that the *Cross-PUF* attacks are highly successful despite temperature misalignments between the reference and target PUFs. It has been shown that the embedded Flip-Flop fed by the arbiter-PUF output is the main source of leakage. The efficiency of the proposed attacks has been validated by simulation using realistic SNR in different temperature conditions. As future works we will investigate countermeasures and the *Cross-PUF* attacks considering aging effects.

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] Q. Jiang et al., "Two-Factor Authentication Protocol Using Physical Unclonable Function for IoV," in *IEEE/CIC ICCC*, 2019, pp. 195–200.
[3] 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.
[4] A. Mars and W. Adi, "New Concept for Physically-Secured E-Coins Circulations," in *Adaptive Hardware and Systems*, 2018, pp. 333–338.
[5] N. Karimi et al., "Impact of aging on the reliability of delay PUFs," *JETTA*, vol. 34, no. 5, pp. 571–586, 2018.
[6] Z. Cherif et al., "An Easy-to-Design PUF based on a single oscillator: the Loop PUF," in *DSD*, September 5-8 2012.
[7] S. Morozov et al., "An analysis of delay based PUF implementations on FPGA," in *ARC*, 2010, pp. 382–387.
[8] U. Rührmair and J. Sölter, "PUF modeling attacks: An introduction and overview," in *DATE*, 2014, pp. 1–6.
[9] D. Merli et al., "Side-channel analysis of PUFs and fuzzy extractors," in *Trust and Trustworthy Computing*, 2011, pp. 33–47.
[10] U. Rührmair et al., "Efficient power and timing side channels for physical unclonable functions," in *CHES*, 2014, pp. 476–492.
[11] R. Elnaggar et al., "Machine learning for hardware security: Opportunities and risks," *J. Elect. Test.*, vol. 34, no. 2, pp. 183–201, Apr. 2018.
[12] U. Rührmair et al., "Modeling attacks on physical unclonable functions," in *CCS*, 2010, pp. 237–249.
[13] I. Verbauwhede, *Secure Integrated Circuits and Systems*, ser. Integrated Circuits and Systems. Springer US, 2010.
[14] G. T. Becker and R. Kumar, "Active and passive side-channel attacks on delay based PUF designs," *IACR Cryptology Archive*, p. 287, 2014.
[15] A. Mahmoud et al., "Combined modeling and side channel attacks on strong pufs," *IACR Cryptology ePrint Archive*, vol. 2013, p. 632, 2013.
[16] 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.
[17] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *DAC*, 2007, pp. 9–14.
[18] T. Kroeger et al, "Effect of aging on PUF modeling attacks based on power side-channel observations," in *DATE*, 2020, pp. 454–459.
[19] "Nangate 45nm open cell library," "http://www.nangate.com".
[20] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, 2006.