Assessment and Mitigation of Power Side-Channel-Based Cross-PUF Attacks on Arbiter-PUFs and Their Derivatives

Trevor Kroeger, *Student Member, IEEE*, Wei Cheng, *Student Member, IEEE*, Sylvain Guilley, *Senior Member, IEEE*, Jean-Luc Danger, *Member, IEEE*, and Naghmeh Karimi, *Member, IEEE*

***Abstract*— Unintentional uncontrollable variations in the manufacturing process of integrated circuits are used to realize silicon primitives known as physical unclonable functions (PUFs). These primitives are used to create unique signatures for security purposes. Investigating the vulnerabilities of PUFs is of utmost importance to uphold their usefulness in secure applications. One such investigation includes exploring the susceptibility of PUFs to modeling attacks that aim at extracting the PUFs' behavior. To date, these attacks have mainly focused on a single PUF instance where the targeted PUF is attacked using the model built based on the very same PUF's challenge–response pairs or power side channel. In this article, we move one step forward and introduce *Cross-PUF* attacks where a model is created using the power consumption of one PUF instance to attack another PUF created from the same GDSII file. Through SPICE simulations, we show that these attacks are highly effective in modeling PUF behaviors even in the presence of noise and mismatches in temperature and aging of the PUF used for modeling versus the targeted PUF. To mitigate the *Cross-PUF* attacks, we then propose a lightweight countermeasure based on dual-rail and random initialization logic approaches called DRILL. We show that DRILL is highly effective in thwarting *Cross-PUF* attacks.**

***Index Terms*— Cross-physical unclonable function (PUF) attacks, device aging, modeling attack, power side-channel, PUF.**

I. Introduction

PHYSICAL unclonable functions (PUFs) avoid storing secret keys in digital memory by generating said values on demand, thereby enhancing the security of the integrated circuits (ICs) in which they are instantiated. These primitives generate values unique to each instance [1], which makes them useful for device authentication, or for generating secret keys and random variables in cryptographic devices [2]. Due to their small size and their high resiliency against reverse-engineering attacks, PUFs are well suited for low-cost devices, such as radio frequency identifiers (RFIDs) and smart cards [3]–[5].

PUFs generate unique outputs despite having identical circuit designs due to the random process variations arising from inadvertent technological perturbations [2]. Each PUF instance, even when fabricated from the same blueprint, produces a unique response based on the amplification of imperfections in the manufacturing process.

There is a myriad of security threats that can be addressed via PUFs. An international standard, namely, ISO/IEC 20897, has even been written in this respect [6]. In fact, with the distribution of IC design and manufacturing all over the globe, IC overproduction has become a major threat. To address such a threat, PUFs are utilized to unlock approved devices for regular use [7]. Also, to ensure the integrity of autonomous devices/vehicles, PUFs are being postulated as an authentication mechanism to prevent nefarious activity in the communication and operation of these systems [8]. Due to their small size and unclonability, PUFs have advantages in securing Internet-of-Things (IoT) devices [9], which are highly resource constrained [10]. They are also being considered for securing cryptocurrencies due to their efficacy in generating cryptographic keys [11].

A PUF's signature corresponds to its input and output pairs, known as challenge–response pairs (CRPs). For each PUF, the CRPs are registered during the enrollment phase postfabrication. However, when the PUF is used, in the corresponding reconstruction phase, it is important for the PUF to have a high-reliability level; otherwise, its CRPs can be erroneous due to measurement noise. Thereby, to increase the reliability of PUFs, it is necessary to have a high signal-to-noise ratio (SNR) or perform postprocessing relying on error-correcting codes. SNR can be improved in delay-PUFs [12, Sec. II.B] where n elements are chained, and the total delay of the chain is measured. In this article, we focus on an emblematic type of delay-PUF, the arbiter-PUF, which is broadly studied for device authentication and, most importantly, for master key generation.

Although PUFs are deployed to preserve security and are assumed to be unclonable, even strong PUFs, such as the arbiter-PUF, may be compromised by modeling attacks [13], [14], side-channel attacks [15], or a combination of the two [16], [17]. In traditional modeling attacks, an adversary

Manuscript received June 6, 2021; revised September 14, 2021 and October 25, 2021; accepted November 11, 2021. Date of publication January 4, 2022; date of current version February 7, 2022. This work was supported in part by the bilateral MESRI-BMBF project "APRIORI" from the ANR Cybersecurity 2020 Call, in part by the Horizon 2020 "SPARTA" project under Grant Agreement 830892, and in part by the National Science Foundation Award under Grant 1920079. *(Trevor Kroeger and Wei Cheng contributed equally to this work.) (Corresponding author: Trevor Kroeger.)*

Trevor Kroeger and Naghmeh Karimi are with the Computer Science and Electrical Engineering Department, University of Maryland, Baltimore County, Baltimore, MD 21250 USA (e-mail: trevor.kroeger@umbc.edu; nkarimi@umbc.edu).

Wei Cheng and Jean-Luc Danger are with LTCI, Télécom Paris, Institut Polytechnique de Paris, 75013 Paris, France (e-mail: wei.cheng@telecom-paris.fr; jean-luc.danger@telecom-paristech.fr).

Sylvain Guilley is with LTCI, Télécom Paris, Institut Polytechnique de Paris, 75013 Paris, France, and also with the Think Ahead Business Line, Secure-IC S.A.S., 35510 Cesson-Sévigné, France (e-mail: sylvain.guilley@telecom-paristech.fr).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TVLSI.2021.3129141.

Digital Object Identifier 10.1109/TVLSI.2021.3129141

collects an extensive number of CRPs and uses them to predict the PUF response for other challenges based on statistical methods, including machine learning (ML) techniques [18].

In practice, targeting a PUF using its power traces is of great interest for ML attacks as, once the chip has been enrolled and the response channel is not accessible anymore (generally, this channel is cut by an antifuse), the only *adversarial* way to observe the response is by indirect side-channel captations. 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 in the field with unseen challenges. In a recent research [19], we launched such an attack on a targeted PUF and demonstrated that the target PUF can be successfully modeled using its power traces. We then dove further into power-based modeling attacks, and in particular, we investigated the efficacy of the power side-channel modeling attacks using ML algorithms in the case of the trace misalignments between learned and attack datasets due to different agings of the targeted PUF when it is attacked versus when its power traces were extracted for modeling purposes (during PUF enrollment).

This article explores 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 from another implementation. In other words, we are attacking one PUF using the power traces of a reference PUF implemented from a similar GDSII file. We refer to these attacks as *Cross-PUF* attacks hereafter. Following the observation that these attacks are highly successful, we propose a countermeasure, coined DRILL, which merges dual-rail logic (DRL) and random initialization logic (RIL) to mitigate the effects of these *Cross-PUF* attacks.

When attacking PUFs, the age of the reference PUF and the attacked PUF can be different, which creates a misalignment. In addition, the operating temperature of these two PUFs can be different. Accordingly, this article further investigates the effects that such misalignments have on the success of the attacks and the proposed countermeasures.

The contributions of this article are given as follows:

1) validating the efficacy of power-based modeling attacks (attacking individual PUFs based on their power fingerprints) across temperatures;
2) successful attacks of one PUF based on a model created from a different PUF instance (i.e., *Cross-PUF*);
3) successful *Cross-PUF* attacks in the presence of temperature mismatch;
4) assessment of the susceptibility of the deployed PUFs against power analysis attacks (both individual and *Cross-PUF* attacks) by targeting their arbitration latch or the flip-flop that stores the arbitration result;
5) HSpice MOSRA simulations to evaluate the effect of NBTI and HCI aging mechanisms on the success of *Cross-PUF* modeling attacks in the presence of aging misalignments between the PUFs;
6) evaluation of all said attacking scenarios in the presence of realistic/relatable noise;
7) specification and investigation of countermeasures to protect against *Cross-PUF* attacks, in the presence/absence of the mismatched effects related to temperature and aging.

This article is organized as follows. Section II discusses the backgrounds on aging mechanisms, the targeted PUF instance, and differences from related side-channel attack research. In Section III, we discuss the threat model that motivates this work. The PUF modeling methods are discussed in Section IV, followed by Section V, wherein we discuss the modeling methodology employed in this article. The countermeasures to the *Cross-PUF* attack are proposed in Section VI. Our experimental setup in Section VII is followed by the simulation results presented in Section VIII. Discussions are shared in Section IX. Finally, conclusions and future extensions of this research are drawn in Section X.

II. Preliminary Background

A. Background on Aging

Aging mechanisms result in performance degradation and eventual failure of digital circuits over time. In CMOS technology, the two leading factors of aging are negative bias temperature-instability (NBTI) and hot-carrier injection (HCI) [20]. Both aging sources result in increasing switching voltage and path delays.

1) NBTI Aging: NBTI primarily affects a pMOS transistor when a negative voltage is applied to its gate. The transistor experiences two phases of NBTI depending on its operating condition: the stress phase and the recovery phase. The stress phase occurs when the transistor is ON ($V_{gs} < V_t$). Here, positive interface traps are generated at the Si–SiO_2 interface, which leads to an increase in the threshold voltage of the transistor. When the transistor is OFF ($V_{gs} > V_t$), it enters the recovery phase. The threshold voltage drift that occurred during the stress phase will partially be undone (or be reverted) in the recovery phase. The threshold voltage drift of a pMOS transistor under stress depends on the physical parameters of the transistor, such as the supply voltage, its temperature, and the total time spent in the stress phase. The NBTI effect is high in the first couple of months, but the threshold voltage tends to saturate for long stress times [21].

2) HCI Aging: HCI mainly affects nMOS transistors. HCI occurs when hot carriers are injected into the gate dielectric during transistor switching and remain there. HCI is a function of the amount of switching activity that occurs on the transistor. This switching activity degrades the circuit by shifting the threshold voltage and the drain current of transistors under stress. The HCI threshold voltage drift is highly sensitive to the number of transitions occurring in the gate input of the transistor under stress. In practice, HCI has a sublinear dependence on the clock frequency, usage time, and activity factor of the transistor under stress, where the activity factor represents the ratio of the cycles that the transistor is switching and the total number of cycles the device is utilized. The effect of HCI is dependent on the transistor's operating temperature [20].

B. Background on Arbiter-PUF

The arbiter-PUF is composed of a series of multiplexers that create a top and bottom path from the input, a rising edge, to the output, producing a single bit response for each challenge for each individual single query of the PUF [22]. The structure of the arbiter-PUF is shown in Fig. 1. This PUF takes advantage of the process-variation-induced race between

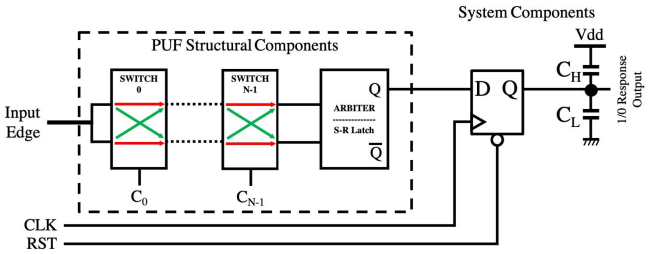

Fig. 1. Structure of an arbiter-PUF [23]. This includes both the PUF structural components and the system components.

the two identical top and bottom paths as the input edge propagates through the switches. The race corresponds to the difference in the delay of these two paths and is adjudicated by the arbiter [23]. 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, at the end of the delay chain, can be realized by an S-R latch implemented through two cross-coupled NAND gates [23].

When embedded in a chip for use, full implementation of the PUF contains a storage mechanism following the PUF's output. This would likely be a flip-flop aiming at registering the response of the PUF for use in downstream components. The capacitors C_H and C_L represent the loading of these downstream components. These leakages play an important role in the overall power consumption of the PUF and affect the total power consumption of the chip [15]. In the following, we will show that the system components create power leakages, which, in fact, reveals the response of the PUF, posing a serious security risk to the arbiter-PUF and its derivatives.

C. Related Work

The literature focuses on ML attacks relying on the knowledge of challenges, responses, and the internal structure of PUFs. The ML attack can be boosted by SCA from exponential to polynomial. For instance, Delvaux and Verbauwhede [24] exploit the PUF reliability, whereas Rührmair *et al.* [17] use the power and the timing to recover the delay model. Even the Interpose PUF [25] that was assumed to be resilient against modeling attacks was defeated later in the literature [26]–[28] using new ML technique and SCA. The difference between these works and the contribution of this article is that the classical modeling attack to get the delay model is not considered here. Our ML application is to model the output DFF behavior, not the internal structure of the PUF. Moreover, we attempt to attack one PUF based on information extracted from another PUF sample while additionally investigating uncontrollable environmental characteristics of the attack, such as operating temperature variation and aging effects. This allows to attack all the PUFs, not a specific one, hence named the *Cross-PUF* attack.

Notably, other hardware security investigations are concerned with side-channel attacks on the registers used of their security-related components. Ring oscillator-based loop PUFs pertain to another popular PUF architecture in which a counter is used to generate the random response of the PUF. Recent work in [29] focused on increasing the randomness of this counter thereby thwarting the side-channel attack.

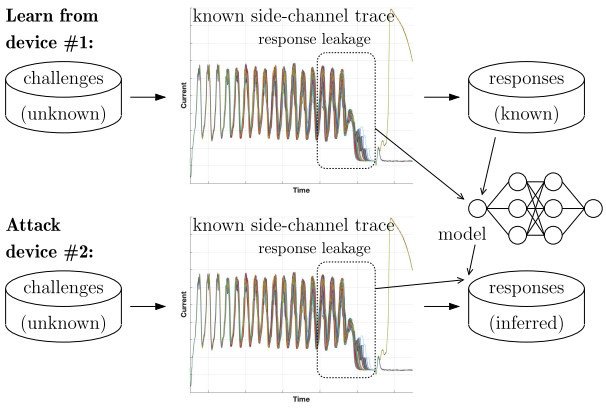

Fig. 2. Illustration of the *Cross-PUF* attack scenario. Top: one PUF is modeled based on known (trace and response) pairs. Bottom: attack consists of leveraging the learned model to infer responses from traces captured on another PUF.

III. Threat Model

PUFs, being an integral part of the security primitives for an IC, are natural targets for adversaries. Indeed, with access to the responses of a target PUF (or intercepting its CRPs), an adversary who aims at breaking the PUF can launch successful modeling attacks allowing for the reconstruction of the PUF's behavior [13], [14]. As a result, the adversary can predict the PUF's responses to any unseen challenges. On the other hand, there are contexts where the attacker does not have access to the PUF responses. The paragon example is the "master key" use case, which is used for the secure boot of root-of-trust chips [30, Sec. 2.2].

In this case, the PUF's responses can notwithstanding be predicted by observing its power side channel [16], [17]. Both of these attacks, proposed previously in the literature, only target one PUF, i.e., the PUF that was used for training the model is the very targeted PUF.

The threat model observed by this work allows for an attacker to monitor the activity of a PUF through its power consumption. The adversary does not have access to the responses of a target PUF; however, he does have access to that of a reference PUF created from the same GDSII file. The attacker, having acquired the PUF from the manufacturer, can assume that it is a valid chip passing required tests (e.g., randomness, unfitness, uniformity, and reliability) to make it to market. That being said, the profile from the storage flip-flop can be used for distinguishing the response (even if the response is biased) as long as the PUF has 0 and 1 responses. This allows the adversary to perform noninvasive "profiling" attacks with a model created from the power traces from the reference PUF used to infer the response of the target PUF. In doing so, the attacker performs the supposed *Cross-PUF* attack. This scenario is illustrated in Fig. 2. The attacked PUF has never been seen before, and the inferred responses can correspond to challenges never encountered either during profiling.

For each PUF instance, the responses for a given challenge are unique; therefore, this form of attack may seem illogical. However, this does not hold true for the power consumption of a PUF. It is observed that the side-channel power consumption from the innate design of the PUF is not inherently unique and, in fact, not only reveals the response of an observed PUF but also characteristic in the responses from PUFs that are

created from the same GDSII file. Therefore, the power traces from one PUF can be used in the creation of a model to infer the response of another PUF without knowledge of the given challenge.

In the investigation of the *Cross-PUF* attack, it is gratuitous to assume that the attacker will not be encumbered by the environmental differences between the reference PUF and the target PUF. We can point to two main such differences. The first is a difference in temperature between the training and target devices, and the second is a difference in the age of these PUFs. Accordingly, in our threat model, we extend the scope of the attack to consider the cases where there is a temperature and/or aging misalignment between these PUFs.

IV. PUF Modeling Schemes

The modeling of PUFs is typically performed through its CRPs. In this form of modeling attack, a large set of CRPs are collected and then used to train a model, so the response from a previously unseen challenge can be predicted [13], [14]. By doing this, the underlying uniqueness of a PUF has revealed such that random mechanism by which the challenge is transformed to a response can be replaced by the model.

Utilizing CRPs for modeling is made more arduous through the difficulty of gaining access to the challenges and their responses; usually, after the device leaves the manufacturing company, the portions of the circuit that would reveal the CRPs of the embedded PUF have already been cut off from being read out, i.e., the adversary would not have access to the CRPs, and therefore, such an attack would not be realistic [31]. Furthermore, the utilization of CRPs is problematic when designers can utilize advanced coding techniques to ensure secrecy of the CRPs [32]. Another issue considers the nature of PUFs, i.e., being unclonable, CRPs are unique for each instance of PUF realized from the same GDSII file. As a result, it is not possible to launch *Cross-PUF* attacks using PUFs' CRPs, while, in this article, we show that *Cross-PUF* attacks are possible through their *power consumption* side channel, i.e., without knowing the applied challenges or their corresponding responses.

Due to the aforementioned reasons, the monitoring and modeling of the power consumption by the PUF turn out to be a more realistic attack on a PUF. From this type of model, the underlying characteristics of the PUF's circuitry can be discerned and then used to infer the PUF's functionality [16], [33], [34].

The power consumption is recorded by monitoring the PUF during the period of time when it is queried with challenges. The recorded traces from the operation are correlated with 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, they can be used to perform *Cross-PUF* attacks, i.e., building a model created on one PUF to attack another PUF realized from the same GDSII file [35].

V. Attack Methodology

There are two components in the arbiter-PUF (as depicted in Fig. 1) that can be targeted for the *Cross-PUF* attacks: the *Latch arbiter* and the *embedded flip-flop*. Fig. 3 shows how the power traces look like when the query propagates through the deployed latch and flip-flop.

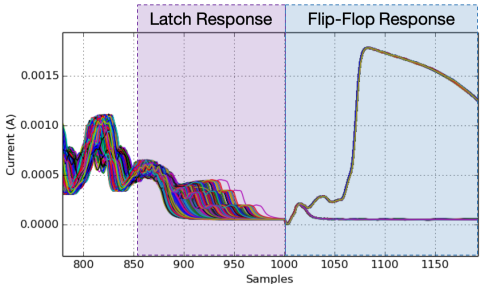

Fig. 3. Highlighted portions of the 12 000 power traces when the latch and flip-flop responses are generated. Each power trace relates to one specific challenge applied to the PUF.

A. Targeting the Latch

The latch is intrinsic to the arbiter-PUF as it carries out the arbitration; therefore, when considering the PUF as a primitive, the leakage of the latch has to be evaluated. As shown in Fig. 3, in the point of time, when the latch is queried, the power traces are highly distinct from each other. In fact, 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.

B. Targeting the Flip-Flop

The flip-flop is extrinsic to the PUF as it depends on the system, which uses the PUF. After the PUF is embedded in a system, the leakage of the flip-flop has also to be assessed. As shown in Fig. 3, 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. However, as we propose in Section VI, system-level countermeasures might reduce or otherwise randomize the leakage of this flip-flop, making it unexploitable. This is the reason why we also study deeply the leakage from the latch stage since this leakage cannot be avoided. Moreover, the latch is part and parcel of the PUF, which could be an IP core/module that is utilized across multiple designs, thus making the attack more portable. An 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.

C. Launching Cross-PUF Attacks

Algorithm 1 explains how the *Cross-PUF* attacks are practically launched. Indeed, it is observed that the *Cross-PUF* attacks are not successful if the leveraged power traces are not aligned [as shown in Fig. 4(a)]. In order to increase the success rate of the attacks, the traces are first shifted appropriately, and then, these aligned traces are targeted. Fig. 4(a) and (b) depicts the power traces of two PUFs (realized from the same GDSII) before and after aligning the latch response, respectively.

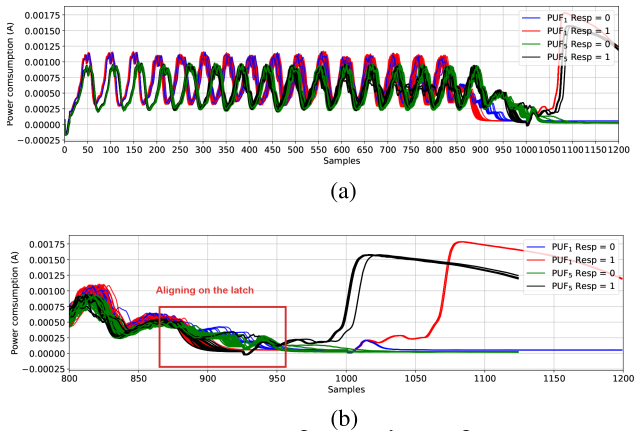

Fig. 4. Superimposing 50 traces of PUF_1 and PUF_5. (a) Full traces of two PUFs. (b) Zoomed-in traces after alignment (by shifting) on the latch.

Algorithm 1 *Cross-PUF* Attacks on the Latch

input : Power traces and related responses for reference PUF (PW_{ref}, R_{ref}) and power traces of the target PUF (PW_{targ})
output: Response bits of the target PUF (R_{targ})

1 PW'_{ref}, $PW'_{targ} \leftarrow$ Select from PW_{ref} and PW_{targ}, the timing windows corresponding to the latch activity
2 Train a model $\mathcal{M}$ by using PW'_{ref} and R_{ref}
3 Shift PW'_{targ} to align it with PW'_{ref}
4 $R_{targ} \leftarrow$ Infer using model $\mathcal{M}$ on PW'_{targ}
5 **return** R_{targ}

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. 4(b) 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. 4(a).

When power traces are used to launch *Cross-PUF* attacks, we need to focus more on the end part of the traces rather than one in their entirety. Thereby, in our attacks, we first select a frame around the point of time in which the latch is queried and only focus on the power traces of both reference and target PUFs in that time frame (line 1 in Algorithm 1). Note that the steps taken for targeting the flip-flop are similar to the steps shown in Algorithm 1 for the latch, except that trace alignment (line 3) is not needed when targeting the flip-flop.

After aligning the traces, we launch the attacks which take advantage of the ML algorithms and consist of two phases: training and evaluation (aka inference). In the training phase, we build the model based on the power traces of the reference PUF and the corresponding responses (line 2 in Algorithm 1). Then, the target traces are shifted in time to be aligned with reference traces (line 3). Finally, in the evaluation phase, unseen inputs (power traces in our case) are tested to investigate whether the model correctly classifies the response (line 4 in Algorithm 1). In this article, we use the support vector machine (SVM) [36], decision tree (DT) [37], and random forest (RF) [38] algorithms to launch the *Cross-PUF* modeling attacks.

VI. Proposed Countermeasure

The proposed countermeasure, DRILL, seeks to reduce the SNR of the power trace leakage from the flip-flop as this is the ideal locale for a modeling attack due to its high current draw during PUF operation [35]. DRILL benefits from deploying dual-rail logic and randomized initialization logic (RIL) to reduce the SNR of the flip-flops' leakage and effectively mitigate the power-based modeling attack. Fig. 5 shows the DRILL countermeasure realized via merging the DRL and RIL techniques.

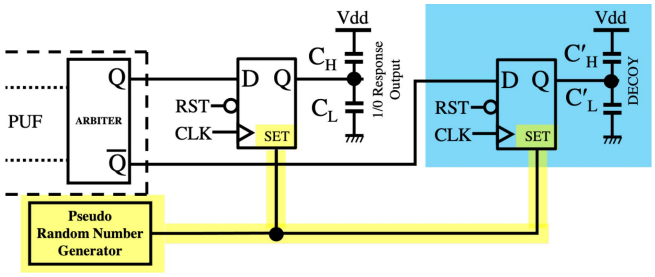

Fig. 5. DRILL countermeasure consisting of DRL (highlighted in blue) and RIL (highlighted in yellow) implemented on a standard arbiter-PUF.

The DRL makes use of two complementary 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 Mangard *et al.* [39, Sec. 7.3]. It is important to note that the loading on the outputs of the flip-flops needs to be balanced to achieve the best protection [40]. As the capacitances are sensitive to process variations, we consider, in our experiments, a great imbalance between capacitances, which is consistent with the worst case scenario of the manufacturing process mismatch. Moreover, the capacitance values must be chosen regarding the relatively high load of the DFF, which is generally a system bus. This leads to choose the following values for the capacitances shown in Fig. 5: $C_H = 200$ fF, $C'_H = 150$ fF, $C_L = 250$ fF, and $C'_L = 200$ fF. On the other hand, 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 [41]. 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 its absence depends on the initial random value of the flip-flop (which is unknown to the adversary) as well.

VII. Experimental Setup

We implemented the targeted arbiter-PUF, as shown in Fig. 1, at the transistor level using a 45-nm technology extracted from the open-source NANGATE library [42]. A 250-fF capacitor was inserted in the PUF's output to demonstrate the load in the unprotected PUF instances.

To investigate the efficacy of *Cross-PUF* attacks, we conducted Monte Carlo transistor-level simulations using

TABLE I
AVERAGE TRAINING TIME (IN SECONDS) USING DATASETS WITH SIZES

ML	Number of power traces exploited during training					
	200	400	600	800	1000	2000
SVM	0.0105	0.0181	0.0330	0.0445	0.0599	0.1645
DT	0.0079	0.0306	0.0568	0.1031	0.1276	0.3779
RF	0.1538	0.2204	0.2811	0.3543	0.4770	0.8753

Synopsys HSpice to realize five PUFs represented as PUF_i, where $i \in \{1, 2, 3, 4, 5\}$. Each simulation was conducted 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 PUF result sets are independently identically distributed. We utilized PUF_1 for the training phase of *Cross-PUF* attacks. The other PUF results are used for validation of the *Cross-PUF* attacks. The HSpice built-in MOSRA Level 3 model [43] was used to capture aging effects. We evaluated the effect of both NBTI and HCI aging for two years of PUF operation in time steps of two months, at the temperature of 80 °C. The 16-stage PUFs were used to perform the analysis in this research. To investigate the effect of temperature misalignments in the success of the *Cross-PUF* attacks, we also simulated one of the designed PUFs at 60 °C. Similar simulations were conducted for the investigation of the proposed DRILL countermeasure.

As already indicated, we deployed three ML algorithms to model each PUF: SVM, DT, and RF. All modeling experiments were performed using a quad-core processor (Intel Core i5-7200U) running at 2.50 GHz with 16 GB of memory. The ML algorithms were implemented with the Python scikit-learn package. The training times of these algorithms are shown in Table I.

1) Data Extraction: Fig. 6(a) shows the timing considered for the data extraction from the targeted PUFs. The entire cycle for querying the PUF is 5 ns. The PUF is fed with a rise transition of 2.5 ns after applying each challenge, starting the response query. The PUF response becomes stable <1 ns after the input edge. In addition, the flip-flop operates at 200 MHz (i.e., clock period: 5 ns) with a rising transition on its clock signal 1 ns after each transition of the PUF input to register the response.

To extract power traces for the attacks, the circuit current is sampled between the time that the PUF is fed with the rise transition and the time that the response becomes stable. Fig. 6(b) shows a set of collected traces, sampled within the aforementioned window.

2) Adding Noise: To be able to account for the noise effects occurring in real silicon experiments, we added artificial noise to the gathered power traces postsimulation. Let X be the original traces and N be the Gaussian noise with four different standard deviations σ, where $\sigma \in \{2.5\text{e}{-4}, 16\text{e}{-4}, 32\text{e}{-4}, 64\text{e}{-4}\}$ to emulate different scenarios. In each case, we add the noises to the original power traces to obtain the noisy traces Y, as shown in the following:

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

The gathered traces after noise insertion are shown in Fig. 7.

To compare the level of the noise added in the experiments conducted in this article with the state-of-the-art research in this area, one can refer to [34], which targets a real arbiter-PUF using its power traces. Similar to [34], the ratio between intervariance and intravariance among the power traces allows us to gauge the susceptibility of the power traces to revealing

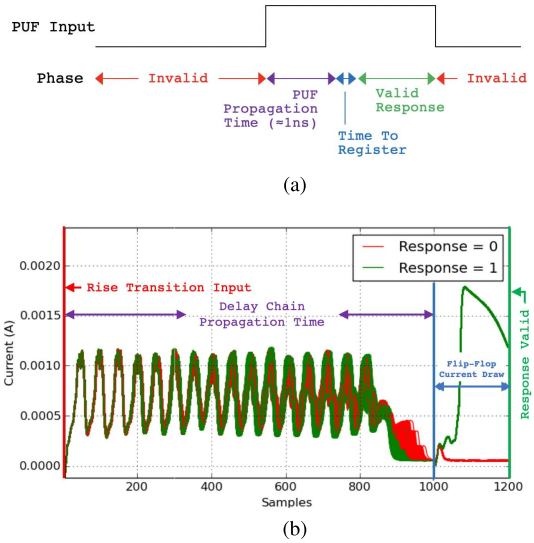

Fig. 6. Timing of the sample window used to create the power traces of the PUFs. (a) Power trace and response sampling window. (b) Power traces collected for the PUFs in the propagation time window.

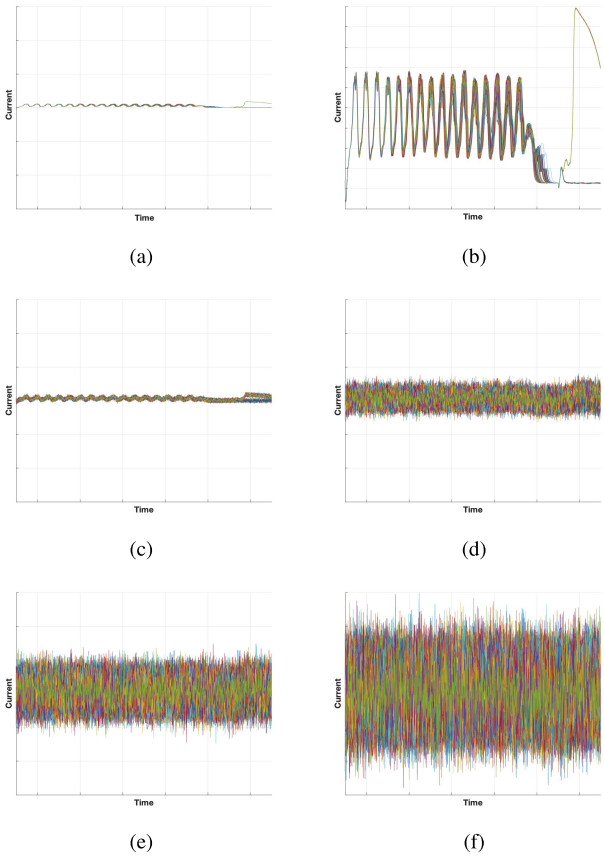

Fig. 7. Power traces (=275) by adding different level of noises. (a) No noise ($\sigma = 0$). (b) Zoom on (a). (c) $\sigma = 2.5e{-}4$. (d) $\sigma = 16e{-}4$. (e) $\sigma = 32e{-}4$. (f) $\sigma = 64e{-}4$.

the response. This ratio is, in fact, the SNR [39, Sec. 4.3.2], which is commonly used in the side-channel analysis.

In order to evaluate the noise level in our experiments, we use SNR to compare with [34]. 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 the two responses of the PUF, i.e., "0" and "1," respectively. 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 detailed comparison of the noise level in this article with [34] is discussed in Section VIII.

3) Modeling Accuracy: The accuracy of the modeling attack, presented in Section VIII, is defined as

$$\text{Accuracy} = \frac{\text{Predicted Correctly}}{\text{Total Tested}}. \quad (2)$$

VIII. Experimental Results

In this section, we present our results and discuss our observations. In the experiments, we target both the latch and the flip-flop shown in Fig. 1 and investigate the efficacy of the attacks when targeting each of these components.

A. Attack Success Rate

1) Self-PUF Attacks: The first set of experiments targets each PUF using its own power traces; we call this the *Self-PUF* attack hereafter. In this set of experiments, the PUFs are new (not-aged). This serves as a baseline for the *Cross-PUF* attack results.

a) Targeting the latch: This set of results investigates the success of the modeling attacks on the targeted PUFs when the attacks are performed through latches, i.e., when the power traces are monitored at the point of time that the latch is queried. Random challenges were given to each PUF; the related PUF responses and power traces were then extracted. The power traces were used to train the models (using the three ML algorithms). Then, the models were tested against 11 000 power traces, and the accuracy was calculated based on the correctness of response prediction. With as few as 40 traces, the targeted PUF could be modeled with high accuracy (≈97%) using the SVM algorithm. The modeling accuracy increased to 99% by using 200 traces for all three algorithms.

In addition, these results showed that the modeling accuracy using each of the three deployed ML algorithms was very close. However, SVM outperformed DTs and RF by around 2% accuracy. Accordingly, we only present the results for applying SVM hereafter.

b) Targeting the flip-flop: Attacking the arbiter-PUF using its own power traces targeting its embedded flip-flop results in 100% accuracy. This can be easy observed in Fig. 6(b). As shown in this figure, the power traces led to responses "0" and "1," which are highly distinct from each other in the point of time, when the flip-flop is queried. Such distinction leads to 100% modeling accuracy. Note that, in this set of experiments, no noise was considered.

The takeaway point from these observations is that an arbiter-PUF can be modeled using its power traces by targeting the point in time in which its arbiter latch is queried or when the underlying flip-flop circuitry is activated. The latter is a stronger attack locale as the power traces are highly dependent on the PUF response when the embedded flip-flop is queried.

TABLE II

Accuracy of *Cross-PUF* Attacks for Each Pair of the Five Implemented PUFs Targeting Their Latch

PUF used for training	# 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
	1000	**0.9998**	0.9705	0.9940	0.9997	0.9619
PUF_2	200	0.9454	**0.9987**	0.9776	0.9517	0.9545
	1000	0.9915	**0.9999**	0.9745	0.9839	0.9349
PUF_3	200	0.9735	0.9997	**0.9983**	0.9775	0.9494
	1000	0.9791	0.9990	**0.9984**	0.9744	0.9449
PUF_4	200	0.9936	0.9700	0.9815	**0.9975**	0.9470
	1000	0.9998	0.9896	0.9903	**1.0000**	0.9637
PUF_5	200	0.9880	0.9951	0.9640	0.9855	**0.9895**
	1000	0.9890	0.9968	0.9785	0.9823	**0.9985**

2) Cross-PUF *Attacks:* In this experiment, a model is built based on the power traces of one PUF and is used to attack other PUFs with the same GDSII file. In the *Cross-PUF* attacks, we again target the sample points, in which either the arbiter latch or the embedded flip-flop has been queried. This threat model is an innovative way to exploit PUFs, which extends the list of threats regarding "physical unclonability" described in Clause 5.5.7 of ISO/IEC 20897-1:2020.

a) Targeting the latch: The results of the *Cross-PUF* attacks when the latch is targeted are shown in Table II. In each experiment, the SVM ML algorithm was used for training a model with 200 and 1000 power traces. Table II demonstrates the *Cross-PUF* attacks accuracy for all PUFs, where one PUF is used as the reference to build the model, and the other PUFs are being targeted. The diagonal of this table shows the *Self-PUF* attacks where each PUF is attacked using its own power traces.

As shown in Table II, the average accuracies of the *Self-PUF* attacks are 99.68% and 99.93% when 200 and 1000 traces are used, respectively, while the attacks accuracies are 97.30% and 97.99% for the *Cross-PUF* attacks with 200 and 1000 traces, respectively. The minimum accuracy for the *Cross-PUF* attacks is ≈93.5%. The takeaway point from this experiment is that *Cross-PUF* attacks can be as strong as *Self-PUF* attacks. This can be a significant threat to the security of devices that are supposed to be secured via PUFs since the adversary can deploy a PUF realized from the same GDSII to break the security of the target PUF, even when the target PUF's response is not observable.

b) 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. 6(b). 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 the arbiter latch or the embedded flip-flop.

B. Attacks' Efficiency in the Presence of Noise

To be able to show the efficacy of the proposed attacks in real silicon experiments, as discussed in Section VII, we artificially added Gaussian noise (with different σ) to the power traces extracted from our HSpice simulations. What

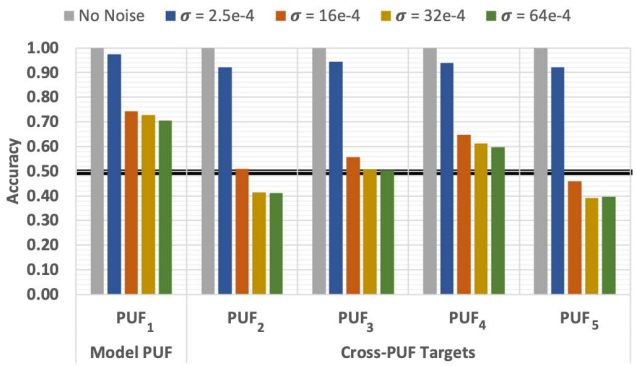

Fig. 8. *Cross-PUF* attacks targeting the arbiter latch in five original PUFs in the presence of different noise levels. PUF_1 was used for training. 2000 power traces were used for training, and 11 000 were deployed for model validation.

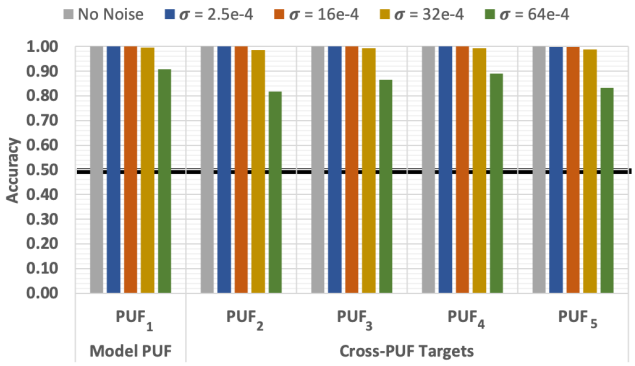

Fig. 9. *Cross-PUF* attacks targeting the embedded flip-flop in the five original PUFs in the presence of different noise levels. PUF_1 was used for training. 2000 power traces were used for training, and 11 000 were deployed for model validation.

follows presents the efficacy of our *Cross-PUF* attacks in the presence of noise.

1) Targeting the Latch: The results of the *Cross-PUF* attacks when the arbiter latch is targeted are shown in Fig. 8. This figure represents the modeling accuracy with different levels of noise. To launch each attack, we trained the model using 2000 traces, and each model was tested against 11 000 traces. Again, we deployed the SVM algorithm and used PUF_1 as the reference PUF. As depicted, the attacks are highly successful when the noise $\sigma = 2.5e{-}4$, i.e., in this case, we obtain 97% accuracy for the *Self-PUF* attacks (attacking PUF_1) and more than 92% accuracy for the *Cross-PUF* attacks. However, the attacks' accuracy for the other noise levels is considerably less. This can be explained via Fig. 7. As shown in that figure, the noise with $\sigma = 2.5e - 4$ is more reasonable, as, with the other noise levels, the power traces themselves are fully concealed by the noise; therefore, the SNR is too low to launch successful *Cross-PUF* attacks. Accordingly, even *Self-PUF* attacks are not possible for those cases. These results show that, in the presence of an acceptable amount of noise (i.e., reasonable SNR), the *Cross-PUF* attacks are still highly accurate.

2) Targeting the Flip-Flop: As discussed earlier, when there is no noise, the embedded flip-flop is a better target for PUF modeling than the arbiter latch. This set of results investigates whether the flip-flops are still better targets than latches in the presence of noise. The attack accuracies are shown in Fig. 9 for the same noise levels in the presence of which we attacked the latches in the previous section. 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 $\sigma = 16e - 4$. With the increase in the noise level to $\sigma = 32e{-}4$, the accuracy drops only marginally to ≈98.5% for both *Self-PUF* and *Cross-PUF* attacks. Increasing the noise even further finally results in a dip when $\sigma = 64e{-}4$. 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% accuracies.

3) Signal-to-Noise Level Comparison: To ensure that the considered noise level is enough, we compared the SNR of our experiments with [34]. The maximum SNR in [34] is estimated to be 1.81, whereas, in our cases, the maximum SNR of PUF_1 is 0.24 for *Targeting the Latch* ($\sigma = 2.5e - 4$) and

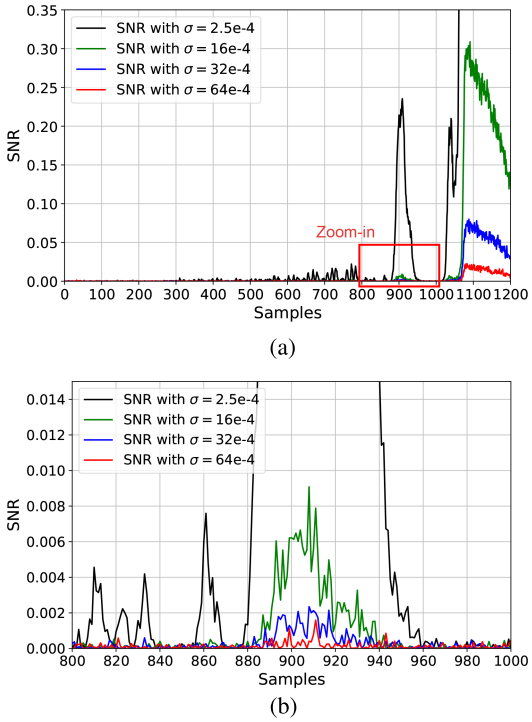

Fig. 10. SNR of the noisy power traces in the original PUF_1 with different noise levels. (a) SNR. (b) Zoom-in at the latch.

0.08 for *Targeting the flip-flop* ($\sigma = 32e - 4$) for the highest noise values in the presence of which the accuracy is above 90%. The maximum SNR for each of the added noise levels in this work, when targeting either the latch or flip-flop, are presented in Table III. The overall SNR for PUF_1 is depicted in Fig. 10. In practice, the SNR in our case is much lower than the one in [34]. This explains the low accuracy of the *Cross-PUF* attacks when targeting the latch in a highly noisy environment.

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).

TABLE III

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

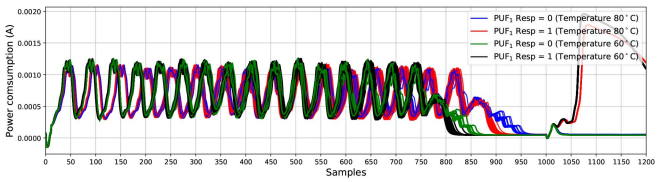

Fig. 11. Superimposing 50 traces of the original PUF_1 under different temperatures to observe the difference in the collected traces.

C. Attacks' Efficiency in the Case of Temperature Misalignment

This set of results demonstrates the accuracy of our modeling attacks when the reference and target PUFs are operating under different temperatures. In this experiment, we consider PUF_1 operating at 60 °C as a reference and target the other five PUFs, including PUF_1, when operating at 80 °C. Fig. 11 shows superimposed traces of PUF_1 operating at different temperatures. As expected, the PUF operates faster at lower temperatures.

1) Targeting the Latch: Fig. 12 shows the effect of temperature misalignments on the attack accuracy when the arbiter latch is targeted. As shown, for the *Self-PUF* attacks, i.e., attacking PUF_1 operating at 80 °C using the model built from the same PUF operating at 60 °C, the PUF can be modeled with 100% accuracy in the case of no noise. The accuracy decreases to 96.11% when Gaussian noise with $\sigma = 2.5e-4$ is added artificially to the power traces. The modeling accuracy of the *Self-PUF* attacks diminishes to 70.84% by increasing σ to $16e-4$. Again, we want to emphasize that the noise levels with $\sigma > 2.5e-4$ result in a very low SNR. In these experiments, the model was trained with 1000 power traces and tested against 11 000 power traces.

The results shown in Fig. 12 confirm that *Cross-PUF* attacks through latches are performed at an accuracy greater than 97% for the SVM algorithm in the case of no noise. When noise is added with $\sigma = 2.5e-4$, the accuracy decreases to 80% for the *Cross-PUF* attacks. Note that the greater the noise level, the lower the accuracy.

2) Targeting the Flip-Flop: The embedded flip-flop is a stronger candidate for attacking even when there is a temperature misalignment between the training and target models. The results of both the *Self-PUF* and *Cross-PUF* attacks targeting the flip-flop are shown in Fig. 13. As depicted, targeting the flip-flop results in 100% accuracy for the *Self-PUF* attacks in the case of no noise or noise with $\sigma \leq 16e-4$ even when there are temperature misalignments between the model and target PUF. The results are very similar for *Cross-PUF* attacks without temperature variation, 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 $\sigma = 64e-4$ for the *Self-PUF* and *Cross-PUF* attacks, respectively.

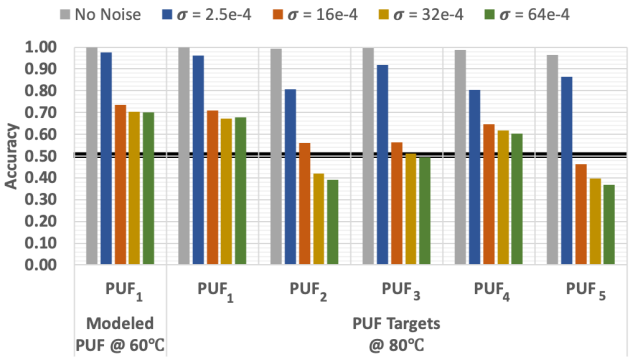

Fig. 12. Modeling results for the *Self-PUF* and *Cross-PUF* attacks on the original PUFs targeting the latch operating at 80 °C. The model was built based on PUF_1 operating at 60 °C.

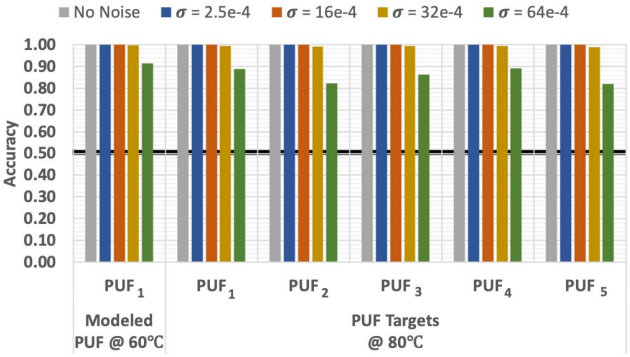

Fig. 13. Modeling results for the *Self-PUF* and *Cross-PUF* attacks on the original PUFs targeting the flip-flop operating at 80 °C. The model was built based on PUF_1 operating at 60 °C.

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

D. Attacks' Efficiency in the Case of Aging Misalignment

This set of results focuses on the aging of the arbiter-PUF and how it affects our ability to model it. In this experiment, the unaged (i.e., age = 0) PUF_1, operating at 80 °C, was used as a reference for modeling, while PUF_2 in different ages ($0 \sim 24$ months' old), operating at the same temperature, was targeted. The training and validation sets included 1000 and 11 000 power traces, respectively.

1) Targeting the Latch: The first observations are made for the attacks targeting the latch. The results are depicted in Fig. 14. As shown, in the case of a low-level noise (i.e., $\sigma = 2.5e-4$), the PUF can be modeled accurately for all aging misalignments, i.e., the accuracy is at least 88.43% across all ages, and the aging-induced accuracy decrease is negligible. However, based on its low accuracy, targeting the latch may not be feasible in cases with higher levels of noise.

2) Targeting the Flip-Flop: Fig. 15 shows the attacks' accuracy for targeting the flip-flop of PUF_2 circuitry, aged

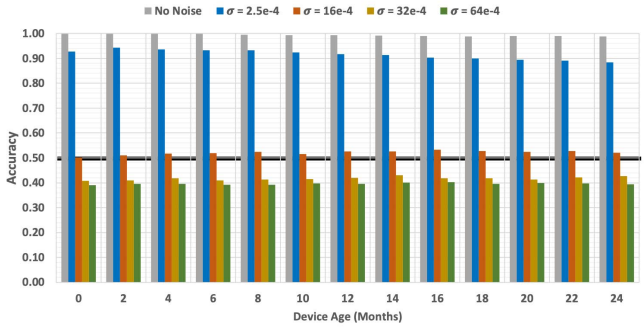

Fig. 14. Modeling accuracy for the *Cross-PUF* attacks on the original PUFs targeting the latch at 80 °C in the presence of aging misalignments. The model was built based on the power traces of the unaged PUF_1 and used to attack PUF_2 operating at the same temperature.

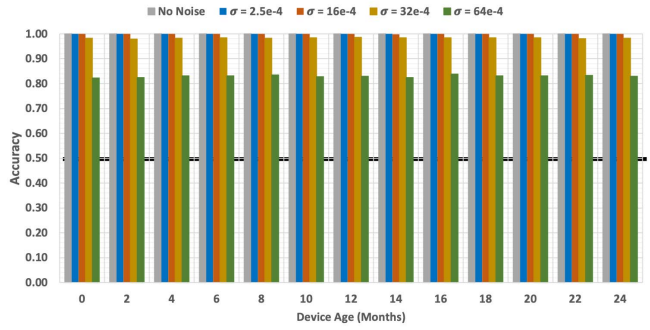

Fig. 15. Modeling accuracy for the *Cross-PUF* attacks on the original PUFs targeting the flip-flop at 80 °C in the presence of aging misalignments. The model was built based on the power traces of the unaged PUF_1 and used to attack PUF_2 operating at the same temperature.

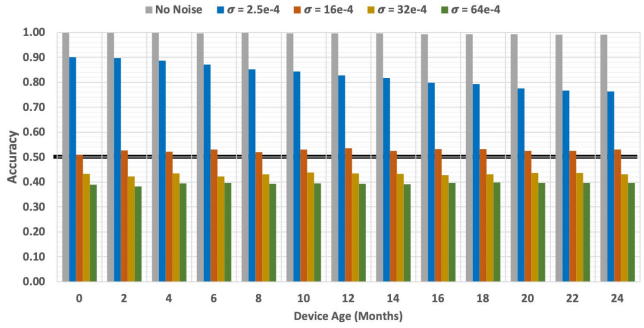

Fig. 16. Modeling accuracy for the *Cross-PUF* attacks on the original PUFs targeting the latch in the presence of both temperature and aging misalignments. The model was built based on the power traces of the unaged PUF_1 in 60 °C and tested based on the traces extracted from the aged PUF_2 operating at 80 °C.

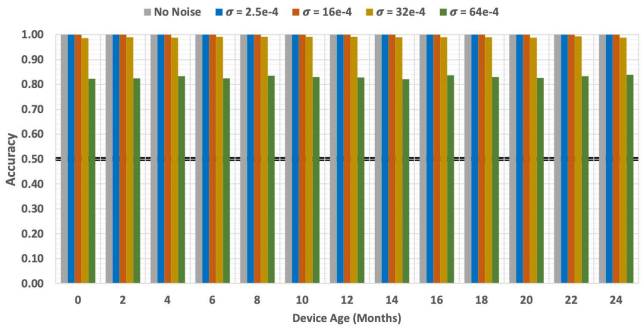

Fig. 17. Modeling accuracy for the *Cross-PUF* attacks on the original PUFs targeting the flip-flop in the presence of both temperature and aging misalignments. The model was built based on the power traces of the unaged PUF_1 in 60 °C and tested based on the traces extracted from the aged PUF_2 operating at 80 °C.

between zero and 24 months and operating at 80 °C. Here, the model was built based on the PUF_1 circuitry operating at the same temperature. As depicted, aging has a negligible effect on the accuracy of the model when targeting the flip-flop even when the noise is at $\sigma = 32e-4$. At this noise level, the accuracy remains almost constant at 98% accuracy over the course of two years.

In another experiment, we used unaged PUF_2 as a reference and aged PUF_1 as the target. The results were very similar to those presented here. The takeaway point from these results is that the misalignment between the age of the training and modeling traces results in slightly higher attack difficulty for the latch (specifically when noise is low). However, there is no noticeable drop in the attack accuracy when targeting the flip-flop.

3) Aging and Temperature Misalignment: To provide a completely realistic scenario for our *Cross-PUF* attacks, this set of experiments deals with the case in which there is both aging and temperature misalignments between the reference and target PUFs. Here, the unaged PUF_1 operating at 60 °C was used as a reference, and the aged PUF_2 operating at 80 °C was targeted. Figs. 16 and 17 demonstrate the attack accuracy when the arbiter latch and the embedded flip-flop were targeted, respectively. These results are very similar to the cases where no temperature misalignment was considered.

When targeting the latch, it can be seen that, in the presence of both aging and temperature misalignment, the decline in accuracy is more pronounced through aging. The results in Fig. 14 show a decline in accuracy of only 5% over 24 months of aging, whereas Fig. 16 shows a decline of 14% for the same aging period when the noise has a $\sigma = 2.5e-4$.

When performing *Cross-PUF* attacks on the flip-flop, the attacks' efficacy is not affected much during the course of aging. Indeed, the accuracy is very close to the no-age *Cross-PUF* attacks. The takeaway point from these observations is that, when the arbiter latch is targeted, the *Cross-PUF* attacks are more difficult in the presence of aging and temperature misalignments compared to the case in which only aging misalignments are observed. However, when targeting the flip-flop, the success rate of our attacks is not diminished significantly even if the target and reference devices operate at different temperatures or have different ages. *This makes the proposed attacks highly applicable according to the considered threat model, where the adversary does not have to take any control of the temperature or age of the target PUF, especially for targeting the flip-flop.*

E. Investigation of Countermeasures

The previous results showed that an arbiter-PUF can be attacked via profiling the traces of a reference PUF realized from the same GDSII. The simulation results clearly confirmed that the flip-flop sampling the arbitration result is the main leakage source compared to the arbitration latch. Accordingly,

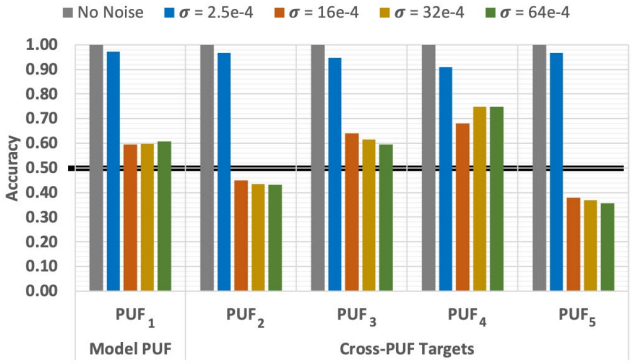

Fig. 18. *Cross-PUF* attacks targeting the flip-flop in five PUFs equipped with the DRILL countermeasure in the presence of different noise levels. 1000 power traces of PUF_1 were used for training. 11 000 power traces were deployed for model validation in each case. All PUFs operate at 80 °C.

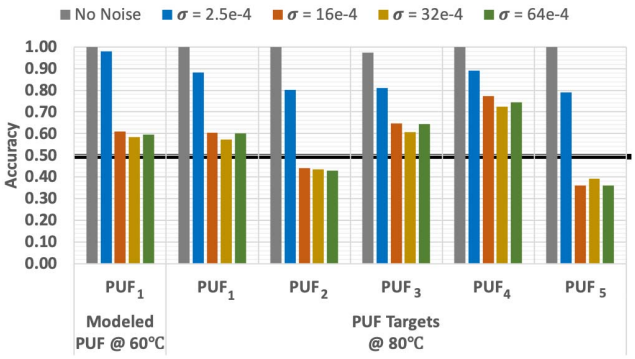

Fig. 19. Modeling results for the *Self-PUF* and *Cross-PUF* attacks targeting the flip-flop for the DRILL protected PUFs operating at 80 °C. The model was built based on PUF_1 operating at 60 °C.

in Section VI, we proposed a countermeasure for mitigating the effects of the *Cross-PUF* attack on the flip-flop. Recall that the goal of the DRILL countermeasure is to reduce the SNR seen in the leakage of the flip-flop by making the transitions or lack of transitions less discernible. The results of implementing the DRILL countermeasure are presented to mirror the investigations of the unprotected PUF.

1) Cross-PUF Attacks on DRILL Protected PUFs: The results of the *Cross-PUF* attacks on the DRILL-protected PUF are shown in Fig. 18. Comparing these results with the ones related to the unprotected PUF, as shown in Fig. 9, reveals that, by using DRILL, the attack accuracy drops significantly when the noise level increases above $\sigma = 2.5e{-}4$. Even with the low levels of noise, the accuracy of the attack is hampered by the countermeasure. At $\sigma = 16e-4$, the level of accuracy drops from being 100% effective to below 60% in most cases. The accuracy of the attack levels off for the higher levels of noise, showing that the model is unable to guess the response when DRILL is used. Also, note that the DRILL countermeasure protects against *Self-PUF* attacks.

The takeaway from these results is that the DRILL countermeasure successfully mitigates both the *Self-PUF* and *Cross-PUF* attacks.

2) Effect of Temperature Mismatch on Cross PUF Attacks Targeting DRILL Protected PUFs: To provide consistency with the previous results, Fig. 19 shows the *Cross-PUF* attack results when there is a temperature mismatch between the modeled PUF and attacked PUF. In this instance, PUF_1 was simulated at 60 °C and used to create the model. This model was used to attack the other PUF instances at 80 °C. Much like previous results, the attack is successfully mitigated, which can be seen by comparing against the results presented in Fig. 13.

When reviewing the results, it is imperative to discuss any bias that the PUFs might have as this affects the model's effectiveness. The following is the percentage of 1's present in the subset of the responses used for training or testing the model: PUF_1:63.82%, PUF_2:39.01%, PUF_3:68.75%, PUF_4:85.56%, and PUF_5:29.90%.

One can see that the bias is not always skewed toward a 1 or a 0. This means that there is no *architectural bias* [44]: our arbiter is an "S-R latch," whose structure is symmetrical. Still, there is *technological dispersion*.

1) This effect makes it possible for each switch to behave differently, depending on whether it is straight (controlled by a 0) or crossed (controlled by a 1), which is leveraged usefully for the sake of the PUF functionality.
2) However, at the same time, the arbiter itself has technological dispersion and, hence, can slightly favor either 1 or 0.

In the results shown in Figs. 18 and 19, one can see the effect of this bias on the model accuracy. First, let us consider PUF_1. When trying to guess the response value of this PUF training on traces captured from itself (*Self-PUF* attack), and when noise is sufficient ($\sigma \geq 16e-4$), the accuracy is not 50%. This does not contradict the soundness of DRILL, though. Indeed, even if the two flip-flops fed by Q and $\bar{Q}$ are randomly reset with probability 50%/50%, the arbiter itself is not masked and is actually biased. In fact, we see that the larger the technological dispersion on the arbiter, the larger its bias (i.e., the more it is unfair since it is unbalanced), and the larger the power difference when it arbitrates 1 or 0. This means that there remains a residual correlation between the response and the power due to the sole technology dispersion at the arbiter level. Such (second order) phenomenon has already been observed previously [45, Fig. 12].

The PUF_3 (resp. 4) is also biased toward 1, by 68.75% (resp. 85.56%); hence, when the attacker guesses them using a (biased, in the same direction) model learned from PUF_1, the accuracy is also slightly greater than $1/2$. The opposite rationale accounts for the slightly less than $1/2$ accuracy for PUF_2 and PUF_5.

3) Effect of Aging Mismatch on Cross PUF Attacks Targeting DRILL Protected PUFs: Fig. 20 shows the results of the *Cross-PUF* attack when there is a mismatch in the age of the PUFs. Similar to the results presented in Fig. 15, an unaged PUF_1 is used to attack an aged PUF_2. Comparing the unprotected results with the DRILL-protected results, it can be seen that the attack is significantly hampered with the accuracy falling to ≈50% when the noise rises above $\sigma = 2.5e-4$. This holds for all ages of the attacked PUF up to two years.

4) Effect of Aging and Temperature Mismatch on Cross PUF Attacks Targeting DRILL Protected PUFs: Much like the previous results, it can be observed in Fig. 21 that the *Cross-PUF* attack accuracy is reduced when realistic noise levels are observed on the power traces of the circuitry (these results can be compared with those of the unprotected PUF

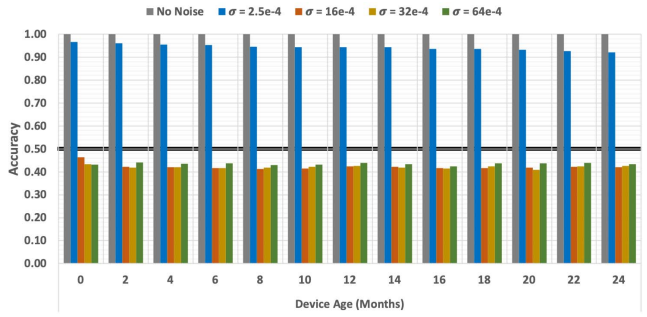

Fig. 20. Modeling accuracy for the *Cross-PUF* attacks targeting the flip-flop of a protected PUF at 80 °C in the presence of aging misalignments. The model was built based on the power traces of the unaged PUF_1 and used to attack PUF_2 operating at the same temperature.

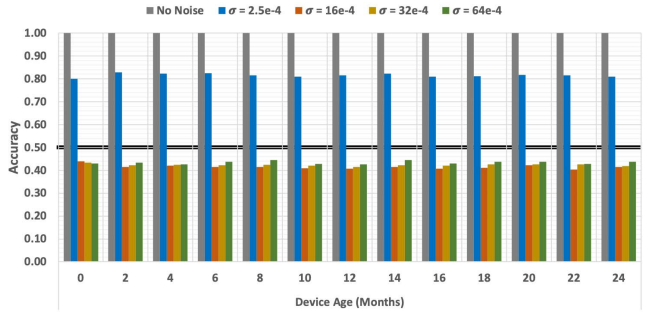

Fig. 21. Modeling accuracy for the *Cross-PUF* attacks targeting the flip-flop of a protected PUF in the presence of both temperature and aging misalignments. The model was built based on the power traces of the unaged PUF_1 in 60 °C and tested based on the traces extracted from the aged PUF_2 operating at 80 °C.

in Fig. 17). At noise levels greater than $\sigma > 2.5e-4$, the accuracy of the attack drops to 50%, showing that the attack is ineffective at classifying the response of the PUF. Also, the accuracy of predicting the response in low noise ($\sigma = 2.5e-4$) is also hindered by the DRILL protection circuitry, which drops the accuracy to ≈80%, as opposed to >95% from the unprotected PUF.

The takeaway from these results is that the DRILL countermeasure successfully mitigates both the *Self-PUF* and *Cross-PUF* attacks. Further investigations into the mismatches of temperature and aging show that the countermeasure is effective even with these eventualities.

IX. Discussion About Protections

The experimental results show that an arbiter-PUF can be attacked via profiling the traces of a reference PUF realized from the same GDSII. Also, note that any kind of arbiter-PUF, even those more robust against modeling attacks (through their CRPs), such as the feedforward PUF or XOR-PUF, can be targeted in a similar way (i.e., via *Cross-PUF* attacks). Note that we are attacking the storage component of these devices, which presents similar behavior for the arbiter-PUF and its derivatives.

As previously stated, the flip-flop presented a significant amount of leakage compared to that of the latch; this makes the flip-flop a more ideal target for inferring the PUF's response. There are several reasons explaining the significant leakage at the flip-flop stage. First, the flip-flop is necessarily connected to the system bus and, thus, more heavily loaded. Second, the output is synchronized with the system clock; hence, there is no need for 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.

It can be stated that the reason for the power modeling attack's success is due to the leakage from the target component. In fact, the balance/imbalance of the loading on the components is critical in the success of performing the modeling attack. In our proposed DRILL countermeasure, if the loading was perfectly balanced, i.e., if, for the four capacitances shown in Fig. 5, we had $C_L = C_H = C'_L = C'_H$, the attack would not be possible since there would be no distinction between the differing power traces. However, perfect balancing on these loads is impossible due to the random variances produced in manufacturing (accordingly our loading is not perfectly equal in the results that we showed). That being said, designers should strive to obtain balancing on these loads to achieve optimum mitigation from the attack.

As already mentioned, the bias that can occur within the PUF is particularly detrimental if the PUFs have "architectural bias" [44], i.e., the PUF may, systematically, have more ones than zeroes, due to a deterministic unbalance between the two paths to arbitrate. However, as shown previously, the PUF instance simulated here does not have bias solely toward one response over the other. The bias caused here is likely in the arbiter embedded in the PUF, as shown in Fig. 1. It can lean toward resolving races in an unfair manner: it can prefer selecting "ones" or "zeros." This is normal, but, actually (opposite to the "constructive" bias in the switches), it is detrimental to the PUF, since it lowers the final entropy of the individual PUF (however, the PUF design as a whole, as previously mentioned, does not skew toward resolving 0's or 1's). Moreover, the bias happens to manifest itself not only in an unbalance in the responses but also in the power it consumes. Thus, a small residual correlation exists between the arbiter's output, and its side-channel leakage. This happens despite the application of the DRILL countermeasure. Still, the countermeasure remains very effective in practice when the noise level is $\sigma \geq 16e-4$. Notice that DRILL protects as against *Cross-PUF* and the more restricted *Self-PUF* scenario.

This article only targets the power side-channel-based attacks and how DRILL protects against such attacks. However, as DRL is susceptible to EM attacks [46], the physical implementations of DRILL should be investigated regarding these EM attacks to assess how RIL affects the effectiveness of EM attacks. Such investigations will be performed in our future work.

X. Conclusion and Perspectives

In this article, the effectiveness of *Cross-PUF* attacks on arbiter-PUFs was explored. These attacks, which utilize the power consumption phenomenon of one PUF to attack another from the same GDSII file, were effective despite variations in temperature and differences in the age of the reference and target PUFs. Furthermore, it can be deduced that the *Cross-PUF* attack can effectively target all the derivatives of the arbiter-PUF (the XOR-PUF, feedforward PUF, and so on). We showed that our proposed DRILL countermeasure,

lightweight addition to the system components storing the PUFs' response, can successfully thwart the *Cross-PUF* attacks.

In the continuation of this work in the future, we will investigate our findings on the *Cross-PUF* attacks in real silicon. We also endeavor to assess the effectiveness of *Cross-PUF* attacks on different PUF targets, as all PUFs need to derive (hence, store, and, subsequently, leak through side channels) response bits. We will also explore the usage of using more advanced ML algorithms to launch the *Cross-PUF* with and the methods that do not require ML. Finally, we opt to investigate if there is a quantitative theoretical evaluation, such as those mentioned in [47], between the power consumption of the PUF design and its susceptibility to the *Cross-PUF* attack.

ACKNOWLEDGMENT

Analysis methods presented in this article are available in Secure-IC's VIRTUALYZR tool [48]. An appendix to this article can be found at https://www.csee.umbc.edu/~nkarimi/papers/karimi_tvlsi_2021.pdf. Contact the co-lead authors Trevor Kroeger and Wei Cheng for inquiries with regards to the code used in this work.

REFERENCES

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, Sep. 2002.

[2] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[3] N. Karimi, J.-L. Danger, and S. Guilley, "Impact of aging on the reliability of delay PUFs," *J. Electron. Test.*, vol. 34, no. 5, pp. 571–586, Oct. 2018.

[4] Z. Cherif, J. Danger, S. Guilley, and L. Bossuet, "An easy-to-design PUF based on a single oscillator: The loop PUF," in *Proc. 15th Euromicro Conf. Digit. Syst. Design*, Izmir, Turkey, Sep. 2012, pp. 156–162.

[5] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and implementation of PUF-based 'Unclonable' RFID ICs for anti-counterfeiting and security applications," in *Proc. IEEE Int. Conf. RFID*, Apr. 2008, pp. 58–64.

[6] N. Bruneau *et al.*, "Development of the unified security requirements of PUFs during the standardization process," in *Proc. SecITC*, Bucharest, Romania, Nov. 2018, pp. 314–330.

[7] A. Cui, X. Qian, G. Qu, and H. Li, "A new active IC metering technique based on locking scan cells," in *Proc. IEEE 26th Asian Test Symp. (ATS)*, Nov. 2017, pp. 40–45.

[8] Q. Jiang, X. Zhang, N. Zhang, Y. Tian, X. Ma, and J. Ma, "Two-factor authentication protocol using physical unclonable function for IoV," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2019, pp. 195–200.

[9] T. Idriss, H. Idriss, and M. Bayoumi, "A PUF-based paradigm for IoT security," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 700–705.

[10] B. Halak, M. Zwolinski, and M. S. Mispan, "Overview of PUF-based hardware security solutions for the Internet of Things," in *Proc. IEEE 59th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Oct. 2016, pp. 1–4.

[11] A. Mars and W. Adi, "New concept for physically-secured E-coins circulations," in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst. (AHS)*, Aug. 2018, pp. 333–338.

[12] A. Schaub, J.-L. Danger, O. Rioul, and S. Guilley, "The big picture of delay-PUF dependability," in *Proc. Eur. Conf. Circuit Theory Design*, Sofia, Bulgaria, Sep. 2020, pp. 1–4.

[13] U. Rührmair and J. Sölter, "PUF modeling attacks: An introduction and overview," in *Proc. DATE*, 2014, pp. 1–6.

[14] U. R Ührmair, F. Sehnke, J. S Ölter, G. Dror, S. Devadas, and J. Ü. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 237–249.

[15] D. Merli et al, "Side-channel analysis of PUFs and fuzzy extractors," in *Trust Trustworthy Computing*. Berlin, Germany: Springer, 2011, pp. 33–47.

[16] A. N. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar, "Combined modeling and side channel attacks on strong PUFs," in *Proc. IACR*, 2013, p. 632.

[17] U. Rührmair, "Efficient power and timing side channels for physical unclonable functions," in *Cryptographic Hardware and Embedded Systems*. Berlin, Germany: Springer, 2014, pp. 476–492.

[18] R. Elnaggar and K. Chakrabarty, "Machine learning for hardware security: Opportunities and risks," *J. Electron. Test.*, vol. 34, no. 2, pp. 183–201, Apr. 2018.

[19] T. Kroeger, W. Cheng, S. Guilley, J. Danger, and N. Karimi, "Effect of aging on PUF modeling attacks based on power side-channel observations," in *Proc. DATE*, 2020, pp. 454–459.

[20] F. Oboril and M. B. Tahoori, "ExtraTime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2012, pp. 1–12.

[21] S. Khan, N. Z. Haron, S. Hamdioui, and F. Catthoor, "NBTI monitoring and design for reliability in nanoscale circuits," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.*, Oct. 2011, pp. 68–76.

[22] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th Annu. Design Autom. Conf.*, 2007, pp. 9–14.

[23] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2002, pp. 148–160.

[24] J. Delvaux and I. Verbauwhede, "Side channel modeling attacks on 65 nm arbiter PUFs exploiting CMOS device noise," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, Dec. 2013, pp. 137–142.

[25] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. Van Dijk, "The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2019, no. 4, pp. 243–290, Aug. 2019.

[26] A. Aghaie and A. Moradi, "TI-PUF: Toward side-channel resistant physical unclonable functions," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3470–3481, 2020.

[27] N. Wisiol *et al.*, "Splitting the interpose PUF: A novel modeling attack strategy," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, vol. 2020, no. 3, pp. 97–120, Jun. 2020.

[28] D. Chatterjee, D. Mukhopadhyay, and A. Hazra, "Interpose PUF can be PAC Learned," in *Proc. IACR*, 2020, p. 471.

[29] L. Tebelmann, "Self-secured PUF: Protecting the loop PUF by masking," in *Constructive Side-Channel Analysis Secure Design*. Cham, Switzerland: Springer, 2021, pp. 293–314.

[30] J.-L. Danger, S. Guilley, M. Pehl, S. Senni, and Y. Souissi, "Highly reliable PUFs for embedded systems, protected against tampering," in *Industrial Networks and Intelligent Systems*, N.-S. Vo, V.-P. Hoang, and Q.-T. Vien, Eds. Cham, Switzerland: Springer, 2021, pp. 167–184.

[31] F.-X. Standaert, "Introduction to side-channel attacks secure integrated circuits and systems," in *Secure Integrated Circuits and Systems* (Integrated Circuits and Systems), I. M. R. Verbauwhede, Ed. Boston, MA, USA: Springer, 2010, ch. 2, pp. 27–42.

[32] O. Günlü and R. F. Schaefer, "An optimality summary: Secret key agreement with physical unclonable functions," *Entropy*, vol. 23, no. 1, p. 16, Dec. 2020.

[33] G. T. Becker and R. Kumar, "Active and passive side-channel attacks on delay based PUF designs," in *Proc. IACR*, 2014, p. 287.

[34] 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 *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 201–207.

[35] T. Kroeger, W. Cheng, S. Guilley, J. Danger, and N. Karimi, "Cross-PUF attacks on arbiter-PUFs through their power side-channel," in *Proc. Int. Test Conf. (ITC)*, 2020, pp. 1–5.

[36] S. Yue *et al.*, "SVM classification: Its contents and challenges," *A. Math. J. Chin. Univ.*, vol. 18, no. 3, pp. 332–342, Sep. 2003.

[37] S. B. Kotsiantis, "Decision trees: A recent overview," *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 261–283, 2013.

[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[39] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards* (Advances in Information Security). Secaucus, NJ, USA: Springer-Verlag, 2007.

[40] T. Kroeger, W. Cheng, S. Guilley, J. Danger, and N. Karimi, "Making obfuscated PUFs secure against power side-channel based modeling attacks," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, 2021, pp. 1000–1005.
[41] L. Tebelmann, J.-L. Danger, and M. Pehl, "Self-secured PUF: Protecting the loop PUF by masking," in *Proc. Construct. Side-Channel Anal. Secure Design (COSADE)*, vol. 12244, 2020, pp. 293–314.
[42] *Nangate 45 nm Open Cell Library*. Accessed: May 6, 2021. [Online]. Available: http://www.nangate.com
[43] *HSPICE User Guide: Basic Simulation and Analysis*, Synopsys, Mountain View, CA, USA, 2016.
[44] D. P. Sahoo, P. H. Nguyen, and R. S. Chakraborty, "Architectural bias: A novel statistical metric to evaluate arbiter PUF variants," Cryptol. ePrint Arch., Tech. Rep. 2016/057, 2016. [Online]. Available: http://eprint.iacr.org/2016/057
[45] N. Karimi, J.-L. Danger, F. Lozach, and S. Guilley, "Predictive aging of reliability of two delay PUFs," in *Proc. Secur., Privacy, Appl. Cryptogr. Eng.*, 2016, pp. 213–232.
[46] V. Immler, "Your rails cannot hide from localized EM: How dual-rail logic fails on FPGAs," in *Proc. Int. Conf. Cryptograph. Hardw. Embedded Syst.*, Aug. 2017, pp. 403–424.
[47] M. Bloch *et al.*, "An overview of information-theoretic security and privacy: Metrics, limits and applications," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 5–22, Mar. 2021.
[48] Secure-IC. *Virtualyzr Tool (VTZ)*. Accessed: Jun. 21, 2017. [Online]. Available: http://www.secure-ic.com/solutions/virtualyzr/

Trevor Kroeger (Student Member, IEEE) received the B.S. degree in computer engineering from the University of Denver, Denver, CO, USA, in 2011, and the M.S. degree in cybersecurity from New York University, New York, NY, USA, in 2015. He is currently working toward the Ph.D. degree in computer engineering at the University of Maryland, Baltimore County, Baltimore, MD, USA, with a focus on hardware security specifically the vulnerabilities of physical unclonable functions (PUFs).

He has worked in the industry designing field-programmable gate arrays (FPGAs) for aerospace applications and developed system-on-chip architectures for distributed nodes in large-scale system frameworks. He also performs communications propagation, clutter analysis, and data collection from unmanned systems.

Wei Cheng (Student Member, IEEE) received the B.S. degree in software engineering from Wuhan University, Wuhan, China, in 2014, and the M.E. degree in computer technology from the Institute of Information Engineering, Chinese Academy of Sciences (also the University of Chinese Academy of Sciences), Beijing, China, in July 2017. He is currently working toward the Ph.D. degree at Télécom Paris, Institut Polytechnique de Paris, Paris, France.

His research interests include information theory, side-channel analysis, and related countermeasures (e.g., code-based masking including inner product masking and direct sum masking and some variants) of embedded systems and implementations. He also works on machine learning-based analysis on physical unclonable functions (PUFs).

Sylvain Guilley (Senior Member, IEEE) is currently the General Manager and the CTO with Secure-IC S.A.S., Cesson-Sévigné, France, a company offering security for connected embedded systems. Secure-IC's flagship technology is the multicertified SECURYZR integrated Secure Element (iSE). He is also a Professor with Télécom Paris, Institut Polytechnique de Paris, Paris, France, an Associate Research with the École Normale Supérieure (ENS), Paris, and an Adjunct Professor with the Chinese Academy of Sciences (CAS), Beijing, China. He is the "High Level Principles for Design/Architecture" Team Leader for the drafting of the Singapore TR68 Standard on Cyber-Security of Autonomous Vehicles. He has coauthored more than 250 research papers and filed more than 40 patents. His research interests are trusted computing, cyber–physical security, secure prototyping in field-programmable gate array (FPGA) and application-specific integrated circuit (ASIC), and formal/mathematical methods.

Dr. Guilley is also a member of the International Association for Cryptologic Research (IACR) and a Senior Member of the CryptArchi club. Since 2012, he has been organizing the PROOFS Workshop, which brings together researchers whose objective is to increase the trust in the security of embedded systems. He is also the Lead Editor of international standards, such as ISO/IEC 20897 (Physically Unclonable Functions), ISO/IEC 20085 (Calibration of Noninvasive Testing Tools), and ISO/IEC 24485 (White Box Cryptography). He is also an Associate Editor of the *Journal of Cryptographic Engineering* (JCEN) (Springer). He is an alumni of the École Polytechnique and Télécom Paris.

Jean-Luc Danger (Member, IEEE) received the Engineering degree in electrical engineering from the École Supérieure d'Électricité, Gif-sur-Yvette, France, in 1981.

After 12 years in industrial laboratories (Philips, France, and Nokia, France), he joined Télécom Paris, Paris, France, in 1993, where he became a Full Professor in 2002. He is currently a Full Professor with Télécom Paris. He is the head of the digital electronic system research team involved in the research in security/safety of embedded systems, configurable architectures, and implementation of complex algorithms in application-specific integrated circuits (ASICs) or field-programmable gate arrays (FPGAs). He is also the Co-Founder and a Scientific Advisor of Secure-IC S.A.S., Cesson-Sévigné, France. He has authored more than 250 scientific publications and patents in architectures of embedded systems and security. His personal research interests are trusted computing, cybersecurity, random number generation, and protected implementations in novel technologies.

Naghmeh Karimi (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer engineering from the University of Tehran, Tehran, Iran, in 1997, 2002, and 2010, respectively.

She was a Visiting Researcher with Yale University, New Haven, CT, USA, from 2007 to 2009, and a Post-Doctoral Researcher with Duke University, Durham, NC, USA, from 2011 to 2012. She has been a Visiting Assistant Professor with New York University, New York, NY, USA, and Rutgers University, New Brunswick, NJ, USA, from 2012 to 2016. She joined the University of Maryland, Baltimore County, Baltimore, MD, USA, as an Assistant Professor, in 2017, where she leads the SECure, REliable and Trusted Systems (SECRETS) Research Lab. She has published three book chapters and authored/coauthored more than 70 papers in refereed conference proceedings and journal manuscripts. Her current research interests include hardware security, VLSI testing, design-for-trust, design-for-testability, and design-for-reliability.

Dr. Karimi was a recipient of the National Science Foundation CAREER Award in 2020. She also serves as an Associate Editor for the *Journal of Electronic Testing: Theory and Applications* (JETTA) (Springer). She is also the Corresponding Guest Editor of the IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS (JETCAS), special issue in Hardware Security in Emerging Technologies.