Physical Unclonable Functions (PUFs): Foundations, Evaluation, and Testing for Secure Hardware Systems

Sergio Vinagrero Gutierrez, Giorgio Di Natale, Ioana Vatajelu

2025 30th IEEE European Test Symposium (ETS)
- Embedded Tutorial Paper -

Physical Unclonable Functions (PUFs): Foundations, Evaluation, and Testing for Secure Hardware Systems

Sergio Vinagrero Gutierrez, Giorgio Di Natale, Elena-Ioana Vatajelu

Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMA, 38000 Grenoble, France
{firstname.lastname}@univ-grenoble-alpes.fr

***Abstract*—Physical Unclonable Functions (PUFs) leverage inherent physical variations in semiconductor devices to generate unique, unpredictable, and unclonable responses, offering promising solutions for secure authentication, key generation, and hardware security. This tutorial provides an in-depth exploration of PUFs, covering their definitions, designs on ASIC and FPGA platforms, and critical evaluation metrics such as uniqueness, reliability, and entropy. Additionally, we address the tradeoffs between reliability and entropy, a fundamental challenge in PUF design, and practical PUF testing methodologies. The content is tailored to the needs of the European Test Symposium (ETS) community, with a focus on testing challenges, and secure deployment in modern applications. Attendees will gain a comprehensive understanding of PUFs potential in advancing hardware security and test technologies.**

***Index Terms*—device fingerprinting, ring oscillator, reliability, entropy**

I. Introduction

The security of embedded systems has become increasingly critical due to the globalization of the semiconductor industry and the pervasive deployment of connected devices. In this context, Physical Unclonable Functions (PUFs) have emerged as a lightweight primitive for securing integrated circuits (ICs) without the need for expensive, tamper-resistant hardware.

A Physical Unclonable Function (PUF) can be viewed as a function $f(c)$ [1], where (c) is an input called the challenge, and the output $r = f(c)$ is known as the response. Unlike conventional functions implemented in software or deterministic hardware, the behavior of a PUF is determined by the microscopic physical variations that inevitably occur during semiconductor manufacturing. These small, random variations (for example, in transistor dimensions, doping concentrations, or interconnect delays) are not fully controllable, even with the same design and fabrication process. As a result, each individual device exhibits unique physical characteristics, making the mapping between challenges and responses slightly different from one chip to another.

The primary advantage of PUFs stems from this physical uniqueness: even if an adversary has complete access to the circuit's design, the exact physical behavior (and therefore the set of responses generated) cannot be perfectly copied or emulated. In other words, PUFs are physically unclonable because it is practically impossible to manufacture two identical instances, and reproducing the device's responses without direct physical access becomes infeasible. This makes PUFs particularly attractive for applications requiring secure identification, key generation, and authentication.

PUFs are generally classified into two major types:

- Weak PUFs: A weak PUF is a type of PUF that can produce only a limited number of challenge-response pairs (CRPs). Typically, a weak PUF is designed to generate a fixed response (or a small set of responses) when queried. Because of this, the number of distinct challenges that can be applied to the device is small. Weak PUFs are primarily used to generate cryptographic keys or unique identifiers for devices. In these applications, it is crucial that the response:
  - Is highly reliable, meaning it can be reproduced identically even when the device is subjected to environmental variations like changes in temperature or supply voltage,
  - Remains secret, because if the response is leaked, the security of the system could be compromised.

  A classic example of a weak PUF is the SRAM PUF, where the initial power-up state of an SRAM memory block is used as a device-specific fingerprint. In short, weak PUFs are well-suited for applications where a device needs to "self-generate" a secure key internally, without having to store it permanently in memory.
- Strong PUFs: they are capable of producing a very large number of challenge-response pairs, typically making it infeasible for an attacker to exhaustively query and store all responses. The key idea behind strong PUFs is that different challenges should produce unpredictable responses, and even after observing many CRPs, it should remain extremely difficult to predict the response to a new, unseen challenge. Because of these properties, strong PUFs are particularly well-suited for device authentication protocols. In such protocols, during the enrollment phase, each strong PUF is interrogated with a large number of randomly selected challenges. The corresponding responses are measured and stored securely on a trusted server, forming a unique challenge-response database for each device. At deployment, authentication is performed by selecting a subset of these stored challenges and issuing them to the device. The device generates responses using

979-8-3315-9450-3/25/$31.00 ©2025 IEEE

its PUF, and the server compares them with the enrolled values. Since PUF responses can exhibit slight variability due to environmental noise or aging, the matching algorithm is typically designed to tolerate a small number of bit errors—ensuring reliable authentication without requiring perfect response reproduction. This error tolerance is a key enabler for practical use of strong PUFs in real-world, noisy conditions.

Beyond traditional applications like key generation and authentication, PUFs have found new roles in lifecycle management of electronic devices. For example, they can help detect recycled or counterfeit ICs, a growing problem in supply chains, by observing changes in a PUF's behavior over time due to aging phenomena. However, strong PUFs face specific security challenges, notably the risk of machine learning attacks: if the challenge-response behavior can be learned from a large enough sample, the PUF can be modeled and cloned. An example of a strong PUF architecture is the Arbiter PUF, where the difference in delay between two signal paths determines the output response.

The effectiveness and security of a PUF are determined by several key characteristics. Each characteristic plays an important role, depending on the application (e.g., key generation, authentication, hardware fingerprinting). Let's briefly go through the main properties:

- Physical Unclonability: A PUF is, by design, physically impossible to clone. Even with complete access to the manufacturing process and the device layout, it is practically impossible to fabricate another device that produces identical responses. This fundamental property stems from uncontrollable variations in the physical structure of each chip, which makes each PUF instance unique at the microscopic level. Physical unclonability is what distinguishes PUFs from traditional stored secrets: the "secret" is not stored explicitly — it is embedded in the hardware itself.
- Uniqueness: Uniqueness measures how different the responses of different PUF instances are when given the same challenge. Ideally, if two chips are asked the same challenge, their responses should be completely independent and different. In practice, this is evaluated by measuring the inter-chip Hamming Distance: the more it tends toward 50% difference, the better. High uniqueness ensures that each device can be individually identified based solely on its PUF responses.
- Reliability: it refers to the PUF's ability to consistently produce the same response when a given challenge is applied, even under varying environmental conditions such as changes in temperature, supply voltage, or device aging over time [2], [3]. Without high reliability, the PUF would produce unstable responses, making it unusable for applications like key generation or secure authentication.
- Randomness (or Entropy): it relates to the distribution of the response bits. For security purposes, the bits produced by the PUF should appear random and unbiased — meaning that the number of 0s and 1s in the responses should be close to equal. High randomness (or high entropy) guarantees that the responses do not reveal patterns that could be exploited by attackers to predict or model the PUF behavior.
- Resistance to Modeling Attacks: Especially for strong PUFs, it is crucial that the mapping from challenges to responses is hard to learn. Even if an attacker can collect a large number of CRPs, they should not be able to build a predictive model that can reproduce the behavior of the PUF. This resistance is critical to defend against machine learning-based attacks, which have proven effective against poorly designed PUFs.

The objective of this tutorial paper is to present an accessible overview of PUFs, covering the essential concepts, metrics for evaluation, challenges like reliability and machine learning attacks, the threat of physical attacks, methods for testing PUF quality, and to offer perspectives on their evolving applications in embedded systems.

The structure of this paper is organized as follows. Section II provides an overview of existing PUF architectures, highlighting their principles, strengths, and limitations. Section III introduces and defines the key metrics used to evaluate PUF quality, such as reliability, uniqueness, and entropy. Section IV discusses the challenges of ensuring both high reliability and high entropy in weak PUFs, and reviews existing approaches and filtering techniques designed to address these challenges. Section V focuses on strong PUFs, explaining how machine learning techniques threaten their security, and presenting countermeasures to resist modeling attacks. Section VI addresses the vulnerability of PUFs to physical attacks, including invasive and non-invasive techniques. Section VII explores the specific problem of testing and evaluating PUFs, both during manufacturing and in mission mode, to guarantee long-term performance. Finally, Section VIII concludes the paper by summarizing the key findings and discussing future research directions.

II. PUF Architectures

Over the past two decades, various families of Physical Unclonable Functions (PUFs) have been proposed. Each architecture relies on different physical phenomena to generate unique and unpredictable responses, and each comes with its own advantages and limitations. In this section, we review the most important PUF types commonly used in embedded systems, focusing on their working principles and practical considerations.

A. SRAM PUFs

SRAM PUFs [4] exploit the behavior of standard Static Random-Access Memory (SRAM) cells at power-up. When an SRAM chip is powered on, each memory cell — consisting of two cross-coupled inverters — spontaneously stabilizes to either a logic '0' or a logic '1'. Due to tiny manufacturing variations, each cell has a slight imbalance that causes it to prefer one state over the other. By capturing the power-up state of a block of SRAM cells, we obtain a device-unique fingerprint.

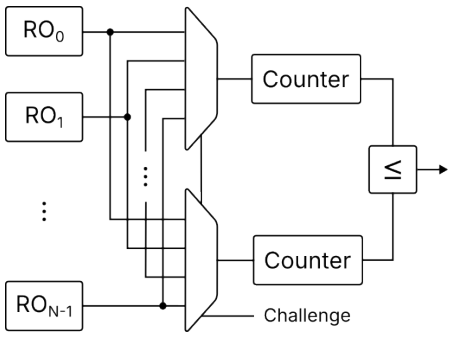

Fig. 1: Basic architecture of a Ring-Oscillator (RO) PUF

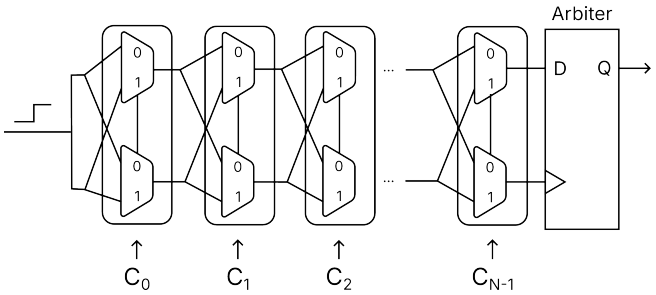

Fig. 2: Basic Architecture of an Arbiter PUF

SRAM PUFs have several important properties:

- They are easy to integrate into existing designs, as SRAM is already a common component.
- They are naturally compact and require no additional circuitry.
- However, their reliability can be sensitive to environmental conditions like temperature or aging, requiring error correction techniques (e.g., fuzzy extractors) to ensure stable responses.

SRAM PUFs are widely used for cryptographic key generation and device identification.

B. Ring Oscillator (RO) PUFs

Ring Oscillator PUFs [5] rely on slight differences in the oscillation frequency of identical ring oscillators implemented on the same chip. A ring oscillator is a circuit made by connecting an odd number of inverters in a loop, allowing a signal to oscillate back and forth indefinitely.

In an RO PUF, multiple ring oscillators are placed side-by-side. Due to manufacturing variations, each oscillator exhibits a slightly different frequency. By comparing the frequencies of two selected oscillators — for example, assigning a '1' if one is faster, or a '0' otherwise — a sequence of bits can be generated.

Key points about RO PUFs:

- They are relatively robust and easy to implement on both ASICs and FPGAs.
- They allow for many different challenge configurations, depending on how oscillators are selected and compared.
- However, they can be affected by environmental noise, and their evaluation time can be relatively high because oscillation frequency must be measured accurately.

The number of available challenge-response pairs grows only quadratically with the number of oscillators (specifically, proportional to $N(N-1)/2$, where N is the number of oscillators, in case any RO can be compared to any other RO). As a result, RO PUFs are not suitable for strong authentication protocols, where a massive number of unpredictable CRPs is necessary.

C. Arbiter PUFs

Arbiter PUFs [6] use signal delay variations in a carefully structured circuit to generate responses. In a typical arbiter PUF, a signal is split into two identical paths that traverse a set of switching elements (multiplexers) controlled by the challenge bits. The two signals "race" through the paths, and an arbiter (a simple latch) determines which signal arrives first, producing a binary response.

Highlights of Arbiter PUFs:

- They can produce a large number of challenge-response pairs, making them ideal candidates for strong PUF applications.
- They are vulnerable to modeling attacks using machine learning if not properly hardened, as the relation between challenge and response is a composition of (often) linear effects based on a small number of elements.

Because of this vulnerability, more complex variants like XOR Arbiter PUFs and Feed-Forward Arbiter PUFs have been developed to enhance security.

D. Other Notable PUF Types

While SRAM, RO, and Arbiter PUFs are the most widely adopted in practice, other types also exist:

- Optical PUFs [1]: These use light scattering through a medium with random microstructure. Although very secure, they are bulky and not suitable for embedded devices.
- Coating PUFs [7]: Random patterns in coating layers are used to generate responses. These are useful for tamper-detection applications.
- Emerging Memories-based PUFs: Emerging approaches use the natural variability in the behavior of MRAMs [8] or memristor [9] devices to create new forms of PUFs with potential for high-density integration.

III. Metrics

The quality of PUFs is commonly assessed through a set of canonical metrics that evaluate both the repeatability and the randomness of the generated responses. These metrics include Reliability, which measures the stability of a response over time under varying environmental conditions, and Uniformity, Bit-aliasing and Uniqueness, which assess the statistical distribution of response values across challenges and devices. Collectively, these metrics aim to provide a comprehensive overview of a PUF's performance.

To support a more intuitive understanding of how these metrics are computed, Figure 3 presents a conceptual diagram based on a 3D array structure of PUF responses. In this standard

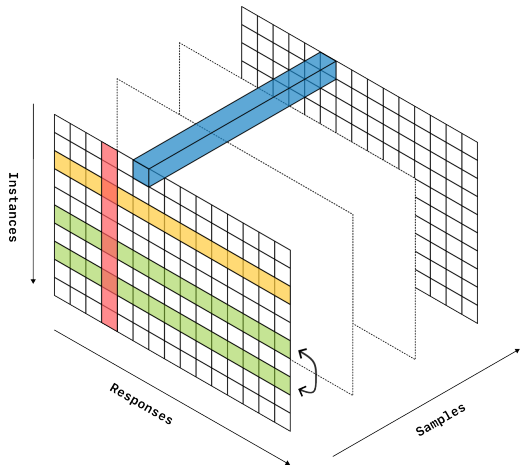

Fig. 3: Diagram representing how the canonical metrics are computed given the PUF responses. Red corresponds to Bit-aliasing, yellow to Uniformity, green to Uniqueness and blue to Reliability

representation, each cell corresponds to a single binary response (either 0 or 1). Each row in the array represents a different device or instance, each column corresponds to a different challenge, and each slice consists of repeated measurements over time.

Reliability is defined as the degree to which a PUF reproduces the same output when the same challenge is applied under different operating conditions, such as changes in temperature, voltage, or ageing. A high Reliability value indicates that the PUF consistently produces the same response and is therefore a desirable property. This metric is computed by comparing each measurement of a device's response to a reference response, typically defined through majority voting. Specifically, for each challenge, the proportion of bit-flips across repeated measurements is calculated, and this is then averaged across all challenges for that device. The process is illustrated in blue in Figure 3, and the formal definition is provided in Equation 1.

In this formulation, T represents the number of repeated measurements, n is the number of total responses, R_{ref} is the set of reference responses, and R_t is the set of t-th measurements. The Hamming distance between vectors, denoted as $HD(\cdot,\cdot)$ in the equation, quantifies how many bits differ between them. Reliability reaches its maximum value of 100% when there are no bit-flips across any measurements.

$$Reliability(d) = 100\% - \frac{1}{T}\sum_{t=1}^{T}\frac{HD(R^d_{ref}, R^d_t)}{n} \times 100\% \quad (1)$$

While Reliability focuses on temporal behaviour, Bit-aliasing, Uniformity and Uniqueness examine the statistical distribution of response bits to assess the randomness of a PUF. Bit-aliasing is computed by averaging the response values across devices for each challenge. Bit-aliasing reaches it's ideal value of 50% when the distribution of 1s and 0s is uniform. The computation of Bit-aliasing is performed per challenge, yielding a distribution across all challenges, and is depicted in red in Figure 3. Equation 2 defines the Bit-aliasing formally, where D represents the number of devices.

$$\text{Bit-Aliasing}(c) = \frac{1}{D}\sum_{d=1}^{D} R^d_c \times 100\% \quad (2)$$

Uniformity, on the other hand, evaluates the average response value for each device. Like Bit-aliasing, the ideal Uniformity value is also 50%, indicating a uniform distribution of 1s and 0s. This metric is computed for each device, resulting in a distribution across devices. The yellow region in Figure 3 highlights where this computation is performed. Equation 3 provides the formula for Uniformity, where R^d_c is the response of device d to challenge c.

$$\text{Uniformity}(d) = \frac{1}{n}\sum_{c=1}^{n} R^d_c \times 100\% \quad (3)$$

Uniqueness measures a PUF's capacity to distinguish one device from another within a population. This characteristic is essential, as it sets the limit on the number of uniquely identifiable devices in a system. The metric is calculated by exhaustively comparing the reference responses of each device with those of all other devices using the Hamming distance, as detailed in Equation 4. Figure 3 illustrates this comparison process, where a pair of devices involved in the Uniqueness calculation is highlighted in green.

$$\text{Uniqueness} = \frac{2}{D(D-1)}\sum_{i=1}^{D-1}\sum_{j=i+1}^{D}\frac{HD(R^i, R^j)}{n} \times 100\% \quad (4)$$

However these metrics can present a series of limitations. First, they inherently produce distributions, yet studies often report only average values without showing variance or full distributions. The second problem, is that complementary values may produce a mean value that is close to ideal, while presenting an overall bad quality. This could be seen by studing in detail the whole distribution of metrics, but these exact distributions are rarely provided. Therefore, we propose to apply the Shannon-Entropy to the canonical metrics to mitigate these issues. The idea behind this is to obtain a new metric that is monotonic and provides a better evaluation. While other functions could be used, Shannon Entropy is interesting due to its links with information theory, cryptography and security. Furthermore, entropy is a nonlinear function, meaning that values close to ideal still retain high entropy, even if they slightly deviate from the optimum.

For the shake of completion, the Shannon entropy of a random variable is computed as shown in Equation 5, where the input value p is the probability of obtaining a 1. By convention, $0\log_2 0$ is defined as 0.

$$H(p) = -\left(p\log_2 p\right) - \left((1-p)\log_2(1-p)\right) \quad (5)$$

Using this entropy-based wrapper, we define several derived metrics: Uniformity per Device (UPD), Uniformity per Challenge (UPC), Entropy-based Uniqueness ($Uniqueness_H$), and Stability. These formulations are captured in the following set of equations:

$$
\begin{aligned}
UPD(d) &= H(Uniformity(d)) \\
UPC(c) &= H(Bit - aliasing(c)) \\
Uniqueness_H &= \frac{2}{D(D-1)} \sum_{i=1}^{D-1} \sum_{j=i+1}^{D} \mathrm{H}\left(\frac{HD(R_i, R_j)}{n}\right) \\
Stability(d, c) &= 1 - H(R_c^d)
\end{aligned}
$$

It is important to emphasise that while these entropy-based adjustments improve upon traditional metrics by addressing distribution-related issues, they are not sufficient to fully characterise the security or unpredictability of a PUF. Entropy alone cannot account for structural patterns or correlations that may exist within the Challenge-Response Pairs. Therefore, it is essential to conduct additional analyses aimed at uncovering such patterns. These investigations can reveal potential vulnerabilities that adversaries may exploit to model or clone the PUF, ultimately enhancing the security assessment of the system.

IV. Reliability and Entropy in Weak PUFs

PUFs suffer from an important limitation: they are not fully reliable. The same device, when exposed to different environmental conditions (such as temperature or voltage variations) or aging, might not always produce exactly the same response to a given challenge. This natural instability must be carefully addressed, especially in weak PUFs that are typically used to generate cryptographic keys or digital signatures, where even a single bit error can lead to catastrophic failures.

The origins of PUF unreliability have been extensively analyzed. As introduced by Maes in [2], early reliability models assumed a uniform random error probability across all PUF cells. However, experiments revealed that certain cells are systematically less stable than others. The newer models consider a distribution of cell reliability, offering a much more accurate understanding of real-world behavior.

The reliability of weak PUFs, particularly Ring Oscillator PUFs (RO-PUFs), is strongly influenced by the physical properties of the compared elements. As discussed in [10] and [11], RO-PUF responses are generated by comparing the frequencies of two nominally identical oscillators. However, when the frequencies are very close, even small environmental variations — such as changes in temperature, supply voltage, or circuit aging — can invert the outcome of the comparison, causing bit flips. To address this instability, the authors propose introducing a differential frequency threshold: only oscillator pairs with frequency differences larger than a defined limit are considered for generating reliable responses. Through extensive experimental measurements on FPGAs, they show that selecting an appropriate threshold — based on worst-case environmental conditions — allows significant improvements in PUF stability. However, they also highlight a trade-off: discarding marginal pairs reduces the total number of usable challenge-response pairs (CRPs) and can slightly degrade the entropy of the system, emphasizing the need to balance reliability against other critical security metrics.

This intrinsic relationship between reliability and entropy is further formalized in [12]. The authors theoretically and experimentally demonstrate that ensuring high reliability by selecting elements with large physical differences (such as large frequency gaps) can unintentionally decrease inter-device entropy. In other words, while stable responses are achieved, they tend to become biased and less random across different chips, weakening the uniqueness property essential for PUF security. Their results, based on large-scale measurements of ring oscillators, confirm a clear correlation: as the reliability of a response increases, the uniformity per challenge (a measure of randomness across devices) tends to decrease. This creates a fundamental tension in weak PUF design: maximizing reliability without excessively sacrificing entropy. Any practical solution — whether based on filtering, architectural improvements, or coding techniques — must carefully balance these two essential properties to ensure robust, unclonable, and secure PUF-based systems.

A. Methods to improve reliability and entropy

Several strategies have been proposed in the literature to address this challenge:

Technological-Level Improvements

One approach is to improve the intrinsic reliability at the circuit level, thus minimizing the need for heavy post-processing. For example, the work on aging-resistant RO-PUFs (ARO-PUFs) [13] proposed specific modifications to the ring oscillator architecture to make it less sensitive to aging mechanisms like Bias Temperature Instability (BTI) and Hot Carrier Injection (HCI). The ARO-PUF design allows the device to maintain stability even after ten years of operation, significantly reducing the number of bit flips compared to traditional RO-PUFs.

Filtering Based on Response Quality

Filtering techniques offer an effective method to optimize both the reliability and entropy of Ring Oscillator PUFs (RO-PUFs). As described in [14], responses generated from RO pairs with small frequency differences are particularly vulnerable to environmental fluctuations and are thus prone to instability. Filtering out such pairs — those with a frequency difference smaller than a threshold $T_{reliability}$ — enhances the reliability of the PUF by removing the most error-prone CRPs. However, the paper also highlights the complementary issue: if the frequency difference between ROs is too large, it may signal the presence of systematic bias (e.g., oscillators coming from different statistical distributions), which can harm entropy by introducing predictable behavior across different devices. Thus, filtering must also exclude CRPs where the frequency difference exceeds a threshold $T_{entropy}$. The study confirms that an optimal set of CRPs exists within a window of frequency differences, neither too small nor too large, allowing simultaneous improvement of both reliability and entropy. Experimental results based on industrial silicon validate this approach, showing that while low thresholds improve stability

at the cost of randomness, and high thresholds enhance entropy at the risk of instability, carefully selecting both limits allows an effective balance to be achieved while maintaining a sufficient number of usable CRPs for practical use.

Similarly, for SRAM-based PUFs, selective use of more stable memory cells can drastically improve reliability. The work in [15] proposed an enhanced stability test methodology at the manufacturing phase. By performing dynamic pre-power-up skewing tests, unstable SRAM cells can be identified and excluded from the PUF response, allowing reliable operation with minimal overhead.

Error Correction Codes and Fuzzy Extractors

Finally, an essential tool is the use of error correction codes (ECC) combined with fuzzy extractors. Fuzzy extractors are cryptographic algorithms designed to produce stable and secure keys from noisy inputs, correcting bit errors during the reconstruction phase.

Typically, the raw PUF response is combined with helper data generated by ECC during enrollment. Later, when reconstructing the key, even if some bits differ due to noise or aging, the original secret can be accurately recovered. State-of-the-art designs leverage combinations of Golay codes, Reed–Muller codes, and BCH codes to achieve high error resilience [16]. For example, it is possible to generate a 128-bit cryptographic key from about 1536 SRAM bits with an error rate as high as 15%, maintaining a reconstruction failure probability below 10^{-6}.

While ECC and fuzzy extractors introduce some area and computation overhead, they allow weak PUFs to be used securely even in harsh environments where physical variations are significant.

V. Machine Learning in Strong PUFs

Strong PUFs, capable of generating a very large number of challenge-response pairs (CRPs), were initially considered resistant to modeling attacks because of their apparent complexity. However, advances in machine learning (ML) techniques over the last decade have demonstrated that many types of strong PUFs are vulnerable to sophisticated learning algorithms.

In a machine learning attack, the adversary collects a set of observed CRPs from the target PUF and trains a predictive model that approximates the PUF's behavior. Once trained, the model can accurately predict the PUF's responses to new, unseen challenges, thereby undermining the security of authentication or key generation schemes based on the PUF.

Early attacks, such as those discussed in [17], employed relatively simple models like Support Vector Machines (SVM) and Logistic Regression (LR) to attack Arbiter PUFs. These methods demonstrated that even supposedly strong PUFs could be broken if enough CRPs were available and if the underlying PUF structure was relatively simple.

More recently, Deep Learning techniques, such as Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), have significantly increased the efficiency and effectiveness of PUF modeling. Neural networks can learn highly non-linear mappings, making them capable of modeling even more complex strong PUF architectures (such as XOR-based, where the output of multiple Arbiter PUFs are XORed).

More in details, [18] showed that even large XOR Arbiter PUFs — previously considered resistant due to their high complexity — can be broken by optimized neural network methods running on ordinary laptops, not just high-performance servers. By adapting training algorithms to handle datasets larger than RAM capacity, they succeeded in breaking 7-XOR and 8-XOR PUFs in under two hours — a task that had previously taken days with traditional ML methods. In the same way, [19] demonstrated that artificial neural networks (especially those using adaptive optimizers like Adam) are particularly effective in modeling responses in on configurable Ring Oscillator PUFs (RO-PUFs). Moreover, the paper [20] introduces an important improvement: Transfer Learning for PUF modeling. Instead of training a model from scratch for every PUF, Transfer Learning reuses internal layers trained on a similar PUF (the "Prime PUF") to initialize the new model (for the "Goal PUF"). This approach reduces the number of CRPs required for training by about 50%, even when some noise is present in the dataset, making modeling attacks faster and more accessible.

Overall, these studies collectively confirm that no current strong PUF architecture is inherently immune to ML attacks if enough CRPs are collected.

Beyond direct attacks, a recent study in [21] proposed a lightweight sensitivity-based metric to assess the vulnerability of strong PUFs. The idea is simple but powerful: if flipping a single bit in the challenge causes a highly predictable change in the response, the PUF is likely easier to model with ML techniques. Conversely, if the PUF exhibits highly sensitive and non-systematic behavior to challenge perturbations, it becomes harder to predict. This sensitivity metric provides a quick, standardized method to estimate a PUF's modeling resistance without having to perform full ML attacks, which are time-consuming and resource-intensive. While not a replacement for exhaustive evaluation, such a tool can be very useful in early-stage PUF design to detect potential vulnerabilities.

To mitigate the growing threat of machine learning attacks, two complementary strategies can be adopted. The first is the use of Controlled PUFs [22], where access to the raw challenge-response behavior is obfuscated: challenges are processed through an input hash function before being applied to the PUF, and the responses are similarly post-processed before output. This obfuscation prevents an attacker from gathering clean CRPs, thus making machine learning-based modeling practically infeasible. The second, simpler approach is to abandon strong PUFs altogether and instead rely on a robust weak PUF to generate a secret cryptographic key. This key can then be used to support a standard symmetric encryption or authentication algorithm, such as AES or HMAC. In this way, the combined system — weak PUF plus symmetric cryptography — effectively behaves like a strong PUF, providing scalable and secure authentication without exposing itself to machine learning vulnerabilities. While this second strategy requires the integration of lightweight cryptographic engines, it leverages

the maturity and proven strength of modern cryptography to complement the intrinsic uniqueness of PUFs.

VI. Physical Attacks

While PUFs are designed to be inherently secure due to their reliance on uncontrollable process variations, they remain susceptible to a variety of physical attacks that exploit their analog nature and environmental dependencies. Unlike purely mathematical attacks (e.g., modeling via machine learning), physical attacks exploit the physical behavior or structure of the device. Physical attacks on PUFs can be categorized as:

- Invasive attacks: Require direct chip access (e.g., delayering, probing) and may irreversibly alter the structure.
- Environmental attacks: Manipulate operating conditions (e.g., temperature, voltage) to influence response stability or induce aging effects.
- Semi-invasive attacks: Interact optically or electromagnetically with the chip (e.g., laser or X-ray fault injection) without modifying its physical layout.
- Side-channel attacks: Extract information through indirect leakage, such as power consumption or EM emissions.

A. Environmental Manipulation Attacks

PUFs are inherently sensitive to environmental conditions, especially temperature and supply voltage. Adversaries can exploit these sensitivities to manipulate the output or degrade the long-term integrity of a PUF.

In [23], the authors demonstrate that elevated temperatures can bias RO-PUF frequencies, making their behavior predictable and reducing entropy. By selectively heating portions of an FPGA, an attacker can induce non-uniform frequency shifts and influence the response bits. This idea is extended in [24] and modeled in [25], where temperature-induced aging effects are deliberately introduced to clone a RO-PUF. By accelerating transistor wear in selected ROs (via thermal stress), the response of a victim PUF can be closely replicated on another device — effectively defeating the unclonability property.

For SRAM PUFs, extreme low temperatures introduce a different vulnerability. In [26], researchers show that below -100°C, data written into SRAM persists after power-off due to data remanence. This allows an attacker to inject a known pattern before reboot and retrieve it as a forged PUF response — even bypassing memory erasure.

To mitigate the effects of physical attacks on PUFs, a variety of countermeasures have been proposed at both the circuit and system levels. One class of defenses is based on monitoring and tamper detection, where embedded sensors track voltage, temperature, or timing variations to detect abnormal behavior caused by physical interference [27]. For example, on-chip delay monitors or timing-to-digital converters (TDCs) can be used to detect shifts in critical signal paths, while analog sensors may alert the system to abnormal environmental conditions. These monitoring techniques are particularly effective for identifying fault injection attempts during runtime. An alternative approach consists in verifying the preservation of the expected statistical characteristics of the generated responses, such as distribution or correlation patterns [28].

B. Semi-Invasive Attacks: Laser, EM, X-ray an Ionizing Particle Injection

Laser injection can target specific logic elements in delay-based PUFs like XOR Arbiter or RO-PUFs. By disturbing latches or comparators, attackers can flip response bits or silence part of the circuit, reducing entropy or simplifying modeling [29].

Another practical and low-cost physical attack vector is electromagnetic fault injection (EMFI), which exploits the sensitivity of integrated circuits to strong, localized EM pulses. In the study presented in [30], the authors analyze how a single EM pulse affects the behavior of a RO PUF. Their results show that the injection of a fast, high-intensity EM pulse can induce transient harmonic effects in the ring oscillator network, effectively disturbing its natural frequency and altering the response of the PUF.

A novel class of attacks involves localized X-ray irradiation, as described in [31]. These attacks modify the electrical behavior of individual ROs through Total Ionizing Dose (TID) effects. Even when the device is powered off, precise irradiation causes semi-permanent shifts in RO frequency, allowing response manipulation without triggering detection mechanisms.

The integrity of SRAM PUFs can also be threatened by ionizing particles. In [32], exposure to cosmic or artificial radiation is shown to flip SRAM startup bits, especially in smaller technology nodes. This compromises response consistency, reliability, and potentially even secret reconstruction when fuzzy extractors are used.

To mitigate such threats, several countermeasures can be adopted, including the use of radiation-hardened memory cells or physical shielding to reduce susceptibility to radiation effects, the implementation of error correction mechanisms and response masking to preserve data integrity in the presence of transient faults, and the integration of fault detection circuits or monitoring logic capable of identifying anomalous behavior indicative of localized attacks.

C. Side-channel attacks

While physical and invasive attacks aim to directly alter the structure or state of a PUF, side-channel attacks represent a more subtle threat: they exploit indirect physical leakage, such as power consumption or electromagnetic emissions, to infer secret information. In particular, power side-channel analysis has emerged as a powerful tool for breaking even obfuscated or strongly protected PUF architectures.

The attack in [33] demonstrates that it is possible to model and break one PUF instance using power traces collected from another, similar instance — even in the presence of noise and environmental variation. Specifically, the attacker uses a reference PUF (fabricated with the same layout and technology) to train a machine learning model based solely on power consumption traces, without needing access to challenge-response pairs from the target PUF. This model is then applied to infer responses of the target device through passive power observation. The authors show that both the latch (arbiter) and the Flip-Flop used to store the response leak distinguishable power signatures, which correlate with the output bit value.

To counteract this attack, the authors propose two circuit-level techniques in [34]. The first is the Dual Flip-Flop mitigation, which balances the power consumption of both outputs of the arbiter (Q and $\overline{Q}$), making it difficult for an attacker to distinguish the stored bit value. The second is Randomized Response Initialization, where the output Flip-Flop is set to a random value before each challenge, introducing noise into the power trace. A hybrid approach combining both strategies significantly reduces the effectiveness of modeling attacks, even under low-noise conditions.

VII. Test of PUFs

Ensuring the reliability and security of Physical Unclonable Functions (PUFs) necessitates rigorous testing and evaluation methodologies. Given the unique nature of PUFs—where their responses are derived from inherent manufacturing variations—traditional testing approaches often fall short.

Indeed, unlike conventional digital circuits that produce deterministic outputs, PUFs generate responses based on unpredictable physical variations. This intrinsic randomness means that there isn't a predefined "golden" output to compare against during testing. Consequently, standard test strategies, such as scan chains, are not directly applicable and may even introduce security vulnerabilities if they expose internal states to potential attackers.

To address this challenge, an alternative methodology has been proposed in [35]. In this approach, a PUF is viewed as a Boolean function mapping binary challenges to binary responses. Rather than testing a single device in isolation, the method analyzes the statistical properties of a population of PUF instances by measuring the correlation between their responses. The key insight is that for a healthy batch of PUFs, the correlation coefficients between different PUF instances should follow a well-defined distribution centered around zero — reflecting the desired uniqueness property, where different PUFs behave independently. If a PUF is defective, for example due to manufacturing faults, systematic biases, or aging-induced degradation, it will exhibit abnormal correlation patterns with respect to the rest of the population. By plotting the correlation spectrum (i.e., the histogram of pairwise correlation coefficients) and identifying outliers, it is possible to efficiently detect faulty or suspicious PUFs.

Another complementary direction is addressed in [36]. This paper focuses on ensuring uniqueness among devices, which is vital for PUF-based authentication schemes. Since exhaustive pairwise comparison is infeasible at scale, the authors propose using Multi-Index Hashing (MIH) to rapidly search for similar responses among a large population. They introduce memory optimization techniques (such as global indexing and Hamming Weight filtering) to make this process viable even for millions of devices. Although still assuming some offline post-processing, this work highlights that uniqueness evaluation could, in principle, be integrated as part of a lightweight on-chip or near-chip test infrastructure, thus reducing the burden on external testers.

However, a practical limitation of these approaches must be noted: it requires reading and analyzing the responses from all manufactured PUFs in order to build the correlation spectrum. In a typical industrial flow, Automated Test Equipment (ATE) is optimized to quickly verify a circuit by comparing its outputs to pre-stored expected values (golden responses) — and does not normally support the collection and statistical processing of large datasets across multiple chips. As a result, applying correlation-spectra analysis in mass production environments would require significant changes to the test flow, including additional data collection steps and post-processing stages outside of standard ATE operations. To address the limitation that conventional ATE systems cannot efficiently test PUFs, various Built-In Self-Test (BIST) strategies have been proposed in the literature. These BIST approaches aim to embed the evaluation capabilities directly into the device itself, allowing real-time, secure, and efficient testing of PUFs without relying on external reference models or extensive off-chip analysis.

A first notable solution is proposed in [37]. This work introduces a hardware-based BIST methodology capable of evaluating both the stability and unpredictability of PUFs at runtime. The architecture monitors the PUF during operation, detecting variations due to environmental changes, aging, or even potential attacks. It combines randomness tests (adapted from NIST standards) with internal structural checks. The scheme was prototyped on FPGA, demonstrating low overhead and practical feasibility, although it requires dedicated hardware resources inside the device.

Building upon similar ideas, in the work [38], the unpredictability of the PUF is assessed using simplified randomness tests adapted for hardware (e.g., frequency, runs, and template matching tests), while the stability is evaluated through sensor-based monitoring, parametric interrogation (where slight variations are induced to check response robustness), and multiple interrogations (voting on repeated challenges). This approach not only allows on-the-fly detection of environmental or malicious effects but also supports adaptive operation, where unstable CRPs can be identified and excluded. One important insight from this work is the recognition that PUF testing must go beyond initial factory characterization and needs continuous, embedded verification to ensure long-term trustworthiness.

Finally, in [39], the authors address the challenge of testing Fuzzy Extractors — critical blocks that process noisy PUF outputs into stable cryptographic keys. They propose a secure scan-chain-free test method, based on a daisy-chained architecture augmented with lightweight hardware structures (such as SISR or MISR registers). This method ensures high stuck-at fault coverage (over 95%) while preventing any leakage of sensitive internal data during testing, an essential property for maintaining the overall security of PUF-based key storage systems.

In summary, testing PUFs presents unique challenges that differentiate them fundamentally from conventional digital circuits. Traditional testing infrastructures, such as Automated Test Equipment (ATE), are optimized to verify deterministic outputs against pre-stored golden responses, which is not feasible for PUFs whose responses are inherently unpredictable and unique to each device. On the other hand, To overcome these

limitations, BIST strategies have emerged as a natural evolution, embedding lightweight evaluation capabilities directly within the device. BIST solutions enable online monitoring of critical PUF properties, such as unpredictability and stability, throughout the lifecycle of the chip. Therefore, a modern and comprehensive testing strategy for PUFs should combine both statistical offline evaluations (during qualification phases) and embedded BIST capabilities (for continuous and secure monitoring in production and field deployment). Such a dual approach ensures that PUF-based security solutions maintain their fundamental properties of unpredictability, reliability, and unclonability over time, even in the face of operational stresses or emerging attack techniques.

VIII. Conclusions

This paper has provided an overview of the fundamental principles and practical challenges surrounding Physical Unclonable Functions (PUFs), focusing on their use in embedded systems for key generation and authentication. We reviewed the distinction between weak and strong PUFs, examined key evaluation metrics, and explored the critical trade-off between reliability and entropy in weak PUFs. For strong PUFs, we emphasized the growing threat of machine learning-based modeling, including recent side-channel-assisted techniques, and discussed architectural countermeasures.

We also addressed the vulnerability of PUFs to physical attacks such as laser, X-ray, electromagnetic, and temperature manipulation. These attacks highlight the need for integrated countermeasures — including obfuscation, monitoring, and circuit-level protections — to ensure PUF security in real-world conditions.

Future directions include the development of standardized evaluation frameworks and scalable test strategies that can reliably assess PUF performance and resilience without compromising their security properties.

References

[1] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002. DOI: 10.1126/science.1074376. eprint: https://www.science.org/doi/pdf/10.1126/science.1074376. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1074376.

[2] R. Maes, "An accurate probabilistic reliability model for silicon pufs," in *Cryptographic Hardware and Embedded Systems - CHES 2013*, G. Bertoni and J.-S. Coron, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 73–89, ISBN: 978-3-642-40349-1.

[3] M. Barbareschi et al, "A ring oscillator-based identification mechanism immune to aging and external working conditions," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. PP, pp. 1–23, Aug. 2017. DOI: 10.1109/TCSI.2017.2727546.

[4] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Fpga intrinsic pufs and their use for ip protection," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, P. Paillier and I. Verbauwhede, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 63–80, ISBN: 978-3-540-74735-2.

[5] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02, Washington, DC, USA: Association for Computing Machinery, 2002, pp. 148–160, ISBN: 1581136129. DOI: 10.1145/586110.586132. [Online]. Available: https://doi.org/10.1145/586110.586132.

[6] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*, 2004, pp. 176–179. DOI: 10.1109/VLSIC.2004.1346548.

[7] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *Cryptographic Hardware and Embedded Systems - CHES 2006*, L. Goubin and M. Matsui, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 369–383, ISBN: 978-3-540-46561-4.

[8] E. I. Vatajelu, G. D. Natale, M. Barbareschi, L. Torres, M. Indaco, and P. Prinetto, "Stt-mram-based puf architecture exploiting magnetic tunnel junction fabrication-induced variability," *J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 1, May 2016, ISSN: 1550-4832. DOI: 10.1145/2790302. [Online]. Available: https://doi.org/10.1145/2790302.

[9] P. Koeberl, Ü. Kocabaş, and A.-R. Sadeghi, "Memristor pufs: A new generation of memory-based physically unclonable functions," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2013, pp. 428–431. DOI: 10.7873/DATE.2013.096.

[10] A. Schaub, J.-L. Danger, S. Guilley, and O. Rioul, "An improved analysis of reliability and entropy for delay pufs," in *2018 21st Euromicro Conference on Digital System Design (DSD)*, 2018, pp. 553–560. DOI: 10.1109/DSD.2018.00096.

[11] H. Martin, E.-I. Vatajelu, G. Di Natale, and O. Keren, "On the reliability of the ring oscillator physically unclonable functions," in *2019 IEEE 4th International Verification and Security Workshop (IVSW)*, 2019, pp. 25–30. DOI: 10.1109/IVSW.2019.8854401.

[12] V. Kulagin, S. V. Gutierrez, T. Kilian, *et al.*, "On the relation between reliability and entropy in physical unclonable functions," *IEEE Design & Test*, vol. 41, no. 6, pp. 46–53, 2024. DOI: 10.1109/MDAT.2024.3425791.

[13] M. T. Rahman, F. Rahman, D. Forte, and M. Tehranipoor, "An aging-resistant ro-puf for reliable key generation," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 3, pp. 335–348, 2016. DOI: 10.1109/TETC.2015.2474741.

[14] E.-I. V. Vasilii Kulagin Giorgio Di Natale, "Optimizing ro-pufs: A filtering approach to reliability and entropy trade-offs," in *2025 IEEE European Test Symposium (ETS)*, 2025.

[15] E. I. Vatajelu, G. Di Natale, and P. Prinetto, "Towards a highly reliable sram-based pufs," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 273–276.

[16] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014. DOI: 10.1109/JPROC.2014.2320516.

[17] M. S. Alkatheiri and Y. Zhuang, "Towards fast and accurate machine learning attacks of feed-forward arbiter pufs," in *2017 IEEE Conference on Dependable and Secure Computing*, 2017, pp. 181–187. DOI: 10.1109/DESEC.2017.8073845.

[18] A. O. Aseeri, Y. Zhuang, and M. S. Alkatheiri, "A machine learning-based security vulnerability study on xor pufs for resource-constraint internet of things," in *2018 IEEE International Congress on Internet of Things (ICIOT)*, 2018, pp. 49–56. DOI: 10.1109/ICIOT.2018.00014.

[19] S. Kumar and M. Niamat, "Machine learning based modeling attacks on a configurable puf," in *NAECON 2018 - IEEE National Aerospace and Electronics Conference*, 2018, pp. 169–173. DOI: 10.1109/NAECON.2018.8556818.

[20] A. Ali-Pour, D. Hely, V. Beroulle, and G. Di Natale, "An efficient approach to model strong puf with multi-layer perceptron using transfer learning," in *2022 23rd International Symposium on Quality Electronic Design (ISQED)*, 2022, pp. 1–6. DOI: 10.1109/ISQED54688.2022.9806257.

[21] W. Stefani, F. Kappelhoff, M. Gruber, *et al.*, *Strong PUF security metrics: Sensitivity of responses to single challenge bit flips*, Cryptology ePrint Archive, Paper 2024/378, 2024. [Online]. Available: https://eprint.iacr.org/2024/378.

[22] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Controlled physical random functions," in *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, 2002, pp. 149–160. DOI: 10.1109/CSAC.2002.1176287.

[23] D. Nedospasov, J.-P. Seifert, C. Helfmeier, and C. Boit, "Invasive puf analysis," in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, IEEE, 2013, pp. 30–38.

[24] H. Cook, J. Thompson, Z. Tripp, B. Hutchings, and J. Goeders, "Cloning the unclonable: Physically cloning an fpga ring-oscillator

puf," in *2022 International Conference on Field-Programmable Technology (ICFPT)*, 2022, pp. 1–10. DOI: 10.1109/ICFPT56656.2022.9974597.

[25] A. Douadi, E.-I. Vatajelu, P. Maistri, D. Hely, V. Beroulle, and G. Di Natale, "Modeling thermal effects for biasing pufs," in *2024 IEEE European Test Symposium (ETS)*, IEEE, 2024, pp. 1–4.

[26] N. A. Anagnostopoulos, T. Arul, M. Rosenstihl, A. Schaller, S. Gabmeyer, and S. Katzenbeisser, "Low-temperature data remanence attacks against intrinsic sram pufs," in *2018 21st Euromicro Conference on Digital System Design (DSD)*, IEEE, 2018, pp. 581–585.

[27] T. Köylü, L. Garaffa, C. Reinbrecht, M. Zahedi, S. Hamdioui, and M. Taouil, "Exploiting puf variation to detect fault injection attacks," in *2022 25th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, 2022, pp. 74–79. DOI: 10.1109/DDECS54261.2022.9770154.

[28] M. R. Heidari Iman, S. Vinagrero Gutierrez, E.-I. Vatajelu, and G. Di Natale, "An innovative data mining technique for automatic anomaly detection in physical unclonable functions," in *2025 28th International Symposium on Design & Diagnostics of Electronic Circuits & Systems (DDECS)*, 2025.

[29] S. Tajik, H. Lohrke, F. Ganji, J.-P. Seifert, and C. Boit, "Laser fault attack on physically unclonable functions," in *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, IEEE, Sep. 2799, pp. 85–96. DOI: 10.1109/FDTC.2015.19.

[30] S. E. Amraoui, A. Douadi, R. Leveugle, and P. Maistri, "Harmonic response of ring oscillators under single electromagnetic pulsed fault injection," in *2024 IEEE 25th Latin American Test Symposium (LATS)*, 2024, pp. 1–6. DOI: 10.1109/LATS62223.2024.10534602.

[31] N.-E. O. Tebina, A. Douadi, L. Salvo, *et al.*, "Non-invasive attack on ring oscillator-based pufs through localized x-ray irradiation," in *2024 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, IEEE, 2024, pp. 01–11.

[32] U. Surendanathan, H. Wilson, L. R. Cao, A. Milenkovic, and B. Ray, "Analysis of sram puf integrity under ionizing radiation: Effects of stored data and technology node," *IEEE Transactions on Nuclear Science*, 2023.

[33] T. Kroeger, W. Cheng, S. Guilley, J.-L. Danger, and N. Karimi, "Cross-puf attacks on arbiter-pufs through their power side-channel," in *2020 IEEE International Test Conference (ITC)*, IEEE, 2020, pp. 1–5.

[34] T. Kroeger, W. Cheng, S. Guilley, J.-L. Danger, and N. Karimi, "Making obfuscated pufs secure against power side-channel based modeling attacks," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2021, pp. 1000–1005.

[35] D. Chatterjee, A. Hazra, and D. Mukhopadhyay, "Testability analysis of pufs leveraging correlation-spectra in boolean functions," *CoRR*, vol. abs/1810.08821, 2018. arXiv: 1810.08821. [Online]. Available: http://arxiv.org/abs/1810.08821.

[36] L. Santiago de Araújo, V. C. Patil, L. Augusto Justen Marzulo, F. Maia Galvão França, and S. Kundu, "Efficient testing of physically unclonable functions for uniqueness," in *2019 IEEE 28th Asian Test Symposium (ATS)*, 2019, pp. 117–1175. DOI: 10.1109/ATS47505.2019.00022.

[37] S. U. Hussain, M. Majzoobi, and F. Koushanfar, "A built-in-self-test scheme for online evaluation of physical unclonable functions and true random number generators," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 1, pp. 2–16, 2016. DOI: 10.1109/TMSCS.2016.2519902.

[38] S. U. Hussain, S. Yellapantula, M. Majzoobi, and F. Koushanfar, "Bist-puf: Online, hardware-based evaluation of physically unclonable circuit identifiers," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 162–169. DOI: 10.1109/ICCAD.2014.7001347.

[39] M. Cortez, G. Roelofs, S. Hamdioui, and G. di Natale, "Testing puf-based secure key storage circuits," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2014, pp. 1–6. DOI: 10.7873/DATE.2014.207.