PUF Enrollment and Life Cycle Management: Solutions and Perspectives for the Test Community

Amir Ali Pour*, Vincent Beroulle*, Bertrand Cambou†, Jean-Luc Danger‡,
Giorgio Di Natale§, David Hely*, Sylvain Guilley‡ and Naghmeh Karimi¶
Univ. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France
†Northern Arizona University
‡Secure-IC and TELECOM-Paris
§Univ. Grenoble Alpes, CNRS, Grenoble INP*, TIMA, 38000 Grenoble, France
¶University of Maryland

***Abstract*—Physically Unclonable Functions (PUFs) allow to extract unique fingerprints from silicon chips. The applications are numerous: chip identification, chip master key extraction, authentication protocol, unique seeding, etc. However, secure usage of PUF requires some precautions. This paper reviews industrial concerns associated with PUF operation, including those occurring before and after market. Namely, starting from PUF "secure" specifications, aligned with state-of-the-art standards, we explore innovative techniques to handle enrollment and subsequent PUF queries, in nominal as well as in adversarial environment.**

Key words: Physically Unclonable Functions (PUF), Standardization, Enrollment, Challenge/Response Pair Database Management, On Line Test.

I. Introduction

Silicon PUF-based schemes have been proposed in the literature in order to provide security services such as chip identification, master key generation, user and device authentication, pseudo-random number generator seeding, etc. Although PUF solutions are very diverse and rely on a variety of technologies exhibiting different characteristics, they require unified techniques to securely manage the PUF hardware primitives during the whole life cycle.

A PUF is a function that generates an output (also called response) starting from an input (also called challenge). The set of Challenge-Response Pairs (CRPs) generated by a device must be unique within the family of manufactured circuits. PUFs can be used in all applications where a different signature is required for every single circuit, without the need for programming the signature itself. More in particular, it can be used: (i) to generate an identifier of the circuit; (ii) to generate secret keys; (iii) to be used in authentication protocols. In the rest of the paper, we often refer to CRP to identify the PUF secret data (or signature) although in some cases it might only be a secret identifier extracted from the PUF.

The set of CRPs for a given PUF should be stable in time and no matter the variations of physical and electrical parameters (e.g., temperature and supply voltage). This property (also known as *reliability*) is a mandatory requirement at system-level. Indeed, if the PUF does not deliver the correct response to a given challenge, then the subsequent applications fail: if the PUF is used to generate a secret key, the effect of unstable response will mimic the behavior of a fault injection attack, which is known to be extremely powerful to break ciphers [1]; on the other hand, if the PUF is used in an authentication protocol, the protocol will fail. In the rest of the paper, we consider the PUF architecture as a whole, i.e., the PUF is supposed to be designed together with an associated processing to enhance its reliability.

In order for a PUF-based circuit to be used, its signature has to be first generated and stored in a secure server after its fabrication, and before its deployment over the target application. This procedure is called *enrollment*, which is a critical operation for PUF-based system. During enrollment, challenges and responses should be fully controllable and observable in order to create and store a relevant signature. However, since the set of CRPs represents the secret information of the PUF, such a procedure must be performed in a secure environment, the set of CRPs must be stored in a secure database, and secure access methods must guarantee that a non-authorized person cannot reuse this mechanism to build the same database. Moreover, during the mission mode (corresponding to the usage of PUF), it is necessary to guarantee that the PUF fulfills security criteria to build trustworthy security services. In order to ease the development of secure and efficient PUF-based security schemes, it is then necessary to address the following points with advanced and methodical techniques:

- How to extract the PUFs secret information (i.e., the set of CRPs) after fabrication in a secure way?
- How to to store the CRPs in the server database?
- How to optimize PUF data extraction (in terms of time required to extract the CRPs and space to store them) in order to make it suitable with economical constraints?
- How to secure the device in mission mode so that an attacker cannot hijack the device to extract the physical properties and then build an image of the database?
- How to securely update the database in mission mode to compensate for natural variations of the CRPs (due for instance to aging)?
- How to test the quality of the PUF after manufacturing?
- How to assess the security properties of the PUF primitive during mission mode?

These challenges are very similar to those addressed by the test community for the test of classical electronic devices. Offline and Online test challenges (such as test time and test coverage optimization, security issues during test among other) have been solved by the test community thanks to methodical

*Institute of Engineering Univ. Grenoble Alpes

978-1-7281-4312-5/20/$31.00 ©2020 IEEE

and well-structured techniques. Dedicated EDA tools or design techniques have been proposed to tackle these challenges. Nevertheless, manufacturing and online test of PUFs, as well as security assessment of their properties is fundamental different from classical devices since the responses of the PUFs are not monastically known at design time.

In this paper we propose an overview of existing solutions for PUF enrollment and life-cycle management, and we propose novel techniques inherited or inspired by the existing practices coming from the test community, to define structured tools and architectures dedicated to the PUF life-cycle management. Our considerations are general: our state-of-the-art reviews and innovative techniques are agnostic about the PUF kind. We assume that the PUF is suitable for industrial applications, in particular that it meets the reliability expectations required for commercial applications.

The main contributions of this paper are:

- An overview of existing solutions for PUF secure life cycle management. It encompasses operations (such as enrollment) to be carried out before the PUF is put on the market, and maintenance operations (re-enrollment, test). Those operations can consist in functional evolution or in adaptation to adversarial conditions (aging, attacks, etc.).
- The state-of-the-art of the canonical method to describe a PUF, according to the framework of ISO/IEC 20897. Indeed, a well-defined PUF is a mandatory building block for a subsequent in-field reliable usage.
- An on-chip enrollment infrastructure inherited from secure integrated circuit testing methodologies. We also highlight the similitude between secure IC testing and PUF enrollment.
- A proposal to exploit modeling attacks against PUF in order to provide a new enrollment method, which can allow replacing the usual PUF database within the secure server by an accurate and agile PUF model.
- The description of existing techniques to make the PUF more robust during its life cycle. The first one is the digital sensor which represents a generic protection to detect the Fault Injection Attacks (FIA) by measuring its impact on the propagation time. The other presented techniques are specific to the considered threat, as aging and side-channel analysis, and the PUF architecture, as SRAM-PUF and Delay-PUF.

The rest of the paper is structured as follows. First, the test and evaluation of PUFs in the framework of international standardization is detailed in Sec. II. Second, the question of off-line PUF secure and efficient PUF enrollment is tackled with in Sec. III. Third, pragmatic challenge-response database management schemes are analyzed and improved thanks to neural networks in Sec. IV. The secure usage of PUFs facing challenging environments is detailed in Sec. V. Finally, conclusions and perspectives are drawn in Sec. VI.

II. PUF STANDARDIZATION

In the first place, before considering how the PUF will be used, we must set the ground to a PUF we can trust. It is the prerequisite we present in this Sec. II, under the terms of on-going international standardization process.

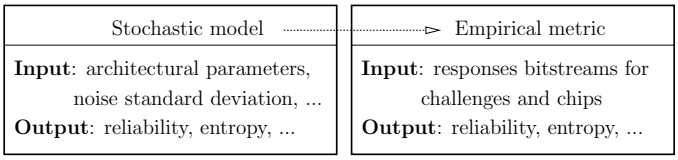


Figure 1. PUF empirical metrics are derived from a stochastic model of security properties

The ISO/IEC 20897 standard [2] specifies the test and evaluation methods for physically unclonable functions. The test and evaluation methods consist of inspection of the design rationale of the PUF and comparison between statistical analyses of the responses from a batch of PUFs or a unique PUF versus specified thresholds. A scientific overview of standardization effort is detailed in the article [3]. This article focuses more particularly on the part 1 of the ISO/IEC 20897 standard.

This section reviews some key points about Physically Unclonable Functions (PUF) metrics and stochastic models, which is the topic of ISO/IEC 20897 part 2. At this stage, the contents is mostly mathematical: it aims at easing discussions between experts involved in PUF evaluation. The key points are highlighted in **boldface text**, and shall be reviewed first.

A large part of the material in this note is directly influenced by the seminal work by Yohei Hori et al. [4]. New directions are also indicated, as less error-prone alternatives.

A. Introduction

The goal of ISO/IEC 20897 part 2 (in Committee Draft stage at the time of writing this paper) is to present some meaningful metrics to quantify the quality of a PUF. Such metrics shall have two desirable properties:

1) be easily computable in practice, thanks to experiments, and
2) relate to concepts which can be computed in theory from a model of the PUF (so-called stochastic model).

In this respect, **conceptual metrics will be computed on the stochastic model**, and **real-world metrics will consist in estimation of the conceptual metrics on PUF measurements**. This is illustrated in Fig. 1. Actually, stochastic models, addressed in Sec. II-B can attest of the security level (but on an ideal model), whereas empirical metrics, addressed in Sec. II-C, checks whether something fails (in real silicon).

B. Stochastic model

A PUF is encountering two kinds of noise:

1) dynamic noise (= thermal noise), and
2) static noise (= process noise).

Assumptions shall be done on those noise probability distributions, e.g., they are independent and normal. For numerical applications in the model, the values of the variance such noises shall be estimated. Dynamic noise requires an experimental chip, so as to carry out measurements (e.g., the phase noise in oscillating structures). Static noise is usually characterized by the silicon founder, as Pelgrom coefficients [5], and can be gotten by Monte-Carlo simulations.

We detail hereafter the three relevant metrics:

n bits	m bits
challenges (in the API)	responses (in the API)

Figure 2. PUF models as an (n, m)-function

1) the reliability (computed as a Bit Error Rate, or BER),
2) the randomness (the property of a PUF responses to be balanced), and
3) the decorrelation (the property of responses to be independent).

Notice that reliability is actually primarily a safety metric, as it quantifies the proportion of responses which fail. However, it can also be seen as a security metric insofar as incorrect responses can open the door to attacks, for instance in the context of devices authentication. A device might indeed inadvertently "impersonate" another one. The randomness and the decorrelation are actually metric innate to the PUF (irrespective of its environment), and are facets of its entropy. Clearly, the randomness shall be ideal (balanced) and as a second criteria, the responses shall be decorrelated, within one PUF (from one challenge to the other one – intra-chip decorrelation), and between two PUFs (inter-chip decorrelation, which is akin uniqueness).

1) Reliability: Let us assume that:

a) dynamic noise is distributed as $D \sim \mathcal{N}(0, \sigma_D^2)$, and
b) static noise is distributed as $S \sim \mathcal{N}(0, \sigma_S^2)$.

The PUF model is $X = D + S$, and one measurement is $x = t_D + t_S$. Then, for one realization, we have:

$$\text{BER} = \mathbb{P}(\text{correct}) = \frac{1}{2}\left[1 - \text{erf}\left(\frac{t_S}{\sqrt{2}\sigma_D}\right)\right].$$

The average BER (averaged over t_S, normally distributed) is:

$$\langle \text{BER} \rangle = \frac{1}{\pi} \arctan\left(\frac{1}{\sqrt{\text{SNR}}}\right),$$

where $\text{SNR} = \sigma_S^2/\sigma_D^2$ [6], [7].

2) Randomness: We model a PUF a **vectorial Boolean function** $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ [8]. The integers n and m are defined as follows:

- n is the number of bits in the challenge (N_{chal} in [4]), that is the total number of challenges is 2^n, and
- m is the number of bits in the response; it is possible that $m = 1$. In RAM PUFs, m is the memory word bitwidth.

Let us notice that this modelization, illustrated in Fig. 2 is fairly generic. The responses are designed to be m bit long. Those could be the m bits from a weak PUF (e.g., $m \in \{8, 16, 32, 64\}$ for SRAM PUFs), or the m successive derivation of bits from a PUF primitive which delivers only 1 bit by basic challenge. The later PUF can be upgraded into a PUF as represented in Fig. 2 by grouping m challenges together, which leaving the other free parts of the plaintext for other purposes.

The randomness is simply:

$$\mathbb{E}_{i,X}(f_i(X)) \in [0, 1].$$

Ideally, the randomness is equal to $\frac{1}{2}$.

This metric is ideal when data is averaged in mean and i.i.d. Otherwise, more generic tests (not bit by bit) can be carried out on the bitstream formed by the concatenation of the responses.

3) Decorrelation: Let two challenges $x, x' \in \mathbb{F}_2^n$. Then, the signed correlation between $f(x)$ and $f(x')$ is:

$$\sum_{i=1}^{m} (-1)^{f_i(x)+f_i(x')} \in [-m, +m]. \tag{1}$$

This value shall be as close as zero as possible. Notice that (1) is a sibling notion to centered Hamming Distance (HD) in Yohei Hori's work [4]. Indeed, for all integer a, we have $(-1)^a = 1 - 2(a \bmod 2)$; hence $\sum_{i=1}^{m}(-1)^{f_i(x)+f_i(x')} = m - 2 \cdot \mathsf{HD}(f(x), f(x'))$.

However, **one shall not consider the average of these correlation over all x, x' pairs**. Indeed, as a motivating counter-example, let us consider a function which has correlated (both positively and negatively) responses, but with such property that in average, the correlations cancel out.

Consider for instance the three responses (written in binary) to three challenges:

- R1: 10101100
- R2: 01101100
- R3: 10100011

Those bitstrings are balanced (there are as many 0s as 1s). But their Hamming distances (or signed correlation) are not optimal:

- HD(R1, R2) = 2
- HD(R2, R3) = 6
- HD(R1, R3) = 4

However, the average Hamming distance is $(2 + 6 + 4)/3 = 4$, which might give a false confidence that responses to different challenges are "independent".

Instead, we recommend the **average of the square of the correlation over all x, x' pairs**. This yields a metric:

Definition 1 (Average correlation)**.**

$$\sum_{x,x'} \left(\sum_{i=1}^{m} (-1)^{f_i(x)+f_i(x')}\right)^2.$$

We have the following lemma:

Lemma 1 (Equivalence of average correlation between challenges and between bits)**.**

$$\sum_{x,x'} \left(\sum_{i=1}^{m} (-1)^{f_i(x)+f_i(x')}\right)^2 = \sum_{i,j} \left(\sum_{x \in \mathbb{F}_2^n} (-1)^{f_i(x)+f_j(x)}\right)^2.$$

Proof. Develop the square, swap sum symbols, and reorganize terms. □

This lemma 1 shows that the average correlation between challenges is equal to the average correlation between bits of the same challenge.

C. Metrics

In this section, we define metrics which can be computed on any PUF, irrespective of a PUF stochastic model.

1) Reliability: Reliability simply consists in repeating experiments for a given challenge, and recording the number of times the PUF responses differ. The proportion of failing elements is referred to as the empirical BER. It is clear that the accuracy of such evaluation is limited. If the target **significance level is 10^{-6} [i.e., one PUF out of one million chip fails]**, then we know from AIS31 [9] that PUF is reliable have having analyzed $20,000$ bits, and that the number of zeroes is comprised between 9654 and 10346.

2) Randomness: The same reasoning applies: the randomness is the empirical proportion of zeroes. The behavior is abnormal if the number of zeroes is outside of range $[9654, 10346]$.

3) Decorrelation: It is first interesting to notice that **decorrelation contradicts with entropy**. Indeed, let us assume $n = 2$ and $m = 4$. Then, assuming the PUF is:

- $f(00) = (0011)_2$,
- $f(01) = (0101)_2$,
- $f(10) = (0110)_2$.

There is no other independent vector $f(11) \in \mathbb{F}_2^m$ of weight $m/2 = 2$ such that $f(11)$ is orthogonal to all vectors $f(00)$, $f(01)$ and $f(10)$. Therefore, **there is a tradeoff between entropy and decorrelation**.

Regarding decorrelation, the confidence interval $\sum_{x,x'} \left(\sum_{i=1}^{m} (-1)^{f_i(x)+f_i(x')} \right)^2$ is computed. **This value cannot to equal to zero, but can be made as small as possible, e.g., under constraint of orthogonality.**

Now that we have set the ground of a trustworthy PUF, we can analyse how to have its usage be secure in "mission mode".

III. Secure IC testing and PUF enrollment

A. PUF enrollment versus Secure IC testing: similitude and differences

IC testing and PUF enrollment are both critical operations performed just after manufacturing in order to respectively guarantee the IC has been properly manufactured without defect and to create a database containing the PUF data which will be further used in mission mode for security purpose.

First, from a functional point of view, IC testing and PUF enrolment both need have high observability and controllability respectively on the design under test and on the Physical Unclonable Function in order respectively test the design and enroll the PUF. During integrated circuit testing, one applies patterns and observe a test response which is compared to a golden model. During PUF enrollment, one applies a challenge and gets the corresponding response which is stored in a secure database. Both operations need to be performed in a timely manner in order to reduce the cost of post manufacturing operations which is high in the semiconductor industry. Nevertheless, some differences may be worth to mention:

- During IC testing, it is important to check as soon as possible the correctness of the response (during enrollment the responses can be stored and processed later within the server).
- During IC testing, the test patterns need to be exhaustive for a given fault model, while enrollment can be partial (i.e. the database can store a subset of the whole PUF CRPs space).

Secondly, from a security point view, IC testing and PUF enrollment have a common threat. The PUF enrollment is performed after fabrication to store within a database the secret related to each PUF units. As a consequence, a secure mechanism is required to access to the PUF within the SoC to create the corresponding database. Nevertheless, this should be made in such a way that a non-authorized entity should not be able to reuse this mechanism to build a copy of the database. During the Integrated circuit life cycle, such a mechanism needs to be activated at two different stages: during post manufacturing operations for the initial enrollment and during mission mode in order to update the database if needed. During post manufacturing enrollment, in a secure environment, one has full access to the PUF in order to efficiently create the database while in mission mode this access must be securely restricted to authorized users only. On one hand, the PUF access needs thus to be as efficient as possible in order to facilitate the PUF database creation after manufacturing. On the other hand, when the device is deployed in mission mode, this access can become a severe threat against PUF-based security schemes if an attacker can activate it. Secure Integrated Circuit (IC) testing is facing the same dilemma, for post-production testing it is required to provide both observability and controllability of the internal nodes of the circuit under test, while in mission mode security requirements imply to restrict to the minimum the access of the internal content of the IC [10]. PUFs enrollment and Secure IC post-fabrication testing have indeed a lot in commons. PUF enrollment methods can inherit many features from the test community. IC testing relies on standardized (or at least broadly shared) methods which ease the access to the on-chip test mechanisms and the design of on-chip test circuitry in order to guarantee a high quality of testing without scarifying cost, performance and security constraints. Could we leverage these IC testing techniques to design efficient and secure PUF enrollment infrastructures? Scan based testing methods are very good candidate in order to meet the requirements of efficient and secure PUF enrollment.

B. Leveraging Secure IC testing methods for PUF enrollment

Considering the similarities previously discussed between IC testing and enrollment, we propose to leverage and to modify existing secure test methods to perform the PUF enrollment. The proposed method described hereafter reuses scan chain-based test methods and Logic built in self-test (LBIST) methods. Leveraging these methods speed up the PUF enrollment operation thanks to the known efficiency of scan chain operations and also allows to benefit from the dedicated security add-ons proposed in the literature to secure the IC testing infrastructures [11] and consequently address the security issues related to the enrollment. Connecting the PUF enrollment hardware module to the test infrastructure makes the use of standardized test access mechanism available to perform the enrollment during the post manufacturing operations. IC test standards define intra chip and off-chip access in order to

efficiently apply the test patterns and to observe the response [12], [13]. These standards have been defined so that it is easy to integrate a new hardware block within the test access mechanism whatever the nature of the block. Such mechanisms could be leveraged to access the PUF during enrollment. This would help the on-chip PUF integration (avoiding additional hardware) and also ease the engineering work related to the enrollment after fabrication.

We thus propose to integrate the PUF enrollment infrastructure within the test infrastructure of the system as depicted on Fig. 3

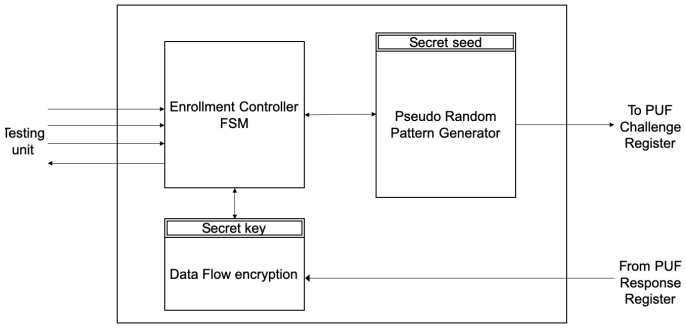


Figure 3. Enrollment Controller Architecture.

The enrollment controller can simply be integrated within the on-chip test controller by adding extra instructions dedicated to the PUF enrollment. This one manages the enrollment operations from both a functional and security point of view as the secure test controller proposed in [14].

The enrollment controller manages an on-chip generator to generate PUF challenges and then uses scan test methods to output the associated response value. The controller of course integrates security mechanisms in order to give access to the PUF enrollment hardware only after a strong authentication as this is performed for secure IC testing. Additional security features initially designed to secure IC testing operations (such as encryption) can be integrated to get additional layers of security to guarantee the confidentiality of the PUF data.

Security add-ons discussed in the following are designed to protect the data being scanned making them hardly exploitable without a secret key. This secret key which needs to be known only by the manufacturer can be deeply embedded in the design and do not need to be shared with other stakeholders. The database can then be offline decrypted and encrypted again with another mean (and key) to be shared with a stakeholder (at the application level) building a secure application leveraging the PUF. Indeed, the main advantage of PUF is to avoid to have to store application related secret keys in order to avoid the use of costly secure nonvolatile memory and above all expensive key injection operations. The PUF response encryption techniques discussed here are inspired from secure scan methodologies described in[15] and [16]. As a result, the proposed mechanism makes enrollment data not exploitable by non-authorized user. The controller has a limited number of inputs and outputs to limit the attack surface while in mission mode. Like commonly used test controller, the controller exchanges data with the tester with simple serial I/Os. Once the tester has been correctly authenticated, the controller manages a dedicated hardware circuit which generates the challenges (the pseudo random pattern generator on Fig. 3). The PUF response is then shifted out to the controller and can be ciphered before to be sent back to the tester via the scan serial interface. Such an architectures advantages are twofold. First from a security point of view the attack surface against the PUF is dramatically reduced, one needs only to secure the controller, no access through the software is possible. Second, the enrollment time is dramatically reduced compare to a traditional approach based on microcontroller unit and can be fully integrated in the test operations after manufacturing.

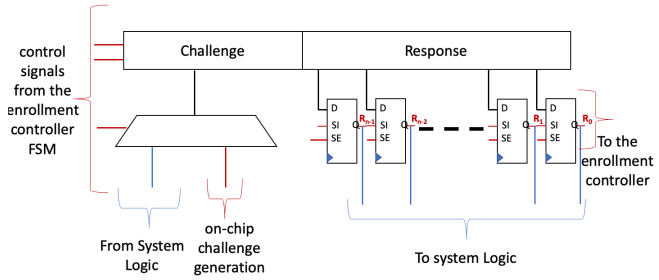


Figure 4. PUF Logic Circuitry with Enrollment logic.

The PUF digital interface depicted in Fig. 4 can be divided into two parts:

- The PUF Challenge register: these flip-flops directly drive the PUF to apply a challenge;
- The PUF response register: these flip-flops are connected to the PUF output and are used to capture the PUF response value (Ri on Fig. 4).

In mission mode, the digital logic of the system controls the PUF avoiding to expose brute data from the PUF to rest of the system. During Enrollment, one should set the PUF challenge register at all the necessary values to apply enough challenges. For each value of PUF challenge register, the PUF responses need to be scanned out and further be stored within the database. The enrollment can be very time consuming since a lot of data need to be read from the PUF. In order to optimize it, we propose to use a full hardware solution based on scan chain and on-chip signal generator. A hardware module generates on-chip the challenge data to sense the PUF and a scan register is used to output the acquired data as depicted in Fig. 4. The enrollment is thus done by repeating the following sequence for each challenge:

1) The on-chip challenge generator generates a challenge data;
2) The corresponding PUF response value R_i is acquired in the PUF Response register;
3) The scan chain is activated and in n clock cycles the values Ri are shifted out via the configured serial register.

This hardware enrollment mechanism needs to be secure to:

- Be activated only upon a strong authentication;
- Protect the confidentiality of the data being shifted out.

The extracted database can be exploited at two conditions (1) decrypting the response value and (2) linking each response to

a challenge. This will not be possible if the attacker does not know the initial seed of the pseudo random pattern generator and the initial key of the stream cipher. A simple protocol to establish unique PRPG seed and encryption key for each device database during the enrollment based on a secret key buried in the device is proposed. Both the test unit and the devices agree on a common key for the stream cipher and a random seed for the challenge generator, which are both securely generated on each side thanks to a secure Key and a shared exchanged random numbers. The controller initiates the enrollment and the challenge generator produces the value Gen0 which drives the PUF logic control, as a result the value R_0 is generated by the PUF and shifted to the stream cipher. The cipher value R'_0 is then shifted out to the test unit. The operation is performed n times with the n different values generated by the on-chip generator. The database contains then n ciphered PUF responses. On the server side, the PUF data-base can be re-constructed knowing the secret enrollment key and the exchanged random value. The database is first decrypted, then the sequence generated on-chip is re-generated on the server in order to associate a challenge to each value which has been acquired. Finally, the PUF values are re-ordered to build the secure database which will be used by the server in mission mode. Based on secure IC testing, this PUF enrollment architecture provides a secure method to generated the PUF database in a timely manner thanks to the efficiency of scan chain access. It is a first step towards a secure and standardized secure access for PUF enrollment being totally independent of the PUF technology.

IV. Deep learning techniques for Challenge/Response Pair Database Management during the Life Cycle

The mandatory requirement for correctly using a PUF-based system is to have a secure server, which is in charge of storing the secret information extracted from the PUF during the enrollment. In mission mode, the secure server is then involved in the establishment of secure protocols requiring the use of the PUF secret information. These protocols will, for instance, guarantee the authenticity of the hardware device, or will allow confidentiality through cryptographic primitives using the PUF value as master key.

While several techniques have been proposed in literature for the integration of PUF-based protocols to enhance authentication and communication protocols, the practical and industrial issues related to the enrollment management are not yet addressed. In particular, both the reduction of the time required to enroll a huge amount of hardware devices, and the reduction of the amount of data that must be stored in the secure server are still open problems.

To mitigate such issues, a possible solution would be to rely on modeling techniques to generate a software model of the PUF instance as a reference, which would replace the traditionally stored set of CRPs. In this case, the enrollment procedure would require the measurement of a subset of the possible CRPs. This subset would be smaller than the one required in the traditional approach. Starting from this subset of CRPs, the model of the PUF is built and it is then used to predict the entire possible CRPs of that instance.

To this day, several works have been conducted and proved successful in modeling PUF with machine learning, and especially deep learning modeling techniques. However, at first hand, these modeling techniques have been introduced as potential tools for attackers that try to model-build a clone of a PUF instance [17]. For instance, it has been proved that a simple Multi-Layer Perceptron model can be trained to predict the response of several architecture of arbiter PUFs, by being trained with just a very small set of CRPs of the PUF-enabled device. For this reason, many variations and countermeasures are proposed for every family of PUF architectures (such as double k-XOR arbiter PUF [18]) with the goal of avoiding model-building attacks using machine learning and deep learning techniques.

However, the potential of using modeling techniques in favor of PUF enrollment and authentication is promising, to save storage and time. Indeed, replacing the PUF secret database by a trained model present several advantages. It would make practical (in terms of enrollment time and database size) security applications (e.g., protocols, authentication mechanisms) requiring a large amount of PUF secret data. The enrollment needs indeed to focus only on a subset of the data and then the trained model should be capable of generating any data that can provide the given PUF. Moreover, the PUF model could also be enhanced by additional variables and parameters related to environmental conditions or the aging of the integrated circuit. These additional properties should make possible the self-adjustment of the PUF responses during the mission mode in order to increase the reliability of the protocols and algorithms using the PUF.

To realize this potential, a re-introduction of PUF modeling to facilitate PUF utilization is required. Neural Network and deep learning in specific, are models and techniques that can introduce fundamental changes in the outcome. Commonly Neural Network models are potential to handle great deal of noise and complex data models, and they can achieve optimal behavior with acceptable amount of data, if properly trained. Their use cases are for many different purposes mainly including classification and pattern recognition. Their use case in the field of PUF modeling can greatly affect the storage and time of enrollment compared to those in the traditional methods.

Replacing the database by a model induce a serious security threat. Indeed, since during enrollment the trusted party is able to model the PUF, this naturally induces that this PUF is vulnerable to modeling attack. Even worst, this methodology requires to use easy to model PUF in order to guarantee the correctness of the model. Nevertheless, solutions already exist, which embed at system level countermeasures against modeling attack. Fig. 6 shows a generic scheme which allows the modeling of the PUF by a trusted party during the enrollment, and which impedes the modeling attacks in mission mode. During enrollment, the PUF is accessed in order to get enough data to model it. Then when the PUF is in mission mode (deployed within the system), the PUF data exchanged with the external world are altered to prevent the modeling. This alteration (the

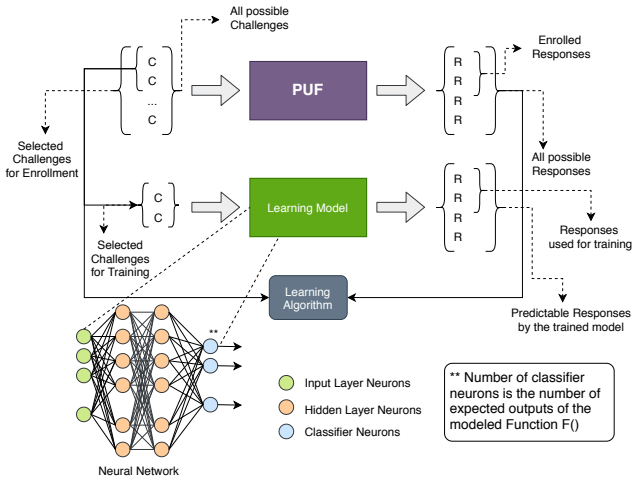


Figure 5. Deep Learning based PUF Enrollment.

so-called poisoning data) modifies the PUF responses (based on a secret protocol shared between the server and the device) so that it makes not possible to correctly model the PUF.

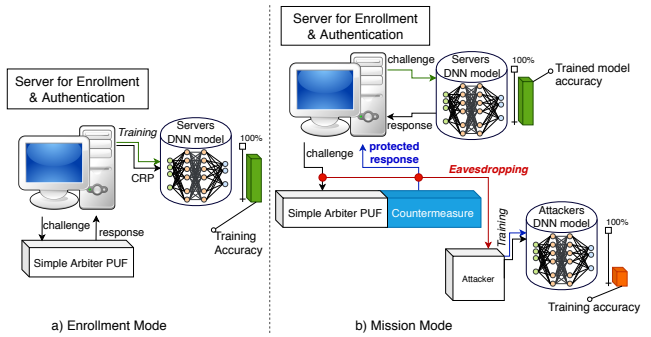


Figure 6. Deep Learning based PUF Management.

V. On line Test to check the PUF Integrity and Security in mission mode

The integrity and security of PUF can be seriously compromised either by harsh environment, aging or malevolent actions as fault injection attacks (FIA) or Side-Channel Analysis (SCA). Regardless the security requirement, the PUF has to use a post-processing block to enhance its reliability which is natively very low, around 3 to 10% of Bit Error Rate [19]. For cryptographic key generation, an efficient post-processing block called "fuzzy extraction" has been proposed by Dodis et al. [20]. It exploits Error Correcting Codes (ECC) and generates a public word, the "Helper Data" during the enrollment phase in order to correct errors in mission mode. Another solution is to filter out unreliable bits as proposed by Schaub et al. in [6], [7]. The PUF can be jeopardized either on its native structure and/or at the post-processing blocks, thus creating a serious integrity problem. In the following subchapters are presented first a generic fault detection and some specific solutions to reduce the impact of aging, FIA and SCA.

A. A generic protection to detect integrity Problems in PUFs: the digital sensor

The loss of integrity or FIA are caused or reflected in transient environmental changes, like temperature and voltages. A digital sensor able to detect a significant change in propagation time can be particularly efficient for PUFs and notably delay-PUFs as this latter relies on delay or frequency measurements. Digital sensors (DS) consist of a delay chain inserted near the senstive primitive to check, as the PUF [21] [22, Fig. 14,page 189]: in case the chip is operated in abnormal conditions, setup time violations occur in the first place on the digital sensor's intentionally long path. In order to characterize the amplitude of the timing violation, the delay chain is sampled in many places. Such a snapshot allows to digitize the amount of stress applied to the circuit, as the change in temperature/voltage or a FIA. Figure 7 represents the digital sensor architecture. In this example, the sensor includes a chain of 64 buffers. The last 33 buffers in this chain each feeds an individual flip-flop. The sensor outcome would be the output of this set of flip-flops. All flip-flops are operating under the same clock signal at frequency: F. In addition, the first buffer is fed with a toggle flip-flop generating a periodic signal $a0$ working at $F/2$. The clock frequency should be determined precisely such that when the circuit is fed with $a0$, the first half of flip-flops are in phase A (say $0 \rightarrow 1 \rightarrow 0$) and the second half in the complementary phase $\overline{\text{A}}$ (say $1 \rightarrow 0 \rightarrow 1$) . Note that the clock frequency is determined for the nominal condition (i.e., Power Voltage Vdd = 1.2V and Temperature = 25°C) for a new device. Figure 8 shows the waveforms representing the

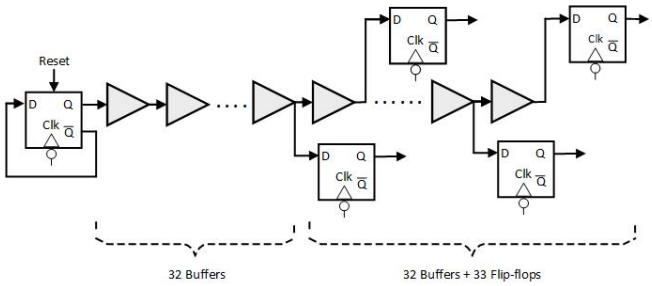
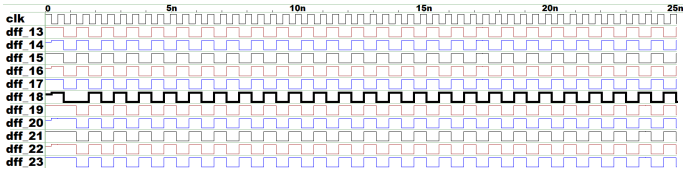


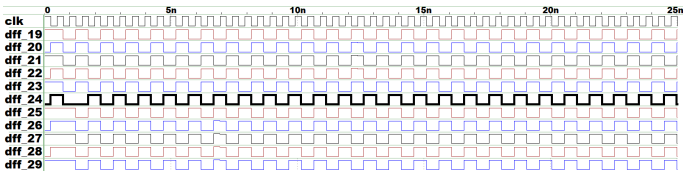
Figure 7. Architecture of the deployed digital sensor.

flip-flop outputs in different conditions. In particular, Fig. 8(a) depicts the flip-flop outputs in the nominal condition of the new (no-aged) sensor. As shown, the first phase change occurs in the 18th flip-flop (*dff_18*), the first 17 flip-flops have the same phase A and the last 16 flip-flops are in complementary phase $\overline{\text{A}}$. This trend is changed when the circuit operates under different voltage/temperature or aging conditions. For instance, Fig. 8(b) represents the sensor outcome when the temperature is 0°C. In this case, the first change occurs in the 24th flip-flop (*dff_24*).

The characterization scheme is to detect the first flip-flop that experiences a change in its output (compared to its prior flip-flop) at every clock cycle of CC_i. The index of that flip-flop is referred to as Flip-flop Number (FN_i). Then the average of all FN_is over all clock cycles is calculated. This average, so-called Average value of Flip-flop Number (AFN) is used



(a) Voltage = 1.2V, Temperature = 25°C, Age = 0



(b) Voltage = 1.2V, Temperature = 0°C, Age = 0

Figure 8. Flip-flop outputs in different conditions.

for characterization. To detect an abnormal change or FIA, the AFN is calculated at runtime and compared with the nominal AFN. In case of a mismatch, an alarm is fired. The digital sensor has to be placed near the PUF in case of high spatial resolution as laser shots.

In [23], the digital sensor has been tested using 45-nm NANGATE technology [24]. Synopsys HSpice has been used for the simulations. Both sensor and S-Box outputs were extracted under different voltage and temperatures, i.e., temperatures between $-10°$C and $150°$C with $1°$C steps, and for the voltage (Vdd) between $0.65V$ to $1.4V$ with $0.05V$ steps. The AFN is extracted based on environmental conditions (from *worst* to *best*) in temperature and voltage. To take the effect of process into account, the threshold value of AFN is calibrated after fabrication, and is stored locally in a One-Time Programmable (OTP) memory to be used as a reference during the chip lifetime. Figure 9 shows the AFN in different voltage/temperature combinations. As expected, AFN is lower for the conditions in which the underlying circuit operates slower, i.e., in low voltages and high temperatures, while its value increases by moving towards lower temperatures and higher voltages.

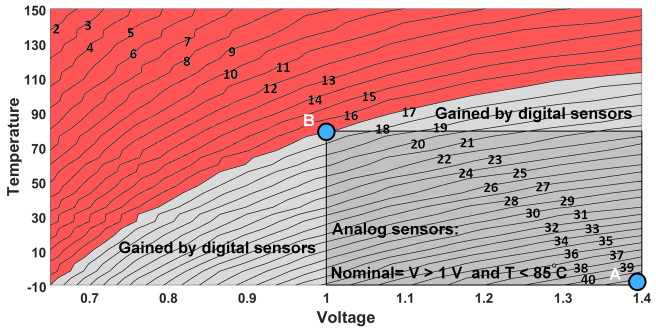


Figure 9. AFN variation in different voltage and temperature pairs.

B. Specific protections to enhance integrity and security in PUFs

These protections take advantage of specific structures of PUFs like SRAM-PUF [25], RO-PUF [26] and Loop-PUF [27], [28]. We present here some solutions to enhance the integrity to face Aging and Side-Channel Analysis.

1) Aging: CMOS aging is a cause of reliability decrease over time. The two main factors of aging in CMOS technology are Negative Bias Temperature-Instability (**NBTI**) and Hot-Carrier Injection (**HCI**) [29]. Both aging sources result in increasing switching and path delays. Theses phenomena cause a Silicon Oxide wearout between the gate and the conduction channel. It has been shown in [30] that the PUF based on combinatorial logic, as delay-PUFs, are not significantly impacted, as the increase in delay caused by aging is partly compensated by the differential measurements. However, memory elements like latches, D-FF or SRAM cell are more impacted. This is mainly due to the NBTI phenomenon. Considering an SRAM cell, the threshold voltage V_{Th} of the PMOS transistors of the two inverters continue to change with age. More precisely the difference $V_{Th1} - V_{Th2}$ of the PMOS transistors tends to decrease and makes the PUF less reliable.

A simple anti-aging protection specific to SRAM PUF has been proposed by Mael et al. in [31]. It relies on storing the inverse of the initial value at power up. This solution still requires ECC to get a high level of reliability, but avoid to increase of errors, which on the contrary decreases over age. This study also shows that the SRAM zeroization for security reasons is not recommended to face aging.

2) Side-Channel Analysis (SCA): Many SCA have been proposed on PUFs. Several semi-invasive attacks relying on focused ion beam or photonic emissions have been carried out on SRAM PUF and Arbiter PUF [32], [33]. But the most common and low-cost attacks use electromagnetic (EM) or Power observation on Delay-PUF. This is particularly efficient for Ring-Oscillator-based PUFs as RO-PUF and Loop-PUF. Merli et al [34] have notably showed that ring oscillator frequencies from simultaneously activated ring oscillators can be identified and exploited. A proposed countermeasure is to measure multiple, more than two, ring oscillators at the same time. But multiplexers and frequency counters exhibit leakage about the ring oscillator frequencies that can be resolved spatially. To impede the attack on counters and multiplexers, measurement path randomization, using different counters or multiplexers for each evaluation, and interleaved placement of the components are proposed [35]. However, depending on the measurement position on the decapsulated die, the counter frequencies have different amplitudes and can be distinguished.

A simpler countermeasure against SCA for RO-PUF and Loop-PUF is to use temporal masking [36]. The main principle is to perform a single measurement at a time, the order being defined by a random sequence depending on a random variable called "mask". For instance the Loop-PUF which implements a single instance of the primitive, requires two measurements that will be carried out according to the mask value. This methods also applied to other RO-based PUFs. However, special care should be taken not to use the same RO for different challenges, because the attacker could identify when the same frequencies occur in time and thereby deduce which RO they belong to.

VI. Conclusion and Perspectives

PUFs constitute very interesting objects, as attested by their wide adoption by the industry. Still, many of their features can be improved. First and foremost, their reliability is key, which requires therefore high efforts in their test. This implies proper configuration before being put on the market. Operations such as enrollment shall be carried out cautiously and efficiently. In operational mode, the PUFs must remain evolvable and managed versus changing conditions, aging and potential attacks. Digital Sensors represent a generic protection against FIA, as it is able to sense abnormal time evolution which is well appropriate for delay-PUFs. Some protections specific to the architecture can be devised to avoid the loss of integrity over time. For instance simple anti-aging for SRAM consist in reprogramming the RAM, or the frequency measurement of RO-PUF and Loop-PUF can be carried out by using temporal masking.

We nonetheless underline that PUF usage is still ad hoc nowadays. Indeed, there is an adherence of the service rendered by the PUF on the PUF architecture. Following the roadmap of test community, tools have been developed to bring the gap between different approaches. Indeed, it is in the global interest to have interoperability and agnostic methods across different PUF technologies. Testing infrastructures (on-chip and off-chip) have been developed over the year with this goal of interoperability and efficiency, the PUF management through the life cycle can benefit from these techniques by adapting them to its specific characteristics. Moreover what matters in the end is the trust in the system-level service (authentication, security, etc.). In particular, a need for agility is important, as underlying technology PUF might evolve. Fortunately, machine-learning techniques for enrollment allow flexibility, which make up for PUF objects variability. Similarly, machine-learning techniques allow to accommodate for a given PUF technology to evolve as a function of attack state-of-the-art improvement and mitigation techniques that are implemented accordingly.

References

[1] S. Ali, D. Mukhopadhyay, and M. Tunstall, "Differential fault analysis of AES: towards reaching its limits," *J. Cryptographic Engineering*, vol. 3, no. 2, pp. 73–97, 2013.

[2] S. Guilley, S. Hamaguchi, and Y. Kang, "ISO/IEC NP 20897. Information technology – Security techniques – Security requirements, test and evaluation methods for physically unclonable functions for generating nonstored security parameters," https://www.iso.org/standard/76353.html.

[3] N. Bruneau, J. Danger, A. Facon, S. Guilley, S. Hamaguchi, Y. Hori, Y. Kang, and A. Schaub, "Development of the Unified Security Requirements of PUFs During the Standardization Process," in *Innovative Security Solutions for Information Technology and Communications - 11th International Conference, SecITC 2018, Bucharest, Romania, November 8-9, 2018, Revised Selected Papers*, ser. Lecture Notes in Computer Science, J. Lanet and C. Toma, Eds., vol. 11359. Springer, 2018, pp. 314–330. [Online]. Available: https://doi.org/10.1007/978-3-030-12942-2_24

[4] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, "Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs," *Reconfigurable Computing and FPGAs, International Conference on*, vol. 0, pp. 298–303, 2010.

[5] M. J. Pelgrom, A. C. Duinmaijer, and A. P. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid State Circuits*, vol. 24, no. 5, pp. 1433–1439, 1989, DOI: 10.1109/JSSC.1989.572629.

[6] A. Schaub, J. Danger, S. Guilley, and O. Rioul, "An Improved Analysis of Reliability and Entropy for Delay PUFs," in *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018*, M. Novotný, N. Konofaos, and A. Skavhaug, Eds. IEEE Computer Society, 2018, pp. 553–560. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/DSD.2018.00096

[7] J. Danger, S. Guilley, and A. Schaub, "Two-Metric Helper Data for Highly Robust and Secure Delay PUFs," in *IEEE 8th International Workshop on Advances in Sensors and Interfaces, IWASI 2019, Otranto, Italy, June 13-14, 2019*. IEEE, 2019, pp. 184–188. [Online]. Available: https://doi.org/10.1109/IWASI.2019.8791249

[8] C. Carlet, "Vectorial Boolean Functions for Cryptography: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering." Cambridge University Press, Y. Crama and P. Hammer eds, 2010, pp. 398–469, preliminary version available at // www.math.univ-paris13.fr/~carlet/chap-vectorial-fcts-corr.pdf.

[9] W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators," Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn, September 2011, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile.

[10] D. Hely, M. . Flottes, F. Bancel, B. Rouzeyre, N. Berard, and M. Renovell, "Scan design and secure chip," in *Proceedings. 10th IEEE International On-Line Testing Symposium*, July 2004, pp. 219–224.

[11] D. Hély, K. Rosenfeld, and R. Karri, "Security challenges during vlsi test," in *2011 IEEE 9th International New Circuits and systems conference*, June 2011, pp. 486–489.

[12] L. Whetsel, "Inevitable use of TAP domains in SOCs," in *Proceedings. International Test Conference*, 2002, pp. 1191–.

[13] S. K. Goel and E. J. Marinissen, "Effective and efficient test architecture design for SOCs," in *Proceedings. International Test Conference*, 2002, pp. 529–538.

[14] D. Hely, F. Bancel, M. . Flottes, and B. Rouzeyre, "a secure scan design methodology," in *Proceedings of the Design Automation Test in Europe Conference*.

[15] M. Da Silva, E. Valea, M. Flottes, S. Dupuis, G. Di Natale, and B. Rouzeyre, "Encryption of test data: which cipher is better?" in *2018 14th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, July 2018, pp. 85–88.

[16] ——, "A new secure stream cipher for scan chain encryption," in *2018 IEEE 3rd International Verification and Security Workshop (IVSW)*, July 2018, pp. 68–73.

[17] J.-Q. Huang, M. Zhu, B. Liu, and W. Ge, "Deep learning modeling attack analysis for multiple FPGA-based APUF protection structures," in *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, pp. 1–3, ISSN: null.

[18] M. Khalafalla and C. Gebotys, "PUFs deep attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs," in *2019 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 204–209, ISSN: 1530-1591.

[19] S. Katzenbeisser, U. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon," in *CHES 2012*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7428, pp. 283–301. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33027-8_17

[20] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *EUROCRYPT*, 2004, pp. 523–540.

[21] N. Selmane, S. Guilley, and J.-L. Danger, "Setup Time Violation Attacks on AES," in *EDCC, The seventh European Dependable Computing Conference*, Kaunas, Lithuania, May 7-9 2008, pp. 91–96, ISBN: 978-0-7695-3138-0, DOI: 10.1109/EDCC-7.2008.11.

[22] N. Selmane, S. Bhasin, S. Guilley, and J.-L. Danger, "Security evaluation of application-specific integrated circuits and field programmable gate arrays against setup time violation attacks," *IET Information Security*, vol. 5, no. 4, pp. 181–190, December 2011, DOI: 10.1049/iet-ifs.2010.0238. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-82055176045&partnerID=40&md5=06115dddc18329ee1ff40f6af025c9ce

[23] T. H. Anik, R. Saini, J.-L. Danger, S. Guilley, and N. Karimi, "Failure and attack detection by digital sensors," in *European Test Symposium*. IEEE, 2020.

[24] "Nangate 45nm open cell library," "http://www.nangate.com".

[25] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, "Brand and ip protection with physical unclonable functions," in *ISCAS*, 2008, pp. 3186–3189.

[26] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007*. IEEE, 2007, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/1278480.1278484

[27] Z. Cherif, J. Danger, S. Guilley, and L. Bossuet, "An easy-to-design PUF based on a single oscillator: The loop PUF," in *15th Euromicro Conference on Digital System Design, DSD 2012, Çeşme, Izmir, Turkey, September 5-8, 2012*. IEEE Computer Society, 2012, pp. 156–162. [Online]. Available: http://dx.doi.org/10.1109/DSD.2012.22

[28] Z. Cherif, J.-L. Danger, F. Lozac'h, Y. Mathieu, and L. Bossuet, "Evaluation of Delay PUFs on CMOS 65 nm Technology: ASIC vs FPGA," in *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, ser. HASP '13. New York, NY, USA: ACM, 2013, pp. 4:1–4:8. [Online]. Available: http://doi.acm.org/10.1145/2487726.2487730

[29] F. Oboril et al., "Extratime: Modeling and analysis of wearout due to transistor aging at microarchitecture-level," in *DSN*, 2012, pp. 1–12.

[30] N. Karimi, J. Danger, and S. Guilley, "Impact of Aging on the Reliability of Delay PUFs," *J. Electronic Testing*, vol. 34, no. 5, pp. 571–586, 2018. [Online]. Available: https://doi.org/10.1007/s10836-018-5745-6

[31] R. Maes and V. van der Leest, "Countering the effects of silicon aging on SRAM PUFs," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014*. IEEE Computer Society, 2014, pp. 148–153. [Online]. Available: http://dx.doi.org/10.1109/HST.2014.6855586

[32] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning Physically Unclonable Functions," in *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on*, 2013, pp. 1–6.

[33] S. Tajik, E. Dietz, S. Frohmann, J. Seifert, D. Nedospasov, C. Helfmeier, C. Boit, and H. Dittrich, "Physical Characterization of Arbiter PUFs," in *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, L. Batina and M. Robshaw, Eds., vol. 8731. Springer, 2014, pp. 493–509. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3_27

[34] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, "Semi-invasive EM attack on FPGA RO PUFs and countermeasures," in *Proceedings of the Workshop on Embedded Systems Security*, ser. WESS '11. New York, NY, USA: ACM, 2011, pp. 2:1–2:9. [Online]. Available: http://doi.acm.org/10.1145/2072274.2072276

[35] D. Merli, J. Heyszl, B. Heinz, D. Schuster, F. Stumpf, and G. Sigl, "Localized electromagnetic analysis of RO pufs," in *2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013, Austin, TX, USA, June 2-3, 2013*, 2013, pp. 19–24.

[36] L. Tebelmann, J.-L. Danger, and M. Pehl, "Self-Secured PUF: Protecting the Loop PUF by Masking," in *Constructive Side-Channel Analysis and Secure Design - 10th International Workshop, COSADE*, 2020.