A survey on silicon PUFs

Fahem Zerrouki, Samir Ouchani, Hafida Bouarfa

A Survey on Silicon PUFs

Fahem Zerrouki[a,*,1], Samir Ouchani[b,2] and Hafida Bouarfa[a,1]

[a] Université Blida 1, Laboratoire LRDSI ,Faculté des Sciences, BP 270, Route de Soumaa, Blida, Algérie
[b] Lineact CESI, Aix-en-Provence, France

ARTICLE INFO

Keywords:
Physical Unclonable Function
Silicon PUFs
Performance
Security
Cryptographic Key Generation
Authentication

ABSTRACT

Integrated Circuits (ICs) and electronic devices have become an integral part of daily human life (mobile, home, car, etc.). However, specific security measures should be taken to protect the communicated information to and from these devices. However, the existing conventional security primitives require large amounts of memory capacity, processing power, and energy resources that contradict the specific nature of devices. On the other hand, they store secret keys on the devices for future use, making them vulnerable to physical attacks. A new concept, known as Physically Unclonable Functions (PUFs), has been recently investigated to mitigate this problem. A PUF is a hardware-specific security primitive uses the randomness found in the disorder of physical media caused by the manufacturing variation process to provide cryptographic functionalities. Consequently, PUFs are inexpensive to fabricate, prohibitively challenging to duplicate, admit no compact mathematical representation, and are intrinsically tamper-resistant. This manuscript gives a complete survey of PUFs as a promising research field in security with a wide application, especially with connected devices. First, we motivate our contribution by comparing it with the existing surveys about PUFs. Then we provide the needed background to understand PUF architectures and applications by covering: the variability and randomness concepts, their classes, and properties. Then, we survey the existing initiatives related to silicon PUFs in terms of implementation and design used to extract unique secret information from the physical characteristics of an integrated circuit. In addition, we compare the surveyed works in terms of performance and security. Furthermore, we classify and compare the existing silicon PUF applications and use cases. Before concluding, we give the principal metrics used to evaluate the PUFs' performance and present some related attacks. Finally, we talk about the current limitations of silicon PUF architectures and applications, and we look at and talk about research opportunities and major trends.

1. Introduction

Nowadays, information security is taking a great interest in the information technology field [97]. However, many researchers and developers are working on making programs more secure and reliable, facing people who want to use leaks in an unethical way and breaking (or trying to break) primarily used one-way functions such as MD5, SHA, or RSA-based algorithms. Hash functions take, in general, a string as input and produce a hash code (fingerprint) with a fixed length that is difficult to interpret by a human. These deterministic functions may choose specific operations based on computed values from the source string or the computational environment. These functions are facing many challenges: the possibility of breaking or reverse-engineering codes and also factoring the product of two large prime numbers can be accomplished in polynomial time on a quantum computer [101]. Furthermore, another practical challenge goes beyond the cost and packaging constraints of hashing functions, especially in embedded authentication and identification hardware technologies. On the other hand, electronic devices have become increasingly used in our everyday lives, especially in the IoT field. Traditionally, the secret keys used by those devices as a unique identifier are embedded immediately after manufacturing into the integrated circuits in non-volatile memory, making them vulnerable to many kinds of attacks, such as invasive, semi-invasive, and side-channel attacks [37]. An attacker could steal the secret key or make a full copy of the device and use it in identity theft attacks. On the other hand, it is expensive, complicated, and impossible to avoid these attacks with classical cryptography systems based on a secret binary key. Hence, a more attractive alternative has recently become a hot topic in research and development that relies on the physical disorder by giving birth to the *Physical Unclonable Functions* (PUFs) [100].

A PUF is a one-way function that is derived from the behaviour of a complex physical object. A corresponding response (output) will be generated when a challenge (input) is presented to a PUF. The response is determined by a complex physical function unique to each device, and it is impossible to duplicate because they have uncontrollable physical parameter variations that occur during hardware device manufacture. Nowadays, PUFs are widely used in identification and authentication.

Due to the physical disorder of integrated circuits (ICs) caused by the manufacturing process during their fabrication. Silicon PUFs [65] are one of the most proposed and discussed PUF classes to generate a unique digital signature used as the fingerprint of an IC. This initiative aims to survey the existing categories and contributions related to silicon PUS. In-depth, we survey, study, and compare the state-of-the-art related to the challenges mentioned above. We split our review into the following six directions:

1. Studying and comparing the existing reviews and sur-

* Corresponding author
✉ ze.fahem@gmail.com (F. Zerrouki); souchani@cesi.fr (S. Ouchani); hafidabouarfa@hotmail.com (H. Bouarfa)
ORCID(s): 0000-0002-9877-413X (F. Zerrouki)

veys about PUFs.
2. Detailing the needed background to understand PUFs.
3. Surveying the recent silicon PUF architectures and applications.
4. Showing how to evaluate PUF's performance and presenting different attacks related to PUFs.
5. Comparing the studied contributions and presenting the prominent research directions related to silicon PUFs.

This paper is organized as follows : First, Section 2 presents and compares the existing surveys on PUFs. Then, Section 3 presents the needed background that allows the use of the PUF as a security primitive and also describes the PUFs as well as their properties and classes. In Section 4, we survey the existing silicon PUF architectures and compare them by considering their performance and security. After that, section 5 classifies and compares the existing Silicon PUF applications. Furthermore, Section 6 presents the primary metrics used to evaluate the PUF's performance and lists the possible PUF attack scenarios. Finally, Section 7 summarizes this survey with critical remarks and enumerates a list of promising research directions in Physical Unclonable Functions.

2. A Review of Existing Surveys

This section presents an overview of the existing surveys and reviews about PUFs and their related research and application areas. A comparison is given in Table 1 by considering ten criteria: *background, application, evaluation, properties, attacks, classes, type, error correction, comparison* and *the surveyed PUFs*.

- *Background* indicates if the survey gives the needed background to understand the PUF technologies.
- *Application* shows if PUF applications and related use cases are mentioned.
- *Evaluation* checks if the performance evaluation metrics were detailed within the survey.
- *Properties* indicate if the PUF characteristics are considered as a feature to understand and/or compare the surveyed contributions.
- *Attacks* show if PUF-related attacks are presented and included within the cited study.
- *Surveyed PUFs* quantify how many papers were studied in the selected survey.
- *classes* check if the mentioned survey classifies silicon PUFs into delay-based, memory-based, and analog electronic PUFs.
- *Type* specifies if the type of PUF, weak or strong, is included within the comparison.
- *Error correction* is to show if error detection and correction, and noise elimination solutions were presented and considered for comparison.
- *Comparison* indicates if the survey gives a comparison of the reviewed papers.

In the literature, several surveys and reviews are presented [83], most of them focusing on the PUF terminologies, architectures, applications and attacks. McGrath et al.[83] gave a PUF taxonomy and summarized the existing PUF implementations from 2000 to 2017. Zhang et al.[144] selected some examples of the existing silicon and non-silicon PUFs, especially RO PUFs, and presented some evaluation criteria and related attacks. Maes and Verbauwhede[75] surveyed and compared a selection of nine papers published between 1992 and 2007 in terms of properties, evaluations, and attacks. Herder et al.[40] presented a survey to categorize PUFs into strong and weak classes, then showed their main application areas. Also, Rührmair and Holcomb[108] provided a brief overview of PUFs. Then, they discussed security features, implementations, attacks, protocol uses, and the applications of weak and strong PUFs. van Dijk and Rührmair [22] presented a brief survey about attacks and countermeasures of strong PUFs protocols. After presenting PUF related background, Liang et al. [64] summarized the existing PUF dedicated to intellectual property (IP) protection. Chang et al.[16] compared strong and weak PUFs published between 2002 and 2017, and also presented their weaknesses, vulnerabilities, and sources of variation. In [94] Ning et al. gave an in-depth review of non-silicon and silicon-based PUF by considering the architecture, applications, requirements, and challenges of PUF that provide security solutions. Delvaux et al. [21] have analysed the proposed PUF-based authentication protocols between 2001 and 2014. Gao et al. [30] presented a survey on recent emerging nanotechnology based PUFs.

Maiti [79] surveyed the existing methods to evaluate and compare the performance of PUFs. In [2], Adames et al. gave a brief review of PUFs regarding CMOS compatibility by comparing them in terms of PUF properties. Noor et al. [96] presented a review that categorizes defense mechanisms against machine learning modeling attacks (ML-MA) on strong PUFs for IoT authentication. Al-Haidary and Nasir [3] presented a brief review that includes seven schemes of PUFs and four types of attacks. Papakonstantinou and Sklavos [99] provided a brief survey of the existing PUF schemes. Also, Joshi et al. [48] summarized the basic concepts, applications, and architecture behind PUFs. Gebali and Mamun [32] gave a review of the common types of PUFs, discussed their performance, and reviewed some PUF-based algorithms that can provide stable authentication and secret key generation.

The comparison presented in Table 1 shows that most of the discussed work gave a brief or non-complete survey regarding the selected criteria. In this survey, we try to fill the gap identified within this comparison by presenting in-depth the needed backgrounds and different PUF terminologies, PUFs use cases, the metrics used to evaluate PUFs, the properties of PUFs, the attacks proper to PUFs, the existing error correction techniques, and finally, we survey and compare the existing contributions related to silicon PUFs.

Table 1
Summary of reviews and surveys about PUFs .

	Work	Year	Background	Application	Evaluation	Properties	Attacks	Surveyed PUF	classes	Type	Error Correction	Comparison
1	Maes and Verbauwhede[75]	2010	●	◐	○	●	○	◐	●	○	◐	●
2	Maiti[79]	2012	○	○	●	○	○	○	◐	○	◐	○
3	Zhang et al.[144]	2014	○	◐	○	◐	◐	◐	●	●	○	○
4	Herder et al.[40]	2014	◐	◐	○	○	◐	◐	○	◐	◐	○
5	Rührmair and Holcomb[108]	2014	◐	◐	○	○	◐	◐	○	◐	○	○
6	van Dijk and Rührmair[22]	2014	○	○	○	○	◐	◐	○	○	○	○
7	Delvaux et al.[21]	2015	◐	◐	○	○	◐	◐	○	◐	●	●
8	Gao et al.[30]	2016	◐	◐	◐	○	◐	◐	◐	●	○	●
9	Liang et al.[64]	2016	◐	◐	◐	◐	◐	◐	●	○	○	○
10	Adames et al.[2]	2016	◐	○	◐	◐	○	◐	●	○	○	●
11	Chang et al.[16]	2017	◐	◐	◐	◐	◐	◐	○	◐	○	○
12	Noor et al.[96]	2017	◐	○	○	◐	◐	◐	○	●	○	●
13	Joshi et al.[48]	2017	◐	◐	●	◐	○	◐	◐	◐	○	○
14	Papakonstantinou and Sklavos[99]	2018	◐	◐	●	○	◐	◐	●	○	○	●
15	Al-Haidary and Nasir[3]	2019	○	○	○	○	●	◐	●	○	○	○
16	McGrath et al.[83]	2019	◐	◐	○	○	○	●	●	◐	○	●
17	Ning et al.[94]	2020	◐	●	◐	◐	●	●	●	●	○	●
18	Anandakumar et al.[7]	2021	◐	◐	◐	●	●	●	●	●	○	●
19	Gebali and Mamun[32]	2022	◐	○	◐	○	○	◐	○	○	○	●
	Proposed work	-	●	●	●	●	●	●	●	●	●	●

In the giving comparison, ○ means that the survey does not consider the indicated criteria, whereas ● means the inverse. Also, ◐ means that the authors of the cited work considered only a part of the criteria.

3. PUFs background

Some of the most common terms and measurements that describe PUFs are shown in this section. They help us understand PUFs and how they work.

3.1. Physical disorder

Physical disorder refers to the random imperfections found in the structure of physical objects. This phenomenon is typically observed at the nano-scale level of the physical objects' structures. Many fascinating randomnesses exist around us, taking various forms such as biological, physical, chemical entities, and so on, caused by nature or any manufacturing process [37].

As a naturally physical disorder example, the surface with three-dimensional random structures of a coffee bean is presented as a microscopic image as shown in Figure 1. (a). Figure 1. (b) represents the microscopic image of a biological physical disorder example of a human tooth. Figure 1. (c) shows the irregular structure of the metal conductors in a semi-conductor chip fabricated using 90 nm technology.

This physical disorder is unique to each object and is hard or impossible to replicate, and it can be used as an identity for this object or the device embedded in it.

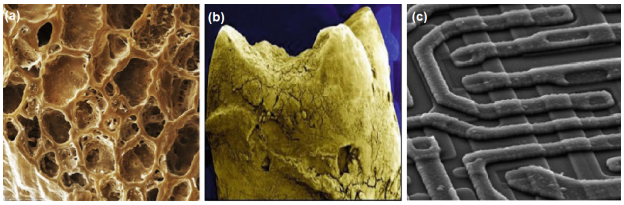


Figure 1: Examples of physical disorder: (a) a coffee bean, (b) a tooth, and (c) an integrated circuits [37].

3.2. Manufacturing Variation

As a main principle, the manufacturing process of any product should be identical in its shape and structure to the needed product design. However, this is not the case in most modern chips and integrated circuits due to the manufacturing process and continuous scaling of semiconductor technologies [37].

The manufacturing process variability is affected mainly by four factors: physical geometric structure, internal material parameters, interconnect geometry, and interconnect material structures [37]. The impact of variability on the electrical parameters of very large-scale integrated (VLSI) circuits is expected to be significant. Figure 2 shows the magnitude of variation in device threshold voltage (Vth) and the performance of VLSI circuits. We observe that the impact of variations on threshold voltage increases significantly compared to the performance evaluation of the VLSI circuit, which endangers the stability of the circuit operations. However, these variations can be exploited to design a physically unclonable function.

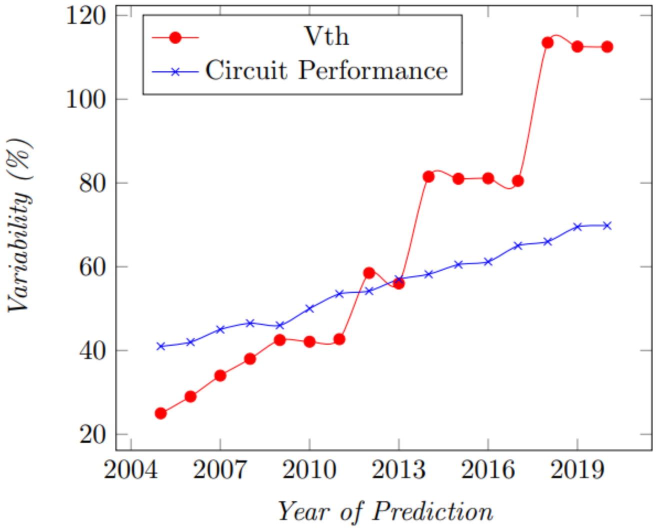


Figure 2: The impact of variability on the electrical parameters of VLSI circuits [37].

3.3. Challenges and Responses

Challenges are entries given as inputs to an instance of a PUF. When a challenge stimulates a device where an instance of a PUF is embedded, the latter will interpret it in its internal system using the complex physical function unique to each device or PUF instance. Then, the PUF will produce unpredictable but repeatable data, called a response. The PUF's design determines the forms of the challenges and responses.Also, as a PUF is derived from the concept of one-way function, it should be impossible to revert the system, meaning that an adversary cannot predict a response as an entry to find the original challenge or vice versa. Finally, as a PUF will always produce the same response to a given challenge, we will talk about the Challenge-Response Pair (CRP), representing the link between a challenge and its response [142].

However, the CRP will change if we build another instance of a PUF (meaning that we take the same design and the same blueprints but build another one with random components and in another environment). Indeed, the way the PUF works is always the same, but due to the manufacturing variation, its internal components are never identical, causing each PUF to (ideally) always produce different responses compared to other instances. This uniquely allows the PUF to play the role of a perfect identification system, where the set of CRPs is the fingerprint of the PUFs or the device embedded in [30]. The particular dependence of responses on physical parameters and challenges for a given PUF was generally called the challenge-response behaviour of that PUF [132]. Figure 3 shows the PUF's challenge-response behaviour.

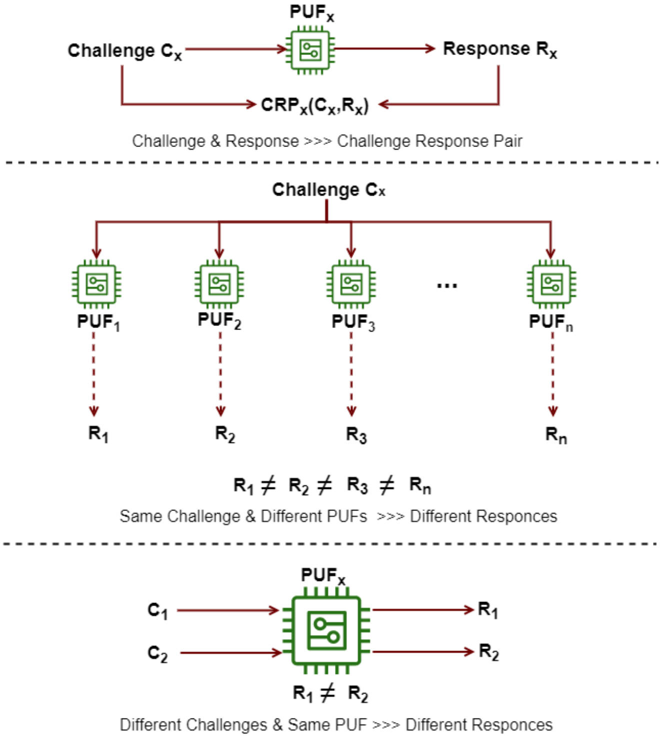

PUF$_x$
Challenge C$_x$
Response R$_x$
CRP$_x$(C$_x$,R$_x$)
Challenge & Response >>> Challenge Response Pair
Challenge Cx
PUF$_1$
PUF$_2$
PUF$_3$
...
PUF$_n$
R$_1$
R$_2$
R$_3$
R$_n$
R$_1$ ≠ R$_2$ ≠ R$_3$ ≠ R$_n$
Same Challenge & Different PUFs >>> Different Responces
PUF$_x$
C$_1$
C$_2$
R$_1$
R$_2$
R$_1$ ≠ R$_2$
Different Challenges & Same PUF >>> Different Responces

Figure 3: The challenge response behaviour.

3.4. Intra-distance

The intra-distance, also called intra-chip or intra-die of a PUF response, is described by a random variable describing the distance between two responses from the same PUF instance and the same challenge [105]. By taking two evaluations $R_i(c)$ and $R'_i(c)$ of the same PUF instance i and the same challenge c, let dist [. , .] to be any distance metric over the response set R, the intra-distance of a PUF i is given by Equation 1 [105].

$$Intra-distance_i \overset{\Delta}{=} dist\left[R_i(c), R'_i(c)\right] \tag{1}$$

In this survey, responses are always considered as bit vectors, and the hamming distance (HD) is used as a distance metric. Therefore, Equation 1 will be:

$$Intra-distance_i \overset{\Delta}{=} HD\left[R_i(c), R'_i(c)\right]$$

For a range of [0,1], when the $Intra-distance_i$ result is close to "zero", that means the PUF is highly reliable. Conversely, if the result is close to "one, " the PUF is least reliable. This, due to the environmental conditions under which responses are generated, such as temperature variation and supply voltage [72]. Where the intra-distance between two responses generated from the same challenge with the same PUF instance under the same environmental condition is less than the intra-distance between the same responses generated under two different conditions [76]. Figure 4 shows the intra-distance of a PUF.

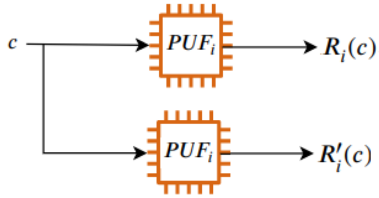


Figure 4: Symbolic representation of intra-distance of a single PUF.

3.5. Inter-distance

The inter-distance, inter-chip, or inter-die of a PUF response is described by a random variable [105]. It is the distance between two responses generated by two different PUF instances, PUF_i and PUF_j, stimulated by the same challenge c. For two responses $R_i(c)$ and $R_j(c)$ of two different PUF instances, i and j, for the same challenge c; Equation 2 uses HD as a distance metric to measure the inter-distance of $R(c)$.

$$Inter - distance_{R(c)} \triangleq HD\left[R_i(c), R_j(c)\right] \tag{2}$$

If the result of Equation 2 is close to "zero" for a range of [0,1] that means the PUF is less unique. Conversely, if the result is close to "one" the PUF is highly unique. The inter-distance between PUF responses is also susceptible to variations in environmental conditions. Figure 5 depicts the inter-distance of a PUF.

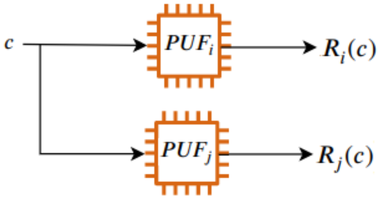


Figure 5: Symbolic representation of inter-distance between two PUFs.

3.6. Environmental effects

In addition to the manufacturing process, which makes the integrated circuits physical disorder objects, environmental variation or variability in the environmental conditions plays a significant role in the circuit operating conditions, and it has a significant impact on the stability and the reliability of the output of the PUF or the device where it is embedded in. The factors that cause this variation can be temperature, power supply, ground bounce, crosstalk, radiation hits, or even aging[1] [37].

3.7. PUF properties

To show a PUF's strength and robustness, we use its CRP, which acts as a signature or fingerprint. The function $\sqcap : C \rightarrow R$ such that $\sqcap(c) = r$ expresses the relationship between the challenge and the response, where $c \in C$ and $r \in R$. Figure 6 describes the basic PUF properties that we consider [89, 105, 74, 76].

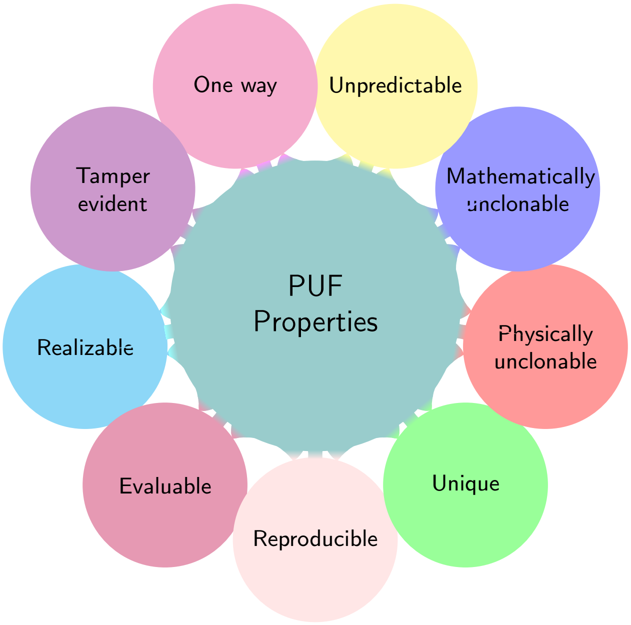


Figure 6: The basic properties of PUF.

- Realizable: A given PUF is realizable if it is easy to invoke its creation procedure and produce a random and unclonable PUF instance given its physical properties.

- Evaluable: It means a PUF can easily produce a response to a random challenge. For a given $\sqcap$ and c, a PUF should be easy to evaluate according to the function $r = \sqcap(c)$ since it does not need any specific requirements.

- Reproducible: For a given challenge, the response may diverge due to the physical environment or the PUF characteristics. Hence, reproducibility means that the PUF must be able to correct this divergence to generate the same response at any time. Thus, a response $r = \sqcap(c)$ can be reproduced with a small error.

- Unique: The function $\sqcap$ contains the identity-related information about the physical entity embedding the PUF, which means the CRPs can be used as a unique identifier of the PUF.

- Physically unclonable: A PUF was considered unclonable when it was not possible to find a corresponded response r of challenge c without the physical PUF. Even if an adversary has the PUF, it is not possible to make a PUF copy. For a given $\sqcap$, it was difficult to fabricate a physical element containing another PUF $\sqcap$' where $\forall c \in C : \sqcap'(c) \simeq \sqcap(c)$.

[1] In some literature, aging is not considered an environmental effect.

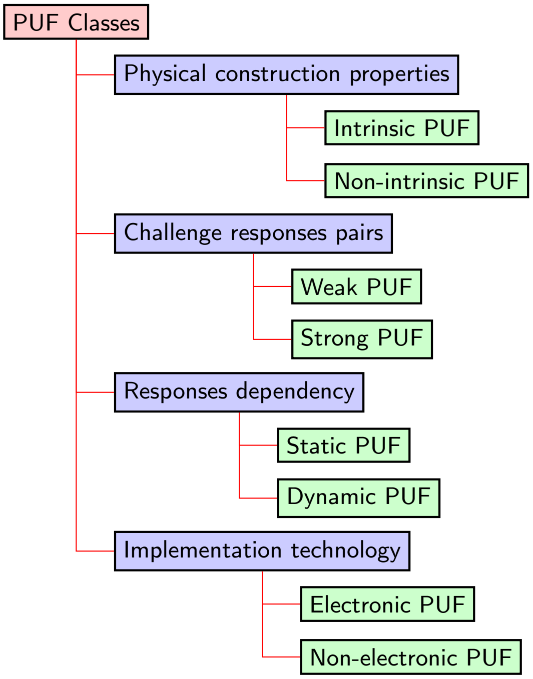


Figure 7: The classification of PUFs [**?**].

- Mathematically unclonable: For a given PUF $\sqcap$, it is hard to construct a mathematical procedure $f_{\sqcap}$ such that $\forall c \in C : f_{\sqcap}(c) \simeq \sqcap(c)$.

- Unpredictable: For a set of CRPs $Q = \{(c_i, r_i) : i > 0 \wedge r = \sqcap(c)\}$, it is hard to predict $r = \sqcap(c)$ up to a small error ($r \approx \sqcap$(c)) for a random challenge c which did not appear in Q.

- One way: For a given r and $\sqcap$, it is not possible to find $c \in C$ such that $\sqcap(c) = r$.

- Tamper evident: Since a PUF is embedded into a physical entity, any alteration of this entity will convert $\sqcap$ into $\sqcap$' and with high probability we got $\exists c \in C : \sqcap(c) \neq \sqcap'(c)$ even with a small error ($\sqcap(c) \not\approx \sqcap'(c)$).

3.8. PUF classes

As shown in Figure 7, we classify PUFs into four classes concerning the implementation technology, the size of challenge-response pairs, the response's dependency, and the physical construction properties.

- Physical construction properties: This class is based on the physical structure properties of the PUF that can be intrinsic or non-intrinsic. In the first case, PUF's construction needs to meet at least two conditions: its uniqueness must be assured during the manufacturing processes, and it must internally evaluate itself from embedded measurement equipment. Otherwise, it is considered non-intrinsic [99].

- Challenge-response pairs (CRPs): The size of challenge-response pairs (CRPs) directly impacts PUF applications among metrics that determine their strength and quality. For the size of CRPs, the results exhibited strong or weak PUFs [41]. The weak have a small number of CRPs due to the lower number of symmetric component blocks used to create the PUF [89]. Thus, an attacker can observe the pairs if he gains physical access to the PUF. Responses from a weak PUF are not public and not unpredictable [131]. Strong PUFs support a massive number of CRPs that grow exponentially with the primary cells or the symmetric component blocks, forming PUFs [89]. This property makes it robust against physical attacks if an attacker has physical access to the PUF. In this case, it is impossible to read all the CRPs since an adversary cannot derive a response to an unknown challenge even with the reverse engineering modelling attacks [25].

- Response dependency: This class is based on the response generation dependency by taking into account the time factor. Practically, the most existing and used PUFs are static, meaning that the generated response is independent of the generation time. In addition to the challenges and the physical features, dynamic PUFs use time as a third dependency, which means dynamic PUFs give different responses to the same challenge at different time slots. Hence, two categories exist in this class: static and dynamic.

- Implementation technology: Various materials and technologies such as glass, plastic, paper, electronic components, and silicon integrated circuits are used to construct PUFs. Thus, each type of material that can be either electronic or not was considered a class of PUFs. The non-electronic PUFs can use electronic subsystems to accomplish their secondary functions [99]. Whereas electronic PUFs use electronic components for their essential operation, such as resistance and capacitance [58].

4. Silicon PUFs Architectures

The PUF is a one-way function that exploits the unique random imperfections found at the nano-scale level of the structure of physical objects [37]. A PUF could be defined as a “digital fingerprint” that is derived from a complex physical object. It is like a black box that takes a challenge as input and produces a response that can be used as a unique identity of the subject or as a cryptographic key.

The term "silicon PUF" has been introduced in [65], which refers to physical unclonable objects built using conventional integrated circuits. Silicon PUF forms a major subclass of electronic PUFs considered integrated circuits (ICs). They can be embedded in silicon chips to accomplish PUF's goals by exploiting their manufacturing process [58]. The Silicon PUF is certainly the simplest PUF as it does not require any modification in the manufacturing process to be used. It exploits the inherent manufacturing variations of transistors and wires that differ from one circuit to another, even if they are part of the same silicon wafer. The Arbiter PUF is the first silicon PUF, introduced by Gassend et al. [31].

According to the different sources of variation, silicon PUFs can be categorized into three major classes: delay-based PUFs, memory-based PUFs, and analog electronic PUFs.

4.1. Delay-based PUFs

The response generated by the delay PUFs depends on the propagation delay between the different delay paths of the PUF's circuits, and it can be affected by the temperature changes of the circuit [2]. Mainly, this type of PUF includes:

4.1.1. Arbiter PUF

Due to the inherent manufacturing variations of transistors and wires, each IC has its own unique delay characteristics, Lee et al. [61] used this property to build secret information unique to each IC, which is called arbiter-based PUF or multiplexer (MUX) PUF.

The idea behind the arbiter PUF is to explicitly introduce a race condition between two digital paths on a silicon chip. It consists of the two delay paths as chains of switch blocks (multiplexers) and an arbiter block at the end of the chain. As shown in Figure 8, the switch block has two possible configurations depending on the challenge bit; straight if the challenge bit is 0 and crossed if it is 1. Each switch block has three outputs: the two outputs from the previous stage and a single bit of the challenge. The inputs of the first switch block are connected to a common enable signal, and the outputs of the last switch block are connected to the arbiter block, which determines which signal arrived first. The arbiter generates a single bit known as the response bit based on this result.

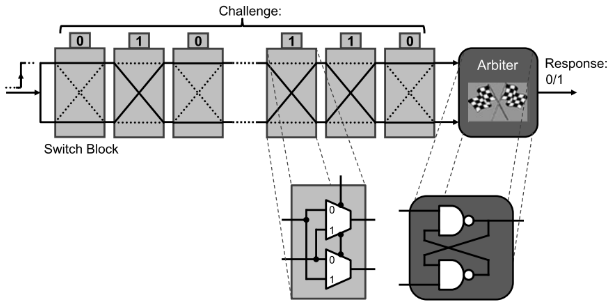


Figure 8: First structure of Arbiter PUF [61].

The same authors of [61] showed that by exploiting the linearity of delay paths, an arbiter PUF was not secure against machine learning attacks. To introduce non-linearity into the PUF scheme, they proposed the feed-forward arbiter PUF (FF APUF) [66], which is an extension of their primary arbiter PUF, where an intermediate arbiter internally generates some challenge bits. Then, these challenges are hidden from an adversary.

Figure 9 depicts the concept of a feed-forward arbiter PUF scheme with one feed-forward arbiter.

In the same direction, several constructions based on the Arbiter PUF have been proposed, such as: XOR PUF or XOR-Arbiter PUF [124], Feed-Forward XOR PUFs [9, 10], Lightweight PUF [81], $m-n$ APUF [70], Multiplexer-based arbiter PUF [113], multi-PUF (MPUF) [69], multi-PUF [115], and Interpose PUF (IPUF) [93].

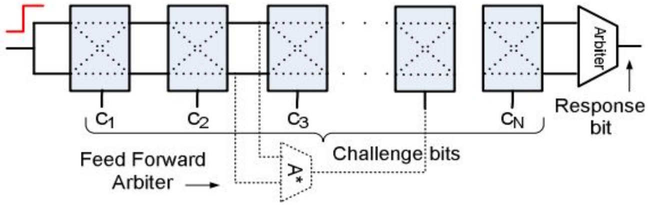


Figure 9: The feed forward arbiter PUF [81].

XOR PUF or XOR-Arbiter PUF [124] combines several rows of the basic arbiter PUF by XORing the outputs of each arbiter PUF into a one-bit response. The length of this implementation is measured by two factors (the length of the challenge's number of switch blocks and the number of rows that indicate the input size of the XOR). Figure 10 shows a 2-XOR PUF with two rows and n switch blocks.

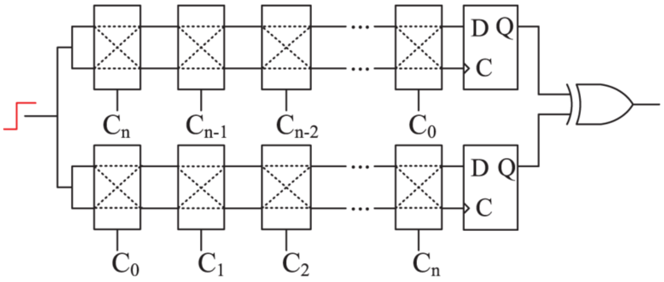


Figure 10: An example of 2-XOR PUF [106].

Recently, Avvaru and Parhi [9, 10] proposed Feed-Forward XOR PUF, which is a combination between Feed-Forward APUF and XOR PUF. Instead of using APUF as a component of XOR PUF, FFXOR PUF uses FF APUF as a new component. According to [9] [9, 10], FFXOR PUF has shown good reliability, uniqueness, and resistance against attacks compared with the classical XOR PUF. However, no document has proposed or analyzed the safety and reliability aspects of this proposed PUF. Figure 11 shows the general architecture of the Feed-Forward XOR PUF.

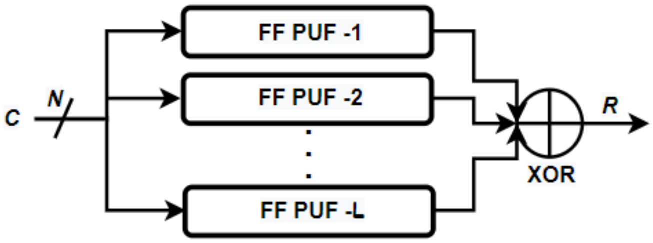


Figure 11: The architecture of Feed-Forward XOR PUF [10].

Lightweight Secure PUFs or Lightweight PUFs have been introduced by Majzoobi et al. [81]. It is a variant of the XOR APUF based on several APUF arranged in parallel. However, the challenge bits are rearranged and modified for each chain. Also, the output response bits of each chain are XORed to obtain a multi-bit response. Figure 12 shows the general architecture of the LSPUF. Since LSPUF outputs are generated using x-XOR PUF, most of the attack strategies developed for XOR PUF can also be applicable to LSPUF, which consequently makes it vulnerable to LR[114] when x≤9.

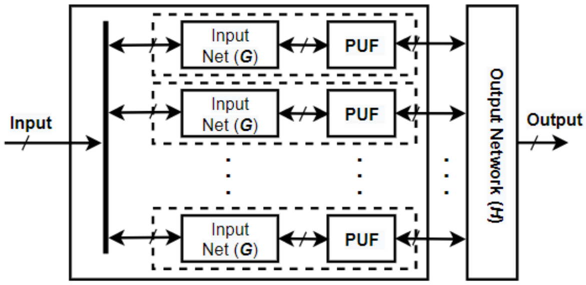


Figure 12: The architecture of LSPUF [81].

From a security perspective, Rührmair et al. [109] showed that all the previously presented PUF implementations can be attacked using ES and LR machine learning attacks, and recently, in [43] authors proved that APUF, XOR APUF, and FF APUF are vulnerable to Deep Learning (DL) modeling attacks.

To enhance the unpredictability of APUF's responses, Machida et al. [70] proposed $m - n$ APUF or double arbiter PUF (DAPUF). Like $n-$XOR PUF, it is based on APUF, where m refers to the number of chains and n to the length of the response.

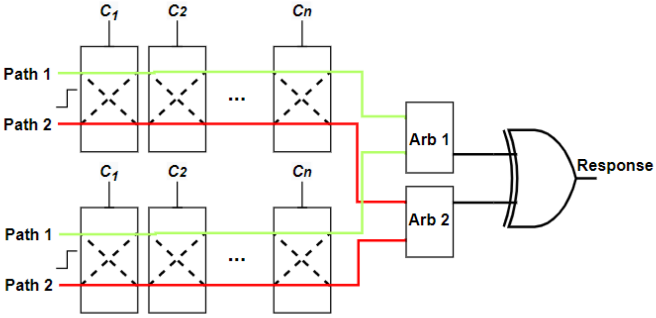


Figure 13: The structure of the $2-1$ Double Arbiter PUF [53].

Instead of comparing the propagation delays of two paths of the same chain like APUF do, DAPUF compares the propagation delays of the same paths across m chains. The response of DAPUF is obtained by XORing all the results of the last comparison process. The experimental results showed that the uniqueness of the proposed 3-1 DAPUFs was approximately 50%, which is much superior to that of 3-1 APUFs. In [71], Machida et al. proposed a 4-1 Double Arbiter PUF and compared 3-1 DAPUF with 3−XOR PUF. This comparison showed that 85% of the responses from the second design could be predicted with machine learning. Contrarily, a 3-1 DAPUF resulted in a prediction rate of 57%, and recently, modeling attacks have been successful against different DAPUFs architectures [53] except for the 4-1 DAPUF. Figure 13 shows the structure of the 2-1 DAPUF.

Sahoo et al. [113] proposed a Multiplexer-based arbiter PUF (MPUF) built with multiplexers and APUFs. An $(n, k)-$MPUF consists of a $2^k - 1$ MUX and $2^k + k$ APUFs where each APUF receives n bit challenge. The outputs of $2^k + k$ APUFs are used as inputs of MUX, where each MUX of the $2^k - 1$ MUXs has three inputs, two data inputs from 2^k APUFs, and one selection input from k APUFs. The $2^k - 1$ MUX selects one of the data inputs as the final response. The robustness of this PUF is that an adversary does not have access to responses of $2^k + k$ APUFs. Figure 14 shows the architectural overview of an $(n, 3)-$MPUF which generates a one-bit response to an n bit challenge by using 7 MUXs and 11 APUFs.

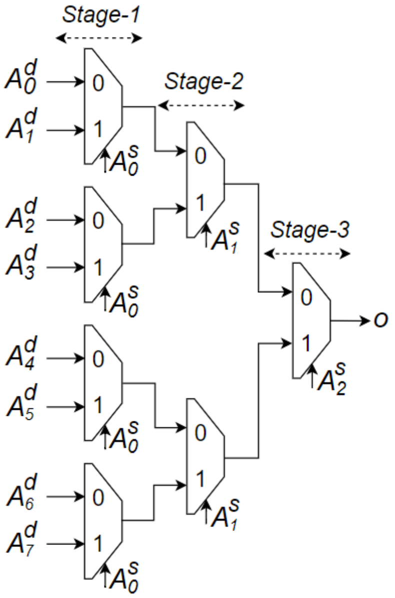
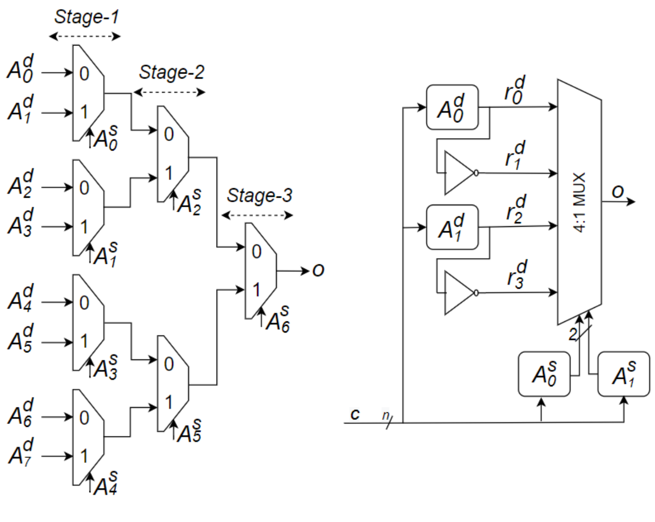


Figure 14: The structure of $(n, 3)-$MPUF [113].

Based on PUF composition principles, two major challenges have been identified to overcome vulnerability against modeling and statistical attacks and lack of reliability. In the same paper, Sahoo et al. [113] proposed two other variants, rMPUF and cMPUF, to ensure reliability and to resist respectively to ML-based attacks and linear cryptanalysis (LC) attacks. Unfortunately, MPUF and its variants can be broken by two recently proposed attacks: logical approximation method and filter-based global approximation attacks [118]. Figure 15 shows an example of (n,3)-rMPUF and (n,2)-cMPUF MPUF variants.

Using the same names but with different implementations, Ma et al. proposed a new arbiter-based multi-PUF (MPUF) [69] as a combination of weak and strong PUF. As shown



(a) (n,3)-rMPUF

(a) (n,2)-cMPUF

Figure 15: Example of MPUF variants: (a) the basic (n,3)-rMPUF and (b) (n,2)-cMPUF. [113].

in Figure 16, MPUF is composed of n PicoPUF [34] and one Arbiter PUF with n switch blocks. To mask the original challenge bit C_i, it is XORed with the response k_i of the i^{th} PicoPUF to generate the new challenge $C*_i$, which is used as the challenge for APUF. As the input of the strong PUF is depending on the output of weak PUF(s), the response of this strong PUF has a strong uniqueness and reliability. MPUF is vulnerable to Deep Learning (DL) modeling attacks [43].

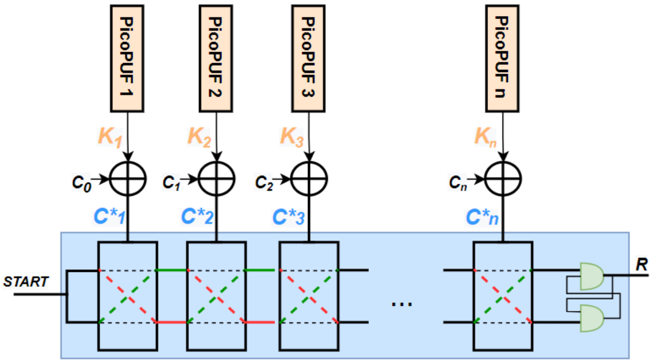


Figure 16: The multi-PUF design based on a PicoPUF and APUF [69].

Huang et al. [44] showed that the uniqueness and the reliability of PUFs could not be guaranteed due to the low hardware resources and the small CRP space. Thus, to enhance the performance of PPUF, they proposed a reconfigurable Pico-PUF (RPPUF) composed of two configurable logic structures, as shown in Figure 17. The RPPUF is a simple NAND-based SR latch with two flip-flop structures and two configurable logic circuits connected before the set-reset latch.

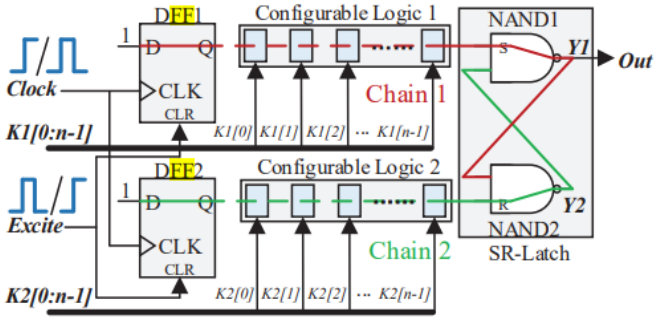


Figure 17: The proposed RPPUF design with configurable logic. [44].

Another multi-PUF implementation was proposed by Sahoo et al. [115] by combining the Ring-Oscillator PUF [124] and Arbiter PUF. The composed PUF is called a Composite PUF, and it is characterized by a larger challenge space and superior quality metrics for each of its components. However, this combined PUF is not secure against cryptanalysis, and modeling attacks [114].

Nguyen et al. proposed one of the most recently designed strong PUFs, called Interpose PUF (IPUF) [93], a combination of two XOR PUF. As shown in Figure 18, *an (x, y)*-IPUF consists of two layers, the upper layer and the lower layer. The upper layer is a x-XOR APUF (x arbiter PUFs) with n challenge bits, whereas the lower one is a y-XOR APUF (y arbiter PUFs) with $n + 1$ challenge bits. The response r_x of the x-XOR APUF is interposed in the nth challenge bit to form $n + 1$ challenge bits.

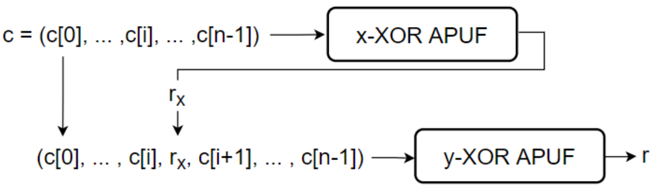

c = (c[0], ... ,c[i], ... ,c[n-1])
x-XOR APUF
r_x
(c[0], ... , c[i], r_x, c[i+1], ... , c[n-1])
y-XOR APUF
r

Figure 18: The structure of the (x, y)-iPUF [93].

The experimental results showed that iPUF is not vulnerable to the reliability-based machine learning attack (CMA-ES) and the classical machine learning attack (Logistic Regression). But, the iPUF of 64-bit challenge length and size of 8 APUF in both layers is broken by the modeling attacks [134].

In order to improve APUF's security against machine learning attacks, Li et al. recently proposed a complex model of APUF, called the Racing APUF (R-APUF) [63]. R-APUF consists of two symmetric paths. However, instead of MUX, the path of R-APUF consists of sub-chains. Each sub-chain has a series of stages based on MUX. the sub-chain is ended by a route selector such as an AND gate or OR gate. R-APUF is characterized by the number of sub-chain in each path and the number of channels in each sub-chain. The structure depicted in Figure 19 can be referred to as a 2-channel 2-stage

R-APUF.

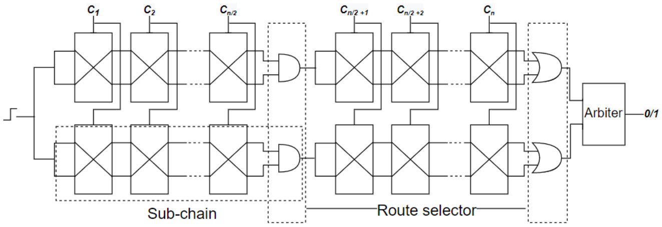


Figure 19: 2-channel 2-stage racing APUF [63].

4.1.2. Ring-Oscillator PUF

Rather than the basic arbiter PUF and its derivatives, ring oscillator PUF (RO PUF) is another PUF design based on the delay difference of identical electrical paths initially proposed by Suh and Devadas [124].

As represented in Figure 20, a typical RO PUF consists of N identically laid-out delay loops, or ring oscillators (ROs), two multiplexers, two counters, and an arbiter. Theoretically, each RO oscillates at the same frequency, but due to manufacturing variations and environmental conditions, it oscillates at a slightly different frequency. To generate a one-bit response from these N ROs, a pair of ROs needs to be selected. This selection is determined by the input (challenges) applied to both MUX and a comparison of the frequency of the selected RO pair. The response bit is set to 1 or 0 depending on the comparison, 0 if the first oscillates faster than the second, and 1 if it is not. From N ring oscillators, RO-PUF can produce $log_2(N!)$ bits [124]. For example, 32 oscillators can produce 118 bits. Compared to APUFs, RO PUFs allow easier implementation for FPGAs and ASICs, easier evaluation of entropy, and higher reliability. Nevertheless, RO PUFs took longer, used more power, and needed more space to make the responses.

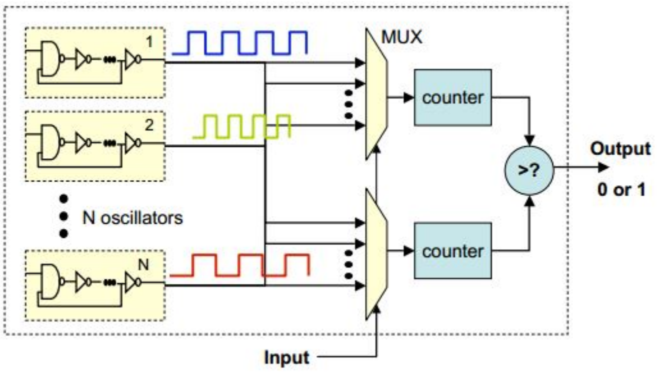


Figure 20: Ring oscillator based PUF circuit [124].

Due to the low number of CRPs generated by RO PUF, it was classified as a weak PUF and is vulnerable to cryptographic analysis attacks. In [109] they showed that machine learning algorithms could model RO PUFs, and in [85] they used electromagnetic attacks to break the security of RO PUFs. Therefore, several variants of RO PUFs have been proposed.

To reduce the noise in RO PUF responses and increase the number of CRPs of the basic RO PUF, the first configurable ring oscillator PUF (CRO PUF) has been introduced in [80]. As shown in Figure 21, a multiplexer has been added after each stage of the RO to check if the inverter will be selected as a member of the RO. According to the input selection bit, each MUX selects one output of the two inverters. So for RO with three stages, eight configurations are possible.

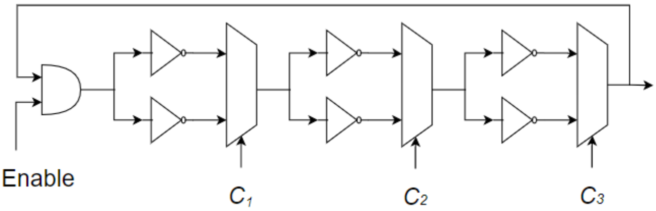


Figure 21: Configurable RO [80].

Based on the same idea, Gao et al. [29] proposed another configuration of RO PUF, called configurable RO PUF or flexible RO PUF [29]. Figure 22 shows that the selection of an inverter from the ring is chosen dependingn the input selection bit. If the bit is 0, the corresponding inverter is discarded, else it will be used in the ring. So, for a RO with three inverters, eight configurations are possible. CRO PUF is vulnerable to modeling attacks while it is characterized by a low number of CRPs as well as to machine learning attacks [87]. Figure 22 depicts the architecture of one ring of CRO PUF.

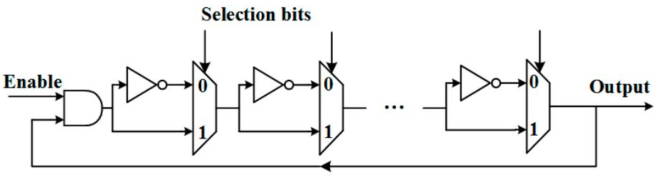


Figure 22: Architecture of the configurable RO [29].

Yu and Devadas [139] proposed the k-sum PUF that consists of k pairs of ring oscillators. To generate the one-bit response, k-sum PUF measures the difference between two delay terms, each produced by the sum of k ring oscillator values. To build these two terms for each k stage (Figure 23), the challenge bit C_i defines which RO is used to compute the bottom and top delay terms. However, K-sum PUF is vulnerable to machine learning attacks [121].

In [82], Marchand et al. proposed the Transient Effect Ring Oscillator (TERO) PUFs as an alternative to RO PUFs with a similar structure, but it is constructed from TERO cells that have two states: stable and transient oscillating. As shown in Figure 24, the basic structure of a TERO PUF is an

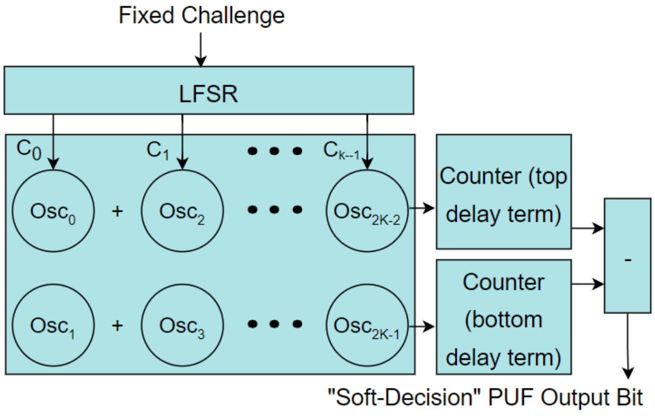


Figure 23: K-sum PUF [41].

RS flip flop, where the TERO cell is composed of two identical and symmetrical branches (Branch 1, Branch 2). Each branch is designed with an initialization stage and inverters whose exact number is used for both branches. The circuit starts oscillation for a short time by setting the init signal to one and depending on the mismatch in the delays between the two branches of the TERO cell caused by CMOS process variations. This behavior results in a finite number of oscillations of the TERO cell output that is considered as the TERO PUF response. Also, they showed that TERO PUF is not as susceptible to frequency injection and cloning attacks through electromagnetic analysis. But in [129], Tebelmann et al. showed that using non-invasive electromagnetic measurements and tailored attack methodology could recover up to 25% of the TERO PUF response's bits without errors.

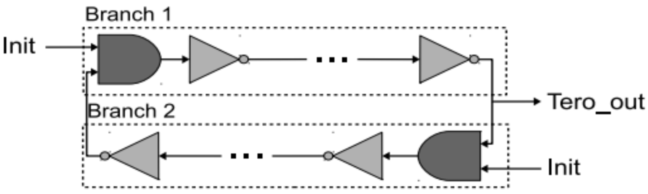


Figure 24: Generic Structure of a TERO cell [82].

Recently, Della Sala et al. [20] proposed a new FPGA-compatible design named the Delay Difference PUF (DD-PUF), which requires a minimal area footprint and provides excellent reproducibility under temperature and supply voltage variations. Figure 25 describes a single DD-PUF cell composed of two inverters (I_1 and I_2), interposed between two D-Latches (L_1 and L_2) forming two identified paths that can be identified ($P_1 = L_1$-I_1 and $P_2 = L_2$-I_2). The DD-PUF needs two control signals, START and RESET, connected to the enabling gate and used to clear the pins of the two latches. When the asynchronous RESET is set to 1, both latches' output pins are forced to 0. When the START signal is set to 1 for a period of time interval, an oscillatory state is produced within the DD-PUF cell. At this point, only the small delay difference between P_1 and P_2 determines the resulting stable bit (response).

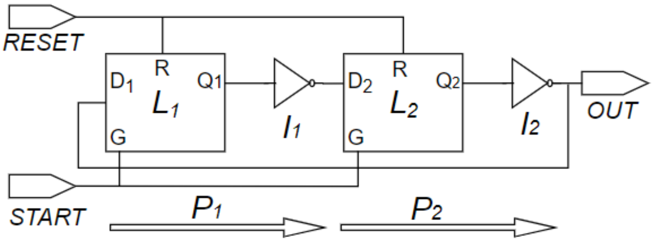


Figure 25: The architecture of a single DD-PUF cell [20].

4.1.3. Glitch PUF

It is the first FPGA-specific PUF [127] design proposed to reduce the ease of predicting the relationship between challenges and responses in delay PUFs. GPUF exploits glitch waveforms caused by variations in the delay between gates to generate the responses. Its architecture consists of three parts: 1) a combinational circuit for generating glitch waveforms, 2) a sampling circuit for Glitch, and 3) a response generator. First, the input value of the glitch generator is presented to a data register as a challenge. Then, the acquisition of the glitch waveforms. Finally, the conversion of the waveforms into response bits. Compared to other PUF designs, GPUF has good performance, and it is ranked among the most secure PUFs against modeling attacks. Figure 26 represents the whole structure of Glitch PUF.

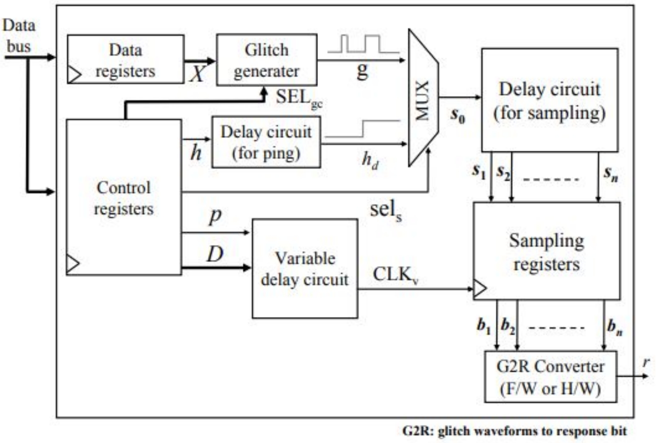


Figure 26: Whole structure of Glitch PUF [127].

As shown in Figure 26, the circuit area of the discussed glitch PUF is large. Hence, Shimizu et al. [119] have proposed a simplified glitch PUF called the second glitch PUF. As shown in Figure 27 the second PUF glitch is simplified in terms of eliminating certain circuit blocks. More precisely, the sampling circuit. In addition, the output of the glitch generator is connected directly to the toggle flip-flop converter

(TFF). From the security side , no successful machine learning attack model against the two glitch PUF designs has been proposed.

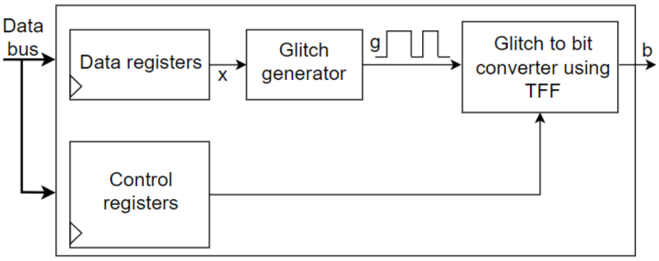


Figure 27: Second glitch PUF [119].

4.1.4. *Intellectual property PUF (IP-PUF)*

To ensure the intellectual property (IP) of personal use, Nithyanand et al. [95] proposed the use of a set of silicon circuits embedded on a personal computer (PC) as a PUF named Intellectual Property PUF (IP-PUF). Mainly, the authors used the intrinsic features found in silicon circuits to exploit mismatches in frequencies of oscillators of the CPU clock or the timer interrupt clock. Then, by exploiting the value of the time period needed to load instructions from the processor cache into the register memory that varies from one PC to another one.

4.1.5. *Clock PUF*

The clock network routes a timing signal from the clock to various sections of the circuit design. It ensures synchronicity by respecting the time taken by the signal from the clock to reach any given area of the circuit. Otherwise, the issue of clock latency variation is known as clock skew. Based on these variations and skewing, Yao et al.[138] proposed the clock PUF (CLK-PUF) similar to an arbiter PUF since it uses MUXes to select two paths of the clock network and compares their delays using an arbiter to generate a response bit (Figure 28). CLK-PUF has been broken by machine learning based attacks [93] and is vulnerable to non-invasive attacks [140].

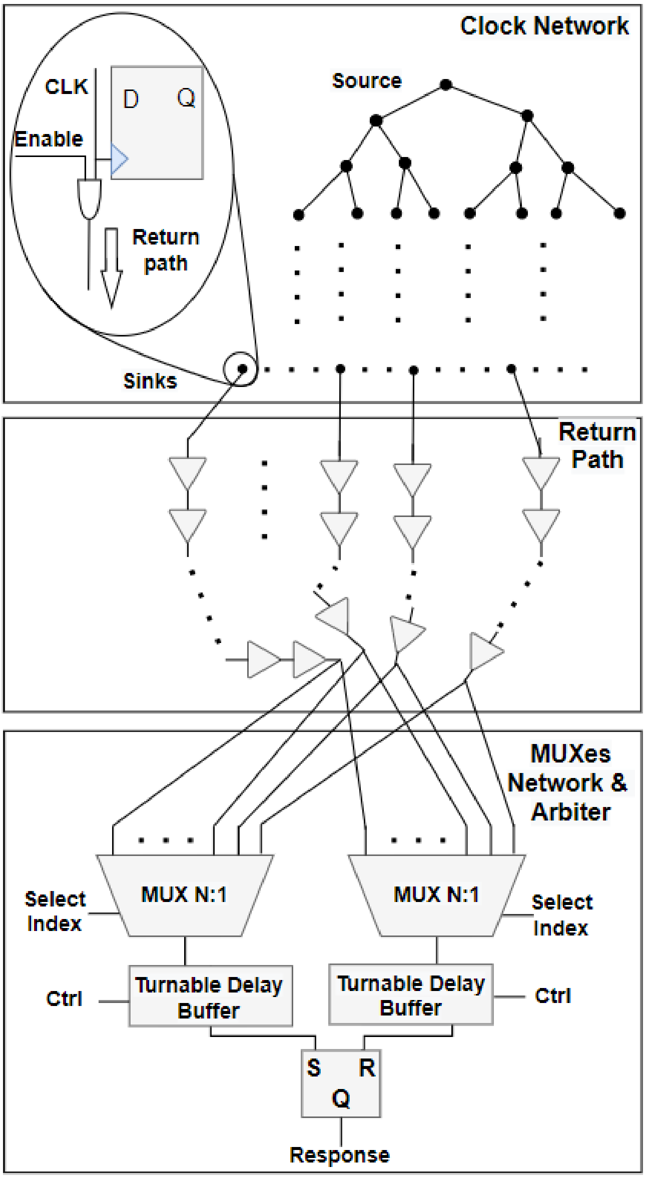


Figure 28: The architecture of Clock PUF [138].

4.2. Memory-based PUFs

The response generated by the memory-based PUFs depends on the initial state of the memory structures. At a power-up, the structures are set in an unstable state, and the response corresponds to the stable state of the structures caused by an external data signal input [2]. This type of PUF family mainly includes:

4.2.1. *SRAM PUF*

Guajardo et al. [35] proposed static random-access memory, or SRAM PUF, as the first intrinsic PUF construction based on the power-up state of an FPGA's SRAM memory. It does not need any modifications in the manufacturing process. It is based on the static-noise margin (SNM) that requires a memory cell to change its logical value. A SRAM cell is logically constructed as two cross-coupled inverters, hence leading to two stable states [76]. During the start-up, the initial value 0/1 of a SRAM cell is given randomly and independently by the SNM. This randomness is due to the manufacturing process of the SRAM cell. In order to generate the response, SRAM PUF uses a range of memory locations of an SRAM memory block as a challenge, and the responses are the start-up values of the whole SRAM cells that compose the challenge. In its first implementation, SRAM PUF was used in protocols for the IP protection problems implemented on FPGAs. Figure 29 shows the design of the SRAM PUF cell with six transistors. In [38], authors showed that it is possible to clone SRAM PUF.

The start-up values of the SRAM cells are controlled by the IC manufacturer, which renders SRAM PUF useless for

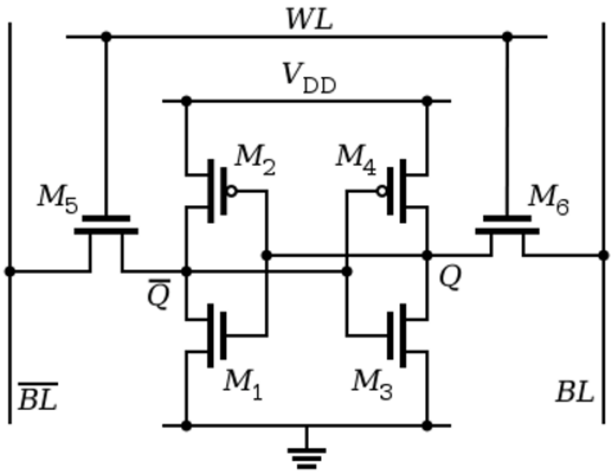


Figure 29: SRAM cell with 6 transistors.

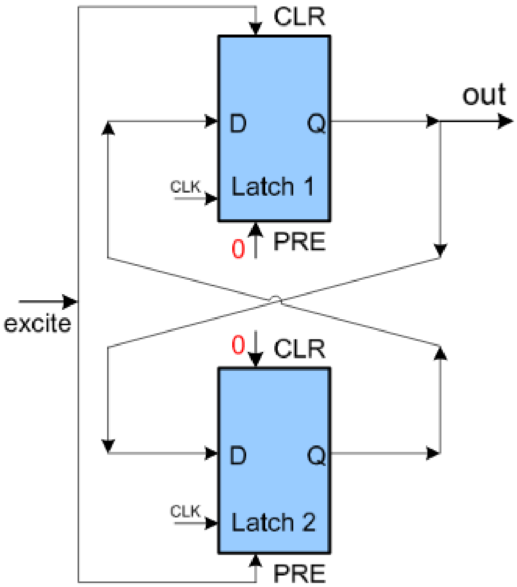


Figure 30: Butterfly PUF cell [59].

FPGAs [73]. To overcome this issue, many improved implementations of the SRAM PUF have been proposed, such as the Butterfly PUF [59], Flip-flop PUF [73], Latch PUF [122] and Buskeeper PUF [120].

SRAM PUFs [35] are used only on FPGAs that support initialized SRAM memory. In order to resolve this problem, Kumar et al. proposed replacing the inverter with latches or flip-flops to build a cross-coupled circuit, and they called it Butterfly PUF [59]. As shown in Figure 30 the structure of the BPUF cell consists of two latches, where each latch is a cross-coupled circuit, which represents a fundamental building block used in all types of storage elements in electronic circuits. This cross-coupled circuit has two different stable operating points, 0/1 and an unstable operating point. An unstable state can be introduced, after which the circuit converges back to one of the two stable states. BPUF exploits this random assignment of a stable state from an unstable one to generate the secret key. This assignment can be comparable to the stat of the SRAM cell after power-up. After experimentation, they found that the proposed PUF can be used in IP protection and in cryptographic applications by generating a secret volatile key.

Maes et al. used the power-up values of the flip-flops present on the FPGA as a PUF, named Flip-flop or D flip-flop PUFs [73], in the same way as an SRAM PUF. This is due to manufacturing variations. When the IC is powered up, the output state of each flip-flop has a random value; hence, it can be zero or one. The experimental results found that the amount of randomness present in the power-up values of the flip-flops is limited, so power-up bits cannot be used directly. So, to increase the quality of responses, post-processing is required [73]. The main advantage of this design is that it is easily spread over an IC and it is challenging to locate it, so it is robust against a reverse-engineering attack.

As we have seen, SRAM and Flip-flop PUF require being powered-up to generate the response bits. This means the cells of these two PUFs should be repowered whenever the responses are needed. Furthermore, Flip-Flop PUF requires some extra processing to extract uniform randomness.

Unlike SRAM and flip-flop PUFs, Su et al. introduced the Latch, or SR-Latch PUF [122], which generates the response when its input is simultaneously enabled. The SR-Latch PUF consists of two cross-coupled NOR gates. Using the metastable value of these gates, LPUF can generate responses without an actual device power-up. As shown in Figure 31, when the input is triggered with the rising edge, the SR-Latch starts oscillating and enters into a metastable state. After a period of time, the SR-Latch stops oscillating and becomes stable. Due to the manufacturing variation, the state that the SR-Latch falls into is unknown, and it can be used as a response bit [8]. LPUF can be implemented on both ASIC and FPGA. But it is not appropriate for low-cost implementation of a PUF. Hence another approach is proposed to address this issue [8].

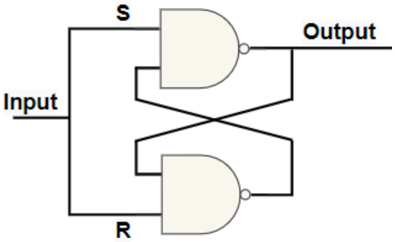


Figure 31: Basic structure of SR-Latch cell [8].

In order to improve the D Flip-flop PUF, Simons et al. were the first to exploit the existing buskeeper cell as a viable alternative to the D-Flip-flop one. The big advantage over using a DFF cell for constructing a PUF is that the Buskeeper cell is minimal, and it does not require any additional circuits or processes to generate a reliable response bit [120]. As shown in Figure 32, the Buskeeper or busholder PUF [120], consists of two inverters. The principle of BPUF is similar to all memory-based PUFs, where the initial patterns are read at the memory power-up. The authors' experiments prove that BPUFs have better reliability and uniqueness compared to DFF-PUFs [120].

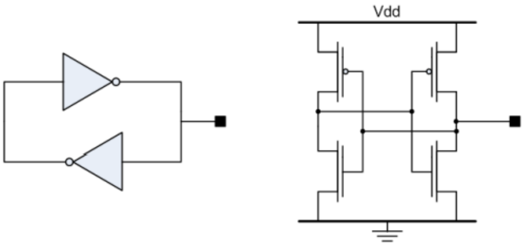


Figure 32: Buskeeper cell structure [120].

4.2.2. *Bistable Ring PUF*

SRAM, Butterfly, Flip-flop, and Buskeeper PUF possess an even smaller number of CRPs, which is proportional to their size. Hence, they can be used as so-called Weak PUFs. The Bistable Ring PUF [19] was the first strong memory-based PUF proposed by Chen et al. As shown in Figure 34 BR-PUF consists of an even number of inverters connected to each other to build a ring. When the device is powered up, each inverter in the ring tries to force its output from an initial value of 0 to 1. For a BR-PUF of 6 inverters, the ring has two possible, stable states, 101010 or 010101. Hence, the output of the last inverter is the one-bit response generated according to which state the ring falls into, and this initial state corresponds to the response. In order to generate an exponential number of CRPs, they proposed an architecture where the inverter count was duplicated to be used as a strong PUF. BR PUFs can be vulnerable to modeling attacks [116, 17].

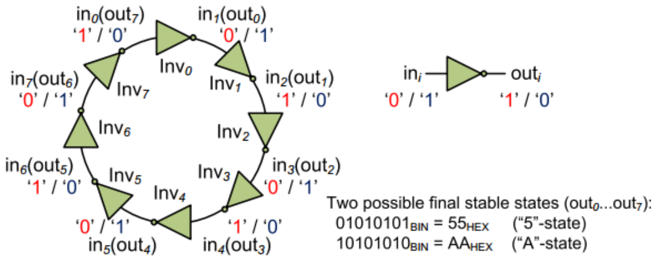


Figure 33: Two possible stable states of an eight-stage bistable ring [19].

4.3. Analog electronic PUFs

The response generated by the analog electronic PUFs depends on the analog movements of the electronic components such as resistance and capacitance [58]. This type of PUF mainly includes:

4.3.1. *ICID PUF*

Integrated Circuit IDentification (ICID) was proposed by Lofstrom et al. [68]. It consists of some transistors with identical designs arranged in an addressable array. Each addressed transistor drives a resistive load due to the voltage thresholds, a random placement function of the doping atoms in the impurities of the silicon channels. The voltage on the load is measured and converted into a bit response where the challenge is the number of the transistor component. Figure 34 shows a block diagram of the ICID PUF.

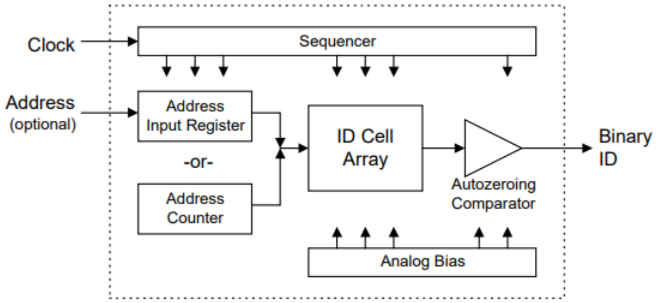


Figure 34: Block diagram of ICID PUF [68].

4.3.2. *Coating PUF*

Based on the idea "thou shalt not store secret keys in digital memory", Tuyls et al. introduced the first Coating PUF [130] using the randomness contained in the protective coating of an IC that is introduced during the manufacturing process. They drive the key, which could be used as the device's fingerprints. This is depicted in Figure 35. The proposed experimental security evaluation says that the proposed PUF is safe from physical attacks.

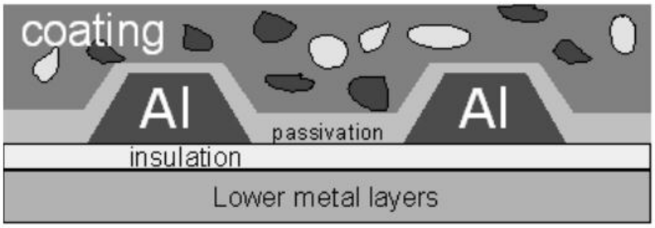


Figure 35: Schematic cross-section of a Coating PUF IC [130].

4.3.3. *Power grid PUF*

Since the voltage drops and the equivalent resistances are affected by random variations in the manufacturing process, Helinski et al. have introduced a new PUF, called the Power Grid or Power Distribution PUF [39], which is based on the resistance variations in the electrical network of an IC. PG-PUF is susceptible to machine learning attacks [107]. Figure 36 shows a circuit for the generation of the response using a power grid.

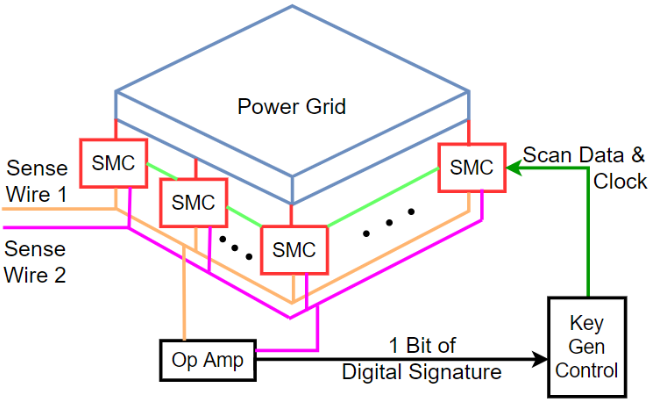


Figure 36: Circuit for generation of the signature using power grid [123].

4.4. Comparison

Table 2 classifies the surveyed PUF schemes into their classes in terms of their strength (weak'W' or strong 'S'), performance (uniqueness and reliability), and resistance to different attacks. Based on this classification, we observe that.

- Arbiter PUF [61] is one of the most used PUF, and its improved architectures [66, 124, 9, 10, 81, 70, 113, 69, 115, 93, 63] achieve good performance in the two well-defined quantitative metrics: uniqueness (≈ 50) and reliability (≈ 100) (see Table 4 for more details). However, they do not perform well in other equally important metrics, especially security which is the most important metric that determines its acceptability in real-life systems. Further, since a relationship between the challenge and the signal propagation time of the arbiter PUF can be represented as the linear model, exploiting this weakness, APUFs are vulnerable to many types of modeling attacks such as machine learning and deep learning attacks. However, strong silicon PUF is suitable for authentication by using many CRPs. Even though APUF is simple and easy to implement, its production process is precise, while the lines must be of the same length.

- Ring-Oscillator PUF [124, 80, 29, 139] is another widely used daily based PUF due to the simplicity of its design and ease of CRP extraction. However, the path between the oscillators and the counters should be exactly the same. As it is classified as a weak PUF, it is suitable for secret key generation, but is also vulnerable to modeling attacks. Compared with other daily-based PUF, RO PUFs are more considerable and consume more power, but provide higher reliability.

- Glitch PUF [127, 119] is predominant compared to other delay-based PUFs in terms of resistance against modeling attacks. It is suitable for secret key generation, but its design and glitch acquisition process are crucial.

- SRAM PUFs [35, 59, 73, 122, 120] are one of the most popular weak PUFs. Due to their simplicity and intrinsic categories, they do not require any extra hardware. However, as they have a restricted number of CRP, they are suitable for secret key generation and are widely used for identification. Compared with other PUFs, SRAM PUFs are sensitive to environmental conditions such as temperature and voltage. Therefore, error correction techniques are vital to moderate these impacts and provide reliable keys. SRAM PUFs are secure against modeling attacks, but are more susceptible to cloning attacks and invasive attacks in general.

- Bistable Ring PUF [19] is a strong memory-based PUF suitable for authentication. The BR PUF has a good uniqueness and reliability, and generally, it is reliable against aging, but it is also vulnerable to modeling attacks.

- ICID PUF or VT PUF [68] has limited IDs and fewer CRPs. Thus, it is suitable for secret key generation and identification. It is cheap and small in size, but vulnerable to cloning attacks, and It needs a particular design.

- Coating PUF [130] is suitable for secret key generation, identification, and for detecting physical tampering. It is small, fast, and cheap, but it needs a special design.

- Power Grid PUF [39] is suitable for secret key generation. It needs a special design, and it is vulnerable to cloning attacks.

We can resume that:

1. Delay-based PUFs are a class based on frequency variations or digital race conditions to generate PUF responses within integrated circuits (ICs) resulting from manufacturing variations. Several delay-based PUFs are made of arbiter PUF as a basic element. All delay-based PUFs are extrinsic PUFs, meaning they need specific extra hardware to be used in a silicon chip. The latter needs a precise process to generate a unique and reliable response. The number of the responses of several delay-based PUF is not limited, making them suitable for authentication. Whereas delay-based PUFs are not proficient in material resources and are subject to modeling attacks, this allows an attacker to build a mathematical clone of a PUF to estimate the PUF's responses.
2. Memory-based PUFs are based on the metastable state of memory cells and unpredictable start-up values. The generation of the response is limited by the number of memory cells. So, most of the memory-based PUFs are weak and have fewer CRP. However, they are suitable for identification and secret key generation. Memory-based PUFs are intrinsic PUFs (except BPUF) because their circuits are implanted within the design itself and do not require any additional hardware.

Table 2
Comparison of Silicon PUFs Architectures.

Class	Scheme	W/S	Year	Uniqueness(%)	Reliability(%)	Attacks
Delay-based	APUF [61]	s	2004	23	99.76	[109, 43]
	FF APUF [66]	s	2004	38	90.16	[109, 43]
	XOR APUF [124]	s	2007	46.15	99.52	[109, 43]
	FFXOR PUF [10]	s	2020	≈ 50	89	-
	R-APUF [63]	s	2019	-	94.74	-
	LSPUF [81]	s	2008	46.16	92.32	[114]
	$m-n$ APUF [70]	s	2014	≈ 50	-	[53]
	MPUF [113]	s	2017	50	99.70	[118]
	rMPUF [113]	s	2017	50	99.55	[118]
	cMPUF [113]	s	2017	49.99	99.68	[118]
	MPUF [69]	s	2018	40.60	-	[43]
	CPUF [115]	s	2014	49.04	97.48	[114]
	IPUF [93]	s	2019	≈ 50	≈ 100	[134]
	RO-PUF [124]	w	2007	46.15	99.52	[109]
	CRO-PUF [80]	w	2011	47.31	99.14	-
	TERO PUFs [82]	w	2017	49.65	96.32	-
	DD-PUF [20]	w	2021	49.48	98.33	-
	RPPUF [44]	S	2021	45.80	99.23	-
	FR-PUF [29]	w	2014	46.88	-	[87]
	k-sum PUF [139]	s	2010	-	-	[121]
	G-PUF [127]	s	2010	41.50	> 93.40	-
	SG-PUF [119]	s	2012	35	> 93	-
	IP-PUF [95]	s	2011	-	-	-
	CLK-PUF [138]	s	2013	-	-	[93, 140]
Memory-based	SRAM PUF [35]	w	2007	49.97	> 88	[38]
	B-PUF [59]	w	2008	≈ 50	> 96	-
	FF-PUF [73]	w	2008	≈ 50	95	-
	L-PUF [122]	w	2008	50.55	96.96	-
	Buskeeper PUF [120]	s	2012	48.27	80.98	-
	BR-PUF[19]	s	2011	≈ 50	97.81	[116, 17]
Analog electronic	ICID-PUF [68]	w	2000	49	> 95	-
	C-PUF [130]	w	2006	≈ 50	> 88	-
	PG-PUF [39]	w	2009	-	-	[107]

3. Analog electronic PUFs are a class of PUFs that exploit the analog measurement of an electric component to generate a response. Analog electronic PUFs are more suitable for integrated circuit identification and physical tampering. Generally, they are represented by power grid PUFs and coating PUFs. Analog electronic PUFs are vulnerable to cloning attacks.

5. Silicon PUFs Applications

PUFs have been used in a wide range of applications to secure devices depending on the PUF class (weak or strong) of the embedded chip within the device. This section surveys the existing application areas and use cases of Silicon PUFs that are illustrated in Figure 37. Two applications were widely found: Secret key generation and authentication.

5.1. Authentication Protocols

One of the main objectives of any security system is to achieve robust authentication, which refers to verifying the device's identities and preventing malicious ones from accessing a trusted area or a network. However, numerous works have been proposed in the literature demonstrating various PUF-based authentication protocol schemes.Before surveying these works, we present a basic scheme for achieving authentication between a server and a device equipped with a PUF chip.

Figure 38 depicts a conceptual PUF-based authentication process between a device equipped with a PUF and a trusted server. PUF-based authentication protocols can be accomplished in two distinct phases. Firstly, during the enrolment phase, the server has access to the IoT device to apply a set of random challenges and then stores their corresponding sets of responses that are extracted from the PUF circuit integrated with the IoT device. The second phase is verification, in which the device verifies the identity of the IoT device. Next, the server randomly selects from its CRP database a challenge that has never been used. Then, the IoT device generates its corresponding response and sends it back to the server. If the response from the server side matches the one that was stored for the challenge that was used, then the IoT device is real and can connect to the IoT network.

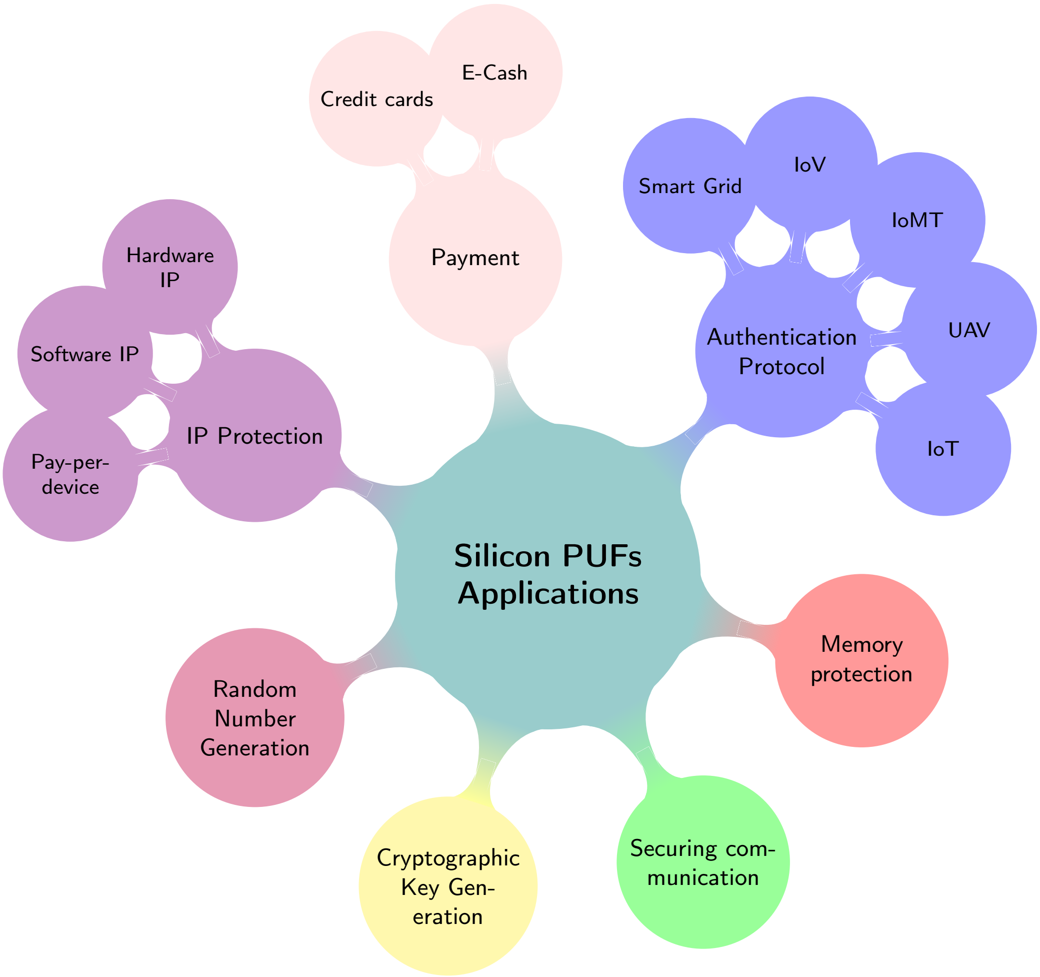


Figure 37: Silicon PUFs Applications Areas and Use Cases.

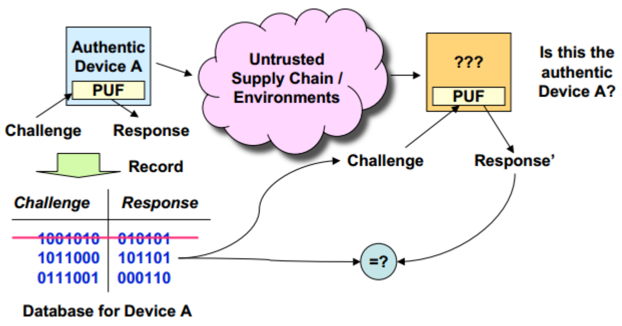


Figure 38: A PUF-based Authentications Protocol Overview.

Over the last decade, a considerable amount of research has been conducted in the PUF-based authentication field. These protocols use a variety of silicon PUF types and different authentication mechanisms and aim to provide a lightweight and secure authentication scheme under various settings.

5.1.1. *Internet of Things (IoT)*

Idriss et al. [45] proposed a lightweight PUF-based protocol that offers mutual authentication for IoT devices. Instead of storing the generated CRPs on the server, this scheme stores a so-called CRP soft model that can be obtained by performing a machine learning attack on the generated CRPs. This protocol does not ensure the reliability of communication, especially the error correction. Najafi et al. [92] presented a PUF-based authentication protocol that does not offer mutual authentication, and many attacks were not considered in their scheme. However, they used a Convolutional Neural Networks (CNN) as a solution to eliminate the need for error correction mechanisms. Using elliptic curve cryptography (ECC) as a second security primitive, Aman et al. proposed a PUF-based authentication protocol [5] that does not consider noise elimination. Muhal et al. [91] proposed a PUF-based authentication protocol that is vulnerable to physical attacks since the device stores an initial session secret key that will be used in the authentication phase. Rather than that, the proposed scheme does not use any noise elimination technique, making it impractical in a real application. Mostafa et al. [90] proposed a mutual two-factor authentication mechanism between a device and a server, where the device is equipped with a strong and weak PUF. The first one is used for authentication and the second for encryption.

This scheme does not present noise elimination, making it impractical in real applications and different environments. Aman et al. [6] presented a light-weight mutual PUF-based authentication protocol for IoT systems, including device-to-device or device-to-server communication. However, the proposed protocol does not consider error correction in the authentication steps.

5.1.2. *Unmanned Aerial Vehicles (UAVs)*

Nowadays, Unmanned Aerial Vehicles (UAVs) are becoming very popular due to the emergence of their areas of application: delivery, first-aid emergency, military, etc. Nevertheless, the communication between a UAV and its ground station (GS) is critical (sensitive data, weather, environmental changes, etc.) . In [102], Pu and Li proposed a mutual authentication protocol between a drone equipped with a PUF and its ground station without the support of error correction. Also, the authors do not show the details regarding the security analysis of the proposed protocol. Alladi et al. [4] proposed UAV-GS and UAV-UAV PUF-based authentication mechanisms. The ground station plays an important role in the authentication phase, and it is also responsible for session key generation and delivery. The noise elimination process has not been considered, making both schemes impractical. Also, Bansal and Sikdar [12] presented mutual authentication in UAV swarm networks using PUFs. The proposed protocol uses a spanning tree protocol to identify the flow of authentication request messages in dynamic typologies and mobile UAVs.

5.1.3. *Internet of Medical Things (IoMT)*

For the safety of patients, PUFs have been used to secure the communication between devices, sensors and the health care monitoring system. Yanambaka et al. [136] presented a PUF-based authentication scheme between the IoMT devices and the server, where the server is also equipped with its proper PUF. In addition, a secure database was used as a third party to store collected CRPs. However, the exchanged messages between the device and the server have not been subject to any encryption or camouflage techniques that facilitate easily launching modeling attacks. Wang et al. [133] proposed a lightweight and reliable authentication protocol for wireless medical sensor networks, that is composed of cutting-edge blockchain technology and a PUF. Also, Lee and Chen [62] used a one-way cryptographic hash function and BS-PUF to ensure lightweight authentication between IoMT sensors and fog devices. Gope et al. [33] introduced a new lightweight anonymous authentication protocol for IoMT that is resilient against machine learning attacks on PUFs. To prevent various security weaknesses such as user anonymity, offline passwords, smart device theft, privileged insiders, and cloning attacks in WMSN, Kwon et al. [60] proposed a three-factor-based mutual authentication scheme using PUFs.

5.1.4. *Internet of Vehicles (IoV)*

To guarantee the security and privacy of driving data in IoV, Jiang et al. [46] introduced a secure authentication and key exchange protocol for IoV using two-factor security that combines PUFs with the user's password. The second factor is used if an advisory could hold the vehicle equipped with a PUF. Then, they added biometrics as a third factor to the same protocol [47]. In [86] vehicles and roadside units (RSU) use PUFs to authenticate themselves to the certificate authority. In this scheme, the authentication process depended on the reception of the silicon PUFs' unique fingerprint and the valid delivery certificate.

5.1.5. *Smart Grid*

A smart grid (SG) can provide reliable, secure, economic, efficient, clean and high-quality electricity services. Smart meters are devices collecting data on smart grids, that can also receive instructions from the control center. However, the communication between smart meters and the control center confronts security and privacy challenges. Instead of storing a set of CRPs on the server, Kaveh et al. [51] proposed a PUF-based authentication protocol where only one pair of CRPs is stored on the server. The used pair is updated at the end of each successful authentication phase. This protocol is vulnerable to physical attacks since it stores secret information on the device's memory. To protect smart meters from physical attacks, Cao et al. [15] addressed the security and privacy problems in collecting metering data by proposing a lightweight privacy-preserving authenticated data collection scheme based on PUFs. In the case of fault or improper behaviour due to the high-tension power lines of the smart city, sensor nodes deployed on these lines send information to the control center to request in an emergency the recovery team. In [11], Badar et al. introduced an identity PUF-based lightweight authentication protocol for supply-line surveillance system between the sensor nodes and the control center.

5.2. Cryptographic Key Generation

In any cryptographic primitive, it is recommended that the key must stay constant and can be reproduced several times. As silicon PUFs are a source of high randomness, their generated responses could be used as cryptographic keys in different security applications. However, the change in environmental conditions will cause noise in the output of the PUFs. This noise can cause one or more PUF output bits toggle, resulting in an incorrect and unusable key because it is not the same as the original key. Therefore, the response cannot be directly used as a cryptographic key. Hence, error correction must be used in order to tackle this issue [103]. Fuzzy Extractor (FE) and many coding techniques for error correction are being employed in order to improve the reliability of PUFs' applicability [117].

Fuzzy Extractor (FE) [23] is designed for extracting nearly uniform random strings from noisy and non-uniform random data with high entropy. FE is built from a pair of algorithms to extract stable, reproducible information from the PUF responses; generation (*Gen*) and reproduction (*Rep*). Gen takes the initial response and outputs uniform random string data (refer to the cryptographic key) and non-secret data called public helper data. To reproduce the key from a noisy response, the reproduction algorithm, Rep, takes two inputs: the noisy response and the public helper data. The reproduction succeeds only if the initial and noisy responses are close enough. As shown in Figure 39, given the same challenge c as input to the same PUF module PUF_i, in different temperatures $m_1 = 30K$ and $m_2 = 80k$, the PUF generates two different responses $R_i(c)$ and $R'_i(c)$. We consider the first response as the reference and the second one as noisy. We use the Gen procedure to generate the secret key k and the public helper data P. Then, for the reproduction of the same key, we use the Rep procedure, which takes the noisy response and P as input [141].

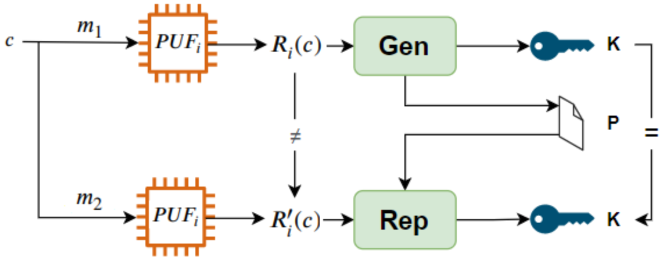


Figure 39: Fuzzy Extractor.

5.3. Intellectual Property Protection

Electronic products suffer from many security challenges such as counterfeiting, cloning, reverse engineering, and the vicious addiction of components, making semiconductor companies suffer tremendous financial losses. Consequently, it is crucial to protect the intellectual property (IP) components of an IC design. Guajardo et al. [35] introduced a protocol for the hardware IP protection problem on FPGAs based on SRAM PUF. Also, Zheng and Potkonjak [145] presented a PUF-based mechanism for firmware tempering protection to prevent the software and the hardware IP from being copied by third parties. To protect IPs from being copied, cloned, or used with unauthorized integration, Zhang et al. [143] proposed PUF-based IP protection mechanism that restricts IP's execution only on specific FPGA devices, and enforces the pay-per-device licensing. Guo et al. [36] proposed a PUF-based pay-per-device scheme for protecting IPs from attacks based on CNN models.

5.4. Random Number Generation

The fourth application of PUFs is the generation of random numbers used in cryptography as an encryption key. Pseudo-Random Number Generators (PRNGs) were not truly random since the pattern repeated itself after a certain value. In fact, the Hardware Random Number Generators (HRNGs) are used to generate a true random without any initial condition [82]. By exploiting the randomness found in the inherent nature of the silicon PUFs, it could be used as a source of random number generation. Kalanadhabhatta et al. [49] used the PUF response as an initial seed and Kaya [52] combined Chua circuits with PUFs. The former are a type of chaotic system that has the ability to produce different results from a fractional change in the initial conditions. PUFs are used as a random number generation mechanism to be used in cryptographic systems.

5.5. Payment

Electronic money (e-money or e-Cash) is the digital representation of physical banknotes where authentication, encryption, privacy, and anonymity play central roles. To not steal, predict, and/or clone tokens used by a device, Calhoun et al. [14] introduced an e-Cash based on PUFs called PUF-Cash, where the PUFs response is used in the authentication bit-strings, encryption keys, and e-Cash token generation. In [137], Yang et al. combined Leveraging TrustZone and SRAM PUFs technology to design the architecture of trusted mobile, which can be used in e-payment schemes while guaranteeing the anonymity of the users' identities to other entities, such as banks and merchants. Kish et al. [54] proposed a credit/debit PUF equipped with a weak PUF chip responsible for secure communication, data authentication and a private key stored by a customer.

5.6. Memory protection

In this field, the PUF's output is used to secure program execution by protecting the confidentiality and integrity of the memory instructions and the stored data against physical and software attacks [125].

5.7. Software licensing

Software licensing is a way to protect software from unauthorized modifications and from running on unauthorized platforms. To achieve this protection, many approaches were proposed such as the use of a hash function or checksum, where each block code has its own hash value to be checked in the next bloc by verifying the integrity of the first one. Other solutions include the use of the obfuscation method to protect software from reverse engineering and malicious modification. Meanwhile, software protection-based PUF has been proposed. The idea is to provide software with the possibility to communicate with the PUF to perform some operations based on the generated keys where both static and dynamic PUF are used with more intention on the dynamic one [135]. Suresh and Manimegalai [126] proposed a software licensing mechanism based on SRAM PUF. In this scheme, the user's PC is equipped with the SRAM PUF, giving it a unique identity. When the user needs to buy a needed software, a company will initiate a connection with the user's PC to have the SRAM PUF's outputs and make them available in the software as a license. The customer installs the software, and during installation, an authentication mechanism occurs between the software and the PC, where the embedded license is compared to the SRAM PUF's outputs. Kohnhäuser et al. [55] combined self-checksumming code techniques with PUFs to establish hardware-assisted software protection. The self-checksumming code is used to check the program's integrity and protect it against tampering attacks. Then, PUFs guarantee the execution of the software instance only on the specific device (hardware) equipped with the right PUF.

5.8. Securing communication

As we presented before, PUF is widely used in the authentication protocol, especially when launching communication in many use cases. Another application of silicon PUF is communication, where the generated response will be used to guarantee secure communication by ensuring the confidentiality and integrity as well as non-repudiation of the exchanged messages.Zheng et al. [146] proposed a PUF-based key-exchange protocol between IoT devices without the need for a trusted entity. After a successful authentication phase, both devices use PUF data to construct and exchange the session key to secure the communication. Also, Mahmood et al. [77] used Elliptic Curve Cryptography and PUFs to secure device-to-device communication. The registration centre is responsible for authentication and session key generation in this scheme. However, error corrections have not been considered in this mechanism.

5.9. Comparison

Table 3 classifies and compares the surveyed contributions related to silicon PUF-based applications. We consider different criteria, including the area of application and the specification of use cases. We also show if the surveyed work relies only on silicon PUFs or uses other security primitives like hashing and cryptographic functions. Also, we consider the integration of noise cleaning and error correction. Further, PUF implementation is verified by checking if the proposed work indicates the architecture of the used PUFs. Based on this comparison, PUFs are used in many applications and fields, from cryptographic key generation to e-payment. Most of the proposed work in different application areas uses extra security primitives to achieve their objectives such as the hashing function and XORing operation. Also, sometimes PUFs are combined with elliptic curve cryptography or blockchains. From the reliability side, most of the discussed works do not consider error correction and noise elimination process in their proposed scheme, making their solution impractical in any application and use case area since the PUFs reliability is considered as a principal metric that could gauge the efficiency of any proposed PUFs based scheme. Also, we observed that most of the discussed works do not indicate the architecture of the deployed PUF in their proposed scheme.

6. PUFs Performances and Attacks

In this section, we surveyed the existing metrics used to evaluate PUFs and studied the existing attacks and countermeasures related to PUFs.

6.1. PUF performance

To evaluate the performance of a given PUF, we consider the metrics shown in Figure 40: uniqueness, steadiness, randomness, correctness, bit aliasing, uniformity, reliability, diffuseness and security [79, 42, 104].

For a better mathematical formulation of these metrics, we first use the notation shown in Table 4.

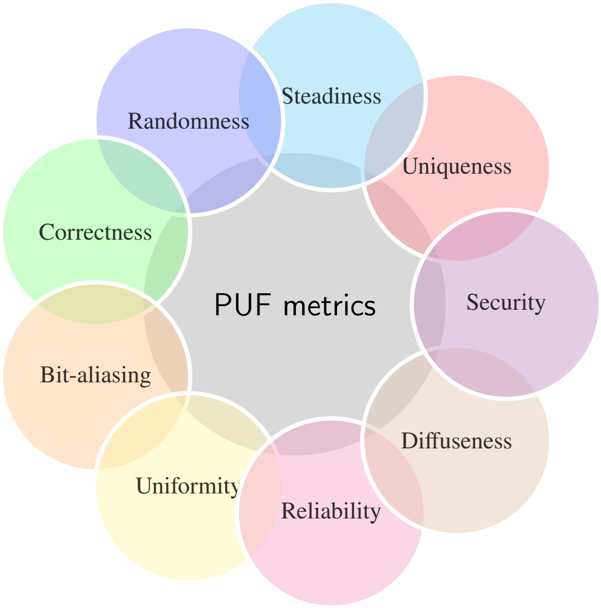


Figure 40: The metrics of PUFs.

6.1.1. *Uniqueness*

Let us consider two PUFs with the exact implementation embedded into two devices d_1 and d_2 that generate respectively responses $R_{d1,m}$ and $R_{d2,m}$ to the same challenge c under the same measurement m where both responses must be very different. Thus, the uniqueness requirement measures how much a PUF instance is different from others by evaluating the uncorrelated responses across dying. When the same challenge sets are presented to different PUFs, the response of each PUF is expected to be different. Uniqueness indicates that responses resulting from evaluating the same challenge on different PUF instances should be dissimilar with a high probability.

Since the generated response can be (or be transformed into) a vector of bits, the Hamming distance (HD) will compare two-bit vectors. The HD is the number of positions in which two PUF responses are different, e.g. “11011” and “11011” is 0, while the HD of “11011” and “10101” is 3. The device uniqueness is defined as follows:

$$Uniqueness = \frac{2}{D(D-1)} \frac{1}{P} \sum_{d_1=1}^{D-1} \sum_{d_2=d_1+1}^{D} HD(R_{d_1,m}, R_{d_2,m})$$

The main function in this formula is the hamming distance calculation given by HD. It calculates the sum of XOR operations between each binary response bit $r_{p,d_1,m}$ and $r_{p,d_2,m}$ of two responses $R_{d_1,m}$ and $R_{d_2,m}$ in the m measurement. Ideally, the uniqueness should be close to 50%.

$$HD = \sum_{p=1}^{P} (r_{p,d_1,m} \oplus r_{p,d_2,m})$$

As illustrated in Figure 41, the two instances d_1 and d_2 of a given PUF are assumed to be implemented on two different chips. When a challenge (c_1= 1010) is presented in both

Table 3
Comparison of the Applications and Use Cases of Silicon PUF.

Application	Use case	Works	Year	Only PUFs	Error Correction	Type PUF
Authentication	IoT	Idriss et al.[45]	2021	No	No	Delay-based PUF
		Najafi et al.[92]	2021	Yes	No	DRAM PUF
		Aman et al.[5]	2020	No	No	-
		Muhal et al.[91]	2018	No	No	-
		Mostafa et al.[90]	2020	No	No	Arbiter and SRAM PUF
		Aman et al.[6]	2017	No	No	-
	UAV	Pu and Li[102]	2020	No	No	-
		Alladi et al. [4]	2020	No	No	-
		Bansal and Sikdar [12]	2021	No	No	-
	IoMT	Yanambaka et al. [136]	2019	No	No	Arbiter PUF
		Wang et al. [133]	2021	No	yes	-
		Lee and Chen [62]	2021	No	No	BS-PUF
		Gope et al. [33]	2021	-	-	-
		Kwon et al. [60]	2021	No	Yes	-
	IoV	Jiang et al. [46]	2019	No	Yes	-
		Jiang et al. [47]	2021	No	Yes	-
		Mershad et al. [86]	2021	No	No	-
	Smart grid	Kaveh et al. [51]	2020	-	-	-
		Cao et al. [15]	2021	No	Yes	-
		Badar et al.[11]	2021	No	No	-
IP Protection	Hardware IP	Guajardo et al.[35]	2007	-	Yes	SRAM PUF
	Software IP	Zheng and Potkonjak[145]	2014	-	No	drPUFs
	Pay-per-device	Zhang et al.[143]	2015	-	No	Delay-based PUF
		Guo et al.[36]	2018	-	-	-
Payment	e-Cash	Calhoun et al.[14]	2019	-	No	HELP [1]
		Yang et al. [137]	2016	No	yes	SRAM PUF
	Credit cards	Kish et al. [54]	2017	-	No	-
Licensing	Software	Xiong et al. [135]	2019	No	No	-
		Suresh and Manimegalai [126]	2018	No	Yes	SRAM PUF
		Kohnhäuser et al. [55]	2015	No	Yes	SRAM PUF
Securing communication	IoT	Zheng et al.[146]	2021	No	Yes	-
		Mahmood et al. [77]	2021	No	No	-
Memory protection	-	Suh et al.[125]	2007	No	No	RO PUF

instances, with the same measurement m, each PUF generates a unique response. In this case, the HD between both of them is 2 which means the uniqueness of this PUF is 37.5%.

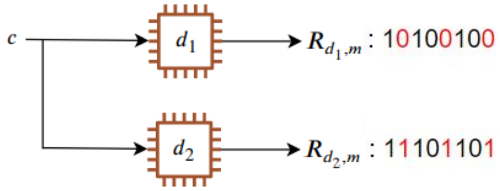


Figure 41: An example of the uniqueness evaluation of a given PUF design.

6.1.2. *Reliability*

This requirement shows how stable a PUF design is when the same challenge values are stimulated for a given PUF instance while the latter should generate the same response values. It measures the repeatability and the consistency with which a PUF generates its response across environmental variations such as ambient noise and aging. To measure the reliability of a PUF, we evaluate the deviation/bias degree of a response generated from the same challenge across different measurements.

For a given device d_1 which has a response R_{d,m_1} of P-bit reference at normal operating conditions m_1 and the response R_{d,m_2} of P-bit at different conditions m_2 for the same challenge c, the reliability is defined as follows using Hamming distance, where less reliability means more changes and instability. The optimal value of the reliability indicator should be 100%.

$$Reliability = 1 - \frac{1}{RMP}\sum_{m_2=2}^{M}\sum_{p=1}^{P}(r_{p,d,m_1} \oplus r_{p,d,m_2})$$

By taking the example of the device d_1 depicted in Figure 42, when a challenge (c_1 = 1010) is applied on this device at two different temperatures, m_1 (30k) and m_2 (80k), we observe that the initial response '10100101' differs from the second one '10101101' only on one bit. Then, the reliability

Table 4
The notation symbols.

Symbol	Description
d	Index of a device ($1 \leq d \leq D$).
D	Number of devices.
c	Index of a challenge ($1 \leq c \leq C$).
C	Number of challenges.
p	Index of a PUF bit within a response vector ($1 \leq p \leq P$).
P	Length of a PUF response vector.
M	Number of measurements.
m	Index of a measurement at a specific time under an environmental condition ($1 \leq m \leq M$).
R	Number of responses generated by a device.
r	Index of a response $1 \leq r \leq R$.
$r_{p,d,m}$	Binary response bit p of a device d within a measurement m.
$R_{d,m}$	Response vector of a device d for a measurement m with $R_{d,m} = \{0,1\}^P$.

of the PUF design embedded on d_1 is 93.75, which means it is not a very reliable design.

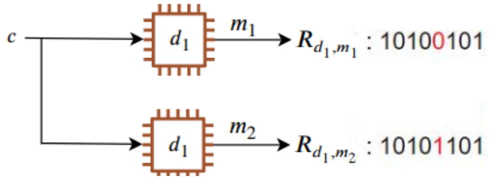


Figure 42: An example of the reliability evaluation of a PUF design.

6.1.3. Uniformity

This metric estimates how uniform the n-bit of a response R$_{d,m}$ are distributed by measuring the percentage of '0's and '1's in the response bits. For an excellent response, the proportion of '0's and '1's in its responses should be equal to 50%. The PUF instance is biased towards '0' or '1' in its responses. In this case, the attacker can guess that response. The uniformity of the response bits $R_{d,m}$ is defined as the percentage of Hamming weight (HW) of the n-bit response. So, the uniformity of a response R$_{d,m}$ for a PUF instance d, generated with m measurement, is defined by:

$$Uniformity = \frac{1}{P}\sum_{p=1}^{P}(r_{p,d,m})$$

Taking for example the 8-bit response of '01010101'. The HW of this response is 4 and its uniformity is 50%, which makes it uniform. This due to the same ratio of ones and zeros in the given 8-bit response.

6.1.4. Randomness

The P-bit response of a PUF is expected to be uniformly distributed, so the randomness measures the balance of ones and zeros of the response bits value r$_{p,d1,m}$. The optimal value of the randomness metric is 100%, and for a device d, it is calculated by:

$$Randomness = -log_2 max(p_d, 1-p_d)$$

p_d is the relative frequency of '1' appearing in all the response bits R generated in a device d at different measurements m, and it is given by:

$$p_d = \frac{1}{RMP}\sum_{r=1}^{R}\sum_{m=1}^{M}\sum_{p=1}^{P} r_{p,d,m}$$

6.1.5. Correctness

This requirement gauges how well the PUF responses are accurate. Imagine that a part of the device where a PUF is embedded is broken for some reason after the correct response r for a given challenge c is determined. Then, compared with the correct response, a PUF instance on the device could always generate a wrong response bit value for the same challenge c. In this case, the new response r' becomes stable but incorrect. The correctness requirement is to determine if such a device is defective or degraded by aging. Correctness is similar to reliability, and its ideal value is 100%. It is calculated as follows.

$$Correctness = 1 - \frac{2}{RMP}\sum_{m_2=2}^{M}\sum_{p=1}^{P}(r_{p,d,m_1} \oplus r_{p,d,m_2})$$

The relationship between reliability and correctness is defined by:

$$Correctness = (2 * Reliability) - 1$$

6.1.6. Bit-aliasing

This metric estimates the bias of a particular response bit among the set of PUF instances. The bit-aliasing of $r_{p,d_1,m}$ for a challenge c is estimated as the average Hamming weight of the p^{th} bit across different PUF instances. Ideally, this value should be around 50%. The bit-aliasing of the p^{th} response bit generated on the same measurement m, across D different devices is given by:

$$Bit - Aliasing = \frac{1}{D}\sum_{d=1}^{D} r_{p,d,m}$$

Taking for example the challenge ($c_1 = 1010$) applied to three different devices d_1, d_2 and d_3 with the same measurement m (see Figure 43), the HW value of the 1^{th} bit is 2. so, the bit-aliasing of this bit is 67% which means that this bit is biased towards binary value 1.

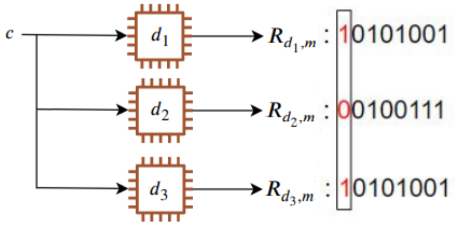


Figure 43: An example of bit-aliasing evaluation of the 1^{th} bit.

6.1.7. *Steadiness*

When applying the same challenge c to the same device d_1 with different measurements m, the output responses R are expected to be identical. The steadiness measures the degree of bias of the p^{th} response bit $r_{p,d_1,m}$ for the given challenge. Steadiness is how strongly $\mathrm{r}_{p,d_1,m}$ is biased toward 0 or 1, and its optimal value is 100%. That is calculated as follows.

$$Steadiness = 1 + \frac{1}{RP}\sum_{r=1}^{R}\sum_{p=1}^{P} log_2 max(p_d, 1 - p_d)$$

where

$$p_d = \frac{1}{M}\sum_{m=1}^{M} r_{p,d,m}$$

Figure 44 represents the steadiness metrics of the 1^{th} bit of the response generated for the same challenge c_1 on the same device d_1 with two different measurements, m_1 (30k) and m_2 (80k). The steadiness of this bit challenge is 25% which means that this bit is biased towards binary value 0.

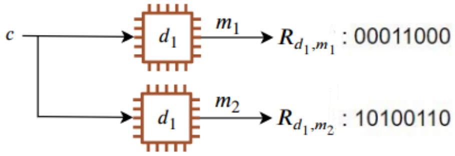


Figure 44: An example of the steadiness evaluation of the 1^{th} bit.

6.1.8. *Diffuseness*

As a result of applying different challenges to the same PUF instance, the generated responses should be different. The diffuseness represents the difference among the generated responses from different challenge sets in the same PUF instance. Using the mean Hamming distance (HD) of the generated responses R from the challenges C, on the same device d_1 and with the same measurement m. The diffuseness of the device d is defined by:

$$Diffuseness = \frac{4}{P.R^2}\sum_{r_1=1}^{R-1}\sum_{r_2=r_1+1}^{R}\sum_{p=1}^{P}(r_{1(p,d,m)} \oplus r_{2(p,d,m)})$$

when applying the two challenges (c_1 = 1010) and (c_2 = 0101) shown in the Figure 45 on the same device d_1, with the same measurement m, this device produces two different responses ($R_{d_1,m}$ = 10001010) and ($R'_{d_1,m}$ = 10100110). The diffuseness of this device is 37.5% which means both responses differ by 37.5%.

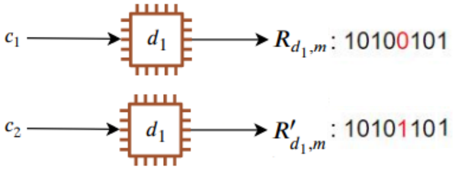


Figure 45: An example of the diffuseness evaluation of a PUF design.

Any cryptosystem is exposed to classical cryptosystem attacks like trying to read out secret keys from memory and communication attacks, in addition to two new threats: Side-Channel Attacks, where an attacker has physical access to the device, and modeling attacks, where the adversary has a large number of CRPs. In addition to these metrics, some researchers consider security as a metric too[57].

6.1.9. *Security*

It is the ability of a PUF to resist all attacks. In contrast to the previous metrics, there is no specific formula to evaluate the security of PUFs.

6.2. PUF Attacks

PUFs are a great solution to replace the actual protection mechanisms, such as hash functions and secret-key algorithms. Unfortunately, they are not entirely secure and suffer from some vulnerabilities. Strong PUFs are very difficult to break, and the current technologies are not advanced enough to manage to break them compared to weak PUFs that can be easily broken by different types of attacks. Security attacks are generally classified into invasive, semi-invasive, and non-invasive attacks [37].

6.2.1. *Non-invasive attacks*

Non-invasive are low-cost attacks based only on observation and speculation about the device without harming it. This type of attack does not need much equipment as the field of action is restricted. The attackers extract secret information by exploiting only the observed data (e.g., power consumption, delay time, and CRPs) without direct access to PUF components. The two most celebrated types of non-invasive attacks are machine learning (ML), and side-channel attacks (SCAs) [78].

- Machine learning attacks: Strong PUF's challenge-response behaviour is vulnerable to modeling attacks, which use machine learning algorithms to predict PUF responses. The principle of machine learning is to create a specific algorithm and give it some sample data to train the algorithm and create a statistical model that will simulate the PUF's behavior. Then, it will be easy to request the model to predict new data. After constructing an adversary machine and collecting a subset of all the CRPs of the target PUF, the attacker can build a numerical model from the collected CRP data. Thus, the model can be used for future response prediction to arbitrary challenges with high probability. In [109], Rührmair et al. showed that several PUF architectures [81, 9, 106, 124, 61, 31] can be broken using various machine learning techniques, including Support Vector Machines (SVMs), Evolution Strategies (ES), Logistic Regression (LR), and also briefly Neural Nets and Sequence Learning [110]. In [27], probably approximately correct (PAC) learning has been used to develop attack models for Arbiter, XOR Arbiter, RO-, and BR-PUFs. To check the robustness of PUFs towards ML attacks, Ganji et al. [28] developed a testing environment, called PUFmeter, to evaluate the security of PUFs under ML attacks. Also, Chatterjee et al. [18] relied on the PAC-learnability of PUFs to derive the PAC-learnability bound from the representation of a PUF architecture described in the PUF-G language [18]. Then, they verify the robustness and resilience of the given PUF against ML-based attacks.

As a countermeasure against ML attacks, Rührmair et al. [111] proposed raising the number of XORs in an XOR Arbiter PUF and a Lightweight PUF. In order to mitigate the PUF modeling vulnerability, Vatajelu et al. [131] proposed to encrypt challenge-bit via the AES algorithm, where the encryption key is generated using a weak PUF. Similarly in [26], instead of storing the response, the hash value of the PUF response is stored on the server. To resist against ML attacks, Dubrova et al. [24] proposed a new PUF construction called a CRC-PUF where the input challenges are desynchronized from the output responses. Mispan et al. [88] proposed a challenge permutation and substitution techniques to increase the ML-attack resistance of Strong-PUFs. They showed that the predictability of Arbiter-PUF and TCO-PUF responses could be reduced to less than 70%. Kroeger et al. [56] showed that challenge obfuscation schemes implemented on a standard arbiter-PUF makes it secure against modeling attacks.

- Side-channel attacks: Another non-invasive example that is (hopefully) more complex is the side-channel attack [13], well-known in cryptanalysis. The idea is to exploit leaked information from the physical implementation of a cryptographic primitive of the device running the algorithm or the physical system regarding PUFs and extract information as much as possible to understand the algorithm's behaviour. For example, when attempting to break RSA, some attackers tried to measure their CPU usage to understand when the most extensive computation occurs, which one, and even the produced data. This type of attack is classified into two groups: passive and active attacks. The attacker observes side channels in the first case, such as timing delays, power consumption, temperature, and electromagnetic noise. In the second, the attacker requires information on the internal structure and operation of the PUF, such as fault injection methods [98]. Various s-channel techniques have been applied to PUFs, such as Helper Data Leakage, Power Analysis Attacks, and Fault Injection Attacks [37]. Karakoyunlu and Sunar [50] exploited the information leaked through the power side-channel in the initial step in the syndrome decoding phase of BCH and Reed-Solomon decoder fuzzy extractor implementations to recover the fuzzy extractor's input that refers to the PUF's response. Merli et al. [84] have demonstrated how RO PUFs can be attacked using electromagnetic (EM) attacks. In [128], Tebelmann et al. analyzed side-channel vulnerabilities of the Loop PUF and showed that it is vulnerable to side-channel analysis (SCA) attacks. Rührmair et al. [112] proposed a power consumption and time-side channel attack method for XOR PUF and lightweight security PUF. Kroeger et al. [56] showed that arbiter-PUFs based challenge obfuscation schemes are vulnerable to power side-channel attacks. In [78], authors have also proposed a combined side-channel and modeling attack.

  As a countermeasure against PUF Side-channel attacks, Merli et al. [84] proposed a measurement path randomization by randomizing the RO selection logic and the interleaved placement to disguise RO EM emission as two countermeasures. Also, to mitigate the attack presented in [128], they introduced a countermeasure based on temporal masking to thwart side-channel analysis that requires only one bit of randomness per a PUF response bit. Kroeger et al. [56] proposed dual flip-flop mitigation and randomized response settings to improve the resiliency of challenge obfuscation PUFs against power trace attacks.

6.2.2. Invasive attacks

This type of attack is among the most expensive ones where the attacker can extract information from the system, understand the internal behavior, and access the device. However, the required equipment to achieve this type of attack is expensive and requires more knowledge and time. However, invasive attacks are less popular, and they need more sophisticated equipment and precise expertise. Micro-probing and reverse engineering are two types of invasive attacks [67]. The first one requires a micro-probing station, a gigantic microscope with some probes to get information, such as the electric signals, from the circuits to understand the interactions between the different components and when they need to communicate. This station can also be used to alter the device, as we can manipulate the system. And the second one requires observing and manipulating the device or the software to derive some information about its behaviour. Then, the attacker can attempt to reproduce it.

6.2.3. Semi-invasive attacks

This class is a compromise between the attacks categories mentioned above. In terms of requirements (affordability and knowledge), it is between invasive and non-invasive. In addition, there are many other types of attacks appropriate to a specific target of a PUF design. They have access to some parts of the internal devices in a system without damaging them. For example, an attacker can add extra circuitry with malicious functionality into a PUF design. When the PUF is used, the attacker uses this circuit to access the PUF. This attack is called 'Trojan insertion'. Also, the PUF can be attacked by exploiting the vulnerabilities of the application domain, like the man-in-the-middle attack, where the attacker tries to intercept the transport data used in the authentication protocol [37].

7. Discussion and perspectives

There have been almost two decades of intensive research on PUFs since the concept was first introduced by Pappu *et al.* [100]. Physical unclonable functions have a completely different system than any other one-way function, especially with the challenge-response pair sets providing better reliability. Also, the level of defence is good with PUFs, and the variability of PUF systems allows users to choose the function with the best characteristics according to the needs of the applications. Silicon PUF is one of the widely accepted hardware security primitives that finds application in authentication and secret key generation. It generates a secured key by the physical disorder nature of an electronic system. The physical structure of every electronic system is unique due to the inheriting differences during the manufacturing process using the same technology.

This paper surveyed Physical Unclonable Functions in general and specifically the silicon PUFs. First, we presented a comparison of the existing PUFs to show the novelty of the present survey. Then, we presented the different aspects and concepts that allowed the birth of PUFs. Also, we provided the needed PUF properties and their classifications regarding the implementation technology, the size of challenge-response pairs (CRPs), the response's dependency, and the physical construction properties. After that, we surveyed the state-of-the-art silicon PUFs architectures and classified them into delay-based PUFs, memory-based PUFs, and analog electronic PUFs. Additionally, we listed the most existing implementations for each class and explained their operating processes with graphical representations. Furthermore, we have also surveyed and classified the existing Silicon PUF applications and use cases. In addition, we have presented the nine metrics used to evaluate PUFs by giving the mathematical formula and the graphical representation for each metric, except the security one. Also, we classified the existing PUF attacks into non-invasive, invasive, and semi-invasive.

Compared with memory-based PUFs that are primarily intrinsic, delay-based PUFs and analog electronic PUFs require dedicated circuits with a complex design and manufacturing process. Also, it is hard, if not impossible, to find these dedicated circuits in existing silicon circuits, and sometimes they require additional hardware to generate a good response. This makes memory-based PUF a preferable class for applications where the silicon chip is part of the integrated circuits.

PUFs are mainly used for authentication and secret key generation but, still with some drawbacks regarding attacks because there are multiple ways to attack, and there is no function able to block every method. More generally, this creates truly random numbers that are highly difficult to clone and predict. This could be helpful in many cases for science and physics, where this is not a question of security. Also, these PUF methods are scalable even in environments such as the IoT field, which allows devices to be more secured than before. In conclusion, we think that this method is, for now, one of the most promising that exists, but the implementation and conception require considerable work in comparison with other "regular" protocols.

The next decade will follow the evolution of PUF usage because the high capacity to evolve makes this technology promising, especially in fields where security is not good yet, such as the authentication methods for IoT systems. Many open questions and research directions might be investigated soon, especially:

- *Standardisation:* As we presented in Section 6.1, nine metrics exist to evaluate the PUF's performance, some of which look similar. We believe it is the moment to standardize how to measure PUFs' performance regarding the nine existing metrics. It is also important to standardize the way they can be deployed, especially their use will be more widespread in the near future.

- *Security:* As discussed in Section 6.2, several attacks have been reported to break the PUF security, especially for strong PUFs susceptible to modeling attacks. So, it is interesting to develop practical and effective solutions to address modeling attacks and investigate more about the security of memory-based PUFs since they are widely used in our daily lives.

- *Environmental Influences*: Error correction techniques are essential to developing any PUF-based security mechanism. As discussed in Section 5.2, the PUF outputs are affected by the environmental conditions, causing errors in the generated response. Consequently, this phenomenon makes PUFs impractical in security and cryptographic applications that require the reproduction of the exact key several times. From this perspective, any new or existing PUF-based scheme should consider robust and efficient error correction techniques.

- *Hybrid PUFs*: When two PUFs are combined, they exhibit strong security characteristics; thus, it is worthwhile to examine which existing PUF design is suitable for pairing, how to evaluate the composition, and whether a particular application area is the most appropriate.

- *New PUFs:* Analyzing possible variations of existing chips (IoT, smartphones, smart TVs, etc.) and categorizing PUF classes and designs that are might be more robust and secure for specific applications.
- *Constrained devices:* In this survey, especially in Section 3, PUF is described as a hardware security primitive that does not require any information to be stored in the device memory. However, most of the current works, particularly in the authentication applications, contradict this criterion. Therefore, PUF-based authentication protocols should consider the capabilities of constrained objects.
- *Application:* Looking at how we can exploit better the existing PUFs on the different areas of interest already classified in section 5 and searching for new applications' area such as agriculture.
- *Tooling:* Research on PUFs of various inputs (audio, images, videos, etc.) should benefit from a free, open-source, and online solution that helps develop experimental and benchmarking data sets and helps researchers test the performance and attacks on simulated and concrete PUFs implementations.

References

[1] Aarestad, J., Ortiz, P., Acharyya, D., Plusquellic, J., 2013. Help: A hardware-embedded delay puf. IEEE Design & Test 30, 17–25.

[2] Adames, I.A.B., Das, J., Bhanja, S., 2016. Survey of emerging technology based physical unclonable funtions, in: 2016 International Great Lakes Symposium on VLSI (GLSVLSI), IEEE. pp. 317–322.

[3] Al-Haidary, M., Nasir, Q., 2019. Physically unclonable functions (pufs): A systematic literature review, in: 2019 Advances in Science and Engineering Technology International Conferences (ASET), IEEE. pp. 1–6.

[4] Alladi, T., Bansal, G., Chamola, V., Guizani, M., et al., 2020. Secauthuav: A novel authentication scheme for uav-ground station and uav-uav communication. IEEE Transactions on Vehicular Technology 69, 15068–15077.

[5] Aman, M.N., Chaudhry, S.A., Al-Turjman, F., 2020. Rapidauth: Fast authentication for sustainable iot, in: International Conference on Forthcoming Networks and Sustainability in the IoT Era, Springer. pp. 82–95.

[6] Aman, M.N., Chua, K.C., Sikdar, B., 2017. Mutual authentication in iot systems using physical unclonable functions. IEEE Internet of Things Journal 4, 1327–1340.

[7] Anandakumar, N.N., Hashmi, M.S., Tehranipoor, M., 2021. Fpga-based physical unclonable functions: A comprehensive overview of theory and architectures. Integration 81, 175–194.

[8] Ardakani, A., Shokouhi, S.B., Reyhani-Masoleh, A., 2018. Improving performance of fpga-based sr-latch puf using transient effect ring oscillator and programmable delay lines. Integration 62, 371–381.

[9] Avvaru, S.S., Parhi, K.K., 2019. Feed-forward xor pufs: Reliability and attack-resistance analysis, in: Proceedings of the 2019 on Great Lakes Symposium on VLSI, pp. 287–290.

[10] Avvaru, S.S., Zeng, Z., Parhi, K.K., 2020. Homogeneous and heterogeneous feed-forward xor physical unclonable functions. IEEE Transactions on Information Forensics and Security 15, 2485–2498.

[11] Badar, H.M.S., Qadri, S., Shamshad, S., Ayub, M.F., Mahmood, K., Kumar, N., 2021. An identity based authentication protocol for smart grid environment using physical uncloneable function. IEEE Transactions on Smart Grid 12, 4426–4434.

[12] Bansal, G., Sikdar, B., 2021. S-maps: Scalable mutual authentication protocol for dynamic uav swarms. IEEE Transactions on Vehicular Technology 70, 12088–12100.

[13] Cai, X., Xie, G., Kuang, S., Li, R., Li, S., 2021. Efficient dpa side channel countermeasure with mim capacitors-based current equalizer. Journal of Systems Architecture 118, 102146.

[14] Calhoun, J., Minwalla, C., Helmich, C., Saqib, F., Che, W., Plusquellic, J., 2019. Physical unclonable function (puf)-based e-cash transaction protocol (puf-cash). Cryptography 3, 18.

[15] Cao, Y.N., Wang, Y., Ding, Y., Zheng, H., Guan, Z., Wang, H., 2021. A puf-based lightweight authenticated metering data collection scheme with privacy protection in smart grid, in: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE. pp. 876–883.

[16] Chang, C.H., Zheng, Y., Zhang, L., 2017. A retrospective and a look forward: Fifteen years of physical unclonable function advancement. IEEE Circuits and Systems Magazine 17, 32–62.

[17] Chatterjee, D., Mukhopadhyay, D., Hazra, A., 2020a. Interpose puf can be pac learned. IACR Cryptol. ePrint Arch. 2020, 471.

[18] Chatterjee, D., Mukhopadhyay, D., Hazra, A., 2020b. Puf-g: A cad framework for automated assessment of provable learnability from formal puf representations, in: Proceedings of the 39th International Conference on Computer-Aided Design, pp. 1–9.

[19] Chen, Q., Csaba, G., Lugli, P., Schlichtmann, U., Rührmair, U., 2011. The bistable ring puf: A new architecture for strong physical unclonable functions, in: 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, IEEE. pp. 134–141.

[20] Della Sala, R., Bellizia, D., Scotti, G., 2021. A novel ultra-compact fpga puf: The dd-puf. Cryptography 5, 23.

[21] Delvaux, J., Peeters, R., Gu, D., Verbauwhede, I., 2015. A survey on lightweight entity authentication with strong pufs. ACM Computing Surveys (CSUR) 48, 1–42.

[22] van Dijk, M., Rührmair, U., 2014. Protocol attacks on advanced puf protocols and countermeasures, in: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE. pp. 1–6.

[23] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A., 2008. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM journal on computing 38, 97–139.

[24] Dubrova, E., Näslund, O., Degen, B., Gawell, A., Yu, Y., 2019. Crc-puf: A machine learning attack resistant lightweight puf construction, in: 2019 IEEE European symposium on security and privacy workshops (EuroS&PW), IEEE. pp. 264–271.

[25] El-Hajj, M., Fadlallah, A., Chamoun, M., Serhrouchni, A., . Secure puf: Physically unclonable function based on arbiter with enhanced resistance against machine learning (ml) attacks, in: SEIA'2019 Conference Proceedings, Lulu. com. p. 216.

[26] Farha, F., Ning, H., Ali, K., Chen, L., Nugent, C., 2020. Sram-puf-based entities authentication scheme for resource-constrained iot devices. IEEE Internet of Things Journal 8, 5904–5913.

[27] Ganji, F., 2018. On the learnability of physically unclonable functions. Springer.

[28] Ganji, F., Forte, D., Seifert, J.P., 2019. Pufmeter a property testing tool for assessing the robustness of physically unclonable functions to machine learning attacks. IEEE Access 7, 122513–122521.

[29] Gao, M., Lai, K., Qu, G., 2014. A highly flexible ring oscillator puf, in: Proceedings of the 51st Annual Design Automation Conference, pp. 1–6.

[30] Gao, Y., Ranasinghe, D.C., Al-Sarawi, S.F., Kavehei, O., Abbott, D., 2016. Emerging physical unclonable functions with nanotechnology. IEEE access 4, 61–80.

[31] Gassend, B., Clarke, D., Van Dijk, M., Devadas, S., 2002. Silicon physical random functions, in: Proceedings of the 9th ACM confer-

ence on Computer and communications security, pp. 148–160.

[32] Gebali, F., Mamun, M., 2022. Review of physically unclonable functions (pufs): Structures, models, and algorithms. Front. Sens. 2: 751748. doi: 10.3389/fsens .

[33] Gope, P., Millwood, O., Sikdar, B., 2021. A scalable protocol level approach to prevent machine learning attacks on physically unclonable function based authentication mechanisms for internet of medical things. IEEE Transactions on Industrial Informatics 18, 1971–1980.

[34] Gu, C., Hanley, N., O'neill, M., 2017. Improved reliability of fpga-based puf identification generator design. ACM Transactions on Reconfigurable Technology and Systems (TRETS) 10, 1–23.

[35] Guajardo, J., Kumar, S.S., Schrijen, G.J., Tuyls, P., 2007. Fpga intrinsic pufs and their use for ip protection, in: International workshop on cryptographic hardware and embedded systems, Springer. pp. 63–80.

[36] Guo, Q., Ye, J., Gong, Y., Hu, Y., Li, X., 2018. Puf based pay-per-device scheme for ip protection of cnn model, in: 2018 IEEE 27th Asian Test Symposium (ATS), IEEE. pp. 115–120.

[37] Halak, B., 2018. Physically Unclonable Functions: From Basic Design Principles to Advanced Hardware Security Applications. Springer.

[38] Helfmeier, C., Boit, C., Nedospasov, D., Seifert, J.P., 2013. Cloning physically unclonable functions, in: 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE. pp. 1–6.

[39] Helinski, R., Acharyya, D., Plusquellic, J., 2009. A physical unclonable function defined using power distribution system equivalent resistance variations, in: 2009 46th ACM/IEEE Design Automation Conference, IEEE. pp. 676–681.

[40] Herder, C., Yu, M.D., Koushanfar, F., Devadas, S., 2014a. Physical unclonable functions and applications: A tutorial. Proceedings of the IEEE 102, 1126–1141.

[41] Herder, C., Yu, M.D., Koushanfar, F., Devadas, S., 2014b. Physical unclonable functions and applications: A tutorial. Proceedings of the IEEE 102, 1126–1141.

[42] Hori, Y., Yoshida, T., Katashita, T., Satoh, A., 2010. Quantitative and statistical performance evaluation of arbiter physical unclonable functions on fpgas, in: 2010 International Conference on Reconfigurable Computing and FPGAs, IEEE. pp. 298–303.

[43] Huang, J.Q., Zhu, M., Liu, B., Ge, W., . 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), IEEE. pp. 1–3.

[44] Huang, Z., Li, L., Chen, Y., Li, Z., Wang, Q., Jiang, X., 2021. Rp-puf: An ultra-lightweight reconfigurable pico-physically unclonable function for resource-constrained iot devices. Electronics 10, 3039.

[45] Idriss, T.A., Idriss, H.A., Bayoumi, M.A., 2021. A lightweight puf-based authentication protocol using secret pattern recognition for constrained iot devices. IEEE Access .

[46] Jiang, Q., Zhang, X., Zhang, N., Tian, Y., Ma, X., Ma, J., 2019. Two-factor authentication protocol using physical unclonable function for iov, in: 2019 IEEE/CIC International Conference on Communications in China (ICCC), IEEE. pp. 195–200.

[47] Jiang, Q., Zhang, X., Zhang, N., Tian, Y., Ma, X., Ma, J., 2021. Three-factor authentication protocol using physical unclonable function for iov. Computer Communications 173, 45–55.

[48] Joshi, S., Mohanty, S.P., Kougianos, E., 2017. Everything you wanted to know about pufs. IEEE Potentials 36, 38–46.

[49] Kalanadhabhatta, S., Kumar, D., Anumandla, K.K., Reddy, S.A., Acharyya, A., 2020. Puf-based secure chaotic random number generator design methodology. IEEE transactions on very large scale integration (VLSI) systems 28, 1740–1744.

[50] Karakoyunlu, D., Sunar, B., 2010. Differential template attacks on puf enabled cryptographic devices, in: 2010 IEEE International Workshop on Information Forensics and Security, IEEE. pp. 1–6.

[51] Kaveh, M., Aghapour, S., Martin, D., Mosavi, M.R., 2020. A secure lightweight signcryption scheme for smart grid communications using reliable physically unclonable function, in: 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), IEEE. pp. 1–6.

[52] Kaya, T., 2020. A true random number generator based on a chua and ro-puf: design, implementation and statistical analysis. Analog Integrated Circuits and Signal Processing 102, 415–426.

[53] Khalafalla, M., Gebotys, C., 2019. 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), IEEE. pp. 204–209.

[54] Kish, L.B., Entesari, K., Granqvist, C.G., Kwan, C., 2017. Unconditionally secure credit/debit card chip scheme and physical unclonable function. Fluctuation and Noise Letters 16, 1750002.

[55] Kohnhäuser, F., Schaller, A., Katzenbeisser, S., 2015. Puf-based software protection for low-end embedded devices, in: International Conference on Trust and Trustworthy Computing, Springer. pp. 3–21.

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

[57] Kumar, A., Mishra, R.S., Kashwan, K., 2016. Puf based challenge response pair for secured authentication. International Journal of Control Theory and Applications 9, 115–121.

[58] Kumar, S., et al., 2018. Analysis of Machine Learning Modeling Attacks on Ring Oscillator based Hardware Security. Ph.D. thesis. University of Toledo.

[59] Kumar, S.S., Guajardo, J., Maes, R., Schrijen, G.J., Tuyls, P., 2008. The butterfly puf protecting ip on every fpga, in: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, IEEE. pp. 67–70.

[60] Kwon, D., Park, Y., Park, Y., 2021. Provably secure three-factor-based mutual authentication scheme with puf for wireless medical sensor networks. Sensors 21, 6039.

[61] Lee, J.W., Lim, D., Gassend, B., Suh, G.E., Van Dijk, M., Devadas, S., 2004. 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), IEEE. pp. 176–179.

[62] Lee, T.F., Chen, W.Y., 2021. Lightweight fog computing-based authentication protocols using physically unclonable functions for internet of medical things. Journal of Information Security and Applications 59, 102817.

[63] Li, Z., Zhu, L., Huang, M., Chen, Z., Chen, S., Li, B., 2019. Racing apuf: A novel apuf against machine learning attack with high reliability, in: 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), IEEE. pp. 722–726.

[64] Liang, W., Liao, B., Long, J., Jiang, Y., Peng, L., 2016. Study on puf based secure protection for ic design. Microprocessors and Microsystems 45, 56–66.

[65] Lim, D., Lee, J.W., Gassend, B., Suh, G.E., Van Dijk, M., Devadas, S., 2005a. Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13, 1200–1205.

[66] Lim, D., Lee, J.W., Gassend, B., Suh, G.E., Van Dijk, M., Devadas, S., 2005b. Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13, 1200–1205.

[67] Ling, M., Wu, L., Li, X., Zhang, X., Hou, J., Wang, Y., 2012. Design of monitor and protect circuits against fib attack on chip security, in: 2012 Eighth International Conference on Computational Intelligence and Security, IEEE. pp. 530–533.

[68] Lofstrom, K., Daasch, W.R., Taylor, D., 2000. Ic identification circuit using device mismatch, in: 2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056), IEEE. pp. 372–373.

[69] Ma, Q., Gu, C., Hanley, N., Wang, C., Liu, W., O'Neill, M., 2018. A

machine learning attack resistant multi-puf design on fpga, in: 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE. pp. 97–104.

[70] Machida, T., Yamamoto, D., Iwamoto, M., Sakiyama, K., 2014. A new mode of operation for arbiter puf to improve uniqueness on fpga, in: 2014 Federated Conference on Computer Science and Information Systems, IEEE. pp. 871–878.

[71] Machida, T., Yamamoto, D., Iwamoto, M., Sakiyama, K., 2015. A new arbiter puf for enhancing unpredictability on fpga. The Scientific World Journal 2015.

[72] Maes, R., 2013. Physically unclonable functions: Concept and constructions, in: Physically Unclonable Functions. Springer, pp. 11–48.

[73] Maes, R., Tuyls, P., Verbauwhede, I., 2008. Intrinsic pufs from flip-flops on reconfigurable devices, in: 3rd Benelux workshop on information and system security (WISSec 2008), p. 2008.

[74] Maes, R., Verbauwhede, I., 2010a. A discussion on the properties of physically unclonable functions, in: TRUST 2010 Workshop, Berlin.

[75] Maes, R., Verbauwhede, I., 2010b. Physically unclonable functions: A study on the state of the art and future research directions, in: Towards hardware-intrinsic security. Springer, pp. 3–37.

[76] Maes, R., Verbauwhede, I., 2010c. Physically unclonable functions: A study on the state of the art and future research directions, in: Towards Hardware-Intrinsic Security. Springer, pp. 3–37.

[77] Mahmood, K., Shamshad, S., Rana, M., Shafiq, A., Ahmad, S., Akram, M.A., Amin, R., 2021. Puf enable lightweight key-exchange and mutual authentication protocol for multi-server based d2d communication. Journal of Information Security and Applications 61, 102900.

[78] Mahmoud, A., Rührmair, U., Majzoobi, M., Koushanfar, F., 2013. Combined modeling and side channel attacks on strong pufs. IACR Cryptol. ePrint Arch. 2013, 632.

[79] Maiti, A., 2012. A systematic approach to design an efficient physical unclonable function. Ph.D. thesis. Virginia Polytechnic Institute and State University.

[80] Maiti, A., Schaumont, P., 2011. Improved ring oscillator puf: An fpga-friendly secure primitive. Journal of cryptology 24, 375–397.

[81] Majzoobi, M., Koushanfar, F., Potkonjak, M., 2008. Lightweight secure pufs, in: 2008 IEEE/ACM International Conference on Computer-Aided Design, IEEE. pp. 670–673.

[82] Marchand, C., Bossuet, L., Mureddu, U., Bochard, N., Cherkaoui, A., Fischer, V., 2017. Implementation and characterization of a physical unclonable function for iot: a case study with the tero-puf. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 37, 97–109.

[83] McGrath, T., Bagci, I.E., Wang, Z.M., Roedig, U., Young, R.J., 2019. A puf taxonomy. Applied Physics Reviews 6, 011303.

[84] Merli, D., Heyszl, J., Heinz, B., Schuster, D., Stumpf, F., Sigl, G., 2013. Localized electromagnetic analysis of ro pufs, in: 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE. pp. 19–24.

[85] Merli, D., Schuster, D., Stumpf, F., Sigl, G., 2011. Semi-invasive em attack on fpga ro pufs and countermeasures, in: Proceedings of the Workshop on Embedded Systems Security, pp. 1–9.

[86] Mershad, K., Cheikhrouhou, O., Ismail, L., 2021. Proof of accumulated trust: A new consensus protocol for the security of the iov. Vehicular Communications 32, 100392.

[87] Miskelly, J., Gu, C., Ma, Q., Cui, Y., Liu, W., O'Neill, M., 2018. Modelling attack analysis of configurable ring oscillator (cro) puf designs, in: 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), IEEE. pp. 1–5.

[88] Mispan, M.S., Su, H., Zwolinski, M., Halak, B., 2018. Cost-efficient design for modeling attacks resistant pufs, in: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE. pp. 467–472.

[89] Moreno, J.H.M., 2019. Memristive modeling for hardware security applications. Master's thesis. National Institute of Astrophysics, Optics and Electronics. Research institution in San Andres Cholula, Puebla, Mexico.

[90] Mostafa, A., Lee, S.J., Peker, Y.K., 2020. Physical unclonable function and hashing are all you need to mutually authenticate iot devices. Sensors 20, 4361.

[91] Muhal, M.A., Luo, X., Mahmood, Z., Ullah, A., 2018. Physical unclonable function based authentication scheme for smart devices in internet of things, in: 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), IEEE. pp. 160–165.

[92] Najafi, F., Kaveh, M., Martín, D., Reza Mosavi, M., 2021. Deep puf: A highly reliable dram puf-based authentication for iot networks using deep convolutional neural networks. Sensors 21, 2009.

[93] Nguyen, P.H., Sahoo, D.P., Jin, C., Mahmood, K., Rührmair, U., van Dijk, M., 2019. The interpose puf: Secure puf design against state-of-the-art machine learning attacks. IACR Transactions on Cryptographic Hardware and Embedded Systems , 243–290.

[94] Ning, H., Farha, F., Ullah, A., Mao, L., 2020. Physical unclonable function: architectures, applications and challenges for dependable security. IET Circuits, Devices & Systems 14, 407–424.

[95] Nithyanand, R., Sion, R., Solis, J., 2011. Poster: Making the case for intrinsic personal physical unclonable functions (ip-pufs), in: Proceedings of the 18th ACM conference on Computer and communications security, pp. 825–828.

[96] Noor, N.Q.M., Daud, S.M., Ahmad, N.A., Maarop, N., Sa'at, N., Aziz, N., 2017. Defense mechanisms against machine learning modeling attacks on strong physical unclonable functions for iot authentication: a review. Int. J. Adv. Comput. Sci. Appl 8, 99–111.

[97] Ouchani, S., 2021. A security policy hardening framework for socio-cyber-physical systems. Journal of Systems Architecture 119, 102259.

[98] Ozturk, E., Hammouri, G., Sunar, B., 2008. Physical unclonable function with tristate buffers, in: 2008 IEEE International Symposium on Circuits and Systems, IEEE. pp. 3194–3197.

[99] Papakonstantinou, I., Sklavos, N., 2018. Physical unclonable functions (pufs) design technologies: Advantages and trade offs, in: Computer and Network Security Essentials. Springer, pp. 427–442.

[100] Pappu, R., Recht, B., Taylor, J., Gershenfeld, N., 2002. Physical one-way functions. Science 297, 2026–2030.

[101] Petrenko, K., Mashatan, A., Shirazi, F., 2019. Assessing the quantum-resistant cryptographic agility of routing and switching it network infrastructure in a large-size financial organization. Journal of Information Security and Applications 46, 151–163.

[102] Pu, C., Li, Y., 2020. Lightweight authentication protocol for unmanned aerial vehicles using physical unclonable function and chaotic system, in: 2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN, IEEE. pp. 1–6.

[103] Puchinger, S., Müelich, S., Bossert, M., Hiller, M., Sigl, G., 2015. On error correction for physical unclonable functions, in: SCC 2015; 10th International ITG Conference on Systems, Communications and Coding, VDE. pp. 1–6.

[104] Ravishankar, Y., 2015. PUFs – An Extensive Survey. Masters thesis. ECE Department, George Mason University. Fairfax, Virginia, USA.

[105] Roel, M., 2012. Physically unclonable functions: Constructions, properties and applications. Katholieke Universiteit Leuven, Belgium .

[106] Rostami, M., Majzoobi, M., Koushanfar, F., Wallach, D.S., Devadas, S., 2014. Robust and reverse-engineering resilient puf authentication and key-exchange by substring matching. IEEE Transactions on Emerging Topics in Computing 2, 37–49.

[107] Rührmair, U., Busch, H., Katzenbeisser, S., 2010a. Strong pufs: models, constructions, and security proofs, in: Towards hardware-intrinsic security. Springer, pp. 79–96.

[108] Rührmair, U., Holcomb, D.E., 2014. Pufs at a glance, in: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE. pp. 1–6.

[109] Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J., 2010b. Modeling attacks on physical unclonable functions, in: Proceedings of the 17th ACM conference on Computer and com-

munications security, pp. 237–249.

[110] Ruhrmair, U., Solter, J., 2014. Puf modeling attacks: An introduction and overview, in: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE).

[111] Rührmair, U., Sölter, J., Sehnke, F., Xu, X., Mahmoud, A., Stoyanova, V., Dror, G., Schmidhuber, J., Burleson, W., Devadas, S., 2013. Puf modeling attacks on simulated and silicon data. IEEE transactions on information forensics and security 8, 1876–1891.

[112] Rührmair, U., Xu, X., Sölter, J., Mahmoud, A., Majzoobi, M., Koushanfar, F., Burleson, W., 2014. Efficient power and timing side channels for physical unclonable functions, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer. pp. 476–492.

[113] Sahoo, D.P., Mukhopadhyay, D., Chakraborty, R.S., Nguyen, P.H., 2017. A multiplexer-based arbiter puf composition with enhanced reliability and security. IEEE Transactions on Computers 67, 403–417.

[114] Sahoo, D.P., Nguyen, P.H., Mukhopadhyay, D., Chakraborty, R.S., 2015. A case of lightweight puf constructions: Cryptanalysis and machine learning attacks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34, 1334–1343.

[115] Sahoo, D.P., Saha, S., Mukhopadhyay, D., Chakraborty, R.S., Kapoor, H., 2014. Composite puf: A new design paradigm for physically unclonable functions on fpga, in: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE. pp. 50–55.

[116] Schuster, D., Hesselbarth, R., 2014. Evaluation of bistable ring pufs using single layer neural networks, in: International Conference on Trust and Trustworthy Computing, Springer. pp. 101–109.

[117] Shamsoshoara, A., Korenda, A., Afghah, F., Zeadally, S., 2019. A survey on hardware-based security mechanisms for internet of things. arXiv preprint arXiv:1907.12525 .

[118] Shi, J., Lu, Y., Zhang, J., 2019. Approximation attacks on strong pufs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems .

[119] Shimizu, K., Suzuki, D., Kasuya, T., 2012. Glitch puf: extracting information from usually unwanted glitches. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 95, 223–233.

[120] Simons, P., van der Sluis, E., van der Leest, V., 2012. Buskeeper pufs, a promising alternative to d flip-flop pufs, in: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, IEEE. pp. 7–12.

[121] Strieder, E., Frisch, C., Pehl, M., 2021. Machine learning of physical unclonable functions using helper data: Revealing a pitfall in the fuzzy commitment scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems , 1–36.

[122] Su, Y., Holleman, J., Otis, B.P., 2008. A digital 1.6 pj/bit chip identification circuit using process variations. IEEE Journal of Solid-State Circuits 43, 69–77.

[123] Sudhanya, P., Krishnammal, P.M., 2016. Study of different silicon physical unclonable functions, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE. pp. 81–85.

[124] Suh, G.E., Devadas, S., 2007. Physical unclonable functions for device authentication and secret key generation, in: 2007 44th ACM/IEEE Design Automation Conference, IEEE. pp. 9–14.

[125] Suh, G.E., O'Donnell, C.W., Devadas, S., 2007. Aegis: A single-chip secure processor. IEEE Design & Test of Computers 24, 570–580.

[126] Suresh, V., Manimegalai, R., 2018. Spic-sram puf intergrated chip based software licensing model, in: International Symposium on Security in Computing and Communication, Springer. pp. 377–388.

[127] Suzuki, D., Shimizu, K., 2010. The glitch puf: A new delay-puf architecture exploiting glitch shapes, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer. pp. 366–382.

[128] Tebelmann, L., Danger, J.L., Pehl, M., 2020. Self-secured puf: protecting the loop puf by masking, in: International Workshop on Constructive Side-Channel Analysis and Secure Design, Springer. pp. 293–314.

[129] Tebelmann, L., Pehl, M., Immler, V., 2019. Side-channel analysis of the tero puf, in: International Workshop on Constructive Side-Channel Analysis and Secure Design, Springer. pp. 43–60.

[130] Tuyls, P., Schrijen, G.J., Škorić, B., Van Geloven, J., Verhaegh, N., Wolters, R., 2006. Read-proof hardware from protective coatings, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer. pp. 369–383.

[131] Vatajelu, E.I., Di Natale, G., Mispan, M.S., Halak, B., 2019. On the encryption of the challenge in physically unclonable functions, in: 2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS), IEEE. pp. 115–120.

[132] Verbauwhede, I., Maes, R., 2011. Physically unclonable functions: manufacturing variability as an unclonable device identifier, in: Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI, pp. 455–460.

[133] Wang, W., Qiu, C., Yin, Z., Srivastava, G., Gadekallu, T.R., Alsolami, F., Su, C., 2021. Blockchain and puf-based lightweight authentication protocol for wireless medical sensor networks. IEEE Internet of Things Journal .

[134] Wisiol, N., Mühl, C., Pirnay, N., Nguyen, P.H., Margraf, M., Seifert, J.P., van Dijk, M., Rührmair, U., 2020. Splitting the interpose puf: A novel modeling attack strategy. IACR Transactions on Cryptographic Hardware and Embedded Systems , 97–120.

[135] Xiong, W., Schaller, A., Katzenbeisser, S., Szefer, J., 2019. Software protection using dynamic pufs. IEEE Transactions on Information Forensics and Security 15, 2053–2068.

[136] Yanambaka, V.P., Mohanty, S.P., Kougianos, E., Puthal, D., 2019. Pmsec: Physical unclonable function-based robust and lightweight authentication in the internet of medical things. IEEE Transactions on Consumer Electronics 65, 388–397.

[137] Yang, B., Yang, K., Zhang, Z., Qin, Y., Feng, D., 2016. Aep-m: Practical anonymous e-payment for mobile devices using arm trustzone and divisible e-cash, in: International Conference on Information Security, Springer. pp. 130–146.

[138] Yao, Y., Kim, M., Li, J., Markov, I.L., Koushanfar, F., 2013. Clockpuf: Physical unclonable functions based on clock networks, in: 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE. pp. 422–427.

[139] Yu, M.D.M., Devadas, S., 2010. Recombination of physical unclonable functions. 35th Annual GOMACTech Conference , 1–4.

[140] Yu, W., Wen, Y., Köse, S., Chen, J., 2018. Exploiting multi-phase on-chip voltage regulators as strong puf primitives for securing iot. Journal of Electronic Testing 34, 587–598.

[141] Zerrouki, F., Ouchani, S., Bouarfa, H., 2021a. A generation and recovery framework for silicon pufs based cryptographic key, in: International Conference on Model and Data Engineering, Springer. pp. 121–137.

[142] Zerrouki, F., Ouchani, S., Bouarfa, H., 2021b. Towards a foundation of a mutual authentication protocol for a robust and resilient puf-based communication network. Procedia Computer Science 191, 215–222.

[143] Zhang, J., Lin, Y., Lyu, Y., Qu, G., 2015. A puf-fsm binding scheme for fpga ip protection and pay-per-device licensing. IEEE Transactions on Information Forensics and Security 10, 1137–1150.

[144] Zhang, J.L., Qu, G., Lv, Y.Q., Zhou, Q., 2014. A survey on silicon pufs and recent advances in ring oscillator pufs. Journal of computer science and technology 29, 664–678.

[145] Zheng, J.X., Potkonjak, M., 2014. A digital puf-based ip protection architecture for network embedded systems, in: 2014 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), IEEE. pp. 255–256.

[146] Zheng, Y., Liu, W., Gu, C., et al., 2021. Puf-based mutual authentication and key-exchange protocol for peer-to-peer iot applications .