Modeling of Physical Unclonable Functions (PUF)

A Systematic Literature Review

Ferens, Mieszko; Dushku, Edlira; Kosta, Sokol

Publication date:
2025

Document Version
Early version, also known as pre-print

Modeling of Physical Unclonable Functions (PUF): A Systematic Literature Review

Mieszko Ferens, Edlira Dushku, Sokol Kosta

***Abstract*—Hardware fingerprinting technologies are an integral part for security of interconnected devices, for which the Physical Unclonable Function (PUF) has attracted attention in industry and academia for over 20 years. PUFs exploit uncontrollable manufacturing variations to provide hardware-intrinsic secrets that are highly sensitive to physical tampering. However, many questions remain on the applicability of PUFs given the prominent existence of modeling techniques that allow to predict or manipulate these secret fingerprints. In this survey, we analyze the trends and state-of-the-art in PUF modeling from 222 papers obtained from a systematic search and screening process. Our results provide an extensive list of PUF designs and protocols, which we classify based on three main perspectives: application, operational, and defensive. Similarly, we list and classify modeling techniques based on the defined PUF models and learning algorithms. Most of the surveyed papers consider modeling techniques purely as a vulnerability. However, we also include the perspective of modeling as an enabler for lightweight sharing of PUF secrets. Finally, we provide an exhaustive knowledge base and identify gaps and promising directions for future work in the field.**

***Index Terms*—physical unclonable function, modeling, machine learning, authentication, hardware security.**

I. Introduction

In hardware security, a key goal is the development of effective primitives that enable secure supply chains, support a root-of-trust, and support cryptographic operations. Starting from the Integrated Circuit (IC) manufacturing, the manufacturer should not illegally use the design and Intellectual Property (IP) to, e.g., overproduce and sell ICs. It is also important to guarantee that the final device has not been tampered by, e.g., a hardware trojan. Moving to the operation of the final device, it should run on a root-of-trust which ensures a controlled hardware and software platform. Finally, the device should also be capable of cryptographic functions allowing it to authenticate and encrypt its communications.

In this landscape, Physical Unclonable Function (PUF) is a promising hardware fingerprinting technology that serves as a building block for the solutions that tackle these aforementioned problems [1]. PUFs exploit random manufacturing variations to create a hardware-intrinsic fingerprint, similar to how biometrics can be used in humans. The generated fingerprint is essentially a secret that may be used for identification and hardware validation. A standout feature of PUFs is the abundance of lightweight designs targeting low-cost devices [2]. Additionally, PUFs are inherently tamper-evident due to their sensitivity to hardware variations which are inevitably affected by invasive or probing techniques [3].

However, a large portion of PUF research has focused on modeling techniques for PUFs, which has become a significant point of controversy for the technology. Modeling techniques are required to support PUF applications in a few cases (e.g., for sharing a large PUF secret at a low memory cost) [4], but generally they are a liability due to the vulnerabilities they introduce [5]. PUF modeling is based on the prediction or manipulation of the PUF fingerprint, for which many different algorithms and threat models have been proposed [6]–[9]. As a result, in the last decade, the field of PUFs has sparked an arms race between secure PUF designs and algorithms capable of cloning them, which deserves an in-depth analysis.

In this paper, we perform a systematic literature review of PUF modeling starting from 2013 until the end of 2024, where most of the works in this domain have been published. The goal is to provide a comprehensive evolution of modeling techniques and PUF designs until the current state-of-the-art. Additionally, the scale of this study enables us to identify clear trends and gaps that we make apparent to the research community. We aim to aid any researcher or developer in selecting the direction of future work on PUFs by providing a comprehensive overview of PUF modeling. As such, the contributions of this paper are as follows:

- We systematically survey the literature on PUF modeling corresponding to more than 400 papers, and we classify different PUF architectures and protocols in regard to their main security features, implementation, and performance metrics.
- We identify the different types of modeling attacks and defensive measures proposed in the literature.
- We present a thorough analysis of modeling attempts (successful and unsuccessful) against different types of PUFs with and without different defensive measures.
- Based on the literature analysis, we discuss the trends and gaps in current PUF research and conclude with recommendations for future directions in both PUF system designs and modeling techniques.

A. Related surveys and motivation

The systematic literature review of this paper (described in the next subsection) provides a list of surveys related to PUF modeling, and through a broader search, a larger list is obtained. As can be seen in Table I, previous surveys have focused on various aspects of PUF research, including general overview [11], [15], [17], [20], [22], protocols [10], [21], implementation [16], and recently, IoT applications [7],

Mieszko Ferens (corresponding author), Edlira Dushku and Sokol Kosta are with the Department of Electronic Systems, Aalborg University, Denmark (emails: mjfm@es.aau.dk, edu@es.aau.dk, sok@es.aau.dk, address: A. C. Meyers Vænge 15, 2450 København)

TABLE I
RELEVANT SURVEYS ON PUF IN THE LAST DECADE.

Survey	Year	Focus	Security analysis	Modeling attacks
Delvaux et al. [10]	2014	PUF protocols	Yes	Yes
Herder et al. [11]	2014	PUF overview	Yes	Yes
Ruhmair et al. [12]	2014	PUF modeling attacks	Yes	Yes
Zhang et al. [13]	2014	Silicon PUFs and RO PUF	Yes	Yes
Adames et al. [14]	2016	Emerging PUF designs	No	No
Halak et al. [15]	2016	PUF overview	No	No
Alkatheiri et al. [16]	2017	FPGA implementations	No	No
Chang et al. [17]	2017	PUF overview	Yes	Yes
Babaei et al. [18]	2019	IoT applications	Yes	No
Shamsoshoara et al. [19]	2020	IoT applications	Yes	No
Ruhmair et al. [9]	2020	PUF side-channel attacks	Yes	Yes
Yehoshuva et al. [8]	2021	PUF invasive attacks	Yes	Yes
Lokhande et al. [20]	2021	PUF overview	No	No
Mall et al. [21]	2022	PUF protocols	Yes	No
Khalafalla et al. [22]	2022	PUF overview	Yes	Yes
Al-Meer et al. [7]	2023	IoT applications	Yes	Yes
Santikellur et al. [6]	2023	PUF modeling attacks	Yes	Yes
Ferens et al. [23]	2023	PUF modeling attacks	No	Yes
Alhamarneh et al. [24]	2024	IoT applications	Yes	No
This work	**2025**	**PUF modeling**	**Yes**	**Yes**

[24]. While most of these surveys include some remarks or summary of modeling attacks, they do not provide a depth that is proportional to the amount of research currently available on PUF modeling. Some recent surveys focus on PUF modeling attacks [6], [23], however, they do not provide a complete overview, choosing to focus on a limited set of PUF designs. Similarly, there are some surveys on specialized PUF attacks [8], [9], although they lack some recent developments in the field. Moreover, the beneficial perspective of modeling techniques is missing. Thus, there is a noticeable gap in the literature for a survey that provides a complete and updated overview.

B. Methodology

To systematically search the literature, we focus on the Web of Science and Scopus databases. These databases index all the major scientific databases where PUF literature is published, including but not limited to IEEE, ACM, ScienceDirect, Springer, and Wiley. To identify relevant works to PUF modeling, we defined a set of keywords which encompass the PUF topic while including mentions of modeling or attacks:

((“*physical unclonable function*” OR “*PUF*”) AND (“*modeling*” OR “*modelling*” OR “*attack*”))

We also limited the search criteria to only include papers from 2013 until the end of 2024. Initially, we attempted to search the databases on a topic basis, where the set of keywords can appear in the title, abstract or keywords of the documents. This search provided us with 2055 and 1950 results in Web of Science and Scopus, respectively, showing that the query was too broad. Thus, we limited the appearance of the keyword set to only the document title for our second query, which provided 197 and 216 results, respectively.

After combining these results into 413 papers, we **removed all duplicates** without excluding similar or extended papers, leaving 236. Then, we **screened all remaining papers** for relevancy in our study. Our criteria for inclusion in this paper were the following:

1) Proposing a novel PUF design, protocol or defensive mechanism that may tackle modeling attacks.
2) Evaluating some modeling technique on some PUF design, protocol or defensive mechanism.

While rare, not all papers approach the PUF modeling from a security perspective. These papers are included in our study as, fundamentally, they fit the topic. Additionally, some papers focus on other aspects, e.g., PUF performance, without considering modeling. These papers are also included in our study, as long as their contents somehow align with the inclusion criteria, even if it was not the intention of the authors. On the other hand, we exclude papers that:

1) Are not about PUFs.
2) Propose PUF applications, design extensions, or protocols that do not affect the modeling in any manner.
3) Are the original and more limited works of other extended papers included in this study.
4) Are not in English language.
5) Are not accessible to us[1].

The screening process left us with 176 papers, to which we added 46 papers from a **snowballing process** performed in parallel, for a total of 222 papers. This total includes 6 of the survey papers[2] included in Table I. Our methodology is illustrated in Fig. 1.

C. Organization

The rest of this paper is organized as follows. Section II presents the necessary background on PUFs and modeling

[1]We have access to all major publishers including IEEE, ACM, ScienceDirect, Springer and Wiley, as well as many others. In some rare cases, we are not able to access papers published in less known venues.

[2]The rest of the surveys from Table I were found through a general search on the topic of PUFs without following the described methodology.

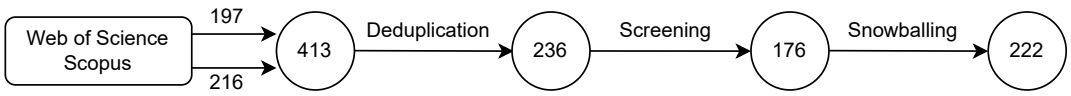

Fig. 1. Systematic literature review methodology with deduplication, screening, and snowballing steps. The numbers indicate the number of papers after each step.

techniques. In Section III, the applications of PUFs are introduced, including the problems and solutions that arise from modeling techniques. Section IV presents the classification of PUF designs and shows the trends that have developed. Following, Section V complements the previous section on PUF designs by considering protocols that extend their functionality. Then, in Section VI the types of currently available modeling techniques and their details are provided. Section VII discusses the challenges that remain unsolved and provides a discussion with remarks towards the future goals of research in the field of PUFs. Finally, Section VIII concludes the paper.

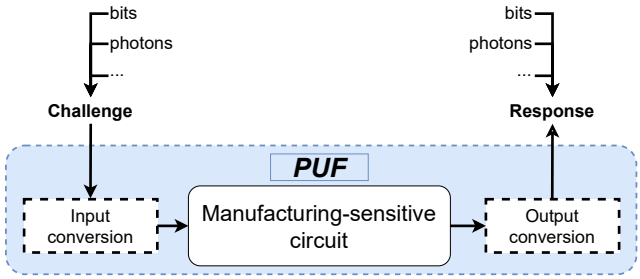

Fig. 2. Generic PUF module.

II. Preliminaries

This section presents the necessary background on PUFs to understand the proposed designs and protocols up to date. It also serves to better understand the modeling of PUFs discussed later in the paper.

A. Definition and basic operation

PUF is based on the idea that it is not possible to manufacture two exact replicas of the same hardware. Random manufacturing variations occur due to many factors such as physical impurities, temperature or pressure difference, among many others. Given that a manufacturer is not able to control these variations with current manufacturing technologies, it is widely accepted that PUFs are physically unclonable. Additionally, since a PUF implementation depends on its hardware, any physical tampering or probing inevitably changes the PUF, proving resilience against such attacks. Note that, we discuss some works that challenge this assumption in Section VI.

At a high level, a PUF module is expected to function by providing responses to challenges, as shown in Fig. 2. A challenge can be any type of stimuli applied to the PUF, ranging from a binary input to other physical stimuli, such as photons, depending on the PUF type. Similarly, the response of a PUF can also take any of the formats the challenge does. For many applications, it is convenient to provide both challenges and responses in binary form, thus, if required, some conversion circuitry can be included in the PUF system (optional input and output conversion in Fig. 2).

Ultimately, a commonly used term is the Challenge-Response Pair (CRP), which describes a single input challenge to the PUF, and its corresponding response. Based on their CRPs, PUF are evaluated according to a set of properties. While different works consider different sets of properties, we choose to present the fundamental properties that are always considered in PUF literature in one way or another, and that are essential to all PUF applications.

- **Reliability**: The ability of a PUF to provide the same responses to the same challenges, i.e., the consistency of the CRPs. Since the CRPs are used to identify the device that possesses a PUF, or as inputs to other cryptographic functions, they are expected to be consistent. Given the correct response r to a challenge, the reliability of a PUF is calculated by obtaining t samples of the same response r_i' in different environmental conditions. These environmental conditions can be a difference in temperature, supply voltage, time (due to aging), among others. Assuming that the responses are converted to bits, and that each response is of m bits:

$$Reliability = (1 - \frac{1}{t}\sum_{i=1}^{t}\frac{HD(r, r_i')}{m}) * 100\% \quad (1)$$

with $HD(\cdot,\cdot)$ being the Hamming Distance of two binary strings. The ideal value for this metric is 100%.

- **Uniformity**: Assuming responses in binary form, the uniformity is the ratio of 1s to 0s in the responses of a PUF. It can be calculated from a set of responses r_i obtained from l different challenges:

$$Uniformity = (\frac{1}{l}\sum_{i=1}^{l} r_i) * 100\% \quad (2)$$

To ensure that the responses are hard to predict, uniformity has an ideal value of 50%.

- **Uniqueness**: The ability of a PUF to identify two devices, each with their own instance of the PUF, when responding to the same challenge. For d devices, each with their own response r_c of m bits, it can be calculated as the average Hamming Distance of the responses from the d PUFs, averaged over t different challenges:

$$Uniqueness = \frac{1}{t}\sum_{c=1}^{t}(\frac{2}{d*(d+1)}\sum_{i=1}^{d-1}\sum_{j=i+1}^{d}\frac{HD(r_c^i, r_c^j)}{m}) * 100\% \quad (3)$$

The ideal value for this metric is 50%.

B. Basics of modeling PUFs

A well-researched problem of PUFs is their clonability [6], [12], [23]. In essence, while a PUF is designed to provide unpredictable outputs to specific inputs (i.e., CRPs), it is possible to predict them through different methods. This is the fundamental idea of modeling PUFs and is the basis of modeling attacks. Note that PUF modeling can also be done in a trusted manner to, e.g., create a software copy of the PUF for a server that can then authenticate the PUF. The main difference between modeling attacks and the latter is that the restrictions on an attacker are usually considered to be stricter due to limited access to information on the PUF itself [25]. For example, an attacker may collect CRPs passively by eavesdropping on a PUF communication channel or actively by directly communicating with a PUF [26]. Meanwhile, a manufacturer could have access to measurements that are internal to the PUF circuit [27].

Regardless of the case, there are two main components for PUF modeling: (i) the model of the PUF and (ii) the learning algorithm. The model is a general representation of a specific PUF design that possesses some parameters that, when randomized, can perfectly define the behavior of a physical instance of the respective PUF design. On the other hand, the learning algorithm is the component responsible for approximating these parameters for a specific PUF instance. This process obviously requires data on the PUF instance (CRPs or side-channel information), e.g., fitting a model with a Machine Learning (ML) algorithm. With a good fit, the resulting fitted model can approximate the physical PUF instance so closely that other CRPs not present in the training data can also be predicted.

The success of PUF modeling depends on the accuracy of the model, the use of an adequate learning algorithm, and the availability of sufficient training data. For example, a popular model for the so-called Arbiter PUF [28] and XOR PUF [29] designs is the additive delay model [30], [31] (these PUFs and models are explained in more detail in Sections IV-B1 and VI-A1). However, while this model is linear with the former design, it is nonlinear with the latter. This means that one of the most popular ML modeling techniques, namely Logistic Regression (LR) [32], becomes ineffective in the nonlinear case despite it being the best modeling approach to date against the Arbiter PUF [33]. Instead, another popular approach is to use the Multi-Layer Perceptron (MLP) algorithm which adapts better to nonlinear models [29], [34], [35], but requires more data [36].

III. Key Consequences of PUF Modeling on PUF Applications

Looking back at the fundamental operation of PUFs from Section II-A, the core concept of a PUF is to provide a unique output to a specific input in the form of a CRP. A CRP functions as a unique token that only the device in possession of the corresponding PUF should be able to generate. This observation shows that a PUF is essentially a tamper-proof storage, which is widely applicable for different security applications. Fig. 3 shows a taxonomy of PUF applications with examples for each. The first division in PUF applications separates the functional environment in which a PUF (more specifically its CRPs) operates: (i) internal or "on-device", and (ii) external or "off-device". Following, both categories and their specific subcategories (applications) are described.

A. On-device applications

In the case of on-device applications the focus is on physical security and anti-tampering properties, with CRPs used only internally in the device that holds the PUF. Recall that a PUF acts as a secure storage on the device. However, the contents of the storage are randomized through the manufacturing process and cannot be controlled. Nevertheless, randomized tamper-proof memory is a valuable resource for the following applications.

- **Cryptographic key storage**: An obvious use case for tamper-proof memory is the storage of cryptographic keys [3]. The key could be used as a pre-shared secret (assuming a copy is distributed to other communication nodes) or to initialize a public-key cryptographic scheme. The main advantage of using PUF in this use case, is that adversaries cannot easily obtain this key from the device by invasive methods. This is especially relevant when dealing with memory readout attacks on powered-off devices, as the key is not stored in persistent memory and the PUF needs to generate it instead [25], [37].
- **Tamper-proof memory encryption**: Encrypting the data in a device's memory is an effective method to minimize the risk of adversaries obtaining some information, such as Intellectual Property (IP) [38]. To do so, a device would normally need an encryption key that should not be physically vulnerable on the same device. This proves challenging in absence of secure storage, but PUFs can solve the issue by generating the key on demand.
- **Detection of physical tampering**: Taking a step back and generalizing, a PUF can simply be used as a physical detection layer on an IC that measures its integrity [25]. An example can be found in [27], where a PUF measuring capacitance forms a cage that envelopes a circuit. Such a cage makes it exceedingly difficult to physically breach the circuit without affecting the PUF, which would then output different CRPs, allowing for tamper detection.
- **True Random Number Generation**: Granted a sufficiently large number of CRPs, they can be used as random numbers [39], [40]. The main advantage of PUFs as Random Number Generators (RNGs) is that they implement this function as a lightweight True RNG (TRNG) instead of a Pseudo RNG (PRNG).

Overall, on-device PUF applications can be a building block toward secure supply chain and early-stage integrity validation in a root-of-trust mechanism. Additionally, over-building and the sale of ICs on the black market can be prevented through identification of the generated unique PUFs [41]. Moreover, cryptographic functions can be supported with secure storage modules with guarantees on the randomness of the keys. Finally, PUFs have great flexibility due to the overwhelmingly large number of use cases for RNGs.

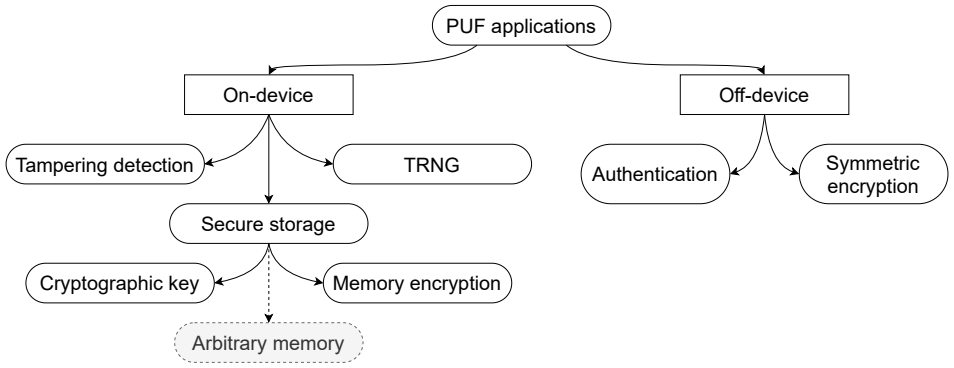

Fig. 3. PUF application taxonomy.

The assumption for the uncontrollable randomness of the CRPs of on-device PUFs is important for the previous applications. However, recent work has shown that this assumption may not always hold, when the CRPs of a Static Random Access Memory (SRAM) PUF were manipulated through a process called BTI aging [42]. Such a vulnerability would allow a manufacturer to maliciously control the contents of a PUF, although the authors of [42] proposed an anti-invasive countermeasure which fixed the vulnerability. As will be seen later, anti-invasive techniques form a small portion of the overall modeling techniques in PUF literature, showing the lack of knowledge in this regard. A positive note is that when not performed maliciously, the technique presented by [42] can be used to implement nonrandom secure and tamper-proof memory, enabling other potential applications that have not been explored to date.

B. Off-device applications

An arguably much more challenging scenario is when CRPs are exchanged and leave the tamper-proof premises of a device. When CRPs leave a device, one cannot guarantee that they do not fall into the hands of an adversary. Generally, this would not be a problem if CRPs were fully independent; however, many off-device applications only make sense in the presence of a large number of unique CRPs. As we will explain in Section IV-A, a Strong PUF can provide an exponential number of CRPs for its size at the cost of a correlation between them. Combine this with a malicious adversary and you get the majority of modeling attacks (see Section VI). Even so, many works have attempted to push through this limitation to provide the following applications.

- **Authentication**: An obvious use for the PUF's device-unique secret is to identify devices by checking the values of known CRPs [39], [43].
- **Pre-shared key symmetric encryption**: Extending the use of secure storage for cryptographic keys in on-device applications, a device could use a pre-shared symmetric key stored in a PUF for encrypted communications [21].

The main challenge of off-device applications is their susceptibility to modeling attacks due to exposed CRPs. The unified response from the research community to this problem has been to develop obfuscation techniques that prevent CRPs from being sent in the clear (despite the potential lack of encrypted communications). Obfuscation applies especially to authentication applications, since authentication tends to precede encryption. However, for symmetric encryption schemes, the CRPs are not easily exploitable since they are not sent in the clear by default.

The remaining challenge is then to deploy a single unique PUF in a shared manner, since any communication requires both endpoints to share the CRPs provided by the PUF. The secure deployment of a PUF clone remains an open challenge, with previous work assuming that it is possible through the manufacturing process. One major concern is the memory complexity of storing millions or even billions of CRPs without a physical PUF. Ironically, a good solution to this problem comes from modeling techniques, as software PUF clones can be distributed instead [4]. However, for additional security the controlled physical manipulation of PUFs (as demonstrated by [42] on SRAM PUF) could be employed as well.

IV. PUF Classification

The idea to use unique and intrinsic characteristics of materials has existed for many decades [44]. However, the term "Physical Unclonable Function" was not introduced until the early 2000s, when the use of intrinsic delays of wires and electronic components to identify ICs was proposed [30]. Since then, a plethora of ideas to uniquely identify electronic devices (typically involving ICs) have emerged [7], accompanied by a variety of applications.

Due to the large number of PUF designs that have been proposed in the literature, it becomes necessary to classify them into groups. The classification process is not straight forward, as no standardized method exists. However, based on the large volume of literature surveyed in this paper, three dimensions to the classification have been identified: (i) **type** of PUFs based on their CRPs (application perspective), (ii) **category** of PUFs based on their operational principles (operational perspective), and (iii) **countermeasures** (if any) implemented into PUFs designs (defensive perspective). Fig. 4 shows a taxonomy for the classification with each perspective having subcategories that are explained in the following subsections.

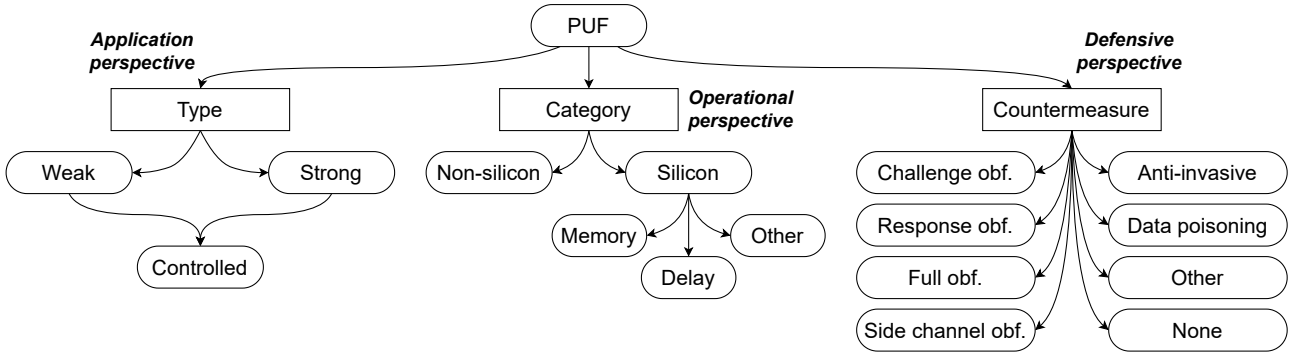

Fig. 4. PUF classification taxonomy.

Additionally, Table II provides a list of PUF designs organized primarily based on their operational principle, with the other perspectives included as standard columns. Due to the large number of implementations (281 in total), Table II is limited to a single page, prioritizing designs that (i) appear in multiple studies, (ii) include complete performance data, and lastly (iii) are among the most recent. Note that this means that PUFs that appear multiple times in the literature have only one entry in Table II to save space. The full list is available in the appendices of this survey (see Appendix A).

A. Application perspective

Depending on the number of unique CRPs that can be extracted from a PUF, they are classified into Weak or Strong PUFs. Moreover, originating from these two classes, Controlled PUF can be defined.

- **Weak PUFs**: Weak PUFs are those that possess a number of unique CRPs linearly proportional to the number of basic components of the PUF. The advantage of a Weak PUF is that its CRPs have low correlation and may even be considered independent [102]. This makes it challenging to model the unknown CRPs based on known ones. However, note that a Weak PUF does not guarantee uncorrelated CRPs [103], [104]. Moreover, due to the limited number of CRPs, an attacker could clone the PUF in linear time by exhaustively recording all CRPs. Thus, the CRPs cannot be exchanged in the clear through an untrusted channel and are normally reserved for on-chip applications such as key generation [11], [13].
- **Strong PUFs**: Strong PUFs are those that possess a number of unique CRPs exponentially proportional to the number of basic components of the PUF. This characteristic prevents PUF cloning in linear time by exhaustively recording all CRPs, making Strong PUFs practical for off-chip applications where CRPs are sent in the clear over an untrusted channel, e.g., authentication. However, since different unique CRPs are generated from the same components of the PUF, they are correlated and exhibit limited entropy [2], [105]. The consequence is that Strong PUFs can become vulnerable to modeling attacks in which unknown CRPs are predicted based on previously known CRPs [102].
- **Controlled PUFs**: Controlled PUFs are those that use either a Weak or (typically) Strong PUF as their core. In addition to this, they implement control logic, e.g., a Random Number Generator (RNG) [106], Linear Feedback Shift Register (LFSR) [107], hash function [108], or fuzzy extractors [109]. Overall, a Controlled PUF enables complex modes of operation that allow for increased security (by implementing countermeasures) and reliable applications. It is worth noting that the definition of Controlled PUF is not consistent throughout the literature. For instance, some PUFs with additional modules might be called Strong PUFs instead [70], [72]. In this paper, we choose to consider as a Controlled PUF any PUF that, besides a single core PUF, incorporates additional elements that increase the difficulty of modeling. However, additional modules that do not affect the modeling do not classify the PUF as a Controlled PUF (e.g., only for increasing reliability [38], [110], [111] or uniqueness [112], [113]).

Looking at the types of PUFs that are addressed in the literature, the largest volume goes to Controlled PUFs with 65.1% of designs. Meanwhile, only 26% and 8.9% are purely Strong and Weak PUFs, respectively. As a security primitive, PUFs are expected to be unclonable and tamper-proof. However, repeated vulnerabilities have necessitated additional security mechanisms, leading mainly to the development of Controlled PUFs. Additionally, the extensive number of CRPs in Strong PUFs provides greater versatility, justifying their popularity in the literature over Weak PUFs. This popularity is reinforced by Weak PUFs already showing good resilience to attacks due to their low correlation CRPs. Since only a handful of recent studies successfully compromise them [42], [114], [115], researchers typically focus on the more challenging security problem of Strong PUFs.

B. Operational perspective

Based on their manufacturing, PUFs can be classified as:

- **Non-silicon PUFs**: Non-silicon PUFs are those that are based on the manufacturing variations of non-electrical components. The most common example are optical structures that modify light patterns that pass through them [99], [116], [117], although microscopic randomly

TABLE II

SHORT OVERVIEW OF PUF IMPLEMENTATIONS (SEE APPENDIX A FOR FULL VERSION). THE DISPLAYED AND TOTAL NUMBER OF ENTRIES ARE SHOWN AS (X / Y). PERFORMANCE VALUES ARE SELECTED BASED ON A REPRESENTATIVE WORST CASE SCENARIO. A '∼' SYMBOL INDICATES THAT THE VALUES ARE TAKEN FROM A VISUAL GRAPH AND ARE NOT EXACT. BLANK SPACES ARE INFORMATION THAT WAS NOT OBTAINABLE FROM THE REFERENCE. A '-' SYMBOL INDICATES NO COUNTERMEASURE.

Category	PUF	Ref ID	Type	Platform	Reliability	Uniformity	Uniqueness	Countermeasure
Delay (32 / 186)	Arbiter PUF	[28]	Strong	Xilinx Spartan-6	98.42%		51.19%	-
	BST-RPUF	[45]	Controlled	Xilinx Artix-7	∼100%	46.78%	48.64%	Challenge obf.
	CBDC-PUF	[46]	Strong	Xilinx Artix-7	100%	49.6%	49.8%	-
	CO-PUF	[47]	Controlled	Xilinx Artix-7	95.37%	∼50%	50.01%	Response obf.
	CT PUF	[48]	Controlled	Xilinx Zynq-7000	∼92%	∼50%	∼50%	Data poisoning
	DAPUF	[49]	Controlled	Xilinx Virtex-5	∼90.31%	∼45.74%	∼46.37%	Response obf.
	DCH PUF	[50]	Controlled	Xilinx Artix-7	98.68%		41%	Response obf.
	Dual-mode PUF	[51]	Strong	Xilinx Artix-7	∼86%	44.65%		Data poisoning
	DyAdv PUF	[52]	Controlled	Xilinx Artix-7	97.97%	49.67%	50.13%	Response obf.
	FF-PUF	[53]	Controlled	Xilinx Artix-7				Challenge obf.
	FLAM-PUF	[54]	Controlled	Simulated	95.59%	49.73%	49.81%	Challenge obf.
	FOM CDS-PUF	[55]	Controlled	Xilinx Artix-7	92.09%	52.54%	50.50%	Challenge obf.
	IPUF	[56]	Controlled	Xilinx Artix-7	97.9%	25-50%		Challenge obf.
	Ising-PUF	[57]	Controlled	Simulated	97.74%		50.1%	Challenge obf.
	LBIST-PUF	[58]	Controlled	Xilinx Artix-7	89.6%	∼60%	∼70%	Challenge obf.
	LEE PUF	[59]	Controlled	Xilinx Spartan-6	99.74%	49%	58.73%	Response obf.
	LP-PUF	[60]	Controlled	Simulated	61-96%		∼99%	Full obf.
	LSPUF	[61]	Controlled	Xilinx Artix-7	97.08%	49.50%		Response obf.
	MPUF	[62]	Controlled	Simulated	98.67%	49.80%	50.01%	Challenge obf.
	NoPUF	[63]	Controlled	Simulated	91.98%		49.65%	Challenge obf.
	OI-PUF	[64]	Controlled	Xilinx Zynq-7000	99.2%	48.9%	30%	Challenge obf.
	P-2APUF	[65]	Controlled	Xilinx Virtex-7	∼53%	∼50%		Challenge obf.
	PFO PUF	[66]	Controlled	Xilinx Spartan-6	96.91%	49.00%	56.02%	Full obf.
	RO PUF	[67]	Weak	180nm CMOS	99.95%		∼50%	-
	RPUF	[68]	Controlled	Xilinx Zynq-7000	94.80%	48.9%	52.2%	Challenge obf.
	SCD-PUF	[69]	Controlled	Xilinx Artix-7	97.79%	49.56%	49.95%	Full obf.
	SOI PUF	[70]	Controlled	Xilinx Artix-7	98.6%	48.2%	29.1%	Response obf.
	SRPUF	[71]	Controlled	Xilinx Artix-7	91.87%	50.14%	50.03%	Challenge obf.
	SW PUF	[72]	Controlled	Xilinx Artix-7	99.97%	49.74%	49.89%	Full obf.
	TP PUF	[73]	Controlled	Xilinx Artix-7	96.59%		57.97%	Response obf.
	XMPUF	[74]	Controlled	Xilinx Artix-7		37.03%	40.6%	Challenge obf.
	XOR PUF	[75]	Controlled	Simulated		49.23%	53.21%	Response obf.
	...	...	...	...	...	...	...	...
Memory (8 / 27)	2SPUF	[76]	Weak	Simulated	96.19%	49.34%	48.1%	Challenge obf.
	BR PUF	[77]	Strong	Xilinx Spartan-6	99%	47%	49%	-
	Feedback SPN PUF	[78]	Controlled	130nm CMOS	99.27%		∼50%	Challenge obf.
	MRAM PUF	[79]	Weak	Fabricated	97.25%		47%	-
	SRAM PUF	[42]	Weak	Simulated	99.14%	50.15%	47.71%	Anti-invasive
	Suresh PUF	[80]	Controlled	14nm CMOS	99.74%	∼50%		Full obf.
	TBR PUF	[77]	Strong	Xilinx Spartan-6	97%	54%	50%	-
	eFlash PUF	[81]	Strong	55nm CMOS	5%	50.3%		-
	...	...	...	...	...	...	...	...
Other (17 / 63)	3D NbOx PUF	[82]	Weak	NbOx	97.59%		49.97%	-
	AM-PUF	[83]	Controlled	Xilinx Zynq-7010	98.1%	51.19%	53.51%	Full obf.
	Arbiter MRAM PUF	[84]	Strong	Simulated	99.55%		49.76%	-
	CIS PUF	[85]	Weak	Simulated	97.23%	47.62%	47.91%	-
	Current Mirror PUF	[86]	Strong	Simulated	98%	47%	49%	-
	ESP-PUF	[87]	Controlled	Altera Cyclone 2	∼100%	50.10%	49.96%	Full obf.
	HP mem-PUF	[88]	Weak	PrPyrl	100%		48.1%	-
	Hybrid PUF	[89]	Controlled	Xilinx Spartan-6	99.7%	49.78%	49.38%	Full obf.
	Lattice PUF	[90]	Controlled	Xilinx Spartan-6	98.74%	49.98%	50.00%	Other
	MR-PUF	[91]	Weak	Simulated		51.10%	49.40%	-
	Neuron-PUF	[92]	Controlled	65nm CMOS	100%	47.49%	48.42%	-
	SC PUF	[93]	Weak	180nm CMOS	∼100%	46.72%	50.38%	-
	SCA PUF	[94]	Strong	130nm CMOS	97.4%	52.8%	49.9%	-
	SPN PUF	[95]	Controlled	130nm CMOS	∼100%		∼50%	Full obf.
	STT-MRAM PUF	[96]	Weak	Simulated	∼97%	49.09%	49.96%	-
	Sponge PUF	[97]	Controlled	Simulated	∼50%	50.03%	50.00%	Full obf.
	VTC PUF	[98]	Strong	Simulated	97.9%	50.1%	49.8%	-
	...	...	...	...	...	...	...	...
Non-silicon (3 / 5)	NOS PUF	[99]	Strong	Simulated			∼50%	-
	QuPUF	[100]	Strong	IBM Quantum	13.82%		55.13%	-
	SBS-CAN PUF	[101]	Strong	Fabricated	96%	49.20%	49.60%	-
	...	...	...	...	...	...	...	...

generated structures in certain materials can also be used as a fingerprint [101]. Finally, an emerging idea is to fingerprint the components of quantum computers [100], [118].

- **Silicon PUFs**: Silicon PUFs are those that are based on the manufacturing variations in the fabrication of electrical components, mainly in ICs [81]. Typically, two subcategories are defined for Silicon PUFs: (i) memory-based PUFs and (ii) delay-based PUFs. Memory-based PUFs exploit the stabilization bias of memory components (memory cells) towards either 0 or 1 when left in an unstable state. Meanwhile, delay-based PUFs utilize the variation in latency of the propagation of electrical signals through electrical components and wires. Other subcategories could be defined, but previous work has not naturally converged towards further definitions. Thus, we consider a third general subcategory for "other" designs, as shown in Fig 4.

Comparing silicon and non-silicon PUFs shows a big difference in the ease of implementation. While a few proposals that could support optical systems have appeared [99], [116], they do not come close to the number of silicon PUFs which account for 98.1% of designs (observed in this survey). Even with the new emerging ideas to use PUFs as "quantum" primitives [100], [117], [118], the reliability and particularly specific use cases of non-silicon PUFs are a major challenge that holds them back. Meanwhile, the potential of a few practical silicon PUFs (mainly Arbiter PUF and SRAM PUF) has largely driven most PUF research over the past decade.

1) The Arbiter PUF: First, the Arbiter PUF [30] is the first and most promising Strong delay-based PUF to appear in the literature. It uses a chain of switch elements to generate an exponential number of CRPs at a low hardware cost. However, two problems have consistently persisted for Arbiter PUFs: the difficulty of implementing balanced delay paths in hardware, especially Field-Programmable Gate Arrays (FPGA) [49], and the correlation between CRPs viewed as a vulnerability. The implementation challenge is out of scope for this paper, but note that many works have successfully manufactured PUFs in a variety of FPGA platforms [55], [119]–[124] and even some Application-Specific Integrated Circuits (ASIC) [80], [125], [126].

More importantly, the correlation between CRPs is a problem that has been shown to be more challenging that could initially appear. This challenge alone has motivated an enormous number of Controlled PUF designs stemming from the Arbiter PUF. Firstly, designs such as FF-PUF [127], XOR PUF [128], [129], IPUF [56], [128], [130], LSPUF [128], [131], MPUF [62], [128], [132], Dual-mode PUF [51], and CT-PUF [48] (just to name a few), all extend the Arbiter PUF and have one thing in common: they increase the complexity of the model through more complicated wiring and additional basic components. This design strategy attempted to retain the main idea of the Arbiter PUF, that is, lightweight hardware identity with exponential CRPs. However, due to the escalating effectiveness of modeling attacks, most of the previous designs were shown to be vulnerable as well. Thus, a different approach which incorporates other modules was also strongly argued in many papers leading to designs such as CRC-PUF [133], DFM-APUF [134], FLAM-PUF [54], RPUF [68], PUF-FSM [135], SRPUF [71], DCH PUF [50], etc. This approach (similarly to protocols in Section V) was more effective against modeling attacks, but certain vulnerabilities were still exposed (see Section VI).

2) The SRAM PUF: Aside from delay-based PUFs, memory-based are an important class. The earliest PUF of this type is the SRAM PUF [42], [136], which exploits the stabilization bias of SRAM memory cells towards a certain value. Based on this principle, the use of other memory technologies can also be exploited, e.g., with DRAM [137], [138], MRAM [79], [139], or flash memory [81]. The use of memory cells provides an important advantage to security as the bits generated by the memory cells bias is uncorrelated between them. However, the limited number of cells leads to a lower number of CRPs. To combat this limitation, other designs such as the BR PUF [140]–[142] and TBR PUF [140], [141] take inspiration from the Arbiter PUF and create a chain of memory cells, but this can obviously lead to some security concerns due to correlation between cells.

That said, overall, the amount of research into memory-based PUFs is lower than for delay-based. Fundamentally, this is because the best designs are the simple ones such as SRAM PUF that provide clear use cases for on-device applications with few security concerns regarding physical aspects [42].

3) Other PUF designs: As can be expected, there are efforts to improve PUFs by providing fundamentally new designs. Aside from delays and memory stabilization bias, capacitance has been exploited [27], [111], [143]. Similarly, voltage, current, or resistance (basically Ohm's law) through electronic components can be applied to PUFs [110], [144].

Additionally, emerging technologies such as memristors can be leveraged to create PUFs [3], [145]. Another interesting example is the use of CMOS image sensors [146]. Finally, non-silicon materials have also been proposed to fabricate specialized PUFs [101]. Interestingly, many new designs share a striking similarity to the structure of an Arbiter PUF [147], [148].

C. Defensive perspective

Given the existence of different attacks against PUF, many papers have proposed protocols, schemes, and techniques to mitigate them. The large-scale literature review conducted in this paper has allowed for the identification of common techniques. Thus, in this subsection, we provide a classification of the types of countermeasures that can be found in the literature, with explanations for each class. There are four fundamental countermeasures to modeling techniques: challenge obfuscation, response obfuscation, full obfuscation, and data poisoning. Additionally, special countermeasures to prevent side-channel and invasive attacks have also been proposed. Finally, certain countermeasures that cannot be grouped with the rest have also been proposed.

- **Challenge obfuscation**: Most modeling attacks on PUFs rely on the use of clear CRPs for training of an ML model (see Section VI). For this reason, a popular approach in PUF protocols as well as many Controlled

PUFs is to hide the challenge fully or partially from the attacker [149]–[153]. This obfuscation is typically done with some common control logic on both the verifier and the prover, so that they can generate matching challenges without the need to publicly exchange them through an insecure communication channel [154]. For example, a common PRNG can be used on both endpoints to generate the same challenges [155]. Note that there can be vulnerabilities associated with the added control logic.

- **Response obfuscation**: Similar to challenge obfuscation, response obfuscation is based on the idea that if an attacker does not possess clear CRPs, they cannot perform a modeling attack. In this case, instead of hiding the challenge information from the attacker, the response information is hidden [156]–[159]. This can be done by, e.g., hashing the response value with some other public information through some common hash function [160], or sending decoy responses so that the attacker does not know which corresponds to which challenge [161].
- **Full obfuscation**: As the name implies, full obfuscation combines both challenge and response obfuscation. By hiding both challenge and response information from the attacker, creating a sufficiently accurate CRP training set becomes even more challenging [80], [162]–[165]. It is important to mention that the more complex the PUF additional control logic becomes, the larger the possibility of other vulnerabilities. Additionally, the hardware and the protocol overhead of the design increases, which is an undesirable side effect.
- **Data poisoning**: Different from the previous approaches, data poisoning is based on the idea that the verifier and prover can control the CRPs that are used for the PUF application [26], [166]. The ability to control CRPs can be exploited to perform, e.g., adversarial counterattacks against the modeling attacks. The key characteristic of these countermeasures is that they do not limit the attacker's ability to obtain CRPs in any way, but regardless, the modeling difficulty increases.
- **Side-channel obfuscation**: Not all modeling attacks, and attacks in general, are designed with CRPs in mind. A common idea to extract information from a PUF is to use side-channel information such as power traces [167], [168], Electromagnetic (EM) radiation [169], or reliability [119]. Side-channel information can be a grave threat to PUF security; thus, countermeasures that hide these traces have been proposed [170], [171].
- **Anti-invasive**: While widely considered to be physically unclonable, in rare cases PUFs have been successfully exploited through physical tampering [42], [115]. In response to this, specific anti-invasive countermeasures have been developed. Specifically, tampering is detected through capacitive cages [27], [111], [143], hardware trojans are prevented through an integrated state machine [172], and memory bit modification through Focused Ion Beam (FIB) aging is handled with specialized pre-charge [42] or transistor test circuits [115].
- **Other**: There exist a handful of countermeasures which do not fit into any of the previous groups [90], [173]. We specify these countermeasures as "Other" in Tables II, III, and IV.
- **None**: Obviously, many designs do not incorporate any specific countermeasure against modeling. We specify these cases as "-" in Tables II, III, and IV.

The need for resilience against modeling attacks has clearly motivated the implementation of countermeasures, since looking at the papers included in this survey only 28.5% do not include any countermeasures. Meanwhile, obfuscation techniques are extremely popular, with 29.4%, 23.4%, and 10.6% of PUFs implementing challenge, response, and full obfuscation, respectively. The lower proportion of full obfuscation techniques can be attached to the larger and undesirable hardware cost. In contrast, data poisoning techniques providing extremely low cost are very scarce with a 2.6%, arguably do the fact that they do not fundamentally prevent modeling attacks and only make them harder. Additionally, although side-channel and invasive techniques are a serious threat, their countermeasures are also rarely seen in literature with 2.1% and 3%, respectively. Finally, other not categorized countermeasures make up the remaining 0.4% of papers.

It is obvious that a popular trend has been to tackle modeling attacks with obfuscation [39]. However, the appearance of advanced attacks that break some of these methods can argue against their use. It is worth noting that some obfuscation techniques have not been broken or have had some initial vulnerability fixed by the same authors that proved them vulnerable. Thus, further research into this direction is not advisable. Instead, the lighter data poisoning solutions that may be more practical in low-cost applications deserve more attention. Moreover, side-channel and invasive vulnerabilities are also a mostly unexplored avenue.

V. Protocols

The use of PUFs requires protocols depending on the application. Typically, these protocols are tied to functionality rather than security. However, in the topic of modeling of PUFs we can find papers that propose a PUF protocol as a countermeasure to modeling. These cases are different from a Controlled PUF because it is the way in which the PUFs are used that changes, not its operation. As such, protocols deserve their own section, but we will consider a similar classification to PUFs. This can be seen in Table III.

The first observation we can make is that all protocols that we found use an Arbiter PUF or derivative (except for two where it is unclear) for testing. Note that these protocols tend to be agnostic to the PUF (although they might require a Strong PUF) so this is just a choice by the authors. The second important observation is that when classifying the type of countermeasure they provide, almost all give some form of obfuscation to the CRPs. These observations make sense based on our previous statistics showing that Arbiter-based PUFs and obfuscation techniques have been by far the most researched. The only exceptions to this are an exchangeless key protocol [185] providing Weak PUF functionality, and a data poisoning technique [26] based on selective use of CRPs. Based on these data, our conclusions for the state of research

TABLE III
FULL OVERVIEW OF PUF PROTOCOLS. PERFORMANCE VALUES ARE SELECTED BASED ON A REPRESENTATIVE WORST CASE SCENARIO. A '∼' SYMBOL INDICATES THAT THE VALUES ARE TAKEN FROM A VISUAL GRAPH AND ARE NOT EXACT. BLANK SPACES ARE INFORMATION THAT WAS NOT OBTAINABLE FROM THE REFERENCE. A '-' SYMBOL INDICATES NO COUNTERMEASURE.

Protocol	Test target	Ref ID	Type	Platform	Reliability	Uniformity	Uniqueness	Countermeasure
Selective CRPs	Arbiter PUF	[26]	Strong	Simulation	97%	54%	50%	Data poisoning
	FF-PUF	[26]	Controlled	Simulation				Data poisoning
	IPUF	[26]	Controlled	Simulation				Data poisoning
	LSPUF	[26]	Controlled	Simulation				Data poisoning
	XOR PUF	[26]	Controlled	Simulation	86%	51%	50%	Data poisoning
Deception	Arbiter PUF	[108]	Controlled	Xilinx Artix-7	93.90%	∼47%	∼40%	Full obf.
	FF-PUF	[108]	Controlled	Simulated				Full obf.
	XOR PUF	[108]	Controlled	Simulated				Full obf.
AES	XOR PUF	[174]	Controlled	Simulated				Challenge obf.
	XOR PUF	[160]	Controlled	Simulated				Challenge obf.
DES	XOR PUF	[174]	Controlled	Simulated				Challenge obf.
	XOR PUF	[160]	Controlled	Simulated				Challenge obf.
Lockdown	Arbiter PUF	[155]	Controlled	0.18um CMOS				Challenge obf.
	XOR PUF	[75]	Controlled	Simulated				Challenge obf.
Noise bifurcation	XOR PUF	[161]	Controlled	Simulated				Challenge obf.
	XOR PUF	[175]	Controlled	Simulated				Challenge obf.
Permutation Interface	Arbiter PUF	[176]	Controlled	Simulated				Challenge obf.
	XOR PUF	[176]	Controlled	Simulated				Challenge obf.
Response inversion	Arbiter PUF	[177]	Controlled	Xilinx Artix-7				Response obf.
	XOR PUF	[178]	Controlled	Simulated				Response obf.
Slender PUF	LSPUF	[109]	Controlled	Simulated				Full obf.
	XOR PUF	[179]	Controlled	Xilinx Virtex-5	75.3%			Full obf.
Reverse Fuzzy Extractor	Arbiter PUF	[109]	Controlled	Simulated				Response obf.
Challenge Splitting	Arbiter PUF	[180]	Controlled	Xilinx Artix-7				Challenge obf.
Challenge Permutation	Arbiter PUF	[181]	Controlled	Xilinx Artix-7	99.99%	61.00%		Challenge obf.
CoLAC	Arbiter PUF	[182]	Controlled	Xilinx Artix-7	98.22%			Response obf.
DAUP	Arbiter PUF	[183]	Controlled	Xilinx Artix-7				Challenge obf.
ECC	XOR PUF	[184]	Controlled	Simulated				Response obf.
Exchangeless Key		[185]	Weak					-
Noise Injection	Arbiter PUF	[186]	Strong	Simulated				Data poisoning
RFID Auth		[187]	Controlled					Full obf.

AES: Advanced Encryption Standard, **DES**: Data Encryption Standard, **CoLAC**: Coordinated and Lightweight Adversarial Machine Learning-based Countermeasure, **DAUP**: Distributed Authentication Using PUFs, **ECC**: Error Correction Code, **RFID**: Radio Frequency Identification

of protocols are the same as for PUF designs in the previous section.

VI. MODELING TECHNIQUES

This section describes the types of PUF models that have been applied to date. We also classify the modeling techniques that have been presented in the literature. The result of this study highlights which research directions in PUF modeling are successful and which are not. Additionally, we identify the research gaps in the literature.

Table IV lists the modeling attempts on different PUF designs and protocols. Due to the large scale of this study and the fact that many papers perform modeling attempts on the same PUF implementation but with varying parameters, we choose to include in Table IV only one modeling attempt from each paper. Regarding cases where a PUF implementation is tested with and without a countermeasure, we include both. However, if many countermeasures are considered and evaluated in different combinations, we only include the most challenging countermeasure combination for modeling. Our decision for including a modeling instance is based on the following priority order: (i) the largest (and most complex) parameters for implementation where the accuracy shows a successful modeling attack (above 90%), (ii) the largest (and most complex) parameters for implementation where the accuracy show improvement from random guessing (not close to 50%), and (iii) the smallest (and least complex) parameters for implementation where the accuracy shows a failed modeling attack (around 50%). Even with the previous criteria, the total number of entries scales to 275 modeling attempts, so Table IV is cut to one page with priority for the more popular techniques and successful attempts. We again refer readers interested in the full table to the appendices (see Appendix B).

A. Types of PUF models

An essential element for effectively modeling a PUF is the use of a model. From analyzing the literature, in this survey, we identify five main model types: (i) parametric, (ii) Deterministic Finite-State Machine, (iii) black-box, (iv) white-box, and (v) physical. Fig. 5 shows the taxonomy for these models, also used in Table IV for each entry.

- **Parametric model**: This model is based on the idea that certain PUFs can be defined as mathematical functions with a set of randomly generated parameters. These parameters are randomized by the manufacturing process of the PUF and considered to be unknown. Thus, the modeling problem becomes a problem of approximating

TABLE IV

SHORT OVERVIEW OF PUF MODELING TECHNIQUES (SEE APPENDIX B FOR FULL VERSION). THE DISPLAYED AND TOTAL NUMBER OF ENTRIES ARE SHOWN AS (X / Y). PERFORMANCE VALUES ARE SELECTED BASED ON A REPRESENTATIVE WORST CASE SCENARIO. A '∼' SYMBOL INDICATES THAT THE VALUES ARE TAKEN FROM A VISUAL GRAPH AND ARE NOT EXACT. BLANK SPACES ARE INFORMATION THAT WAS NOT OBTAINABLE FROM THE REFERENCE. A '-' SYMBOL INDICATES NO COUNTERMEASURES.

	Attack	Ref ID	PUF/protocol	Countermeasure	Model	CRPs	Time†	Accuracy
Machine Learning (33 / 246)	MLP (7 / 88)	[188]	DEPUF	-	Parametric	$6*10^4$		∼94%
		[189]	FF-IPUF	Challenge obf.	Parametric + SC	10^6		96.85%
		[128]	MPUF	Challenge obf.	Black-box	$3.2*10^5$	0:15:23.0	96.54%
		[177]	Response inversion	Response obf.	Black-box	$5*10^3$		95.70%
		[190]	RF-PUF	-	White-box			99.99%
		[191]	SCA PUF	-	Parametric	$3.5*10^2$		97%
		[26]	Selective CRPs	Data poisoning	Black-box	$9*10^4$		∼95%
	LR (5 / 54)	[31]	Arbiter PUF	-	Parametric	$6.5*10^3$	0:0:0.76	99%
		[192]	CRO XMPUF	-	Parametric + SC	10^4		∼98%
		[60]	LP-PUF	Full obf.	Parametric	$5*10^5$		∼100%
		[161]	Noise bifurcation	Challenge obf.	Parametric	$5*10^5$	05:00:00	92%
		[193]	XbarPUF	-	Black-box	$5*10^3$		99.0%
	SVM (3 / 30)	[194]	FF-PUF	Challenge obf.	Parametric + SC	10^3		∼91%
		[195]	STT-MRAM PUF	-	Black-box	10^5		∼95%
		[141]	XOR BR PUF	Response obf.	Parametric	$7.2*10^3$	0:0:24.0	95%
	CMA-ES (2 / 25)	[108]	Deception	Full obf.	Parametric + SC	$2*10^6$	228 years	99%
		[196]	PUF-FSM	Response obf.	Parametric	10^2		∼100%
	ES (1 / 9)	[179]	Slender PUF	Full obf.	Black-box	$6.4*10^4$		∼93%
	RF (1 / 7)	[58]	LBIST-PUF	Challenge obf.	Black-box	$6*10^4$		∼93%
	Boosting (1 / 4)	[197]	MRO-PUF	Response obf.	Black-box	$5*10^4$	00:00:02	∼98%
	CNN (1 / 4)	[198]	RO PUF	-	Parametric + SC	$2*10^5$		92.8%
	VAE (1 / 4)	[174]	AES	Challenge obf.	Black-box	$2.5*10^2$	00:01:37	63.9%
	DQN (1 / 3)	[199]	Arbiter PUF	-	DFSM	10^4		94.98%
	DT (1 / 3)	[88]	HP mem-PUF	-	Black-box	$1.2*10^5$		∼85%
	ECP-TRN (1 / 3)	[75]	Lockdown	Challenge obf.	Black-box	$4.8*10^6$	00:48:12	98.09%
	GA (1 / 3)	[200]	Current Mirror PUF	-	Parametric	$5*10^4$	26:18:00	97.07%
	SLP (1 / 2)	[140]	TBR PUF	-	Black-box	$5*10^4$		∼95%
	GRNN	[201]	XOR PUF	Response obf.	Parametric	$7*10^4$	01:07:10	99.64%
	LM	[202]	RNN SCA-PUF	Challenge obf.	Parametric	$2*10^6$		∼54%
	LiR	[196]	RPUF	Challenge obf.	Parametric	$1.024*10^3$		∼90%
	NB	[52]	DyAdv PUF	Response obf.	Black-box	$7*10^4$		∼52%
	PAC	[203]	Arbiter PUF	-	DFSM			
	RBFNN	[46]	CBDC-PUF	-	Parametric	10^4		50.9%
	...	...	...	...	...	...	...	...
Invasive (5 / 8)	FIB (1 / 3)	[204]	SRAM PUF	-	Physical	1.6*10	00:05:00	Success
	BTI (1 / 2)	[42]	SRAM PUF	-	Physical			98.64%
	Bypass	[27]	Capacitive PUF	Anti-invasive	Physical			
	MD	[27]	Capacitive PUF	Anti-invasive	Physical			
	MP	[27]	Capacitive PUF	Anti-invasive	Physical			
	...	...	...	...	...	...	...	...
Other (10 / 21)	Sorting (1 / 3)	[205]	CIS PUF	-	Parametric	∼20		∼90%
	LP (1 / 3)	[206]	Xbar PUF	-	Parametric + SC	$1.288*10^3$	00:12:48	99.9%
	Cryptoanalysis (1 / 2)	[207]	RO PUF	Response obf.	Parametric + SC			∼100%
	Characterization	[208]	Arbiter PUF	-	Parametric	10^6		83.7%
	EM analysis	[67]	RO PUF	-	Parametric + SC	0		94.2%
	Emulation	[208]	Arbiter PUF	-	Physical			75.5%
	QS	[31]	RO PUF	-	Parametric	$8.39*10^4$		99%
	-	[63]	NoPUF	Challenge obf.	Parametric			∼92%
	-	[99]	NOS PUF	-	Physical			∼100%
	-	[209]	XMPUF	Challenge obf.	Black-box	10^4		94%
	...	...	...	...	...	...	...	...

†Format is "hours : minutes : seconds" (hh:mm:ss), unless directly specified with other units in the case of times longer than 48 hours.

BTI: Bias Temperature Instability, **CMA-ES**: Covariance Matrix Adaptation Evolutionary Strategies, **CNN**: Convolutional Neural Network, **DQN**: Deep Q-Network, **DT**: Decision Tree, **ECP-TRN**: Efficient CANDECOM/PARAFAC-Tensor Regression Network, **ES**: Evolutionary Strategies, **FIB**: Focused Ion Beam, **GA**: Genetic Algorithm, **GRNN**: General Regression Neural Network, **LM**: Lagrange Multiplier, **LiR**: Linear Regression, **LP**: Linear Programming, **LR**: Logistic Regression, **MD**: Micro-drilling, **MLP**: Multi-Layer Perceptron, **MP**: Magnetic Probing, **NB**: Naive Bayes, **PAC**: Probably Approximately Correct, **QS**: Quick Sort, **RBFNN**: Radial Basis Function Neural Network, **RF**: Random Forest, **SLP**: Single-Layer Perceptron, **SVM**: Support Vector Machine, **VAE**: Variational Auto-Encoder

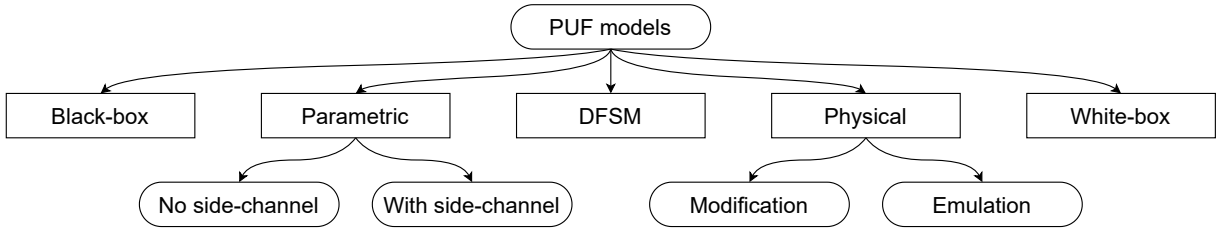

Fig. 5. PUF model type taxonomy.

the values of these parameters until full emulation of the PUF is achieved. An important note is that certain parametric models utilize side-channel information [168], [210]–[212], which is specified in Table IV as "+ SC". Many different parametric models have been defined, although the specifics of all parametric models that are used in the literature surveyed by this paper are too broad (and not included in Table IV). However, we do provide an explanation of the most popular option in Section VI-A1 - the so-called additive delay model [30], [213].

- **Deterministic Finite-State Machine (DFSM) model**: As the name suggests, this model is based on the idea that a PUF's operation for generating an output can be modeled by states and transitions. Similarly to the parametric model, the challenge is to define a state machine that adequately conveys the inner working of the PUF in question. This is hard for a PUF such as the SRAM PUF [214], since you would require an independent state machine for each bit where the state transitions are dependent on factors internal to the physical components of the SRAM module. Instead, for, e.g., an Arbiter PUF, a DFSM representation makes far more sense as the transitions will be dependent on the input challenge. Despite many candidates for such a model existing (see Table II), very few works make such attempts. However, we were able to find two examples [199], [203].
- **Black-box model**: When no known mathematical model can be applied to a PUF (or even if one exists), a different option is simply treating the PUF as a black-box. As a black-box, the PUF turns into an unknown function which must be approximated using an adequate technique [215]–[217]. This modeling is enabled by the advances in ML techniques, and especially Neural Networks (NNs) capable of theoretically approximating any function given enough data [218], [219]. However, this form of modeling is generally less effective than when a pre-defined model is available, since a larger amount of input-output data is required and there is no guarantee of successfully approximating the unknown function [220], [221]. That said, this black-box approximation remains a realistic threat to PUF that is enhanced by further advances in the field of ML [5], [35]. It is important to note that some papers describe an existing model for their target PUF but proceed to perform an attack using a black-box model. In this paper we consider any modeling technique that does not explicitly define a model and tailored technique as black-box.
- **White-box model**: In complete opposition to the black-box approach, the white-box model assumes that modeling is performed with complete access and design information for the PUF. The main argument for this approach is to create a PUF clone legally to support the applications of PUFs. On the other hand, it is obvious that white-box modeling is unrealistic for adversaries (aside from a single party charged with all steps of PUF manufacturing and supply).
- **Physical model**: These methods differ from the previous approaches as they use the physical design and characteristics of the PUF instead of defining mathematical models. The methods to physically model a PUF are either through emulation of physical characteristics [208] or by physically modifying a PUF [42], [115].

When looking at the presence of the previous models in the literature, black-box modeling overshadows the rest with 52.8% of cases. It is easy to conclude that the ease of use and availability of advanced ML techniques influences this statistic. Meanwhile, parametric modeling is a strong second choice with 29.2%, to which we should add another 12% when including side-channel information. The popularity in this case is explained with the undeniable effectiveness of the techniques when properly applied. On the other hand, the use of DFSM was only shown to be theoretically possible in [203], and further simplified and experimentally proven in [199]. The lack of research in this direction puts it at a very low 0.4%. Similarly, white-box modeling is only seen 0.8% of the time, because the threat model is less realistic for adversaries. Finally, physical cloning shows promise with 4.8% but moderated by the stricter entry barrier for researchers. Following, we describe some representative examples of each model type, excluding black-box as all such models are inherently unknown internally.

1) (Parametric) additive delay model: Starting with parametric models, the most well-known PUF model is the additive delay model, initially proposed for the Arbiter PUF [30], [213]. Its importance comes from the fact that the Arbiter PUF, as shown in Fig. 6, is the building block for most other delay-based PUF designs (see Section IV-B1).

In the context of the Arbiter PUF, the additive delay model considers each stage as an individual switch element (e.g., multiplexer). Consider the i-th stage. The delay between both inputs to their corresponding outputs is determined by four critical delay parameters: p_i, q_i, r_i, s_i. The key idea of the model is that each stage adds a delay between its outputs

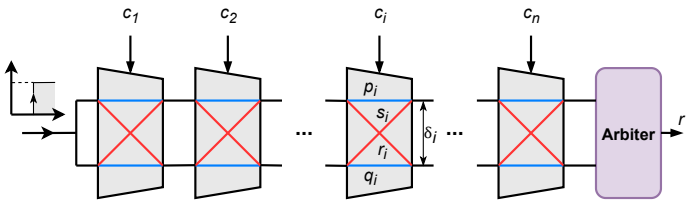

Fig. 6. Schematic view of n-bit Arbiter PUF.

without interference from the other stages (hence the term "additive" model). Following the previous example, the i-th stage's delay can be expressed as δ_i. Based on the input to the Arbiter PUF, i.e., the challenge $\vec{C} = [c_1, ..., c_n]$, this delay can take two values:

$$\delta_i^{c_i} = \begin{cases} p_i - q_i, & c_i = 0 \\ r_i - s_i, & c_i = 1 \end{cases} \tag{4}$$

Using both these values, one can define each stage based on the difference between these delays, while adding the common delays that all paths have from the previous stage. The only exception is the first stage, since it does not have any stages prior to it. Thus, we can define the Arbiter PUF as $\vec{w} = [w_1, ..., w_{n+1}]$, with each parameter defined as:

$$w_i = \begin{cases} \frac{\delta_1^0 - \delta_1^1}{2}, & i = 1 \\ \frac{\delta_{i-1}^0 + \delta_{i-1}^1 + \delta_i^0 - \delta_i^1}{2}, & \forall i \in [2, n] \\ \frac{\delta_n^0 + \delta_n^1}{2}, & i = n+1 \end{cases} \tag{5}$$

Notice that there are $n + 1$ elements instead of n. This is because the common delay from previous stages must be considered for all stages except the first, however the term w is defined including only the previous stage. Thus, we need the additional w_{n+1} term to include the common delay of the n-th stage. For additional information regarding the additive delay model, including the details for how to define w_i, we refer the reader to [213].

Finally, given the Arbiter PUF parameters ($\vec{w}$), the output r can be calculated as:

$$r = sign(\vec{w}^T \vec{\Phi}) \tag{6}$$

where Φ represents the input challenge $\vec{C}$ transformed from its binary form[3]:

$$\vec{\Phi}(\vec{C}) = [\Phi_1(\vec{C}), ..., \Phi_n(\vec{C}), 1] \tag{7}$$

$$\Phi_i(\vec{C}) = \prod_{i=1}^{n} (-1)^{c_i} \tag{8}$$

The additive delay model can obviously be extended to delay-based PUF designs that incorporate Arbiter PUFs. Clear examples are the FF-PUF [222], XOR PUF [223], LSPUF [61], MPUF [132], IPUF [224] or DAPUF [225]. We note that many other delay-based designs from Table II use Arbiter PUFs, although in many cases they include modules (Controlled PUFs) which make it challenging to adapt the additive delay model. Apart from this, delay-based PUFs that are not based on the Arbiter PUF can also have an additive delay model defined for them. We can find obvious examples of this in the BR PUF [141], TBR PUF [141], and Current Mirror PUF [200], all of which had mathematical models defined for them in a similar fashion to the Arbiter PUF.

2) DFSM model: This approach for modeling the Arbiter PUF from Fig. 6 is taken in [203]. The model begins by following the same approach as the additive delay model, where the total delay at the two outputs of each stage can be described as δ_i^t and δ_i^b, for the top and bottom paths respectively. These values are the sum of the delays of each path until the i-th stage, and their difference marks the delay difference $\delta_i = \delta_i^t - \delta_i^b$ (Fig. 4).

However, the authors realize that the four key wire delays of each stage (p_i, q_i, s_i, and r_i) follow a Gaussian distribution $N(\mu_i, \sigma_i)$. Assuming that all stages have the same mean and standard deviation (which is reasonable since they are manufactured together in close proximity), for an n-bit Arbiter PUF, $\mu = \mu_1 = ... = \mu_n$ and $\sigma = \sigma_1 = ... = \sigma_n$. By utilizing this assumption, the distribution of the total delay of all stages must follow the Gaussian distribution $N(n\mu, \sqrt{n}\sigma)$. Additionally, we can also consider that the Arbiter at the end of the chain has a limited precision threshold γ for differentiating the final path delays. Therefore, the final output of the Arbiter PUF is given by (9).

$$r = \begin{cases} 1, & \delta_n^t - \delta_n^b > \gamma \\ 0, & \delta_n^t - \delta_n^b < \gamma \\ X, & |\delta_n^t - \delta_n^b| \leq \gamma \end{cases} \tag{9}$$

Notice that there exists a metastable state for the Arbiter if the delay difference is too low. This is simply an unreliable response, and the lower the value of γ is, the more reliable the Arbiter PUF will be. Moreover, due to the limited precision of the Arbiter and the fact that 99.7% of a Gaussian distribution values fall into an interval of 6σ, the real values of the delays at the outputs of the stages can be mapped to integer values by $f : \mathbb{R} \to \mathbb{Z}$ as shown by (10).

$$f(\delta) = \left\lceil \frac{\delta - n\mu + 3\sigma\sqrt{n}}{\gamma} \right\rceil \tag{10}$$

This mapping means that $\delta_i \in \{0, M\}$ for all delay values that are in the interval $[n\mu - 3\sigma\sqrt{n}, n\mu + 3\sigma\sqrt{n}]$ (99.7% of values), and that M can be expressed by (11).

$$M = \left\lceil \frac{6\sqrt{n}\sigma}{\gamma} \right\rceil \tag{11}$$

With the previous observations it is possible to construct a DFSM representing an Arbiter PUF. Note that it was always possible to represent an Arbiter PUF as a DFSM by simply starting from an initial state, and then for each state transition the DFSM grows by twice as many states as it already has. Basically, this representation considers each state as all the possible combinations of previous stages, where at each stage the paths are either straight or crossed, and the resulting delay difference δ_i is either lower or higher than 0. However, such a DFSM grow exponentially with n (number of stages). That is why the DFSM constructed in [203] is an improvement,

[3]By binary form we refer to the fact that the challenge bits $c_i = \{0, 1\}$. Some authors convert this to $c_i = \{-1, 1\}$, respectively. In that case, the definition of $\Phi_i(\vec{C})$ in (8) should be adjusted accordingly.

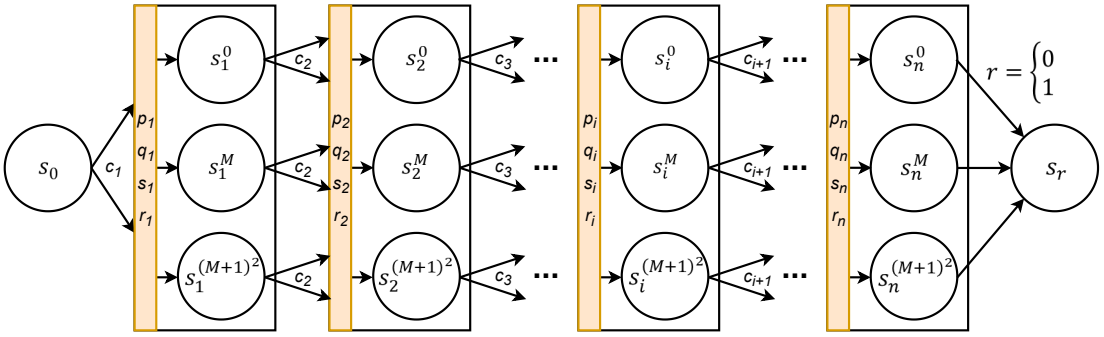

Fig. 7. Polynomial size DFSM representation of a n-bit Arbiter PUF.

as it is of polynomial size in n, as shown in Fig 7. This DFSM has an initial and terminal state, and for each stage of the Arbiter PUF a total of $(M+1)^2+1$ states. Each of these states represents a unique differentiable pair of δ_i^t and δ_i^b, i.e., the granularity of δ_i that the Arbiter can differentiate due to its precision of γ. Note that each state has two possible transitions depending on the challenge bit c_i for that stage, and the state to which this transition connects is determined by the delay parameters p_i, q_i, r_i, and s_i. In other words, each Arbiter PUF instance transitions are determined by the random manufacturing variations.

To model an Arbiter PUF with the polynomial DFSM representation, one would have to learn the state transitions. Such a problem is solvable with different algorithms and as such is a valid Arbiter PUF model. Additionally, it is possible to extend the DFSM to other PUF architectures, although it may come at the cost of a significant increase in complexity. On the other hand, it is also possible to reduce the number of states per stage further for easier learnability, such as in [199], at the cost of a lower fidelity towards the real PUF model.

3) White-box PUF probing: In this literature review, a few works that assume unrestricted access to certain PUF parameters can be found [33], [190]. Generally, this approach is only interesting for developing functional PUF clones for applications, not as an attack method. There exist obvious and simple examples of white-box modeling (not seen in any papers in this review), namely, exhaustive modeling. For example, we can model an SRAM PUF by recording all output bits with privileged access during manufacturing. Another example would be to probe the delays of each connection on an Arbiter PUF.

4) Physical modification and emulation of PUFs: There are two alternatives when physically modeling a PUF. The first is to physically modify a PUF to manipulate its output values in a predictable manner. An excellent work for such a technique was presented in [42], where the authors managed to modify the outputs of an SRAM PUF via a process called Bias Temperature Instability (BTI) aging. Using this technique, any SRAM PUF could be configured to have a specific set of CRPs, which means that a supplier could maliciously control, e.g., a cryptographic key embedded in a device. Alternatively, a supplier could also create legitimate clones of SRAM PUFs for symmetric cryptography purposes. The second option for physical modeling of PUFs is emulation [208]. This method is similar to parametric modeling, but involves modeling the PUF characteristics at a physical level, instead of mathematical.

B. PUF modeling techniques

Aside from selecting a model, PUF modeling requires an algorithm or other technique that approximates the exact PUF instance based on observed data. In this survey, these techniques are broadly grouped into: (i) ML-based, (ii) invasive, and (iii) other. However, we need to specify algorithms and techniques to provide a full overview, thus the classification goes into a second level for each, as illustrated in Fig. 8.

- **ML**: By far the most popular approach towards approximating a PUF model is the use of ML algorithms. These algorithms are trained on data collected from the PUFs, i.e., CRPs. Data collection is considered viable for adversaries as they can collect CRPs passively from the communication channel in off-device applications. Additionally, side-channel information can enhance the model, but may require additional pre-processing or physical access to the target PUF. There has been a wide variety of ML algorithm applied to PUF modeling; however, the majority follow a supervised learning approach. The current highlights are optimized LR [31], [32] and MLP [32], [189] for passive modeling, and the reliability-based Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES) [212] considering side-channels.
- **Invasive**: This type of technique assumes physical unrestricted access to the PUF, and its goal is to challenge the "physically unclonable" property of PUFs. Invasive modeling is different from simply collecting side-channel information, as the latter typically only involves measurement on the physical device with no manipulation or invasive probing. As a highly specialized group of techniques, invasive modeling is less accessible and popular. However, some work has shown the viability of such techniques [114], [204] and proves the need for further research into the unique fingerprinting capabilities of PUFs, even for Weak PUFs and on-device applications.
- **Other**: While reviewing the literature we encountered papers that had a unique approach to the modeling of PUFs which had no clear connection to the previous two categories of techniques [211]. As each is unique and strongly specialized to the context of the respective

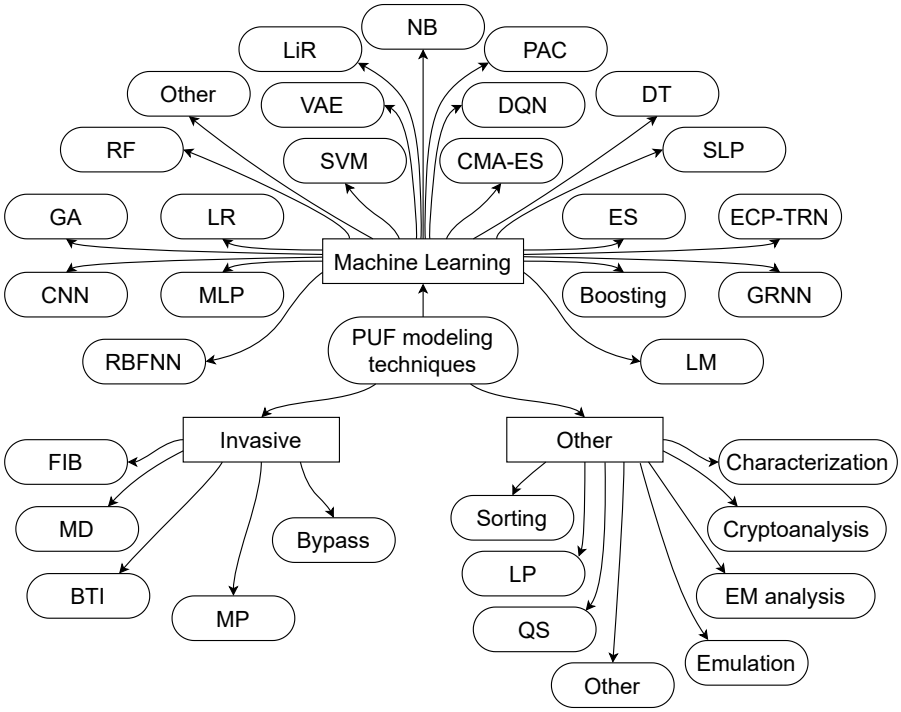

Fig. 8. PUF modeling technique taxonomy. **BTI**: Bias Temperature Instability, **CMA-ES**: Covariance Matrix Adaptation Evolutionary Strategies, **CNN**: Convolutional Neural Network, **DQN**: Deep Q-Network, **DT**: Decision Tree, **ECP-TRN**: Efficient CANDECOM/PARAFAC-Tensor Regression Network, **ES**: Evolutionary Strategies, **FIB**: Focused Ion Beam, **GA**: Genetic Algorithm, **GRNN**: General Regression Neural Network, **LM**: Lagrange Multiplier, **LiR**: Linear Regression, **LP**: Linear Programming, **LR**: Logistic Regression, **MD**: Micro-drilling, **MLP**: Multi-Layer Perceptron, **MP**: Magnetic Probing, **NB**: Naive Bayes, **PAC**: Probably Approximately Correct, **QS**: Quick Sort, **RBFNN**: Radial Basis Function Neural Network, **RF**: Random Forest, **SLP**: Single-Layer Perceptron, **SVM**: Support Vector Machine, **VAE**: Variational Auto-Encoder.

papers, they require individual attention, which leads us to group them as "other".

Given the large number of techniques, Fig. 9 illustrates the proportion of use of each technique in the literature. We can observe how ML techniques take up most of the work with an 89.2%. Given that the problem of modeling PUFs can be considered as a binary classification problem, the application of many ML algorithms is not surprising. Looking at some specific techniques, it is clear that Neural Networks (NN), especially feed-forward NNs with back propagation, are the most popular. We can see that MLP occupies almost a third of the literature, with other NN-based algorithms being considerably less frequent. A strong second place goes to LR, followed by Support Vector Machine (SVM) and CMA-ES or Evolutionary Strategies (ES).

Meanwhile, only 3.2% are invasive and 7.6% are other specialized techniques. Regarding other techniques, we believe that this is an adequate amount of research towards niche approaches that are highly specialized. However, the low number of invasive techniques is more concerning. Given the successful invasive techniques shown in [42], [114], [115], we would argue that this direction of research is significantly underdeveloped. This is significant since it affects all applications of PUFs due to the possible breaking of the physical unclonability property. An interesting observation in modeling is that seemingly unrelated metrics can have significant effects

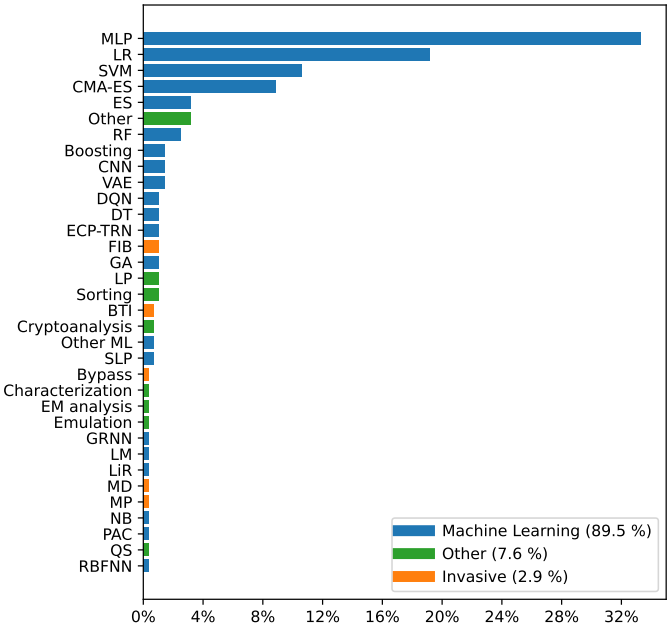

Fig. 9. Percentages of modeling attempts for each modeling technique. See Fig 8 for the acronyms.

on the modeling algorithms, e.g., uniformity [226].

VII. Open challenges

The open challenges in PUF modeling and PUF applications limited by modeling attacks can be approached from different perspectives. First, PUF designs focus mainly on creating secure Strong PUFs, as they have a broader set of applications compared to Weak PUFs and are thus more desirable. However, this leads to overheads through large designs or Controlled PUFs with additional modules [227]. Moreover, most designs are based on delay-based PUFs, with a majority being extensions of a previously proposed architecture called the Arbiter PUF. This creates a problem, as the modeling vulnerability of the original PUF is transmitted to its extensions, making modeling attacks more challenging but not infeasible.

While some work into Strong PUF designs besides delay-based is ongoing [228]–[230], it is yet to provide a widely accepted proposal. This is partially because a Strong PUF design with uncorrelated CRPs is unlikely to be possible, as a limited number of hardware components cannot generate unlimited entropy [104], [105]. Regardless, a lightweight Strong PUF capable of providing an extremely large number of CRPs could still prove practical for off-device applications [26]. An important element to this is a formal security proof accompanying a practical implementation [231], but to the best of our knowledge, no work has combined both these aspects for modeling.

A different approach towards practical PUF off-device applications is through countermeasures. The majority choose to obfuscate CRP information from an attacker. In most cases this prevents modeling attacks but requires more complex and expensive hardware and protocol design. Despite the cost, authors typically argue that their Controlled PUFs remain very lightweight [169], [232], [233]. However, the vulnerabilities of additional logic have become a liability in the past [109], [174], [177], [234].

Nowadays, obfuscation countermeasures have so many similar proposals that further significant improvements seem unlikely. Alternatively, data poisoning methods can be adopted, where an attacker has access to handpicked CRPs, making training accurate ML models difficult. PUF designs and protocols that acknowledge the limited entropy and aim to provide a provable large number of CRPs are an interesting research direction [26], [186], [235] with few contributions thus far. Additionally, side-channel and invasive countermeasures have only recently gained importance due to vulnerabilities related to the physical properties of PUFs [42], [67], [115], [192], [204].

Finally, we should not forget the open challenges in modeling techniques themselves. Almost all modeling techniques for PUFs are based on ML techniques. Despite this, there is an illusion that many secure PUFs exist as of the current state-of-the-art. The problem is that most new PUF designs aim to prevent ML-based modeling and use it as a benchmark. However, most of these models cannot be applied to many new designs [236], and thus the results are unsurprising. This is shown by the fact that some outliers that successfully exploit Controlled PUFs have been shown in literature [237]. Thus, most current PUF designs and protocols cannot provide security guarantees, since better modeling techniques could be developed [32], [212], [221], [222], [224], [238].

A silver bullet approach which can model many PUF designs as a black-box is the end goal of the research topic. For the purpose of achieving such a model, further research into black-box models of PUF should be performed. A first step in this direction would be to create a model which can successfully model two different types of PUFs without more than a few tweaks to its configuration. A different idea would be to attempt to model some of the Controlled PUF that obfuscate CRP information as a black-box.

Alternatively, tailor-made modeling for specific PUF designs or protocols is rare [211], [224], [237]–[239], leaving a large gap for new attacks towards many of the current "secure" designs. However, given the lack of widely adopted PUF designs and protocols aside from a chosen few, this research direction has low impact. Meanwhile, invasive modeling techniques, although more challenging, are gaining traction as their viability challenges even the physical unclonability property of PUFs [42], [115].

VIII. Conclusion

The Physical Unclonable Function (PUF) supports security applications through its unique hardware fingerprinting capabilities. Unique hardware physical characteristics can be leveraged to provide enhanced physical security on devices and enable resource-constrained security applications including cryptography and authentication. However, due to the existence of many modeling techniques that can effectively clone PUFs, the applicability of this technology is limited in practice. In this paper, we screened over 400 papers into 222 relevant papers and classified the different proposals for PUFs, the countermeasures against modeling attacks, and the modeling techniques (not necessarily adversarial) that have been applied so far.

From our results, we can conclude that many PUF designs with no known modeling attacks exist, but there is no guarantee that an improved or tailor-made attack will not appear in the future. Even for PUFs employed in applications that only use the primitive on the device, recent invasive techniques have raised questions about their security. Despite this, for every vulnerability, new designs and protocols have been developed, making PUFs retain their potential as a lightweight security primitive. Thus, we believe that this field is waiting for a PUF design with formal security or a powerful modeling technique that disproves the technology's viability.

Acknowledgements

This work is partially supported by the IoTalentum project, funded by the European Union Horizon 2020 research and innovation program within the framework of Marie Skłodowska-Curie Actions ITN-ETN with grant number 953442. This work is also partially supported by the Thomas B. Thriges Fond.

REFERENCES

[1] K. Y. Annapurna and D. Koppad, "A lightweight hardware secure and reliable framework using secure and provable puf for iot devices against the machine learning attack," *International Journal of Circuits*, vol. 16, pp. 699–709, 2022.

[2] S. Rajput and J. Dofe, "Counteracting modeling attacks using hardware-based dynamic physical unclonable function," *IEEE International Conference on Cyber Security and Resilience*, pp. 586–591, 2023.

[3] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, "Memristive crypto primitive for building highly secure physical unclonable functions," *Scientific Reports*, vol. 5, no. 1, p. 12785, 2015.

[4] M. S. Mispan, A. Z. Jidin, H. M. Nasir, N. M. A. Brahin, and I. M. Nawi, "Modeling arbiter-puf in nodemcu esp8266 using artificial neural network," *International Journal of Reconfigurable and Embedded Systems*, vol. 11, no. 3, pp. 233–239, 2022.

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

[6] P. Santikellur and R. S. Chakraborty, *Modeling Attacks on PUF*, ser. Studies in Computational Intelligence. Springer Science and Business Media Deutschland GmbH, 2023, vol. 1052, pp. 35–53.

[7] A. Al-Meer and S. Al-Kuwari, "Physical unclonable functions (puf) for iot devices," *ACM Computing Surveys*, vol. 55, no. 14s, 2023.

[8] C. Yehoshuva, R. Raja Adhithan, and N. Nalla Anandakumar, "A survey of security attacks on silicon based weak puf architectures," in *Security in Computing and Communications*, S. M. Thampi, G. Wang, D. B. Rawat, R. Ko, and C.-I. Fan, Eds. Singapore: Springer Singapore, 2021, pp. 107–122.

[9] Y. Li, J. Shen, W. Liu, and W. Zou, "A survey on side-channel attacks of strong puf," pp. 74–85, 2020.

[10] J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede, "Secure lightweight entity authentication with strong pufs: Mission impossible?" in *Cryptographic Hardware and Embedded Systems*, L. Batina and M. Robshaw, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 451–475.

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

[12] U. Rührmair and J. Sölter, "Puf modeling attacks: An introduction and overview," Technische Universität München, 80333 München, Germany; Freie Universität Berlin, 14195 Berlin, Germany. Institute of Electrical and Electronics Engineers Inc, 2014.

[13] J.-L. Zhang, G. Qu, Y.-Q. Lv, and Q. Zhou, "A survey on silicon pufs and recent advances in ring oscillator pufs," *Journal of Computer Science and Technology*, vol. 29, no. 4, pp. 664–678, 2014.

[14] I. A. Bautista Adames, J. Das, and S. Bhanja, "Survey of emerging technology based physical unclonable funtions," in *Proceedings of the 26th Edition on Great Lakes Symposium on VLSI*. New York, NY, USA: Association for Computing Machinery, 2016, p. 317–322.

[15] B. Halak, M. Zwolinski, and M. S. Mispan, "Overview of puf-based hardware security solutions for the internet of things," in *IEEE 59th International Midwest Symposium on Circuits and Systems*, 2016, pp. 1–4.

[16] M. S. Alkatheiri, Y. Zhuang, M. Korobkov, and A. R. Sangi, "An experimental study of the state-of-the-art pufs implemented on fpgas," in *IEEE Conference on Dependable and Secure Computing*, 2017, pp. 174–180.

[17] C.-H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: Fifteen years of physical unclonable function advancement," *IEEE Circuits and Systems Magazine*, vol. 17, no. 3, pp. 32–62, 2017.

[18] A. Babaei and G. Schiele, "Physical unclonable functions in the internet of things: State of the art and open challenges," *Sensors*, vol. 19, no. 14, 2019.

[19] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (puf)-based security solutions for internet of things," *Computer Networks*, vol. 183, p. 107593, 2020.

[20] P. Lokhande and A. Shah, "Strong authentication and encryption modeling using physical unclonable function based on fpga," in *6th International Conference on Communication and Electronics Systems*, 2021, pp. 192–195.

[21] P. Mall, R. Amin, A. K. Das, M. T. Leung, and K.-K. R. Choo, "Puf-based authentication and key agreement protocols for iot, wsns, and smart grids: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8205–8228, 2022.

[22] M. Khalafalla and C. Gebotys, *The Physical Unclonable Functions Fight: State-of-the-Art Architectures and Their Performance Against Advanced Deep Learning Modeling Attacks*, ser. Machine Learning for Embedded System Security. Springer International Publishing, 2022, pp. 67–102.

[23] M. Ferens, E. Dushku, and S. Kosta, "Securing pufs against ml modeling attacks via an efficient challenge-response approach," in *IEEE Conference on Computer Communications Workshops*, 2023, pp. 1–6.

[24] R. A. Alhamarneh and M. Mahinderjit Singh, "Strengthening internet of things security: Surveying physical unclonable functions for authentication, communication protocols, challenges, and applications," *Applied Sciences*, vol. 14, no. 5, 2024.

[25] U. Rührmair and M. van Dijk, "Pufs in security protocols: Attack models and security evaluations," in *IEEE Symposium on Security and Privacy*, 2013, pp. 286–300.

[26] M. Ferens, E. Dushku, and S. Kosta, "When random is bad: Selective crps for protecting pufs against modeling attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, p. 1, 2024.

[27] K. Garb, M. Schink, M. Hiller, and J. Obermaier, "Attacks and countermeasures for capacitive puf-based security enclosures," *Proceedings of the IEEE International Conference on Physical Assurance and Inspection on Electronics*, 2021.

[28] T. Kroeger, W. Cheng, J. L. Danger, S. Guilley, and N. Karimi, "Cross-puf attacks: Targeting fpga implementation of arbiter-pufs," *Journal of Electronic Testing: Theory and Applications*, vol. 38, no. 3, pp. 261–277, 2022.

[29] K. T. Mursi, B. Thapaliya, Y. Zhuang, A. O. Aseeri, and M. S. Alkatheiri, "A fast deep learning method for security vulnerability study of xor pufs," *MDPI Electronics*, 2020.

[30] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*. New York, NY, USA: Association for Computing Machinery, 2002, p. 148–160.

[31] U. Rührmair, J. Sölter, F. Sehnke, X. L. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "Puf modeling attacks on simulated and silicon data," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[32] N. Wisiol, B. Thapaliya, K. T. Mursi, J. P. Seifert, and Y. Zhuang, "Neural network modeling attacks on arbiter-puf-based designs," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2719–2731, 2022.

[33] Y. He, Q. Yu, and K. Yang, "A lossless and modeling attack-resistant strong puf with ¡4e-8 bit error rate," vol. 2022-April, Rice University, Houston, TX, United States. Institute of Electrical and Electronics Engineers Inc, 2022.

[34] B. Thapaliya, K. T. Mursi, and Y. Zhuang, "Machine learning-based vulnerability study of interpose pufs as security primitives for iot networks," in *IEEE International Conference on Networking*, 2021, pp. 1–7.

[35] M. Khalafalla and C. Gebotys, "Pufs deep attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter pufs," in *Design*, 2019, pp. 204–209.

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

[37] Y. J. Zhang, P. J. Wang, Y. Li, X. X. Zhang, Z. Y. Yu, and Y. B. Fan, "Model and physical implementation of multi-port puf in 65 nm cmos," *International Journal of Electronics*, vol. 100, no. 1, pp. 112–125, 2013.

[38] S. S. Zalivaka, A. A. Ivaniuk, and C.-H. Chang, "Reliable and modeling attack resistant authentication of arbiter puf in fpga implementation with trinary quadruple response," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, 2019.

[39] M. Ebrahimabadi, M. Younis, W. Lalouani, and N. Karimi, "A novel modeling-attack resilient arbiter-puf design," *34th International Conference on VLSI Design and 20th International Conference on Embedded Systems*, no. 34, pp. 123–128, 2021.

[40] A. Dheeraj, P. Das, K. A. Kiran, S. Kalanadhabhatta, and A. Acharyya, "Modeling attacks resilient multiple puf-cprng architecture design methodology," *IEEE 35th International System-on-Chip Conference*, pp. 154–159, 2022.

[41] Y. Nozaki, K. Asahi, and M. Yoshikawa, "Puf id generation method for modeling attacks," in *IEEE 3rd Global Conference on Consumer Electronics*, 2014, pp. 393–394.

[42] S. Y. Duan and G. Sai, "Bti aging-based physical cloning attack on sram puf and the countermeasure," *Analog Integrated Circuits and Signal Processing*, vol. 117, no. 1-3, pp. 45–55, 2023.

[43] S. Alahmadi, K. Khalil, and M. Bayoumi, "Enhancing arbiter puf against modeling attacks using constant weight encoding," 2024, p. 173–177.

[44] R. Maes and I. Verbauwhede, *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 3–37.

[45] Z. Q. He, C. Wang, T. Ke, Y. J. Zhang, W. J. Cao, and J. C. Jiang, "A highly reliable fpga-based ro puf with enhanced challenge response pairs resilient to modeling attacks," *IEICE Electronics Express*, vol. 18, no. 20, 2021.

[46] Y. J. Liu, J. W. Li, T. Z. Qu, and Z. B. Dai, "Cbdc-puf: A novel physical unclonable function design framework utilizing configurable butterfly delay chain against modeling attack," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 5, 2023.

[47] G. Li, X. L. Shao, P. J. Wang, X. J. Ma, H. Li, and H. Ye, "Anti-machine-learning-attack strong puf design based on multi-path delay selection strategy," *Microelectronics Journal*, vol. 153, 2024.

[48] J. L. Zhang, C. Q. Shen, Z. Y. Guo, Q. Wu, and W. L. Chang, "Ct puf: Configurable tristate puf against machine learning attacks for iot security," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 452–14 462, 2022.

[49] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A new arbiter puf for enhancing unpredictability on fpga," *The Scientific World Journal*, vol. 2015, p. 864812, 2015.

[50] Y. Wang, C. Wang, C. Gu, Y. Cui, M. O'Neill, and W. Liu, "A dynamically configurable puf and dynamic matching authentication protocol," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 1091–1104, 2022.

[51] Q. Wang, M. Z. Gao, and G. Qu, "A machine learning attack resistant dual-mode puf," *Proceedings of the Great Lakes Symposium on VLSI*, pp. 177–182, 2018.

[52] Y. Kou, H. Jiang, D. Deng, and W. Mou, "Da puf: dynamic adversarial puf against machine learning attacks," vol. 13176, 2024.

[53] A. O. Aseeri, "A problem-tailored adversarial deep neural network-based attack model for feed-forward physical unclonable functions," *ACM Transactions on Design Automation of Electronic Systems*, vol. 28, no. 4, 2023.

[54] L. J. Wu, Y. P. Hu, K. H. Zhang, W. J. Li, X. L. Xu, and W. L. Chang, "Flam-puf: A response-feedback-based lightweight anti-machine-learning-attack puf," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4433–4444, 2022.

[55] H. Li, W. J. Cao, C. Wang, X. R. Zhu, G. S. Liao, and Z. Q. He, "Fom-cds puf: A novel configurable dual state strong puf based on feedback obfuscation mechanism against modeling attacks," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E106A, no. 10, pp. 1311–1321, 2023.

[56] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose puf: Secure puf design against state-of-the-art machine learning attacks," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 4, pp. 243–290, 2019.

[57] H. Awano and T. Sato, "Ising-puf: A machine learning attack resistant puf featuring lattice like arrangement of arbiter-pufs," in *Design, Automation & Test in Europe Conference & Exhibition*, 2018, pp. 1447–1452.

[58] M. Shintani, T. Mino, and M. Inoue, "Lbist-puf: An lbist scheme towards efficient challenge-response pairs collection and machine-learning attack tolerance improvement," *IEEE 29th Asian Test Symposium*, no. 29, pp. 192–197, 2020.

[59] L. R. Zhou, J. J. Wang, Z. Huang, L. Fan, Q. Wang, J. H. Liu, and B. Wan, "A logic encryption-enhanced puf architecture to deceive machine learning-based modeling attacks," *IEEE 32nd Asian Test Symposium*, no. 32, pp. 225–230, 2023.

[60] N. Wisiol, *Modeling Attack Security of Physical Unclonable Functions based on Arbiter PUFs*, ser. T-Labs Series in Telecommunication Services. Springer Science and Business Media B.V, 2023, pp. 1–116.

[61] D. P. Sahoo, P. H. Nguyen, D. Mukhopadhyay, and R. S. Chakraborty, "A case of lightweight puf constructions: Cryptanalysis and machine learning attacks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1334–1343, 2015.

[62] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, and P. H. Nguyen, "A multiplexer-based arbiter puf composition with enhanced reliability and security," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 403–417, 2018.

[63] A. T. Wang, W. H. Tan, Y. J. Wen, and Y. J. Lao, "Nopuf: A novel puf design framework toward modeling attack resistant pufs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2508–2521, 2021.

[64] C. Y. Xu, L. T. Zhang, M. K. Law, X. J. Zhao, P. I. Mak, and R. P. Martins, "Modeling-attack-resistant strong puf exploiting stagewise obfuscated interconnections with improved reliability," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 16 300–16 315, 2023.

[65] X. Zhang, L. Ye, X. R. Zhu, and Z. Q. He, "Low-cost anti-modeling attack puf circuit based on configurable apuf," *Ieice Electronics Express*, vol. 21, no. 22, 2024.

[66] Z. F. Huang, Y. K. Lin, F. S. Zeng, J. C. Bian, Z. Yang, H. G. Liang, Y. C. Lu, L. Yao, X. Q. Wen, and T. M. Ni, "Pfo puf: A lightweight parallel feed obfuscation puf resistant to machine learning attacks," *8th International Test Conference in Asia*, 2024.

[67] M. Shiozaki and T. Fujino, "Simple electromagnetic analysis attack based on geometric leak on asic implementation of ring-oscillator puf," *Journal of Cryptographic Engineering*, vol. 11, no. 3, pp. 201–212, 2021.

[68] J. Ye, Y. Hu, and X. Li, "Rpuf: Physical unclonable function with randomized challenge to resist modeling attack," in *IEEE Asian Hardware-Oriented Security and Trust*, 2016, pp. 1–6.

[69] Y. J. Peng, D. Deng, Z. Y. Wang, and Y. Guo, "Scd-puf: Shuffled chaotic-dual-puf with high machine learning attack resilience," *8th International Test Conference in Asia*, 2024.

[70] C. Y. Xu, L. T. Zhang, P. I. Mak, R. P. Martins, and M. K. Law, "Fully symmetrical obfuscated interconnection and weak-puf-assisted challenge obfuscation strong pufs against machine-learning modeling attacks," *IEEE Transactions on Information Forensics and Security*, vol. 19, p. 3927–3942, 2024.

[71] S. Hou, D. Deng, Z. Y. Wang, J. H. Shi, S. Q. Li, and Y. Guo, "A dynamically configurable lfsr-based puf design against machine learning attacks," *CCF Transactions on High Performance Computing*, vol. 3, no. 1, pp. 31–56, 2021.

[72] L. Tang, H. Ji, K. Liu, J. Zhou, and W. Fan, "A hybrid strong-weak puf against machine learning attacks with high reliability," in *9th International Conference on Integrated Circuits and Microsystems*, 2024, pp. 402–406.

[73] C. Y. Xu, J. Y. Zhang, M. K. Law, X. J. Zhao, P. I. Mak, and R. P. Martins, "Transfer-path-based hardware-reuse strong puf achieving modeling attack resilience with ¿200 million training crps," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2188–2203, 2023.

[74] Q. Ma, C. Gu, N. Hanley, C. Wang, W. Liu, and M. O'Neill, "A machine learning attack resistant multi-puf design on fpga," in *23rd Asia and South Pacific Design Automation Conference*, 2018, pp. 97–104.

[75] P. Santikellur and R. S. Chakraborty, "A computationally efficient tensor regression network-based modeling attack on xor arbiter puf and its variants," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1197–1206, 2021.

[76] V. K. Rai, S. Tripathy, and J. Mathew, "2spuf: Machine learning attack resistant sram puf," Indian Institute of Technology Patna, Department of Comp. Science Engg., Patna, India. Institute of Electrical and Electronics Engineers Inc, 2020, pp. 149–154.

[77] M. Khalafalla, M. A. Elmohr, and C. Gebotys, "Going deep: Using deep learning techniques with simplified mathematical models against xor br and tbr pufs (attacks and countermeasures)," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2020, pp. 80–90.

[78] K. Liu, Z. Fu, G. Li, H. Pu, Z. Guan, X. Wang, X. Chen, and H. Shinohara, "36.3 a modeling attack resilient strong puf with feedback-spn structure having ¡0.73

[79] J. Das, K. Scott, S. Rajaram, D. Burgett, and S. Bhanja, "Mram puf: A novel geometry based magnetic puf with integrated cmos," *IEEE Transactions on Nanotechnology*, vol. 14, no. 3, pp. 436–443, 2015.

[80] V. Suresh, R. Kumar, M. Anders, H. Kaul, V. De, and S. Mathew, "A 0.26% ber, 1028 challenge-response machine-learning resistant strong-puf in 14nm cmos featuring stability-aware adversarial challenge selection," in *IEEE Symposium on VLSI Circuits*, 2020, pp. 1–2.

[81] M. Mahmoodi, H. Nili, S. Larimian, X. J. Guo, and D. Strukov, "Chipsecure: A reconfigurable analog eflash-based puf with machine

learning attack resiliency in 55nm cmos," *Proceedings of the 56th ACM/EDAC/IEEE Design Automation Conference*, no. 56, 2019.

[82] Q. Ding, H. Jiang, J. Li, C. Liu, J. Yu, P. Chen, Y. Zhao, Y. Ding, T. Gong, J. Yang, Q. Luo, Q. Liu, H. Lv, and M. Liu, "Unified 0.75pj/bit trng and attack resilient 2f2/bit puf for robust hardware security solutions with 4-layer stacking 3d nbox threshold switching array," in *IEEE International Electron Devices Meeting*, 2021, pp. 39.2.1–39.2.4.

[83] P. Li, Z. Hou, H. Gao, B. Wang, and Z. Wang, "A reconfigurable and machine learning attack resistant strong puf based on arbiter mechanism and sot-mram," in *Proceedings of the 18th ACM International Symposium on Nanoscale Architectures*. New York, NY, USA: Association for Computing Machinery, 2024.

[84] R. Ali, D. M. Zhang, H. Cai, W. S. Zhao, and Y. Wang, "A machine learning attack-resilient strong puf leveraging the process variation of mram," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp. 2712–2716, 2022.

[85] A. Kamal, V. Mishra, S. Mittal, M. Rathor, C. Kumar, and U. Chatterjee, "Sorting attacks resilient authentication protocol for cmos image sensor based puf," in *Asian Hardware Oriented Security and Trust Symposium*, 2024, pp. 1–6.

[86] R. Kumar and W. Burleson, "On design of a highly secure puf based on non-linear current mirrors," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2014, pp. 38–43.

[87] J. H. Liu, Y. Zhu, C. H. Chan, and R. P. Martins, "An entropy-source-preselection-based strong puf with strong resilience to machine learning attacks and high energy efficiency," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 12, pp. 5108–5120, 2022.

[88] R. A. John, N. Shah, S. K. Vishwanath, S. E. Ng, B. Febriansyah, M. Jagadeeswararao, C.-H. Chang, A. Basu, and N. Mathews, "Halide perovskite memristors as flexible and reconfigurable physical unclonable functions," *Nature Communications*, vol. 12, no. 1, p. 3681, 2021.

[89] B. Narayanapuram and J. Panda, "Reconfigurable low weight hybrid puf design against side channel attacks," in *4th International Informatics and Software Engineering Conference*, 2023, pp. 1–5.

[90] Y. Wang, X. D. Xi, and M. Orshansky, "Lattice puf: A strong physical unclonable function provably secure against machine learning attacks," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 273–283, 2020.

[91] H. M. Ibrahim, H. Skovorodnikov, and H. Alkhzaimi, "Resilience evaluation of memristor based puf against machine learning attacks," *Scientific Reports*, vol. 14, no. 1, 2024.

[92] M. Elshamy and H.-G. Stratigopoulos, "Neuron-puf: Physical unclonable function based on a single spiking neuron," in *IEEE 27th International Symposium on On-Line Testing and Robust System Design*, 2021, pp. 1–6.

[93] Y. Zhang, Z. Q. He, M. L. Wan, J. Y. Liu, H. S. Gu, and X. C. Zou, "A sc puf standard cell used for key generation and anti-invasive-attack protection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3958–3973, 2021.

[94] H. Y. Zhuang, X. D. Xi, N. Sun, and M. Orshansky, "A strong subthreshold current array puf resilient to machine learning attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 1, pp. 135–144, 2020.

[95] K. Liu, Y. Tang, S. Xu, R. Zhang, and H. Shinohara, "A 100-bit-output modeling attack-resistant spn strong puf with uniform and high-randomness response," in *IEEE Custom Integrated Circuits Conference*, 2023, pp. 1–2.

[96] M. J. Adel, M. H. Rezayati, M. H. Moaiyeri, A. Amirany, and K. Jafari, "A robust deep learning attack immune mram-based physical unclonable function," *Scientific Reports*, vol. 14, no. 1, 2024.

[97] Z. Z. Chen, T. Sato, and H. Shinohara, "Spongepuf: A modeling attack resilient strong puf with scalable challenge response pair," *IEEE International Symposium on Hardware Oriented Security and Trust*, p. 244–253, 2024.

[98] A. Vijayakumar and S. Kundu, "A novel modeling attack resistant puf design based on non-linear voltage transfer characteristics," in *Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 653–658.

[99] R. J. Hui, F. L. Chen, M. Li, and J. Zhang, "Non-linear optical scattering puf: enhancing security against modeling attacks for authentication systems," *Optics Express*, vol. 31, no. 24, pp. 40 646–40 657, 2023.

[100] K. Phalak, A. A. . Saki, M. Alam, R. O. Topaloglu, and S. Ghosh, "Quantum puf for security and trust in quantum computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 333–342, 2021.

[101] M. P. Ma, Z. T. Jiang, T. J. Ma, X. X. Gao, J. Li, M. H. Liu, J. C. Yan, and X. S. Jiang, "Robust puf label authentication system synergistically constructed by hierarchical pattern of self-assembled phase-separation encrypted wrinkle and deep learning model," *Advanced Functional Materials*, vol. 34, no. 44, 2024.

[102] J. H. Liu, Y. Z. Zhao, Y. Zhu, C. H. Chan, and R. P. Martins, "A weak puf-assisted strong puf with inherent immunity to modeling attacks and ultra-low ber," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 12, pp. 4898–4907, 2022.

[103] T. Umeda, Y. Nozaki, and M. Yoshikawa, "Scalability and performance evaluation of ga based modeling analysis for ro puf," in *IEEE 8th Global Conference on Consumer Electronics*, 2019, pp. 1133–1134.

[104] F. Amsaad, M. Niamat, A. Dawoud, and S. Kose, "Reliable delay based algorithm to boost puf security against modeling attacks," *Information*, vol. 9, no. 9, 2018.

[105] A. Maiti, I. Kim, and P. Schaumont, "A robust unclonable function with enhanced challenge-response set," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 333–345, 2012.

[106] P. Wortman, W. Yan, J. Chandy, and F. Tehranipoor, "P2m-based security model: security enhancement using combined puf and prng models for authenticating consumer electronic devices," *IET Computers and Digital Techniques*, vol. 12, no. 6, pp. 289–296, 2018.

[107] W. L. Che, M. Martinez-Ramon, F. Saqib, and J. Plusquellic, "Delay model and machine learning exploration of a hardware-embedded delay puf," *Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 153–158, 2018.

[108] C. Y. Gu, C. H. Chang, W. Q. Liu, S. C. Yu, Y. L. Wang, and M. O'Neill, "A modeling attack resistant deception technique for securing lightweight-puf-based authentication," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 6, pp. 1183–1196, 2021.

[109] G. T. Becker, "On the pitfalls of using arbiter-pufs as building blocks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1295–1307, 2015.

[110] H. Lin, H. Zuo, Q. Peng, and X. Zhao, "A 690fj/bit ml-attack-resilient strong puf based on subthreshold voltage attenuator ring with closed-loop feedback," in *IEEE 49th European Solid State Circuits Conference*, 2023, pp. 113–116.

[111] Z. He, M. Wan, J. Deng, C. Bai, and K. Dai, "A reliable strong puf based on switched-capacitor circuit," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 26, no. 6, pp. 1073–1083, 2018.

[112] H. Kang, Y. Hori, T. Katashita, A. Satoh, and K. Iwamura, "Puf evaluation with post-processing and modified modeling attack," *International Journal of Security and Its Applications*, vol. 7, no. 4, pp. 231–242, 2013.

[113] L. Yu, X. Wang, F. Rahman, and M. Tehranipoor, "Interconnect-based puf with signature uniqueness enhancement," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 28, no. 2, pp. 339–352, 2020.

[114] A. Roelke and M. R. Stan, "Attacking an sram-based puf through wearout," *IEEE Computer Society Annual Symposium on VLSI*, pp. 206–211, 2016.

[115] S. B. Dodo, R. Bishnoi, S. M. Nair, and M. B. Tahoori, "A spintronics memory puf for resilience against cloning counterfeit," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 27, no. 11, pp. 2511–2522, 2019.

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

[117] Y. Yao, M. Gao, M. Li, and J. Zhang, "Quantum cloning attacks against puf-based quantum authentication systems," *Quantum Information Processing*, vol. 15, no. 8, pp. 3311–3325, 2016.

[118] B. Skoric, "Quantum readout of physical unclonable functions," *International Journal of Quantum Information*, vol. 10, no. 01, p. 1250001, 2012.

[119] J. Ye, Y. Hu, and X. Li, "Vpuf: Voter based physical unclonable function with high reliability and modeling attack resistance," in *IEEE 23rd International Symposium on On-Line Testing and Robust System Design*, 2017, pp. 74–79.

[120] Z. T. Chang, S. S. Shi, B. W. Song, W. B. Fan, and Y. Wang, "Modeling attack resistant arbiter puf with time-variant obfuscation scheme," *31st International Conference on Field-Programmable Logic and Applications*, pp. 60–63, 2021.

[121] A. Oun and M. Niamat, "Design of a delay-based fpga puf resistant to machine learning attacks," in *IEEE International Midwest Symposium on Circuits and Systems*, 2021, pp. 865–868.

[122] J. Ye, Y. Gong, Y. Hu, and X. Li, "Polymorphic puf: Exploiting reconfigurability of cpu+fpga soc to resist modeling attack," in *Asian Hardware Oriented Security and Trust Symposium*, 2017, pp. 43–48.

[123] Y. Nozaki and M. Yoshikawa, "Tamper resistance evaluation of puf implementation against machine learning attack," in *Proceedings of the 2017 International Conference on Biometrics Engineering and Application*. New York, NY, USA: Association for Computing Machinery, 2017, p. 1–6.

[124] A. M. Venkata, D. R. Jeeru, and V. K P, "Design and modelling an attack on multiplexer based physical unclonable function," *International Journal of Engineering Trends and Technology*, vol. 68, no. 6, p. 63–67, 2020.

[125] H. B. Su, M. Zwolinski, and B. Halak, "A machine learning attacks resistant two stage physical unclonable functions design," *IEEE 3rd International Verification and Security Workshop*, no. 3, pp. 52–55, 2018.

[126] K. Liu, G. Li, Z. Fu, X. Z. Wang, and H. Shinohara, "A 2.17-pj/b 5b-response attack-resistant strong puf with enhanced statistical performance," *IEEE 48th European Solid State Circuits Conference*, no. 48, pp. 513–516, 2022.

[127] A. Gupta, S. Manhas, and B. P. Das, "Highly non-linear feed-forward arbiter puf against machine learning attacks," in *VLSI Design and Test*, A. P. Shah, S. Dasgupta, A. Darji, and J. Tudu, Eds. Cham: Springer Nature Switzerland, 2022, pp. 234–248.

[128] P. Santikellur, A. Bhattacharyay, and R. S. Chakraborty, "Deep learning based model building attacks on arbiter puf compositions," Cryptology ePrint Archive, Paper 2019/566, 2019. [Online]. Available: https://eprint.iacr.org/2019/566

[129] R. Yashiro, T. Machida, M. Iwamoto, and K. Sakiyama, "Deep-learning-based security evaluation on authentication systems using arbiter puf and its variants," in *Advances in Information and Computer Security*, K. Ogawa and K. Yoshioka, Eds. Cham: Springer International Publishing, 2016, pp. 267–285.

[130] R. R. Adhithan and N. N. Anandakumar, "Modeling attacks and efficient countermeasures on interpose puf," *Foundations and Practice of Security*, vol. 12637, no. 13, pp. 149–162, 2021.

[131] Y. Nozaki and M. Yoshikawa, "Countermeasure of lightweight physical unclonable function against side-channel attack," in *Cybersecurity and Cyberforensics Conference*, 2019, pp. 30–34.

[132] J. Shi, Y. Lu, and J. Zhang, "Approximation attacks on strong pufs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2138–2151, 2020.

[133] E. Dubrova, O. Näslund, B. Degen, A. Gawell, and Y. Yu, "Crc-puf: A machine learning attack resistant lightweight puf construction," *4th IEEE European Symposium on Security and Privacy Workshops*, no. 4, pp. 264–271, 2019.

[134] S. Shi, Z. Chang, B. Guo, and Y. Wang, "Modeling attack resistant arbiter puf based on dynamic finite field matrix multiplication scheme," in *IEEE 16th International Conference on Solid-State Integrated Circuit Technology*, 2022, pp. 1–3.

[135] Y. Gao, H. Ma, S. F. Al-Sarawi, D. Abbott, and D. C. Ranasinghe, "Puf-fsm: A controlled strong puf," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 5, p. 1, 2018.

[136] K. Y. Liu, K. Takeuchi, and H. Shinohara, "Statistical modeling of sram puf cell mismatch shift distribution after hot carrier injection burn-in," *IEEE 34th International Conference on Microelectronic Test Structures*, no. 34, pp. 93–96, 2022.

[137] A. Pugazhenthi, N. Karimian, and F. Tehranipoor, "Dla-puf: Deep learning attacks on hardware security primitives," *Autonomous Systems: Sensors, Processing*, vol. 11009, 2019.

[138] S. Sutar, A. Raha, and V. Raghunathan, "D-puf: an intrinsically reconfigurable dram puf for device authentication in embedded systems," in *Proceedings of the International Conference on Compilers*, ser. CASES '16. New York, NY, USA: Association for Computing Machinery, 2016.

[139] R. Ali, H. Ma, Z. Hou, D. Zhang, E. Deng, and Y. Wang, "A reconfigurable arbiter mpuf with high resistance against machine learning attack," *IEEE Transactions on Magnetics*, vol. 57, no. 10, pp. 1–7, 2021.

[140] D. Schuster and R. Hesselbarth, "Evaluation of bistable ring pufs using single layer neural networks," in *Trust and Trustworthy Computing*, T. Holz and S. Ioannidis, Eds. Cham: Springer International Publishing, 2014, pp. 101–109.

[141] X. Xu, U. Rührmair, D. E. Holcomb, and W. Burleson, "Security evaluation and enhancement of bistable ring pufs," in *Radio Frequency Identification*, S. Mangard and P. Schaumont, Eds. Cham: Springer International Publishing, 2015, pp. 3–16.

[142] J. Han and C. Shin, "An extensive puf of bistable rings feed-forward chains with lightweight secure architecture for enhanced ml attack resistance," *Journal of Semiconductor Technology and Science*, vol. 24, no. 2, p. 76–83, 2024.

[143] M. L. Wan, Z. Zhang, Y. Zhang, Z. Q. He, H. S. Gu, K. Dai, and X. C. Zou, "A chip-pcb hybrid sc puf used for anti-desoldering and depackaging-attack protection," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 7, p. 2330–2344, 2024.

[144] J. Y. Zhang, C. Y. Xu, M. K. Law, Y. Jiang, X. J. Zhao, P. I. Mak, and R. P. Martins, "A 4t/cell amplifier-chain-based xor puf with strong machine learning attack resilience," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 1, pp. 366–377, 2022.

[145] X. H. Yang, S. Khandelwal, A. Q. Jiang, and A. Jabir, "A modelling attack resistant low overhead memristive physical unclonable function," *33rd IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, no. 33, 2020.

[146] H. B. Zuo, J. C. Hao, H. T. Lin, X. J. Zhao, Y. T. Yang, and L. Huang, "A 3.02 pj/bit 3t-aps-based in-sensor strong puf featuring near-100% hardware reuse ratio and high resilience to machine learning attacks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 11, pp. 4206–4210, 2023.

[147] Y. Tanaka, S. Bian, M. Hiromoto, and T. Sato, "Coin flipping puf: A novel puf with improved resistance against machine learning attacks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 5, pp. 602–606, 2018.

[148] E. Elmitwalli, K. Ni, and S. Köse, "Machine learning attack resistant area-efficient reconfigurable ising-puf," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 30, no. 4, pp. 526–538, 2022.

[149] Y. J. Cui, C. Y. Gu, Q. Q. Ma, Y. Fang, C. H. Wang, M. O'Neill, and W. Q. Liu, "Lightweight modeling attack-resistant multiplexer-based multi-puf (mmpuf) design on fpga," *Electronics*, vol. 9, no. 5, 2020.

[150] X. J. Ma, P. J. Wang, G. Li, and Z. Y. Zhou, "Machine learning attacks resistant strong puf design utilizing response obfuscates challenge with lower hardware overhead," *Microelectronics Journal*, vol. 142, 2023.

[151] V. S. Balijabudda, D. Thapar, P. Santikellur, R. S. Chakraborty, and I. Chakrabarti, "Design of a chaotic oscillator based model building attack resistant arbiter puf," in *Asian Hardware Oriented Security and Trust Symposium*, 2019, pp. 1–6.

[152] N. Pundir, N. A. Hazari, F. Amsaad, and M. Niamat, "A novel hybrid delay based physical unclonable function immune to machine learning attacks," in *IEEE National Aerospace and Electronics Conference*, 2017, pp. 84–87.

[153] W. Ge, S. Hu, J. Huang, B. Liu, and M. Zhu, "Fpga implementation of a challenge pre-processing structure arbiter puf designed for machine learning attack resistance," *IEICE Electronics Express*, vol. 17, no. 2, pp. 1–6, 2020.

[154] J. Zhang, L. Wan, Q. Wu, and G. Qu, "Dmos-puf: Dynamic multi-key-selection obfuscation for strong pufs against machine learning attacks," *Arxiv Preprint Arxiv:1806.02011*, 2018.

[155] M.-D. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede, "A lockdown technique to prevent machine learning on pufs for lightweight authentication," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 3, pp. 146–159, 2016.

[156] Z. Zhou, G. Li, P. Wang, and M. Ye, "Matrix encryption based anti-machine learning attack algorithm for strong puf," in *IEEE 14th International Conference on ASIC*, 2021, pp. 1–4.

[157] L. Dai, Q. Yan, S. Yi, W. Liu, and H. Qian, "A novel rram based puf for anti-machine learning attack and high reliability," *Journal of Shanghai Jiaotong University*, vol. 24, no. 1, pp. 101–106, 2019.

[158] J. C. Bian, Z. F. Huang, R. X. Liu, Y. K. Lin, Z. Yang, H. G. Liang, and A. B. Yan, "A ro-integrated-lfsr-based nonlinear strong puf with intrinsic modeling attacks resilience," *8th International Test Conference in Asia*, 2024.

[159] S. Alahmadi, K. Khalil, H. Idriss, and M. Bayoumi, "Fortifying strong pufs: A modeling attack-resilient approach using weak puf for iot device security," 2024.

[160] V. R. Laguduva, S. Katkoori, and R. Karam, "Machine learning attacks and countermeasures for puf-based iot edge node security," no. 5, 2020.

[161] M.-D. Yu, D. M'Raïhi, I. Verbauwhede, and S. Devadas, "A noise bifurcation architecture for linear additive physical functions," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2014, pp. 124–129.

[162] C. C. Lin and M. S. Chen, "Enhancing reliability and security: A configurable poisoning puf against modeling attacks," *IEEE Transactions*

on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 11, pp. 4301–4312, 2022.

[163] Y. M. Wen, S. F. Ahamed, and W. Z. Yu, "A novel puf architecture against non-invasive attacks," *ACM/IEEE International Workshop on System Level Interconnect Prediction*, 2019.

[164] C. C. Lin and M. S. Chen, "Attack is the best defense: A multi-mode poisoning puf against machine learning attacks," *Advances in Knowledge Discovery and Data Mining*, vol. 12712, pp. 176–187, 2021.

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

[166] H. L. França, C. B. Prado, V. C. Patil, and S. Kundu, "Defeating strong puf modeling attack via adverse selection of challenge-response pairs," *Proceedings of the Asian Hardware Oriented Security and Trust Symposium*, no. 3, pp. 25–30, 2018.

[167] K. Fukushima, Y. Souissi, S. Hidano, R. Nguyen, J.-L. Danger, S. Guilley, Y. Nakano, S. Kiyomoto, and L. Sauvage, "Delay puf assessment method based on side-channel and modeling analyzes: The final piece of all-in-one assessment methodology," in *IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 201–207.

[168] A. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar, "Combined modeling and side channel attacks on strong pufs," Cryptology ePrint Archive, Paper 2013/632, 2013. [Online]. Available: https://eprint.iacr.org/2013/632

[169] Y. Cao, X. J. Zhao, W. B. Ye, Q. B. Han, and X. F. Pan, "A compact and low power ro puf with high resilience to the em side-channel attack and the svm modelling attack of wireless sensor networks," *Sensors*, vol. 18, no. 2, 2018.

[170] Z. J. Lu, D. F. Li, H. L. Liu, M. Y. Gong, and Z. L. Liu, "An anti-electromagnetic attack puf based on a configurable ring oscillator for wireless sensor networks," *Sensors*, vol. 17, no. 9, 2017.

[171] M. Shiozaki, T. Kubota, T. Nakai, A. Takeuchi, T. Nishimura, and T. Fujino, "Tamper-resistant authentication system with side-channel attack resistant aes and puf using mdr-rom," in *IEEE International Symposium on Circuits and Systems*, 2015, pp. 1462–1465.

[172] M. Xue, J. Wang, Y. Wang, and A. Hu, "Security against hardware trojan attacks through a novel chaos fsm and delay chains array puf based design obfuscation scheme," in *Cloud Computing and Security*, Z. Huang, X. Sun, J. Luo, and J. Wang, Eds. Cham: Springer International Publishing, 2015, pp. 14–24.

[173] T. Kroeger, W. Cheng, S. Guilley, J.-L. Danger, and N. Karimi, "Effect of aging on puf modeling attacks based on power side-channel observations," in *Design, Automation & Test in Europe Conference & Exhibition*, 2020, pp. 454–459.

[174] V. L. Ramnath, S. N. Aakur, and S. Katkoori, "Latent space modeling for cloning encrypted puf-based authentication," in *Internet of Things: A Confluence of Many Disciplines*, A. Casaca, S. Katkoori, S. Ray, and L. Strous, Eds. Cham: Springer International Publishing, 2020, pp. 142–158.

[175] J. Tobisch and G. T. Becker, "On the scaling of machine learning attacks on pufs with application to noise bifurcation," 2015.

[176] Y. Zhuang, G. Li, and K. T. Mursi, "A permutation challenge input interface for arbiter puf variants against machine learning attacks," *IEEE Computer Society Annual Symposium on VLSI*, pp. 418–421, 2022.

[177] M. Ebrahimabadi, W. Lalouani, M. Younis, and N. Karimi, "Countering puf modeling attacks through adversarial machine learning," *IEEE Computer Society Annual Symposium on VLSI*, pp. 356–361, 2021.

[178] S.-J. Wang, Y.-S. Chen, and K. S.-M. Li, "Adversarial attack against modeling attack on pufs," in *56th ACM/IEEE Design Automation Conference*, 2019, pp. 1–6.

[179] M. Rostami, M. Majzoobi, F. Koushanfar, D. S. Wallach, and S. Devadas, "Robust and reverse-engineering resilient puf authentication and key-exchange by substring matching," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 1, pp. 37–49, 2014.

[180] M. Ebrahimabadi, M. Younis, and N. Karimi, "A puf-based modeling-attack resilient authentication protocol for iot devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3684–3703, 2022.

[181] M. H. B. Ishak, M. S. Mispan, Y. C. Wong, M. R. Kamaruddin, and M. Korobkov, "Fpga-based obfuscated delay puf for security enhancement against ml-attack," *6th IEEE International Conference on Recent Advances and Innovations in Engineering*, no. 6, 2021.

[182] W. Lalouani, M. Younis, M. Ebrahimabadi, and N. Karimi, "Countering modeling attacks in puf-based iot security solutions," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 3, 2022.

[183] M. Ebrahimabadi, M. Younis, W. Lalouani, and N. Karimi, "An attack resilient puf-based authentication mechanism for distributed systems," *35th International Conference on VLSI Design held Concurrently with 21st International Conference on Embedded Systems*, no. 35, pp. 108–113, 2022.

[184] R. Yashiro, Y. Hori, T. Katashita, and K. Sakiyama, "A deep learning attack countermeasure with intentional noise for a puf-based authentication scheme," in *Innovative Security Solutions for Information Technology and Communications*, E. Simion and R. Géraud-Stewart, Eds. Cham: Springer International Publishing, 2020, pp. 78–94.

[185] M. S. E. Quadir and J. A. Chandy, "Embedded systems authentication and encryption using strong puf modeling," vol. 2020-January, University of Connecticut, Electrical and Computer Engineering, Storrs, 06269, CT, United States. Institute of Electrical and Electronics Engineers Inc, 2020.

[186] N. P. Bhatta and F. Amsaad, "Advancing puf security machine learning assisted modeling attacks," *IEEE Computer Society Annual Symposium on VLSI*, p. 805–808, 2024.

[187] X. Zheng, S. Xie, C. Xie, and W. Zhu, "An rfid lightweight authentication technology based on puf-rfid structure model," *Blockchain and Trustworthy Systems*, vol. 1156, no. 1, p. 305, 2020.

[188] S. Duan and C. Cao, "Differential emulatable puf design against machine learning attacks for hardware security," in *3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering*, 2023, pp. 1–6.

[189] W. Xu, L. Pang, Y. Tang, and M. Chen, "Security evaluation of feed-forward interpose puf against modelling attacks," in *IEEE 4th International Conference on Power, Electronics and Computer Applications*, 2024, pp. 871–877.

[190] B. Chatterjee, D. Das, and S. Sen, "Rf-puf: Iot security enhancement through authentication of wireless nodes using in-situ machine learning," in *IEEE International Symposium on Hardware Oriented Security and Trust*, 2018, pp. 205–208.

[191] S. Zhou, Y. Chen, X. Cui, and Y. Liu, "Modeling attack tests and security enhancement of the sub-threshold voltage divider array puf," in *Design, Automation & Test in Europe Conference & Exhibition*, 2024, pp. 1–6.

[192] J. Miskelly, C. Y. Gu, Q. Q. Ma, Y. J. Cui, W. Q. Liu, and M. O'Neill, "Modelling attack analysis of configurable ring oscillator (cro) puf designs," *IEEE 23rd International Conference on Digital Signal Processing*, no. 23, 2018.

[193] M. Uddin, B. Majumder, and G. S. Rose, "Robustness analysis of a memristive crossbar puf against modeling attacks," *IEEE Transactions on Nanotechnology*, vol. 16, no. 3, pp. 396–405, 2017.

[194] Y. Nozaki and M. Yoshikawa, "Power consumption aware machine learning attack for feed-forward arbiter puf," *Computer and Information Science*, vol. 791, no. 17, pp. 49–62, 2019.

[195] Z. Hou, Y. Wang, D. Zhang, C. Wang, and H. Cai, "A modeling attack resilient physical unclonable function based on stt-mram," in *Proceedings of the Great Lakes Symposium on VLSI*. New York, NY, USA: Association for Computing Machinery, 2020, p. 65–70.

[196] J. Delvaux, "Machine-learning attacks on polypufs, ob-pufs, rpufs, lhs-pufs, and puf-fsms," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2043–2058, 2019.

[197] M. Hiromoto, M. Yoshinaga, and T. Sato, "Mro-puf: Physically unclonable function with enhanced resistance against machine learning attacks utilizing instantaneous output of ring oscillator," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E101A, no. 7, pp. 1035–1044, 2018.

[198] W. Yu and Y. Wen, "Malicious attacks on physical unclonable function sensors of internet of things," in *IEEE 28th North Atlantic Test Workshop*, 2019, pp. 206–211.

[199] M. Ferens, E. Dushku, and S. Kosta, "On the feasibility of deep reinforcement learning for modeling delay-based pufs," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, p. 421–428, 2024.

[200] Q. L. Guo, J. Ye, Y. Gong, Y. Hu, and X. W. Li, "Efficient attack on non-linear current mirror puf with genetic algorithm," *IEEE 25th Asian Test Symposium*, no. 25, pp. 49–54, 2016.

[201] Y. J. Liu, G. F. Huang, J. W. Li, P. F. Guo, C. S. Zhu, and Z. B. Dai, "Ma-grnn: a high-efficient modeling attack approach utilizing generalized regression neural network for xor arbiter physical unclonable functions," *IEICE Electronics Express*, vol. 20, no. 13, 2023.

[202] Z. K. Peng, N. Y. Sun, J. F. Cheng, W. R. Liu, C. Y. Wang, Y. J. Bi, C. B. Sun, Y. F. Wang, Y. M. Wen, Y. B. Wang, and W. Z. Yu, "A sequential strong puf architecture based on reconfigurable neural

networks (rnns) against state-of-the-art modeling attacks," *Integration*, vol. 92, pp. 83–90, 2023.
[203] F. Ganji, S. Tajik, and J.-P. Seifert, "Pac learning of arbiter pufs," *Journal of Cryptographic Engineering*, vol. 6, no. 3, pp. 249–258, 2016.
[204] C. Helfmeier, C. Boit, D. Nedospasov, and J. P. Seifert, "Cloning physically unclonable functions," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, 2013, pp. 1–6.
[205] H. Yamada, S. Okura, M. Shirahata, and T. Fujino, "Modeling attacks against device authentication using cmos image sensor puf," *IEICE Electronics Express*, vol. 18, no. 7, 2021.
[206] Y. T. Liu, Y. Xie, C. X. Bao, and A. Srivastava, "A combined optimization-theoretic and side-channel approach for attacking strong physical unclonable functions," *IEEE Transactions on Very Large Scale Integration VLSI Systems*, vol. 26, no. 1, pp. 73–81, 2018.
[207] P. H. Nguyen, D. P. Sahoo, R. S. Chakraborty, and D. Mukhopadhyay, "Efficient attacks on robust ring oscillator puf with enhanced challenge-response set," in *Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 641–646.
[208] T. Xu, D. F. Li, and M. Potkonjak, "Adaptive characterization and emulation of delay-based physical unclonable functions using statistical models," *52nd ACM/EDAC/IEEE Design Automation Conference*, no. 52, 2015.
[209] D. Canaday, W. Barbosa, and A. Pomerance, "A novel attack on machine-learning resistant physical unclonable functions," *IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 25–28, 2022.
[210] Y. Nozaki and M. Yoshikawa, "Em based machine learning attack for xor arbiter puf," *2nd International Conference on Machine Learning and Soft Computing*, no. 2, pp. 19–23, 2015.
[211] J. Delvaux and I. Verbauwhede, "Key-recovery attacks on various ro puf constructions via helper data manipulation," in *Design, Automation & Test in Europe Conference & Exhibition*, 2014, pp. 1–6.
[212] G. T. Becker, "The gap between promise and reality: On the insecurity of xor arbiter pufs," in *Cryptographic Hardware and Embedded Systems*, T. G uneysu, H, and H. schuh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 535–555.
[213] B. Gassend, D. Lim, D. Clarke, M. van Dijk, and S. Devadas, "Identification and authentication of integrated circuits," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 11, pp. 1077–1098, 2004.
[214] D. E. Holcomb and K. Fu, "Bitline puf: Building native challenge-response puf capability into any sram," in *Cryptographic Hardware and Embedded Systems*, L. Batina and M. Robshaw, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 510–526.
[215] J. Ye, Y. Hu, and X. W. Li, "Poster: Attack on non-linear physical unclonable function," *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, no. 23, pp. 1751–1753, 2016.
[216] S. Kiryu, K. Asahi, and M. Yoshikawa, "Vulnerability evaluation of multiplexing puf for svm attacks," *Progress in Systems Engineering*, vol. 366, no. 23, pp. 205–210, 2015.
[217] M. A. Alamro and K. T. Mursi, "Machine learning attack on a multiplexer puf variant using silicon data: a case study on rmpufs," in *IEEE 6th International Conference on Computer and Communication Systems*, 2021, pp. 1017–1022.
[218] Y. Ikezaki, Y. Nozaki, and M. Yoshikawa, "Deep learning attack for physical unclonable function," *IEEE 5th Global Conference on Consumer Electronics*, no. 5, 2016.
[219] S. Kumar and M. Niamat, "Machine learning based modeling attacks on a configurable puf," vol. 2018-July, Electrical Engineering and Computer Science, University of Toledo, Toledo, OH, United States. Institute of Electrical and Electronics Engineers Inc, 2018, pp. 169–173.
[220] Q. Wang, O. Aramoon, P. F. Qiu, and G. Qu, "Efficient transfer learning on modeling physical unclonable functions," *Proceedings of the 21st International Symposium on Quality Electronic Design*, no. 21, pp. 1–6, 2020.
[221] Y. J. Wen and Y. J. Lao, "Puf modeling attack using active learning," *IEEE International Symposium on Circuits and Systems*, 2018.
[222] M. S. Alkatheiri and Y. Zhuang, "Towards fast and accurate machine learning attacks of feed-forward arbiter pufs," in *IEEE Conference on Dependable and Secure Computing*, 2017, pp. 181–187.
[223] A. O. Aseeri, "Noise-resilient neural network-based adversarial attack modeling for xor physical unclonable functions," *Journal of Cyber Security and Mobility*, vol. 9, no. 2, pp. 331–354, 2020.
[224] N. Wisiol, C. Mühl, N. Pirnay, P. H. Nguyen, M. Margraf, J. P. Seifert, M. van Dijk, and U. Rührmair, "Splitting the interpose puf: A novel modeling attack strategy," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 3, pp. 97–120, 2020.
[225] S. Matsumi, Y. Nozaki, and M. Yoshikawa, "Feature extraction driven modeling attack against double arbiter puf and its evaluation," *Proceedings of Artificial Intelligence and Cloud Computing Conference*, pp. 94–99, 2018.
[226] A. Aghaie and A. Moradi, "Inconsistency of simulation and practice in delay-based strong pufs," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, no. 3, pp. 520–551, 2021.
[227] Z. J. Jin, S. Y. Chen, and L. M. Yan, "Low-overhead xor multi-puf against machine learning attacks," *Microelectronics Journal*, vol. 141, 2023.
[228] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A taxonomy of puf schemes with a novel arbiter-based puf resisting machine learning attacks," *Computer Networks*, vol. 194, 2021.
[229] R. C. Surita, M. L. Côrtes, D. F. Aranha, and G. Araujo, "Crpuf: A modeling-resistant delay puf based on cylindrical reconvergence," *Microprocessors and Microsystems*, vol. 60, pp. 185–195, 2018.
[230] S. Y. Chen and L. M. Yan, "A low-overhead puf for anti-clone attack of rfid tags," *Microelectronics Journal*, vol. 126, 2022.
[231] Y. H. Xu, Y. J. Lao, W. Q. Liu, Z. C. Zhang, X. H. You, and C. Zhang, "Mathematical modeling analysis of strong physical unclonable functions," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4426–4438, 2020.
[232] S. Tripathy, V. K. Rai, and J. Mathew, "Marpuf: physical unclonable function with improved machine learning attack resistance," *IET Circuits*, vol. 15, no. 5, pp. 465–474, 2021.
[233] A. Rajan and S. Sankaran, "Lightweight and attack-resilient puf for internet of things," *IEEE 6th International Symposium on Smart Electronic Systems*, pp. 139–142, 2020.
[234] J. Ye, Q. Guo, Y. Hu, H. Li, and X. Li, "Modeling attacks on strong physical unclonable functions strengthened by random number and weak puf," in *IEEE 36th VLSI Test Symposium*, 2018, pp. 1–6.
[235] S. S. Mansouri and E. Dubrova, "Protecting ring oscillator physical unclonable functions against modeling attacks," *Information Security and Cryptology*, vol. 8565, no. 16, pp. 241–255, 2014.
[236] Y. L. Chen, X. L. Cui, W. Q. Ye, and X. X. Cui, "The security enhancement techniques of the double-layer puf against the ann-based modeling attack," *IEEE International Test Conference*, pp. 63–72, 2021.
[237] M. Kojage, N. Hassan, and U. Chatterjee, "Machine learning attacks on low-cost reconfigurable xrro and xrbr puf designs," *Security, Privacy, and Applied Cryptography Engineering*, vol. 13783, no. 12, pp. 204–224, 2022.
[238] Y. Liu, Y. L. Chen, and X. L. Cui, "A modeling attack on the sub-threshold current array puf," *IEEE International Symposium on Hardware Oriented Security and Trust*, pp. 169–172, 2022.
[239] Y. Nozaki and M. Yoshikawa, "Security evaluation of ring oscillator puf against genetic algorithm based modeling attack," *Innovative Mobile and Internet Services in Ubiquitous Computing*, vol. 994, no. 13, pp. 338–347, 2020.

APPENDIX A
CLASSIFICATION OF PUF IMPLEMENTATIONS

TABLE V: Full overview of PUF implementations.

Category	PUF	Year	Ref ID	Type	Platform	Reliability	Uniformity	Uniqueness	Countermeasure
Delay (186)	Arbiter PUF (23)	2024	[186]	Strong	Simulated				-
		2024	[199]	Strong	Simulation				-
		2022	[28]	Strong	Xilinx Spartan-6	98.42%		51.19%	-
		2022	[28]	Controlled	Xilinx Spartan-6	98.42%		51.19%	Response obf.
		2022	[4]	Strong	Simulated				-
		2020	[124]	Strong	Xilinx Artix-7				-
		2020	[173]	Strong	Simulated				-
		2020	[173]	Strong	Simulated				Side-channel obf.
		2019	[128]	Strong	Simulated		51.04%	50.19%	-
		2018	[206]	Strong	Simulated				-
		2018	[166]	Strong	Simulated				Data poisoning
		2018	[221]	Strong	Simulated	96.5%			-
		2017	[123]	Strong	Xilinx Virtex-5				-
		2017	[123]	Strong	Xilinx Virtex-5				-
		2016	[218]	Strong	Simulated				-
		2016	[167]	Strong	65nm CMOS		65%		-
		2016	[167]	Controlled	65nm CMOS		65%		Response obf.
		2016	[203]	Strong					-
		2015	[208]	Strong	Xilinx Spartan-6				-
		2015	[208]	Strong	Xilinx Spartan-6				-
		2013	[31]	Strong	Xilinx Spartan-6	95.13%			-
		2013	[31]	Strong	45nm CMOS	96.82%			-
		2013	[112]	Strong	Xilinx Spartan-3A	97.11%		∼50%	-
	XOR PUF (20)	2024	[199]	Controlled	Simulation				Response obf.
		2023	[201]	Controlled	Xilinx Artix-7				Response obf.
		2022	[36]	Controlled	Simulated				Response obf.
		2022	[32]	Controlled	Simulated				Response obf.
		2021	[75]	Controlled	Simulated		49.23%	53.21%	Response obf.
		2021	[226]	Controlled	Simulated		50.31%		Response obf.
		2020	[220]	Controlled	Simulated				Response obf.
		2020	[220]	Controlled	Simulated				Response obf.
		2020	[223]	Controlled	Simulated				Response obf.
		2020	[29]	Controlled	Xilinx Artix-7		49.93%		Response obf.
		2019	[128]	Controlled	Simulated		50%	49.98%	Response obf.
		2018	[206]	Controlled	Simulated				Response obf.
		2018	[5]	Controlled	Simulated				Response obf.
		2016	[129]	Controlled	Xilinx Virtex-5				Response obf.
		2015	[212]	Controlled	Simulated	76.0%			Response obf.
		2015	[175]	Controlled	Simulated				Response obf.
		2015	[210]	Controlled	Xilinx Spartan-6				Response obf.
		2013	[31]	Controlled	Xilinx Spartan-6				Response obf.
		2013	[31]	Controlled	45nm CMOS				Response obf.
		2013	[168]	Controlled	Simulation				Response obf.
	RO PUF (11)	2021	[67]	Weak	180nm CMOS	99.95%		∼50%	-
		2020	[239]	Weak	180nm CMOS				-
		2019	[198]	Weak	Simulated				-
		2019	[103]	Weak	Simulated				-
		2018	[169]	Controlled	65nm CMOS	98.30%		50.17%	Challenge obf.
		2018	[104]	Strong	Xilinx Spartan-3E				-
		2015	[207]	Controlled	Xilinx Spartan-3E				Response obf.
		2014	[211]	Weak	Mixed				-
		2014	[235]	Weak					Data poisoning
		2013	[31]	Weak	Simulated				-
		2012	[105]	Controlled	Xilinx Spartan-3E	∼90%	50.02%	49.99%	Response obf.
	IPUF (10)	2024	[199]	Controlled	Simulation				Challenge obf.
		2021	[130]	Controlled	Xilinx Artix-7				Challenge obf.
		2021	[226]	Controlled	Xilinx Artix-7		46%		Challenge obf.
		2021	[34]	Controlled	Xilinx Artix-7				Challenge obf.
		2021	[34]	Controlled	Xilinx Artix-7				Challenge obf.
		2020	[224]	Controlled	Simulated				Challenge obf.
		2020	[224]	Controlled	Simulated				Challenge obf.
		2019	[56]	Controlled	Xilinx Artix-7				Challenge obf.
		2019	[56]	Controlled	Xilinx Artix-7	97.9%	25%		Challenge obf.
		2019	[128]	Controlled	Simulated		50.50%	50.92%	Challenge obf.
	LSPUF (6)	2021	[75]	Controlled	Simulated				Response obf.
		2019	[131]	Controlled	Xilinx Virtex-5				Side-channel obf.
		2019	[128]	Controlled	Simulated		50.62%	51.01%	Response obf.
		2015	[61]	Controlled	Xilinx Artix-7	97.08%	49.50%		Response obf.
		2013	[31]	Controlled	Simulated				Response obf.
		2013	[168]	Controlled	Simulation				Response obf.
	DAPUF (4)	2019	[35]	Controlled	Xilinx Spartan-6		40.2%		Response obf.

TABLE V: Full overview of PUF implementations (continued).

Delay (186)

DAPUF (4)	2018	[225]	Controlled	Xilinx Virtex-5				Response obf.
	2016	[129]	Controlled	Xilinx Virtex-5				Response obf.
	2015	[49]	Controlled	Xilinx Virtex-5	∼90.31%	∼45.74%	∼46.37%	Response obf.
FF-PUF (4)	2023	[53]	Controlled	Xilinx Artix-7				Challenge obf.
	2019	[194]	Controlled	Xilinx Virtex-5				Challenge obf.
	2017	[222]	Controlled	Simulated				Challenge obf.
	2013	[31]	Controlled	Simulated				Challenge obf.
MMPUF (4)	2022	[209]	Controlled	Simulated				Challenge obf.
	2020	[149]	Controlled	Simulated				Challenge obf.
	2020	[149]	Controlled	Xilinx Artix-7	94%		40.10%	-
	2020	[149]	Controlled	Xilinx Kintex-7	96.65%		47.30%	-
XOR FF-PUF (4)	2022	[127]	Controlled	Xilinx Zynq-7000				Full obf.
	2022	[32]	Controlled	Xilinx Artix-7				Full obf.
	2020	[165]	Controlled	Xilinx Artix-7	∼93%		∼50%	Full obf.
	2020	[165]	Controlled	Xilinx Artix-7	∼93%		∼50%	Full obf.
CRO PUF (3)	2018	[219]	Strong	Xilinx Spartan-3E				-
	2018	[192]	Strong	Simulated				-
	2017	[170]	Controlled	Xilinx Artix-7	89%	∼50%	48.85%	Side-channel obf.
LP-PUF (3)	2023	[60]	Controlled	Simulated	61-96%		∼99%	Full obf.
	2023	[60]	Controlled	Simulated				Full obf.
	2022	[209]	Controlled	Simulated				Full obf.
MPUF (3)	2020	[132]	Controlled	Simulated				Challenge obf.
	2019	[128]	Controlled	Simulated		50.05%	54.20%	Challenge obf.
	2018	[62]	Controlled	Simulated	98.67%	49.80%	50.01%	Challenge obf.
DCH PUF (2)	2022	[50]	Controlled	Xilinx Artix-7	98.68%		41%	Response obf.
	2022	[50]	Controlled					Full obf.
FF-IPUF (2)	2024	[189]	Controlled	Simulated	78.60%	50.11%	49.70%	Challenge obf.
	2024	[189]	Controlled	Simulated	78.60%	50.11%	49.70%	Challenge obf.
Ising PUF (2)	2022	[148]	Controlled	Simulated	99.81%	∼50%	50%	Challenge obf.
	2018	[57]	Controlled	Simulated	97.74%		50.1%	Challenge obf.
NoPUF (2)	2021	[63]	Controlled	Simulated	91.98%		49.65%	Challenge obf.
	2021	[63]	Controlled	Simulated	56.75%		49.65%	Data poisoning
OI-PUF (2)	2023	[64]	Controlled	Xilinx Zynq-7000	99.2%	48.9%	30%	Challenge obf.
	2023	[64]	Controlled	Simulated				Challenge obf.
PUF-FSM (2)	2019	[196]	Controlled	Simulated				Response obf.
	2018	[135]	Controlled	FPGA				-
rMPUF (2)	2021	[217]	Controlled	Xilinx Artix-7	99%	51%		Response obf.
	2018	[62]	Controlled	Simulated	98.67%	50.04%	49.95%	Challenge obf.
RPUF (2)	2019	[196]	Controlled	Simulated				Challenge obf.
	2016	[68]	Controlled	Xilinx Zynq-7000	94.80%	48.9%	52.2%	Challenge obf.
TP PUF (2)	2023	[73]	Controlled	Xilinx Artix-7	96.59%		57.97%	Response obf.
	2023	[73]	Strong	Xilinx Artix-7				-
VPUF (2)	2017	[119]	Controlled	Xilinx Kintex-7	∼93%	49.8%	49.7%	-
	2017	[119]	Controlled	Xilinx Kintex-7	∼93%	49.8%	49.7%	-
XMPUF (2)	2022	[209]	Controlled	Simulated				Challenge obf.
	2018	[74]	Controlled	Xilinx Artix-7		37.03%	40.6%	Challenge obf.
XOR RO PUF (2)	2022	[230]	Strong	Xilinx Zynq-7000	98%	49.68%	49.62%	-
	2020	[233]	Strong	Xilinx Artix-7	92.87%	46.04%	44.64%	-
AES-PUF	2019	[163]	Controlled	Simulated	97.4%	47.1%	52.4%	Full obf.
AML PUF	2021	[177]	Controlled	Xilinx Artix-7				Response obf.
AROPUF	2017	[152]	Controlled	Xilinx Spartan 3E				Challenge obf.
Alahmadi PUF 1	2024	[43]	Controlled	Simulated				Challenge obf.
BST-RPUF	2021	[45]	Controlled	Xilinx Artix-7	∼100%	46.78%	48.64%	Challenge obf.
Bent PUF	2021	[228]	Controlled	Simulated	∼80%	54.6%	∼49.95%	Challenge obf.
CBDC-PUF	2023	[46]	Strong	Xilinx Artix-7	100%	49.6%	49.8%	-
CO-PUF	2024	[47]	Controlled	Xilinx Artix-7	95.37%	∼50%	50.01%	Response obf.
CP PUF	2022	[162]	Controlled	Simulated	98%	49.94%	49.99%	Full obf.
CPP-APUF	2020	[153]	Controlled	Altera FPGA	99.67%	50.18%	51.06%	Challenge obf.
CRC-PUF	2019	[133]	Controlled	Simulated		50.08%	50.00%	Challenge obf.
CRO XMPUF	2018	[192]	Controlled	Simulated				-
CRPUF	2018	[229]	Strong	Xilinx Spartan-3E	86.4%	43%		-
CT PUF	2022	[48]	Controlled	Xilinx Zynq-7000	∼92%	∼50%	∼50%	Data poisoning
Composite PUF	2015	[61]	Controlled	Xilinx Spartan-3	98.85%	54.76%	36.87%	Response obf.
DC MUX PUF	2020	[231]	Strong	Simulated	94.10%	99.93%	96.77%	-
DCA PUF	2015	[172]	Controlled	Altera Cyclone II				Anti-invasive
DEPUF	2023	[188]	Strong	Simulated		48.02%		-
DFM-APUF	2022	[134]	Controlled	Xilinx Artix-7	94.82%	50.69%	50.17%	Challenge obf.
DMOS-PUF	2018	[154]	Controlled	Xilinx Artix-7	94.8%		49.7%	Full obf.
DPUF	2023	[2]	Controlled	Xilinx Artix-7				Response obf.
Domino IPUF	2020	[224]	Controlled	Simulated				Challenge obf.
Dual-mode PUF	2018	[51]	Strong	Xilinx Artix-7	∼86%	44.65%		Data poisoning
DyAdv PUF	2024	[52]	Controlled	Xilinx Artix-7	97.97%	49.67%	50.13%	Response obf.

TABLE V: Full overview of PUF implementations (continued).

Delay (186)	Ebrahimabadi PUF	2021	[39]	Controlled	Xilinx Artix-7	96%	50.02%	45.66%	Challenge obf.
	FLAM-PUF	2022	[54]	Controlled	Simulated	95.59%	49.73%	49.81%	Challenge obf.
	FOM CDS-PUF	2023	[55]	Controlled	Xilinx Artix-7	92.09%	52.54%	50.50%	Challenge obf.
	HELP PUF	2018	[107]	Controlled	Simulated				Full obf.
	LBIST-PUF	2020	[58]	Controlled	Xilinx Artix-7	89.6%	∼60%	∼70%	Challenge obf.
	LEE PUF	2023	[59]	Controlled	Xilinx Spartan-6	99.74%	49%	58.73%	Response obf.
	LHS-PUF	2019	[196]	Controlled	Simulated				Response obf.
	LROBIPUF	2021	[130]	Controlled	Xilinx Artix-7				Challenge obf.
	MARPUF	2021	[232]	Controlled	Simulated	87.5%	48.23%	47.12%	Challenge obf.
	ME-PUF	2021	[156]	Controlled	Simulated		∼50%		Response obf.
	MMP PUF	2021	[164]	Controlled	Simulated				Full obf.
	MRO-PUF	2018	[197]	Controlled	Simulated	56-93%	∼50%	∼45%	Response obf.
	MUX PUF	2015	[216]	Strong	Simulated				-
	Ma PUF	2023	[150]	Controlled	Xilinx Artix-7	99.8%		49.89%	Challenge obf.
	PUF-CPRNG	2022	[40]	Controlled	Xilinx Zynq-7000				Full obf.
	OB-PUF	2019	[196]	Controlled	Simulated				Challenge obf.
	OBCIPUF	2021	[130]	Controlled	Xilinx Artix-7				Challenge obf.
	Oun PUF	2021	[121]	Controlled	Xilinx Artix-7				Challenge obf.
	P-2APUF	2024	[65]	Controlled	Xilinx Virtex-7	∼53%	∼50%		Challenge obf.
	P2M-Sec	2018	[106]	Controlled					Response obf.
	PFO PUF	2024	[66]	Controlled	Xilinx Spartan-6	96.91%	49.00%	56.02%	Full obf.
	Poly PUF	2019	[196]	Controlled	Simulated				Full obf.
	Polymorphic PUF	2017	[122]	Controlled	Xilinx Artix-7		48.0%	61.4%	-
	ROinLFSR PUF	2024	[158]	Controlled	Xilinx Virtex-7	94.67%	50.06%	50.49%	Response obf.
	SCD-PUF	2024	[69]	Controlled	Xilinx Artix-7	97.79%	49.56%	49.95%	Full obf.
	SOI PUF	2024	[70]	Controlled	Xilinx Artix-7	98.6%	48.2%	29.1%	Response obf.
	SP-PUF	2022	[1]	Controlled		∼90%		∼49%	Challenge obf.
	SRPUF	2021	[71]	Controlled	Xilinx Artix-7	91.87%	50.14%	50.03%	Challenge obf.
	SW PUF	2024	[72]	Controlled	Xilinx Artix-7	99.97%	49.74%	49.89%	Full obf.
	TVO-APUF	2021	[120]	Controlled	Xilinx Spartan-6	94.2%	53.2%	47.1%	Challenge obf.
	Tree IPUF	2020	[224]	Controlled	Simulated				Full obf.
	Trit PUF	2019	[38]	Controlled	Xilinx Artix-7	100%		49.7%	Challenge obf.
	Two-stage PUF	2018	[125]	Controlled	TSMC 65nm	70.50%	39.06%	47.76%	Challenge obf.
	XOR Cascaded IPUF	2020	[224]	Controlled	Simulated				Full obf.
	XOR Domino IPUF	2020	[224]	Controlled	Simulated				Full obf.
	XOR IPUF	2020	[224]	Controlled	Simulated				Full obf.
	XOR Multi-PUF	2023	[227]	Controlled	Xilinx Zynq-7000	97%	∼50%	48.85%	Challenge obf.
	XOR OPUF	2018	[234]	Controlled	Xilinx Kintex-7	86.3%			Challenge obf.
	XOR RPUF	2018	[234]	Controlled	Xilinx Kintex-7	93.5%			Challenge obf.
	XRBR PUF	2022	[237]	Strong					-
	XRRO PUF	2022	[237]	Weak					-
	cMPUF	2018	[62]	Controlled	Simulated	98.10%	50.23%	50.00%	Challenge obf.
	cSOI PUF	2024	[70]	Controlled	Xilinx Artix-7	98.3%	49.1%	49.8%	Full obf.
Memory (27)	SRAM PUF (5)	2023	[42]	Weak	Simulated	96.99%	50.03%	47.67%	-
		2023	[42]	Weak	Simulated	99.14%	50.15%	47.71%	Anti-invasive
		2022	[136]	Weak	130nm CMOS				-
		2016	[114]	Weak	Alliance AS6C6264				-
		2013	[204]	Weak	Atmel ATmega328P				-
	BR PUF (3)	2020	[77]	Strong	Xilinx Spartan-6	99%	47%	49%	-
		2015	[141]	Strong	Xilinx Spartan-6	∼97%	∼20%		-
		2014	[140]	Strong	Xilinx Spartan-6	∼80%	∼3-32%		-
	TBR PUF (3)	2020	[77]	Strong	Xilinx Spartan-6	97%	54%	50%	-
		2015	[141]	Strong	Xilinx Spartan-6				-
		2014	[140]	Strong	Xilinx Spartan-6	∼78%	∼14%		-
	XOR BR PUF (3)	2020	[77]	Controlled	Xilinx Spartan-6				Response obf.
		2020	[77]	Controlled	Xilinx Spartan-6				Full obf.
		2015	[141]	Controlled	Xilinx Spartan-6	∼92%	∼70%	∼50%	Response obf.
	2SPUF	2020	[76]	Weak	Simulated	96.19%	49.34%	48.1%	Challenge obf.
	Bitline PUF	2014	[214]	Strong	Simulated	92.4%		50.03%	-
	D-PUF	2016	[138]	Weak	Altera Stratix IV GX	¡30°C OK		¡40°C OK	-
	DRAM PUF	2019	[137]	Weak	Xilinx Spartan-6				-
	Feedback SPN PUF	2021	[78]	Controlled	130nm CMOS	99.27%		∼50%	Challenge obf.
	LS-BR PUF	2024	[142]	Controlled	Xilinx FPGA	98.15	49.49%	48.31%	Full obf.
	Liu PUF	2022	[126]	Controlled	130nm CMOS	∼100%		∼50%	Challenge obf.
	MRAM PUF	2015	[79]	Weak	Fabricated	97.25%		47%	-
	Multi-port PUF	2013	[37]	Weak	65nm CMOS	98.15%			-
	Suresh PUF	2020	[80]	Controlled	14nm CMOS	99.74%	∼50%		Full obf.
	XM XOR BR PUF	2020	[77]	Controlled	Xilinx Spartan-6				Challenge obf.
	XOR TBR PUF	2020	[77]	Controlled	Xilinx Spartan-6				Response obf.
	eFlash PUF	2019	[81]	Strong	55nm CMOS	5%	50.3%		-
Other (63)	SCA PUF (6)	2024	[191]	Strong	Simulated	97.2%	48.3%	50.3%	-
		2024	[191]	Controlled	Simulated	97.0%	55.6%	49.9%	Challenge obf.
		2024	[191]	Controlled	Simulated	96.3%	49.9%	50.0%	Challenge obf.
		2023	[202]	Controlled	Simulated	99.4%	49.3%	50.7%	Challenge obf.
		2022	[238]	Strong					-
		2020	[94]	Strong	130nm CMOS	97.4%	52.8%	49.9%	-
	Arbiter MRAM PUF (4)	2022	[84]	Strong	Simulated	99.55%		49.76%	-
		2022	[84]	Controlled	Simulated	99.55%		49.76%	Response obf.
		2021	[139]	Strong	Simulated	99.85%	∼50%	50.21%	-
		2021	[139]	Controlled	Simulated				Response obf.

TABLE V: Full overview of PUF implementations (continued).

Other (63)	STT-MRAM PUF (4)	2020	[195]	Strong	Simulated	∼100%	51.05%	50.01%	-
		2019	[115]	Weak	Simulated				-
		2019	[115]	Controlled	Simulated				Anti-invasive
		2024	[96]	Weak	Simulated	∼97%	49.09%	49.96%	-
	VTC PUF (4)	2020	[173]	Strong	Simulated				-
		2020	[173]	Strong	Simulated				Side-channel obf.
		2016	[215]	Strong	Simulated				-
		2015	[98]	Strong	Simulated	97.9%	50.1%	49.8%	-
	Alahmadi PUF 2 (3)	2024	[159]	Controlled	Xilinx Artix-7				Response obf.
		2024	[159]	Controlled	Xilinx Artix-7				Response obf.
		2024	[159]	Controlled	Xilinx Artix-7				Response obf.
	CIS PUF (3)	2024	[85]	Weak	Simulated	97.23%	47.62%	47.91%	-
		2024	[85]	Weak	Simulated				Response obf.
		2021	[205]	Weak	Simulated				-
	Current Mirror PUF (3)	2016	[200]	Strong	Simulated				-
		2016	[215]	Strong	Simulated				-
		2014	[86]	Strong	Simulated	98%	47%	49%	-
	SC PUF (3)	2021	[93]	Weak	180nm CMOS	∼100%	46.72%	50.38%	-
		2024	[143]	Weak	7nm CMOS	95.24%	49.34%	50.22%	Anti-invasive
		2018	[111]	Strong	0.18um CMOS	100%	47.85%	50.26%	-
	HP mem-PUF (3)	2021	[88]	Weak	PrPyrl	100%		48.1%	-
		2021	[88]	Strong	PrPyrl	100%		40.07%	-
		2021	[88]	Controlled	PrPyrl	100%			Challenge obf.
	Xbar PUF (3)	2018	[206]	Strong	Simulated				-
		2017	[193]	Strong	Simulated				-
		2017	[193]	Controlled	Simulated				Response obf.
	He PUF (2)	2022	[33]	Controlled	65nm CMOS	∼100%		∼50%	Challenge obf.
		2022	[33]	Controlled	65nm CMOS				-
	3D NbOx PUF	2021	[82]	Weak	NbOx	97.59%		49.97%	-
	AC-XOR PUF	2022	[144]	Strong	65nm CMOS	99.42%		49.92%	-
	AM-PUF	2023	[83]	Controlled	Xilinx Zynq-7010	98.1%	51.19%	53.51%	Full obf.
	Capacitive PUF	2021	[27]						Anti-invasive
	CF-PUF	2018	[147]	Strong	Simulated	80%	50.70%	53.96%	-
	Chaotic PUF	2019	[151]	Controlled	Xilinx Artix-7	97.01%	53.16%	41.17%	Challenge obf.
	ESP-PUF	2022	[87]	Controlled	Altera Cyclone 2	∼100%	50.10%	49.96%	Full obf.
	Hybrid PUF	2023	[89]	Controlled	Xilinx Spartan-6	99.7%	49.78%	49.38%	Full obf.
	Lattice PUF	2020	[90]	Controlled	Xilinx Spartan-6	98.74%	49.98%	50.00%	Other
	Lin PUF	2023	[110]	Strong	65nm CMOS	99.8%		∼50%	-
	MDR-ROM PUF	2015	[171]	Controlled	180nm CMOS	98.36%		44.44%	Side-channel obf.
	MR-PUF	2024	[91]	Weak	Simulated		51.10%	49.40%	-
	Memristive PUF	2020	[145]	Strong	Simulated		56.33%	49%	-
	Neuron-PUF	2021	[92]	Controlled	65nm CMOS	100%	47.49%	48.42%	-
	PUF ID	2014	[41]	Controlled	Simulated				Response obf.
	RF-PUF	2018	[190]	Strong	Simulated	∼99.97%		∼99.97%	-
	RRAM PUF	2021	[236]	Controlled	Simulated				Challenge obf.
	SPN PUF	2023	[95]	Controlled	130nm CMOS	∼100%		∼50%	Full obf.
	Sponge PUF	2024	[97]	Controlled	Simulated	∼50%	50.03%	50.00%	Full obf.
	Weak-assist SPUF	2022	[102]	Controlled	Altera Cyclone 2	∼100%	50.08%	49.99%	Response obf.
	XOR MRAM PUF	2021	[139]	Controlled	Simulated				Response obf.
	XOR RRAM PUF	2019	[157]	Controlled	1Kb RRAM array	95.5%	∼50%	∼50%	Response obf.
	Zuo PUF	2023	[146]	Strong	65-nm CMOS	∼98%	∼50%	∼50%	-
	iPUF	2020	[113]	Controlled	55nm CMOS	90.07%	48.53%	48.03%	Challenge obf.
	mrSPUF	2015	[3]	Strong	Simulated	92.5%	50.76%	50.07%	-
Non-silicon (5)	NOS PUF	2023	[99]	Strong	Simulated			∼50%	-
	Optical PUF	2016	[117]	Strong					-
	QR-PUF	2012	[118]	Strong					-
	QuPUF	2021	[100]	Strong	IBM Quantum	13.82%		55.13%	-
	SBS-CAN PUF	2024	[101]	Strong	Fabricated	96%	49.20%	49.60%	-

APPENDIX B
CLASSIFICATION OF PUF MODELING TECHNIQUES

TABLE VI: Full overview of PUF modeling techniques.

Attack		Year	Ref ID	PUF	Countermeasure	Model	CRPs	Time	Accuracy
Machine Learning (246)	MLP (88)	2021	[82]	3D NbOx PUF	-	Black-box	$2*10^6$		∼50%
		2022	[144]	AC-XOR PUF	-	Black-box	10^6		50.99%
		2023	[83]	AM-PUF	Full obf.	Black-box	$7*10^4$		∼50%
		2021	[177]	AML PUF	Response obf.	Black-box	10^5		∼64%
		2017	[152]	AROPUF	Challenge obf.	Black-box			50.36%
		2024	[159]	Alahmadi PUF 2	Response obf.	Parametric	10^7		57%
		2024	[159]	Alahmadi PUF 2	Response obf.	Parametric	10^7		54%
		2024	[159]	Alahmadi PUF 2	Response obf.	Parametric	10^7		52%
		2022	[84]	Arbiter MRAM PUF	-	Black-box	$2*10^4$		79.7%
		2022	[84]	Arbiter MRAM PUF	Response obf.	Black-box	$2*10^4$		53.8%
		2022	[4]	Arbiter PUF	-	Parametric	$5*10^3$		∼100%
		2016	[218]	Arbiter PUF	-	Black-box	$5*10^4$		58%
		2019	[128]	Arbiter PUF	-	Black-box	$6.8*10^2$	00:00:11	99.50%
		2024	[47]	CO-PUF	Response obf.	Black-box	$5*10^5$		58.89%
		2022	[162]	CP PUF	Full obf.	Black-box	$2.5*10^6$		∼50%
		2020	[153]	CPP-APUF	Challenge obf.	Black-box	$1.5*10^5$		∼62%
		2018	[219]	CRO PUF	-	Black-box	$3.264*10^3$		88.6%
		2022	[180]	Challenge Splitting	Challenge obf.	Parametric	$6*10^4$		∼51.92%
		2021	[181]	Challenge Permutation	Challenge obf.	Black-box	$3*10^4$		44.50%
		2019	[35]	DAPUF	Response obf.	Black-box	$1.7*10^7$		81.5%
		2022	[183]	DAUP	Challenge obf.	Black-box	$2*10^5$		∼50%
		2023	[188]	DEPUF	-	Parametric	$6*10^4$		∼94%
		2018	[154]	DMOS-PUF	Full obf.	Black-box	10^5		63.61%
		2023	[2]	DPUF	Response obf.	Parametric	10^6		81.11%
		2018	[51]	Dual-mode PUF	Data poisoning	Black-box	$5*10^3$		61.05%
		2020	[184]	ECC	Response obf.	Black-box	10^5		∼50%
		2021	[39]	Ebrahimabadi PUF	Challenge obf.	Black-box	10^6		∼58%
		2024	[189]	FF-IPUF	Challenge obf.	Black-box	10^6		65.42%
		2024	[189]	FF-IPUF	Challenge obf.	Parametric + SC	10^6		96.85%
		2023	[53]	FF-PUF	Challenge obf.	Parametric	$1.5*10^6$		92.33%
		2017	[222]	FF-PUF	Challenge obf.	Parametric	$2*10^4$	00:00:08	91.01%
		2021	[226]	IPUF	Challenge obf.	Parametric	10^6	00:14:00	92.56%
		2021	[34]	IPUF	Challenge obf.	Black-box	$3*10^6$	07:00:00	93%
		2021	[34]	IPUF	Challenge obf.	Black-box	$2*10^6$	01:20:00	95%
		2019	[128]	IPUF	Challenge obf.	Black-box	$3.19*10^5$	00:05:23	97.44%
		2022	[148]	Ising-PUF	Challenge obf.	Black-box	$4*10^6$		∼50%
		2018	[57]	Ising-PUF	Challenge obf.	Black-box	$5*10^4$		∼50%
		2023	[60]	LP-PUF	Full obf.	Parametric			∼80%
		2019	[128]	LSPUF	Response obf.	Black-box	$8*10^5$	00:33:24	97.42%
		2020	[90]	Lattice PUF	Other	Black-box	10^6		∼50%
		2022	[126]	Liu PUF	Challenge obf.	Black-box	$4*10^7$		50.03%
		2021	[232]	MARPUF	Challenge obf.	Black-box	$2*10^4$		58.9%
		2021	[164]	MMP PUF	Full obf.	Black-box	$2.7*10^6$		50.09%
		2020	[132]	MPUF	Challenge obf.	Parametric	$8*10^4$		96.04%
		2019	[128]	MPUF	Challenge obf.	Black-box	$3.2*10^5$	00:15:23	96.54%
		2023	[150]	Ma PUF	Challenge obf.	Black-box	$5*10^5$		56%
		2022	[40]	PUF-CPRNG	Full obf.	Black-box	$6*10^5$		48.07%
		2023	[64]	OI-PUF	Challenge obf.	Parametric	10^7		61%
		2021	[121]	Oun PUF	Challenge obf.	Black-box	$5*10^3$		7.5%
		2024	[66]	PFO PUF	Full obf.	Black-box	10^6		∼67%
		2022	[176]	Permutation Interface	Challenge obf.	Parametric	10^7		∼80%
		2019	[196]	Poly PUF	Full obf.	Parametric	10^5		∼95%
		2018	[190]	RF-PUF	-	White-box			99.99%
		2021	[177]	Response inversion	Response obf.	Black-box	$5*10^3$		95.70%
		2021	[236]	RRAM PUF	Challenge obf.	Parametric	$\sim 1.3*10^7$		92.33%
		2022	[238]	SCA PUF	-	Parametric	10^4		∼95%
		2020	[94]	SCA PUF	-	Black-box	10^4		∼60%
		2024	[191]	SCA PUF	-	Parametric	$3.5*10^2$		97%
		2024	[191]	SCA PUF	Challenge obf.	Parametric	$8*10^2$		∼90%
		2024	[69]	SCD-PUF	Full obf.	Black-box	10^6		51.29%
		2024	[70]	SOI PUF	Response obf.	Black-box	$4*10^7$	20:04:12	69.19%
		2023	[95]	SPN PUF	Full obf.	Black-box	10^7		∼50%
		2021	[71]	SRPUF	Challenge obf.	Black-box	$5*10^5$	00:54:30	50.28%
		2024	[26]	Selective CRPs	Data poisoning	Black-box	$2.5*10^5$		∼70%
		2024	[97]	Sponge PUF	Full obf.	Black-box	$2*10^7$		∼50%
		2020	[80]	Suresh PUF	Full obf.	Black-box	$7*10^5$		99%
		2023	[73]	TP PUF	Response obf.	Black-box	$\sim 2*10^8$		50.90%

MLP: Multi-Layer Perceptron

TABLE VI: Full overview of PUF modeling techniques (continued).

Machine Learning (246)	MLP (88)	2023	[73]	TP PUF	-	Black-box	$\sim 3*10^7$		71.11%
		2018	[125]	Two-stage PUF	Challenge obf.	Black-box	$9*10^3$		59.85%
		2022	[102]	Weak-assist SPUF	Response obf.	Black-box	$2*10^7$		$\sim$50%
		2020	[77]	XM XOR BR PUF	Challenge obf.	Black-box	10^5		99.1%
		2020	[77]	XOR BR PUF	Response obf.	Black-box	10^6	00:04:48	99.5%
		2020	[77]	XOR BR PUF	Full obf.	Black-box	10^5		82.3%
		2022	[32]	XOR FF-PUF	Full obf.	Parametric	$1.8*10^7$	24:00:00	90%
		2022	[127]	XOR FF-PUF	Full obf.	Parametric	10^5	00:01:51	67.75%
		2020	[165]	XOR FF-PUF	Full obf.	Black-box	10^6		$\sim$80%
		2022	[36]	XOR PUF	Response obf.	Parametric	$1.75*10^4$		$\sim$97.5%
		2022	[32]	XOR PUF	Response obf.	Parametric	$3.25*10^8$	53 days	98.1%
		2020	[223]	XOR PUF	Response obf.	Parametric	$4*10^6$	00:09:15	99.07%
		2020	[29]	XOR PUF	Response obf.	Black-box	$3.4*10^6$	00:12:51	96%
		2019	[128]	XOR PUF	Response obf.	Black-box	$6.8*10^5$	00:20:52	97.68%
		2018	[5]	XOR PUF	Response obf.	Black-box	$3*10^7$	00:23:18	99.17%
		2019	[157]	XOR RRAM PUF	Response obf.	Black-box			$\sim$57%
		2020	[77]	XOR TBR PUF	Response obf.	Black-box	10^6	00:06:48	98.8%
		2023	[146]	Zuo PUF	-	Black-box	10^7		$\sim$51.84%
		2024	[70]	cSOI PUF	Full obf.	Black-box	$4*10^7$	17:12:07	58.71%
		2019	[81]	eFlash PUF	-	Black-box	10^5		$\sim$50%
		2021	[217]	rMPUF	Response obf.	Black-box	$8.55*10^5$		93.12%
	LR (54)	2020	[76]	2SPUF	Challenge obf.	Black-box	$5*10^3$		60.4%
		2021	[139]	Arbiter MRAM PUF	Response obf.	Black-box	$7.5*10^3$		$\sim$50%
		2021	[139]	Arbiter MRAM PUF	-	Black-box	$7.5*10^3$		$\sim$65%
		2020	[124]	Arbiter PUF	-	Black-box	$4.182*10^3$		50.41%
		2013	[31]	Arbiter PUF	-	Parametric	$6.5*10^3$	830 ms	99%
		2013	[31]	Arbiter PUF	-	Parametric	$6.5*10^3$	760 ms	99%
		2013	[112]	Arbiter PUF	-	Parametric	$1.8*10^4$		$\sim$88%
		2019	[133]	CRC-PUF	Challenge obf.	Black-box			75%
		2018	[192]	CRO XMPUF	-	Parametric + SC	10^4		$\sim$98%
		2018	[229]	CRPUF	-	Black-box	10^5		$\sim$85%
		2019	[151]	Chaotic PUF	Challenge obf.	Black-box	$5*10^4$		52.53%
		2022	[182]	CoLAC	Response obf.	Black-box	$4*10^4$		$\sim$50%
		2015	[61]	Composite PUF	Response obf.	Parametric	$1.25*10^4$		94.68%
		2020	[231]	DC MUX PUF	-	Parametric	10^4	00:08:00	90.77%
		2022	[134]	DFM-APUF	Challenge obf.	Black-box	10^5		50.71%
		2019	[137]	DRAM PUF	-	Black-box			51.8%
		2020	[224]	Domino IPUF	Challenge obf.	Parametric	$2*10^7$	24:00:00	95%
		2022	[87]	ESP-PUF	Full obf.	Black-box	10^6		$\sim$53%
		2022	[54]	FLAM-PUF	Challenge obf.	Black-box	10^6		53.4%
		2021	[130]	IPUF	Challenge obf.	Parametric	10^5		68.59%
		2020	[224]	IPUF	Challenge obf.	Parametric	$7.5*10^8$	56 days	95%
		2020	[224]	IPUF	Challenge obf.	Parametric	$3*10^8$	18 days	95%
		2023	[60]	LP-PUF	Full obf.	Parametric	$5*10^5$		$\sim$100%
		2021	[130]	LROBIPUF	Challenge obf.	Black-box	10^5		62.12%
		2019	[131]	LSPUF	Side-channel obf.	Parametric + SC	$5*10^3$		49.2%
		2015	[61]	LSPUF	Response obf.	Parametric	$3*10^4$		36.30%
		2013	[31]	LSPUF	Response obf.	Black-box	10^6	267 days	99%
		2013	[168]	LSPUF	Response obf.	Parametric + SC	$2*10^6$	00:01:45	96.0%
		2016	[155]	Lockdown	Challenge obf.	Black-box	$1.2*10^7$		Fail
		2020	[149]	MMPUF	Challenge obf.	Black-box	10^5		$\sim$60%
		2020	[145]	Memristive PUF	-	Black-box	$5.4*10^4$		$\sim$54%
		2015	[175]	Noise bifurcation	Challenge obf.	Parametric	$4*10^6$	01:00:00	88%
		2014	[161]	Noise bifurcation	Challenge obf.	Parametric	$5*10^5$	05:00:00	92%
		2021	[130]	OBCIPUF	Challenge obf.	Black-box	10^5		63.57%
		2024	[65]	P-2APUF	Challenge obf.	Black-box	$1.4*10^4$		52.16%
		2024	[158]	ROinLFSR PUF	Response obf.	Black-box	10^6		$\sim$50%
		2022	[1]	SP-PUF	Challenge obf.	Black-box	10^4		$\sim$68%
		2024	[26]	Selective CRPs	Data poisoning	Parametric	$3*10^4$		$\sim$86%
		2021	[120]	TVO-APUF	Challenge obf.	Black-box	10^4		$\sim$54%
		2020	[224]	Tree IPUF	Full obf.	Parametric	$5*10^6$	08:48:00	95%
		2017	[119]	VPUF	-	Parametric			81.4%
		2020	[224]	XOR Cascaded IPUF	Full obf.	Parametric	10^7	02:42:00	95%
		2020	[224]	XOR Domino IPUF	Full obf.	Parametric	$4*10^7$	2 days	95%
		2020	[224]	XOR IPUF	Full obf.	Parametric	$4*10^7$	3 days	95%
		2021	[139]	XOR MRAM PUF	Response obf.	Black-box	$7.5*10^3$		$\sim$45%
		2023	[227]	XOR Multi-PUF	Challenge obf.	Black-box	10^5		90%
		2013	[31]	XOR PUF	Response obf.	Parametric	$7.8*10^4$	00:39:00	99%
		2013	[31]	XOR PUF	Response obf.	Parametric	$7.8*10^4$	00:18:09	99%
		2021	[226]	XOR PUF	Response obf.	Parametric	$5*10^4$		96.87%
		2015	[175]	XOR PUF	Response obf.	Black-box	$3.5*10^8$	37:46:00	98%
		2013	[168]	XOR PUF	Response obf.	Parametric + SC	$5*10^5$	00:04:07	95%
		2022	[237]	XRBR PUF	-	Parametric	10^4		99.40%
		2022	[237]	XRRO PUF	-	Parametric	$8*10^6$		98.06%
		2017	[193]	XbarPUF	-	Black-box	$5*10^3$		99.0%

LR: Logistic Regression, **MLP**: Multi-Layer Perceptron

TABLE VI: Full overview of PUF modeling techniques (continued).

Machine Learning (246)	SVM (30)	2022	[28]	Arbiter PUF	-	Parametric + SC	10^3		92.59%
		2022	[28]	Arbiter PUF	Response obf.	Parametric + SC	10^3		53.18%
		2020	[173]	Arbiter PUF	-	Parametric + SC	$8*10^3$	00:00:09	96.65%
		2018	[221]	Arbiter PUF	-	Parametric	$3.5*10^2$		92.72%
		2017	[123]	Arbiter PUF	-	Parametric	10^3		97.4%
		2017	[123]	Arbiter PUF	-	Parametric	10^3		55.1%
		2015	[141]	BR PUF	-	Parametric	$1.3*10^3$		95%
		2014	[214]	Bitline PUF	-	Black-box	$5*10^2$		∼90%
		2018	[147]	CF-PUF	-	Black-box	10^4		∼50%
		2014	[86]	Current Mirror PUF	-	Black-box	$2*10^6$	00:20:40	70%
		2018	[169]	RO PUF	Challenge obf.	Black-box	10^4		∼50%
		2018	[225]	DAPUF	Response obf.	Parametric	$5*10^4$		∼70%
		2015	[49]	DAPUF	Response obf.	Black-box	10^3		80.72%
		2019	[194]	FF-PUF	Challenge obf.	Parametric + SC	10^3		∼91%
		2022	[33]	He PUF	Challenge obf.	Black-box	10^7		∼50%
		2024	[142]	LS-BR PUF	Full obf.	Black-box	10^4		∼51%
		2021	[156]	ME-PUF	Response obf.	Black-box	$6.4*10^4$		50.11%
		2015	[216]	MUX PUF	-	Black-box	$5*10^4$		∼10%
		2021	[63]	NoPUF	Challenge obf.	Black-box	$5*10^4$		∼58%
		2014	[41]	PUF ID	Response obf.	Black-box	$5.12*10^2$		∼0%
		2024	[96]	STT-MRAM PUF	-	Black-box	$1.675*10^4$		54.04%
		2020	[195]	STT-MRAM PUF	-	Black-box	10^5		∼95%
		2015	[141]	TBR PUF	-	Parametric	$7.5*10^2$		95%
		2020	[173]	VTC PUF	-	Parametric + SC	$8*10^3$	00:00:32	85.35%
		2020	[173]	VTC PUF	Side-channel obf.	Parametric + SC	$4*10^3$		∼60%
		2015	[98]	VTC PUF	-	Black-box	10^5		79.2%
		2017	[193]	XbarPUF	Response obf.	Black-box	$5*10^3$		57.9%
		2015	[141]	XOR BR PUF	Response obf.	Parametric	$7.2*10^3$	00:00:24	95%
		2015	[210]	XOR PUF	Response obf.	Parametric + SC	10^3		87%
		2020	[233]	XOR RO PUF	-	Black-box	10^4		62.4%
	CMA-ES (24)	2024	[43]	Alahmadi PUF 1	Challenge obf.	Black-box	10^6		56%
		2018	[192]	CRO PUF	-	Parametric + SC	10^4		∼100%
		2022	[48]	CT PUF	Data poisoning	Black-box	10^5		∼61%
		2022	[50]	DCH PUF	Response obf.	Parametric	$8*10^7$	40 days	96.89%
		2022	[50]	DCH PUF	Full obf.	Parametric	$4*10^4$		52.38%
		2021	[108]	Deception	Full obf.	Parametric + SC	$2*10^6$	228 years	99%
		2023	[55]	FOM CDS-PUF	Challenge obf.	Black-box	10^5		55%
		2019	[56]	IPUF	Challenge obf.	Black-box	$2*10^5$		∼50%
		2019	[196]	LHS-PUF	Response obf.	Parametric	10^4		∼98%
		2023	[110]	Lin PUF	-	Black-box	$5*10^7$		50.6%
		2020	[149]	MMPUF	Challenge obf.	Parametric + SC	10^4		∼73%
		2018	[62]	MPUF	Challenge obf.	Black-box	$2*10^5$		76%
		2023	[64]	OI-PUF	Challenge obf.	Parametric	$5*10^5$	12:54:00	77%
		2019	[196]	PUF-FSM	Response obf.	Parametric	10^2		∼100%
		2019	[178]	Response inversion	Response obf.	Black-box	10^6		72.33%
		2015	[109]	Reverse Fuzzy Extractor	Response obf.	Parametric	$1.785*10^3$	00:23:00	97%
		2015	[109]	Reverse Fuzzy Extractor	Response obf.	Parametric + SC	$5.355*10^3$	00:01:00	97%
		2024	[72]	SW PUF	Full obf.	Black-box	10^5		∼58%
		2015	[109]	Slender PUF	Full obf.	Parametric	$3*10^5$	30:18:00	96.9%
		2019	[38]	Trit PUF	Challenge obf.	Black-box	$1.6*10^4$	1000 years	∼62%
		2017	[119]	VPUF	-	Parametric + SC			86.2%
		2018	[74]	XMPUF	Challenge obf.	Parametric + SC	10^4		∼80%
		2020	[165]	XOR FF-PUF	Full obf.	Parametric + SC			Fail
		2015	[212]	XOR PUF	Response obf.	Parametric + SC	$5*10^5$	30:30:00	90.8%
		2018	[62]	rMPUF	Challenge obf.	Parametric + SC	$6*10^5$		99.68%
	ES (9)	2016	[215]	Current Mirror PUF	-	Black-box	10^4	01:52:00	99.26%
		2021	[108]	Deception	Full obf.	Parametric	$5*10^4$	6 years	99%
		2013	[31]	FF-PUF	Challenge obf.	Black-box	$5*10^4$	03:15:00	99%
		2017	[122]	Polymorphic PUF	-	Parametric	$1.024*10^5$		∼68%
		2016	[68]	RPUF	Challenge obf.	Black-box	$2*10^2$		57.3%
		2014	[179]	Slender PUF	Full obf.	Black-box	$6.4*10^4$		∼93%
		2016	[215]	VTC PUF	-	Black-box	10^4	01:23:00	99.31%
		2018	[234]	XOR OPUF	Challenge obf.	Black-box	$2.1*10^5$		86.5%
		2018	[234]	XOR RPUF	Challenge obf.	Black-box	$1.11*10^5$	04:51:36	93.7%
	RF (7)	2020	[173]	Arbiter PUF	Side-channel obf.	Parametric + SC	$4*10^3$		∼61%
		2016	[167]	Arbiter PUF	-	Black-box	$2*10^4$		97.5%
		2016	[167]	Arbiter PUF	Response obf.	Parametric + SC	$2*10^4$		∼87%
		2024	[186]	Arbiter PUF	-	Black-box	$6*10^3$		89%
		2020	[58]	LBIST-PUF	Challenge obf.	Black-box	$6*10^4$		∼93%
		2024	[91]	MR-PUF	-	Black-box	$5.0323*10^3$		52.30%
		2024	[186]	Noise injection	Data poisoning	Black-box	$6*10^3$		∼80%
	Boosting (4)	2018	[166]	Arbiter PUF	Data poisoning	Black-box	10^5		∼78%
		2018	[107]	HELP PUF	Full obf.	Black-box	$6.5536*10^4$		∼96%
		2018	[197]	MRO-PUF	Response obf.	Black-box	$5*10^4$	00:00:02	∼98%

LR: Logistic Regression, **SVM**: Support Vector Machine, **CMA-ES**: Covariance Matrix Adaptation Evolutionary Strategies, **ES**: Evolutionary Strategies, **RF**: Random Forest

TABLE VI: Full overview of PUF modeling techniques (continued).

Machine Learning (246)	Boosting (4)	2021	[88]	HP mem-PUF	Challenge obf.	Black-box	$1.2*10^5$	∼52%
	CNN (4)	2019	[163]	AES-PUF	Full obf.	Black-box	10^5	51.8%
		2019	[198]	RO PUF	-	Parametric + SC	$2*10^5$	92.8%
		2020	[220]	XOR PUF	Response obf.	Parametric	$1.15*10^5$	80%
		2020	[220]	XOR PUF	Response obf.	Parametric + SC	$1.5*10^5$	∼90%
	VAE (4)	2020	[174]	AES	Challenge obf.	Black-box	$2.5*10^2$	00:01:37 63.9%
		2020	[160]	AES	Challenge obf.	Black-box	$2.5*10^2$	00:00:47 59.2%
		2020	[174]	DES	Challenge obf.	Black-box	$2.5*10^2$	00:01:22 60.3%
		2020	[160]	DES	Challenge obf.	Black-box	$2.5*10^2$	00:00:42 57.3%
	DQN (3)	2024	[199]	Arbiter PUF	-	DFSM	10^4	94.98%
		2024	[199]	IPUF	Challenge obf.	Black-box	$1.5*10^5$	∼93%
		2024	[199]	XOR PUF	Response obf.	Parametric	$1.9*10^5$	∼90%
	DT (3)	2021	[78]	Feedback SPN PUF	Challenge obf.	Black-box	$2*10^7$	50.26%
		2023	[59]	LEE PUF	Response obf.	Black-box	10^5	49.74%
		2021	[88]	HP mem-PUF	-	Black-box	$1.2*10^5$	∼85%
	ECP-TRN (3)	2021	[75]	LSPUF	Response obf.	Parametric	$4.8*10^6$	00:25:07 96.73%
		2021	[75]	Lockdown	Challenge obf.	Black-box	$4.8*10^6$	00:48:12 98.09%
		2021	[75]	XOR PUF	Response obf.	Parametric	$2.7*10^6$	15:30:00 97.42%
	GA (3)	2016	[200]	Current Mirror PUF	-	Parametric	$5*10^4$	26:18:00 97.07%
		2020	[239]	RO PUF	-	Black-box	$5*10^4$	96%
		2019	[103]	RO PUF	-	Black-box	$5*10^4$	08:20:00 79.3%
	SLP (2)	2014	[140]	BR PUF	-	Black-box	$5*10^4$	∼100%
		2014	[140]	TBR PUF	-	Black-box	$5*10^4$	∼95%
	GRNN	2023	[201]	XOR PUF	Response obf.	Parametric	$7*10^4$	01:07:10 99.64%
	LM	2023	[202]	SCA-PUF	Challenge obf.	Parametric	$2*10^6$	∼54%
	LiR	2019	[196]	RPUF	Challenge obf.	Parametric	$1.024*10^3$	∼90%
	NB	2024	[52]	DyAdv PUF	Response obf.	Black-box	$7*10^4$	∼52%
	PAC	2016	[203]	Arbiter PUF	-	DFSM		
	RBFNN	2023	[46]	CBDC-PUF	-	Parametric	10^4	50.9%
	Other	2016	[129]	DAPUF	Response obf.	Black-box	$5*10^4$	02:24:42 ∼91%
	Other	2016	[129]	XOR PUF	Response obf.	Black-box	$5*10^4$	02:24:42 ∼97%
Invasive (8)	FIB (3)	2019	[115]	STT-MRAM PUF	-	Physical		Success
		2019	[115]	STT-MRAM PUF	Anti-invasive	Physical		Fail
		2013	[204]	SRAM PUF	-	Physical	1.6*10	00:05:00 Success
	BTI (2)	2023	[42]	SRAM PUF	-	Physical		98.64%
		2023	[42]	SRAM PUF	Anti-invasive	Physical		51%
	Bypass	2021	[27]	Capacitive PUF	Anti-invasive	Physical		
	MD	2021	[27]	Capacitive PUF	Anti-invasive	Physical		
	MP	2021	[27]	Capacitive PUF	Anti-invasive	Physical		
Other (21)	LP (3)	2018	[206]	Arbiter PUF	-	Parametric + SC	$2.415*10^3$	00:24:12 99.9%
		2018	[206]	XOR PUF	Response obf.	Parametric + SC	$5.718*10^3$	01:01:48 99%
		2018	[206]	Xbar PUF	-	Parametric + SC	$1.288*10^3$	00:12:48 99.9%
	Sorting (3)	2024	[85]	CIS PUF	-	Parametric	$2.66*10^3$	87.7%
		2024	[85]	CIS PUF	Response obf.	Parametric	$2.66*10^3$	66%
		2021	[205]	CIS PUF	-	Parametric	∼20	∼90%
	Cryptoanalysis (2)	2015	[207]	RO PUF	Response obf.	Parametric + SC		∼100%
		2015	[207]	RO PUF	Response obf.	Black-box		∼50%
	Characterization	2015	[208]	Arbiter PUF	-	Parametric	10^6	83.7%
	EM analysis	2021	[67]	RO PUF	-	Parametric + SC	0	94.2%
	Emulation	2015	[208]	Arbiter PUF	-	Physical		75.5%
	QS	2013	[31]	RO PUF	-	Parametric	$8.39*10^4$	99%
	-	2022	[33]	He PUF	-	White-box		∼50%
	-	2022	[209]	LP-PUF	Full obf.	Black-box	10^4	86%
	-	2022	[209]	MMPUF	Challenge obf.	Black-box	10^4	93%
	-	2023	[99]	NOS PUF	-	Physical		∼100%
	-	2021	[63]	NoPUF	Challenge obf.	Parametric		∼92%
	-	2014	[211]	RO PUF	-	Parametric + SC		
	-	2016	[114]	SRAM PUF	-	Physical		0%
	-	2022	[136]	SRAM PUF	-	Physical		
	-	2022	[209]	XMPUF	Challenge obf.	Black-box	10^4	94%

BTI: Bias Temperature Instability, **CMA-ES**: Covariance Matrix Adaptation Evolutionary Strategies, **CNN**: Convolutional Neural Network, **DQN**: Deep Q-Network, **DT**: Decision Tree, **ECP-TRN**: Efficient CANDECOM/PARAFAC-Tensor Regression Network, **FIB**: Focused Ion Beam, **GA**: Genetic Algorithm, **GRNN**: General Regression Neural Network, **LM**: Lagrange Multiplier, **LiR**: Linear Regression, **LP**: Linear Programming, **MD**: Micro-drilling, **MP**: Magnetic Probing, **NB**: Naive Bayes, **PAC**: Probably Approximately Correct, **QS**: Quick Sort, **RBFNN**: Radial Basis Function Neural Network, **SLP**: Single-Layer Perceptron, **VAE**: Variational Auto-Encoder