cryptography

MDPI

Review

FPGA-Based PUF Designs: A Comprehensive Review and Comparative Analysis

Kusum Lata [1] and Linga Reddy Cenkeramaddi [2,*]

[1] Department of Electronics and Communication Engineering, The LNM Institute of Information Technology, Jaipur 302031, India; kusum@lnmiit.ac.in
[2] Department of Information and Communication Technology, University of Agder, 4879 Grimstad, Norway
* Correspondence: linga.cenkeramaddi@uia.no

Abstract: Field-programmable gate arrays (FPGAs) have firmly established themselves as dynamic platforms for the implementation of physical unclonable functions (PUFs). Their intrinsic reconfigurability and profound implications for enhancing hardware security make them an invaluable asset in this realm. This groundbreaking study not only dives deep into the universe of FPGA-based PUF designs but also offers a comprehensive overview coupled with a discerning comparative analysis. PUFs are the bedrock of device authentication and key generation and the fortification of secure cryptographic protocols. Unleashing the potential of FPGA technology expands the horizons of PUF integration across diverse hardware systems. We set out to understand the fundamental ideas behind PUF and how crucially important it is to current security paradigms. Different FPGA-based PUF solutions, including static, dynamic, and hybrid systems, are closely examined. Each design paradigm is painstakingly examined to reveal its special qualities, functional nuances, and weaknesses. We closely assess a variety of performance metrics, including those related to distinctiveness, reliability, and resilience against hostile threats. We compare various FPGA-based PUF systems against one another to expose their unique advantages and disadvantages. This study provides system designers and security professionals with the crucial information they need to choose the best PUF design for their particular applications. Our paper provides a comprehensive view of the functionality, security capabilities, and prospective applications of FPGA-based PUF systems. The depth of knowledge gained from this research advances the field of hardware security, enabling security practitioners, researchers, and designers to make wise decisions when deciding on and implementing FPGA-based PUF solutions.

Keywords: hardware security; physically unclonable functions (PUFs); PUF applications; field-programmable gate arrays (FPGAs); FPGA-based PUFs

Citation: Lata, K.; Cenkeramaddi, L.R. FPGA-Based PUF Designs: A Comprehensive Review and Comparative Analysis. *Cryptography* **2023**, *7*, 55. https://doi.org/10.3390/cryptography7040055

Academic Editor: Jim Plusquellic

Received: 27 September 2023
Revised: 24 October 2023
Accepted: 30 October 2023
Published: 1 November 2023

1. Introduction

In light of modern computing systems permeating every facet of daily life, the need for robust security in these systems has grown significantly. This holds especially true for the Internet of Things (IoT) [1–3], which represents one of the central challenges, as it emerges in our personal lives and future industrial systems [4]. Physical unclonable functions (PUFs), which are used to create identifying strings and cryptographic keys, have become crucial as a hardware protection measure in recent years [5]. Physical unclonable functions (PUFs) have drawn a lot of interest because of their potential to offer strong security primitives for device authentication, safe key generation, and cryptographic protocols [6–10]. PUFs generate device-specific identifiers by taking use of the distinctive manufacturing differences available in hardware components, adding an extra layer of defense against various types of attacks.

Field-programmable gate arrays (FPGAs), with their reconfigurable nature and versatility, have emerged as a prominent platform for implementing PUFs. FPGAs, when compared to conventional application-specific integrated circuits (ASICs), give designers

the ability to build unique digital circuits at the hardware level, which makes them ideal for incorporating security measures made for particular applications. The combination of PUFs and FPGA technology not only strengthens the security of hardware systems but also opens a way to include security primitives in a variety of hardware components, from consumer electronics to vital infrastructure.

In today's embedded systems environment, the utilization of FPGAs for PUF implementation is in line with the demand for scalable, secure, and flexible solutions. It gives designers the ability to raise the security of their creations while retaining the adaptability and effectiveness needed for various applications. The generation of secret keys [11–13], the generation of random numbers [14–17], the protection of FPGA intellectual property [18,19], the identification of devices [20], chip authentication [8,17], key exchange/agreement protocols [8,21–23], the prevention of counterfeiting [24], and IoT security [8,22–27] are a few intriguing applications of FPGA-based PUFs.

1.1. Motivations

Secure storage is indeed a growing demand in various industries and applications. As the volume of sensitive data increases, there is a need for robust and reliable methods to protect this data from unauthorized access and tampering. PUFs have emerged as one of the solutions to address this demand for secure storage [28]. Semiconductor companies and researchers have been promoting the use of PUFs as a hardware-based security feature to enhance the security of data storage systems. PUFs leverage the inherent physical variations present in semiconductor devices during the manufacturing process. These variations are unique and practically impossible to replicate accurately. As a result, PUFs can be used to generate device-specific cryptographic keys or identifiers that serve as secure storage mechanisms. Knowmade [29] has analyzed the evolution of PUF-related patent applications which have been published in last two decades in Figure 1.

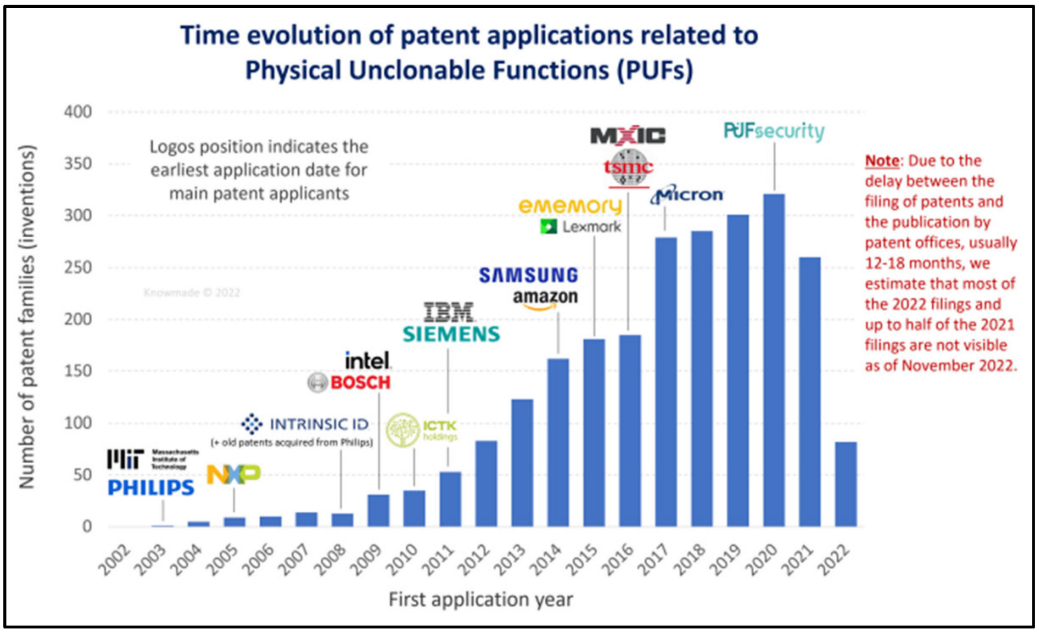

Figure 1. Evolution of patent applications related to PUF published in last two decades [29].

As also reported by Secure-IC in [30], while designing a good PUF, many parameters should be taken into account, as demonstrated in Figure 2.These parameters are considered to be broadly in five categories, namely no amount of helper data, proven reliability, security, uniqueness of generated responses, and flexibility and portability. These parameters also indicate the robustness of a good PUF design. PUF is also known to be a solution for anti-copy applications at a silicon level.

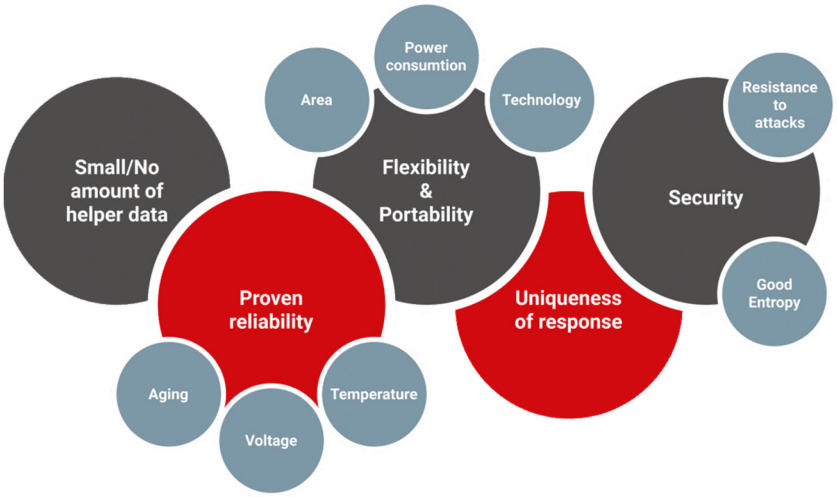

Figure 2. Typical characteristics of a good PUF (adapted from [30]).

According to a recent study by Markets and Markets Inc., Northbrook, IL, USA [31], the FPGA market is anticipated to increase from USD 9.7 billion in 2023 to USD 19.1 billion by 2028, as demonstrated in Figure 3, growing at a CAGR of 14.6% over that time. In a variety of industries, including automotive and consumer electronics, medical technology, remote sensing, military and aerospace, industrial control systems, etc., FPGAs have emerged as the platform of choice for many security-related applications [32–34] thanks to their inherent flexibility configurability in comparison to application specific integrated circuits (ASICs) as well as their quick time-to-market, lower design costs, and availability of third-party intellectual property (IPs). The prevalence of FPGAs in embedded applications and their ability to facilitate rapid product customization make them an ideal platform for implementing PUFs. The integration of PUFs with FPGAs offers a powerful synergy that aligns with the demands of modern embedded applications. It combines the security benefits of PUFs with the flexibility, cost efficiency, and rapid development cycles enabled by FPGAs, making it a highly desirable approach for securing a wide array of electronic devices and systems.

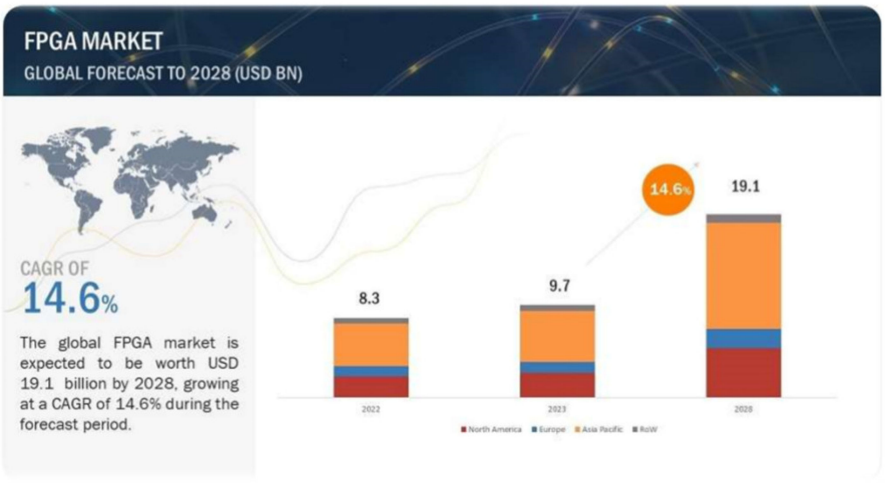

Figure 3. FPGA market global forecast to 2028 (adapted from [31]).

PUFs indeed have a broad range of applications across various fields due to their unique characteristics and capabilities. Their ability to harness the inherent randomness and uniqueness found in physical systems makes them a versatile solution in many areas. Figure 4 demonstrates a few examples to illustrate the breadth of PUF applications.

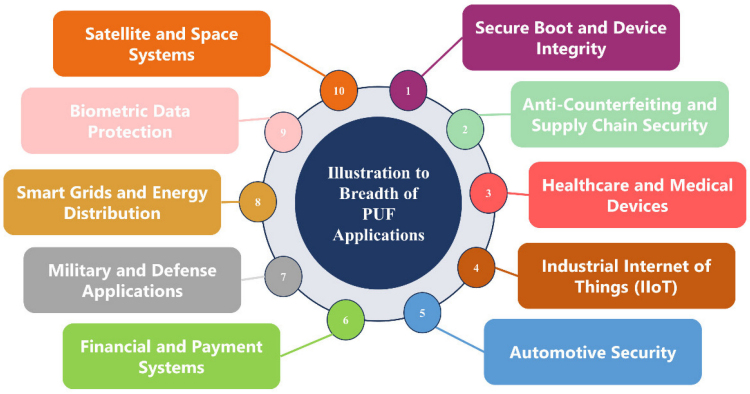

Figure 4. Illustration of breadth of PUF applications.

1.2. Contributions

FPGA-based PUF designs provides methods to assist in bridging the gap between the PUF usages for security purposes in many applications and FPGA growing market for design implementations. A review of the pertinent existing articles is necessary to have a thorough picture of the new techniques in this emerging sector. To do this, we give a full explanation of how PUF implementation for diverse security applications has been conducted using FPGA-based design methodologies. We think that our study can serve as a model for future field studies. These are the pivotal contributions of our work:

- Comprehensive Survey of FPGA-Based PUFs: We present a detailed survey encompassing the current and emerging methodologies for FPGA-based physical unclonable function (PUF) designs. Our focus centers on the security applications of PUFs across a wide array of product designs. To ensure the utmost relevance, we conducted a thorough analysis of more than 100 pertinent publications, encapsulating the very latest advancements in the field.
- In-Depth Categorization of PUF Applications: Our study delves deeper by categorizing research papers according to the diverse applications of FPGA-based PUFs, as illustrated in Figure 5. For each application, we place a spotlight on task-specific challenges and proffer well-informed solutions drawn from extensive literature research.

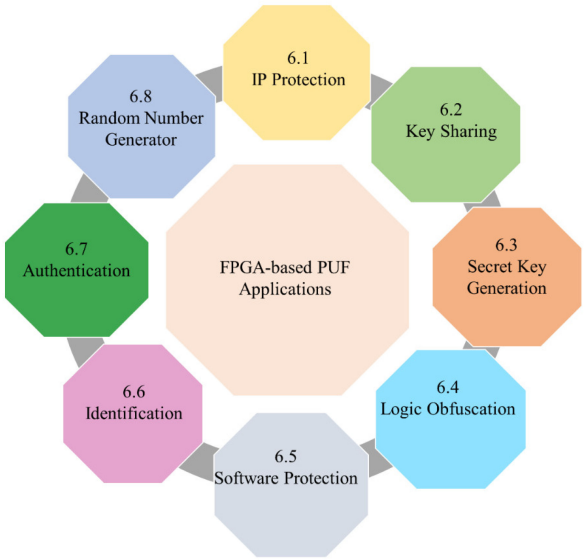

Figure 5. Overview of FPGA-based PUF applications.

- Critical Field Assessment: In the final section of our work, we provide a critical evaluation of the current state of the field. We not only highlight critical issues but

also bring attention to those that remain unresolved. Furthermore, we offer potential solutions, paving the way for further advancements in the field of FPGA-based PUFs. Our research aims to provide clarity and direction, shedding light on the path forward for this critical technology.

1.3. Paper Organization

The remaining sections of the paper are organized as follows. Putting particular emphasis on a key idea that forms the basis of PUF comprehension, we present the types of PUFs and their implementations in Section 2 of our paper. FPGA-based PUF implementation of different types is discussed in Section 3. Section 4 defines the metrics and criteria used to evaluate and compare different FPGA-based PUF designs. It also explains the importance of these metrics in assessing the performance and security of PUFs. We thoroughly explore the applications of FPGA-based PUFs in various domains for security purposes, as illustrated in Figure 5. Section 5 presents the various applications of FPGA-based PUF designs. Section 5 defines the metrics and criteria used to evaluate and compare different FPGA-based PUF designs. It also explains the importance of these metrics in assessing the performance and security of PUFs. Comparative analysis of FPGA-based PUF designs based on these metrics is presented in Section 6. Section 7 presents the discussion and potential future developments for the entire field. We offer solutions for coping with the field's rapid development in Section 8, which concludes the paper.

2. Physical Unclonable Functions (PUFs)

PUFs are hardware security structures that leverage the random and unique physical variations inherent in semiconductor devices during the manufacturing process. The manufacturing process introduces inherent variations in the physical properties of transistors, capacitors, resistors, and other components on the chip. As a result, even chips produced on the same production line within the same batch will have minute differences in their electrical characteristics. These variations are uncontrollable and arise due to imperfections and process fluctuations, making them difficult to replicate or clone intentionally. These small but significant variations can be exploited to create a unique response or identifier for each individual chip. When a challenge is presented to the PUF, it generates a response based on its inherent variations. Since the variations are practically impossible to reproduce with high precision, cloning or counterfeiting the PUF response becomes infeasible. PUFs have gained popularity in recent years as a cost-effective and efficient way to enhance hardware-based security. They find applications in various domains, such as device authentication, secure key generation, secure booting, anti-counterfeiting measures, and cryptographic operations.

2.1. Classification of PUFs

PUFs can be classified based on fabrication method and security strength [35], as shown in Figure 6.

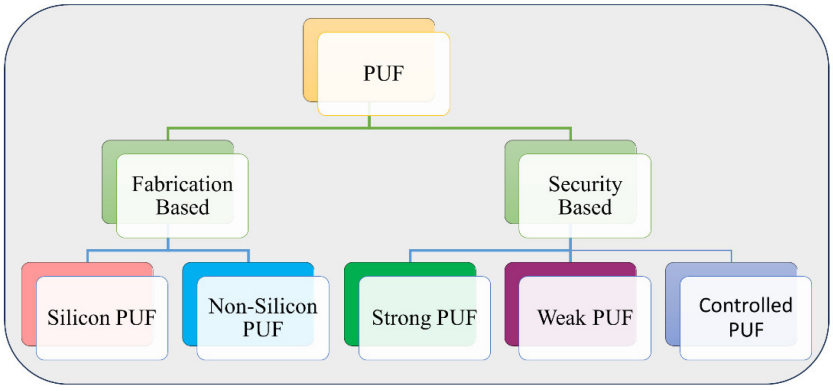

Figure 6. Different types of PUFs (controlled PUF) (adapted from [35]).

2.1.1. Based on Fabrication

PUFs are often categorized into two major groups based on the fabrication method [36]:

Silicon-Based PUFs: Silicon-based PUFs are created using variations in the manufacturing process of silicon chips or integrated circuits [5]. These variations are uncontrollable and unique to each device, making them suitable for generating device-specific and unpredictable responses. Silicon PUFs are widely used and are generally considered highly secure due to the inherent manufacturing variations.

Non-Silicon PUFs: Non-silicon PUFs encompass all other types of PUFs that are not based on silicon, such as optical, magnetic, or micro-electro-mechanical systems (MEMS) PUFs. These PUFs leverage variations in properties other than silicon, such as light, magnetism, or mechanical characteristics, to generate unique responses. Non-silicon PUFs may have different characteristics and security properties compared to silicon-based PUFs, depending on the underlying physical properties they exploit [5].

This classification helps differentiate between PUFs based on the materials and properties they utilize for generating unique responses.

2.1.2. Based on Security

PUFs can also be categorized using security characteristics, particularly those that relate to the quantity of obstacles and the reachability of solutions to the outside world. This classification aids in determining the degree of security and privacy offered by a PUF implementation. According to this criterion, there are two main categories:

Strong PUFs (sPUFs): High levels of security are offered by strong PUFs. sPUFs have many unique responses they can produce, which is known as the "responses space". It is appropriate for applications in which a range of cryptographic keys or authentication tokens are required because each challenge yields a distinct response. Strong PUFs often prevent outsiders from directly accessing the responses they produce. They are kept private and can be applied to cryptographic applications, secure key creation, and device authentication [37].

Weak PUFs (wPUFs): Strong PUFs may provide a higher level of security than weak PUFs. There might be a cap on how many distinct responses wPUFs can provide. A limited response space could be the result of challenges. The outside world might have access to any or all of the responses produced by weak PUFs [38,39]. This may be a trait of some implementations which could in some circumstances compromise their security. It is vital to remember that a PUF's security cannot be determined in absolute terms by its designation as strong or weak.

Controlled PUFs: The decision between strong and weak PUFs depends on the application's particular security requirements. While weak PUFs may be used in less security-critical situations where cost, efficiency, or accessibility considerations are important, strong PUFs are often favored for applications requiring a high level of protection. A full security evaluation that considers the specific use case, prospective threats, and the acceptable degree of risk should be used to determine whether to utilize strong or weak PUFs [40,41]. The PUF responses must be safeguarded and used safely in applications. Hence, it is imperative to implement additional security protocols.

3. FPGA-Based PUF Implementations

FPGA-based PUF implementations exhibit diverse types, each characterized by its distinctive methodology for generating hardware-derived identifiers that are inherently unique. FPGA-based PUF designs can be categorized into three broader groups: delay-based PUFs, memory-based PUFs, and combined PUFs implemented on FPGAs. Here are several prevalent categories of these FPGA-based PUF implementations:

3.1. Delay-Based PUFs

One kind of PUF, called a delay based PUF, uses changes in signal delays within a digital circuit to produce distinctive and unpredictable responses [42–44]. When it is

critical to have a hardware-based root of trust that is challenging to replicate or copy, delay-based PUFs are useful tools for hardware security. To ensure their effectiveness, however, proper design and management are required. Environmental factors and the specific implementation can have an impact on their performance and reliability.

3.1.1. Arbiter PUF

Each integrated circuit (IC) has a distinct delay characteristic due to the manufacturing variances of transistors and wires. Lee et al. [45] exploited this feature to create arbiter-based PUF or multiplexer (MUX) PUFs, which are secret information specific to each IC.

The arbiter PUF's goal is to consciously induce a race condition between two silicon chip digital paths. It consists of two chains of switch blocks (multiplexers) representing the two delay pathways, with an arbiter block at the conclusion of each chain [46–48]. According to Figure 7, the switch block can be in one of two configurations, straight if the challenge bit is 0, or crossing if it is 1.

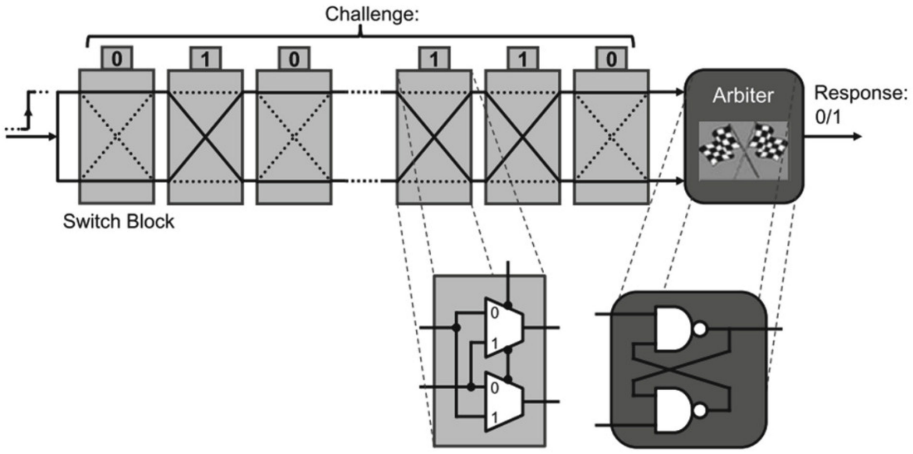

Figure 7. First structure of Arbiter PUF [45].

The two outputs from the stage before and one bit from the challenge make up the three outputs of each switch block. The outputs of the final switch block are connected to the arbiter block, which decides which signal arrived first, while the inputs of the first switch block are connected to a common enable signal. Based on this outcome, the arbiter creates a single bit known as the response bit.

3.1.2. XOR Arbiter PUF

By XORing their output into a single answer bit, the XOR arbiter PUF [9] combines many rows of the fundamental arbiter PUFs, as seen in Figure 8. This increases the circuit's defense against modeling attacks [49,50]. The more rows there are, though, the more faults this method adds into the PUF answer.

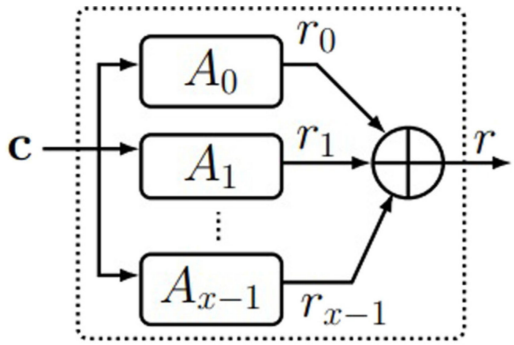

Figure 8. XOR arbiter PUF [9].

3.1.3. Free-Forward Arbiter PUF

Recent proposals for a feed-forward XOR PUF, which combines feed-forward APUF and XOR PUF, were from Avvaru and Parhi [51] and Avvaru et al. [52], respectively. FFXOR PUF employs FF APUF as a new component in place of APUF, which was a part of XOR PUF. In comparison to the traditional XOR PUF, the FFXOR PUF has demonstrated strong dependability, uniqueness, and resilience to attacks, according to claims by Refs. [51,52]. However, the safety and reliability features of this suggested PUF have not been proposed or examined in any publication. The feed-forward XOR PUF's main architecture is depicted in Figure 9.

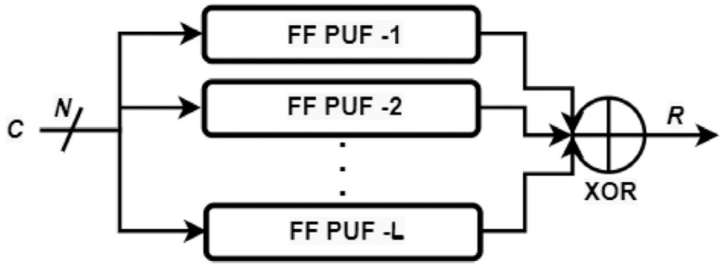

Figure 9. Feed-forward XOR PUF [52].

3.1.4. Ring Oscillator PUF

Ring oscillator PUF (RO PUF), an alternative PUF design as shown in Figure 10 to the fundamental arbiter PUF and its derivatives, was first put out by Suh and Devadas [9] and is based on the delay difference of identical electrical paths.

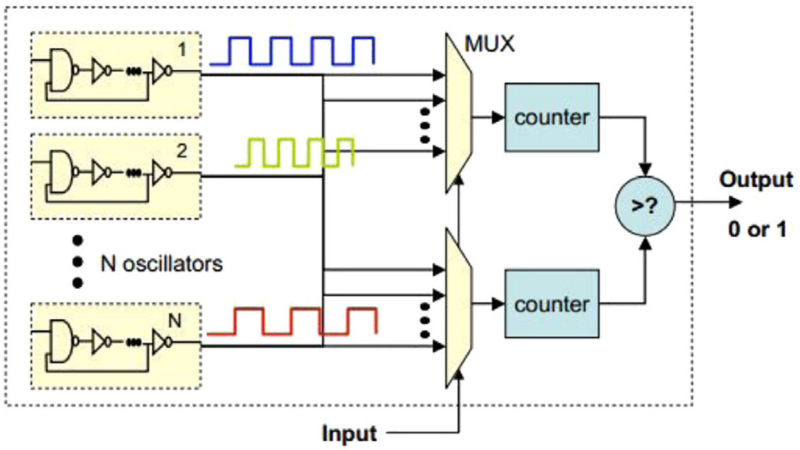

Figure 10. Ring-oscillator-based PUF circuit [9].

N identically arranged delay loops—also known as ring oscillators (ROs)—two multiplexers, two counters, and an arbiter make up a standard RO PUF [53–55]. Theoretically, each RO oscillates at the same frequency, but in practice, each RO oscillates at a slightly variable frequency due to manufacturing differences and environmental factors. A pair of these N ROs must be chosen in order to produce a one-bit answer. The input (challenges) applied to both MUX and a comparison of the frequency of the chosen RO pair serve to determine this choice.

3.1.5. Configurable RO PUF

The first reconfigurable ring oscillator PUF (CRO PUF) was introduced in [56] with the goals of lowering response noise and increasing the number of CRPs of the fundamental RO PUF. In order to determine if the inverter will be chosen as a member of the RO, a

multiplexer has been added after each stage of the RO, as illustrated in Figure 11. Each MUX chooses one output of the two inverters based on the input selection bit. Eight combinations are thus feasible for RO with three steps.

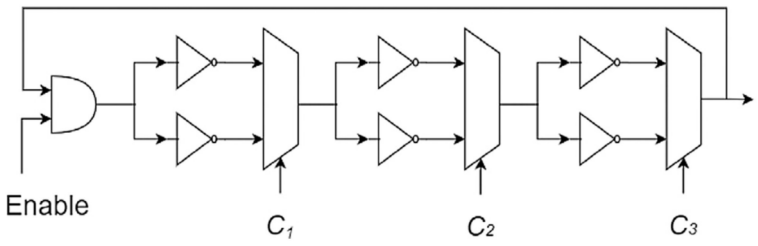

Figure 11. Configurable RO PUF [56].

Another RO PUF structure, referred to as configurable RO PUF or flexible RO PUF, was suggested by Gao et al. [57] based on the same concept. The selection of an inverter from the ring is made in accordance with the input selection bit. Inverters are discarded if the bit is 0; else, they are used in the ring.

3.1.6. Glitch PUF

Suzuki et al. [58] first suggested this in 2010. The glitch PUF architecture makes use of glitches that exhibit nonlinear behavior as a result of gate delay variation and gate pulse propagation [59,60]. Using a delay line to create distinct and random bits, combinational circuit errors can be identified. The complete glitch PUF structure is shown in Figure 12.

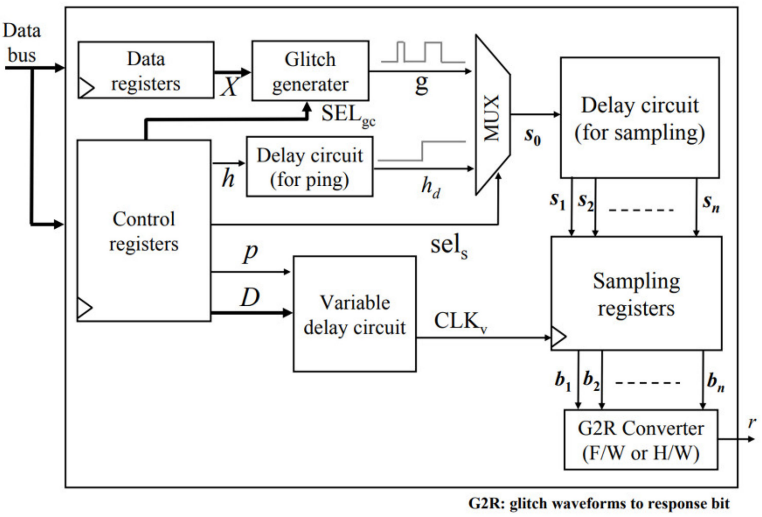

Figure 12. Whole structure of glitch PUF [58].

3.1.7. Bistable Ring PUF

The RO PUF and the bistable ring PUF (BR PUF) [61]; both include inverter rings, but the BR PUF maintains a stable state throughout time. An odd number of inverter gates make up a RO PUF. Instead of oscillating, the BR PUF circuit has two types of potential states: "101010…" and "010101…", formed by an even number of inverter gates [61–64]. Figure 13 displays the schematic diagram of one such PUF.

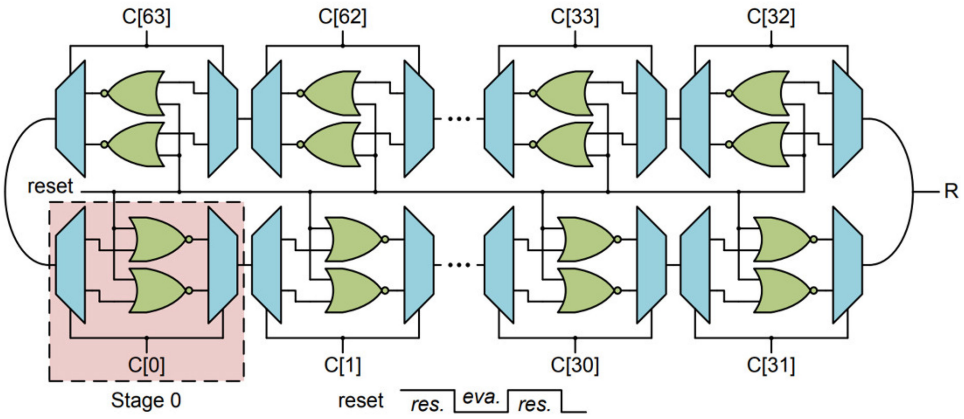

Figure 13. Bistable ring PUF [61].

3.1.8. PLFSR PUF

The Pseudo-LFSR PUF (PLFSR PUF) [65] is based on the Linear Feedback Shift Register (LFSR), however, instead of a shift register, it consists of huge combinational logic [66,67]. Figure 14 displays the PLFSR PUF's schematic diagram. The PLPUF provides a changeable ID, outputs a long bit with efficiency, and has a reasonably compact PLPUF circuit.

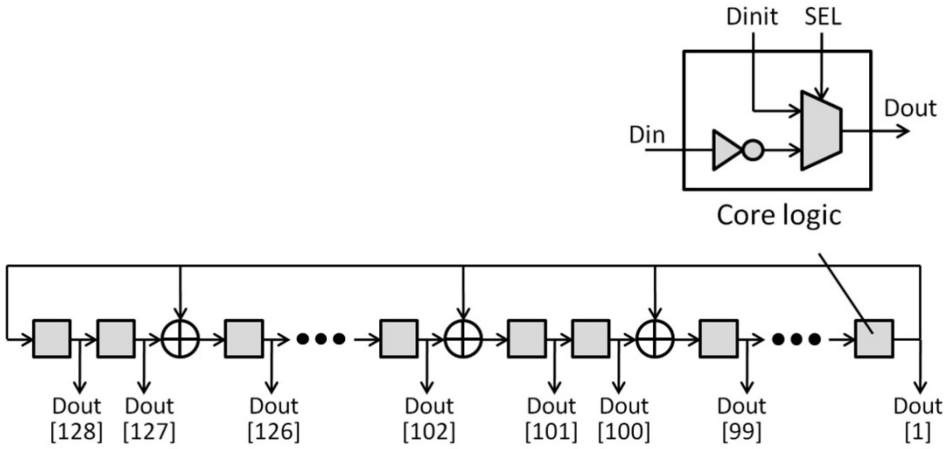

Figure 14. Pseudo-LFSR PUF [65].

3.2. Memory-Based PUFs

The starting state of the memory structures affects the response that the memory-based PUFs produce. The structures are initially in an unstable state at power-up, and the reaction reflects the stabilization of the structures brought about by an external data signal input [68]. This particular PUF family primarily consists of the following.

3.2.1. SRAM PUF

Static random-access memory, or SRAM PUF, was proposed by Guajardo et al. [19] as the first intrinsic PUF construction based on the power-up state of an FPGA's SRAM memory. No adjustments to the production process are necessary. The static–noise margin (SNM), which necessitates a memory cell to change its logical value, is the basis of this system. Two stable states result from the logical construction of an SRAM cell, which is two cross-coupled inverters [69]. A SRAM cell's initial value, 0/1, is randomly and independently assigned by the SNM during startup. The SRAM cell's manufacturing process is to blame for this randomness. These PUFs make use of the conventional SRAM memory in digital chips' random power-up behavior. Each memory cell in the SRAM block, which is made up of an array of memory cells, has the capacity to store one binary digit.

According to Figure 15, an SRAM cell logically consists of two cross coupled inverters and two access transistors for reading and writing data.

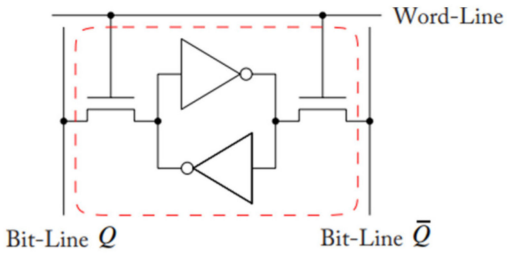

Figure 15. SRAM PUF [19].

3.2.2. Flip-Flop PUF

In 2008, Maes et al. suggested reading the startup values of conventional flip-flops (FFs) on FPGAs, which is equivalent to the SRAM PUFs [70]. Uninitialized flip-flops, like SRAM cells, are in an undefined state that is influenced by manufacturing process differences and can be employed as PUF response. Flip-flop PUFs are based on the ability to stop the initialization of the flip-flops' power-up values on Xilinx FPGAs, but each response must first be generated before the device can be used. Nevertheless, Gu et al. [71] proposed an effective, scalable PUF ID generator architecture on Xilinx Spartan-6 FPGA that needs two cross-coupled NAND gates, two D flip-flops, and one multiplexer to generate the single response bit.

3.2.3. Butterfly PUF

In 2008, Kumar et al. put out the Butterfly PUF [72], which consists of two cross-coupled latches and attempts to replicate the starting behavior of cross-coupled inverters in an SRAM cell. In order to create a Butterfly PUF as shown in Figure 16, the inverter in the SRAM PUF is essentially replaced by a latch or flip-flop. A random state will be produced utilizing the clear (CLR) and preset (SET) functionalities based on the physical (delay) mismatch between the latches and the cross-coupling connection [73]. A power-up is required in the case of an SRAM cell, however it is not required for the butterfly PUF cell to provide a response.

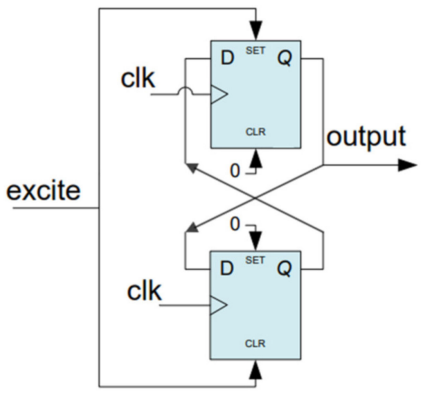

Figure 16. Butterfly PUF [72].

3.3. Combined PUFs

The performance metrics and security of PUFs are improved by combining several PUF primitives, such as hybrid [74] and composite PUFs. These methods use the advantages of various PUFs to strengthen each one's flaws and offer a more reliable security solution.

3.3.1. Hybrid PUF

The goal of a hybrid PUF design is to increase the uniqueness and randomness of the responses by combining two separate sources of randomness (i.e., smaller PUF units) in a PUF [75,76]. The author of [77] rates the effectiveness of two different hybrid PUF design types. The RO PUF and Anderson PUF are merged in the first one, while the SR latch PUF and RO PUF are mixed in the second. A hybrid PUF on FPGA with an area-efficient architecture was recently disclosed in [33]. To increase uniqueness and entropy properties, it mixes units of traditional RS latch PUF and APUF.

3.3.2. Composite PUF

Sahoo et al. presented the composite PUF in 2011 [78] in order to improve quality measures employing RO and arbiter PUFs, as shown in Figure 17. First, m-bit sub challenges are created from the applied master challenge. Then, each sub-challenge component is applied to n separate RO PUFs. The combined output of the ROPUFs serve as an internal challenge for the APUF, which ultimately produces the final response [79]. In this case, the reactions of the RO PUFs regulate the path switching activities of the APUF. Additionally, Ref. [80]'s writers analyze the effectiveness of several composite PUF design types.

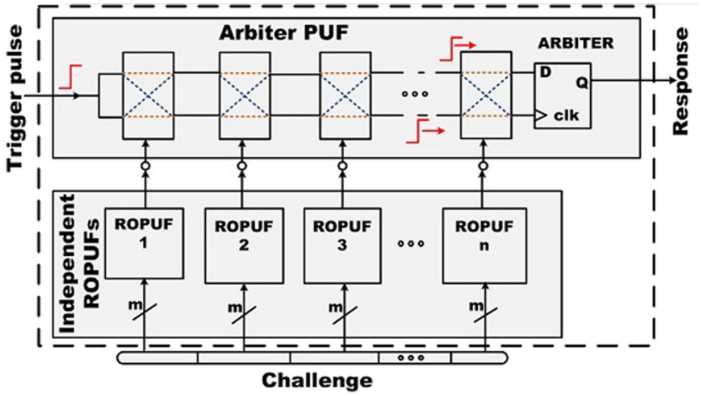

Figure 17. Composite PUF [78].

3.3.3. Double Arbiter PUF

In order to increase the uniqueness of arbiter PUF on FPGAs where the place and route are strictly constrained, Machida et al. [81] proposed a DAPUF in 2015. When compared to the original Arbiter PUF, the difference in wiring bias can be cancelled out in the DAPUF. Additionally, in that work, the authors examined various double arbiter PUF variants, including 2-1, 3-1, and 4-1 DAPUFs, which contain two, three, and four duplicated APUFs, respectively. For instance, the 2-1 DAPUF in Figure 18. Be aware that every DAPUF type produces a response by XORing a number of one-bit outputs [82–84].

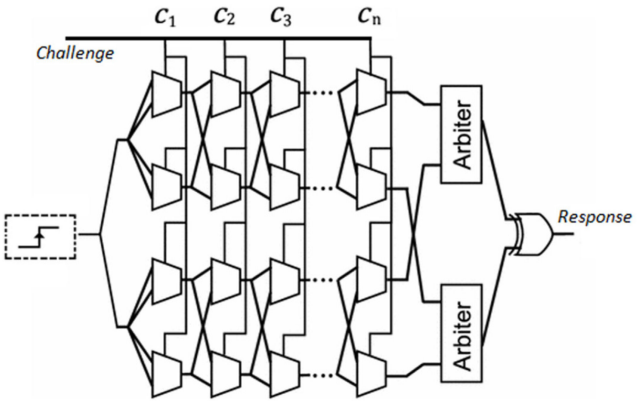

Figure 18. Double arbiter PUF [81].

3.4. FPGA-Based PUF Advantages and Challenges

It is important to note that the PUF type selection should be in accordance with the application's specific needs, taking into account issues like security, resource limitations, and the potential dangers of various sorts of assaults. To deal with the problems that come with each form of PUF and guard against potential weaknesses, security measures and countermeasures should be implemented. Each type of PUF has its corresponding advantages and challenges, presented in Table 1:

Table 1. FPGA-based PUF advantages and challenges.

Types	Design	Ref.	Advantages	Challenges
Delay based PUFs	Arbiter PUF	Morozov et al. [43]	Identical instantiation of building blocks may not be necessary.	Requires symmetric routing in a building block.
		Papakonstantinou et al. [85]	-	Pair of multiplexers in an APUF needs to be identical.
	RO-PUF	Morozov et al. [43]	Does not require symmetric routing in a building block.	Building blocks require identical instantiation.
		Xin et al. [86]	-	RO PUFs is that they require one pair of ring oscillators per bit of output. Therefore, in order to collect enough output bits for a safe security level, a large number of ring oscillators is needed.
		Papakonstantinou et al. [85]	Easier evaluation of entropy and higher reliability than a simple arbiter PUF.	It has slower response, though, it requires larger area and consumes more power.
	Butterfly PUF	Morozov et al. [43]	Identical instantiation of building blocks may not be necessary.	Requires symmetric routing in a building block.
		Papakonstantinou et al. [85]	Choosing a settling time for the CRPs makes the design of a BR-PUF easier since the symmetry of the layout will not be so necessary.	Pair of latches in a BPUF cell needs to be identical.
			Since the difference between intra and inter-chip variation increases with longer settling times, the identification and authentication become more efficient.	Experimental results show that there is a trade-off between reliability and uniqueness. Choosing a short settling time favors reliability whereas longer settling times favor uniqueness.
Memory based PUFs	Latch PUF		-	LPUF has some RS latches that generate inconsistent (random) numbers (i.e., "random latches"). This randomness causes a problem in that the reliability of the response is reduced.
			-	The response bits become lower as the number of random latches increases, which reduces the variety and entropy of responses.
	SRAM PUF	Sklavos et al. [87]	SRAM PUFs require less area and they are easily implemented to FPGAs.	-
			Already existing SRAM of the device can be used for its construction.	-
		Nam et al. [88]	SRAM PUF has the advantages of a rapid response and a small area.	There is a disadvantage related to the number of CRPs required for authentication of an entity.

Table 1. *Cont.*

Types	Design	Ref.	Advantages	Challenges
Combined PUFs	Hybrid PUF	Devika et al. [76]	Combines the advantages of both Arbiter PUF and Butterfly PUF.	Since delay arising out of manufacturing variations are beyond the control it can be exploited to generate keys that are more unique than those produced by normal arbiter PUF designs.
	Composite PUF	Sahoo et al. [80]	Utilizes smaller PUFs as design building blocks to create a "Composite PUF" with larger challenge-space and superior performance at reasonable resource overhead.	The composition is useful if at least one of the component PUFs possesses an unbiased response.
	CRO-PUF	Miskelly et al. [89]	Main advantage of CRO over RO is that it is much more efficient in terms of space and component usage.	Traditional CRO architecture is vulnerable to ML attacks.

4. PUF Quality Metrics

The evaluation of PUFs is typically based on the collected challenge–response pairs (CRPs) from multiple instances of the same PUF type on a specific platform. These CRPs serve as the foundation for assessing the PUF's performance and security characteristics. Evaluating the performance of a physical unclonable function (PUF) is crucial to determine its effectiveness and suitability for specific applications. The metrics mentioned in Figure 19 provide a comprehensive framework for assessing the performance of PUFs. Here is an overview of each of these metrics:

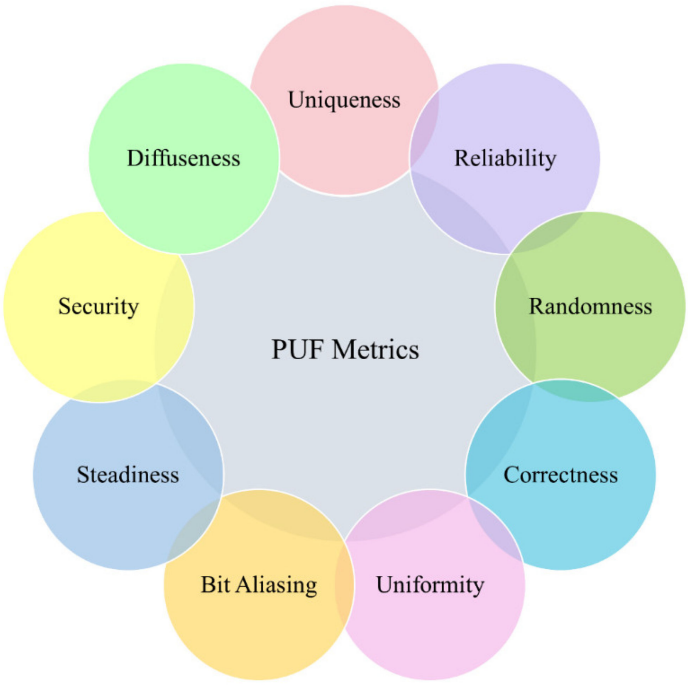

Figure 19. PUF quality metrics.

Uniqueness: Uniqueness measures the extent to which the responses generated by different instances of the same PUF type are distinct from each other. High uniqueness is crucial for ensuring that each PUF instance produces a unique and unclonable response, enhancing security and device authentication.

The uniqueness (HD_{INTER}) is represented as the average inter-chip hamming distance (HD) across k devices provided by (1) if X_i and X_j are the n-bit responses of the ith and jth chips, respectively, for the identical challenge C.

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(X_i, X_j)}{n} \times 100\% \quad (1)$$

The Hamming distance between the n bit strings X_i and X_j is HD (X_i, X_j), and k is the number of chips (devices). Uniqueness should be at least 50%.

Reliability: Reliability assesses the consistency of a PUF's responses over time and under varying operating conditions (e.g., temperature and voltage fluctuations). High reliability ensures that the PUF can consistently generate the same response for a given challenge, making it suitable for reliable authentication and key generation.

It determines how efficiently a PUF can generate the same response at different operating conditions (ambient temperatures or supply voltages) over a period of time for a given challenge. For the ith chip, the average intra-chip HD is estimated using (2). Then, the reliability of a PUF is defined by (3). Here, X_i is the reference response of the ith chip, $X_{i,t}$ is the response generated by it at time t, and s is the number of responses of same set of challenges. Note that the intra-chip HD (2) can be also used to calculate the bit error rate (BER).

$$HD_{INTRAi} = \frac{1}{s} \sum_{t=1}^{s} \frac{HD(X_i, X_{i,t})}{n} \times 100\% \quad (2)$$

$$Reliability_i = 100\% - HD_{INTRAi} \quad (3)$$

The ideal value of reliability is 100% (i.e., ideal value for HD_{INTRA} is 0%), and the average reliability of k chips can be calculated using (4).

$$Average\ Reliability = \frac{1}{k} \sum_{i=1}^{k} Reliability_i \quad (4)$$

Randomness: Randomness evaluates the unpredictability of the PUF responses. PUFs should generate responses that appear random and are statistically indistinguishable from truly random data. It is a measure of balance between "0"s and "1"s in the response bits of the PUF and measures the randomness. The ideal value is 100% (i.e., perfect balance). This property is important for cryptographic applications and random number generation.

Correctness: Correctness measures the accuracy of the PUF responses. PUFs should generate responses that are correct and match the expected values for the given challenges. It is a measure of correctness of the response under different operating conditions. The ideal value is 100%. Ensuring correctness is essential for reliable device authentication.

Uniformity: Uniformity assesses how evenly the response distribution is across the entire response space. It measures whether responses are uniformly distributed or if there are clusters or gaps. It is a measure of the proportion of zeros and ones across the whole of the response set of the PUF and uses Equation (5)

$$Uniformity_i = \frac{1}{n} \sum_{j=1}^{n} u_{i,j} \times 100\% \quad (5)$$

where $u_{i,j}$ is the jth bit of the n-bit response of the ith chip. The ideal value of uniformity is 50%. Uniformity is important to prevent bias in response selection and to ensure that the entire response space is effectively utilized for security purposes.

Bit Aliasing: Bit aliasing occurs when two or more challenges result in the same response. It is a measure of how many challenges map to the same response. The is calculated using Equation (6):

$$Bit - aliasing_j = \frac{1}{k}\sum_{i=1}^{k} b_{i,j} \times 100\% \quad (6)$$

where k is the number of chips/devices and $b_{i,j}$ is the jth bit of the n-bit response of the ith chip. The ideal value of bit aliasing is 50%.

Low bit aliasing is desirable because it enhances the uniqueness of responses, reducing the risk of collisions and vulnerabilities in authentication and key generation.

Steadiness: Steadiness measures the stability of PUF responses over time and in the face of environmental variations. All the responses should be identical when the same challenges feed to a PUF. Ideally, the value of intra-HD should be 0%.

Security: This metric evaluates the PUF's resilience against various attacks, including modeling attacks, machine-learning-based attacks, and physical attacks aimed at extracting or manipulating PUF responses. PUFs should be resistant to these attacks to maintain their security.

Diffuseness: The difference between the uniqueness and the diffuseness is that the uniqueness is determined across several chips, while the diffuseness is defined within a single chip among many possible IDs.

These performance metrics collectively help assess the quality and effectiveness of a PUF implementation. Achieving high values for uniqueness, reliability, randomness, correctness, uniformity, and resistance against attacks, while minimizing bit aliasing and ensuring steadiness are essential goals when evaluating PUFs. The evaluation process may involve statistical analyses, modeling, and testing under various conditions to comprehensively assess the PUF's performance and suitability for specific security applications. Additionally, researchers and designers may develop countermeasures to address any identified vulnerabilities or weaknesses in the PUF's performance.

5. Performance Evaluation and Comparative Analysis

Performance evaluation for the various suggested PUF designs has been conducted on several FPGA process technologies, ranging from 150 nm to 28 nm, and has been conducted in terms of uniqueness, uniformity, bit aliasing, and reliability. Additionally, there are considerable differences in the number of FPGA testbeds used for PUF performance evaluation across the relevant literature. Additionally, we compare the performance of current FPGA-based PUF implementations in Table 2 in terms of uniqueness, uniformity, bit aliasing, reliability, area, and response bit length. In this table we have also shown the targeted FPGA board for the PUF implementation and its resilience towards attacks. These tabulated results lead to the following important conclusions:

In terms of uniqueness and uniformity [81,90,91], it is evident that the delay-based arbiter PUF technique beats the others [92–94], but when compared to other arbiter PUF designs, the dependability of delay-based arbiter PUF [91] exceeds them.

In terms of reliability, the standard RO PUF design [95] performs better than the alternative RO PUF design strategies. Depending upon the applications requirements suitable type of the PUF may be selected as in all categories one or more PUFs are designed, which gives more than 99% reliability, as shown in Table 2 also. Additionally, in terms of uniqueness, the customizable based RO PUF design [96] performs better than other RO PUF designs as shown in Table 1. If utilized as powerful PUFs, the common RO PUF designs [91,96–99] are vulnerable to modeling attacks based on machine and deep learning. However, as compared to other RO PUF designs, some of the designs [93,95,100] offer greater modeling robustness.

Table 2. Comparison of PUF evaluation metrics and resource utilizations of FPGA based PUF designs.

Types	Design		Uniqueness	Reliability	Uniformity	Bit-Alias	Area (Total Slices)	Res. Bit Length	Target FPGA	Attack-Resistant	Year
		Ideal Value	**50%**	**100%**	**50%**	**50%**					
Delay based PUFs	Arbiter PUF	Naveenkumar et al. [92]	49.88	99.20	48.74%	-	-	64	Artix-7	ML-based-modeling attack	2022
		Sahoo et al. [101]	45.25	95.93	48.30	-	-	90	Spartan-3	-	2015
		Maiti et al. [91]	7.20	99.76	55.69	19.57	-	128	Virtex-5	-	2013
		Machida et al. [81]	4.70	99.32	54.78	-	177	128	Virtex-5	-	2015
		Anandakumar et al. [32]	44.30	96.00	48.45	-	234	256	Spartan-6	-	2017
		Hori et al. [102]	36.75	98.48	42.34	-	-	128	Virtex-5	-	2010
		Mahalat et al. [94]	51.34	97.57	57.64		132	64	Artix-7	ML-based-modeling attack	2021
		Anandakumar et al. [49]	48.69	99.41	50.73	-	279	64	Artix-7	ML-based-modeling attack	2022
		Maiti et al. [91]	47.24	99.14	50.56	50.56	-	511	Spartan-3E	-	2013
		Anandakumar et al. [32]	47.13	99.16	50.61	-	82	256	Spartan-6	-	2017
		Merli et al. [97]	48.51	98.28	-	-	512	128	Spartan-3E	-	2010
		Anandakumar et al. [93]	48.91	97.91	49.55	49.55	107	256	Artix-7	ML-based-modeling attack	2022
		Habib et al. [103]	48.30	97.88	50.13	51.80	747	283	Spartan-3E	-	2013
	RO-PUF	Yu et al. [104]	47	-	-	-	420	64	Spartan-3E	-	2012
		Lee et al. [98]	50.1	-	49.4	-	-	64	Artix-7	-	2018
		Dang et al. [96]	50.23	95.92	52.64	-	-	32	Artix-7	-	2022
		Zhang et al. [105]	49.33	95.45	49.50	-	186	136	Virtex-5	ML-based-modeling attack	2018
		Choudhury et al. [99]	47.40	-	49.20	49.10	-	124	Artix-7	-	2017
		Rabiei et al. [95]	48.49	99.55	50.99	-	210	256	Spartan-6	Modeling and MITM attack	2022
		Karmakar et al. [106]	49	85.95	50	-	626 (LUT)	8	Artix-7	-	2023
	Butterfly PUF	Huang et al. [54]	48.74	98.91	49.3	49.2	12 (LUT)	128	Virtex-7	-	2023
		Kumar et al. [72]	43.16	96.20	-	-	130	50	Virtex-5	-	2008
		Anandakumar et al. [32]	48.10	99.19	50.20	-	54	256	Spartan-6	-	2017
	Latch PUF	Ardakani et al. [107]	49.32	98.80	44.65	44.65	128	127	Spartan-3	-	2018
		Anandakumar et al. [93]	49.47	98.29	51.02	51.02	101	256	Artix-7	ML-based-modeling attack	2022
Memory based PUFs		Habib et al. [108]	49.24	98.87	-	-	324	256	Spartan-6	-	2015
		Stanciu et al. [109]	34.73	92.00	-	-	-	-	Spartan-6	-	2016
	SRAM PUF	Guajardo et al. [19]	49.97	88.00	-	-	-	128	-	-	2007
	DFF PUF	Khan et al. [110]	50.2	97.89	-	49.613	-	128	Artix-7	-	2020
	Hybrid PUF	Vega et al. [111]	48.59	97.89	50.97	-	-	1024	Artix-7	ML-based-modeling attack	2023
		Khoshroo et al. [77]	39.63	93.63	48.30	25.20	-	128	Virtex-2	-	2013
		Anandakumar et al. [33]	49.41	99.22	50.09	50.09	257	256	Spartan-6	ML-based-modeling attack	2020
		Tanamoto et al. [112]	32.52	96.96	55.66	-	-	256	Spartan-6	-	2017
Combined PUFs	Composite PUF	Sahoo et al. [90]	36.87	98.85	54.76	-	-	64	Spartan-3	ML-based-modeling attack	2015
		Sahoo et al. [80]	49.04	97.48	50.07	-	1051	-	Spartan-3	Chosen-challenge attack	2014
	Multi-PUF	Ma et al. [113]	40.60	-	30.03	-	-	32	Artix-7	ML-based-modeling attack	2018
	DAPUF	Machida et al. [81]	50.24	88.20	53.94	-	436	128	Virtex-5	ML-based-modeling attack	2015
	FF-APUF	Gu et al. [114]	41.53	95.50	-	-	2816	64	Artix-7	ML-based-modeling attack	2021
Cascaded PUF	CaPUF	Nassar et al. [115]	55.63	92.54	50.06	-	5723 (LUT)	16	Virtex-7	ML-based-modeling attack	2022
Configurable PUF	CT-PUF	Zhang et al. [100]	~49	~99	~48	-	731 (LUT)	64	Zedboard	ML-based-modeling attack	2022
	CRO-PUF	Jeeru et al. [116]	35.78	99.21	49.86	45	-	128	Spartan-3E	-	2019
Feedback-based Lightweight PUF	FLAM-PUF	Wu et al. [117]	49.85	96.58	49.99	-	(1955) Gate Equivalent	128	-	ML-based-modeling attack	2022

The uniqueness of SRAM PUF [19] and DFF PUF [110] for memory-based PUFs is equivalent to that of other delay-based PUF designs and comes very close to the ideal value (50%) for this type of PUF. Additionally, in terms of dependability, the compact implementation of the FPGA-based PUF design [32] exceeds all other memory-based PUF designs. When compared to other memory-based PUF designs, Ref. [93] offers the best performance in terms of uniformity and bit-aliasing.

For combined PUFs, the DAPUF design [81] outperforms other combined PUFs in terms of uniqueness, while the [33,112] exceed other composite and hybrid PUF designs in terms of dependability and uniformity. When compared to other combination PUFs, the composite and hybrid PUFs offer greater modeling robustness in terms of security.

Other than those already mentioned, several other types of PUFs have been reported, including cascaded PUF [115], configurable PUF [100], and feedback-based lightweight PUF [117], all of which have been shown to perform reasonably well in terms of uniqueness, reliability, and uniformity. However, the reported attack resilience capability is particularly strong against attacks involving machine learning modeling.

One of the most important performance metric characteristics, along with those already mentioned, is reliability and another is security, as the authors discovered while putting together a comparison table of FPGA-based PUF designs in terms of several performance metric parameters. PUFs' reliability is their greatest barrier to practical application, although it is important to note that many efforts [118–120] have been made to overcome this hurdle. An approach based on auxiliary data masking with variable positioning as a defense was put out by Ali Pour et al. [121]. Reproducing the key using the fuzzy extractor approach and analyzing it for the helper data manipulation (HDM) attack is how experimental assessment is carried out. These experimental findings demonstrate that masking with changing location can effectively reduce the likelihood that an HDM attack would succeed, even after many unsuccessful attempts. Zheng et al. [122] proposed a novel dual-state analog PUF (DA PUF) which has been fabricated in 55 nm process. Authors have generated 40,960 bits by using fabricated DA PUF which passes NIST randomness test with the reliability over 99.99%. Authors have also given the range of the working environment for the proposed DA PUF.

As is evident from the table, most of the researchers who have reported attack-resistant mechanisms in their approaches studied the ML-based modelling attack. In their studies, the consequences of that ML-based-attack-resistant PUF structure [123,124] are also studied and reported. Sahoo et al. [124] proposed multiplexer based composition of APUFs also called as MPUF to overcome the challenges towards modeling and statistical attacks and lack of reliability. Authors have also proposed two variants of MPUF viz. cMPUF and rMPUF to improve the robustness against modeling attack. Theoretical results are also validated using MATLAB based simulations of MPUFs. Shi et al. [125] analyzed the structural characteristics of MPUFs and proved that these variants proposed by Sahoo et al. can still be broken and for that authors have proposed two novel modeling attacks namely logical approximation and global approximation. Their experimental results also show that MPUF and its variant can be modeled by the new proposed approximation attack up to the accuracy of 96.85%.

Zhang et al. [126] have proposed a random-set-based obfuscation (RSO) for strong PUFs to resist ML-based attacks. In the proposed approach, a true random number is used to select any two random numbers to obfuscate the challenges and responses of the strong PUF to prevent from the ML-based attacks. Their experimental results demonstrate that the proposed approach has strong resistance to ML-based attacks with low hardware overhead.

6. FPGA Based PUF Applications

FPGAs have found application scenarios for PUFs in various domains due to their unique security features and flexibility. Depending on the PUF class of the embedded chip inside the device, PUFs have been employed to secure a variety of devices. This section

examines the current use cases and application domains for FPGA-based PUFs, as shown in Figure 20.

6.1. IP Protection

Electronic devices have a number of security issues, including component addiction, cloning, reverse engineering, and counterfeiting. As a result, semiconductor businesses sustain significant financial losses. Therefore, it is essential to safeguard an IC design's intellectual property (IP) elements. FPGAs often contain valuable intellectual property (IP). PUFs can be used to protect this IP by ensuring that only authorized users can access and use it. A hardware IP protection scheme based on SRAM PUF was introduced by Guajardo et al. [19] for FPGAs. Zhang et al.'s [127] PUF-based IP protection solution restricts IP execution to only certain FPGA devices and upholds pay-per-device licensing in order to prevent IPs from being duplicated, cloned, or utilized with unlawful integration. To stop third parties from stealing the intellectual property (IP) for the program and the hardware, Zheng and Potkonjak [128] presented a PUF-based firmware tempering protection system. To defend IPs from attacks based on CNN models, Guo et al. [129] presented a PUF-based pay-per-device system. Kumar et al. [72] provide a detailed explanation of a public/private key pair-based strategy for safeguarding the use of intellectual property cores in FPGAs.

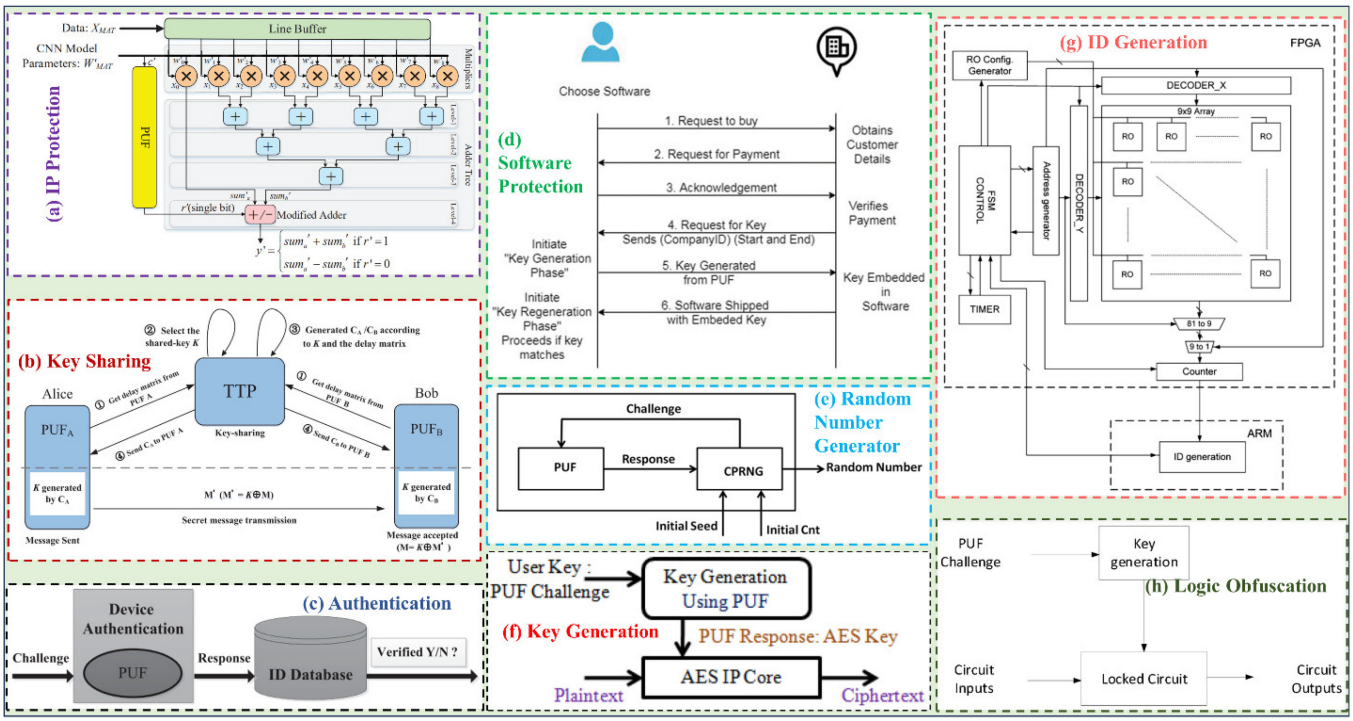

Figure 20. Applications of FPGA-based PUFs in various domains (**a**) IP protection [129], (**b**) key sharing [130], (**c**) authentication [131], (**d**) software protection [132], (**e**) random number generation [16], (**f**) key generation [133], (**g**) ID generation [104] and (**h**) logic obfuscation [134].

6.2. Key Sharing

PUFs can facilitate secure key sharing between devices. FPGAs with PUFs can generate and exchange keys securely, enabling encrypted communication between devices. Due to differences in manufacturing processes, traditional PUFs produce chip-unique keys for each device that are not transferable to another device. However, in multiparty communication between several devices in many IoT security applications, a shared key is necessary. Zhang et al. [130] developed a PUF-based key-sharing mechanism based on the crossover configurable RO PUF to solve this problem. Each level of the RO PUF's inverters has a configurable configuration of problems with inter stage crossover structures. Therefore,

different devices can generate the same response as the shared key for resource-constrained devices with the proper selection of configurations and challenges.

6.3. Secret Key Generation

Cryptographic primitives including message authentication codes, digital signatures, and encryption play crucial roles in many security applications. The secrecy of the key is essential to the cryptographic protection's security. FPGAs with PUFs can generate unique cryptographic keys based on the device's inherent variations. These keys can be used for secure data encryption and decryption. Few practical PUF-based key generator designs on FPGAs have been reported in the literature. These designs leverage the unique properties of PUFs to generate cryptographic keys securely. The PUF-based key generator was fully implemented by the authors [13] using a ring-oscillator PUF, BCH error correcting, and a cryptographic entropy accumulator. On a Xilinx Spartan-6 FPGA, this whole solution uses 1162 slices, of which 82% (952 slices) are taken up by the ROPUF and just 18% by the key generation logic (error-correcting and hashing). Ref. [11] describes an effective PUF-based key generation method employing a per-device setup in FPGAs. This per-device PUF setup step uses 2689 flip-flops and 2199 LUTs on the Xilinx Virtex-7 FPGA to limit the amount of space needed for PUFs and error correction during key generation. Chhabra et al. [133] have integrated with the key-based hardware obfuscation of AES IP core by generating the cryptographic key using PUFs. Their experimental results also show that the proposed model is resilient against side-channel and SAT-based attacks. These types of practical implementations of PUF-based key generators on FPGAs demonstrate their effectiveness in enhancing the security of embedded systems, IoT devices, and other applications where secure key management is crucial.

6.4. Logic Obfuscation

One method for preventing piracy and counterfeiting of integrated circuits is logic obfuscation. By adding built-in locking mechanisms to the original circuit, logic obfuscation aims to conceal the circuit's true purpose. Only when the proper key is applied will the locking circuit revert to being transparent and resume working properly. PUFs can be employed to obfuscate the logic of an FPGA design. By using PUF-generated values as part of the logic, it becomes challenging for attackers to reverse-engineer or understand the design. Chhabra et al. [133] have integrated with the key-based hardware logic obfuscation of AES IP core by generating the cryptographic key using PUFs. Their experimental results also show that the proposed model is resilient against side-channel and SAT-based attacks. In order to increase obfuscation while minimizing space and power overhead, authors also investigate PUFs as a viable seed for pseudo-random number generators (PRNGs). To further provide stronger authentication and data encryption in IoT devices, the PUF-based obfuscation model is applied to the advanced encryption standard (AES) IP core. Quadir et al. [134] proposed the key generation using strong PUFs for a hardware obfuscation scheme to uniquely lock each chip. This PUF-based hardware obfuscation method is evaluated on different ISCAS 85 benchmarks and also analyzed based on performance analysis. In order to generate a chip-dependent license to thwart overbuilding assaults and piracy while also offering pay-per-device licensing service, Jiliang Zhang in [135] configures obfuscation cells of obfuscated design to XOR with the PUF response. In that situation, an attacker is unable to determine the proper license to unlock the counterfeit/overproduced chips because the attacker is ignorant of the secret code used by the obfuscation cells. Therefore, the only person who can activate the chip is the designer, i.e., the person who can grant the license.

6.5. Software Protection

PUFs can enhance software protection by generating unique device-specific keys that are used to encrypt and secure software. This prevents unauthorized copying or tampering with software programs. PUF features are used in a PUF-entangled software security

method to shield software code against malicious tampering before or during run-time. Suresh et al. [132] proposed SRAM-PUF-based key generation for a software licensing model. The generated key validates the veracity of the program installation copy and permits the user to legitimately purchase the software package. A full, end-to-end design cycle to link software code to an FPGA using a RO PUF was put forth by Gora et al. in 2010 [136]. They extract a 128-bit AES key from the PUF upon device startup. Then use this key to decrypt the software's real code, which was previously saved encrypted, and then run the decrypted software. Developers want to prevent unauthorized parties from accessing or altering their software code. Thus, they take precautions to do so.

6.6. Identification

PUFs can be used to uniquely identify FPGAs and devices. This is useful for device tracking, inventory management, and ensuring that only authorized devices are used in a network. PUFs' physical unclonability trait makes using them for device identification more intriguing for anti-counterfeiting systems. In this case, much like in a biometric identification method, the PUF answer can be used directly for unique identification (chip ID). A server queries various PUF devices during the enrollment phase to compile a database of their CRPs, which it then keeps in a database together with information about the PUF's physical system embedder. The server selects a random CRP from the CRPs kept in the database for the current device and issues challenges to the current PUF device during the identification phase. The identification (ID) of the device is successful if the observed response matches the responses that are stored in its database; otherwise, it fails. On FPGAs, some useful chip identification generators have been proven [104,131,137]. A DRAM-based PUF for chip identification was provided in [131], and the most reliable bits were obtained to be used as a 128-bit identifier. A chip ID creation approach with customizable RO, power-up initialization, and adaptive re-initialization was provided by the authors in [104], which can significantly increase its repeatability. In [137], a PUF-based identification method using Xilinx Spartan-6 FPGA devices is provided. This method uses the absolute values of ROs frequencies and is resistant to aging and working conditions.

6.7. Authentication

FPGAs with PUFs can provide strong authentication mechanisms. The unique responses generated by the PUFs can be used for device authentication, preventing unauthorized access to systems and networks. Since a server once more builds a database of a PUF device's CRPs, PUF-based authentication is essentially comparable to PUF-based identification. The PUF gadget, though, is exposed to unreliable surroundings this time around, where it may be misrepresented by knockoffs. Therefore, if a client wants to confirm the legitimacy of a particular PUF device, it can do so by submitting an authentication request to the server. A challenge to identify the PUF device is then issued by the server in response. The device is considered authenticated if the provided answer corresponded with the pertinent response contained in the server's database. If not, it is regarded as a fake device.

Additionally, certain real-world PUF-based authentication systems have been demonstrated using FPGAs [8,17,21,138–140]. In [21], Rostami et al. presented the slim PUF protocol, which uses pattern matching to authenticate the output of an arbitrator PUF. In Xilinx Virtex 5 FPGAs, this PUF protocol implementation requires 1400 registers and 652 LUTs. In [8], Chatterjee et al. combined the ideas of PUF, IBE, and keyed hash function to demonstrate authentication and key exchange protocol using FPGA. The double arbiter PUF takes up 456 slices of this complete implementation's 1733 total slices on the Xilinx Artix-7 FPGA, while the BCH error-correcting logic takes up 1277 slices. Aysu et al. showed off an FPGA implementation of a protocol that can provide privacy-preserving reciprocal authentication between a server and a restricted device in [141]. In Xilinx Virtex 5 FPGAs, this PUF protocol implementation requires 3543 LUTs, 1275 registers, and eight-block RAMs.

6.8. Random Number Generator

Random numbers are used in cryptographic applications to generate authentication protocols' random nonces, padding bits, encryption keys, and initial parameter values. The majority of the time, these random numbers are generated using a PRNG, a deterministic software technique that mimics randomness. A PRNG uses a stream of random bits created by secret seed values. A PRNG's seed enables an attacker to produce and forecast the full stream of random numbers if they possess it. Therefore, each PRNG seed should be entirely distinct, random, and unpredictable. The PUF answers can be utilized to create a stream of genuine random bits and serve as a secret seed for the PRNG. FPGAs can use PUF responses to generate high-quality random numbers. PUFs have been used for random number generation in several FPGA-based implementations [16,142–144]. The first seed in [16] is produced using PUF and has dynamic refreshing logic to guarantee that the generated random numbers are nonperiodic. This solution uses 352 less resources and achieves 832 Mbps performance on the Xilinx Virtex-7 FPGA. The true random seeds are extracted out of the noise on the SRAMs' startup pattern in [142–144] and used as an input to a nondeterministic random number generator.

7. Challenges and Future Directions

Following an overview and analysis of PUF features, designs, attacks, performance comparisons, and application scenarios in previous sections, we address some challenges and unresolved issues. Although the field of PUFs has expanded greatly over the last ten years while facing many challenges application wise, much more work is still needed to fulfill future demands. We attempt to highlight a few frequent issues with FPGA-based PUF designs and intriguing future research directions in this section.

7.1. Challenges in FPGA-Based PUF Designs

FPGA-based PUF implementation presents a unique set of difficulties. These difficulties may affect PUFs' dependability, security, and usefulness. Some frequent issues with PUF designs using FPGAs follow.

7.2. Variability and Aging

PUFs with an FPGA foundation rely on manufacturing variances within the hardware. The reliability of the PUF may be impacted over time by changes in these variations brought on by aging, temperature swings, and other external conditions.

7.3. Response Stability

Particularly in dynamic situations with temperature and voltage variations, it can be difficult to guarantee consistent responses from FPGA-based PUFs. Unstable responses may result from changes in the FPGA's working circumstances.

7.4. Resource Utilization

The number of logic components, flip-flops, and lookup tables in an FPGA is constrained. It requires a delicate balance to use these resources effectively while still preserving security.

7.5. Randomness Assurance

To produce high-quality random numbers, some applications need PUFs. It is difficult to ensure that the PUF's responses are completely random and objective in terms of statistics.

7.6. Secure Key Storage

Using PUFs for cryptographic purposes requires securely storing PUF-generated keys within the FPGA. Due to the sensitivity of the keys and the potential security hazards, protecting these keys from manipulation or extraction is a very difficult task.

7.7. Testability

Assuring the stability and security of FPGA-based PUFs throughout their lifecycle requires effective testing and validation. It improves confidence in the PUF's behavior, aids in the early detection and resolution of problems, and adds to the general security of systems using PUFs for authentication and cryptographic activities.

7.8. Configuration Management

For PUF-based systems to continue to be reliable, secure configuration management of reconfigurable PUFs is essential. You can increase the system's overall security by putting in place stringent security controls, access restrictions, and encryption procedures to safeguard the PUF's configuration data from unwanted access, manipulation, or misuse.

Hardware design skills, cryptography methods, as well as careful testing and validation, are all necessary to meet these obstacles. For a variety of applications, such as device authentication, key generation, and secure communication, FPGA-based PUF designers must take these difficulties into account while developing reliable and secure solutions.

Future Directions

The development of PUFs based on FPGAs continues to offer intriguing opportunities for improving security, dependability, and versatility in a variety of applications. We can anticipate the following developments and future directions for FPGA-based PUF designs.

7.9. Improved Reliability and Stability

PUFs based on FPGA must be protected against aging and environmental influences in order to maintain long-term stability and reliability. PUFs are employed in a variety of crucial applications, therefore any performance deterioration over time could have serious security repercussions.

7.10. Enhanced Security against Modeling Attacks

One of the key challenges facing PUFs is their vulnerability to modeling attacks, including machine learning-based attacks. To maintain the security of PUFs, advancements in PUF design should focus on bolstering resistance against these types of attacks.

7.11. Resource-Efficient Designs

Developing resource-efficient PUF designs is critical for enabling PUF implementation in resource-constrained FPGAs. These resource-efficient PUFs should aim to minimize FPGA resource utilization while maintaining strong security.

7.12. Post-Quantum Security

In a world in which quantum computing threatens conventional cryptographic systems, quantum-resistant PUFs have the potential to offer strong security. Researchers and developers can produce PUFs that are still useful and secure in a post-quantum cryptography environment by fusing novel design methodologies, quantum-safe algorithms, and thorough security assessments.

7.13. Standardization

Industry-wide efforts to standardize FPGA-based PUFs foster security, interoperability, and reliability. These standards boost the dependability and efficiency of FPGA-based PUFs in a variety of applications by creating defined guidelines, testing processes, and compliance methods, which helps to create a more secure digital ecosystem.

As FPGA technology advances and the need for robust security solutions grows, FPGA-based PUFs will continue to evolve and play a critical role in enhancing the security, dependability, and variety of embedded systems and secure applications.

8. Conclusions

This article offers a thorough evaluation and comparative analysis of physical unclonable function (PUF) designs based on FPGA. The investigation of FPGA-based PUFs has brought to light their importance in strengthening security, authenticity, and cryptographic applications across numerous fields. The diversity and complexity of these designs have been highlighted in this research by a thorough analysis of various FPGA-based PUF implementations. The delay-based, memory-based, and combined PUFs—each offering certain benefits and security features—are the three categories into which the article has divided FPGA-based PUFs. This paper has presented the benefits and drawbacks of FPGA-based PUFs, which enables the researchers and practitioners to select the appropriate PUF type for their specific applications. The research has also investigated the various metrics to assess the performance of FPGA-based PUFs, such as uniqueness, randomness, dependability, correctness, uniformity, bit aliasing, stability, attack resistance, etc. Though useful in many situations, FPGA-based PUF designs have a number of drawbacks that must be taken into account. Listed below are some typical limitations of PUF designs built on FPGAs:

- Environmental variables including temperature, voltage, and electromagnetic interference can influence PUF reactions, making them less dependable under difficult circumstances.
- Due to manufacturing irregularities, FPGA-based PUFs may display exceptions in their responses, which may affect their dependability and consistency.
- It can be difficult to choose the best design for a given application due to the diversity of available FPGA-based PUF designs, which can prevent standardization.
- PUFs on FPGAs can increase the overall cost of a device or system, which makes them less cost-effective for particular applications.
- When integrating PUFs into already-existing FPGA designs, compatibility, design changes, and potential interaction with other functions may need to be carefully taken into account.
- Some FPGA-based PUF designs could be difficult to scale in order to support applications that demand large numbers of PUF instances.

When contemplating the use of FPGA-based PUFs in a particular application, it is critical to be aware of these constraints and carefully weigh the trade-offs. The challenges associated with FPGA-based PUF designs are also highlighted in this paper. These challenges ensure the requirement of ongoing research in the field of PUD designs for addressing the emerging threats and ensure the long-term reliability of these designed PUFs for real-world applications readiness. The article also focuses on improving security features and robust efficiency while assessing the future directions of FPGA-based PUF designs.

The article concludes with the emphasis upon the enormous potential of FPGA-based PUFs for improving the security feature of various range of applications. This study offers the complete overview of PUF basic understanding, and their types followed by their design metrics, applications, comparison of different types of available FPGA-based PUF designs, making it an invaluable resource for helping the researchers and practitioners in suitable PUF selection for their applications. FPGA-based PUFs are positioned to continue playing a key role in protecting sensitive data and systems from ever-evolving threats along with technological advancements.

Author Contributions: Conceptualization, K.L. and L.R.C.; formal analysis, K.L.; methodology, K.L.; writing—original draft, K.L.; writing—review and editing, K.L. and L.R.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sicari, S.; Rizzardi, A.; Grieco, L.A.; Coen-Porisini, A. Security, Privacy and Trust in Internet of Things: The Road Ahead. *Comput. Netw.* **2015**, *76*, 146–164. [CrossRef]
2. Radomirovic, S. Towards a Model for Security and Privacy in the Internet of Things. In Proceedings of the First International Workshop on the Security of the Internet of Things, Tokyo, Japan, 29 November 2010.
3. Wurm, J.; Hoang, K.; Arias, O.; Sadeghi, A.-R.; Jin, Y. Security Analysis on Consumer and Industrial IoT Devices. In Proceedings of the 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, 25–28 January 2016; pp. 519–524.
4. Shrouf, F.; Ordieres, J.; Miragliotta, G. Smart Factories in Industry 4.0: A Review of the Concept and of Energy Management Approached in Production Based on the Internet of Things Paradigm. In Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management, Selangor, Malaysia, 9–12 December 2014; pp. 697–701.
5. Zerrouki, F.; Ouchani, S.; Bouarfa, H. A Survey on Silicon PUFs. *J. Syst. Archit.* **2022**, *127*, 102514. [CrossRef]
6. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical One-Way Functions. *Science* **2002**, *297*, 2026–2030. [CrossRef] [PubMed]
7. Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S. Silicon Physical Random Functions. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; Association for Computing Machinery: New York, NY, USA, 2002; pp. 148–160.
8. Chatterjee, U.; Govindan, V.; Sadhukhan, R.; Mukhopadhyay, D.; Chakraborty, R.S.; Mahata, D.; Prabhu, M.M. Building PUF Based Authentication and Key Exchange Protocol for IoT Without Explicit CRPs in Verifier Database. *IEEE Trans. Dependable Secur. Comput.* **2019**, *16*, 424–437. [CrossRef]
9. Suh, G.E.; Devadas, S. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In Proceedings of the 44th Annual Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; Association for Computing Machinery: New York, NY, USA, 2007; pp. 9–14.
10. Qureshi, M.A.; Munir, A. PUF-RAKE: A PUF-Based Robust and Lightweight Authentication and Key Establishment Protocol. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 2457–2475. [CrossRef]
11. Usmani, M.A.; Keshavarz, S.; Matthews, E.; Shannon, L.; Tessier, R.; Holcomb, D.E. Efficient PUF-Based Key Generation in FPGAs Using Per-Device Configuration. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2019**, *27*, 364–375. [CrossRef]
12. Assaf, T.; Al-Dweik, A.; Iraqi, Y.; Jangsher, S.; Pandey, A.; Giacalone, J.-P.; Abulibdeh, E.E.; Saleh, H.; Mohammad, B. High-Rate Secret Key Generation Using Physical Layer Security and Physical Unclonable Functions. *IEEE Open J. Commun. Soc.* **2023**, *4*, 209–225. [CrossRef]
13. Maes, R.; Van Herrewege, A.; Verbauwhede, I. PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2012, Leuven, Belgium, 9–12 September 2012; Prouff, E., Schaumont, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 302–319.
14. Anchana, U.K.; Mogireddy, M.; Kadavergu, E.; Singh, S. Design of PUF Based Chaotic Random Number Generator. In Proceedings of the 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichirappalli, India, 5–7 April 2023; pp. 1–7.
15. Dheeraj, A.; Das, P.; Kumar, K.; Kalanadhabhatta, S.; Acharyya, A. Modeling Attacks Resilient Multiple PUF-CPRNG Architecture Design Methodology. In Proceedings of the 2022 IEEE 35th International System-on-Chip Conference (SOCC), Belfast, UK, 5–8 September 2022; pp. 1–6.
16. Kalanadhabhatta, S.; Kumar, D.; Anumandla, K.K.; Reddy, S.A.; Acharyya, A. PUF-Based Secure Chaotic Random Number Generator Design Methodology. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 1740–1744. [CrossRef]
17. Sutar, S.; Raha, A.; Kulkarni, D.; Shorey, R.; Tew, J.; Raghunathan, V. D-PUF: An Intrinsically Reconfigurable DRAM PUF for Device Authentication and Random Number Generation. *ACM Trans. Embed. Comput. Syst.* **2017**, *17*, 17. [CrossRef]
18. Roy, D.B.; Bhasin, S.; Nikolić, I.; Mukhopadhyay, D. Combining PUF with RLUTs: A Two-Party Pay-per-Device IP Licensing Scheme on FPGAs. *ACM Trans. Embed. Comput. Syst.* **2019**, *18*, 1–22. [CrossRef]
19. Guajardo, J.; Kumar, S.S.; Schrijen, G.-J.; Tuyls, P. FPGA Intrinsic PUFs and Their Use for IP Protection. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2007, Vienna, Austria, 10–13 September 2007; Paillier, P., Verbauwhede, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 63–80.
20. Suragani, R.; Nazarenko, E.; Anagnostopoulos, N.A.; Mexis, N.; Kavun, E.B. Identification and Classification of Corrupted PUF Responses via Machine Learning. In Proceedings of the 2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), McLean, VA, USA, 27–30 June 2022; pp. 137–140.
21. Rostami, M.; Majzoobi, M.; Koushanfar, F.; Wallach, D.S.; Devadas, S. Robust and Reverse-Engineering Resilient PUF Authentication and Key-Exchange by Substring Matching. *IEEE Trans. Emerg. Top. Comput.* **2014**, *2*, 37–49. [CrossRef]
22. Roy, S.; Das, D.; Mondal, A.; Mahalat, M.H.; Sen, B.; Sikdar, B. PLAKE: PUF-Based Secure Lightweight Authentication and Key Exchange Protocol for IoT. *IEEE Internet Things J.* **2023**, *10*, 8547–8559. [CrossRef]
23. Sun, D.-Z.; Gao, Y.-N.; Tian, Y. On the Security of a PUF-Based Authentication and Key Exchange Protocol for IoT Devices. *Sensors* **2023**, *23*, 6559. [CrossRef] [PubMed]
24. Yang, K.; Forte, D.; Tehranipoor, M.M. CDTA: A Comprehensive Solution for Counterfeit Detection, Traceability, and Authentication in the IoT Supply Chain. *ACM Trans. Des. Autom. Electron. Syst.* **2017**, *22*, 1–31. [CrossRef]
25. Shamsoshoara, A.; Korenda, A.; Afghah, F.; Zeadally, S. A Survey on Physical Unclonable Function (PUF)-Based Security Solutions for Internet of Things. *Comput. Netw.* **2020**, *183*, 107593. [CrossRef]

26. Yalli, J.S.; Hasan, M.H. A Unique PUF Authentication Protocol Based Fuzzy Logic Categorization for Internet of Things (IOT) Devices. In Proceedings of the 2023 12th International Conference on Software and Computer Applications, Kuantan, Malaysia, 23–25 February 2023; Association for Computing Machinery: New York, NY, USA, 2023; pp. 246–252.
27. Babaei, A.; Schiele, G. Physical Unclonable Functions in the Internet of Things: State of the Art and Open Challenges. *Sensors* **2019**, *19*, 3208. [CrossRef]
28. Al-Meer, A.; Al-Kuwari, S. Physical Unclonable Functions (PUF) for IoT Devices. *ACM Comput. Surv.* **2023**, *55*, 314. [CrossRef]
29. Knowmade Physical Unclonable Functions for Securing Our Digital World. Available online: https://www.knowmade.com/technology-news/semiconductor-news/memory-news/physical-unclonable-functions-pufs-a-short-review-of-innovators-who-are-making-the-digital-revolution-more-secure/ (accessed on 31 July 2023).
30. Physically Unclonable Function—PUF Solution. Available online: https://www.secure-ic.com/products/issp/security-ip/key-management/puf-ip/ (accessed on 12 August 2023).
31. FPGA Industry Worth $19.1 Billion by 2028. Available online: https://www.marketsandmarkets.com/PressReleases/fpga.asp (accessed on 17 August 2023).
32. Anandakumar, N.N.; Hashmi, M.S.; Sanadhya, S.K. Compact Implementations of FPGA-Based PUFs with Enhanced Performance. In Proceedings of the 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), Hyderabad, India, 7–11 January 2017; pp. 161–166.
33. Anandakumar, N.N.; Hashmi, M.S.; Sanadhya, S.K. Efficient and Lightweight FPGA-Based Hybrid PUFs with Improved Performance. *Microprocess. Microsyst.* **2020**, *77*, 103180. [CrossRef]
34. Nalla Anandakumar, N.; Sanadhya, S.K.; Hashmi, M.S. FPGA-Based True Random Number Generation Using Programmable Delays in Oscillator-Rings. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 570–574. [CrossRef]
35. Joshi, S.; Mohanty, S.P.; Kougianos, E. Everything You Wanted to Know About PUFs. *IEEE Potentials* **2017**, *36*, 38–46. [CrossRef]
36. Zhang, J.-L.; Qu, G.; Lv, Y.-Q.; Zhou, Q. A Survey on Silicon PUFs and Recent Advances in Ring Oscillator PUFs. *J. Comput. Sci. Technol.* **2014**, *29*, 664–678. [CrossRef]
37. Strong PUFs: Models, Constructions, and Security Proofs | SpringerLink. Available online: https://link.springer.com/chapter/10.1007/978-3-642-14452-3_4 (accessed on 23 October 2023).
38. Mukhopadhyay, D. PUFs as Promising Tools for Security in Internet of Things. *IEEE Des. Test* **2016**, *33*, 103–115. [CrossRef]
39. Yehoshuva, C.; Raja Adhithan, R.; Nalla Anandakumar, N. A Survey of Security Attacks on Silicon Based Weak PUF Architectures. In Proceedings of the Security in Computing and Communications, Chennai, India, 14–17 October 2020; Thampi, S.M., Wang, G., Rawat, D.B., Ko, R., Fan, C.-I., Eds.; Springer: Singapore, 2021; pp. 107–122.
40. Wachsmann, C.; Sadeghi, A.-R. *Physically Unclonable Functions (PUFs): Applications, Models, and Future Directions*; Synthesis Lectures on Information Security, Privacy, and Trust; Springer International Publishing: Cham, Switzerland, 2015; ISBN 978-3-031-01216-7.
41. Gassend, B.; Clarke, D.; van Dijk, M.; Devadas, S. Controlled Physical Unknown Functions: Applications to Secure Smartcards and Certified Execution. Available online: https://api.semanticscholar.org/CorpusID:9153005 (accessed on 23 October 2023).
42. Morozov, S.; Maiti, A.; Schaumont, P. An Analysis of Delay Based PUF Implementations on FPGA. In Proceedings of the Reconfigurable Computing: Architectures, Tools and Applications, Bangkok, Thailand, 17–19 March 2010; Sirisuk, P., Morgan, F., El-Ghazawi, T., Amano, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 382–387.
43. Morozov, S.; Maiti, A.; Schaumont, P. A Comparative Analysis of Delay Based PUF Implementations on FPGA. 2009. Available online: https://eprint.iacr.org/2009/629 (accessed on 23 October 2023).
44. Zhang, J.; Wu, Q.; Lyu, Y.; Zhou, Q.; Cai, Y.; Lin, Y.; Qu, G. Design and Implementation of a Delay-Based PUF for FPGA IP Protection. In Proceedings of the 2013 International Conference on Computer-Aided Design and Computer Graphics, Guangzhou, China, 16–18 November 2013; ISBN 978-1-4799-2576-6.
45. Lee, J.W.; Lim, D.; Gassend, B.; Suh, G.E.; van Dijk, M.; Devadas, S. A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications. In Proceedings of the 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525), Honolulu, HI, USA, 17–19 June 2004; pp. 176–179.
46. Hemavathy, S.; Bhaaskaran, V.S.K. Arbiter PUF—A Review of Design, Composition, and Security Aspects. *IEEE Access* **2023**, *11*, 33979–34004. [CrossRef]
47. Kulkarni, S.; Vani, R.M.; Hunagund, P.V. Designing of Arbiter PUF for Securing IP and IoT Devices. In Proceedings of the Data Intelligence and Cognitive Informatics, Tirunelveli, India, 8–9 July 2020; Jeena Jacob, I., Kolandapalayam Shanmugam, S., Piramuthu, S., Falkowski-Gilski, P., Eds.; Springer: Singapore, 2021; pp. 131–138.
48. Shariffuddin, S.k.; Sivamangai, N.M.; Napolean, A.; Naveenkumar, R.; Kamalnath, S.; Saranya, G. Review on Arbiter Physical Unclonable Function and Its Implementation in FPGA for IoT Security Applications. In Proceedings of the 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 21–22 April 2022; pp. 369–374.
49. Anandakumar, N.N.; Hashmi, M.S.; Chaudhary, M.A. Implementation of Efficient XOR Arbiter PUF on FPGA With Enhanced Uniqueness and Security. *IEEE Access* **2022**, *10*, 129832–129842. [CrossRef]
50. Becker, G.T. The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2015, Saint-Malo, France, 13–16 September 2015; Güneysu, T., Handschuh, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 535–555.

51. Avvaru, S.V.S.; Parhi, K.K. Feed-Forward XOR PUFs: Reliability and Attack-Resistance Analysis. In Proceedings of the 2019 on Great Lakes Symposium on VLSI, Tysons Corner, VA, USA, 9–11 May 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 287–290.
52. Avvaru, S.V.S.; Zeng, Z.; Parhi, K.K. Homogeneous and Heterogeneous Feed-Forward XOR Physical Unclonable Functions. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2485–2498. [CrossRef]
53. Maiti, A.; Casarona, J.; McHale, L.; Schaumont, P. A Large Scale Characterization of RO-PUF. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 13–14 June 2010; pp. 94–99.
54. Huang, Z.; Bian, J.; Lin, Y.; Liang, H.; Ni, T. Design Guidelines and Feedback Structure of Ring Oscillator PUF for Performance Improvement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**. *Early Access*. [CrossRef]
55. Martínez-Rodríguez, M.C.; Rojas-Muñoz, L.F.; Camacho-Ruiz, E.; Sánchez-Solano, S.; Brox, P. Efficient RO-PUF for Generation of Identifiers and Keys in Resource-Constrained Embedded Systems. *Cryptography* **2022**, *6*, 51. [CrossRef]
56. Maiti, A.; Schaumont, P. Improved Ring Oscillator PUF: An FPGA-Friendly Secure Primitive. *J. Cryptol.* **2011**, *24*, 375–397. [CrossRef]
57. Gao, M.; Lai, K.; Qu, G. A Highly Flexible Ring Oscillator PUF. In Proceedings of the 51st Annual Design Automation Conference, San Francisco, CA, USA, 1–5 June 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 1–6.
58. Suzuki, D.; Shimizu, K. The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes. In Proceedings of the Cryptographic Hardware and Embedded Systems, CHES 2010, Santa Barbara, CA, USA, 17–20 August 2010; Mangard, S., Standaert, F.-X., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 366–382.
59. Glitch PUF: Extracting Information from Usually Unwanted Glitches. Available online: https://www.jstage.jst.go.jp/article/transfun/E95.A/1/E95.A_1_223/_article (accessed on 23 October 2023).
60. Nozaki, Y.; Takemoto, S.; Ikezaki, Y.; Yoshikawa, M. Performance Evaluation of Unrolled Cipher Based Glitch PUF Implemented on Virtex-7. In Proceedings of the 2021 International Symposium on Devices, Circuits and Systems (ISDCS), Higashihiroshima, Japan, 21 March 2021; pp. 1–4.
61. Chen, Q.; Csaba, G.; Lugli, P.; Schlichtmann, U.; Rührmair, U. The Bistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions. In Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, San Diego, CA, USA, 5–6 June 2011; Available online: https://ieeexplore.ieee.org/document/5955011 (accessed on 27 September 2023).
62. Yamamoto, D.; Takenaka, M.; Sakiyama, K.; Torii, N. Security Evaluation of Bistable Ring PUFs on FPGAs Using Differential and Linear Analysis. In Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 7–10 September 2014; pp. 911–918.
63. Chen, Q.; Csaba, G.; Lugli, P.; Schlichtmann, U.; Rührmair, U. Characterization of the Bistable Ring PUF. In Proceedings of the 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 12–16 March 2012; pp. 1459–1462.
64. Xu, X.; Rührmair, U.; Holcomb, D.E.; Burleson, W. Security Evaluation and Enhancement of Bistable Ring PUFs. In Proceedings of the Radio Frequency Identification, New York, NY, USA, 23–24 June 2015; Mangard, S., Schaumont, P., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 3–16.
65. Hori, Y.; Kang, H.; Katashita, T.; Satoh, A. Pseudo-LFSR PUF: A Compact, Efficient and Reliable Physical Unclonable Function. In Proceedings of the 2011 International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 30 November–2 December 2011; pp. 223–228.
66. Ogasahara, Y.; Hori, Y.; Katashita, T.; Iizuka, T.; Awano, H.; Ikeda, M.; Koike, H. Implementation of Pseudo-Linear Feedback Shift Register-Based Physical Unclonable Functions on Silicon and Sufficient Challenge–Response Pair Acquisition Using Built-In Self-Test before Shipping. *Integration* **2020**, *71*, 144–153. [CrossRef]
67. Zhou, T.; Ji, Y.; Chen, M.; Li, Y. PL-MRO PUF: High Speed Pseudo-LFSR PUF Based on Multiple Ring Oscillators. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
68. Bautista Adames, I.A.; Das, J.; Bhanja, S. Survey of Emerging Technology Based Physical Unclonable Funtions. In Proceedings of the 26th Edition on Great Lakes Symposium on VLSI, Boston, MA, USA, 18–20 May 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 317–322.
69. Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions | SpringerLink. Available online: https://link.springer.com/chapter/10.1007/978-3-642-14452-3_1 (accessed on 27 September 2023).
70. Maes, R.; Tuyls, P.; Verbauwhede, I. Intrinsic PUFs from Flip-Flops on Reconfigurable Devices. In Proceedings of the 3rd Benelux Workshop on Information and System Security (WIS-Sec2008), Eindhoven, The Netherlands, 13–14 November 2008.
71. Improved Reliability of FPGA-Based PUF Identification Generator Design | ACM Transactions on Reconfigurable Technology and Systems. Available online: https://dl.acm.org/doi/abs/10.1145/3053681 (accessed on 27 September 2023).
72. Kumar, S.S.; Guajardo, J.; Maes, R.; Schrijen, G.-J.; Tuyls, P. Extended Abstract: The Butterfly PUF Protecting IP on Every FPGA. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, 9 June 2008; pp. 67–70.
73. Xu, X.; Liang, H.; Huang, Z.; Jiang, C.; Ouyang, Y.; Fang, X.; Ni, T.; Yi, M. A Highly Reliable Butterfly PUF in SRAM-Based FPGAs. *IEICE Electron. Express* **2017**, *14*, 20170551. [CrossRef]
74. Design of Hybrid Strong PUF Circuit against Machine Learning Attacks. Available online: https://journal.ecust.edu.cn/article/doi/10.14135/j.cnki.1006-3080.20221009003?pageType=en (accessed on 23 October 2023).

75. Cao, R.; Mei, N. A Fpga Hybrid Cro Puf Based on Three-State Gate for Improving Reliability and Hardware Overhead. *Arab. Econ. Bus. J.* **2014**, 1–6. [CrossRef]
76. Devika, K.N.; Bhakthavatchalu, R. FPGA Implementation of Programmable Hybrid PUF Using Butterfly and Arbiter PUF Concepts. *J. Phys. Conf. Ser.* **2022**, *2312*, 012033. [CrossRef]
77. Khoshroo, S. Design and Evaluation of FPGA-Based Hybrid Physically Unclonable Functions. In *Electronic Thesis and Dissertation Repository*; University of Western Ontario: London, ON, Canada, 2013.
78. Sahoo, D.P.; Mukhopadhyay, D.; Chakraborty, R.S. Design of Low Area-Overhead Ring Oscillator PUF with Large Challenge Space. In Proceedings of the 2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 9–11 December 2013; Available online: https://ieeexplore.ieee.org/document/6732277 (accessed on 27 September 2023).
79. Wu, Z.; Patel, H.; Sachdev, M.; Tripunitara, M.V. Strengthening PUFs Using Composition. In Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, 4–7 November 2019; pp. 1–8.
80. Sahoo, D.P.; Saha, S.; Mukhopadhyay, D.; Chakraborty, R.S.; Kapoor, H. Composite PUF: A New Design Paradigm for Physically Unclonable Functions on FPGA. In Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Arlington, VA, USA, 6–7 May 2014; pp. 50–55.
81. Machida, T.; Yamamoto, D.; Iwamoto, M.; Sakiyama, K. A New Arbiter PUF for Enhancing Unpredictability on FPGA. *Sci. World J.* **2015**, *2015*, e864812. [CrossRef] [PubMed]
82. Alamro, M.A.; Zhuang, Y.; Aseeri, A.O.; Alkatheiri, M.S. Examination of Double Arbiter PUFs on Security against Machine Learning Attacks. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3165–3171.
83. Alamro, M.A.; Mursi, K.T.; Zhuang, Y.; Alkatheiri, M.S.; Aseeri, A.O. Does Sophisticating Double Arbiter PUF Design Ensure Its Security? Performance and Security Assessments on 5-1 DAPUF. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 1788–1795.
84. Hou, S.; Ma, Y.; Deng, D.; Wang, Z.; Ren, G. Modeling and Physical Attack Resistant Authentication Protocol with Double PUFs. *J. Inf. Secur. Appl.* **2023**, *76*, 103543. [CrossRef]
85. Papakonstantinou, I.; Sklavos, N. Physical Unclonable Functions (PUFs) Design Technologies: Advantages and Trade Offs. In *Computer and Network Security Essentials*; Daimi, K., Ed.; Springer International Publishing: Cham, Switzerland, 2018; pp. 427–442. ISBN 978-3-319-58424-9.
86. Xin, X.; Kaps, J.P.; Gaj, K. A Configurable Ring-Oscillator-Based PUF for Xilinx FPGAs. In Proceedings of the 2011 14th Euromicro Conference on Digital System Design, Oulu, Finland, 31 August–2 September 2011; Available online: https://ieeexplore.ieee.org/abstract/document/6037472 (accessed on 24 October 2023).
87. Sklavos, N.; Chaves, R.; Di Natale, G.; Regazzoni, F. (Eds.) *Hardware Security and Trust*; Springer International Publishing: Cham, Switzerland, 2017; ISBN 978-3-319-44316-4.
88. Nam, J.-W.; Ahn, J.-H.; Hong, J.-P. Compact SRAM-Based PUF Chip Employing Body Voltage Control Technique. *IEEE Access* **2022**, *10*, 22311–22319. [CrossRef]
89. Miskelly, J.; Gu, C.; Ma, Q.; Cui, Y.; Liu, W.; O'Neill, M. Modelling Attack Analysis of Configurable Ring Oscillator (CRO) PUF Designs. In Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 19–21 November 2018; Available online: https://ieeexplore.ieee.org/abstract/document/8631638 (accessed on 24 October 2023).
90. Sahoo, D.P.; Nguyen, P.H.; Mukhopadhyay, D.; Chakraborty, R.S. A Case of Lightweight PUF Constructions: Cryptanalysis and Machine Learning Attacks. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 1334–1343. [CrossRef]
91. Maiti, A.; Gunreddy, V.; Schaumont, P. A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions. In *Embedded Systems Design with FPGAs*; Athanas, P., Pnevmatikatos, D., Sklavos, N., Eds.; Springer: New York, NY, USA, 2013; pp. 245–267. ISBN 978-1-4614-1362-2.
92. Naveenkumar, R.; Sivamangai, N.M.; Napolean, A.; Priya, S.S.S. Design and Evaluation of XOR Arbiter Physical Unclonable Function and Its Implementation on FPGA in Hardware Security Applications. *J. Electron. Test* **2022**, *38*, 653–666. [CrossRef]
93. Anandakumar, N.N.; Hashmi, M.S.; Sanadhya, S.K. Design and Analysis of FPGA-Based PUFs with Enhanced Performance for Hardware-Oriented Security. *J. Emerg. Technol. Comput. Syst.* **2022**, *18*, 72. [CrossRef]
94. Mahalat, M.H.; Mandal, S.; Mondal, A.; Sen, B.; Chakraborty, R.S. Implementation, Characterization and Application of Path Changing Switch Based Arbiter PUF on FPGA as a Lightweight Security Primitive for IoT. *ACM Trans. Des. Autom. Electron. Syst.* **2021**, *27*, 26. [CrossRef]
95. Rabiei, H.; Kaveh, M.; Mosavi, M.; Martín, D. MCRO-PUF: A Novel Modified Crossover RO-PUF with an Ultra-Expanded CRP Space. *CMC* **2022**, *74*, 4831–4845. [CrossRef]
96. Dang, T.-K.; Serrano, R.; Hoang, T.-T.; Pham, C.-K. A Novel Ring Oscillator PUF for FPGA Based on Feedforward Ring Oscillators. In Proceedings of the 2022 19th International SoC Design Conference (ISOCC), Gangneung-si, Republic of Korea, 19–22 October 2022; pp. 87–88.
97. Merli, D.; Stumpf, F.; Eckert, C. Improving the Quality of Ring Oscillator PUFs on FPGAs. In Proceedings of the 5th Workshop on Embedded Systems Security, Scottsdale, Arizona, 24 October 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 1–9.

98. Lee, S.; Oh, M.-K.; Kang, Y.; Choi, D. Implementing a Phase Detection Ring Oscillator PUF on FPGA. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 17–19 October 2018; pp. 845–847.
99. Choudhury, M.; Pundir, N.; Niamat, M.; Mustapa, M. Analysis of a Novel Stage Configurable ROPUF Design. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 942–945.
100. Zhang, J.; Shen, C.; Guo, Z.; Wu, Q.; Chang, W. CT PUF: Configurable Tristate PUF Against Machine Learning Attacks for IoT Security. *IEEE Internet Things J.* **2022**, *9*, 14452–14462. [CrossRef]
101. Sahoo, D.P.; Chakraborty, R.S.; Mukhopadhyay, D. Towards Ideal Arbiter PUF Design on Xilinx FPGA: A Practitioner's Perspective. In Proceedings of the 2015 Euromicro Conference on Digital System Design, Madeira, Portugal, 26–28 August 2015; Available online: https://ieeexplore.ieee.org/abstract/document/7302326/ (accessed on 25 September 2023).
102. Hori, Y.; Yoshida, T.; Katashita, T.; Satoh, A. Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs. In Proceedings of the 2010 International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 13–15 December 2010; pp. 298–303.
103. Habib, B.; Gaj, K.; Kaps, J.-P. FPGA PUF Based on Programmable LUT Delays. In Proceedings of the 2013 Euromicro Conference on Digital System Design, Los Alamitos, CA, USA, 4–6 September 2013; pp. 697–704.
104. Yu, H.; Leong, P.H.; Xu, Q. An FPGA Chip Identification Generator Using Configurable Ring Oscillators. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2011**, *20*, 2198–2207. [CrossRef]
105. Zhang, J.; Tan, X.; Zhang, Y.; Wang, W.; Qin, Z. Frequency Offset-Based Ring Oscillator Physical Unclonable Function. *IEEE Trans. Multi-Scale Comput. Syst.* **2018**, *4*, 711–721. [CrossRef]
106. Karmakar, M.; Naz, S.F.; Shah, A.P. Fault-Tolerant Reversible Logic Gate-Based RO-PUF Design. *Mem. Mater. Devices Circuits Syst.* **2023**, *4*, 100055. [CrossRef]
107. Ardakani, A.; Shokouhi, S.B.; Reyhani-Masoleh, A. Improving Performance of FPGA-Based SR-Latch PUF Using Transient Effect Ring Oscillator and Programmable Delay Lines. *Integration* **2018**, *62*, 371–381. [CrossRef]
108. Habib, B.; Kaps, J.-P.; Gaj, K. Efficient SR-Latch PUF. In Proceedings of the Applied Reconfigurable Computing, Bochum, Germany, 13–17 April 2015; Sano, K., Soudris, D., Hübner, M., Diniz, P.C., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 205–216.
109. Stanciu, A.; Cirstea, M.N.; Moldoveanu, F.D. Analysis and Evaluation of PUF-Based SoC Designs for Security Applications. *IEEE Trans. Ind. Electron.* **2016**, *63*, 5699–5708. [CrossRef]
110. Khan, S.; Shah, A.P.; Chouhan, S.S.; Gupta, N.; Pandey, J.G.; Vishvakarma, S.K. A Symmetric D Flip-Flop Based PUF with Improved Uniqueness. *Microelectron. Reliab.* **2020**, *106*, 113595. [CrossRef]
111. FLEX PUF: A Flexible Physical Unclonable Function Design Using Configurable Templates. Available online: https://www.researchsquare.com (accessed on 26 September 2023).
112. Tanamoto, T.; Yasuda, S.; Takaya, S.; Fujita, S. Physically Unclonable Function Using an Initial Waveform of Ring Oscillators. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 827–831. [CrossRef]
113. Ma, Q.; Gu, C.; Hanley, N.; Wang, C.; Liu, W.; O'Neill, M. A Machine Learning Attack Resistant Multi-PUF Design on FPGA. In Proceedings of the 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), Jeju, Republic of Korea, 22–25 January 2018; pp. 97–104.
114. Gu, C.; Liu, W.; Cui, Y.; Hanley, N.; O'Neill, M.; Lombardi, F. A Flip-Flop Based Arbiter Physical Unclonable Function (APUF) Design with High Entropy and Uniqueness for FPGA Implementation. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1853–1866. [CrossRef]
115. Nassar, H.; Bauer, L.; Henkel, J. CaPUF: Cascaded PUF Structure for Machine Learning Resiliency. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 4349–4360. [CrossRef]
116. Reddy Jeeru, D.; Panduranga Vittal, K.; Anikethan, H.V.U.; Kumar, A.S. Implementation of Enhanced Parallel Port Interface for Frequency Analysis in a Configurable Ring Oscillator PUF Circuits on Xilinx Spartan 3E Architecture. In Proceedings of the 2019 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 26–27 July 2019; pp. 1–7.
117. Wu, L.; Hu, Y.; Zhang, K.; Li, W.; Xu, X.; Chang, W. FLAM-PUF: A Response–Feedback-Based Lightweight Anti-Machine-Learning-Attack PUF. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 4433–4444. [CrossRef]
118. Yang, S.-H.; Liu, T.-T. A Highly Stable Physically Unclonable Function Using Algorithm-Based Mismatch Hardening Technique in 28-Nm CMOS. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2023**, *70*, 280–289. [CrossRef]
119. Gao, Y.; Su, Y.; Yang, W.; Chen, S.; Nepal, S.; Ranasinghe, D.C. Building Secure SRAM PUF Key Generators on Resource Constrained Devices. In Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 11–15 March 2019; pp. 912–917.
120. Delvaux, J.; Gu, D.; Schellekens, D.; Verbauwhede, I. Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2015**, *34*, 889–902. [CrossRef]
121. Ali Pour, A.; Afghah, F.; Hély, D.; Beroulle, V.; Natale, G.D.; Korenda, A.R.; Cambou, B. Helper Data Masking for Physically Unclonable Function-Based Key Generation Algorithms. *IEEE Access* **2022**, *10*, 40150–40164. [CrossRef]

122. Zhang, J.; Ding, L.; Chen, Z.; Li, W.; Qu, G. DA PUF: Dual-State Analog PUF. In Proceedings of the 59th ACM/IEEE Design Automation Conference, San Francisco, CA, USA, 10–14 July 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 73–78.
123. Liu, Y.; Li, J.; Qu, T.; Dai, Z. CBDC-PUF: A Novel Physical Unclonable Function Design Framework Utilizing Configurable Butterfly Delay Chain Against Modeling Attack. *ACM Trans. Des. Autom. Electron. Syst.* **2023**, *28*, 78. [CrossRef]
124. Sahoo, D.P.; Mukhopadhyay, D.; Chakraborty, R.S.; Nguyen, P.H. A Multiplexer-Based Arbiter PUF Composition with Enhanced Reliability and Security. *IEEE Trans. Comput.* **2018**, *67*, 403–417. [CrossRef]
125. Shi, J.; Lu, Y.; Zhang, J. Approximation Attacks on Strong PUFs. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 2138–2151. [CrossRef]
126. Zhang, J.; Shen, C. Set-Based Obfuscation for Strong PUFs against Machine Learning Attacks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 288–300. [CrossRef]
127. Zhang, J.; Lin, Y.; Lyu, Y.; Qu, G. A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-Per-Device Licensing. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1137–1150. [CrossRef]
128. Zheng, J.X.; Potkonjak, M. A Digital PUF-Based IP Protection Architecture for Network Embedded Systems. In Proceedings of the Tenth ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Los Angeles, CA, USA, 20–21 October 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 255–256.
129. Guo, Q.; Ye, J.; Gong, Y.; Hu, Y.; Li, X. PUF Based Pay-Per-Device Scheme for IP Protection of CNN Model. In Proceedings of the 2018 IEEE 27th Asian Test Symposium (ATS), Hefei, China, 15–18 October 2018; pp. 115–120.
130. Zhang, J.; Qu, G. Physical Unclonable Function-Based Key Sharing via Machine Learning for IoT Security. *IEEE Trans. Ind. Electron.* **2020**, *67*, 7025–7033. [CrossRef]
131. Tehranipoor, F.; Karimian, N.; Yan, W.; Chandy, J.A. DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 1085–1097. [CrossRef]
132. Suresh, V.; Manimegalai, R. SPIC-SRAM PUF Intergrated Chip Based Software Licensing Model. In Proceedings of the Security in Computing and Communications, Bangalore, India, 19–22 September 2018; Thampi, S.M., Madria, S., Wang, G., Rawat, D.B., Alcaraz Calero, J.M., Eds.; Springer: Singapore, 2019; pp. 377–388.
133. Chhabra, S.; Lata, K. Hardware Obfuscation of AES IP Core Using PUFs and PRNG: A Secure Cryptographic Key Generation Solution for Internet-of-Things Applications. *SN Comput. Sci.* **2022**, *3*, 303. [CrossRef]
134. Enamul Quadir, M.S.; Chandy, J.A. Key Generation for Hardware Obfuscation Using Strong PUFs. *Cryptography* **2019**, *3*, 17. [CrossRef]
135. Zhang, J. A Practical Logic Obfuscation Technique for Hardware Security. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2016**, *24*, 1193–1197. [CrossRef]
136. Gora, M.A.; Maiti, A.; Schaumont, P. A Flexible Design Flow for Software IP Binding in FPGA. *IEEE Trans. Ind. Inform.* **2010**, *6*, 719–728. [CrossRef]
137. Barbareschi, M.; Di Natale, G.; Torres, L.; Mazzeo, A. A Ring Oscillator-Based Identification Mechanism Immune to Aging and External Working Conditions. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2017**, *65*, 700–711. [CrossRef]
138. Huang, Z.; Wang, Q. A PUF-Based Unified Identity Verification Framework for Secure IoT Hardware via Device Authentication. *World Wide Web* **2020**, *23*, 1057–1088. [CrossRef]
139. Gope, P.; Millwood, O.; Sikdar, B. A Scalable Protocol Level Approach to Prevent Machine Learning Attacks on Physically Unclonable Function Based Authentication Mechanisms for Internet of Medical Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1971–1980. [CrossRef]
140. Lounis, K.; Zulkernine, M. Lessons Learned: Analysis of PUF-Based Authentication Protocols for IoT. *Digit. Threat.* **2022**, *4*, 19. [CrossRef]
141. Aysu, A.; Gulcan, E.; Moriyama, D.; Schaumont, P.; Yung, M. End-To-End Design of a PUF-Based Privacy Preserving Authentication Protocol. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2015, Saint-Malo, France, 13–16 September 2015; Güneysu, T., Handschuh, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 556–576.
142. Li, D.; Lu, Z.; Zou, X.; Liu, Z. PUFKEY: A High-Security and High-Throughput Hardware True Random Number Generator for Sensor Networks. *Sensors* **2015**, *15*, 26251–26266. [CrossRef]
143. Van der Leest, V.; van der Sluis, E.; Schrijen, G.-J.; Tuyls, P.; Handschuh, H. Efficient Implementation of True Random Number Generator Based on SRAM PUFs. In *Cryptography and Security: From Theory to Applications: Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*; Naccache, D., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; pp. 300–318. ISBN 978-3-642-28368-0.
144. Chen, S.; Li, B.; Zhou, C. FPGA Implementation of SRAM PUFs Based Cryptographically Secure Pseudo-Random Number Generator. *Microprocess. Microsyst.* **2018**, *59*, 57–68. [CrossRef]