electronics

MDPI

Article

Robustness and Unpredictability for Double Arbiter PUFs on Silicon Data: Performance Evaluation and Modeling Accuracy

Meznah A. Alamro [1,*], Khalid T. Mursi [1,2], Yu Zhuang [1], Ahmad O. Aseeri [3] and Mohammed Saeed Alkatheiri [2]

[1] Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA; khalid.mursi@ttu.edu or kmursi@uj.edu.sa (K.T.M.); yu.zhuang@ttu.edu (Y.Z.)
[2] College of Computer Science and Engineering, University of Jeddah, Jeddah 23218, Saudi Arabia; msalkatheri@uj.edu.sa
[3] Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia; a.aseeri@psau.edu.sa
* Correspondence: meznah.alamro@gmail.com or meznah.alamro@ttu.edu

Received: 15 April 2020; Accepted: 21 May 2020; Published: 24 May 2020

Abstract: Classical cryptographic methods that inherently employ secret keys embedded in non-volatile memory have been known to be impractical for limited-resource Internet of Things (IoT) devices. Physical Unclonable Functions (PUFs) have emerged as an applicable solution to provide a keyless means for secure authentication. PUFs utilize inevitable variations of integrated circuits (ICs) components, manifest during the fabrication process, to extract unique responses. Double Arbiter PUFs (DAPUFs) have been recently proposed to overcome security issues in XOR PUF and enhance the tolerance of delay-based PUFs against modeling attacks. This paper provides comprehensive risk analysis and performance evaluation of all proposed DAPUF designs and compares them with their counterparts from XOR PUF. We generated different sets of real challenge–response pairs CRPs from three FPGA hardware boards to evaluate the performance of both DAPUF and XOR PUF designs using special-purpose evaluation metrics. We show that none of the proposed designs of DAPUF is strictly preferred over XOR PUF designs. In addition, our security analysis using neural network reveals the vulnerability of all DAPUF designs against machine learning attacks.

Keywords: physical unclonable functions; double arbiter PUF; machine learning; authentication; FPGA; Internet of Things

1. Introduction

Traditional authentication procedures and protocols were constructed upon the state-of-the-art cryptographic algorithms that fundamentally operate over the use of secret keys stored in non-volatile memory. Such memory-based cryptographic methods are subjected to invasive and side-channel attacks [1–3], increasing the cost of maintaining secure transactions and processes. In addition, the intensive processing nature of such classical cryptographic algorithms makes them unsuitable for the IoT environment. Therefore, Physical Unclonable Functions (PUFs) have emerged as a lightweight memoryless solution to provide security requirements for limited-resource Internet of Things (IoT) devices. PUF utilizes physical variations in integrated circuits to output unique responses that are unreproducible even by manufacturers. Even if the exact design is implemented and the same challenge set is used, the responses generated in one circuit should be uniquely distinguished from another one, making them appropriate for providing security requirements at a low cost, due to the elimination of non-volatile memory.

The notion of PUFs was first proposed in [4,5] which later led to the introduction of delay-based Arbiter PUF (APUF) by [6]. Among all proposed designs of delay-based arbiter PUF, XOR PUF designs have been thought to be one of the most resistant designs against possible attacks due to the use of XOR operation. However, XOR PUFs have been reported to be mathematically cloneable by machine learning-based methods and suffer from a low unpredictability rate [7–10]. Recently, Double Arbiter PUFs (DAPUFs) have been proposed by [11] as an enhancement against machine learning attacks. Authors in [11] argue that both 3-1 DAPUF and 4-1 DAPUF are secure against modeling attacks. Although authors in [12] could successfully break the security of 3-1 DAPUF using a very large set of CRPs, 17 million CRPs, they failed to make a breakthrough 4-1 DAPUF using the same set of CRPs.

In this paper, we examine the robustness and unpredictability of DAPUF using silicon data to evaluate the reliability of DAPUF designs in reproducing the exact key given the same set of input challenges, to what extent challenge sets of a given design is biased, and the uniqueness of a key produced from one circuit to another when the same design and challenge set is used. Since the introduction of DAPUF designs, to the best of our knowledge, the proposed DAPUF designs have not been thoroughly investigated since its introduction, and the performance evaluations were carried out using a very limited number of CRPs. This has motivated us to perform a detailed analysis of DAPUF that contributes to the literature in the following:

- Perform comprehensive performance evaluations on all proposed designs of DAPUF and their counterparts from XOR PUF using different sets of CRPs generated from FPGA chips.
- Examine the security of each proposed design using FPGAs data drawn from the implemented DAPUF designs on FPGA boards. Thus:
  - By reconstructing the 3-1 DAPUF design in our FPGA boards, we were able to break the security of 3-1 DAPUF design and substantially lower the number of CRPs required from 17 million CRPs, as reported by [12], to 500 k only with a higher accuracy rate.
  - We were able to successfully model 4-1 DAPUF that has never been cloned at the time of writing this paper.

The rest of the paper is organized as follows: Section 1.1 gives an overview of delay-based arbiter PUF. Section 1.2 discusses related work on DAPUF, and Section 1.3 points out areas that need to be revisited and our contributions on DAPUF. Section 2 describes performance evaluations developed in [13,14] that we rely on to perform our analysis on DAPUF designs. In Section 3, we explain our experimental setup which includes FPGA implementation, CRP generation, and our machine learning attack model. Section 4 presents our results for performance evaluations, modeling attacks, and detailed discussion on what is the preferred design. Finally, concluding remarks are provided in Section 5.

1.1. Background on Delay Based Arbiter PUFs

Since the design of both XOR PUF and DAPUF consist of multiple arbiter PUFs, it is essential to understand the structure of an arbiter PUF design. The initial design of arbiter PUFs that was introduced in [15,16] is illustrated in Figure 1. It consists of two parallel paths on which electric signals (input challenge) race against each other simultaneously through n-stages to reach the Arbiter. Each stage contains a pair of multiplexers and the challenge bit decides which path will be taken by the signal. The final stage is connected with an arbiter using two latches from top and bottom paths. The signal that reaches the arbiter input bit first will take the value of 1, otherwise 0. The response in n-stage arbiter PUF could take 2^n different paths. However, it turns out that the output of arbiter PUF could be predicted using machine learning attack depending on a model of additive delay model [6] that could be represented as a linear specification problem, which makes it vulnerable against machine learning attacks.

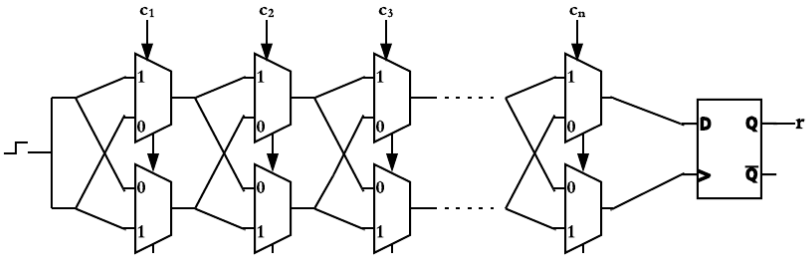

Figure 1. Structure of Arbiter PUF.

XOR PUF, illustrated in Figure 2, was proposed in [17] to enhance the resilience of arbiter PUF. XOR PUF is composed of N number of arbiter PUFs and the response from each arbiter is XORed to generate the output response of XOR PUF. While XOR PUF has evidently added more complexity by introducing nonlinearity to the design [8], the response of XOR PUF can be predicted using machine learning techniques [7–9].

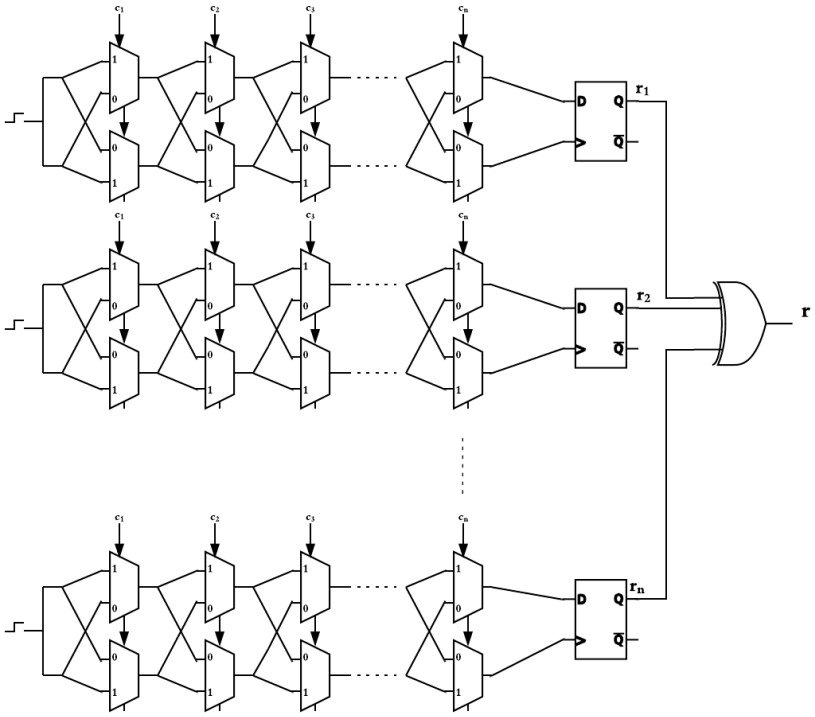

Figure 2. Structure of XOR Arbiter PUF.

DAPUF has recently been proposed in [18] to add further complexity to the modeled design of Arbiter PUF as illustrated in Figure 3. Three DAPUF designs have been introduced in [11,18] which are (2-1, 3-1, and 4-1 DAPUF) shown in Figures 4–6, respectively. In DAPUF, it requires N(N-1) internal bit XORing, one from each arbiter, to get the response of N-1 DAPUF.

DAPUF is similar to XOR PUF in that $N \geq 2$ arbiters are required for any of their designs. However, unlike XOR PUF where each arbiter receives the top and bottom delay signals from the same APUF, each arbiter in DAPUF will take either the top delay lines or the bottom delay lines in similar proportions. DAPUF requires N(N-1) XORing to get the one bit-response of N-1 DAPUF, while it takes N-XORed responses to generate the one-bit output response of N-XOR PUF.

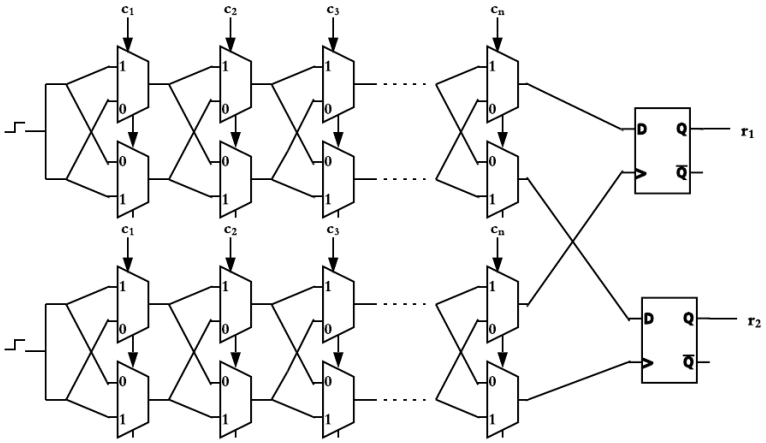

Figure 3. Structure of Double Arbiter PUF.

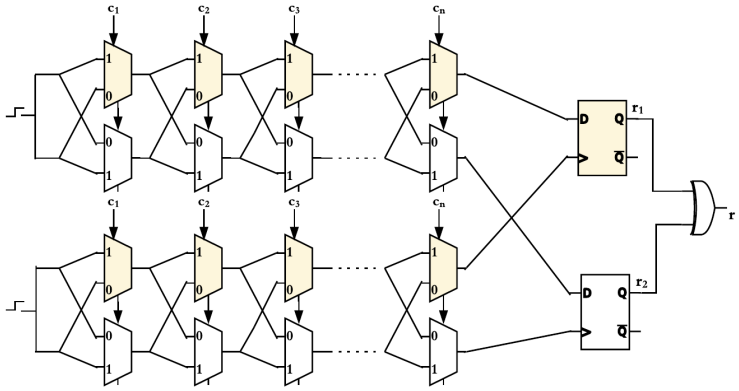

Figure 4. Structure of 2-1 Double Arbiter PUF.

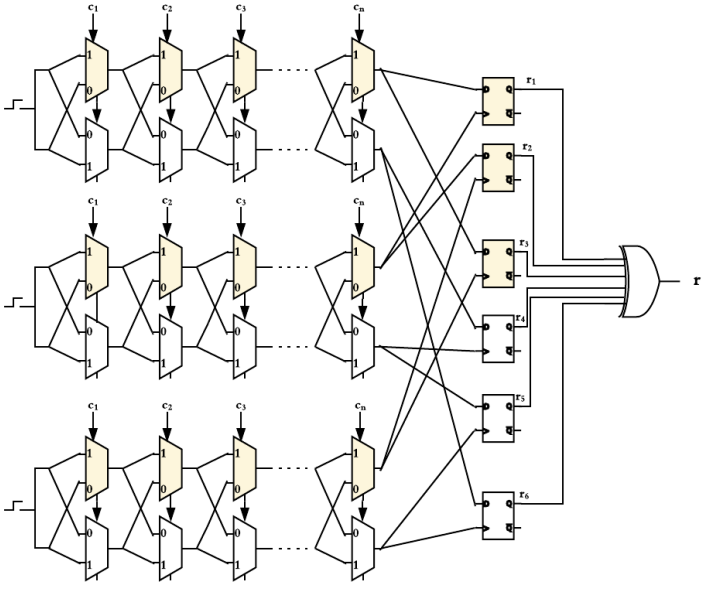

Figure 5. Structure of 3-1 Double Arbiter PUF.

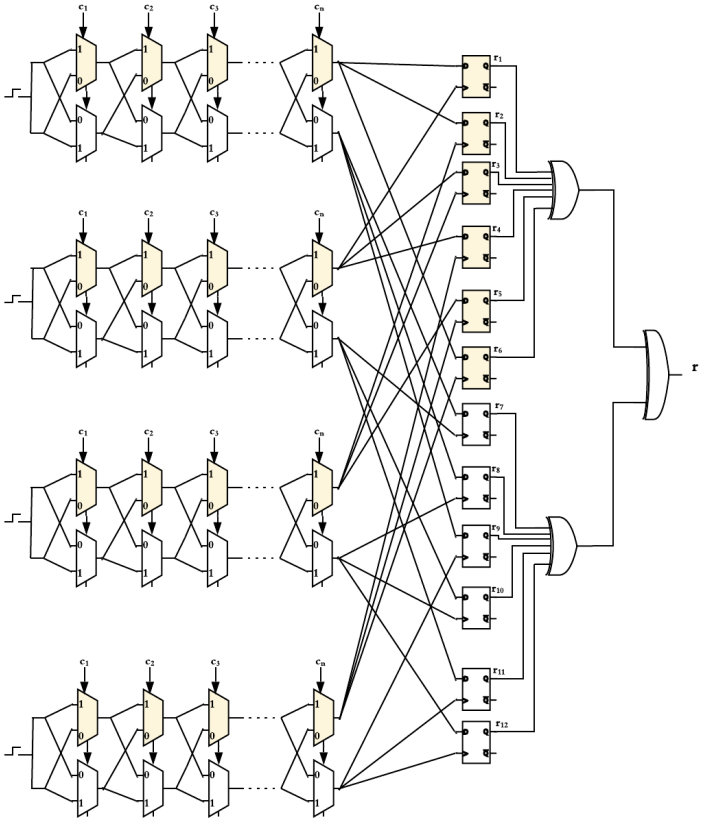

Figure 6. Structure of 4-1 Double Arbiter PUF.

1.2. Related Work

Previous work on XOR PUF and DAPUF can be split into performance evaluation and security analysis. Performance evaluation, on one hand, is based on evaluation metrics introduced in [13,14] that allow us to examine and compare the suitability of each proposed design of delay-based arbiter PUF. Performance analysis is based on hardware primitives in which each proposed design is implemented using FPGA boards. Each implemented design on FPGA utilizes the inherited variation in the ICs to generate the challenge–response pairs.

Performance evaluation on XOR PUF is well reported in the literature [10,19–23] using different components (also known as streams) and number of stages. However, there is very limited number of research conducted on the implementation and experimentation of DAPUFs. Authors of [24] have examined and compared three different designs which are XOR PUF, PAPUF, and DAPUF. However, their implemented design of DAPUF, shown in Figure 3, is not similar to those implemented in [11,18]. In particular, the response from each arbiter has to be XORed to generate the one-bit response of any DAPUF design, a part that has prevented them from conducting their analysis on different DAPUF designs. The only paper that reported performance analysis on all DAPUF designs is from the same authors that proposed the design of DAPUF [11], in which they analyzed the performance of DAPUF and compare it with that of XOR PUF using three evaluation metrics.

Security analysis, on the other hand, is pivotal to the applicability of any PUF design. In fact, XOR PUF and DAPUF, with multiple components and different number stages, were primarily introduced to enhance the resilience of arbiter PUF design against machine learning attacks. The tolerance of XOR PUFs against machine learning attack have been extensively studied in [7,9,10,22] and they were able to impersonate them using different machine learning techniques. DAPUF designs have shown more resilience against machine learning attacks compared to XOR PUF. Authors of [11] reported that 3-1 DAPUF and 4-1 DAPUF are secure against machine learning attacks, but their findings were based on small sets of CRPs. Other work on DAPUF utilized 17 million real CRPs to

successfully attack 3-1 DAPUF, while they report that 4-1 DAPUF remains secure even when deep learning techniques are used and 17 million real CRPs are employed [12]. Authors of [25] have simulated 2-1, 3-1, and 4-1 DAPUF designs and were able to virtually counterfeit all designs using different sets of CRPs.

1.3. Motivation and Contribution

Previous work on DAPUF lack in two essential aspects. First, more comprehensive performance evaluations are required to examine each of the implemented design. Studies on XOR PUF include correctness and diffuseness; these performance metrics are not reported in DAPUF work. Performance analysis on DAPUF conducted in [11] uses a very limited number of CRPs.

Our goal is to evaluate the performance of each proposed design of DAPUF using five evaluation metrics, and the reported results of each evaluated design are based on 60,000 CRPs with 32 iterations and each experiment is repeated 32 times. Our experiments include comparisons between DAPUF designs and their counterparts from XOR PUF using similar experimental setups.

The second area that needs to be revisited is the security analysis on DAPUF. Authors of [11] examined the resilience of DAPUF using small sets of CRPs with 1000 training samples and 10,000 test samples, and they concluded that 3-1 DAPUF and 4-1 DAPUF are secure against machine learning attacks. However, it is important to examine the tolerance of DAPUF when a sufficient amount of CRPs can be acquired. The security of DAPUF designs have also been studied in [12] using deep learning. However, employing deep learning techniques may raise the issue of overfitting on the trained model, and the model will perform worse on the data outside the trained model [26]. Their approach to solving this problem was to dropout layers at the end of the network and increase the set of CRPs to 17 million. While this has helped them to model 3-1 DAPUF with an 86% accuracy rate, their approach had no efficiency gain and failed to model 4-1 DAPUF. Our approach is to use a rather simpler model by employing a neural network-based attack model that enables us to successfully model all the proposed design of DAPUF, namely 2-1, 3-1, and 4-1 DAPUF and their counterparts 2, 3, and 4 XOR PUF. Our modeled attack reduced the number of CRPs required to successfully attack any DAPUF design with a higher accuracy rate compared to the literature.

2. Performance Evaluation of PUF

Several types of PUF designs have been proposed since the initiation of PUFs including delay-based Arbiter PUFs, Ring Oscillator PUFs [17], Lightweight Secure PUFs [27], etc. These designs can be complicated by increasing the number of stages and/or by increasing the number of components embedded in the design. While sophisticating the design can surely enhance the resilience of a PUF instance against attacks, it raises the question of what are the trade-offs brought by these complications. For this reason, a number of evaluation metrics have been proposed to evaluate the performance of PUF designs. These metrics can be classified into intra-chip and inter-chip evaluation metrics. The former is related to examinations of a PUF design performance within a device which are randomness, correctness, steadiness, and diffuseness, while the latter examines how unique the output of a PUF design is when the same challenge is given to different devices, called a uniqueness test. These performance metrics are explained in detail next, and all notations used in these metrics are shown in Table 1.

Table 1. List of Notations used in Performance Metrics.

Notation	Description
S	Set of output keys generated
I	Length of stages implemented
T	Number of testes generated in the exterminates
K	Number of devices used
s	Set of CRPs generated to produce one key
i	The i-bit response to a challenge bit
t	The t-bit test performed to generate a key
k	The index of a device used in the implementation

2.1. Performance Metrics

2.1.1. Randomness

Randomness measures how balanced the output of a PUF in producing numbers of 0′s and numbers of 1′s. The proper output of a PUF should be uniformly distributed between 0′s and 1′s. Randomness, defined in [13], is calculated using the hamming weight from running multiple tests using different sets of CRPs:

$$P_k = \frac{1}{S \cdot I \cdot T} \sum_{s=1}^{S} \sum_{i=1}^{I} \sum_{t=1}^{T} b_{s,i,t}, \tag{1}$$

where P_k is the output frequency of 1′s generated from device k, and $b_{s,i,t}$ is the i-bit response of a challenge set s when examined at test t on device k. The randomness of a device is calculated as follows:

$$U_k = -\log_2 \max(p_k, 1 - p_k). \tag{2}$$

The ideal value is equal to 1 when $P_k = 0.5$ (i.e., when the data are balanced). The higher the deviation to either $P_k = 0$ or $P_k = 1$, the more unbalanced the data are.

2.1.2. Steadiness

Evaluation of a PUF design should consider factors that may influence the ability to reproduce the exact key under different work environments. One factor is related to power voltage that could impact signal variations within the device. Noise due to environmental variability such as temperature in the working area may also affect the delay in ICs. To measure the influence of these variables, multiple tests have to be made using the same devices and keeping the number of CRPs fixed. Hori et al. [13] introduced one way to measure steadiness of a response bit as follows:

$$S_k = 1 + \frac{1}{S \cdot I} \sum_{s=1}^{S} \sum_{i=1}^{I} \log_2 \max(P_{s,i}, 1 - P_{s,i}), \tag{3}$$

where S_k takes the value of one when the output has repeatedly produced the exact key (i.e., when $P_{s,i} = 0$ or $P_{s,i} = 1$). The value of S_k is zero if $P_{s,i} = 0.5$ when it fails to emulate similar responses for the same challenge sets.

2.1.3. Correctness

The reliability of a PUF is also exposed to internal factors that could impact the accuracy of the responses from a given challenge set. One source of incorrect responses is due to damage that does not destroy the IC entirely, but rather influence the permanent variations in the device.

Aging of the examined device could potentially affect identification degradation after lengthy use [16]. The correctness of the device is measured using the fractional Hamming distance as follows [13]:

$$C_K = 1 - \frac{2}{S \cdot I \cdot T} \sum_{s=1}^{S} \sum_{i=1}^{I} \sum_{t=1}^{T} \log_2 \max(b_{s,i} \oplus b_{s,i,t}), \tag{4}$$

where $b_{s,i}$ is the correct i-bit response that is determined by the majority voting from the set of all challenges, while $b_{s,i,t}$ is the response to the same bit challenge through T tests. Thus, correctness is calculated as the sum of the normalized Hamming distance between the "correct" bit and the response of that specific challenge when generated multiple times.

2.1.4. Diffuseness

Each challenge set in a PUF is expected to output unique responses that could be utilized as a secret key. One approach to examining the validity of this key is to evaluate how it is different from any other key produced by the same device. Diffuseness, introduced in [13], is calculated using the fractional Hamming distance of all possible keys generated by the device as follows:

$$D_k = \frac{4}{S^2 \times I} \sum_{s=1}^{S-1} \sum_{m=s+1}^{S} (b_{s,i} \oplus b_{m,i}), \tag{5}$$

where $b_{s,i}$ and $b_{m,i}$ are the bit response of challenge set s and m, respectively. Since it is inevitable for any PUF design to share common bit-responses in such multi-bit settings, Hori et al. [13] prove that the upper bound of the sum of Hamming distance for all possible key is

$$\frac{4}{S^2 \times I}.$$

The highest value is 1 when D_k reaches the upper bound, while it takes the value of 0 if all of the generated keys from different challenge sets are identical.

2.1.5. Uniqueness

All of the above metrics are concerned about the nature of challenge–response pairs within the same device. The metric that measures to what extent CRP sets' behavior vary among different devices is called Uniqueness. It evaluates how different the response of one PUF device from the response generated from another device when the same design implemented and the same set of challenges is given to both devices. Since the response on any PUF design relies on the inherited delay variation, it is crucial to examine if responses from a specific challenge set can be replicated using a different device. Maiti et al. [14] show one way to measure uniqueness using the hamming distance between responses generated from different devices from a given challenge as follows:

$$Q = \frac{2}{S \cdot I \cdot K(K-1)} \sum_{s=1}^{S} \sum_{i=1}^{I} \sum_{n=1}^{K-1} \sum_{m=n+1}^{K} (b_{n,s,i} \oplus b_{m,s,i}), \tag{6}$$

where n and m are two different PUF devices, and $b_{n,s,i}$ and $b_{m,s,i}$ are the i-bit responses generated from a given challenge. The uniqueness of all challenges is averaged out to obtain the uniqueness of the response uniqueness of all experimented tests set. Hori et al. [13] proposed a similar approach, but they differ in the normalization value where the ideal value is one instead of one half as follows:

$$Q_k = \frac{4}{S \cdot I \cdot K} \sum_{s=1}^{S} \sum_{i=1}^{I} \sum_{n=1, n \neq m}^{K} (b_{n,s,i} \oplus b_{m,s,i}), \tag{7}$$

and the upper bound is similar to the one shown in randomness.

3. Experimental Setup

3.1. FPGA Hardware and Utilized Software in the Experiments

Our hardware implementation procedures follow [10] to build in XOR PUF and DAPUF instances. Three Artix-7 FPGA devices have been used to implement both designs which we shall refer to A, B, and C. We used VHDL on Xilinx Vivado, edition 15.4 HL, that is capable of describing signal variations in ICs and synthesize HDL designs. Xilinx SDK is utilized to capture the input/output functioning using C language.

Core voltage in Artix-7 is set to 1V with 30 mv tolerance and the junction temperature is between 0–85 °C. The reported ambient temperature is around 26 °C during all performed experiments. When generating the CRPs, fluctuation in power voltage and variations in the temperature are reflected in the stability metric [14]. In our experiments, correctness and steadiness metrics evaluate how consistent the responses generated from any design are under these conditions. In case of placing 2-1 DAPUFs on the circuit, the first stream gates' placement starts from X1Y100 to X32Y100 and X1Y101 to X32Y101 and connected to an arbiter D-FF, and the second stream starts from X1Y102 to X32Y102 and X1Y103 to X32Y103 connected to arbiter D-FF. The top and bottom MUXes of each stage are placed on the same slice. The same procedure is followed to place all designs. The number of gates allocated in each implemented designs on FPGA is calculated as follows:

$$2 \times \text{number of stages} \times \text{ number of streams} + \text{number of arbiter} + \text{XOR gate}$$

The number of components and arbiters differ in each of the implemented designs, while the number of XOR gate is equal to 1 in all designs. This shows that the cost of implementing DAPUFs is higher than their counterparts from XOR PUFs.

3.2. CRP Generation

We implement all proposed designs of DAPUF (2-1, 3-1, and 4-1 DAPUF) and their counterparts from XOR PUF (2, 3, and 4 XOR PUF). Each design is implemented on three FPGA boards which results in a total of 18 implementations, and the challenge length is 64-bit in all tested designs. Each extracted CRP could take one path ranging from 0–2^n where n is the number of challenge bits in the implementation.

Several papers have evaluated the performance of PUF designs using sets of CRPs ranging from 128 as in [13,14,18,19] to 32,768 CRPs in [21]. Although increasing the number of accumulated CRPs does not imply a better approximation to the true distribution, Ref. [23] shows that the model accuracy increases when more CRPs are used. Considering the trade-offs between enhancing model accuracy and the cost brought by increasing the number of CRPs, we limited the number of CRPs generated in our performance evaluations to 60,000 CRPs with 32 iterations.

Each of the challenge sets is utilized to examine randomness and diffuseness of each design. To evaluate the steadiness of each design, each challenge set is repeated instantly 32 times. The correctness of all designs is evaluated when the whole experiment on each device and for each design is iterated 32 times. We evaluate uniqueness by examining how different the responses of 60,000 CRPs from each design are when generated on each device. This procedure will capture factors that may have some impacts on the reported results such as heat on the device or the frequency of power voltage. The reported results for each performance evaluation metric of each design is the average from all of the three FPGA boards.

Generating the required CRPs for performance experiments in all XOR PUFs and DAPUF designs took an approximately similar time. Each design on each device takes around 128 min, and the total time to generate all required CRPs is 38.4 h. These CRPs are used to evaluate the performance of each design on different devices using five performance metrics, and evaluation of each design takes approximately 9 h, which brings the total time required to examine all devices to 54 h.

3.3. Machine Learning Attacks

Our work also covers the security of all DAPUF instances and relevant XOR PUF designs. To compare our security analysis with the literature, the challenge length is set to 64-bits in all designs. We use neural networks that utilize Scikit-learn [28] in our modeling attack with Multi-Layer Perceptron (MLP) that is capable of handling nonlinearity in both designs. Our machine learning attack model employs Rectified Linear Unit (ReLU) to learn the parameters of each PUF design. The Adaptive Moment Estimation (ADAM) optimizer [29] enabled us to achieve high accuracies and successfully model all examined designs.

It is preferable to find a globally optimal solution using the smallest number of parameters, and that model with higher degree of freedoms is expected to perform worse than smaller networks that are trained to zero error [30]. XOR PUFs and DAPUF designs could be modeled using NN based attack without the need to employ more sophisticated deep learning techniques. In our modeling attack, we found that three hidden layers derive the best accuracy for all implemented designs. The number of neurons employed varies based on the complexity of the examined design, the number of components implemented, and the number of CRPs used. In implementations with large CRPs, the number of neurons in each layer is 2^n for XOR PUF designs [7] and 2^{2n-1} for DAPUF designs where n is the number of streams in each design. When small sets of CRPs are used, the number of neurons should be decreased to model DAPUF designs with higher accuracy rates. A summary of MLP parameters is listed in Table 2.

Table 2. Model parameters.

Parameter	Description
Optimizer	ADAM
Number of Hidden Layers	Three hidden layers
Number of Neurons	
	XOR PUFs = 2^n,
	DAPUFs = 2^{2n-1}, where n is the # of stream
Hidden layer Activation function	ReLU
Output layer Activation function	Sigmoid
Learning Rate	1×10^{-3}

We generated a variety of CRPs sets to examine the security of DAPUFs and XOR PUFs. Our approach is to examine the resilience of each evaluated design when different sets of CRPs are attainable. In all experiments, each CRPs set is split into 85% training set and 15% test set. The breakability of each design is tested using 100 k, 500 k, and one million generated CRPs which allow us to demonstrate the resilience of each examined design. While we are able to make a breakthrough in most of the proposed designs by utilizing only 100 k, the number of CRPs required increases with the number of components embedded in the design.

Since the security of the implemented designs is influenced by the inherited properties of each FPGA board, our experiments use CRPs generated from three FPGA boards. The reported results of all implemented designs are based on the average of 10 trials using more than 28 million CRPs.

4. Results and Analysis

4.1. Performance Results

Each of the proposed DAPUF designs and relevant XOR PUFs designs have been examined using five evaluation metrics. In Table 3, we also present findings of performance evaluations shown in [18] and our work, denoted by A and B, respectively. While the ideal values from both studies differ, the base calculations for these metrics in both studies are alike.

Table 3. Performance evaluation results for both XOR PUF and DAPUF designs.

Metrics	Study	2 XOR PUF	2-1 DAPUF	3 XOR PUF	3-1 DAPUF	4 XOR PUF	4-1 DAPUF	Ideal
Randomness	A	5.32	45.74	54.96	53.39	n/a	55.01	50
	B	0.66	0.63	0.97	0.83	0.90	0.94	1
Steadiness	A	1.1	9.68	1.176	11.79	n/a	26.6	0
	B	0.99	0.98	0.99	0.96	0.99	0.89	1
Uniqueness	A	5.38	46.37	6.34	50.24	n/a	50.02	50
	B	0.03	0.12	0.05	0.33	0.06	0.43	0.5
Normalized Uniqueness *	B	0.04	0.17	0.07	0.44	0.09	0.58	1
Correctness *	B	0.99	0.97	0.99	0.95	0.99	0.86	1
Diffuseness *	B	0.93	0.92	0.99	0.98	0.99	0.99	1

* These metrics are not reported in study A [18].

4.1.1. Randomness Experiment

Randomness metric measures how balanced a set of CRP is between 0′s and 1′s. As discussed earlier, the ideal value is 1 when the data are ideally balanced. Our experimental result in Table 3 shows that XOR PUF is more random than DAPUF when two and three components are used, while the story is reversed when four components are used in the implemented designs. Overall, the value of the randomness of both designs has a positive correlation with the number of components used.

4.1.2. Steadiness Experiment

Steadiness examines the ability to reproduce similar bit-responses when the same challenge is tested multiple times on the same device. The ideal value of steadiness is 1 when the responses from a specific CRPs were identical. In our experiment, XOR PUF designs exhibit much higher steadiness than DAPUF designs. Increasing the number of components in both designs have a negative impact on the steadiness of implemented designs. However, the problem is exacerbated in the case of DAPUF when steadiness declines from 98% in 2-1 DAPUF to less than 90% in 4-1 DAPUF.

4.1.3. Correctness Experiment

Correctness of the device measures how well the i-bit response to the ith challenge matches the benchmark output determined by the majority voting from the set of all challenges. Evaluation of XOR PUF shows that each i-bit is correctly identified in each experiment, regardless of how many components are used in the design. In contrast, increasing the number of components has an adverse effect on identifying the correct responses in DAPUF designs.

4.1.4. Diffuseness Experiment

Diffuseness evaluates how each key generated using a set of CRPs is different from any other key produced by the same device and the same design. Table 3 shows that increasing the number of components in both XOR PUF and DAPUF has a positive impact on how each key is uniquely identified. Overall, both XOR PUF and DAPUF perform well when the number of components is higher than 2.

4.1.5. Uniqueness and Normalized Uniqueness Experiment

Uniqueness evaluates how responses from one device can be distinguished from responses generated from another device when the same design is implemented and the same set of the challenge is given to both devices. Our experiments show that the uniqueness of any DAPUF designs surpasses all XOR PUF. Increasing the number of components has a substantial positive effect on the uniqueness of DAPUF, while it has a negligible influence on XOR PUF. The normalized uniqueness metric proposed in [13] output similar patterns to uniqueness metric proposed in [14] when XOR PUFs suffer from low uniqueness rates compared to DAPUF designs.

When comparing our results with [18], we see that both evaluations report lower randomness for designs that have fewer components, and the randomness for both XOR PUFs and DAPUF designs improves when more components are employed. Our results also match with the findings in [18] that increasing the number of components has an adverse effect on the steadiness of both XOR PUFs and DAPUF designs. When it comes to uniqueness, both works show that XOR PUFs suffer from very low uniqueness. DAPUF designs have a better uniqueness in both studies, but our experiments show that DAPUFs have lower uniqueness than reported in [18] using both uniqueness metric proposed in [14] and the normalized uniqueness proposed in [13]. Correctness, diffuseness, and normalized uniqueness metrics were not reported in [18], and 4-XOR PUF design was not implemented in [18].

4.2. Machine Learning Attack Results

Our study includes examining the resistance of all DAPUF design and relevant XOR PUF designs. As expected, each proposed DAPUF design has clearly shown higher resilience against machine learning attacks compared to their counterparts from XOR PUF as illustrated in Table 4. The responses of all XOR PUF designs are accurately predicted when 100k CRPs are obtained. On the other hand, increasing the number of XORed responses in DAPUF design enhances its resilience against machine learning attacks. When comparing the design of 2-1 DAPUF with that of 2-XOR PUF, we observe that the number of XORed responses is similar, which justifies its vulnerability against modeling attack. The output response of 3-1 DAPUF is the outcome of XORing six internal responses, which have been reported in [18] to be secure against machine learning attacks. Authors of [12] investigated the tolerance of 3-1 DAPUF using deep learning techniques and were able to impersonate 3-1 DAPUF using 17 million CRPs. Our modeling attack reduces the number of CRPs required to successfully attack 3-1 DAPUF design to 100 k CRPs [12].

Table 4. Modeling accuracy results for DAPUF 64-bit.

Training CRP	FPGA	Prediction Accuracy %											
		2 XOR PUF		2-1 DAPUF		3 XOR PUF		3-1 DAPUF		4 XOR PUF		4-1 DAPUF	
		Average	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best
100 K	A	97.82	98.23	92.49	93.16	94.36	96.39	85.21	85.89	94.98	95.33	60.08	60.61
	B	97.87	98.29	93.04	93.56	93.87	95.81	84.52	85.32	94.88	95.46	58.68	62.34
	C	97.74	98.35	92.64	93.23	94.61	96.45	84.10	84.82	95.22	95.95	58.13	58.77
	Average	97.81	98.29	92.72	93.32	94.28	96.22	84.61	85.34	95.03	95.58	58.96	60.57
500 K	A	97.97	98.38	95.14	95.27	96.40	96.84	90.42	90.83	95.94	96.19	63.34	64.27
	B	98.15	98.28	95.25	95.53	96.62	97.08	90.00	90.53	95.92	96.14	79.90	80.13
	C	98.01	98.24	94.14	94.27	96.44	96.74	90.16	90.46	95.96	96.11	80.15	80.57
	Average	98.04	98.30	94.84	95.02	96.49	96.89	90.19	90.61	95.94	96.15	74.46	74.99
1 M	A	98.23	98.38	95.70	95.83	96.91	96.74	92.00	92.24	96.23	96.30	79.31	80.50
	B	98.21	98.25	95.73	95.87	96.74	97.01	91.46	91.86	96.08	96.17	82.28	82.52
	C	98.26	98.35	95.44	95.59	96.51	97.05	91.43	91.87	96.25	96.45	82.81	83.05
	Average	98.23	98.33	95.62	95.76	96.72	96.93	91.63	91.99	96.19	96.31	81.46	82.02

Out of the three proposed designs of DAPUF, 4-1 DAPUF is reported to be secure against machine learning attack [11,12]. 4-1 DAPUF is a sophisticated design that requires 12 XORing internal output bits to generate the output response. In our modeling attack, we are able to successfully model 4-1 DAPUF that have not been impersonated at the time of writing of this paper even when 17 million CRPs are used [12]. Our modeling attack of 4-1 DAPUF achieved 81% average accuracy using only 1 million CRPs [12].

4.3. Preferred Design

Performance evaluation metrics, summarized in Table 5, have shown that there is not a strict preference of one design over the other proposed designs. XOR PUF designs with different components are more reliable in reproducing output responses from the same challenge. On the other hand, DAPUF designs have better uniqueness that differentiates the output key generated in one device from another key produced in a different device when the same challenge set is used. Authors of [18]

compared DAPUF with XOR PUF and conclude that 3-1 DAPUF is their preferred design in terms of resistance, uniqueness, and steadiness.

Experimental results confirm that DAPUFs are more secure against machine learning attacks than XOR PUFs. However, our experiments contradict with findings in [11] that 3-1 DAPUF is the preferred design. Our machine learning attack shows that 4-1 DAPUF is more resistant than 3-1 DAPUF when 1 million CRPs are required to successfully model 4-1 DAPUF while only 100k is enough to successfully attack 3-1 DAPUF. Performance evaluations also indicate that 4-1 DAPUF design has a better uniqueness, diffuseness, and randomness compared to 3-1 DAPUF design, while 3-1 DAPUF remains more stable. On the other hand, performance analysis show that DAPUFs are less stable compared to their counterparts of XOR PUFs, while DAPUFs have a better uniqueness. This leads us to the conclusion that none of the examined designs has a strict preference over other designs.

Table 5. Summary of preferred design.

Metric	Preferred Design	Correlation
Steadiness	XOR PUF	Negative
Correctness	XOR PUF	Negative
Uniqueness	DAPUF	Positive
Diffuseness	3 XOR PUF $\succ$ 3-1 DAPUF 4 XOR PUF $\prec$ 4-1 DAPUF	Positive
Randomness	3 XOR PUF $\succ$ 3-1 DAPUF 4 XOR PUF $\prec$ 4-1 DAPUF	Positive

5. Conclusions

This paper evaluates the performance and the security of all proposed designs of DAPUF, namely 2-1, 3-1, 4-1 DAPUF, and their counterparts from XOR PUF which are 2, 3, 4 XOR PUF. We generated different sets of real CRP to analyze the performance of each design using five evaluation metrics proposed in [13,14]. Our experiments show that DAPUF designs have better uniqueness, but this came at the cost in the stability of the responses generated from given challenge sets. XOR PUFs are more stable but suffer from low uniqueness since over 90% of responses generated in one device could be replicated on another device. Overall, increasing the number of components in any design has a positive impact on both diffusion and uniqueness, while it has an adverse effect on both steadiness and correctness of the responses generated from DAPUF designs.

Our analysis also covers the security of both DAPUF designs and their counterparts from XOR PUFs. Results of the machine learning attack shows that all XOR PUF designs and 2-1 DAPUF and 3-1 DAPUF designs can be impersonated when only 100k CRPs are attainable. Ref. [12] 4-1 DAPUF is a sophisticated design that has not been impersonated at the time of writing this paper even when 17 million CRPs are used. We are able to successfully predict the response of 4-1 DAPUF design with 81% accuracy rate using 1 million CRPs.

Our results contradict with [11] that 3-1 DAPUF is the preferred design and we show that there are no strict preferences toward any DAPUF over other designs. Experimental results in [11] show that the 3-1 DAPUF design has a better randomness and steadiness than 4-1 DAPUF while uniqueness is almost ideal in both designs, in which they conclude that 3-1 DAPUF is the preferred design. However, our experiments show that that 4-1 DAPUF is more secure and has better uniqueness than 3-1 DAPUF, while the responses of 3-1 DAPUF are more stable. Finally, stability of the responses is essential to the applicability of a PUF, and responses from XOR PUFs have been shown to be more stable than DAPUF designs.

Author Contributions: Methodology, M.A.A. and Y.Z.; hardware setup, M.A.A., Y.Z., K.T.M., and M.S.A; data generation, M.A.A., K.T.M.; software, M.A.A. and A.O.A.; formal analysis and writing—original draft, M.A.A.;

validation and supervision, Y.Z.; writing—review and editing, M.A.A., Y.Z., K.T.M., A.O.A. and M.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported in part by the National Science Foundation under Grant No. CNS-1526055, and the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia through project number 383.

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

References

1. Anderson, R.; Bond, M.; Clulow, J.; Skorobogatov, S. Cryptographic Processors-a Survey. *Proc. IEEE* **2006**, *94*, 357–369. [CrossRef]
2. Kocher, P.C. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 104–113.
3. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.
4. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical One-way Functions. *Science* **2002**, *297*, 2026–2030. [CrossRef] [PubMed]
5. 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; pp. 148–160.
6. Lim, D.; Lee, J.W.; Gassend, B.; Suh, G.E.; Van Dijk, M.; Devadas, S. Extracting Secret Keys from Integrated Circuits. *IEEE Trans. Large Scale Integr. Syst.* **2005**, *13*, 1200–1205.
7. Aseeri, A.O.; Zhuang, Y.; Alkatheiri, M.S. A Machine Learning-based Security Vulnerability Study on XOR PUFs for Resource-Constraint Internet of Things. In Proceedings of the 2018 IEEE International Congress on Internet of Things (ICIOT), San Francisco, CA, USA, 2–7 July 2018; pp. 49–56.
8. Rührmair, U.; Sehnke, F.; Sölter, J.; Dror, G.; Devadas, S.; Schmidhuber, J. Modeling Attacks on Physical Unclonable Functions. In Proceedings of the 17th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 4–8 October 2010; pp. 237–249.
9. Tobisch, J.; Becker, G.T. On the Scaling of Machine Learning Attacks on PUFs with Application to Noise Bifurcation. In *International Workshop on Radio Frequency Identification: Security and Privacy Issues*; Springer: Cham, Switzerland, 2015; pp. 17–31.
10. Mursi, K.T.; Zhuang, Y.; Alkatheiri, M.S.; Aseeri, A.O. Extensive Examination of XOR Arbiter PUFs as Security Primitives for Resource-Constrained IoT Devices. In Proceedings of the 2019 17th International Conference on Privacy, Security and Trust (PST), Fredericton, NB, Canada, 26–28 August 2019; pp. 1–9.
11. Machida, T.; Yamamoto, D.; Iwamoto, M.; Sakiyama, K. A New Arbiter PUF for Enhancing Unpredictability on FPGA. *Sci. World J.* **2015**, *2015*. [CrossRef] [PubMed]
12. Khalafalla, M.; Gebotys, C. PUFs Deep Attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs. In Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 25–29 March 2019; pp. 204–209.
13. 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, 6–8 December 2010; pp. 298–303.
14. 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*; Springer: New York, NY, USA, 2013; pp. 245–267.
15. Gassend, B.; Lim, D.; Clarke, D.; Van Dijk, M.; Devadas, S. Identification and authentication of integrated circuits. *Concur. Comput. Pract. Exp.* **2004**, *16*, 1077–1098. [CrossRef]
16. 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.

17. Suh, G.E.; Devadas, S. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In Proceedings of the 2007 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
18. Machida, T.; Yamamoto, D.; Iwamoto, M.; Sakiyama, K. Implementation of Double Arbiter PUF and its Performance Evaluation on FPGA. In Proceedings of the 20th Asia and South Pacific Design Automation Conference, Chiba, Japan, 19–22 January 2015; pp. 6–7.
19. Alkatheiri, M.S.; Zhuang, Y. Towards Fast and Accurate Machine Learning Attacks of Feed-Forward Arbiter PUFs. In Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing, Taipei, Taiwan, 7–10 August 2017; pp. 181–187.
20. Becker, G.T. The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs. In *International Workshop on Cryptographic Hardware and Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 535–555.
21. Maes, R. *Physically Unclonable Functions: Constructions, Properties and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 19 November 2013.
22. Rührmair, U.; Sölter, J.; Sehnke, F.; Xu, X.; Mahmoud, A.; Stoyanova, V.; Dror, G.; Schmidhuber, J.; Burleson, W.; Devadas, S. PUF Modeling Attacks on Simulated and Silicon Data. *IEEE Trans. Inform. For. Secur.* **2013**, *8*, 1876–1891. [CrossRef]
23. Zhou, C.; Parhi, K.K.; Kim, C.H. Secure and Reliable XOR Arbiter PUF Design: An Experimental Study based on 1 Trillion Challenge Response Pair Measurements. In Proceedings of the 54th Annual Design Automation Conference, Austin, TX, USA, 18– 22 June 2017; pp. 1–6.
24. Sahoo, D.P.; Nguyen, P.H.; Chakraborty, R.S.; Mukhopadhyay, D. Architectural Bias: A Novel Statistical Metric to Evaluate Arbiter PUF Variants. *IACR Cryptol. ePrint Arch.* **2016**, *2016*, 57.
25. 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.
26. Hawkins, D.M. The Problem of Overfitting. *J. Chem. Inform. Comput. Sci.* **2004**, *44*, 1–12. [CrossRef] [PubMed]
27. Majzoobi, M.; Koushanfar, F.; Potkonjak, M. Lightweight Secure PUFs. In Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 10–13 November 2008; pp. 670–673.
28. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
29. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980 .
30. Lawrence, S.; Giles, C.L.; Tsoi, A.C. Lessons in neural network training: Overfitting may be harder than expected. In Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA, USA, 27–31 July 1997; pp. 540–545.