SECURE DYNAMIC PUF FOR IOT SECURITY

A Thesis By

SHAILESH RAJPUT
ORCID iD: 0000-0002-2356-3215

California State University, Fullerton
Spring, 2023

In partial fulfillment of the degree:
Master of Science in Electrical and Computer Engineering

Department:
Department of Electrical and Computer Engineering

Committee:
Jaya Dofe, Department of Electrical and Computer Engineering, Chair
John K. Faller, Department of Electrical and Computer Engineering
Kiran George, Department of Electrical and Computer Engineering
Pradeep Nair, Department of Electrical and Computer Engineering

DOI:
10.5281/zenodo.7943535

Keywords:
hardware security, modeling attacks, cryptography function

Abstract:
The widespread adoption of Internet of Things (IoT) devices in various application domains has significantly improved the quality of life. However, the resource-constrained, heterogeneous, and low-power nature of these devices poses challenges in ensuring secure communication and authenticity. Physical Unclonable Functions (PUFs) offer a solution by creating a unique and device-specific identity through manufacturing process variations without requiring additional resources. To authenticate IoT devices, a challenge-response pair (CRP) is generated based on the unique characteristics of each device. However, the CRPs generated by PUFs often exhibit high correlation, making them vulnerable to modeling attacks. Despite the proposal of numerous intricate PUF architectures, such as XOR PUF and Interpose PUF (iPUF), the advancement in machine learning (ML) algorithms has enabled the modeling of attacks on these PUFs. In this study, experiments performed on field programmable gate arrays (FPGAs) demonstrate that the dynamic nature of the proposed PUF architecture makes it challenging for prevalent ML models to predict accurate PUF responses. Moreover, this work compares the performance of Logistic Regression and multilayer perceptron-based modeling attacks on Arbiter PUF, XOR PUF, and a proposed dynamic PUF. The experimental results demonstrate that the proposed dynamic PUF architecture outperforms in resilience to ML-based attacks and resource utilization, making it a viable option for IoT applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I express my profound gratitude to my thesis supervisor, Dr. Jaya Dofe, for allowing me to engage in research work under her mentorship and for her invaluable guidance throughout my graduate studies. Her vast knowledge, insightful perspectives, and unwavering support were instrumental in enabling me to complete my thesis. I would also like to sincerely thank my thesis committee members, Dr. Kenneth Faller, Dr. Kiran George, and Dr. Pradeep Nair, for their generous time and effort in reviewing my thesis. Their constructive feedback and insightful suggestions greatly enhanced the quality of my work.

I am deeply grateful to my family for their love and motivation during my academic journey. I am also incredibly grateful to my friends, Ravi Monani and Kriti Rai Saini, who offered me their expertise and advice on various aspects of my research and academic journey. I cherish their friendship and feedback

CHAPTER 1

INTRODUCTION

Introduction Hardware Secure Key Storage

In the digital age, ensuring the security and authenticity of Internet of Things (IoT) devices and data is crucial due to sensitive information being shared online, such as in banking and defense. IoT devices like monitoring systems, automated cars, medical equipment, home automation, and smart infrastructure devices transmit data online. Furthermore, several IoT devices authenticate a person's identity and can contain personal, social, and banking information. Therefore, if the security of these devices is breached, significant harm can occur. Cryptography secures critical information by encrypting and decryption of data by using cryptography keys. Cryptography and authentication protocols rely on secure key storage in non- volatile electrically erasable programmable read-only memory (EEPROM) and static random-access memory (SRAM). This approach has significant resource and area overhead and is susceptible to invasive attacks such as Side-Channel Attacks (SCA) and non-invasive attacks proposed in [1]. A compromised cryptographic key can jeopardize the authentication process of a device or user, potentially leading to the exposure of critical information. Furthermore, these devices can be breached due to the widespread deployment of IoT devices in public access areas. Storing a key identifier in a device helps identify the device. However, this method must be improved for privacy, authentication, and authorization. To achieve these goals, different approaches must be utilized. To address this security issue, a Physical Unclonable Function (PUF) is proposed as a more secure and cost-effective solution than conventional key storage methods to authenticate a device [2]. PUF relies on manufacturing variations of Integrated Circuits (IC) at the nanoscale. The variations are so unique and small that they cannot be replicated even by the original manufacturer of the IC.

Threats and Challenges on IoT Security

Several IoT devices connect wirelessly over the internet and are usually installed in publicly accessible places, making them susceptible to attacks. Several traditional cryptography algorithms for

encrypted secure data transmission, such as Advanced Encryption Standard (AES), require ample storage. Some IoT devices might have low storage of around 512 bytes, while AES might take storage up to 800 bytes [3]. Low-end cipher encryption algorithms such as RC5 have been proposed for IoT de- vices. But still, it is susceptible and vulnerable to attacks and requires non-volatile memory storage on the device. The cost, area, and vulnerability to invasive and non-invasive attacks are problems IoT security faces. Encryption mechanisms using PUF can be easily built on System-on-Chip (SoC) and Field Programmable-Gate Array (FPGA). PUF requires less area comparatively than traditional cryptographic methods. Initially, FPGA was used to make a prototype and simulate designs for debugging purposes. But due to customization in FPGA, nowadays FPGA based IoT devices have been proposed [4]. However, FPGAs do not have non-volatile on-chip memory making it difficult to store keys on the FPGA. Hence, PUF can be a promising solution for encrypting IoT and FPGA devices.

Motivation

Using the unique manufacturing variation of ICs, PUF serves as the device's finger- print. PUF since its proposal after 2002, there has always been an intense debate on its *Unclonability* feature. The unexpected manufacturing variations are uncontrolled and cannot be cloned in the form of duplicate physical ICs. But a Modeling attack on PUF structure has been proposed, which can predict the response of PUF devices [4], [6]-[9]. This suggests that the electronic cloning of PUF is possible, and the security of IoT and FPGA can be compromised. Since its inception, there has been competitive research between attack models and complex PUF architecture. Also, for making unpredictable PUF, the intricate designs that were proposed earlier leverage more resources which is not beneficial in the case of IoT devices. One way or the other, PUF cannot be proven as a commercial and secure device for cryptography and Security. The desired reliability property of PUF increases the correlation between CRPs. This constant property of PUF has to be reconsidered, and dynamic behavior to avoid chances of correlation is much needed for PUF resiliency against

Modeling Attacks. Our research discusses introducing dynamic behavior in PUF and making complex PUF against machine learning attacks.

IoT Authentication Protocol Using PUF

PUF-based security protocol can authenticate low-energy and smaller IoT devices to reduce the resource overhead for cryptography. Authentication using PUF involves two phases: Enrollment and Verification.

Enrollment

In the enrollment phase, the database of Challenge-Response pairs is generated from the PUF device and stored in the authenticated server. This database is assumed to be stored in a secured space and cannot be accessed by adversaries. When an authentication request is sent from the client PUF, the database CRP is requested and matched against the PUF response.

Verification

During the authentication phase of the device, the server sends the challenge from the database and collects the response generated by the PUF devices. These responses are compared with the response bits in the database. If the generated response matches with the database in the server, the device is verified, and a secure connection is established, as shown in Figure 1. For added security mechanisms, the server, for authentication purposes, uses one CRPs for only single-time authentication. Every new authentication will be a new occurrence of the challenge for verification purposes. These stages are explained in more detail in Chapter 2. PUF can be used as added security primitive with existing cryptography protocols to improve the device’s security.

Thesis Organization

This section outlines the organization of the rest of the thesis to address a better understanding of the article. The concept and background of PUF architecture are discussed in the next chapter. Furthermore, Chapter 2 includes the working principle and classification of PUF, the complex architecture proposed earlier, and quality metrics for PUF.

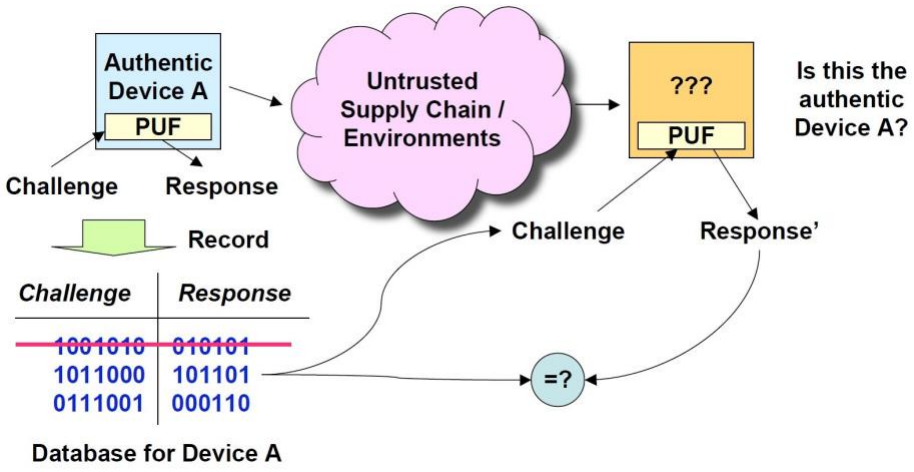

Figure 1. PUF-based Device Authentication [10]

Chapter 3 of the thesis delves into the potential risks and obstacles PUF devices face. This section focuses on the nuances of modeling and traditional attacks, which are elaborated upon in detail. Moreover, this chapter expounds on the susceptibility of PUF devices to machine learning algorithms, particularly logistic regression, and deep learning neural networks.

In Chapter 4, we presented new PUF architecture to increase the vulnerability against machine learning modeling attacks by introducing reliable noise in the PUF model. This chapter also describes the implementation strategy and experimental setup, such as the software and hardware used for the research. Furthermore, we have summarized our results against a machine learning attack on the proposed PUF, along with resource utilization and quality metrics of the PUF.

CHAPTER 2

PUF BACKGROUND

PUF Introduction

There is a slight difference in ICs at the nanoscale range, even though they are manufactured from the same wafers. These differences are unique and so random that the original manufacturer cannot replicate the same behavior. This unique behavior can generate cryptography secure keys using differences in delay, process, frequency, and current [2]. The process variation in ICs is within die and die-to-die variations [11]. These variations are caused due unexpected and uncontrolled differences in wafers within a die or die-to-die. Dopant concentration within the die is a prominent reason for the variation within the die. Thus, these random variations are unique to each device and are termed fingerprints of the device. PUF takes the challenge bits (C) as input and produces the random response (R) bits depending on the device interconnect. This implies if the same challenge C is given to two different IC will produce a different response R.

The stages for authentication using PUF for IoT devices are depicted in detail in Figure 2. The PUF device's Challenge-Response Pairs (CRPs) are stored in the server's database in a secure environment, and the PUF client is installed on IoT devices. When a request is made from the client, the challenge is transmitted from the server to the PUF client device for a secure connection. PUF client sends unique response bits compared with the server's CRP database. If the response generated by the device matches the CRP database on the server, the device is considered authenticated, as illustrated in Figure 2. For added security, the IoT authentication protocol can be designed so that every time unique CRPs are used, if the communication channel is attacked, the actual key cannot be predicted due to the random behavior of PUF. The Figure 3 shows that responses R1 and R2 generated from two ICs for the same challenge are not identical.

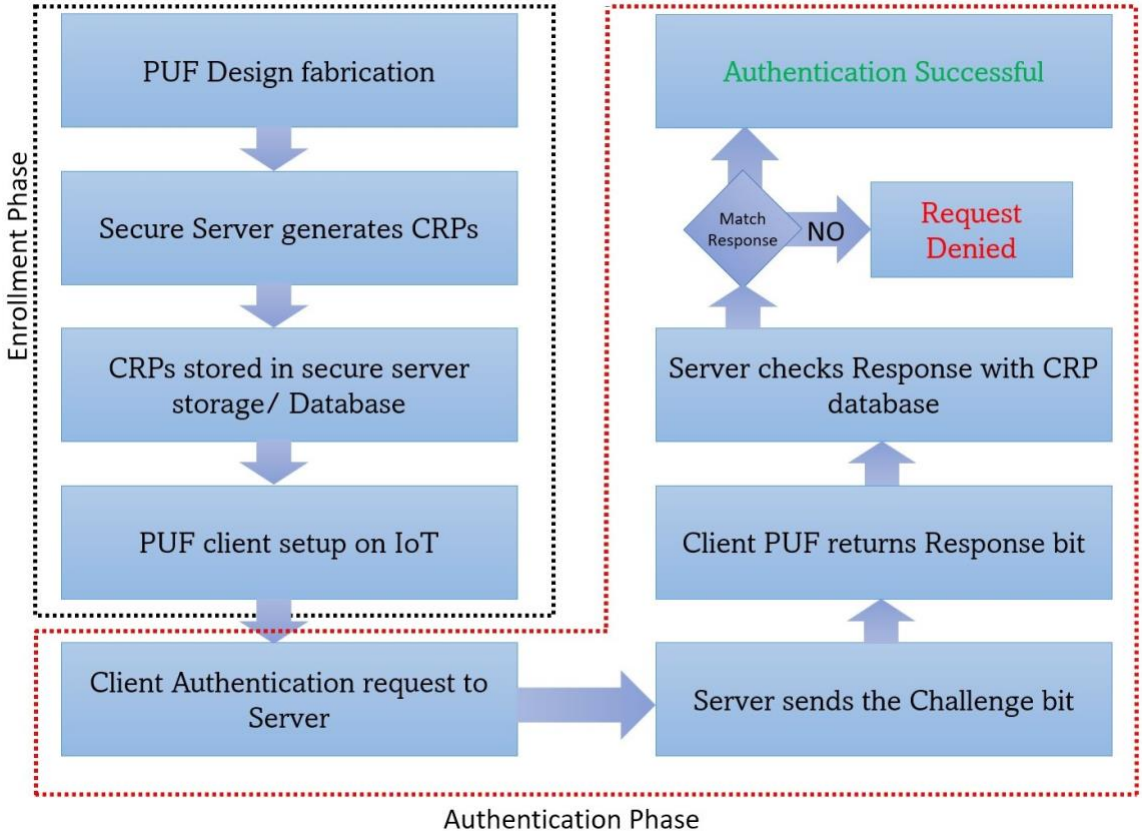

Figure 2. Stages for authentication using PUF

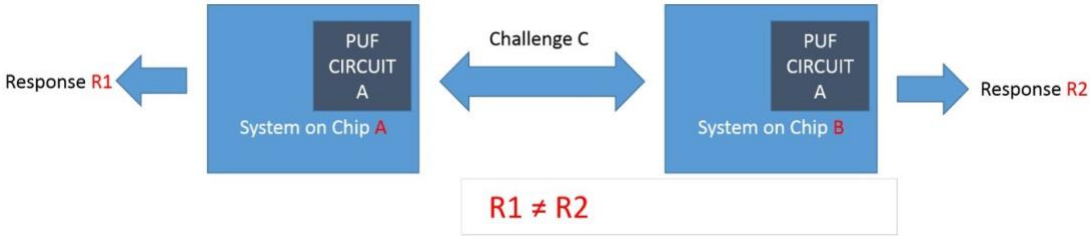

Figure 3. Response produced on PUF circuit of two different ICs [12].

PUF Classification

The classification of PUF differs depending on the security or material required. It doesn't follow explicit classification, but the researcher classified the PUF mainly into two types, classification based on the material used and on the security of the device as shown in Figure 4.

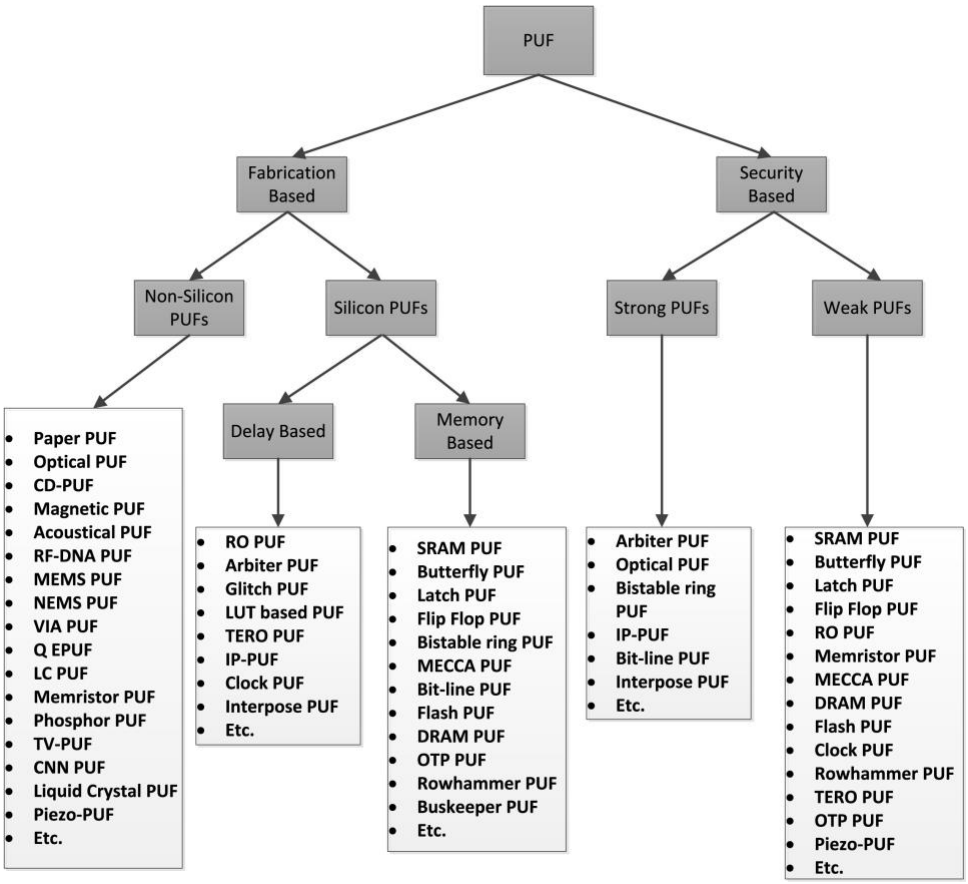

Figure 4. PUF Classification based on security and material [13]

Classification Based on Material

Based on the material used, PUF is further classified into two types: Silicon PUF and Non-Silicon PUF. Optical PUF, Acoustical PUF, magnetic PUF, and PUF are Non-Silicon PUF. While on the other hand, Silicon PUF relies on the process variation of Silicon Devices. Delay Based PUF, Memory-based PUF, and other electronic PUF. The Delay based PUF uses the delay condition between two identical paths to generate response R. Some PUFs, like Ring Oscillator PUF (RO-PUF), uses frequency variation to derive CRPs. This research will focus on Silicon PUF that can enhance IoT and FPGA security.

Classification Based on Security

Depending on the ability to generate the number of CRPs, the PUFs are classified as strong and Weak PUFs. Strong PUFs can generate more CRPs, while weak PUFs generate fewer pairs using the same number of resources. Due to the fewer CRPs, the weak PUF accounts for low security compared to the strong PUF. The attacker cannot replicate the ICs even for the weak PUF,

but due to the lower number of CRPs, it is possible that the attacker can keep a record of all the CRPs generated by weak PUF. APUF, Interpose PUF (iPUF), Double Arbiter PUF (DAPUF), XOR PUF and its variants, and Optical PUF are some examples of Strong PUF. In contrast, SRAM PUF, RO-PUF, Butterfly PUF, and DRAM PUF are examples of weak PUF.

Our research focuses on Delay based Silicon PUFs, as they are strong PUFs and capable of generating a higher number of CRPs. Some of the delay PUFs, such as RO PUF, are incapable of the high number of CRPs. We will discuss it in more detail in the following sections on delay based PUF variants. Due to more security in terms of CRPs, strong PUFs are a strong candidate for authentication, while weak PUFs can be used as identification, random number generators, and key generations.

Our proposed research focuses delay based APUF due to its ability to generate more CRPs. So, we will discuss more in detail on delay based PUF variants in the following section.

Delay Based PUF Variants

The PUF that relies on the delay difference between two paths is generalized as delay based PUF. RO PUF, Arbiter PUF and its Variants, XOR PUF, and Variants, Double Arbiter PUF [14], Anderson PUF and Interpose PUF are some examples of delay based PUF.

Ring Oscillator PUF

This PUF uses frequency comparison between two oscillators. Due to variations in ICs, even though the identical oscillators are compared, there is a slight difference in frequency. The RO PUF was proposed by Suh and Devadas in [10]. As shown in Figure 5, the oscillator frequency is compared using the counter. Two multiplexers are used to select the oscillators for frequency comparison. The select input is challenge C for the RO PUF, and the output after frequency comparison is Response R. The comparison of two oscillator frequencies generates response 10, depending on the manufacturing variations. As the number of oscillators is limited for PUF, very few responses are generated. For a given N number of oscillators in RO PUF around log2(N !) CRPs, but for accuracy and reliability, oscillators can be used only once to avoid correlation between oscillators

[10]. The author in [10] states that RO PUF allows easy implementation and is more reliable but power consuming.

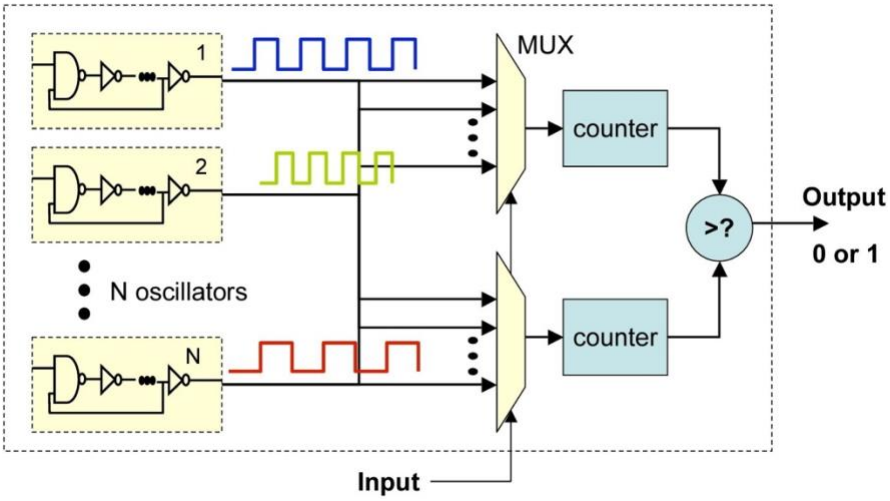

Figure 5. Architecture of Ring Oscillator PUF [10].

Arbiter PUF

Arbiter PUF (APUF) is mostly researched for its easy implementation and ability to produce many CRPs. For n-bit Challenge inputs, the APUF can produce 2n numbers of CRPs. Arbiter PUF contains the chain of Multiplexer as shown in Figure 6 was proposed by Lee et al. in [15]. The race condition between two identical chains of the multiplexer is sent to the arbiter, i.e., flip-flop. There is a difference in the delay of both paths due to the nanoscale manufacturing variation in the ICs, even though they are manufactured from the same wafer. These variations are so random to produce a different response for different challenge inputs.

The same Challenge bit C[i], for APUF, is given to select input of the multiplexer of both paths as shown in Figure 6. The upper and lower paths are identical, with the same number of multiplexers. Figure 7 shows the challenge inputs X of 128- bits, and Response Y is Generated at the end. The value of challenge bits alters the path of multiplexers and has a unique delay path for each challenge bit. If the upper path has a lower delay and arrives first at the flip-flop, the response 1 will be generated, as when the lower path arrives at the clock terminal, the value on D will be 1. While if the

lower path arrives earlier, the response 0 will be generated at Y, as the value at D will be 0 due to a higher delay at the upper path.

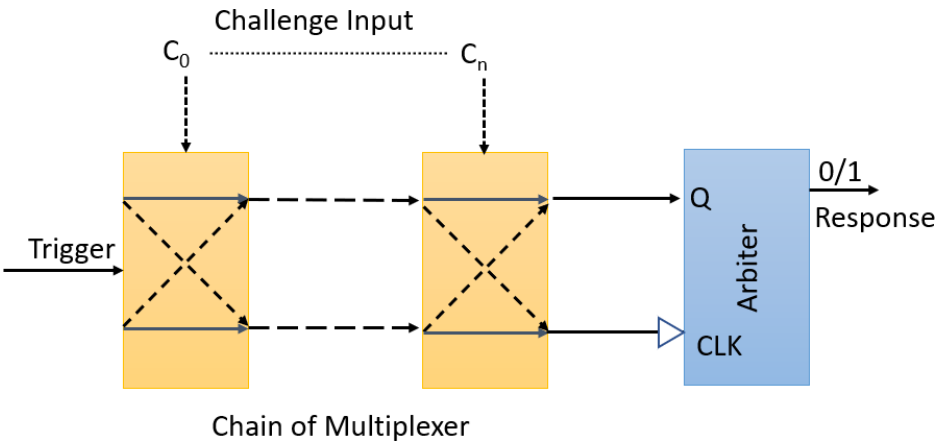

Figure 6. Arbiter PUF

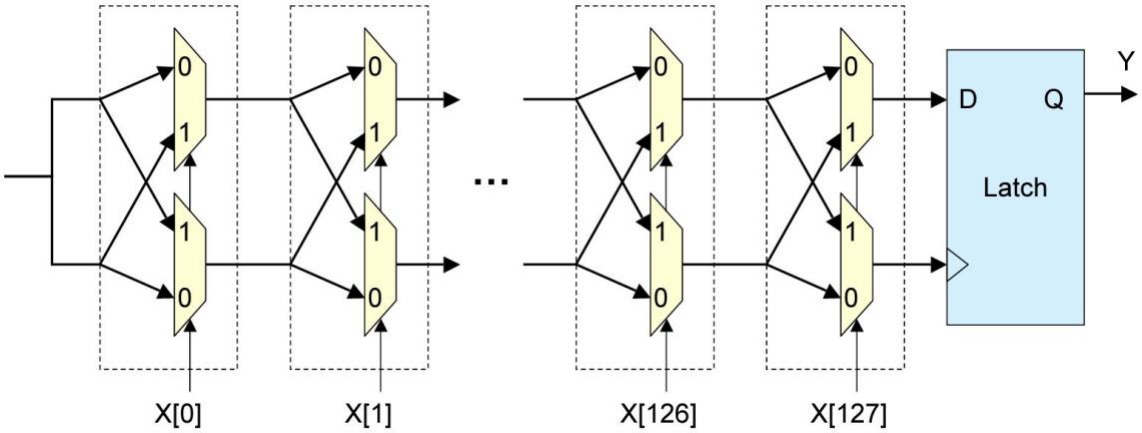

Figure 7. Two identical multiplexer paths for APUF [10]

To generate more unique responses, more path combinations are required. As APUF uses a multiplexer for path design, the select input of the multiplexer can be used to switch paths and generate a unique delay path for every different combination. The multiplexer block for switching of delay path in multiplexer is depicted in detail in Figure 8. As shown in Figure 8, if the challenge or select input to the multiplexer is 1, the path will be cross-coupled, or else the delay path will be in parallel. Due to this switching mechanism by a multiplexer, the APUF and other APUF-derived PUFs are also classified as multi-challenge PUFs (MCPUFs). While PUFs such as ROPUF and SRAM PUF don't allow configurable challenge bits, they are classified as single-challenge PUFs (SCPUFs) [16].

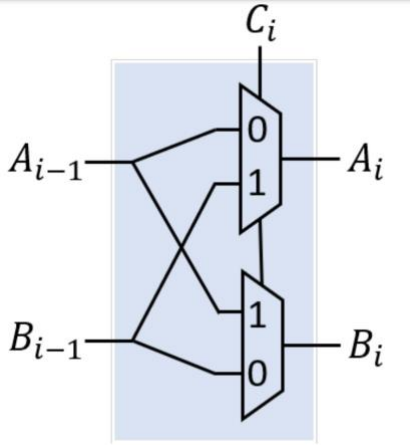

Figure 8. PUF Switching paths in multiplexer [17]

The architecture of APUF discussed generates the one-bit response. To generate multiple-bit responses using APUF, multiple chains of APUF can be implemented on the circuit. Due to its ability to produce more CRPs and, easy implementation, low resource utilization, APUF is a strong candidate for authentication purposes. APUF and its variants can also be implemented on FPGA with careful floorplanning.

As FPGA is effortlessly configurable and cheaper than ASIC, we used FPGA for our research purpose. To improve the security of APUF, many other robust variants of APUF are proposed, such as Double Arbiter PUF (DAPUF) [18], XOR Arbiter PUF Variants [15], and Interpose PUF [19]. These strong APUF variants will be discussed in detail in further sections. Furthermore, we will propose a dynamic PUF architecture in Chapter 4.

XOR PUF

The XOR APUF was initially proposed by Suh and Devadas in [10], to avoid modeling attacks on PUF. APUF architecture allowed direct integration of the input and output of PUF, making it easy for modeling attacks to learn the behavior of PUF. The idea of XORing the output of multiple PUF, as shown in Figure 9, makes the actual output more secure by not exposing the response bit of APUF directly.

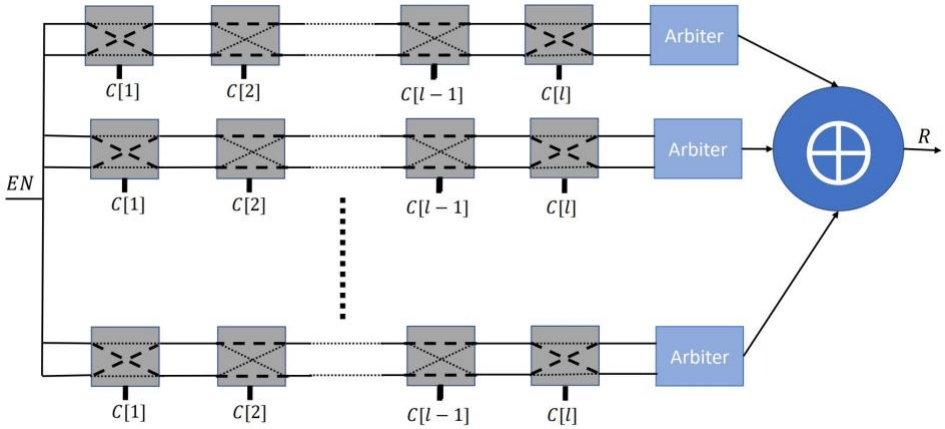

Figure 7. Architecture of n-XOR PUF [20].

The Figure 9 shows the architecture of XOR PUF for n number of APUF rows. The responses are XORed to get the final XOR. XOR PUF is named depending on the number of rows used to make PUF. N-XOR has n rows of APUF, i.e., 2 and 3 rows of XOR are termed 2-XOR PUF and 3-XOR PUF. Adding more rows makes the PUF more secure however decreases the stability of response bits for PUF. At the time of inception, XOR PUF was supposed to provide resiliency against modeling attacks. But later, an ML-based attack was proposed, which could predict the response from XOR PUF with an accuracy of over 95%.

Double Arbiter PUF

In DAPUF, two or more identical APUFs are used to derive the response bits. The DAPUF, as shown in Figure 10, the paths from different chains are used to derive intermediate responses, which are r1 and r2. This response can be XORed to produce the complexity of APUF. DAPUF has similar architecture and resource utilization to XOR PUF. Figure 10 depicts the architecture of 2-1 DAPUF, as two chains of APUF are used to derive r1 and r2 responses. Due to biased responses in APUF, 2-1, DAPUF has comparatively low uniqueness.

Along with 2-1, DAPUF to introduce more uniqueness, Machida et al. proposed 3-1 DAPUF. The 3-1 DAPUF introduces more randomness, reliability, and uniqueness property in the PUF than the 2-1 DAPUF variant. Compared to XOR-PUF, DAPUF showed better results in uniqueness and randomness, while XOR-PUF was more reliable [14].

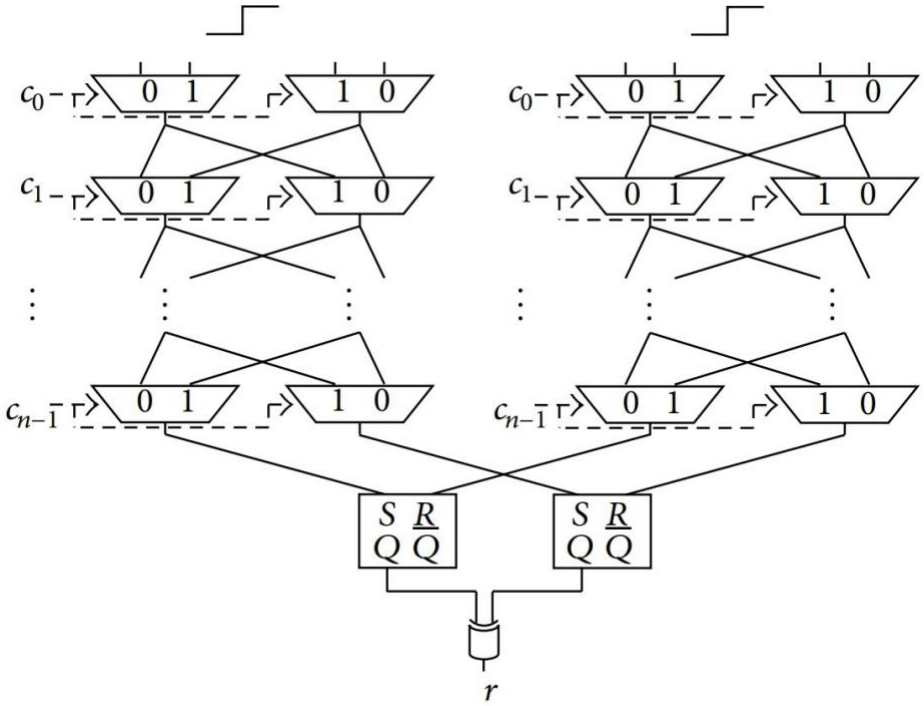

Figure 8. Architecture of 2-1 DA PUF [14]

Interpose PUF

Interpose PUF (iPUF) was proposed by Nguyen et al. for advanced security of PUF devices against modeling attack [19]. The (x,y)-iPUF is derived from two levels of XOR PUF. The upper level consists of x-XOR PUF, and the lower level is derived from y-XOR PUF as shown in Figure 11.

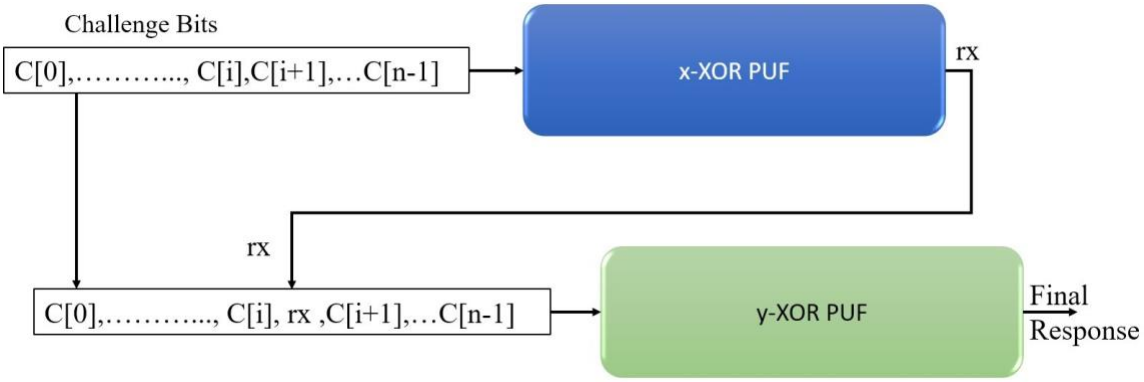

Figure 9. Architecture of (x,y)-Interpose PUF.

The intermediate response generated from the upper-level x-XOR PUF is interposed with the challenge bits of lower-level y-XOR PUF. To interpose one response bit of the upper level to the lower layer of XOR PUF, the upper consist of n-bit XOR PUF, while the lower level consists of (n+1)

y-XOR PUF. The final response of iPUF is the output of y-XOR PUF, as shown in Figure 11. The x and y represent the number of APUFs used to make XOR PUF in the upper and lower levels.

PUF Quality Metrics

To assess the performance of the proposed PUF designs, it is crucial to consider several essential metrics, as discussed in the following sections.

Uniqueness

As PUF is proposed on the concept that every ICs have unique manufacturing behavior, the response generated from PUF must be unique to the device. That suggests if the same challenge is applied to two different PUFs, it must produce different challenge bits. The Hamming Distance (HD) between responses of multiple PUF devices used for deriving uniqueness. For a one-bit response PUF function, the ideal value of PUF for Uniqueness is 50%. If two chips, i and j (such that i ≠ j), generates n number of response bits R_i and R_j for the same challenge C, the uniqueness among k chips is evaluated by the authors Maiti, Gunreddy, and Schaumont as

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

Uniformity

PUF generates responses of 1 or 0; the randomness/uniformity property of PUF is the ability of PUF to not produce the same response bits for all given challenges. The randomness property ensures that PUF does not produce identical response bits for every challenge. The uniformity property ensures that the proportion of 1's and 0's in the response bits is evenly distributed for all the given challenges. In an ideal scenario, PUF responses should exhibit a randomness of 50%.

$$Uniformity = \frac{1}{n} \sum_{l=1}^{n} R_{i,l} \times 100\% \quad (2)$$

Reliability

Reliability, also called the steadiness of the PUF function, is the ability to reproduce response bits. As PUF circuits account for variations in ICs, there are chances of impact on response bits due to environmental noise. An ideal PUF must always produce the same response bit for the same challenge. It should not change due to temperature or other environmental noise. Ideally, the value

for the Reliability/Steadiness of the PUF function is 100%. Intra Hamming Distance (HDINTRA) calculated using the response of device 'i' at different conditions [21]. Equation for HDINTRA given as

$$HD_{INTRA} = \frac{1}{m}\sum_{t=1}^{m}\frac{HD_{R_i,R'_{i,t}}}{n} \times 100\% \quad (3)$$

Where R_i and $R_{i,j}$ are responses of device 'i' at different environmental conditions or voltages. The equation defines reliability as,

$$Reliability = 100\% - HD_{INTRA} \quad (4)$$

As PUFs are mainly designed for low-cost IoT devices, cost and resource utilization has also become the prominent quality metrics for PUFs. With the advent of advanced ML algorithms, it was possible to replicate the PUF behavior. So, for security concerns, along with three fundamentals of PUF devices, PUF must also be evaluated for these properties.

CHAPTER 3

ATTACKS ON PUF

The first PUF was proposed assuming that PUF can be used for IoT security and encryption. The unclonable and unpredictability was the pillar on which the PUF mechanism was suggested as a solution for cryptography and identification. It is almost impossible to replicate the IC at the nanoscale even by its original manufacturer. This belief is true for cloning the IC physically. Even if the attacker tries to clone the IC, it requires extensive lab equipment and time, which cannot be done in public areas. But it was possible to replicate the uncontrollable behavior of ICs by training CRPs on Machine Learning (ML) Modeling algorithms. The PUF is not unclonable anymore as its behavior can be modeled using ML algorithms. In this section, we will discuss the possible attacks on the security of PUF.

The attacks on any ICs are broadly classified into two types invasive and non-invasive attacks. We will discuss the basics of this attack to understand the scope of a possible attack on a PUF device. Successful modeling attack on PUF is discussed in detail in other sections.

Invasive Attacks

For an invasive attack, the attacker requires access to the original IC that needs to be attacked. We will discuss two infamous invasive attacks on ICs in the following subsections.

Physical Attacks

The physical attack involves physically characterizing ICs for intrinsic behavior at the transistor and gate levels. To characterize the IC behavior Tajik et al. proposed photonic emission-based analysis of PUF function [22]. This type of attack is expensive, and exploiting the PUF behavior using this attack may sometimes damage the PUF device.

Cloning Attacks

For a cloning attack, the attacker tries to clone the ICs with identical behavior, with the same manufacturing variation. This kind of attack requires expensive lab equipment and is harder to achieve a successful attack. The authors of Zeitouni et al. have proposed an easy cloning attack

based on the remanence decay method [23] on SRAM PUF. However, due to complex and nanoscale delay connectivity in delay-based PUF, it is still challenging to clone the PUF devices.

Side Channel Analysis

Side Channel Analysis (SCA) exploits the physical information of the ICs, such as leakage current, power consumption, junction temperature, and electromagnetic radiations, to extract encrypted cryptographic keys. Several types of research exploit SCA analysis attacks on PUF function [24]. The SCA attack on PUF and its resiliency against such attack is discussed by Aghaie and Moradi on iPUF [25]. Aghaie and Moradi proposes a Boolean masking method to counter SCA attacks on PUF. Later Radha Krishnan presented secure techniques with less resource overhead by using AES [26]. SCA on PUF can be avoided by Boolean masking, obfuscation, and AES techniques. The current attack strategy, as we discussed, relies on having access to the device in a controlled lab environment. Some proposed attacks carry inherent risks or can be mitigated by proactively implementing countermeasures. However, it's essential to acknowledge that modeling attacks on PUFs (Physical Unclonable Functions) are a substantial risk as they don't require direct access to the device during the attack. An attacker can launch a successful attack by modeling the PUF device earlier by recording CRPs to predict its behavior accurately. Therefore, addressing modeling attacks on PUFs should be a priority in developing effective countermeasures. This will help to enhance the security of PUF-based systems and safeguard against potential threats. So, the proposed modeling attacks on PUF are addressed in the following section.

Modeling Attacks

The machine learning bases modeling attacks on PUF can be sub-classified as semi-invasive attacks. This type of attack requires the adversaries to record CRPs using the original PUF device that needs to be attacked. This attack assumes that the attacker can access the PUF device and generate CRPs for making a dataset and training the ML model. The phases of modeling attacks using ML algorithms are shown in Figure 12. The attacker collects the CRP data by breaking into the connection channel of the PUF client and server. The data is then pre-processed and transformed to

fit into the model and predict the client PUF response. The following section discusses two prominent attacks: ML and LR-based modeling attacks.

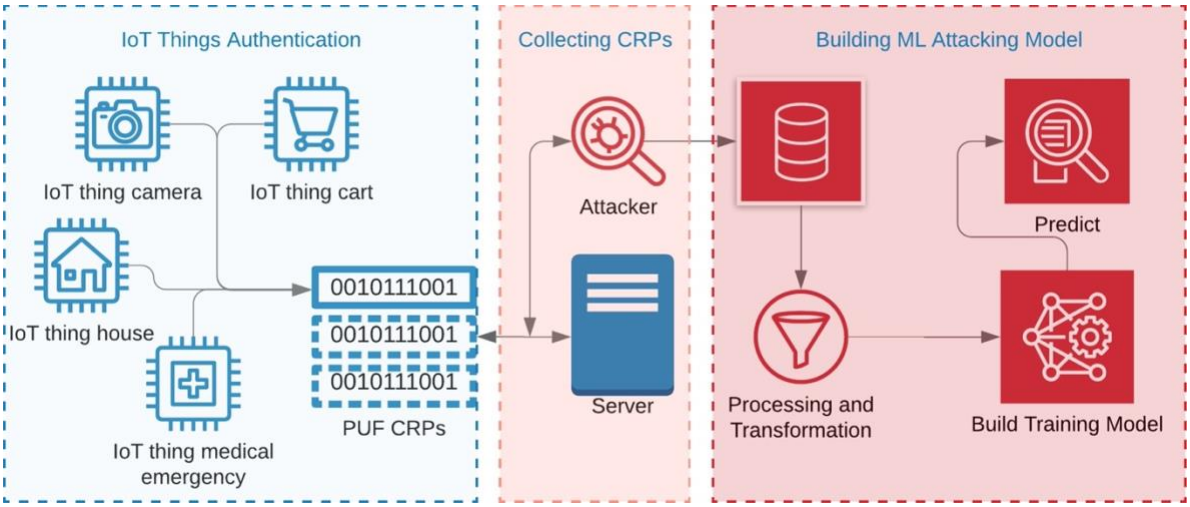

Figure 10. ML-based modeling attack on PUF [27].

Logistic Regression

The logistic regression model is a powerful statistical tool to identify patterns and relationships between input data and output labels. In the context of PUF, where the CRPs do not exhibit linear relationships, the logistic regression model has been demonstrated to be particularly effective. Previous studies have shown the suitability of logistic regression, mainly when using the RProp optimization algorithm, for PUF-based authentication, with reported accuracy rates of around 99% for Arbiter PUFs (APUFs) [8]. This research used the linear additive delay model to generate APUF CRPs, which was considered a reliable simulation model for delay based PUF devices in earlier days. The LR attack by Rührmair et al. was later modified in year 2015 to lower the training time of the model with around 1 to 6.5 hours for large XOR PUF circuits Tobisch and Becker. The LR-based modeling attack had higher training time and low accuracy on breaking iPUF, Feed-Forward PUF, and XOR PUF [4], [8].

Artificial Neural Network

Artificial Neural Network (ANN) is an ML algorithm that uses layers of perceptron/neurons to predict the response bits using challenge bits. ANN contains multiple layers of perceptron (also

referred to as neurons), with each perceptron having some values of weights and biases, as shown in Figure 13. The value of weights and biases are defined by training the model on the dataset and updated to predictthe output. The main goal of the multi-layer perceptron method (MLP) is to map the input data with the output label. Backpropagation and optimizers increase the accuracy of the output label's prediction. Backpropagation is a technique that involves calculating the gradients, later used for updating the weights and biases of neural networks by optimizers to obtain a correct prediction. The layers of ANN can be subdivided into three main layers of neurons: the input layer, the hidden layers, and the output layer. Once the ideal model achieves good accuracy on the training set, it is expected to perform well on unseen data.

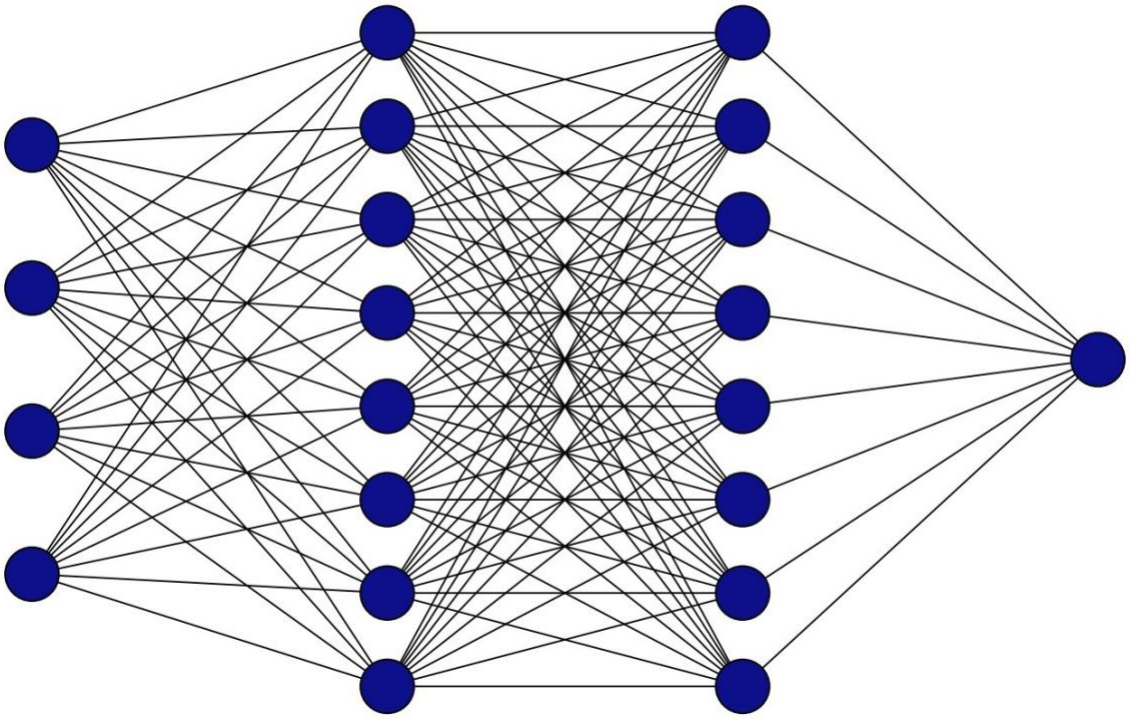

Figure 11. Deep Learning Neural Network Architecture.

The first attack using Deep Learning (DL) modeling was proposed by Ikezaki, Nozaki, and Yoshikawa in 2016 [9]. The authors in this research used the raw dataset of CRPs to train the model. Unfortunately, the attack had low accuracy of around 58%, i.e., near to random guess probability of 50%. Earlier in Chapter 2, it was discussed that complex architecture of PUF such as Feed Forward Arbiter PUF, XOR PUF, and iPUF was proposed for resilience against machine learning attacks. The

first attack model using multi-layer perceptron ANN was proposed in 2017 by Alkatheiri and Zhuang on Feed Forward Arbiter PUF with a maximum accuracy of around 96%, which was secure against modeling attack by Rührmair et al. [6] [8]. Later in year 2018, an MLP-based attack on XOR PUF was proposed by Aseeri, Zhuang, and Alkatheiri with a shallow attack time [30]. This attack by Aseeri, Zhuang, and Alkatheiri successfully break into large XOR PUF with an attack time of around two hours. In year 2019, Santikellur, Bhattacharyay, and Chakraborty proposed successful attacks on APUF and its complex variants, such as IPUF and XOR PUFs. The work by Santikellur, Bhattacharyay, and Chakraborty showed prediction accuracy of over 97% on XOR PUF and IPUF [28]. After that, Wisiol et al. in the year 2021, modified the model proposed by Aseeri, Zhuang, and Alkatheiri to reduce the complexity of the model [7]. The research by Wisiol et al. was faster and more robust in predicting the responses of iPUF [19], Double Arbiter PUF [14], XOR PUF with a lesser number of CRPs [7]. The main difference in the model by Wisiol et al. was tanh activation function was used instead of ReLU. Also, the model has fewer neurons than the model proposed by Aseeri, Zhuang, and Alkatheiri. The details of the optimizer and layers used for both MLP model is in Table 1. The iPUF discussed in Chapter 2 was proposed as robust architecture to tackle modeling attacks on iPUF. But the splitting attack on iPUF was proposed by Wisiol et al. by dividing the attack model for the upper and lower chain of Interpose-PUF with an accuracy of more than 95% [29].

Table 1. Multi Layer Perceptron Parameters

Parameters	MLP 2018 [30]	MLP 2021 [7]	CNN
Optimizer	ADAM	ADAM	ADAM
Learning Rate	1×10^{-3}	1×10^{-3}	1×10^{-3}
Activation Function	ReLU	tanh	ReLU and Softmax

Convolution Neural Network

The Convolution Neural Network (CNN) is robust architecture mainly for image prediction. The very first successful CNN architecture was proposed by Lecun et al., which became the revolution for

modern-age ML techniques [31]. This researchuses Machine Learning Based modeling attacks using Deep Learning Convolution Neural Networks (CNN). CNN is more efficient than other deep learning models, especially on image data. To the best of our knowledge, the application of Convolutional Neural Networks (CNN) to predict the response bits of Arbiter PUFs (APUFs)has yet to be previously explored.

We researched to predict the response of PUF-based authentication using 2d CNN deep learning techniques. For this modeling attack using two-dimensional CNN, weneed to convert and reshape the binary challenge bits into a two-dimensional arrayof 8x8 size and map with response bits. The CNN model under consideration features two convolution layers, three dropout layers, and one dense layer. The activation function used throughout the model is ReLU, except for the final layer, which utilizes a softmax activation function. The input shape for this model is (8,8,1), andthe feature size for the first convolution layer is (3,3). The model utilizes challenge bits as the input sample and response bits generated from an FPGA as the output labels. The CNN model is designed to avoid over-fitting on the training data set by using dropout layers and regularization on the hidden layers. These dropout layers work by randomly deactivating a certain percentage of the hidden neurons, thereby reducing the complexity of the model and preventing over-fitting. Additionally, the model utilizes a softmax activation function in the output layer and employs a sparse categorical cross-entropy loss function. The accuracy of the CNN model in predicting response bits is lower than that of the ANN model. Also, training the CNN model requires a higher training time than the ANN. The parameters used for the CNN-based modeling attack are depicted in Table 1. The maximum accuracy of the 2d CNN model on the 100k CRP dataset was around 72% for 2-XOR PUF and 73% for APUF, which is close to a random guess of 50%. The 2-d CNN model that is believed to perform very well on two-dimensional image data doesn’t perform well on CRPs prediction.

In conclusion, MLP and LR-based attacks were highly robust in predicting the response bits, even for the complex architecture. Though modeling attack initially requires device access to record CRPs, they are still a potential threat to the unclonable feature of PUF. More randomness in response

generation can help reduce modeling attacks on PUF devices. We will discuss our proposed concept of how adding reliable noise in PUF can confuse machine learning algorithms in Chapter 4.

CHAPTER 4

PROPOSED DELAY-BASED PUF

Motivation

The machine learning algorithm is used in a modeling attack to predict the behavior of PUFs based on their static and linear behavior. As PUF responses are directly derived from the challenge inputs of PUF, CRPs are highly correlated. The correlation of response R with challenge bit C should deviate as a countermeasure for the modeling attack. In the past, several researchers attempted to introduce dynamicity into the PUF function. One such software based Dynamic PUF (DPUF) incorporated dynamicity, which was proposed by Xiong et al. in the year 2019. This software utilized the timing of the query and the physical properties of the PUF to generatedynamic response bits [32], [33]. However, this research did not truly involve adding dynamicity at the hardware level. In 2022, Wang et al. proposed a hardware-based dynamically configurable hybrid (DCH) PUF architecture using LFSR. However, the introduced PUF was not resource-constrained and utilized more hardware resource than the underlying basic PUF [34]. The ultimate goal of adding complexity and dynamicity to PUFs with limited resources remains a challenge. In the following section, we introduce the concept of a dynamic architecture with limited resources by incorporating the idea of reliable noise.

Proposed Arbiter Skip Dynamic APUF

For non-linearity in PUF responses, PUF architecture such as Feed Forward and XOR PUF was proposed at the cost of area and resource. We suggest a novel PUF architecture titled the 64-bit Arbiter-Skip Dynamic PUF (DPUF). We will refer to this architecture as DPUF in the rest of the article. This PUF proposes adding reliable and unpredictable noise to the PUF response, as shown in Figure 14.

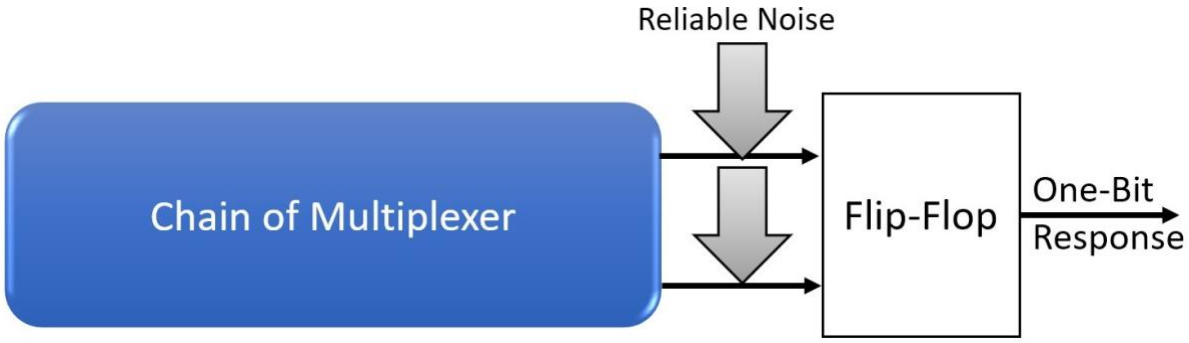

Figure 12. Concept of Proposed Dynamic PUF.

To introduce noise in Arbiter PUF, we have added chain of APUF with fewer multiplexer units, which runs in parallel with the main chain of multiplexers. The noise added due to other chain biases in some of the responses makes it difficult for the modeling attacks to predict the exact correlation between challenge and response bits.

The challenge bits for the parallel skip arbiter are chosen from the original challenge bits. The intermediate response generated from the arbiter-skip PUF is passed to a 1:2 demultiplexer unit (Demux). Depending on the select input of the Demux, the intermediate response is sent to the main flip-flop. The ’select’ input for the Demux is randomly generated from the main chain of 64 MUX using a flip-flop that works as an arbiter. Both the Demux output and the final MUX output are XORed together. The result is biased when the intermediate response is 1, but when it is 0, the PUF functions like a normal arbiter PUF. When the intermediate response bit 1 is generated from the Arbiter skip chain, as shown in Figure 15, the result will be biased depending on the select input of Demux. Assuming the randomness of PUF to generate 50% 1’s and 0’s for both intermediate and main arbiter PUF, then approximately 10% of the final response will be biased, making a complex biased pattern for any modeling attack. The DPUF has less prediction accuracy and is immune to modeling attacks than other complex PUFs like XOR PUF variants and Interpose PUF.

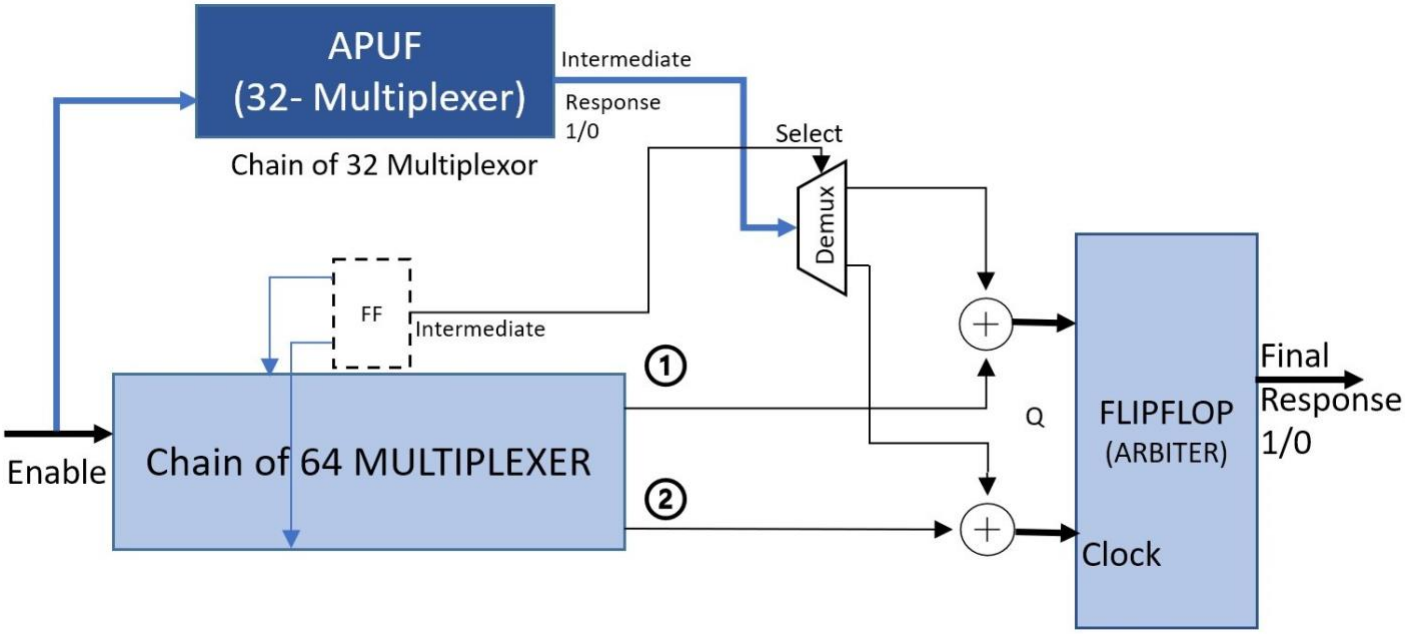

Figure 13. Arbiter Skip DPUF Architecture.

Implementation

To evaluate the metrics, the DPUF and APUF is implemented on FPGA. We have opted for FPGA due to its easy reconfiguration compared to the fabrication of ICs, making it cost-effective. The implementation steps of the PUF function on FPGA are as shown in Figure 16. The first step involves RTL designing of PUF function using HDL and design synthesis of PUF. After running the design synthesis, the PUF structure is ready to program on FPGA. Due to the non-identical path, delay can vary due to wire length differences. To avoid this and ensure the performance of PUF on FPGA, the third step requires floorplanning of design on the FPGA board. After careful floorplanning, implementation is done in next step, and FPGA is programmed using bitstream files. After programming the FPGA, the functionality of PUF must be checked. After a successful functionality check of PUF design, the CRPs are exported for performance evaluation or database generation.

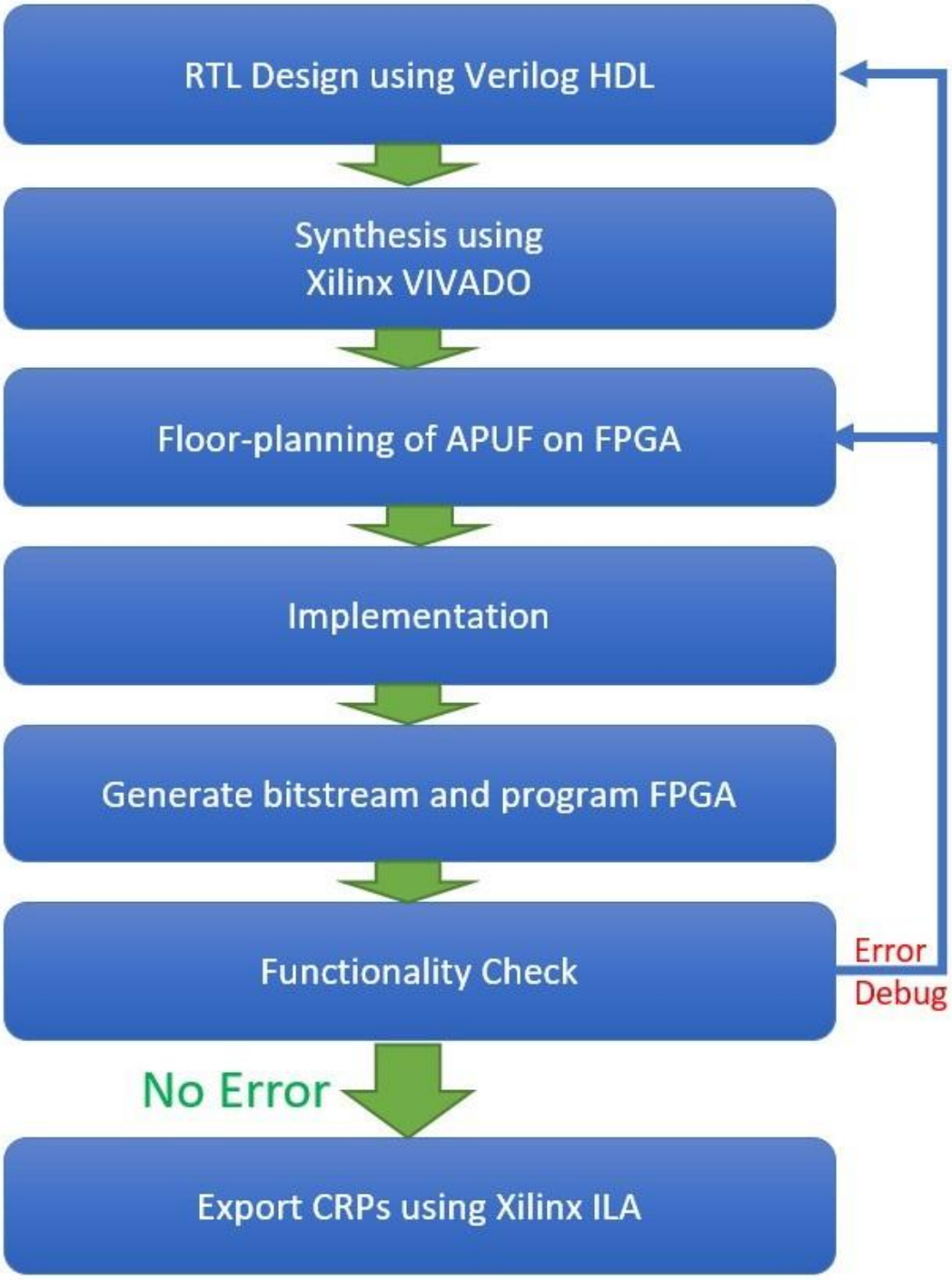

Figure 14. Steps of PUF implementation.

Experimental Setup

The APUF and Arbiter Skip DPUF Architecture were implemented on Xilinx Artix-7 100T board using the Xilinx Vivado tool, and all challenges were captured at room temperature. For performance metrics and training ML models, thousands of CRPs are required. A control unit shown in Figure 17 is designed to generate random challenge bits automatically using Linear Feedback Shift Register

(LFSR) and pass them to APUF. The control unit has three main modules, namely LFSR, PUF, and RAM, which generate a random response bit based on their delay characteristics. Floorplanning is done using the Xilinx Vivado tool to ensure accurate response pairs for the Arbiter PUF. The Vivado tools automatically redesign the synthesis according to behavioral logic. But the PUF functionality cannot be distinguished by behavioral logic as, ideally, the circuit output should remain constant for the identical path. The auto-synthesis tool of Vivado needs to be ensured that it does not change the path design of PUF according to behavioral logic. The ILA tool made debugging the APUF, real-time designing, and recording CRPs easier. For evaluating the resiliency of modeling attacks against PUF, around 1 million CRPs were recorded. The RAM block is used to provide identical challenge bits multiple times to assess the performance of the PUF. Xilinx's Integrated Logic Analyzer (ILA) tool captures around 131K randomly generated responses from PUF and LFSR-generated challenge bits in one round. To record accurate CRPs, ILA must be triggered, and the capture setup must be optimized to record CRPs only when new challenge bits are updated. The waveform for CRPs recorded using ILA is depicted in Figure 18.

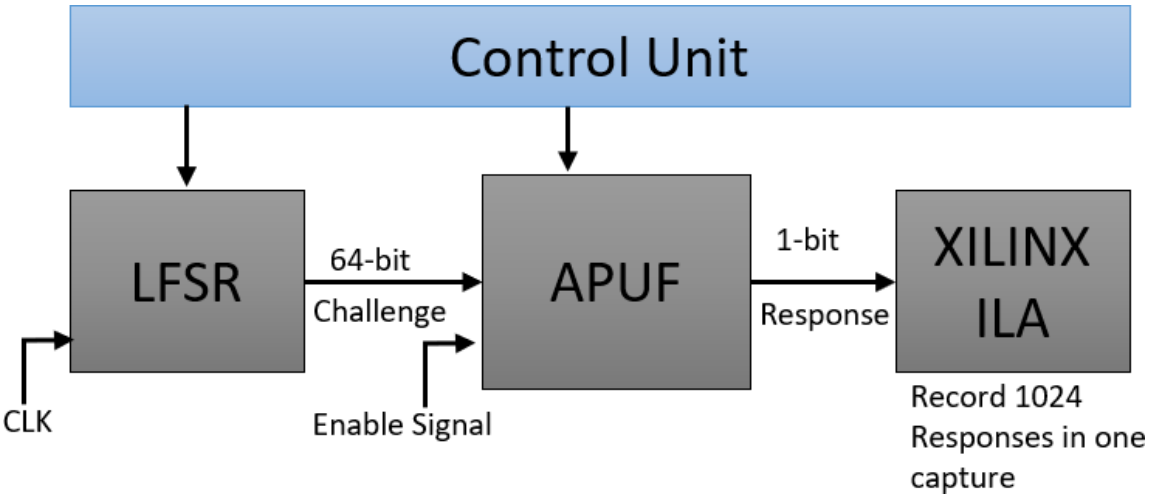

Figure 17. Control unit for generating PUF CRPs.

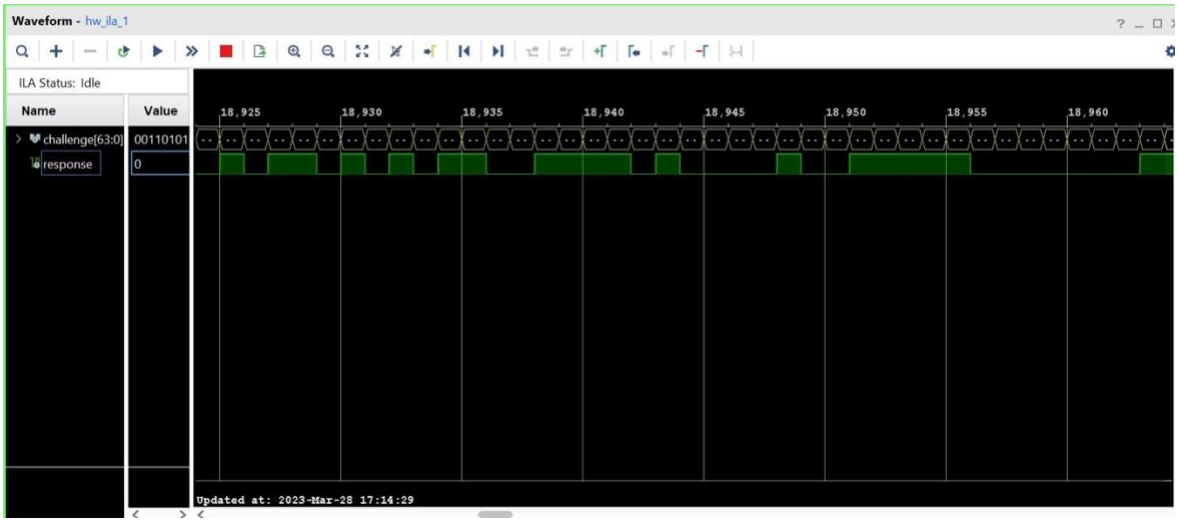

Figure 18. ILA waveform for RPs in Vivado

FPGA Implementation Strategy

The PUF function exhibits non-reliable behavior when implemented on an FPGA. Proper floorplanning must be considered to ensure a reliable PUF implementationon FPGA. Especially in the case of delay based PUF, both multiplexer paths must share identical routing. If both paths are not similar, the connecting wire influencesthe delay. The floorplanning of APUF with two exact routes on the Artix-7 100T board is shown in Figure 19.

Still, as FPGA lacks wire routing configuration, it becomes quite complex to implement an identical path on FPGA. We tried to lower the path difference by using floor planning and reducing the wiring delay difference. The APUF and its variants can generate a higher number of CRPs due to the switching of cross-coupled and parallel paths due to the select line of the multiplexer. The implementation of this multiplexer is depicted in Figure 20 more precisely. As discussed earlier, most of the design tools, including Vivado, auto-optimize HardwareDescriptive Language Code (HDL) according to the behavior of the design. As PUF exhibits manufacturing variation, the behavior model cannot simulate its behavior.To avoid auto-optimization by Vivado DONT_TOUCH attribute of the Vivado toolis used. This attribute allows all the logic from the HDL code without optimization.

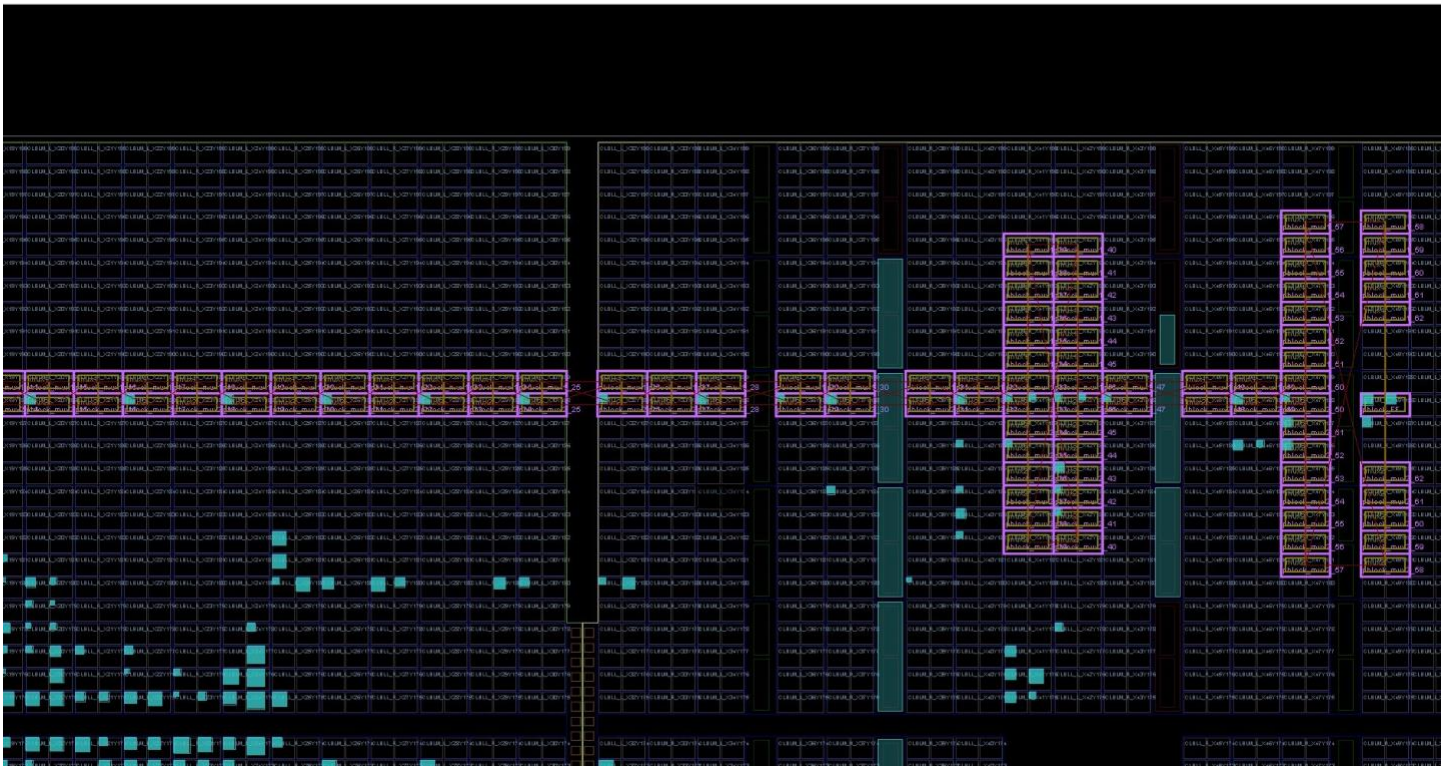

Figure 19. Floorplanning of APUF on Artix-7 FPGA.

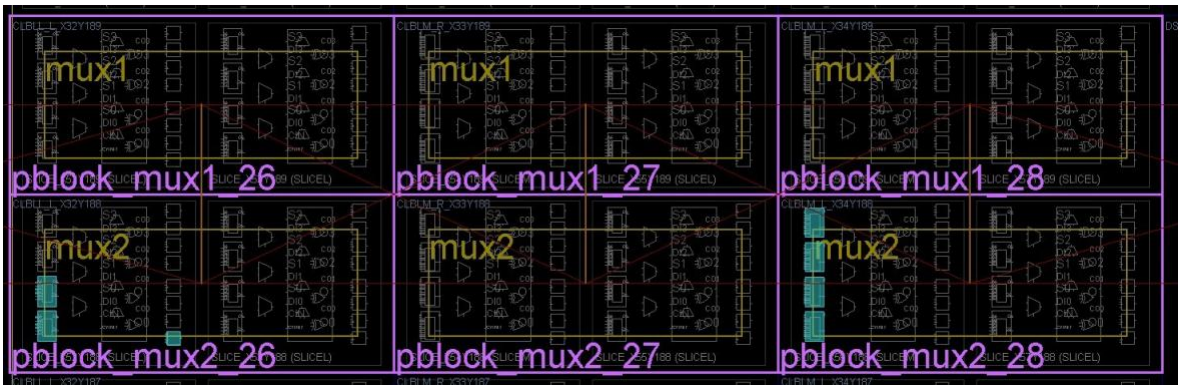

Figure 20. Implementation of Multiplexer Switch on Artix-7 FPGA.

Attack Model

The resiliency of the proposed dynamic PUF was analyzed for Logistic Regression and MLP-based modeling attacks. Instead of developing the attack model from scratch, we have used the attack model developed earlier on pypuf library [35]. The pypuf is a Python library developed by researchers to enhance the research work in PUF, especially in modeling attacks. This library also offers the CRPs dataset for iPUF, XOR PUF variants, and APUF. We have used the iPUF and XOR PUF variants dataset, while the CRPs for APUF and proposed PUF design were recorded from the

implemented design on FPGA. The attack model by pypuf library is used. The modeling attack using logistic regression developed by Tobisch and Becker while using MLP developed by Wisiol et al. [4], [7]. The attack model in pypuf [35] requires preprocessing the CRPs dataset for the model in bipolar format -1,1 instead of 0,1 for better accuracy. So, we have processed our recorded CRPs and converted into desired format for the model published in pypuf.

Results

As previously discussed, researchers have proposed complex PUF architectures to increase their resistance against modeling attacks. However, this approach often results in higher resource utilization, which can be expensive. For instance, an n-XOR PUF variant necessitates n additional chains of multiplexers, which can consume substantial resources. To tackle this problem, researchers have investigated different techniques to decrease resource utilization while ensuring security. We propose a dynamic architecture by adding reliable noise to the previously proposed APUF architecture. The DPUF results in deviation of generated response, which creates significant resistance against modeling attack. The experiment involved modeling attacks using the pypuf library on n-XOR PUF, APUF, and proposed DPUF, with the one-bit response and 64-bit challenge bits.

The findings revealed that an MLP based attack outperformed an LR attack in predicting response bits. The MLP attack was compared on 10k, 100k, and 1 million CRPs of 4-XOR, 5-XOR, and 6-XOR with DPUF, as depicted in Figure 21. The graph shows that the prediction accuracy for all PUF designs, except 6-XOR PUF, was around 60% for 10k challenges, with 6-XOR PUF having a prediction accuracy of around 50%. When trained on 100k CRPs, the model predicted the 4 & 5-XOR variants with over 95% accuracy, DPUF at around 78%, while 6-XOR remained close to 55%. However, when the MLP model trained with 1 million CRPs, it successfully predicted the response of all mentioned XOR variants with over 95% accuracy. Nonetheless, DPUF’s accuracy remained close to 80%, even with one million CRPs. This demonstrates that DPUF poses a more significant challenge to MLP-based algorithms, making it more difficult to predict response bits than XOR PUFs.

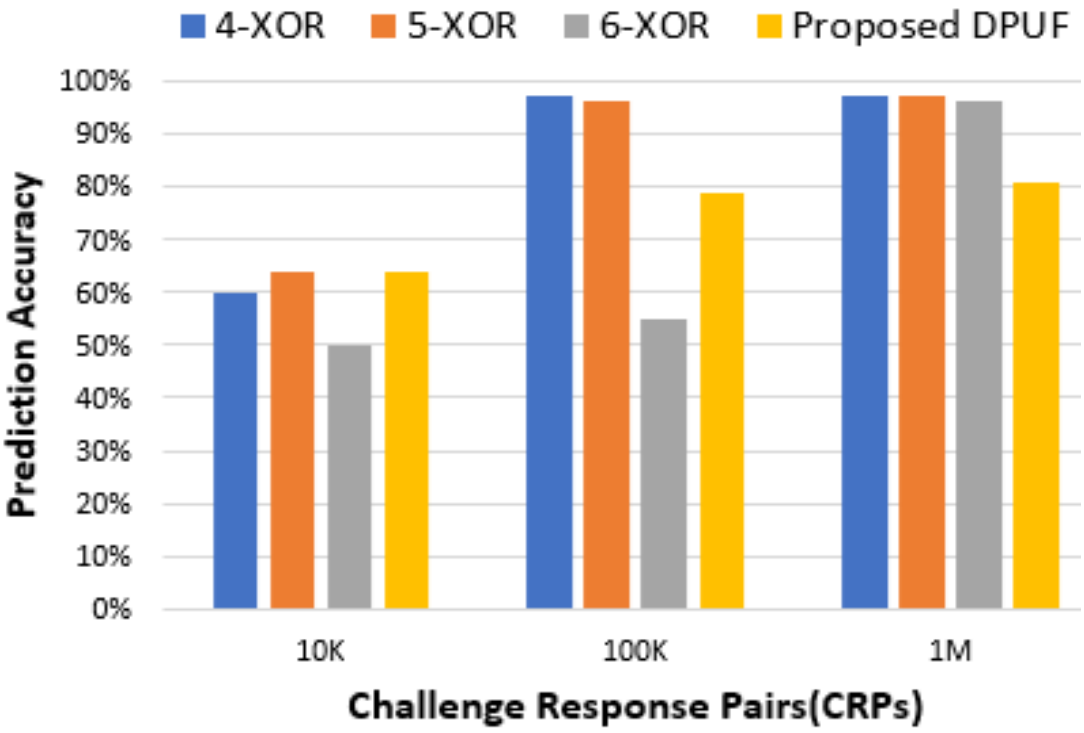

Figure 21. Prediction Accuracy of XOR and proposed PUF using MLP attack

In addition to the MLP-based attack, we assessed the resiliency of DPUF against LR attacks, considered one of the most successful attacks on PUF functions. The results of the LR-based modeling attack on DPUF and APUF are shown in Table 2. Furthermore, we compared the results of the MLP and LR-based attacks with those of the Arbiter PUF design for 10k, 100k, 500k, and 1 million CRPs. We observed that the maximum accuracy for the LR attack on DPUF was 77.97% for 1 million CRPs, whereas that of APUF was over 98%. In Figure 22, we compare the proposed PUF's resilience with n-XOR PUF against the LR attack. The results reveal that DPUF outperforms 4-XOR PUF in terms of resistance to LR attack. However, 5 and 6 XOR PUFs exhibited greater immunity towards LR attack compared to the proposed dynamic PUF. The analysis revealed that DPUF outperforms APUF in terms of resiliency against both types of modeling attacks, as demonstrated by the comparison results.

Table 2. Prediction Accuracy of proposed DPUF with APUF on TestData for MLP and LR attack

	MLP		LR	
CRPs	APUF	DPUF	APUF	DPUF
10k	72.90%	64.00%	88.00%	51.00%
100k	88.40%	79.99%	94.61%	75.20%
500K	98.28%	81.09%	95.22%	75.02%
1M	98.99%	81.11%	98.36%	77.97%

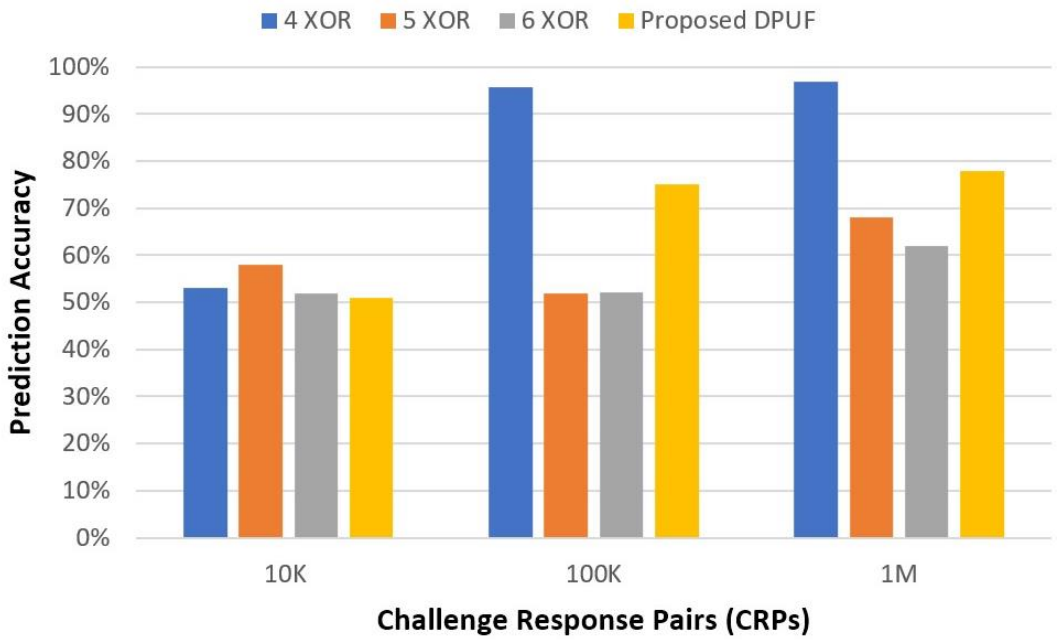

Figure 22. Prediction Accuracy of XOR and proposed PUF using LR attack

In addition to the attacks, we evaluated the desired performance metrics of the proposed DPUF, including randomness, reliability, and uniqueness. Our analysis revealed that the dynamic structure of DPUF is highly reliable, with a reliability of 98%. Moreover, the distribution of 1s and 0s in the response bit is almost uniform, indicating a randomness of 46.81%, as shown in Table 3. We found that the proposed DPUF has a relatively low uniqueness property, as the uniqueness was measured at 7.03%. However, it is worth noting that our analysis was based on a comparison of the PUF's response on only two Artix-7 boards. Evaluating a PUF's reliability and uniqueness requires significant resources. For instance, to measure the reliability at various temperatures, a temperature control unit is necessary. Similarly, assessing the uniqueness property of DPUF would necessitate

testing the PUF on multiple devices rather than just Artix-7 FPGA devices. The PUF was evaluated for all these three metrics on 5000 CRPs. LFSR cannot be used to assess PUF for these metrics as it requires the same challenge bits to compare the generated response bits. We used FPGA ram to provide the same challenge bits for the same PUF to different chips or for other environmental conditions to the same chip. The uniqueness for DPUF was recorded at 7.03%, while the ideal value for uniqueness is 50%. Research by Hori et al. indicates that Artix-7 Boards possess low uniqueness properties for delay-based PUF. The Uniqueness of 9.42 % was recorded by Gu, Hanley, and O'Neill for conventional APUF [36], [37]. Several researchers have reported that the uniqueness property in FPGA yields poor results [21], [38]. The primary reason for this limitation in the uniqueness property of FPGA might be due to the lack of routing inside the slices of FPGA. As the delay difference caused by routing is much greater than that of multiplexers manufacturing variations the, it is challenging to implement ideal PUF on FPGA. Also, it should be noted that we have calculated the uniqueness, reliability, and uniformity of the proposed PUF for one-bit response between two devices. To evaluate these PUF metrics more accurately, the metrics must be evaluated for multiple-bit response between more than two devices.

Table 3. Quality Metrics of Dynamic PUF

Metrics	DPUF
Randomness/Uniformity	46.81 %
Reliability	98%
Uniqueness	7.03%

The PUF is proposed as the technique with low resource and area utilization for cryptography in IoT or FPGA devices. But to increase the resiliency against modeling attacks many complex architectures were proposed that utilize higher resources comparatively to APUF. We have implemented some complex PUF architecture such as XOR PUF and iPUF on Artix-7 FPGA board along-with DPUF to compare the resource utilization. The resource utilization with number of Look Up

Table (LUTs) utilized by different PUFs on FPGA is recorded in Table 4. The arbiter skip DPUF is compared with previously proposed complex PUF architectures in terms or resource utilization in Figure 23 in reference to APUF. The DPUF utilizes around 27% more resources compared to the APUF, while n-XOR PUF variants utilize over 100% to 300% more resources. As iPUF is derived from XOR PUF, it utilizes comparatively higher resources than all the proposed PUF architecture. As depicted in Figure 23, the (3,3)-iPUF utilizes over 510% resource than the basic APUF.

Table 2. Resource utilization of Dynamic PUF, XOR PUF and IPUF on FPGA

	LUTs	SLICE	SLICE REG
DPUF	162	55	3
APUF	127	49	1
2 XOR	257	69	2
3 XOR	385	114	3
4 XOR	513	136	4
(2,2) IPUF	518	141	4
(2,3) IPUF	648	175	5
(3,3) IPUF	776	205	6

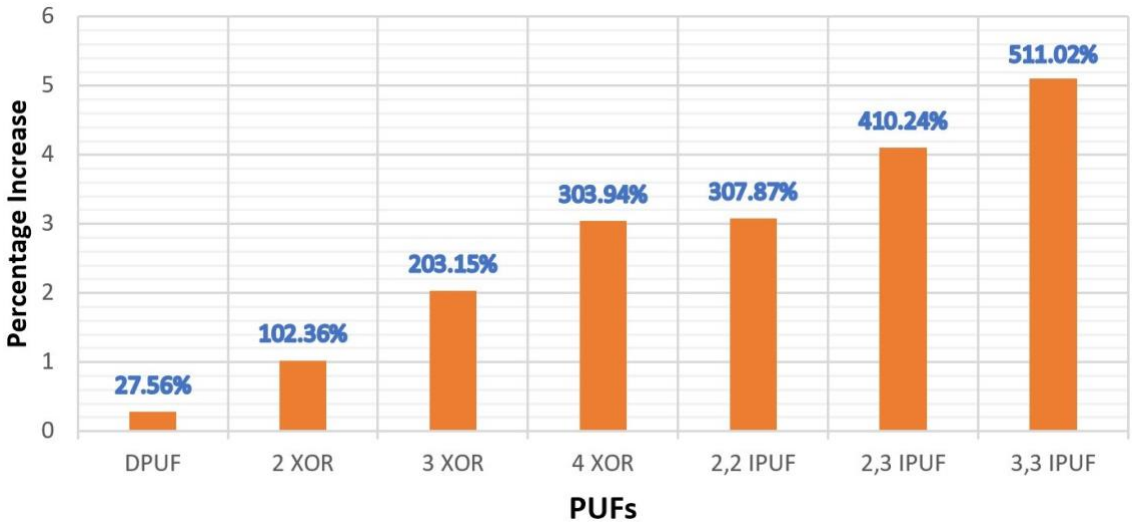

Figure 23. Comparison of Resource Utilization increase with APUF

CHAPTER 5

CONCLUSION AND DISCUSSION

Conclusion

The PUF architecture was proposed as a low-cost alternative to traditional cryptography methods such as AES to address issues like SCA, cloning, and other attacks on IoT devices. However, the high correlation between CRP generated by PUF led to the proposal of a modeling attack using machine learning, which compromises the security of IoT devices. Despite being designed to thwart machine learning algorithms, complex architectures such as iPUF and XOR PUF have the drawbacks ofhigher resource usage and susceptibility to modeling attacks, despite their complexity. This study has discussed previously proposed modeling attacks on PUF designswith higher accuracy. We implemented a 2-D CNN model to evaluate its effectiveness in predicting response bits, but MLP and LR attacks proved to be more efficientthan they were, requiring additional training time.

Moreover, we have discussed the PUF and its resiliency against modeling attacks. Among the demonstrated attacks, the MLP attack outperformed LR based modeling attack in terms of training time and prediction accuracy for all PUF designs. We have introduced the novel concept of adding reliable noise in delay-based Arbiter PUF for resistance against modeling attacks with low resource utilization. The accuracy of predicting one-bit response using MLP or LR attacks was around 96% on the previously proposed complex PUF architectures. However, the accuracy results of DPUF show that attack accuracy has decreased to 81.1% for MLP attacks and 77.97% for LR attacks on DPUF. The proposed Arbiter-skip PUF increased the complexity of PUF with extremely low resource utilization compared to the earlier proposed complex PUF architectures. The resource utilization is reduced by 87% compared to (3,3)-iPUF.

Overall, the proposed dynamic PUF showed better results for vulnerability against modeling attacks with extremely low resource utilization. This method enables the implementation of the PUF

architecture on small IoT devices where IC size is a significant constraint while also addressing security concerns.

Discussion

However, the proposed architecture needs to be further evaluated for basic quality metrics of PUF for multiple devices under different temperature conditions. In particular, the uniqueness property needs to be assessed for other FPGA devices and ASIC fabrication. Furthermore, adding reliable noise can be tried by adding different PUF architectures instead of using basic APUF to increase complexity and reduce resource utilization. Further research on this architecture and implementation on multiple FPGA and IC fabrications will provide a clear understanding of resiliency towards modeling attack and quality metrics of PUF. We strongly believe that our study opens further research in dynamic PUF.

The third promising concept would be that this PUF architecture can be used as a building block of previously proposed complex architecture such as XOR and iPUF as shown in Figure 24 and 25. Currently, the iPUF uses the XOR PUF, which requires multiple chains of APUFs, increasing resource utilization. This idea must be evaluated for ML resiliency and PUF basic quality metrics. These promising findings suggest that the proposed dynamic PUF could be a viable solution for enhancing the security of resource-constrained device.

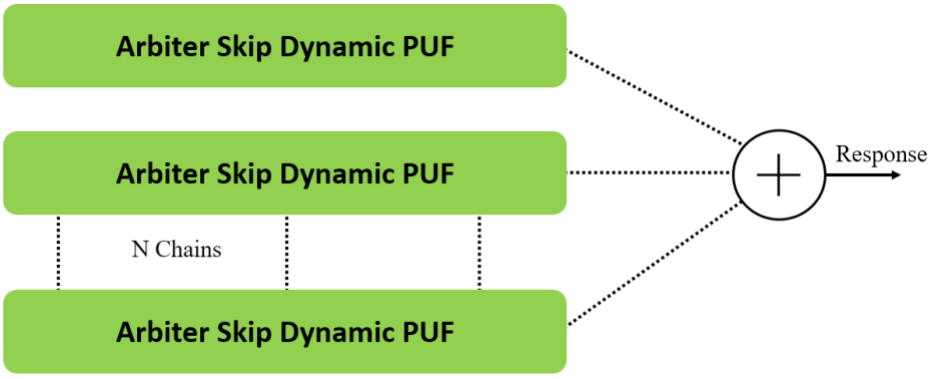

Figure 24. n-XOR PUF using DPUF

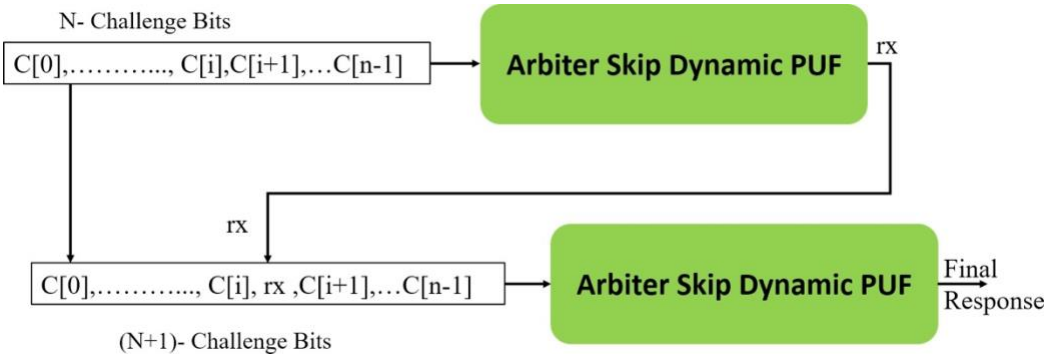

Figure 25. iPUF using DPUF

REFERENCES

[1] R. Anderson and M. Kuhn, “Tamper resistance – A cautionary note”, *2nd USENIX Workshop on Electronic Commerce (EC 96)*. Oakland, CA: USENIX Association, Nov. 1996. URL: https://www.usenix.org/conference/2nd-usenix-workshop-electronic-commerce/tamper-resistance-cautionary-note.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. 2002. “Silicon physical random functions”, *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, Nov. 2002, pp. 148-160. doi: 10.1145/586110.586132.

[3] B. Halak, M. Zwolinski and M. S. Mispan, "Overview of PUF-based hardware security solutions for the internet of things," *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Abu Dhabi, United Arab Emirates, 2016, pp. 1-4, doi: 10.1109/MWSCAS.2016.7870046.

[4] A. Magyari and Y. Chen, "Review of State-of-the-Art FPGA Applications in IoT Networks," *Sensors*, vol. 22, no. 19, pp. 7496, 2022. https://doi.org/10.3390/s22197496.

[5] J. Tobisch and G. T. Becker, "On the Scaling of Machine Learning Attacks on PUFs with Application to Noise Bifurcation," *Radio Frequency Identification (RFIDSec 2015*), S. Mangard and P. Schaumont, Eds., Springer, Cham, 2015, pp. 17-31. doi: 10.1007/978-3-319-24837-0_2.

[6] M. S. Alkatheiri and Y. Zhuang, "Towards fast and accurate machine learning attacks of feed-forward arbiter PUFs," *2017 IEEE Conference on Dependable and Secure Computing*, Taipei, Taiwan, 2017, pp. 181-187, doi: 10.1109/DESEC.2017.8073845.

[7] N. Wisiol, B. Thapaliya, K. T. Mursi, J. -P. Seifert and Y. Zhuang, "Neural Network Modeling Attacks on Arbiter-PUF-Based Designs," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2719-2731, 2022, doi: 10.1109/TIFS.2022.3189533.

[8] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, “Modeling attacks on physical unclonable functions”, *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, 2010, 237–249. https://doi.org/10.1145/1866307.1866335.

[9] Y. Ikezaki, Y. Nozaki and M. Yoshikawa, "Deep learning attack for physical unclonable function," *2016 IEEE 5th Global Conference on Consumer Electronics*, Kyoto, Japan, 2016, pp. 1-2, doi: 10.1109/GCCE.2016.7800478.

[10] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," *2007 44th ACM/IEEE Design Automation Conference*, San Diego, CA, USA, 2007, pp. 9-14.

[11] A. Mutlu, K. J. Le, M. Celik, D. -s. Tsien, G. Shyu and L. -C. Yeh, "An Exploratory Study on Statistical Timing Analysis and Parametric Yield Optimization," *8th International Symposium on Quality Electronic Design (ISQED'07)*, San Jose, CA, USA, 2007, pp. 677-684, doi: 10.1109/ISQED.2007.34.

[12] A. Babaei and G. Schiele, “Physical Unclonable Functions in the Internet of Things: State of the Art and Open Challenges,” *Sensors*, vol. 19, no. 14, p. 3208, Jul. 2019, doi: 10.3390/s19143208.

[13] N. N. Anandakumar, M. S. Hashmi, and M. Tehranipoor, "FPGA-based physical unclonable functions: A comprehensive overview of theory and architectures," Integration, vol. 81, pp. 175-194, 2021. doi: 10.1016/j.vlsi.2021.06.001.

[14] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A new arbiter PUF for enhancing unpredictability on FPGA," The Scientific World Journal, vol. 2015, article ID 864812, 2015. doi: 10.1155/2015/864812.

[15] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," *2004 Symposium on VLSI Circuits. Digest of Technical Papers* (IEEE Cat. No.04CH37525), Honolulu, HI, USA, 2004, pp. 176-179, doi: 10.1109/VLSIC.2004.1346548.

[16] F. Wilde, B. M. Gammel and M. Pehl, "Spatial correlation analysis on physical unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1468-1480, June 2018, doi: 10.1109/TIFS.2018.2791341.

[17] J. Shao, “Characterization of FPGA-based Arbiter Physical Unclonable Functions”, Dissertation, School of Electrical Engineering and Computer Science (EECS), KTH, Sweden, 2019.

[18] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," *2004 Symposium on VLSI Circuits. Digest of Technical Papers* (IEEE Cat. No.04CH37525), Honolulu, HI, USA, 2004, pp. 176-179, doi: 10.1109/VLSIC.2004.1346548.

[19] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose PUF: Secure PUF design against state-of-the-art machine learning attacks," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2019, no. 4, pp. 243-290, 2019. doi: 10.13154/tches.v2019.i4.243-290.

[20] S. Khalfaoui, J. Leneutre, A. Villard, I. Gazeau, J. Ma, and P. Urien, "Security Analysis of Machine Learning-Based PUF Enrollment Protocols: A Review," Sensors (Basel, Switzerland), vol. 21, no. 24, pp. 8415, 2021. doi: 10.3390/s21248415.

[21] A. Maiti, V. Gunreddy, and P. Schaumont, "A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions," in Embedded Systems Design with FPGAs, pp. 245-267, 2012. doi: 10.1007/978-1-4614-1362-2_11.

[22] S. Tajik, E. Dietz, S. Frohmann, H. Dittrich, D. Nedospasov, C. Helfmeier, J.-P. Seifert, C. Boit, and H.-W. Hübers, "Photonic side-channel analysis of arbiter PUFs," in Journal of Cryptology, vol. 30, no. 2, pp. 550-571, Apr. 2017. doi: 10.1007/s00145-016-9228-6.

[23] S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl and A. -R. Sadeghi, "Remanence Decay Side-Channel: The PUF Case," in *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1106-1116, June 2016, doi: 10.1109/TIFS.2015.2512534.

[24] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, “Side-channel analysis of PUFs and fuzzy extractors”, *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST'11)*, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 33–47.

[25] A. Aghaie and A. Moradi, "TI-PUF: Toward Side-Channel Resistant Physical Unclonable Functions," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3470-3481, 2020, doi: 10.1109/TIFS.2020.2986887.

[26] R. A. RadhaKrishnan, "Side-channel resistant implementation using arbiter PUF," Cryptology ePrint Archive, Paper 2023/047. https://eprint.iacr.org/2023/047.

[27] K. T. Mursi, B. Thapaliya, Y. Zhuang, A. O. Aseeri, and M. S. Alkatheiri, "A Fast Deep Learning Method for Security Vulnerability Study of XOR PUFs," Electronics, vol. 9, no. 10, pp. 1715, MDPI AG, 2020. http://dx.doi.org/10.3390/electronics9101715.

[28] P. Santikellur, A. Bhattacharyay, and R. S. Chakraborty, "Deep Learning based Model Building Attacks on Arbiter PUF Compositions," IACR Cryptology ePrint Archive, vol. 2019, p. 566, 2019.

[29] N. Wisiol, C. Mühl, N. Pirnay, P. H. Nguyen, M. Margraf, J.-P. Seifert, M. van Dijk, and U. Rührmair, "Splitting the interpose PUF: A novel modeling attack strategy," IACR Transactions on Cryptographic Hardware and Embedded Systems, vol. 2020, no. 3, pp. 97-120, 2020. doi: 10.13154/tches.v2020.i3.97-120.

[30] A. O. Aseeri, Y. Zhuang and M. S. Alkatheiri, "A machine learning-based security vulnerability study on XOR PUFs for resource-constraint Internet of Things," *2018 IEEE International Congress on Internet of Things (ICIOT)*, San Francisco, CA, USA, 2018, pp. 49-56, doi: 10.1109/ICIOT.2018.00014.

[31] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[32] W. Xiong, A. Schaller, S. Katzenbeisser, and J. Szefer, "Dynamic physically unclonable functions," *Proceedings of the 2019 on Great Lakes Symposium on VLSI (GLSVLSI '19)*, New York, NY, USA, 2019, pp. 311-314. doi: 10.1145/3299874.3318025.

[33] W. Xiong, A. Schaller, S. Katzenbeisser and J. Szefer, "Software protection using dynamic PUFs," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2053-2068, 2020, doi: 10.1109/TIFS.2019.2955788.

[34] Y. Wang, C. Wang, C. Gu, Y. Cui, M. O'Neill and W. Liu, "A dynamically configurable PUF and dynamic matching authentication protocol," *IEEE Transactions on Emerging Topics in Computin*g, vol. 10, no. 2, pp. 1091-1104, 1 April-June 2022, doi: 10.1109/TETC.2021.3072421.

[35] N. Wisiol, C. Gräbnitz, C. Mühl, B. Zengin, T. Soroceanu, N. Pirnay, K. T. Mursi, and A. Baliuka, "pypuf: Cryptanalysis of physically unclonable functions," version 2, August 2021, Zenodo. https://doi.org/10.5281/zenodo.3901410.

[36] C. Gu, N. Hanley and M. O'Neill, "FPGA-based strong PUF with increased uniqueness and entropy properties," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, USA, 2017, pp. 1-4, doi: 10.1109/ISCAS.2017.8050838.

[37] Y. Hori, H. Kang, T. Katashita, A. Satoh, S. Kawamura, and K. Kobara, "Evaluation of Physical Unclonable Functions for 28-nm Process Field-Programmable Gate Arrays," Journal of Information Processing, vol. 22, no. 2, pp. 344-356, 2014. doi: 10.2197/ipsjjip.22.344.

[38] Y. Hori, T. Yoshida, T. Katashita and A. Satoh, "Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs," *2010 International Conference on Reconfigurable Computing and FPGAs*, Cancun, Mexico, 2010, pp. 298-303, doi: 10.1109/ReConFig.2010.24.