How do the Arbiter PUFs Sample the Boolean Function Class?

Animesh Roy[1], Dibyendu Roy[2], and Subhamoy Maitra[1]

[1] Indian Statistical Institute, Kolkata, India
[2] Indian Institute of Information Technology Vadodara (Gandhinagar Campus), India
animesh.roy03@gmail.com, dibyendu.roy@iiitvadodara.ac.in,
subho@isical.ac.in

Abstract. Arbiter based Physical Unclonable Function (sometimes called Physically Unclonable Function, or in short PUF) is a hardware based pseudorandom bit generator. The pseudorandomness in the output bits depends on device specific parameters. For example, based on the delay parameters, an n-length Arbiter PUF can be considered as an n-variable Boolean function. We note that the random variation of the delay parameters cannot exhaust all the Boolean functions and the class is significantly smaller as well as restricted. While this is expected (as the autocorrelation property in certain cases is quite biased), we present a more disciplined and first theoretical combinatorial study in this domain. Our work shows how one can explore the functions achieved through an Arbiter based PUF construction with random delay parameters. Our technique mostly shows limitation of such functions from the angle of cryptographic evaluation as the subclass of the Boolean function can be identified with much better efficiency (much less complexity) than random. On the other hand, we note that under certain constrains on the weights of inputs, such a simple model of Arbiter PUFs provide good cryptographic parameters in terms of differential analysis. In this regard, we theoretically solve the problem of autocorrelation properties in a restricted space of input variables with a fixed weight. Experimental evidences complement our theoretical findings.

Keywords: Bias, Boolean Function, Non-uniformity, Physically Unclonable Function (PUF), Pseudorandomness, Restricted Domain.

1 Introduction

Arbiter based Physically Unclonable Functions (PUFs) were first introduced in [8]. This is a hardware based pseudorandom bit generator which is used to generate cryptographic keys and related applications in device authentications [5,9,10]. PUFs are used to generate keys during the execution of the algorithms without storing them in an insecure memory. To meet the security needs, these constructions must be one-way and should not be cloned in different devices. The design of PUFs basically depends on multiple device parameters. Due to this, such devices supposedly generate uncorrelated output bit-stream. An

n-length Arbiter PUF takes an n-bit long challenge and based on the manufacturing variations, it generates one pseudorandom output bit. Thus an n-length Arbiter based PUF can be treated as a Boolean function from $\{0,1\}^n$ to $\{0,1\}$, as described in [12]. Due to the pseudorandom nature of the output bits, one can exploit them for security related tools and thus PUF has certain practical applications, e.g., smart cards [1]. An Arbiter based PUF supports a large amount of Challenge-Response Pairs (CRPs) and therefore an adversary should not be able to predict the CRPs.

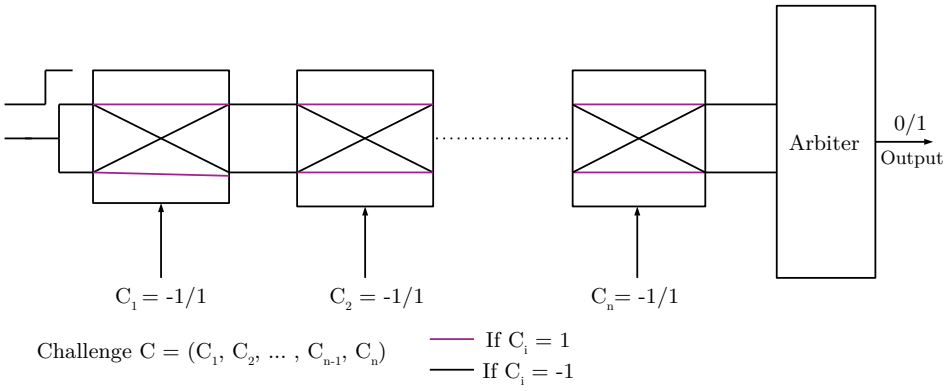


Fig. 1: Basic structure of an Arbiter based PUF.

An ideal PUF should exhibit some important cryptographic features, like uniformity, uniqueness, reliability, etc. Uniformity describes the distribution of the output bits of a PUF. If a PUF produces an equal number of 0's and 1's in the output then that PUF is said to be uniform or balanced. The uniqueness property says that if we provide the same input to different PUFs, then the output of one PUF should not be predicted from the other. That means each PUF should be unique in nature. It will have good reliability if the same device produces the same output for the same input in different instances. In real life, achieving all these cryptographic properties together is difficult as the device specific and environmental parameters may inject noise in the output bits. Thus the reducing noise in the output bits becomes an important task in practice. In [7], Gassend has shown that error correcting codes can be used to tackle the noisy situations.

As we have already described, an Arbiter based PUF can act as an n-variables Boolean function. In most of the cryptographic applications, the input bits of a Boolean function are usually considered independent and taken uniformly from the domain. In such a scenario, the question is on pseudo-randomness measures of the output bits. If this is violated, then the PUF model should not be

accepted for cryptographic applications. There are several attacks in this direction [1,2,18,19,20,21,22] and the recent trend shows significant works in this direction using Machine Learning tools [11,16,20,22]. Several counter-measures are also proposed to resist such attacks and thus new designs are introduced [6,10,11]. On an orthogonal context, we are more interested in combinatorial and statistical aspects in evaluating the Boolean functions generated out of varying delays in Arbiter PUFs. In this direction, we refer to [24], where several non-randomness results had been demonstrated theoretically.

From [24], it can be referred that if one generates output bits corresponding to two challenge inputs $C = (C_1, C_2, \ldots, C_n)$ and $\tilde{C} = (\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_n)$, where C and $\tilde{C}$ belong to $\{-1, 1\}^n$ and differ only at the most significant bit (MSB) position (i.e., $C_1 + \tilde{C}_1 = 0$), then the output bits will match with high probability. The position of the differed challenge bit plays an important role in producing the bias. One can look into Figure 1 to understand the position of the challenge bits. This bias reduces with the location of the bit difference at the inputs. The least bias occurs for the middle-most bit. Naturally, this lack of randomness provides a direction that the PUF devices can only produce a restricted class of Boolean functions, not all. Consequently, the immediate scientific question is to explore the set of Boolean functions such Arbiter PUFs are generating. In this regard, here we present relevant combinatorial results to show certain necessary conditions regarding the existence or non-existence of Boolean functions generated out of the Arbiter PUFs. Then we try to find out for what kinds of combinatorial properties the functions from Arbiter PUFs resemble a randomly chosen Boolean function better. We note that if one considers a certain autocorrelation measure after restricting the input bit pattern to a fixed weight, then such bias disappears. Thus, if one can restrict the attack model with such a constraint, then the use of Arbiter PUFs in certain applications (such as lightweight environment) might be recommended.

As a passing remark, we should also mention the thin connectivity with certain stream ciphers like FLIP [13], which are used as integral components in Fully Homomorphic Encryption (FHE) [3]. In this direction, several properties of Boolean functions over restricted domain (definition of the restricted domain is described in Section 1.2 in more detail) were studied in [4,14,15,17]. Our results show that while there is significant bias in the Arbiter PUFs in certain autocorrelation measures [24], this is absent if challenge inputs are chosen from a restricted domain.

Before proceeding further, let us now present the outline of the paper.

1.1 Contribution and Organization

In Section 1.2, we discuss the basic definitions and notations, introducing the existing results and the problems we consider. The contributions of this paper are the followings, in one case it shows the limitation of Arbiter PUFs, and in another case it demonstrates still how they can be useful in restricted domain.

- In Section 2, we study the limitation of Arbiter PUFs in representing the class of Boolean functions. We provide examples of functions that can or

cannot be generated through different delay parameters. An upper bound on the number of such functions are also provided, which shows that the proportions of different functions will be vanishing compared to the total class of Boolean functions as the number of input variables increases. We show that the ratio of distinct Boolean functions arising out of n-length Arbiter PUFs and the total number of n-variable Boolean functions is less than $\frac{1}{2^{5\cdot 2^{n-4}}}$ for $n \geq 4$. The analysis also identifies the nature of the functions arriving out of the Arbiter PUFs with better efficiency.

– Then, in Section 3, we show that the nature of autocorrelation distributions of Arbiter PUFs and Boolean functions do not differ much if the inputs are chosen from a restricted domain. In particular, we consider when the weight of the inputs are fixed and the inputs must always differ at an already selected bit. We provide a theoretical proof in this regard. This shows that in certain restricted applications, such simple Arbiter PUFs can still be useful.

Section 4 concludes the paper.

1.2 Preliminaries

In this section we talk about some basic terminologies and definitions.

Arbiter PUF. It is a hardware based pseudorandom bit generator model, where the basic idea is to initiate a digital race condition on two paths on the chip and decide which of the two paths won the race. In Arbiter PUF construction, there are n-many Arbiter switches present, one after the other, as shown in Figure 1. Each switch has two multiplexers symmetrically placed. Each input bit is fed to each Arbiter switch. A common pulse is also transmitted through the switches and received at the end by an Arbiter. Based on the input bits, the pulse selects the path inside the Arbiter switches. If an input bit is 1, the path of the pulse remains unchanged. Else it gets swapped. Due to process variations, the pulse will traverse through one path, faster than the other. At the end, the Arbiter finally produces the response 0 or 1 based on the top or bottom path is reached first. For each device, these paths for a given challenge act differently due to delay parameters and hence the output will differ for different devices. This is an informal description of the device. For our purpose, we need to follow the mathematical definition more formally. This is as follows.

An n-length (or n-variable) Arbiter based PUF takes an input of length n from $\{-1, 1\}^n$ and generates either 0 or 1. Note that, by abuse of notation, we interchangeably consider the mapping $a \to (-1)^a$, for $a \in \{0, 1\}$ sometime in Boolean treatment here. The input to the Arbiter PUF is known as challenge and the output is known as response. In [11] it has been shown that an n-length Arbiter PUF can be modelled mathematically in the following form.

$$\Delta(C) = \alpha_1 P_0 + (\alpha_2 + \beta_1)P_1 + \cdots + (\alpha_n + \beta_{n-1})P_{n-1} + P_n\beta_n. \tag{1}$$

Here C is the challenge to the PUF, $C = (C_1, \ldots, C_n) \in \{-1, 1\}^n$, α_i and β_i depend on the delay parameters p_i, q_i, r_i, s_i. Usually in a mathematical model

of PUF, we assume that these delay parameters follows normal distribution with mean μ and standard deviation σ, i.e., the distribution follows $\mathcal{N}(\mu, \sigma)$. The formula through which these α_i, β_i are connected with p_i, q_i, r_i, s_i are $\alpha_i = \frac{p_i - q_i}{2} + \frac{r_i - s_i}{2}$, and $\beta_i = \frac{p_i - q_i}{2} - \frac{r_i - s_i}{2}$. It can be easily verified that if $p_i, q_i, r_i, s_i \sim \mathcal{N}(\mu, \sigma)$ then $\alpha_i, \beta_i \sim \mathcal{N}(0, \sqrt{2}\sigma)$. The term $P_k = \prod_{i=k+1}^{n} C_i$, for $k = 0, \ldots, n-1$ and $P_n = 1$. For a challenge $C \in \{-1, 1\}^n$, the value of $\Delta(C)$ can either be positive or negative. If the sign of $\Delta(C)$ is positive, the output from the PUF will be 0 and if the sign of $\Delta(C)$ is negative then the output from the PUF will be 1. We will be using the notation $\mathcal{B}_n^{\mathsf{PUF}}$ to denote the set of n-variable Boolean functions exhaustively generated through n-step Arbiter PUFs, whereas the set of all Boolean functions involving n-variables are usually denoted by $\mathcal{B}_n$. One can note that implementation of an n-variable Boolean function requires exponential number of gates. In practical life, we always prefer to have those circuits which can be implemented using a polynomial number of gates. Arbiter based PUFs are those class circuits that can be implemented using $\mathcal{O}(n)$ units. Thus the Boolean functions constructed using PUF are of great interest. In this paper, we are first time finding such class of Boolean functions and also showing that in a special case it exhibits good property.

Restricted Domain. Let f be a function from $\{-1, 1\}^n$ to $\{0, 1\}$. Further, let the function be defined over a restricted domain when it takes input from a subset of $\{-1, 1\}^n$. We know that the weight of $\mathbf{x} \in \{0, 1\}^n$ (i.e., $wt(\mathbf{x})$) is considered as the number of 1's present in $\mathbf{x}$. In the similar convention along with the transformation $a \to (-1)^a$ here we define $wt(\mathbf{x})$ for $\mathbf{x} \in \{-1, 1\}^n$. The weight of $\mathbf{x} \in \{-1, 1\}^n$ is the total number of -1's present in $\mathbf{x}$. This is the total number of 1's if we consider the string of 0's and 1's. The set $E_{n,k}$ denotes the set of all n-length points whose weight is k, i.e., $E_{n,k} = \{\mathbf{x} : \mathbf{x} \in \{-1, 1\}^n \text{ and } wt(\mathbf{x}) = k\}$. Here $|E_{n,k}| = \binom{n}{k}$. It can be noticed that $E_{n,k}$ is a restricted domain, where the restriction is that the all the points in $E_{n,k}$ will be of length n and weight k.

Autocorrelation of an n-variable Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ is defined by

$$\mathcal{A}_f(\mathbf{a}) = \sum_{\mathbf{x} \in \{0,1\}^n} (-1)^{f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a})}, \mathbf{a} \in \{0, 1\}^n.$$

It can be noticed this definition of autocorrelation can not be used directly to compute autocorrelation of f in $E_{n,k}$. As if we take any $\mathbf{x} \in E_{n,k}$ and take any $\mathbf{a} \in \{0, 1\}^n$ then $\mathbf{x} \oplus \mathbf{a}$ may not belong to $E_{n,k}$. For an $\mathbf{x} \in E_{n,k}$, we need to select an $\mathbf{a}$ selectively such that $\mathbf{x} \oplus \mathbf{a}$ should also belong to $E_{n,k}$. As we have already pointed out (see the discussion in Section 1.3 below), significant bias could be identified in $\mathcal{A}_f(\mathbf{a})$ when $wt(\mathbf{a}) = 1$. In a similar line, we consider a special case, where a specific input bit will be selected, where the differential will exist. However, the weight of the two inputs should be of the same weight.

Let f be an n-variable Boolean function. Let S_1 and S_2 be two sets defined as $S_1 = \{\mathbf{x} \in E_{n,k} \mid u\text{-th bit of } \mathbf{x} \text{ is } -1\}$, $S_2 = \{\mathbf{x} \in E_{n,k} \mid u\text{-th bit of } \mathbf{x} \text{ is } 1\}$.

Note that $E_{n,k} = S_1 \cup S_2$ and $S_1 \cap S_2 = \phi$. The restricted autocorrelation of f over $E_{n,k}$ is defined as

$$\mathcal{A}_f^{E_{n,k}} = \sum_{\mathbf{x}_\in S_1, \mathbf{x}_2 \in S_2} (-1)^{f(\mathbf{x}_1) \oplus f(\mathbf{x}_2)}.$$

It is evident that $|S_1| = \binom{n-1}{k-1}$ and $|S_2| = \binom{n-1}{k}$. We are not concerned about the bit position u as it will be proved that this expression actually does not depend on u for an n-length Arbiter PUF.

The purpose of defining restricted autocorrelation is to study the autocorrelation spectrum of PUF in a restricted domain, where the simple construction of Arbiter PUF does not provide any bias.

1.3 Motivation of Our Work

Theoretical estimation of autocorrelation of an n-variable PUF over a complete domain $\{-1, 1\}^n$ is discussed in [24]. In the same paper, it has also been shown that the outputs corresponding to inputs are heavily biased when two inputs differ at the first position. It means that the autocorrelation value of $f \in \mathcal{B}_n^{\mathsf{PUF}}$ is not good for certain $\mathbf{a} \in \{0, 1\}^n$.

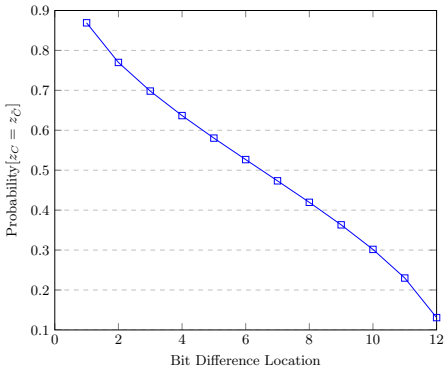


Fig. 2: Representation of Table 1

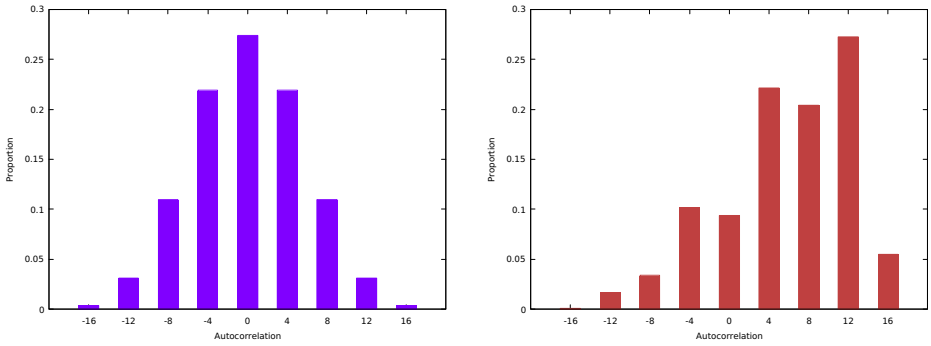
Bit Difference Location	Pr[$z_C = z_{\tilde{C}}$]
1	0.8691
2	0.7699
3	0.6982
4	0.6368
5	0.5804
6	0.5266
7	0.4734
8	0.4196
9	0.3632
10	0.3017
11	0.2300
12	0.1309

Table 1: Experimental Bias of PUFs ($n = 12$) in complete domain (over 1024 randomly chosen Arbiter PUFs) for single bit difference, matching with the theoretical values from [24]

To understand the autocorrelation values we consider a 12-variable PUF and two inputs C and $\tilde{C}$ where C and $\tilde{C}$ differ at only one location. From the result of [24] we know that the output z_C and $z_{\tilde{C}}$ are highly biased for certain

bit difference locations. The experimental $Pr[z_C = z_{\tilde{C}}]$ for different single bit difference locations is provided in Table 1. From Table 1 and Figure 2 it can be observed that the bias is highest when the bit difference location is either first or last and bias is least when the bit difference location is in the middle. Thus for certain values of $\mathbf{a} \in \{0,1\}^n$ the expected autocorrelation value of $f \in \mathcal{B}_n^{\mathsf{PUF}}$ significantly differs from 0.5.

To get a clearer idea about the autocorrelation distribution of PUF we perform statistical analysis. We consider all 4-variable Boolean functions and PUFs and measure the average number of Boolean functions and PUFs corresponding to different possible autocorrelation values $\{-16, -8, -4, 0, 4, 8, 12, 16\}$. From Figure 3 it can be observed that the distribution of PUF differs significantly from the distribution of Boolean function.



(a) Distribution of all Boolean Functions (b) Distribution of all PUFs

Fig. 3: Comparison of Autocorrelation Distribution in Complete Domain

Now we provide a clear answer why the autocorrelation distribution is highly biased for PUF for single bit difference. The basic reason is that the Arbiter PUFs cannot exhaustively generate all possible Boolean functions. This observation motivates us to investigate the following.

- How to estimate the set $\mathcal{B}_n^{\mathsf{PUF}}$?
- Can we obtain a restricted definition of autocorrelation so that the Arbiter PUFs do not expose a significant bias?

2 Relation Between $\mathcal{B}_n^{\mathsf{PUF}}$ and $\mathcal{B}_n$

In this section, we explore the class of Boolean functions generated from n-variable PUFs i.e., $\mathcal{B}_n^{\mathsf{PUF}}$. To compute the number of distinct Boolean functions which can be constructed using PUFs we start with $n = 1$. The total number of Boolean functions involving 1-variable is $|\mathcal{B}_1| = 2^{2^1} = 4$. We all know that a PUF

can be seen as a Boolean function. Thus, the obvious question is if we consider 1-length PUF, can that generate all possible Boolean functions given different delay parameters. To answer this question we state the following proposition.

Proposition 1. *All possible Boolean functions involving* 1*-variable can be generated by using* 1*-length PUFs i.e.,* $\mathcal{B}_1^{\mathsf{PUF}} = \mathcal{B}_1$.

Proof. This proposition can be proven by exhaustively enumerating $\mathcal{B}_1^{\mathsf{PUF}}$. We have considered 1-length PUFs for different random delay parameters and observed that all the possible truth tables are generated in our experiment. Thus $|\mathcal{B}_1^{\mathsf{PUF}}| = |\mathcal{B}_1| = 4$. □

Now we move towards the case for $n = 2$. The total number of Boolean functions in this case is $|\mathcal{B}_2| = 2^{2^2} = 16$. Interestingly, from our experiments, we have observed that 14 many Boolean functions can be constructed from 2-length Arbiter PUFs, i.e., $\mathcal{B}_2^{\mathsf{PUF}} = 14$. Truth tables of two specific Boolean functions can never be constructed using 2-length Arbiter PUFs. In this regard, we will state the following result.

Proposition 2. *The following two Boolean functions* f_1 *and* f_2 *do not belong to* $\mathcal{B}_2^{\mathsf{PUF}}$.

C_1	C_2	f_1	f_2
1	1	1	0
−1	1	0	1
1	−1	1	0
−1	−1	0	1

Proof. The mathematical model of 2-length PUF is $\Delta(C) = \alpha_1 P_0 + (\alpha_2 + \beta_1) P_1 + \beta_2$, where $P_0 = C_1 C_2$ and $P_1 = C_2$. Here α_i, β_i are the delay parameters. We consider the truth table of f_1 first. It can be observed that if the $\text{sign}(\Delta(C))$ and $\text{sign}(C_1)$ are the same then only the truth table f_1 can be generated from 2-length PUF. Thus, to generate the same truth values from a 2-length PUF, we need to have the following scenarios.

C_1	C_2	$\Delta(C)$
1	1	$\alpha_1 + (\alpha_2 + \beta_1) + \beta_2 > 0$
−1	1	$-\alpha_1 + (\alpha_2 + \beta_1) + \beta_2 < 0$
1	−1	$-\alpha_1 - (\alpha_2 + \beta_1) + \beta_2 > 0$
−1	−1	$\alpha_1 - (\alpha_2 + \beta_1) + \beta_2 < 0$

If the above conditions hold for atleast one pair of $\alpha_1, \alpha_2, \beta_1, \beta_2$ then only the truth values of f_1 can be generated. If we add two > 0 inequalities then we will have $\beta_2 > 0$ and if we add two < 0 inequalities then we will have $\beta_2 < 0$. This generates a contradiction. Hence the truth table of f_1 can not be generated from the 2-length Arbiter PUF structure. Similarly, it can be shown that it is not possible to generate the truth table of f_2 using a 2-length PUF. Thus $f_1, f_2 \notin \mathcal{B}_2^{\mathsf{PUF}}$. □

Using the transformation $a \to (-1)^a$ for $a \in \{0, 1\}$, the Algebraic Normal Form (ANF) of f_1, f_2 are $f_1(x_1, x_2) = 1 \oplus x_1$ and $f_2(x_1, x_2) = x_1$ respectively.

From the Figure 1, it can be noticed that x_1 is related to the leftmost block of the Arbiter PUF, furthest from the output.

Proposition 2 justifies that $|\mathcal{B}_2^{\mathsf{PUF}}| = 14$ as we noted from exhaustive experiment and directs us towards the following result.

Lemma 1. *For any n-variable Boolean function $f \notin \mathcal{B}_n^{\mathsf{PUF}}$ if and only if $(1 \oplus f) \notin \mathcal{B}_n^{\mathsf{PUF}}$.*

Proof. To prove this, we assume that there exists an n-variable Boolean function $f \in \mathcal{B}_n^{\mathsf{PUF}}$ but $1 \oplus f \notin \mathcal{B}_n^{\mathsf{PUF}}$. Let the n-length PUF be $\Delta(C) = \alpha_1 P_0 + (\alpha_2 + \beta_1)P_0 + (\alpha_3 + \beta_2)P_2 + \cdots + (\alpha_n + \beta_{n-1})P_{n-1} + \beta_n$. Here, α_i, β_i are the delay parameters. We know that depending on the sign of $\Delta(C)$, the truth table of $1 \oplus f$ is generated. Now if we consider a PUF with the delay parameters $\alpha_i' = -\alpha_i$ and $\beta_i' = -\beta_i$ and construct the PUF $\Delta(C)' = \alpha_1' P_0 + (\alpha_2' + \beta_1')P_0 + (\alpha_3' + \beta_2')P_2 + \cdots + (\alpha_n' + \beta_{n-1}')P_{n-1} + \beta_n'$, then $\text{sign}(\Delta(C))$ and $\text{sign}(\Delta(C)')$ will be opposite for the same challenge values. Thus the truth table generated from $\Delta(C)'$ will be the truth table of $1 \oplus (1 \oplus f) = f$. Which contradicts our assumption. Hence if a Boolean function $f \notin \mathcal{B}_n^{\mathsf{PUF}}$ then $(1 \oplus f) \notin \mathcal{B}_n^{\mathsf{PUF}}$. Similarly if $(1 \oplus f) \notin \mathcal{B}_n^{\mathsf{PUF}}$ then $f \notin \mathcal{B}_n^{\mathsf{PUF}}$. □

We know that any $(n+1)$-variable Boolean function f can be expressed as $f(x_1, \ldots, x_{n+1}) = (1 \oplus x_{n+1}) f_1(x_1, \ldots, x_n) \oplus x_{n+1} f_2(x_1, \ldots, x_n)$, where f_1, f_2 are two Boolean functions involving n variables. This is basically equivalent to $f(x_1, \ldots, x_{n+1}) = f_1(x_1, \ldots, x_n) \parallel f_2(x_1, \ldots, x_n)$, in terms of concatenating the truth tables. That is, the truth table of f can be divided into two halves. In upper half if we consider $x_{n+1} = 0$, then it will contain the truth values of f_1 and in lower half if we consider $x_{n+1} = 1$ then it will contain the truth values of f_2.

Every 3-variable Boolean function f can be written as $f = f_1 \parallel f_2$, where f_1 and f_2 are two Boolean functions involving 2 variables. As the constructions of PUFs depend on parameters from normal distributions, the natural question is that if we consider a 3-variable PUF then can it be of the form $F = f \parallel f_1$ or $F = f_1 \parallel f$, where $f_1 = (1\ 0\ 1\ 0) \notin \mathcal{B}_2^{\mathsf{PUF}}$ (see Proposition 2) and $f \in \mathcal{B}_2$. The mathematical model of 3-variable PUF is $\Delta(C) = \alpha_1 P_0 + (\alpha_2 + \beta_1)P_1 + (\alpha_3 + \beta_2)P_2 + \beta_3$, where $P_0 = C_1C_2C_3$, $P_1 = C_2C_3$ and $P_2 = C_3$. We prepare a truth table of a 3-variable PUF $F = f_1 \parallel f$ where $f_1 = (1\ 0\ 1\ 0) \notin \mathcal{B}_2^{\mathsf{PUF}}$ and $f \in \mathcal{B}_2$. We now break the truth table into two parts. In the upper part $C_3 = 1$ and in lower part $C_3 = -1$. Without loss of generality we consider $f = (0\ 0\ 0\ 0)$. The final truth table of F will be of the following form.

C_1	C_2	C_3	$\Delta(C)$	$F = f_1 \parallel f$
1	1	1	$\alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0$	1
−1	1	1	$-\alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 < 0$	0
1	−1	1	$-\alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0$	1
−1	−1	1	$\alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 < 0$	0
1	1	−1	$-\alpha_1 - (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0
−1	1	−1	$\alpha_1 - (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0
1	−1	−1	$\alpha_1 + (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0
−1	−1	−1	$-\alpha_1 + (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0

We consider the following pairs of equations from the upper part of the above truth table.

$$\begin{cases} -\alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 < 0 \\ \alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 < 0 \end{cases} \tag{2}$$

$$\begin{cases} \alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0 \\ -\alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0 \end{cases} \tag{3}$$

From Equation (2) we get $(\alpha_3 + \beta_2) + \beta_3 < 0$ and from Equation (3) we get $(\alpha_3 + \beta_2) + \beta_3 > 0$, which is a contradiction. Thus for any $f \in \mathcal{B}_3^{\mathsf{PUF}}$ it can not be of the form $f_1 \parallel f_2$ or $f_2 \parallel f_1$ where $f_1 = (1\ 0\ 1\ 0) \notin \mathcal{B}_2^{\mathsf{PUF}}$. Similarly we can prove that for any $f \in \mathcal{B}_3^{\mathsf{PUF}}$ it can not be of the form $(1 \oplus f_1) \parallel f_2$ or $f_2 \parallel (1 \oplus f_1)$ where $f_1 = \notin \mathcal{B}_2^{\mathsf{PUF}}$. In this regard, we present the following important result.

Theorem 1. *If $f_1 \notin \mathcal{B}_n^{\mathsf{PUF}}$, then there does not exist any $F \in \mathcal{B}_{n+1}^{\mathsf{PUF}}$ of the form $f_1 \parallel f$ or $f \parallel f_1$.*

Proof. Assume that there exists an $F \in \mathcal{B}_{n+1}^{\mathsf{PUF}}$ such that $F = f_1 \parallel f$ and $f_1 \notin \mathcal{B}_n^{\mathsf{PUF}}$. Let the challenge input to the $(n+1)$-variable PUF be $C = (C_1, \ldots, C_{n+1})$. The mathematical model of the $(n+1)$-variable PUF corresponding to F is

$$\Delta(C) = \alpha_0 P_0 + (\alpha_1 + \beta_0) P_1 + \cdots + (\alpha_{n+1} + \beta_n) P_n + \beta_{n+1}, \tag{4}$$

where $P_k = \prod_{i=k+1}^{n+1} C_i$. As $F \in \mathcal{B}_{n+1}^{\mathsf{PUF}}$, the inequalities constructed from $\Delta(C)$ in Equation (4) and the truth table corresponding to F will provide a solution for α_i and β_i. Let us look at the truth table of F into two equal parts. In the upper half $C_{n+1} = 1$ and lower half $C_{n+1} = -1$. It can be noticed that the upper half of the truth table of F should be exactly the same as the truth table of f_1 and the lower half should be exactly the same as the truth table of f. Using the values of α_i and β_i we prepare the following model of n-variable PUF

$$\Delta(C)' = \alpha_0' P_0 + (\alpha_1' + \beta_0') P_1 + \cdots + (\alpha_n' + \beta_{n-1}') P_n + \beta_n', \tag{5}$$

with $\alpha_i' = \alpha_i$ for $i = 0, \ldots, n$; $\beta_i' = \beta_i$ for $i = 0, \ldots n-1$ and $\beta_n' = (\alpha_{n+1} + \beta_n) + \beta_{n+1}$. The existence of α_i, β_i guarantees that the PUF described in Equation (5) will be able to generate the truth table of f_1. This is a contradiction as $f_1 \notin \mathcal{B}_n^{\mathsf{PUF}}$. Thus $F = f_1 \parallel f \notin \mathcal{B}_{n+1}^{\mathsf{PUF}}$. Similar arguement works to prove $F = f \parallel f_1 \notin \mathcal{B}_{n+1}^{\mathsf{PUF}}$. □

From Lemma 1 and Theorem 1, it is clear that $\mathcal{B}_n^{\mathsf{PUF}} \subset \mathcal{B}_n$ for $n \geq 2$. In fact we can directly say that if $f \in \mathcal{B}_{n+1}^{\mathsf{PUF}}$ then $f = f_1 \parallel f_2$ where $f_1, f_2 \in \mathcal{B}_n^{\mathsf{PUF}}$. With this we would like to investigate $\mathcal{B}_3^{\mathsf{PUF}}$. Proposition 2 claims that $|\mathcal{B}_2^{\mathsf{PUF}}| = 14$. Now if we prepare a 3-variable Boolean function by concatenating these 14 Boolean functions from $\mathcal{B}_2^{\mathsf{PUF}}$ then we can have maximum 196 Boolean functions. The most natural question is that whether all such Boolean functions belong to $\mathcal{B}_3^{\mathsf{PUF}}$ or not. To answer this, we note the following result.

Proposition 3. *Consider* $f_1 = (1\ 1\ 0\ 1), f_2 = (0\ 1\ 0\ 0) \in \mathcal{B}_2^{\mathsf{PUF}}$ *and* $f = f_1 \parallel f_2$. *The Boolean function* $f \notin \mathcal{B}_3^{\mathsf{PUF}}$.

Proof. We construct a truth table of $f = f_1 \parallel f_2$ for a 3-length PUF, where $f_1 = (1\ 1\ 0\ 1)$, $f_2 = (0\ 1\ 0\ 0) \in \mathcal{B}_2^{\mathsf{PUF}}$.

C_1	C_2	C_3	$\Delta(C)$	$f = f_1 \parallel f_2$
1	1	1	$\alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0$	1
−1	1	1	$-\alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0$	1
1	−1	1	$-\alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 < 0$	0
−1	−1	1	$\alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0$	1
1	1	−1	$-\alpha_1 - (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0
−1	1	−1	$\alpha_1 - (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 > 0$	1
1	−1	−1	$\alpha_1 + (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0
−1	−1	−1	$-\alpha_1 + (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0$	0

First we consider the following pairs of equations from the above truth table.

$$\begin{cases} -\alpha_1 + (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 > 0 \\ \alpha_1 - (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 > 0 \end{cases} \tag{6}$$

$$\begin{cases} -\alpha_1 - (\alpha_2 + \beta_1) + (\alpha_3 + \beta_2) + \beta_3 < 0 \\ \alpha_1 + (\alpha_2 + \beta_1) - (\alpha_3 + \beta_2) + \beta_3 < 0 \end{cases} \tag{7}$$

From Equation (6) we get $\beta_3 > 0$ and from Equation (7) we get $\beta_3 < 0$. This is a contradiction. Thus $f = f_1 \parallel f_2 \notin \mathcal{B}_3^{\mathsf{PUF}}$. □

Proposition 3 shows that even if we take any two Boolean functions f_1, f_2 from $\mathcal{B}_2^{\mathsf{PUF}}$ then $f = f_1 \parallel f_2$ may not belong to $\mathcal{B}_3^{\mathsf{PUF}}$. We have $|\mathcal{B}_1^{\mathsf{PUF}}| = 4$ but $|\mathcal{B}_2^{\mathsf{PUF}}| = 14$. For higher values of n, we have considered the mathematical model of PUF described in Equation (1) for different values of n and exhaustively searched the Boolean functions which belong to $\mathcal{B}_n^{\mathsf{PUF}}$. For $n = 3, 4$ we have observed that $|\mathcal{B}_3^{\mathsf{PUF}}| = 104 < |\mathcal{B}_2^{\mathsf{PUF}}|^2$ and $|\mathcal{B}_4^{\mathsf{PUF}}| = 1882 < |\mathcal{B}_3^{\mathsf{PUF}}|^2$. From this, the following result follows.

Theorem 2. *For any value of* n, $|\mathcal{B}_{n+1}^{\mathsf{PUF}}| \leq |\mathcal{B}_n^{\mathsf{PUF}}|^2$. *Further, for* $n \geq 4$, $\frac{|\mathcal{B}_n^{\mathsf{PUF}}|}{|\mathcal{B}_n|} < \frac{1}{2^{5 \cdot 2^{n-4}}}$.

Proof. The first result follows from Theorem 1. The next result is initiated from exhaustive experiments, where for different values of delay parameters we have observed that $|\mathcal{B}_4^{\mathsf{PUF}}| = 1882$. Regarding the exhaustive experiment supporting the proof we refer to Algorithm 1 below. If we compute $\frac{|B_4^{\mathsf{PUF}}|}{|\mathcal{B}_4|} = \frac{1882}{2^{2^4}} < \frac{1}{2^5} = \frac{1}{2^{5 \cdot 2^{4-4}}}$. Assume that the relation holds for $n = k$, for some $k > 4$, i.e., $\frac{|\mathcal{B}_k^{\mathsf{PUF}}|}{|\mathcal{B}_k|} < \frac{1}{2^{5 \cdot 2^{k-4}}}$. For $n = k + 1$, following Theorem 1 we have,

$$\frac{|\mathcal{B}_{k+1}^{\mathsf{PUF}}|}{|\mathcal{B}_{k+1}|} \leq \frac{|\mathcal{B}_k^{\mathsf{PUF}}|^2}{|\mathcal{B}_k|^2} < \left(\frac{1}{2^{5 \cdot 2^{k-4}}}\right)^2 = \left(\frac{1}{2^{5 \cdot 2^{(k+1)-4}}}\right) \tag{8}$$

Hence for $n \geq 4$, $\frac{|\mathcal{B}_n^{\mathsf{PUF}}|}{|\mathcal{B}_n|} < \frac{1}{2^{5 \cdot 2^{n-4}}}$. □

Although the bound derived in Theorem 2 is not tight, it provides a significant estimation about $\mathcal{B}_n^{\mathsf{PUF}}$. Now the question is how one can obtain $\mathcal{B}_{n+1}^{\mathsf{PUF}}$ exhaustively. One informal way is, consider large number of values varying the delay parameters to construct $(n+1)$ variable PUFs and enumerate the number of distinct ones. However, this cannot be used as a proof.

Below we provide an iterative way of completely enumerating $\mathcal{B}_{n+1}^{\mathsf{PUF}}$ from $\mathcal{B}_n^{\mathsf{PUF}}$. In Algorithm 1 we consider the mathematical model of $(n+1)$-variable PUF, i.e., $\Delta(C) = \alpha_1 P_0 + (\alpha_2 + \beta_1) P_1 + \cdots + (\alpha_{n+1} + \beta_n) P_n + \beta_{n+1}$, $P_i = \prod_{k=i+1}^{n} C_i$. That is $\Delta(C)$ can be considered as a Boolean function on $C = (C_1, C_2, \ldots, C_{n+1})$, the challenge inputs corresponding to $(n+1)$-length PUF. Consider any two $f_1, f_2 \in \mathcal{B}_n^{\mathsf{PUF}}$. Let $f = f_1 \parallel f_2$. For $C_{n+1} = 1$ we prepare the system of inequalities involving α_i, β_i, based on the truth table of f_1. Similarly, for $C_{n+1} = -1$ we construct the system of inequalities involving α_i, β_i based on the truth table of f_2. If this system of equations is solvable then we include the Boolean function f in $\mathcal{B}_{n+1}^{\mathsf{PUF}}$ which corresponds to the $(n+1)$-length PUF $\Delta(C)$. If we continue this process for all $f_1, f_2 \in \mathcal{B}_n^{\mathsf{PUF}}$ then we will have $\mathcal{B}_{n+1}^{\mathsf{PUF}}$.

Algorithm 1: Construction of $\mathcal{B}_{n+1}^{\mathsf{PUF}}$ from $\mathcal{B}_n^{\mathsf{PUF}}$

Input : $\mathcal{B}_n^{\mathsf{PUF}}$
Output: $\mathcal{B}_{n+1}^{\mathsf{PUF}}$

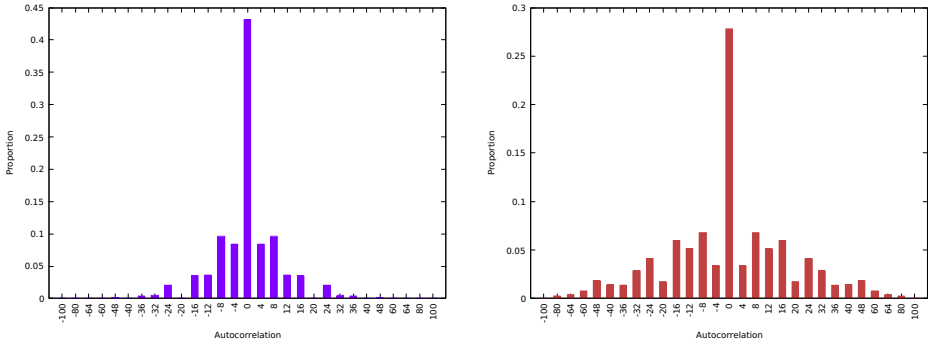
Assign $\Delta(C) = \alpha_1 P_0 + (\alpha_2 + \beta_1) P_1 + \cdots + (\alpha_{n+1} + \beta_n) P_n + \beta_{n+1}$, $P_i = \prod_{k=i+1}^{n} C_i$;
for *each* $f_i \in \mathcal{B}_n^{\mathsf{PUF}}$ **do**
  $F_1 = \{\}$;
  if $C_{n+1} = 1$ **then**
    if $f_i(C_1, \ldots, C_n) = 1$ **then**
      Construct equation $\Delta(C) > 0$ and include $\Delta(C) > 0$ in F_1;
    end
    else
      Construct equation $\Delta(C) < 0$ and include $\Delta(C) < 0$ in F_1;
    end
  end
  for *each* $f_j \in \mathcal{B}_n^{\mathsf{PUF}}$ **do**
    $F_2 = \{\}$;
    if $C_{n+1} = -1$ **then**
      if $f_j(C_1, \ldots, C_n) = 1$ **then**
        Construct equation $\Delta(C) > 0$ and include $\Delta(C) > 0$ in F_2;
      end
      else
        Construct equation $\Delta(C) < 0$ and include $\Delta(C) < 0$ in F_2;
      end
    end
    if $F = F_1 \cup F_2$ *is solvable* **then**
      Construct $f = f_1 \parallel f_2$ and include f in $\mathcal{B}_{n+1}^{\mathsf{PUF}}$;
    end
  end
end
return $\mathcal{B}_{n+1}^{\mathsf{PUF}}$;

We have implemented Algorithm 1 in SageMath 9.2 [23] and enumerated $\mathcal{B}_{n+1}^{\mathsf{PUF}}$ for $n = 1, 2, 3$. Larger values are being tried with further optimization and

several cryptographic properties are being studied. These will be reported in full version of this paper.

3 On Restricted Autocorrelation of Arbiter PUF

In Section 1.3 we have seen that the distribution of Boolean functions and PUFs differs significantly in terms of autocorrelation spectrum. This happens due to the fact that the PUFs depend on multiple device specific parameters and $\mathcal{B}_n^{\mathsf{PUF}} \subset \mathcal{B}_n$ for $n > 2$. Interestingly, if we consider the challenge inputs from $E_{n,k}$ with certain restrictions, then the autocorrelation distributions of random Boolean functions and PUFs become quite close. For measuring this we need to revisit the definition of restricted autocorrelation from Section 1.2.



(a) Distribution of the Boolean functions in the restricted domain $E_{6,3}$

(b) Distribution of the randomly chosen PUFs in the restricted domain $E_{6,3}$

Fig. 4: Comparison of the restricted autocorrelation.

As we have discussed, f is an n-variable Boolean function. S_1 and S_2 are two sets defined as $S_1 = \{\mathbf{x} \in E_{n,k} \mid u\text{-th bit of } \mathbf{x} \text{ is } -1\}$, $S_2 = \{\mathbf{x} \in E_{n,k} \mid u\text{-th bit of } \mathbf{x} \text{ is } 1\}$. Note that $E_{n,k} = S_1 \cup S_2$ and $S_1 \cap S_2 = \phi$. The restricted autocorrelation of f over $E_{n,k}$ is defined as

$$\mathcal{A}_f^{E_{n,k}} = \sum_{\mathbf{x} \in S_1, \mathbf{x}_2 \in S_2} (-1)^{f(\mathbf{x}_1) \oplus f(\mathbf{x}_2)}.$$

Let us explain the scenario for restricted autocorrelation over the domain $E_{6,3}$. We have classified all the $2^{\binom{6}{3}}$ patterns and computed the distribution of Boolean function corresponding to different restricted autocorrelation values in Figure 4. Such autocorrelation values are $\{-100, -80, -64, -60, -48, -40, -36, -32, -24, -20, -16, -12, -8, -4, 0, 4, 8, 12, 16, 20, 24, 32, 36, 40, 48, 60, 64, 80, 100\}$. The frequency of all such functions are normalized by dividing with

$2^{\binom{6}{3}}$. For 6-length PUFs we have randomly searched with 2^{20} different sets of delay parameters (α_i, β_i) and obtained 14100 such distinct functions. For them we also obtained the same set of distinct autocorrelation values. The normalized frequency distribution is drawn in Figure 4. A few blocks corresponding to certain autocorrelation values (such as $-100, -80, 80, 100$) in Figure 4 are not visible due to very small proportion.

From Figures 3 and 4, it can be observed that the restricted autocorrelation distribution of the PUFs demonstrates same behavior as the set of Boolean functions. That is the differential characteristics related to the bias is not observed for this restricted domain. That is, if the choice of two distinct challenge pairs can be restricted over certain domains (here one from S_1 and another from S_2 given a specific input bit location u), then the cryptographic weakness related to the bias might be avoided. Other than this different larger classes should be explored where such improved properties can be observed. Very simple model of Arbiter PUFs can be used there as cryptographic components with better confidence.

3.1 Theoretical Analysis

We now consider an Arbiter PUF whose inputs are from $E_{n,k}$. Let us divide the complete set $E_{n,k}$ into two subsets S_1 and S_2, where $S_1 = \{\mathbf{x} : \text{ MSB of } \mathbf{x} \text{ is } -1\}$ and $S_2 = \{\mathbf{x} : \text{ MSB of } \mathbf{x} \text{ is } 1\}$, i.e., the selected input bit is $u = n$.

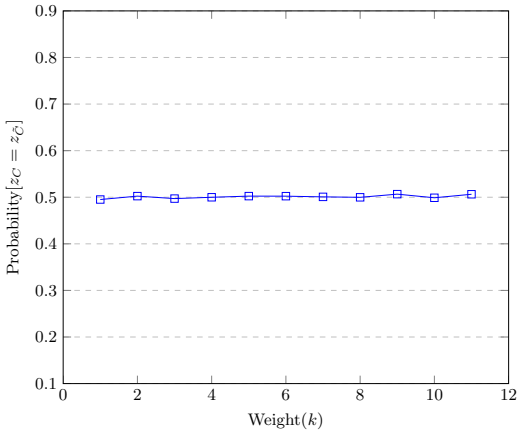


Fig. 5: Distribution of $Pr[z_C = z_{\tilde{C}}]$ in $E_{12,k}$

Weight(k)	$Pr[z_C = z_{\tilde{C}}]$
1	0.495117
2	0.502397
3	0.497111
4	0.499929
5	0.502405
6	0.502307
7	0.500808
8	0.499840
9	0.506651
10	0.498867
11	0.506392

Table 2: $Pr[z_C = z_{\tilde{C}}]$ in $E_{12,k}$

Consider challenge input C from S_1 and $\tilde{C}$ from S_2. For a randomly chosen PUF, let us denote z_C as the output corresponding to C and $z_{\tilde{C}}$ as the output corresponding to $\tilde{C}$. Compute the difference $z_C \oplus z_{\tilde{C}}$ for $C \in S_1$ and $\tilde{C} \in S_2$.

If we calculate the average for all the points $C \in S_1$ and $\tilde{C} \in S_2$, then we can estimate the quantity $p_i = Pr[z_C = z_{\tilde{C}}]$. Here p_i denotes the probability corresponding to i-th PUF say. We compute the average of all these probabilities (p_i's) for of all the different cases $E_{12,1}, E_{12,2}, \ldots, E_{12,11}$. The obtained experimental data is presented in Table 2 and the distribution is plotted in Figure 5. Note that $E_{12,0}$ and $E_{12,12}$ are not considered here as $|E_{12,0}| = |E_{12,12}| = 1$. From this experiment, we observe that the average probability is close to 0.5 for all weights $k = 1, \ldots, 11$. During the experiments, we have also observed that this probabilities do not depend of the choice of input bit t.

We note that this average probability is very close to 0.5 and that motivates us to explore the following theoretical result.

Theorem 3. *Expectation of $\mathcal{A}_f^{E_{n,k}}$ is equal to $\frac{1}{2}$ for $f \in \mathcal{B}_n^{\mathsf{PUF}}$.*

Proof. Consider two distinct challenge inputs $C, \tilde{C} \in E_{n,k}$ such that they must differ at location t_1. Here C and $\tilde{C}$ are of the same weight k, hence they will definitely differ at more than one locations. Let the m locations where C and $\tilde{C}$ differ be $t_1, t_2, \ldots, t_m$.

Let $\alpha = (\alpha_{t_1+1} + \beta_{t_1})P_{t_1} + (\alpha_{t_1+2} + \beta_{t_1+1})P_{t_1+1} + \ldots + (\alpha_{t_2} + \beta_{t_2-1})P_{t_2-1} + (\alpha_{t_3+1} + \beta_{t_3})P_{t_3} + \ldots + (\alpha_{t_4} + \beta_{t_4-1})P_{t_4-1} + \ldots + (\alpha_{t_m} + \beta_{t_m-1})P_{t_m-1}$ and $X = \Delta(C) - \alpha$. Thus the sign of $\Delta(C)$ corresponding to two challenge inputs $C, \tilde{C}$ will be same if and only if $|\frac{\alpha}{X}| < 1$. Hence the output bits corresponding two inputs C and $\tilde{C}$ will be same if and only if $|\frac{\alpha}{X}| < 1$.

As $\alpha_i, \beta_i \sim \mathcal{N}(0, \sigma)$, the quantity α will follow $\mathcal{N}(0, \sigma_\alpha)$ and X will follow $\mathcal{N}(0, \sigma_X)$, where $\sigma_\alpha = \sigma\sqrt{2[(t_2 - t_1) + (t_4 - t_3) + ... + (t_m - t_{m-1})]}$ and $\sigma_X = \sigma\sqrt{2n - 2[(t_2 - t_1) + (t_4 - t_3) + ... + (t_m - t_{m-1})]}$. The probability density functions of α and X will be $f_\alpha(y) = \frac{1}{\sqrt{2\pi}\sigma_\alpha} e^{-\frac{y^2}{2\sigma_\alpha^2}}$, $-\infty < y < \infty$ and $f_X(y) = \frac{1}{\sqrt{2\pi}\sigma_X} e^{-\frac{y^2}{2\sigma_X^2}}$, $-\infty < y < \infty$ respectively. Now we consider $Y_1 = \frac{\alpha}{X}$ and $Y_2 = X$. So $\alpha = Y_1 Y_2$. The joint distribution of α, X will be $f_{\alpha,X}(\alpha, x) = \frac{1}{2\pi\sigma_\alpha\sigma_X} e^{-\left(\frac{\alpha^2}{2\sigma_\alpha^2} + \frac{x^2}{2\sigma_X^2}\right)}$. Similarly, the joint distribution of Y_1, Y_2 will be $f_{Y_1,Y_2}(y_1, y_2) = \frac{1}{2\pi\sigma_\alpha\sigma_X} e^{-\left(\frac{y_1^2 y_2^2}{2\sigma_\alpha^2} + \frac{y_2^2}{2\sigma_X^2}\right)} y_2$, where $-\infty < y_1, y_2 < \infty$. The distribution of Y_1 will be $f_{Y_1}(y_1) = \int_{-\infty}^{\infty} f_{Y_1,Y_2}(y_1, y_2)dy_2 = \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma_\alpha\sigma_X} e^{-\left(\frac{y_1^2 y_2^2}{2\sigma_\alpha^2} + \frac{y_2^2}{2\sigma_X^2}\right)} y_2 dy_2 = \frac{1}{\pi} \frac{\frac{\sigma_\alpha}{\sigma_X}}{y_1^2 + \left(\frac{\sigma_\alpha}{\sigma_X}\right)^2}$, where $-\infty < y_1 < \infty$.

We already know that the output bits corresponding to the two inputs C and $\tilde{C}$ will be the same if and only if $|\frac{\alpha}{X}| < 1$. To calculate $Pr[|\frac{\alpha}{X}| < 1]$ we need to calculate $Pr[|Y_1| < 1]$.

$$Pr[|Y_1| < 1] = \Big| \int_{-1}^{1} \frac{1}{\pi} \frac{\frac{\sigma_\alpha}{\sigma_X}}{y_1^2 + \left(\frac{\sigma_\alpha}{\sigma_X}\right)^2} dy_1 \Big|$$

$$= \frac{1}{\pi}\Big|\{\tan^{-1}\Big(\frac{1}{\frac{\sigma_\alpha}{\sigma_X}}\Big) - \tan^{-1}\Big(\frac{-1}{\frac{\sigma_\alpha}{\sigma_X}}\Big)\}\Big|$$
$$= 1 - \frac{2}{\pi}\tan^{-1}\Big(\frac{\sigma_\alpha}{\sigma_X}\Big)$$
$$= 1 - \frac{2}{\pi}\tan^{-1}\Big(\sqrt{\frac{(t_2-t_1)+(t_4-t_3)+...+(t_m-t_{m-1})}{n-[(t_2-t_1)+(t_4-t_3)+...+(t_m-t_{m-1})]}}\Big)$$
$$= 1 - \frac{2}{\pi}\tan^{-1}\sqrt{\frac{t}{n-t}}.$$

Here $t = (t_2 - t_1) + (t_4 - t_3) + \cdots + (t_m - t_{m-1})$. Note that we have selected two distinct challenge inputs $C, \tilde{C}$ from $E_{n,k}$ with the condition that C and $\tilde{C}$ must differ at location t_1. Without loss of generality, we can assume that t_1-th location of C has -1 and t_1-th location of $\tilde{C}$ has 1. Let $S_1 = \{\mathbf{x} \mid \mathbf{x} \in E_{n,k}$ and t_1-th location of $\mathbf{x}$ has $-1\}$, $S_2 = \{\mathbf{x} \mid \mathbf{x} \in E_{n,k}$ and t_1-th location of $\mathbf{x}$ has $1\}$. That is $C \in S_1$, $\tilde{C} \in S_2$ and we have already noted $|S_1| = \binom{n-1}{k-1}$, $|S_2| = \binom{n-1}{k}$. If we consider the average probability for all choices of $C \in S_1$ and $\tilde{C} \in S_2$ then we will get the expectation of $\mathcal{A}_f^{E_{n,k}}$, where f is an n-length Arbiter PUF chosen uniformly at random. Hence,

$$\text{Expectation of } \mathcal{A}_f^{E_{n,k}} = \frac{1}{\binom{n-1}{k} \times \binom{n-1}{k-1}} \times \sum_{C \in S_1} \sum_{\tilde{C} \in S_2} \Big[1 - \frac{2}{\pi}\tan^{-1}\sqrt{\frac{t}{n-t}}\Big]$$
$$= 1 - \frac{1}{\binom{n-1}{k} \times \binom{n-1}{k-1}} \times \sum_{C \in S_1} \sum_{\tilde{C} \in S_2} \Big[\frac{2}{\pi}\tan^{-1}\sqrt{\frac{t}{n-t}}\Big].$$

We further simplify this. For every pair of inputs $C \in S_1$ and $\tilde{C} \in S_2$, one can compute $(1 - \frac{2}{\pi}\tan^{-1}\sqrt{\frac{t}{n-t}})$ as follows. Note that for every value of t, $\tan^{-1}\sqrt{\frac{t}{n-t}}$ and $\tan^{-1}\sqrt{\frac{n-t}{t}}$ both term will occur in the summation $\sum_{C \in S_1} \sum_{\tilde{C} \in S_2} \frac{2}{\pi}\tan^{-1}\sqrt{\frac{t}{n-t}}$. That means the above summation will contain $\tan^{-1}\sqrt{x} + \tan^{-1}\sqrt{\frac{1}{x}} = \frac{\pi}{2}$, for different values of x. As there are total $\binom{n-1}{k} \times \binom{n-1}{k-1}$ terms in the summation, the final expectation of $\mathcal{A}_f^{E_{n,k}}$ will be equal to $\frac{1}{2}$ for $f \in \mathcal{B}_n^{\mathsf{PUF}}$. This completes our proof. □

Let us provide an example with $n = 9$ and $k = 4$, i.e., $|E_{n,k}| = 126$. Hence $|S_1| = \binom{8}{3} = 56$ and $|S_2| = \binom{8}{4} = 70$. Let $T_i = \{(C, \tilde{C}) \in S_1 \times S_2 : t = i\}$. It can be checked that $|T_i| = |T_{n-i}|$, for $i = 1, \ldots, 8$. In $E_{9,4}$, $|T_1| = |T_8| = 35$, i.e., there are 35 pair of inputs $(C, \tilde{C}) \in S_1 \times S_2$, for which $t = 1$ and another different 35 pairs of inputs $(C, \tilde{C}) \in S_1 \times S_2$, for which $t = 8$. If we add $(1 - \frac{2}{\pi}\tan^{-1}\sqrt{\frac{t}{n-t}})$ for all these 70 pairs of distinct inputs, the final value becomes 35. Similarly $|T_2| = |T_7| = 215, |T_3| = |T_6| = 635$ and $|T_4| = |T_5| = 1075$. Hence the final expectation becomes $\frac{1}{{}^8C_3 \times {}^8C_4} \times [35 + 215 + 635 + 1075] = \frac{1}{2}$.

From the result of Theorem 3 it can be observed that if the challenge pairs are chosen with certain restrictions related to the input weights, then there does not exist any bias in the output of the Arbiter PUF. Thus in such restricted scenarios, such simple models of physically unclonable devices might provide acceptable cryptographic parameters.

In a related note, it has been shown [15] that certain cryptographic properties related to the Walsh spectrum of a Boolean function degrades in the restricted domain. Here we show that in the case of Arbiter PUFs, certain kind of autocorrelation property in a restricted sense, improves. The proposed notion of restricted autocorrelation might be explored for analyzing the security of FLIP [13] type ciphers under differential attack or related key attack.

4 Conclusion

In this paper, we have studied certain limitations of Arbiter PUFs and shown that the class of Boolean functions constructed using n-length ($n > 1$) PUFs is a proper subset of the set of all n variable Boolean functions. It is shown that exhaustively varying the delay parameters, the n-length Arbiter PUFs can only generate a negligible portion of Boolean functions. We present several existence and non-existence results in this direction. Further we have looked at autocorrelation in certain restricted sense and presented relevant results in this direction. It is known that the autocorrelation property of Boolean functions generated out of Arbiter PUFs is quite biased in certain cases. Interestingly, here we note that under certain constrains on the weights of inputs, along with the difference in a specific input bit, such biases vanish. That is, such a simple model of Arbiter PUFs provide good cryptographic parameters in terms of differential analysis if certain restrictions on the input challenge pairs are imposed.

References

1. G. T. Becker. The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs. In: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, pp. 535–555, 2015.
2. C. Brzuska, M. Fischlin, H. Schröder, S. Katzenbeisser. Physical Unclonable Functions in the Universal Composition Framework. In: Advances in Cryptology – CRYPTO 2011. LNCS, Springer, pp. 51–70, 2011.
3. A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier and R. Sirdey. Stream Ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression. Journal of Cryptology 31:3 (2018), `https://doi.org/10.1007/s00145-017-9273-9` (earlier version in FSE 2016, LNCS 9783, pp. 313–333, Springer).
4. C. Carlet, P. Méaux and Y. Rotella. Boolean Functions with Restricted Input and their Robustness; application to the FLIP cipher. IACR Transactions on Symmetric Cryptology, vol 3 (2017), pp. 192–227 (presented at FSE 2018).
5. J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede. Secure Lightweight Entity Authentication with Strong PUFs: Mission Impossible? In: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, pp. 451–475, 2014.

6. S. Devadas. Physical Unclonable Functions and Secure Processors. In: Cryptographic Hardware and Embedded Systems–CHES 2009, Springer, pp. 65–65 2009.
7. B. Gassend. Physical Random Functions. M.S. thesis, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Jan. 2003. Available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.7571&rep=rep1&type=pdf
8. B. Gassend, D. Clarke, M. V. Dijk, and S. Devadas. Silicon Physical Random Functions. In: Proceedings of the 9th ACM conference on Computer and communications security, ACM pp. 148–160, 2002. Available at: https://dl.acm.org/citation.cfm?id=586132.
9. G. Hammouri, and B. Sunar. PUF-HB: A Tamper-Resilient HB Based Authentication Protocol. In: International Conference on Applied Cryptography and Network Security, Springer, pp. 346–365, 2008.
10. J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. V. Dijk and S. Devadas. A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications. In 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525), IEEE, pp. 176–179, 2004. Available at: https://people.csail.mit.edu/devadas/pubs/vlsi-symp-puf.pdf.
11. D. Lim. Extracting Secret Keys from Integrated Circuits. M.Sc. thesis, MIT, 2004.
12. D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. V. Dijk, and S. Devadas. Extracting Secret Keys from Integrated Circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13 (10), pp. 1200–1205, 2005.
13. P. Méaux, A. Journault, F.-X. Standaert and C. Carlet. Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. Advances in Cryptology-35th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT, 2016), Springer, pp. 311–343, 2016.
14. S. Maitra, B. Mandal, T. Martinsen, D. Roy and P. Stănică. Tools in Analyzing Linear Approximation for Boolean Functions Related to FLIP In the Proceeding of the 19th International Conference on Cryptology in India (Indocrypt 2018), Springer, pp. 282–303, 2018.
15. S. Maitra, B. Mandal, T. Martinsen, D. Roy and P. Stănică. Analysis on Boolean Function in a Restricted (biased) domain. IEEE Trans. Inf. Theory, vol 66(2), pp. 1219–1231, 2020.
16. M. Majzoobi, F. Koushanfar, and M. Potkonjak. Testing Techniques for Hardware Security. In International Test Conference (ITC), IEEE, pp. 1–10, 2008.
17. S. Mesnager, Z. Zhou and C. Ding. On the Nonlinearity of Boolean Functions with Restricted Input. Cryptography and Communications, vol. 11(1), pp. 63–76, 2019.
18. U. Rührmair, H. Busch, S. Katzenbeisser. Strong PUFs: Models, Constructions and Security Proofs. In: Towards Hardware Intrinsic Security: Foundation and Practice, Springer, pp. 79–96, 2010.
19. U. Rührmair, S. Devadas, F. Koushanfar. Security based on Physical Unclonability and Disorder. In: Introduction to Hardware Security and Trust, Springer, pp. 65–102, 2012.
20. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, J. Schmidhuber. Modeling Attacks on Physical Unclonable Functions. In: Proceedings of the 17th ACM conference on Computer and communications security, ACM, pp. 237–249, 2010.
21. U. Rührmair, J. Sölter, F. Sehnke. On the Foundations of Physical Unclonable Functions. Cryptology ePrint Archive, 2009:277, 2009. Available at: https://eprint.iacr.org/2009/277.pdf

22. U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, S. Devadas. PUF Modeling Attacks on Simulated and Silicon Data. IEEE Transactions on Information Forensics and Security, vol. 8(11), pp. 1876–1891, 2013.
23. SageMath: A free open-source mathematics software. `https://www.sagemath.org/`
24. A. A. Siddhanti, S. Bodapati, A. Chattopadhyay, S. Maitra, D. Roy, P. Stănică. Analysis of the Strict Avalanche Criterion in Variants of Arbiter-Based Physically Unclonable Functions. In: Progress in Cryptology – INDOCRYPT 2019, LNCS, Springer, 2019.