Designs, Codes and Cryptography (2026) 94:104
https://doi.org/10.1007/s10623-026-01861-7

Multipath PA-PUFs generate all Boolean functions

R. Radheshwar[1,3] · Dibyendu Roy[1] · Pantelimon Stănică[2]

Received: 18 April 2026 / Revised: 20 April 2026 / Accepted: 20 April 2026

Abstract
In this paper, we propose a generalized model of Priority Arbiter-based Physical Unclonable Function (PA-PUF) with an arbitrary number of paths inside each switch. We first develop a mathematical model for this generalized model. Experimentally, we observed that the class of Boolean functions generated from our model of PA-PUF increases proportionally with the number of paths inside each switch, and that motivated us to attempt one of the open challenges proposed by Kansal et al. (Discrete Appl Math 356:71–95, 2024). We first show that the set of Boolean functions generated from i-length PA-PUF with $(i+1)$ number of paths is a proper super set of the set of Boolean functions generated from i-length PA-PUF with i number of paths. Based upon that, we show in our main result that we need at least $(n+1)$ numbers of paths inside each switch of an n-length PA-PUF to generate all the Boolean functions involving n-number of variables. Furthermore, we performed significant software and hardware experimentations to assess the resilience of our model against machine learning based modeling attacks.

Keywords PA-PUF · Boolean function · Delay · Distribution

1 Introduction

Cryptographic primitives rely on the use of one-way functions. Physical Unclonable Functions (PUFs) [5] are hardware based one-way functions with random looking bits. Unlike conventional cryptographic primitives, PUFs depend on the physical attributes of a device, which are unclonable in nature. The arbiter-based PUF was introduced by Gassend et al. [5] in

Communicated by C. Mitchell.

✉ Dibyendu Roy
dibyendu.roy@iiitvadodara.ac.in

R. Radheshwar
202273001@iiitvadodara.ac.in

Pantelimon Stănică
pstanica@nps.edu

[1] Maths & Computing, Indian Institute of Information Technology Vadodara, Gandhinagar, Gujarat 382028, India

[2] Applied Mathematics Department, Naval Postgraduate School, Monterey, CA 93943, USA

[3] Centre for Development of Advanced Computing, Bengaluru, Karnataka 560038, India

Published online: 08 May 2026

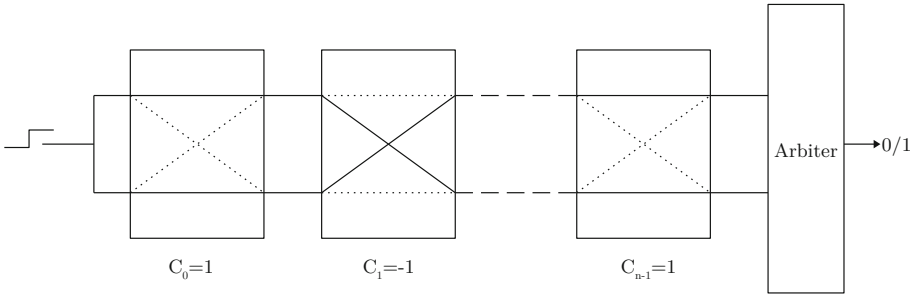

Fig. 1 Arbiter-based PUF

2002. This topic has garnered considerable attention within the cryptology community, leading to the development of several applications such as identification and authentication [11]. PUF-based RFID tags provided robust authentication at low cost. PUFs are used to generate cryptographic keys at runtime and thereby prevent attacks on keys stored in non-volatile memory.

An n-length arbiter-based PUF consists of n switches where each switch is associated with two paths top and bottom (see Fig. 1). One signal is introduced at the first stage of the arbiter-based PUF and the signal is transmitted through each path of every switch from the 1st switch to the n-th switch. Every switch has an input called challenge which is either 1 or -1. If the input supplied to a switch is 1, the signal follows the same path. The signal alters the path if the input to the switch is -1. The challenge input to the i-th switch is denoted by C_i. Since each path has its specific characteristics, the traveling speed of the signal will be different. Thus there will be some amount of delay associated with each path of every switch. The delay parameters are random in nature and follow a normal distribution. At the final stage we have one arbiter which measures the delay differences of the signal between the top and bottom path. Depending upon the sign of the delay difference the arbiter produces either 0 or 1 as output.

The mathematical model of an n-length arbiter-based PUF was derived by Lim et al. [12]. If we consider $C^n = (C_0, \ldots, C_{n-1}) \in \{-1, 1\}^n$ as the challenge input of an arbiter-based PUF, then the delay difference between top path and bottom path after the final switch will be

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

where $P_k = \prod_{i=k+1}^{n} C_i$, for $k = 0, \ldots, n-1$ and $P_n = 1$, and the parameters $\alpha_i = \frac{p_i - q_i}{2} + \frac{r_i - s_i}{2}$, $\beta_i = \frac{p_i - q_i}{2} - \frac{r_i - s_i}{2}$, where, $p_i, q_i, r_i, s_i \sim \mathcal{N}(\mu, \sigma)$. Depending upon the sign of $\Delta(C^n)$, the arbiter returns 0 or 1. An n-length PUF can be considered as an n-variable Boolean function using a simple transformation $\{1, -1\} \mapsto \{0, 1\}$.

Feed forward PUF is an extension of the classical PUF by introducing an arbiter in the middle (say after the i-th switch) and based on the introduced arbiter's output, a challenge (C_{i+1}) value is decided. The XOR-PUF uses multiple PUF circuits and combines the arbiters' responses via the XOR. These new types of PUFs improved some features of PUFs like uniqueness, reliability, etc. An ideal PUF is expected to possess good cryptographic properties like Uniqueness, Uniformity and Reliability [14]. Uniqueness requires that if the same challenge bits are given to two different PUFs then the responses should also be different. Uniformity states that the set of all responses should be balanced. Reliability requires that for a given challenge, a PUF should always produce the same response.

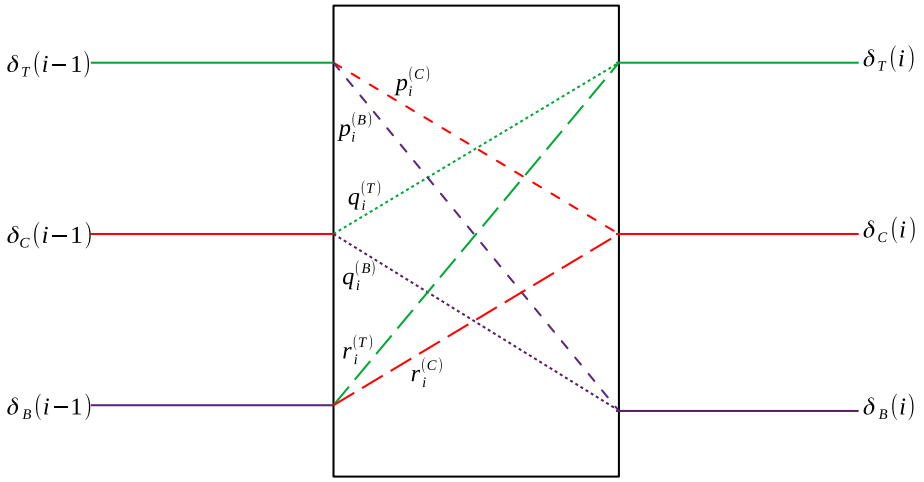

Fig. 2 One switch of PA-PUF [21]

In literature, different types of PUF were introduced, namely feed forward PUF [11], XOR PUF [4], ring oscillator PUF [22], SRAM PUF [6], Butterfly PUF [10], Glitch PUF [23], MEmory Cell-based Chip Authentication (MECCA) PUF [9]. PUFs can also be modeled using machine learning approaches. For example, Mishra et al. [15] used a search optimization based framework to construct a functional model of a k-XOR PUF from $(k-1)$-XOR PUFs, emphasizing structural characterization rather than adversarial analysis.

Strong PUFs [2] are characterized by a large number (more than 64 bits in general) of challenge-response pairs (CRPs) and a public interface to query the PUF, that is, a user should be able to feed any challenge and obtain its corresponding response. Additionally, the challenge-response set should be sufficiently large given the public interface to prevent an attacker from obtaining every potential CRP. The Arbiter PUF, XOR Arbiter PUF, Feed Forward Arbiter PUF, and analogue PUFs [3] are a few common Strong PUF architectures. Weak PUFs have a limited querying interface because they permit comparatively fewer CRPs. Ring oscillator PUFs, SRAM PUFs, butterfly PUFs, buskeeper PUFs [20], transistor PUFs [13], or diode PUFs [17] are a few examples of weak PUF structures.

In 2018, Siddhanti et al. [19] studied the cryptographic properties of arbiter-based PUF. They have observed that the outputs from arbiter-based PUF are highly correlated when the challenge inputs differ at specific position. They have derived the mathematical formulae for computing the probabilities in which output bits will be equal under different circumstances. Late in 2021, Roy et al. [16] have thoroughly looked into the reason behind the presence of non-randomness in the output bits observed by Siddhanti et al. [19]. They have noticed that the set of Boolean function generated from classical arbiter-based PUF is quite narrow compared to the complete set of Boolean functions for a number of variables greater than 1. Various other security analyses [1, 18], were also performed on PUF and other variants of PUF.

In 2022, Singh et al. [21] proposed a three path based PUF called Priority Arbiter-based PUF (PA-PUF). The path alteration is based on the challenge input as in the PUF. Here the path is always altered and the path alteration occurs as shown in Fig. 2. The arbiter has the ability to determine the order in which the signals arrive, which were sent through the three paths namely Top (T), Center (C) and Bottom (B) paths. The arbiter returns 0 or 1 based on

a particular priority order in which the signal arrives to the arbiter. If the order of arrival is $\{T, B, C\}$ or $\{T, C, B\}$ or $\{C, T, B\}$ then response is 0. Otherwise the order of arrival will be $\{B, T, C\}$ or $\{B, C, T\}$ or $\{C, B, T\}$ and in this case the response is 1.

In 2024, Kansal et al. [8] derived a mathematical model for the PA-PUF [21]. Here, the arbiter returns 0 or 1 based on the sign of the delay difference between bottom and top path. The authors [8] have shown that the set of all Boolean functions generated from PA-PUF is notably larger than the set of all Boolean functions generated from classical Arbiter-based PUF, though, the entire class of Boolean functions could still not be completely generated.

1.1 Preliminaries and notations

In this subsection, we introduce all the notations and definitions used throughout our article. A Boolean function f in n variables is a map from $\{0, 1\}^n$ to $\{0, 1\}$. The set of all n-variable Boolean functions is denoted by $\mathcal{B}_n$ and $|\mathcal{B}_n| = 2^{2^n}$. The set of Boolean functions generated from n-length PUFs is denoted by $\mathcal{B}_n^{\mathsf{PUF}}$. We extend the prior concept of PA-PUF to i-paths where $i \in \{2, 3, \ldots\}$. When $i = 2$, the PA-PUF represents a classical 2-path PUF. For $i = 3$, our generalization becomes previously proposed PA-PUF [21]. A j-length PA-PUF with i paths is a hardware device based on j number of switches, where each switch is associated with i symmetrical paths. This switch can take either 1 or -1 as input (challenge, C_j) and finally the arbiter placed in the last stage produces either 0 or 1 as output (response). Every n-length PA-PUF eventually can also be seen as a Boolean function involving n number of variables using the transformation $\{1, -1\} \mapsto \{0, 1\}$. We denote the set of all j-variable Boolean functions that can be generated by i-path PA-PUF as $\mathcal{B}_{i,j}^{\mathsf{PA-PUF}}$.

1.2 Motivation and contributions

In [16], the authors pointed out that all possible Boolean functions involving 1-variable can be generated by using 1-length PUFs, that is, $\mathcal{B}_1^{\mathsf{PUF}} = \mathcal{B}_1$. For $n = 2$, they showed that $\mathcal{B}_2^{\mathsf{PUF}} \subset \mathcal{B}_2$, and in fact, there are two Boolean functions which can not be generated using 2-length PUFs with two paths.

The authors in [8] showed that a 3-path PA-PUF [21] is able to construct all possible 2-variable Boolean functions, i.e $\mathcal{B}_{3,2}^{\mathsf{PA-PUF}} = \mathcal{B}_2$. This model of PA-PUF is able to generate only 214 Boolean functions involving 3 variables, which motivated us to increase the number of paths in the PA-PUF model and check whether it is possible to generate all the possible 3-variable Boolean functions, that is, 256 Boolean functions. Increasing the number of paths led us to the 4-path PA-PUF, but surprisingly, that method in the defined model [21] did not construct all possible 3-variable Boolean functions.

In Sect. 2, we introduce a new model of PA-PUF with 4 paths inside each switch. We also develop its mathematical model for further analysis. We have shown the existence of one 3-variable Boolean function in $\mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$ which was not present in $\mathcal{B}_{3,3}^{\mathsf{PA-PUF}}$ [8]. Our new model of 3-length PA-PUF with 4 paths generates all 3-variable Boolean functions, that is, $\mathcal{B}_{4,3}^{\mathsf{PA-PUF}} = \mathcal{B}_3$. This experimental observation motivates us to extend our work further and attempt the open problem raised by Kansal et al. [8]:

> *What is the optimal number of paths required in the circuit of a PA-PUF to generate the entire set $\mathcal{B}_n$ for every n ?*

Table 1 Comparison with previous models of PUFs

n	$\lvert\mathcal{B}_{2,n}^{\mathsf{PUF}}\rvert$ [16]	$\lvert\mathcal{B}_{3,n}^{\mathsf{PA-PUF}}\rvert$[8]	$\lvert\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}\rvert$ (Our model)
1	4	4	4
2	14	16	16
3	104	214	256
4	1882	16584	65536

From our experimental observations (as in Table 1), we were led to conjecture (now, a theorem) that we need $(n+1)$ paths inside every switch to generate the complete set of all Boolean functions, (that is, 2^{2^n}). In Sect. 3, we generalize our model of PA-PUF to n-paths. We first show that $\mathcal{B}_{i+1,i}^{\mathsf{PA-PUF}} \supset \mathcal{B}_{i,i}^{\mathsf{PA-PUF}}$. Finally, we prove that our model of PA-PUF with $(n+1)$ paths can generate all n-variable Boolean functions, that is, $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}} = \mathcal{B}_n$.

In Sect. 4 we implement one toy model of our proposed PA-PUF in FPGA. For our implementation we consider a 16-length PA-PUF with 17 paths in every switch. Further, we investigate Uniqueness, Uniformity, and Reliability of our considered PA-PUF. Finally, we notice that the predictability rate is also low ($\approx 54.95\%$) for our model of PA-PUF against modern machine learning based attacks.

2 PA-PUF with 4 paths

In this section, we generalize the classical 2-path PUF (as in Fig. 1) to a 4-path PA-PUF. Here, every switch is associated with 4 paths instead of 2 paths as in the classical PUF. We will give one common pulse in the first switch and the pulse moves from the first switch to the last switch via the four paths we have for each switch. After the last switch one arbiter is placed, which measures the delay differences between the paths and outputs either 0 or 1. We will define later how the output bit is decided.

The condition for path alteration in the classical PUF is when the challenge is -1 or else the path is unaltered. Using the same condition, we define one switch of a PA-PUF as in Fig. 3 and 4, where the dashed line represents the path if the challenge is $C_i = -1$, which results in an alteration in the paths. The signal follows the same path if the challenge is $C_i = 1$. The path alteration technique is similar to the classical 2-path PUF either the signal goes unaltered or the path is altered to the immediate below path. One notes that in the case of a 2-path PUF, the path alternation follows the permutation $\sigma_2 : \begin{pmatrix} Pt_1 & Pt_2 \\ Pt_2 & Pt_1 \end{pmatrix}$. We are generalizing this path alteration mechanism to a 4-path PA-PUF. We follow the Fig. 3 to determine the path of the signal. Here we use the following permutation (cycle of order 4) σ_4 for computing the delay in paths when there is an alteration of path of the pulse. $\sigma_4 : \begin{pmatrix} Pt_1 & Pt_2 & Pt_3 & Pt_4 \\ Pt_4 & Pt_1 & Pt_2 & Pt_3 \end{pmatrix}$.

We have already seen that each path is associated with a delay, which is based on device-specific parameters. We denote the delay parameters (which follow normal distribution) by $p_i^{(Pt_X)}, q_i^{(Pt_X)}, r_i^{(Pt_X)}$ and $s_i^{(Pt_X)}$ where $X \in \{1, 2, 3, 4\}$ corresponding to Path 1, Path 2, Path 3 and Path 4, respectively, at the i-th switch. Here, the path number starts from top and ends at bottom. We denote the delay at the i-th switch ($i \geq 0$) by $\delta_{Pt_X}(i)$.

We measure the delay at every i-th switch using the iterative relations described in Eqs. (1), (2), (3), (4) in the four paths as $\delta_{Pt_1}(i)$, $\delta_{Pt_2}(i)$, $\delta_{Pt_3}(i)$ and $\delta_{Pt_4}(i)$ for Path 1,

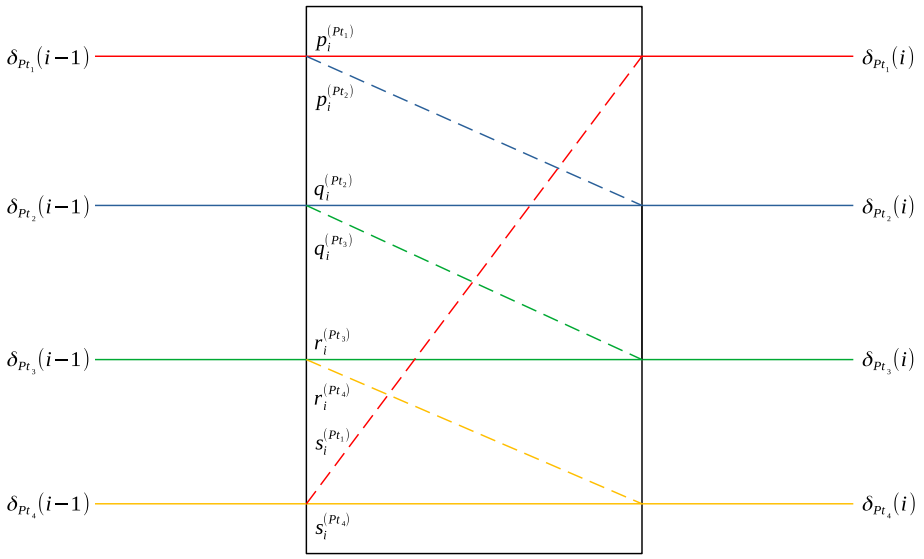

Fig. 3 Path alteration in 4-path PA-PUF

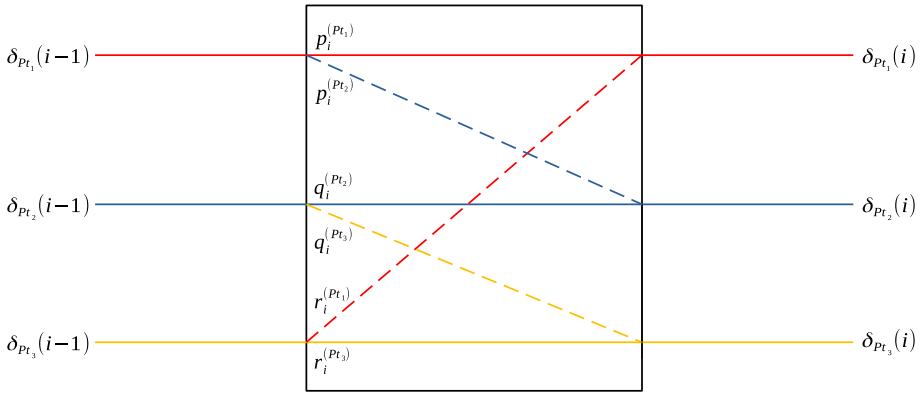

Fig. 4 Path alteration in 3-path PA-PUF

Path 2, Path 3 and Path 4, respectively,

$$\delta_{Pt_1}(i) = \left(\frac{C_i+1}{2}\right)\left(\delta_{Pt_1}(i-1) + p_i^{(Pt_1)}\right) + \left(\frac{1-C_i}{2}\right)\left(\delta_{Pt_4}(i-1) + s_i^{(Pt_1)}\right) \quad (1)$$

$$\delta_{Pt_2}(i) = \left(\frac{C_i+1}{2}\right)\left(\delta_{Pt_2}(i-1) + q_i^{(Pt_2)}\right) + \left(\frac{1-C_i}{2}\right)\left(\delta_{Pt_1}(i-1) + p_i^{(Pt_2)}\right) \quad (2)$$

$$\delta_{Pt_3}(i) = \left(\frac{C_i+1}{2}\right)\left(\delta_{Pt_3}(i-1) + r_i^{(Pt_3)}\right) + \left(\frac{1-C_i}{2}\right)\left(\delta_{Pt_2}(i-1) + q_i^{(Pt_3)}\right) \quad (3)$$

$$\delta_{Pt_4}(i) = \left(\frac{C_i+1}{2}\right)\left(\delta_{Pt_4}(i-1) + s_i^{(Pt_4)}\right) + \left(\frac{1-C_i}{2}\right)\left(\delta_{Pt_3}(i-1) + r_i^{(Pt_4)}\right). \quad (4)$$

Using these delay relations, we derive the general form of the delay difference between one path with the other paths in Eqs. (5), (6), (7), (8). The $\Delta^i_{Pt_Y Pt_X}(C^{i+1})$ is the general form of the delay differences between the Y-th path and X-th path after the i-th switch,

$$\Delta^i_{Pt_1 Pt_X}(C^{i+1}) = \left(\frac{C_i+1}{2}\right)\left(\Delta^{i-1}_{Pt_1 Pt_X}(C^i) + p_i^{(Pt_1)} - v_i^{(Pt_X)}\right) + \left(\frac{1-C_i}{2}\right)\left(\Delta^{i-1}_{Pt_4\sigma_4(Pt_X)}(C^i) + s_i^{(Pt_1)} - u_i^{(Pt_X)}\right) \quad (5)$$

$$\Delta^i_{Pt_2 Pt_X}(C^{i+1}) = \left(\frac{C_i+1}{2}\right)\left(\Delta^{i-1}_{Pt_2 Pt_X}(C^i) + q_i^{(Pt_2)} - v_i^{(Pt_X)}\right) + \left(\frac{1-C_i}{2}\right)\left(\Delta^{i-1}_{Pt_1\sigma_4(Pt_X)}(C^i) + p_i^{(Pt_2)} - u_i^{(Pt_X)}\right) \quad (6)$$

$$\Delta^i_{Pt_3 Pt_X}(C^{i+1}) = \left(\frac{C_i+1}{2}\right)\left(\Delta^{i-1}_{Pt_3 Pt_X}(C^i) + r_i^{(Pt_3)} - v_i^{(Pt_X)}\right) + \left(\frac{1-C_i}{2}\right)\left(\Delta^{i-1}_{Pt_2\sigma_4(Pt_X)}(C^i) + q_i^{(Pt_3)} - u_i^{(Pt_X)}\right) \quad (7)$$

$$\Delta^i_{Pt_4 Pt_X}(C^{i+1}) = \left(\frac{C_i+1}{2}\right)\left(\Delta^{i-1}_{Pt_4 Pt_X}(C^i) + s_i^{(Pt_4)} - v_i^{(Pt_X)}\right) + \left(\frac{1-C_i}{2}\right)\left(\Delta^{i-1}_{Pt_3\sigma_4(Pt_X)}(C^i) + r_i^{(Pt_4)} - u_i^{(Pt_X)}\right). \quad (8)$$

Here σ_4 is a permutation on $\{Pt_1, Pt_2, Pt_3, Pt_4\}$, defined by $\sigma_4 : \begin{pmatrix} Pt_1 & Pt_2 & Pt_3 & Pt_4 \\ Pt_4 & Pt_1 & Pt_2 & Pt_3 \end{pmatrix}$. The quantities $v_i^{(Pt_X)}$ and $u_i^{(Pt_X)}$ are the delay parameters involved in Pt_X path when $C_i = 1$ and $C_i = -1$, respectively. The relations on delays and delay differences are iterative, and so, we need initial conditions. We impose all $\delta(-1)$ to be 0 and initial delay differences are also equal to 0.

The delay difference $\Delta^i_{Pt_1 Pt_2}(C^{i+1})$ after the i-th switch is computed by taking the difference between Eqs. (1) and (2). Similarly $\Delta^i_{Pt_1 Pt_3}(C^{i+1})$, $\Delta^i_{Pt_1 Pt_4}(C^{i+1})$ are also computed. The arbiter returns 1 if $\Delta^{n-1}_{Pt_1 Pt_X}(C^n) < 0$ for all $Pt_X \in \{Pt_2, Pt_3, Pt_4\}$ or $\Delta^{n-1}_{Pt_2 Pt_X}(C^n) < 0$ for all $Pt_X \in \{Pt_1, Pt_3, Pt_4\}$. The arbiter returns 0, otherwise, that is, $\Delta^{n-1}_{Pt_3 Pt_X}(C^n) < 0$ for all $Pt_X \in \{Pt_1, Pt_2, Pt_4\}$ or $\Delta^{n-1}_{Pt_4 Pt_X}(C^n) < 0$ for all $Pt_X \in \{Pt_1, Pt_2, Pt_3\}$. This implies that if the signal reaches first in Path 1 or Path 2, the arbiter returns 1 and returns 0 if the signal reaches first in Path 3 or Path 4. In Proposition 1, we show that a Boolean function which was not present in the set of all 3-variable Boolean functions generated by 3-path PA-PUF [8] is present in $\mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$. Experimentally, we observed that our proposed model is able to generate all possible 3-variable Boolean functions, that is, $\mathcal{B}_{4,3}^{\mathsf{PA-PUF}} = \mathcal{B}_3$ (we show this result later in our paper, for any number of variables).

Table 2 Complete set of conditions for f_1

C_2	C_1	C_0	$\Delta^2_{Pt_j Pt_X}(C^3)$ (or) $\Delta^2_{Pt_{j+1} Pt_X}(C^3)$
1	1	1	$r_0^{(Pt_3)} - v_0^{(Pt_X)} + r_1^{(Pt_3)} - v_1^{(Pt_X)} + r_2^{(Pt_3)} - v_2^{(Pt_X)} < 0$ (or) $s_0^{(Pt_4)} - v_0^{(Pt_X)} + s_1^{(Pt_4)} - v_1^{(Pt_X)} + s_2^{(Pt_4)} - v_2^{(Pt_X)} < 0$
1	1	−1	$q_0^{(Pt_3)} - u_0^{(Pt_X)} + r_1^{(Pt_3)} - v_1^{(Pt_X)} + r_2^{(Pt_3)} - v_2^{(Pt_X)} < 0$ (or) $r_0^{(Pt_4)} - u_0^{(Pt_X)} + s_1^{(Pt_4)} - v_1^{(Pt_X)} + s_2^{(Pt_4)} - v_2^{(Pt_X)} < 0$
1	−1	1	$q_0^{(Pt_2)} - v_0^{\sigma_4(Pt_X)} + q_1^{(Pt_3)} - u_1^{(Pt_X)} + r_2^{(Pt_3)} - v_2^{(Pt_X)} < 0$ (or) $r_0^{(Pt_3)} - v_0^{\sigma_4(Pt_X)} + r_1^{(Pt_4)} - u_1^{(Pt_X)} + s_2^{(Pt_4)} - v_2^{(Pt_X)} < 0$
1	−1	−1	$r_0^{(Pt_4)} - u_0^{\sigma_4(Pt_X)} + s_1^{(Pt_1)} - u_1^{(Pt_X)} + p_2^{(Pt_1)} - v_2^{(Pt_X)} < 0$ (or) $s_0^{(Pt_1)} - u_0^{\sigma_4(Pt_X)} + p_1^{(Pt_2)} - u_1^{(Pt_X)} + q_2^{(Pt_2)} - v_2^{(Pt_X)} < 0$
−1	1	1	$p_0^{(Pt_1)} - v_0^{\sigma_4^2(Pt_X)} + p_1^{(Pt_1)} - v_1^{\sigma(Pt_X)} + s_2^{(Pt_4)} - u_2^{(Pt_X)} < 0$ (or) $q_0^{(Pt_2)} - v_0^{\sigma_4^2(Pt_X)} + q_1^{(Pt_2)} - v_1^{\sigma(Pt_X)} + p_2^{(Pt_1)} - u_2^{(Pt_X)} < 0$
−1	1	−1	$p_0^{(Pt_2)} - u_0^{\sigma_4^2(Pt_X)} + q_1^{(Pt_2)} - v_1^{\sigma(Pt_X)} + q_2^{(Pt_3)} - u_2^{(Pt_X)} < 0$ (or) $q_0^{(Pt_3)} - u_0^{\sigma_4^2(X)} + r_1^{(Pt_3)} - v_1^{\sigma(Pt_X)} + r_2^{(Pt_4)} - u_2^{(Pt_X)} < 0$
−1	−1	1	$p_0^{(Pt_1)} - v_0^{\sigma_4^2(Pt_X)} + p_1^{(Pt_2)} - u_1^{\sigma(Pt_X)} + q_2^{(Pt_3)} - u_2^{(Pt_X)} < 0$ (or) $q_0^{(Pt_2)} - v_0^{\sigma_4^2(Pt_X)} + q_1^{(Pt_3)} - u_1^{\sigma(Pt_X)} + r_2^{(Pt_4)} - u_2^{(Pt_X)} > 0$
−1	−1	−1	$s_0^{(Pt_1)} - u_0^{\sigma_4^2(Pt_X)} + p_1^{(Pt_2)} - u_1^{\sigma(Pt_X)} + q_2^{(Pt_3)} - u_2^{(Pt_X)} < 0$ (or) $p_0^{(Pt_2)} - u_0^{\sigma_4^2(Pt_X)} + q_1^{(Pt_3)} - u_1^{\sigma(Pt_X)} + r_2^{(Pt_4)} - u_2^{(Pt_X)} > 0$

In [8], it is shown that the function $f_1 = (0, 0, 0, 1, 1, 0, 0, 0)$ is not present in the set of all 3-variable Boolean functions generated by a 3-path PA-PUF. In Proposition 1 we prove that f_1 belongs to $\mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$.

Proposition 1 *The function* $f_1 = (0, 0, 0, 1, 1, 0, 0, 0) \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$.

Proof By considering the truth table of f_1 and the Eqs. (5), (6), (7) and (8) we derive $\Delta^2_{Pt_j Pt_X}(C^3)$ and $\Delta^2_{Pt_{j+1} Pt_X}(C^3)$ in a 4-path setting. The conditions are described in Table 2.

We consider one set of possible inequalities from Table 2 in Table 3 to investigate whether the system has any solution or not.

We observe that there exists a solution to the above system described in Table 3, given by: $p_0^{(Pt_2)} < -q_2^{(Pt_3)} - q_1^{(Pt_2)} + u_2^{(Pt_X)} + u_0^{\sigma_4^2(Pt_X)} + v_1^{\sigma_4(Pt_X)}, q_0^{(Pt_3)} < -r_1^{(Pt_3)} - r_2^{(Pt_3)} + u_0^{(Pt_X)} + v_1^{(Pt_X)} + v_2^{(Pt_X)}, p_1^{(Pt_2)} < \min\{-q_2^{(Pt_3)} - s_0^{(Pt_1)} + u_2^{(Pt_X)} + u_1^{\sigma_4(Pt_X)} + u_0^{\sigma_4^2(Pt_X)}, -q_2^{(Pt_3)} - p_0^{(Pt_1)} + u_2^{(Pt_X)} + u_1^{\sigma_4(Pt_X)} + v_0^{\sigma_4^2(Pt_X)}\}, b2 < -r_0^{(Pt_4)} - s_1^{(Pt_1)} + u_1^{(Pt_X)} + u_0^{\sigma_4(Pt_X)} + v_2^{(Pt_X)}, q_1^{(Pt_3)} < -q_0^{(Pt_2)} - r_2^{(Pt_3)} + u_1^{(Pt_X)} + v_2^{(Pt_X)} + v_0^{\sigma_4(Pt_X)}, p_0^{(Pt_1)} < -p_1^{(Pt_1)} - s_2^{(Pt_4)} + u_2^{(Pt_X)} + v_1^{\sigma_4(Pt_X)} + v_0^{\sigma_4^2(Pt_X)}, r_0^{(Pt_3)} < -r_1^{(Pt_3)} - r_2^{(Pt_3)} + v_0^{(Pt_X)} + v_1^{(Pt_X)} + v_2^{(Pt_X)}$. □

Table 3 One set of conditions from Table 2

C_2	C_1	C_0	$\Delta^2_{Pt_j Pt_X}(C^3)$
1	1	1	$r_0^{(Pt_3)} - v_0^{(Pt_X)} + r_1^{(Pt_3)} - v_1^{(Pt_X)} + r_2^{(Pt_3)} - v_2^{(Pt_X)} < 0$
1	1	−1	$q_0^{(Pt_3)} - u_0^{(Pt_X)} + r_1^{(Pt_3)} - v_1^{(Pt_X)} + r_2^{(Pt_3)} - v_2^{(Pt_X)} < 0$
1	−1	1	$q_0^{(Pt_2)} - v_0^{\sigma_4(Pt_X)} + q_1^{(Pt_3)} - u_1^{(Pt_X)} + r_2^{(Pt_3)} - v_2^{(Pt_X)} < 0$
1	−1	−1	$r_0^{(Pt_4)} - u_0^{\sigma_4(Pt_X)} + s_1^{(Pt_1)} - u_1^{(Pt_X)} + p_2^{(Pt_1)} - v_2^{(Pt_X)} < 0$
−1	1	1	$p_0^{(Pt_1)} - v_0^{\sigma_4^2(Pt_X)} + p_1^{(Pt_1)} - v_1^{\sigma_4(Pt_X)} + s_2^{(Pt_4)} - u_2^{(Pt_X)} < 0$
−1	1	−1	$p_0^{(Pt_2)} - u_0^{\sigma_4^2(Pt_X)} + q_1^{(Pt_2)} - v_1^{\sigma_4(Pt_X)} + q_2^{(Pt_3)} - u_2^{(Pt_X)} < 0$
−1	−1	1	$p_0^{(Pt_1)} - v_0^{\sigma_4^2(Pt_X)} + p_1^{(Pt_2)} - u_1^{\sigma_4(Pt_X)} + q_2^{(Pt_3)} - u_2^{(Pt_X)} < 0$
−1	−1	−1	$s_0^{(Pt_1)} - u_0^{\sigma_4^2(Pt_X)} + p_1^{(Pt_2)} - u_1^{\sigma_4(Pt_X)} + q_2^{(Pt_3)} - u_2^{(Pt_X)} < 0$

Similarly, one can follow the similar method to prove that all 42 Boolean functions f_i ($i = 1, \ldots, 42$) which were absent in the $\mathcal{B}_{3,3}^{\mathsf{PA-PUF}}$ [8] can be generated using a 4-path PA-PUF involving 3 variables, that is, $f_i \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$, $i = 1, \ldots, 42$.

3 PA-PUF with arbitrary number of paths

In the previous section, we observed that our model of a PA-PUF with 4-paths generates all Boolean functions involving 3 variables. This observation motivates us to investigate further. In this section, we generalize our new model of PA-PUF with arbitrary number of paths r, denoted by $Pt_1, Pt_2, \ldots, Pt_r$. In the model of classical PUF with 2 paths, the signal either follows the same path or paths are altered depending upon the challenge input to the switch. Recall that the path alternation mechanism for two paths Pt_1, Pt_2, the alternation is done based on the permutation $\begin{pmatrix} Pt_1 & Pt_2 \\ Pt_2 & Pt_1 \end{pmatrix}$. We extend this concept of path following mechanism in the case of r number of paths in every switch. The signal in every path of every switch will either follow same path or it changes its path based on the challenge input to the switch. The path alternation will be done based as per the Fig. 5. We use the following (cycle of order r) permutation $\sigma_r : \begin{pmatrix} Pt_1 & Pt_2 & \cdots & Pt_r \\ Pt_r & Pt_1 & \cdots & Pt_{r-1} \end{pmatrix}$ to compute the delay in all paths when there is an alteration of path of the pulse.

We measure the delay at every i-th switch using the iterative relations described in Eqs. (9) and (10) in any two paths (say Pt_a and Pt_b) as $\delta_{Pt_a}(i)$ and $\delta_{Pt_b}(i)$ for any Path a and Path b, respectively. Here $dl_i^{(Pt_a Pt_b)}$ denotes the introduced delay at the i-th switch, when the signal moves from the path Pt_a to Pt_b,

$$\begin{aligned} \delta_{Pt_a}(i) = &\left(\frac{C_i + 1}{2}\right)\left(\delta_{Pt_a}(i-1) + dl_i^{(Pt_a Pt_a)}\right) \\ &+ \left(\frac{1 - C_i}{2}\right)\left(\delta_{\sigma_r(Pt_a)}(i-1) + dl_i^{(\sigma_r(Pt_a) Pt_a)}\right) \end{aligned} \tag{9}$$

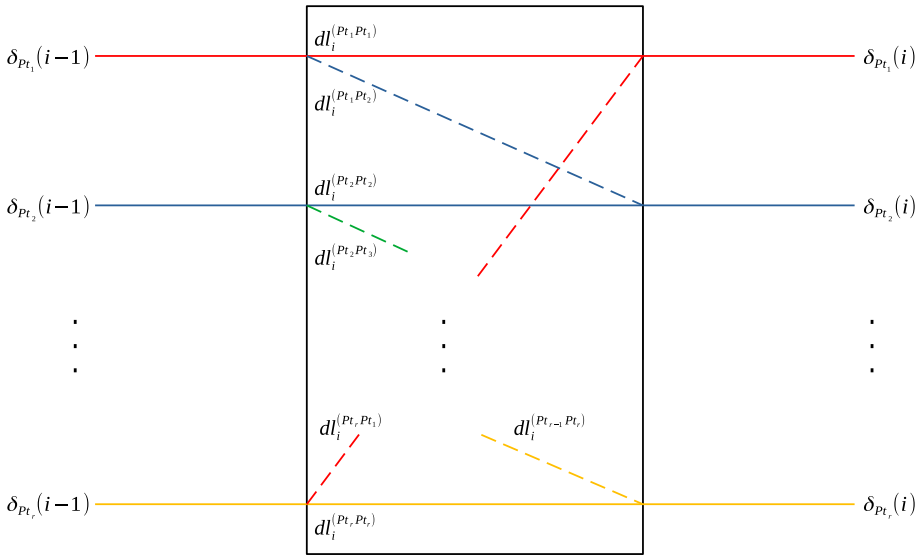

Fig. 5 Path alteration in an r-path PA-PUF

$$\delta_{Pt_b}(i) = \left(\frac{C_i + 1}{2}\right)\left(\delta_{Pt_b}(i-1) + dl_i^{(Pt_b Pt_b)}\right) + \left(\frac{1 - C_i}{2}\right)\left(\delta_{\sigma_r(Pt_b)}(i-1) + dl_i^{(\sigma_r(Pt_b) Pt_b)}\right). \tag{10}$$

The delay difference between any two paths Pt_a and Pt_b at the i-th switch will be the difference between Eqs. (9) and (10) as follows:

$$\Delta^i_{Pt_a Pt_b}(C^{i+1}) = \left(\frac{C_i + 1}{2}\right)\left(\Delta^{i-1}_{Pt_a Pt_b}(C^i) + dl_i^{(Pt_a Pt_a)} - dl_i^{(Pt_b Pt_b)}\right) + \left(\frac{1 - C_i}{2}\right)\left(\Delta^{i-1}_{\sigma_r(Pt_a)\sigma_r(Pt_b)}(C^i) + dl_i^{(\sigma_r(Pt_a) Pt_a)} - dl_i^{(\sigma_r(Pt_b) Pt_b)}\right). \tag{11}$$

In the case of a 2-path PUF, the arbiter outputs 1 if the signal reaches first in the top path and it outputs 0 if the signal reaches first in the bottom path. In our model of n-length PA-PUF (n number of switches) with r number of paths we follow the following two strategies for determining the output.

1. If the number of paths r is even, that is, $r = 2k$, then the condition for arbiter to produce 1 as output is when the signal reaches first in any one of the top $\frac{r}{2} = k$ paths. It outputs 0 if signal reaches first in any one of the bottom $\frac{r}{2} = k$ paths. In terms of sign of delay differences we follow the following, to decide the output from the PA-PUF:
   - If $\Delta^{n-1}_{Pt_1 Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_k Pt_X}(C^n) < 0$ then the output from the PA-PUF is 1.
   - If $\Delta^{n-1}_{Pt_{k+1} Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n) < 0$ then the output from the PA-PUF is 0.

2. If the number of paths r is odd, that is, $r = 2k + 1$, then the condition for the arbiter to produce 1 as output is when the signal reaches first in any one of the top $\frac{r-1}{2} = k$ paths, or if the signal reaches first in the bottom path along with $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) < 0$. Similarly, the arbiter outputs 0 if the signal reaches first in any one of the bottom $\frac{r-1}{2} = k$ paths, or if the signal reaches first in the bottom path along with $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) > 0$. This decision of producing output can also be expressed as the sign of the delay differences, which is described below:
   - If $\Delta^{n-1}_{Pt_1 Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_k Pt_X}(C^n) < 0$ (or) $(\Delta^{n-1}_{Pt_{2k+1} Pt_X}(C^n) < 0$ and $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) < 0)$ then the output is 1.
   - If $\Delta^{n-1}_{Pt_{k+1} Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n) < 0$ (or) $(\Delta^{n-1}_{Pt_{2k+1} Pt_X}(C^n) < 0$ and $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) > 0)$ then the output is 0.

Here, $\Delta^{n-1}_{Pt_j Pt_X}(C^n)$ represents the general form of the delay difference between the path Pt_j and the path Pt_X where $Pt_X \in \{Pt_1, \ldots, Pt_{j-1}, Pt_{j+1}, \ldots, Pt_r\}$.

Now, we are in a position to investigate the class $\mathcal{B}^{\mathsf{PA-PUF}}_{i,j}$, for arbitrary i, j. In [16], Roy et al. showed that the Boolean function $f = (0, 1, 0, 1) = x_0 \notin \mathcal{B}^{\mathsf{PUF}}$. In terms of our generalized notation, $\mathcal{B}^{\mathsf{PA-PUF}}_{2,2} = \mathcal{B}^{\mathsf{PUF}}$, it means that $f = (0, 1, 0, 1) = x_0 \notin \mathcal{B}^{\mathsf{PA-PUF}}_{2,2}$. It is to be noted that the variable x_0 corresponds to the challenge input C_0 to the PA-PUF. In the following theorem, we can find a more general function, which does not belong to the set $\mathcal{B}^{\mathsf{PA-PUF}}_{i,i}$. We start with a few lemmas.

Lemma 1 *At least one delay parameter in $\Delta^{i-1}_{Pt_j Pt_l}(C^i)$ is not present in the expression of $\Delta^{i-1}_{Pt_k Pt_l}(C^i)$ for $j \neq k$ when we have at least $(i + 1)$ paths.*

Proof Let us focus on the equations of $\Delta^{i-1}_{Pt_j Pt_l}(C^i)$ and $\Delta^{i-1}_{Pt_k Pt_l}(C^i)$,

$$\begin{aligned}
\Delta^{i-1}_{Pt_j Pt_l}(C^i) &= \left(\frac{C_{i-1}+1}{2}\right)\left(\Delta^{i-2}_{Pt_j Pt_l}(C^{i-1}) + dl^{(Pt_j Pt_j)}_{i-1} - dl^{(Pt_l Pt_l)}_{i-1}\right) \\
&\quad + \left(\frac{1-C_{i-1}}{2}\right)\left(\Delta^{i-2}_{\sigma_r(Pt_j)\sigma_r(Pt_l)}(C^{i-1}) + dl^{(\sigma_r(Pt_j)Pt_j)}_{i-1} - dl^{(\sigma_r(Pt_l)Pt_l)}_{i-1}\right), \\
\Delta^{i-1}_{Pt_k Pt_l}(C^i) &= \left(\frac{C_{i-1}+1}{2}\right)\left(\Delta^{i-2}_{Pt_k Pt_l}(C^{i-1}) + dl^{(Pt_k Pt_k)}_{i-1} - dl^{(Pt_l Pt_l)}_{i-1}\right) \\
&\quad + \left(\frac{1-C_{i-1}}{2}\right)\left(\Delta^{i-2}_{\sigma_r(Pt_k)\sigma_r(Pt_l)}(C^{i-1}) + dl^{(\sigma_r(Pt_k)Pt_k)}_{i-1} - dl^{(\sigma_r(Pt_l)Pt_l)}_{i-1}\right).
\end{aligned}$$

For $C_{i-1} = 1$, if we look into the expressions of $\Delta^{i-1}_{Pt_j Pt_l}(C^i)$ and $\Delta^{i-1}_{Pt_k Pt_l}(C^i)$, the delay parameters $dl^{(Pt_j Pt_j)}_{i-1}$ and $dl^{(Pt_k Pt_k)}_{i-1}$ are present in $\Delta^{i-1}_{Pt_j Pt_l}(C^i)$ and $\Delta^{i+1}_{Pt_k Pt_l}(C^i)$, respectively. Similarly for $C_{i-1} = -1$, the delay parameters $dl^{(\sigma_r(Pt_j)Pt_j)}_{i-1}$ and $dl^{(\sigma_r(Pt_k)Pt_k)}_{i-1}$ are present in $\Delta^{i-1}_{Pt_j Pt_l}(C^i)$ and $\Delta^{i-1}_{Pt_k Pt_l}(C^i)$, respectively. Thus, there is at least one delay parameter in the expression of $\Delta^{i-1}_{Pt_j Pt_l}(C^i)$ which is not present in $\Delta^{i-1}_{Pt_k Pt_l}(C^i)$ for $j \neq k$. □

Lemma 2 *At least one delay parameter in $\Delta^{i}_{Pt_j Pt_l}(C^{i+1})$ is not present in the expression of $\Delta^{i}_{Pt_j Pt_l}(\widetilde{C}^{i+1})$.*

Table 4 Conditions when $f_3 = x_0 + x_2 \in \mathcal{B}_{3,3}^{\mathsf{PA-PUF}}$

C_2	C_1	C_0	$\Delta^2_{Pt_jPt_k}(C^3)$
1	1	1	$\Delta^2_{Pt_2X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) > 0$
1	1	−1	$\Delta^2_{Pt_1X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) < 0$
1	−1	1	$\Delta^2_{Pt_2X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) > 0$
1	−1	−1	$\Delta^2_{Pt_1X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) < 0$
−1	1	1	$\Delta^2_{Pt_1X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) < 0$
−1	1	−1	$\Delta^2_{Pt_2X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) > 0$
−1	−1	1	$\Delta^2_{Pt_1X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) < 0$
−1	−1	−1	$\Delta^2_{Pt_2X}(C^3) < 0$ (or) $\Delta^2_{Pt_3X}(C^3) < 0$ (&) $\Delta^2_{Pt_1Pt_2}(C^3) > 0$

Proof Let's consider the equations of $\Delta^i_{Pt_jPt_l}(C^{i+1})$ and $\Delta^i_{Pt_kPt_l}(C^{i+1})$,

$$\Delta^{i-1}_{Pt_jPt_l}(C^i) = \left(\frac{C_{i-1}+1}{2}\right)\left(\Delta^{i-2}_{Pt_jPt_l}(C^{i-1}) + dl_{i-1}^{(Pt_jPt_j)} - dl_{i-1}^{(Pt_lPt_l)}\right) + \left(\frac{1-C_{i-1}}{2}\right)\left(\Delta^{i-2}_{\sigma_r(Pt_j)\sigma_r(Pt_l)}(C^{i-1}) + dl_{i-1}^{(\sigma_r(Pt_j)Pt_j)} - dl_{i-1}^{(\sigma_r(Pt_l)Pt_l)}\right),$$

$$\Delta^{i-1}_{Pt_jPt_l}(\widetilde{C}^i) = \left(\frac{\widetilde{C}_{i-1}+1}{2}\right)\left(\Delta^{i-2}_{Pt_jPt_l}(\widetilde{C}^{i-1}) + dl_{i-1}^{(Pt_jPt_j)} - dl_{i-1}^{(Pt_lPt_l)}\right) + \left(\frac{1-\widetilde{C}_{i-1}}{2}\right)\left(\Delta^{i-2}_{\sigma_r(Pt_j)\sigma_r(Pt_l)}(\widetilde{C}^{i-1}) + dl_{i-1}^{(\sigma_r(Pt_j)Pt_j)} - dl_{i-1}^{(\sigma_r(Pt_l)Pt_l)}\right).$$

If $C_{i-1} \neq \widetilde{C}_{i-1}$, then both delay parameters present in $\Delta^i_{Pt_jPt_l}(C^i)$ will be not present in $\Delta^i_{Pt_jPt_l}(\widetilde{C}^i)$. If $C_{i-1} = \widetilde{C}_{i-1}$, we continue the iterative expansion until $C_k \neq \widetilde{C}_k$, where $k \in \{i-2, \ldots, 0\}$ since $C^i \neq \widetilde{C}^i$ in a particular table of conditions. Thus, there is at least one delay parameter in the expression of $\Delta^i_{Pt_jPt_l}(C^i)$, which is not present in $\Delta^i_{Pt_jPt_l}(\widetilde{C}^i)$. □

Theorem 1 *The function f_i of the form $f_i = \begin{cases} x_0 & i = 2 \\ x_0 + x_2 + \cdots + x_{i-1} & i \geq 3 \end{cases}$ does not belong to $\mathcal{B}_{i,i}^{\mathsf{PA-PUF}}$.*

Proof The first case $f_1 = x_0 \notin \mathcal{B}_{2,2}^{\mathsf{PA-PUF}}$ follows from the result in [16], which we have discussed just before this theorem. We will be using mathematical induction for proving $f_i \notin \mathcal{B}_{i,i}^{\mathsf{PA-PUF}}$ for $i \geq 3$. Let us assume that $f_3 = x_0 + x_2 \in \mathcal{B}_{3,3}^{\mathsf{PA-PUF}}$, then the conditions in Table 4 must have a solution for the involved delay parameters.

Here, $\Delta^2_{Pt_1Pt_2}(C^3)\big|_{(1,C_1,C_0)}$ can be expressed as $\Delta^1_{Pt_1Pt_2}(C^2)\big|_{(C_1,C_0)} + dl_2^{(Pt_1Pt_1)} - dl_2^{(Pt_2Pt_2)}$. Since $f_3 \in \mathcal{B}_{3,3}^{\mathsf{PA-PUF}}$, there exists a solution to the above system of equations, say with

$$\Delta^1_{Pt_1Pt_2}(C^2)\big|_{(1,1)} > dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)},$$
$$\Delta^1_{Pt_1Pt_2}(C^2)\big|_{(1,-1)} < dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)},$$

Table 5 Conditions when $f_2 = x_0 \in \mathcal{B}_{2,2}^{\mathsf{PA-PUF}}$

C_1	C_0	$\Delta^1_{Pt_1Pt_2}(C^2)$ Conditions
1	1	$\Delta^1_{Pt_1Pt_2}(C^2) > 0$
1	−1	$\Delta^1_{Pt_1Pt_2}(C^2) < 0$
−1	1	$\Delta^1_{Pt_1Pt_2}(C^2) > 0$
−1	−1	$\Delta^1_{Pt_1Pt_2}(C^2) < 0$

$$\Delta^1_{Pt_1Pt_2}(C^2)\big|_{(-1,1)} > dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)},$$
$$\Delta^1_{Pt_1Pt_2}(C^2)\big|_{(-1,-1)} < dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)}.$$

If we expand each $\Delta^1_{Pt_1Pt_2}(C^2)\big|_{(C_1,C_0)}$, we get the following conditions

$$\begin{aligned}
dl_0^{(Pt_1Pt_1)} - dl_0^{(Pt_2Pt_2)} + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)} &> dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)},\\
dl_0^{(Pt_3Pt_1)} - dl_0^{(Pt_1Pt_2)} + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)} &< dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)},\\
dl_0^{(Pt_3Pt_3)} - dl_0^{(Pt_1Pt_1)} + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)} &> dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)},\\
dl_0^{(Pt_2Pt_3)} - dl_0^{(Pt_3Pt_1)} + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)} &< dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)}.
\end{aligned}$$

This can be re-written as

$$\begin{aligned}
dl_0^{(Pt_1Pt_1)} - (dl_0^{(Pt_2Pt_2)} + dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)}) + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)} &> 0,\\
dl_0^{(Pt_3Pt_1)} - (dl_0^{(Pt_1Pt_2)} + dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)}) + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)} &< 0,\\
(dl_0^{(Pt_3Pt_3)} - dl_2^{(Pt_2Pt_2)} + dl_2^{(Pt_1Pt_1)}) - dl_0^{(Pt_1Pt_1)} + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)} &> 0,\\
(dl_0^{(Pt_2Pt_3)} - dl_2^{(Pt_2Pt_2)} + dl_2^{(Pt_1Pt_1)}) - dl_0^{(Pt_3Pt_1)} + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)} &< 0.
\end{aligned}$$

Further using the substitutions $(dl_0^{(Pt_2Pt_2)} + dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)}) = \widetilde{dl}_0^{(Pt_2Pt_2)}$, $(dl_0^{(Pt_1Pt_2)} + dl_2^{(Pt_2Pt_2)} - dl_2^{(Pt_1Pt_1)}) = \widetilde{dl}_0^{(Pt_1Pt_2)}$, $(dl_0^{(Pt_3Pt_3)} - dl_2^{(Pt_2Pt_2)} + dl_2^{(Pt_1Pt_1)}) = \widetilde{dl}_0^{(Pt_3Pt_3)}$ and $(dl_0^{(Pt_2Pt_3)} - dl_2^{(Pt_2Pt_2)} + dl_2^{(Pt_1Pt_1)}) = \widetilde{dl}_0^{(Pt_2Pt_3)}$ we get the following simplified forms,

$$dl_0^{(Pt_1Pt_1)} - \widetilde{dl}_0^{(Pt_2Pt_2)} + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)} > 0, \tag{12}$$
$$dl_0^{(Pt_3Pt_1)} - \widetilde{dl}_0^{(Pt_1Pt_2)} + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)} < 0, \tag{13}$$
$$\widetilde{dl}_0^{(Pt_3Pt_3)} - dl_0^{(Pt_1Pt_1)} + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)} > 0, \tag{14}$$
$$\widetilde{dl}_0^{(Pt_2Pt_3)} - dl_0^{(Pt_3Pt_1)} + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)} < 0. \tag{15}$$

These four inequalities given in Eqs. (12), (13) (14) and (15) are equivalent to the system of equations corresponding to $\Delta^1_{Pt_1Pt_2}(C^2)$, which are given in Table 5.

We will use the solution to (12), (13) (14) and (15) to form the solution of the inequalities of Table 5. If the solution of inequalities of Table 5 exists then it contradicts the fact that $f_2 = x_0 \notin \mathcal{B}_{2,2}^{\mathsf{PA-PUF}}$ (proven in [16]) as the inequalities of Table 5 corresponds to the PUF which generates $f_2 = x_0$. Thus, $f_3 = x_0 + x_2 \notin \mathcal{B}_{3,3}^{\mathsf{PA-PUF}}$. This proves the base step of our mathematical induction process.

Now, we assume that the result is true for $i = n$, that is, $f_n = x_0 + x_2 + \cdots + x_{n-1} \notin \mathcal{B}_{n,n}^{\mathsf{PA-PUF}}$. With this assumption we need to prove that for $i = n+1$, $f_{n+1} = x_0 + x_2 + \cdots + x_n \notin$

Table 6 Conditions for $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$

C_n	...	C_0	$\Delta^n_{Pt_j Pt_k}(C^{n+1})$
1	...	1	$\Delta^n_{Pt_{k+1} Pt_X}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{Pt_{2k} Pt_X}(C^{n+1}) < 0$
⋮	⋮	⋮	⋮
−1	...	−1	$\Delta^n_{Pt_1 Pt_X}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{Pt_k Pt_X}(C^{n+1}) < 0$

Table 7 Conditions when $f_n \in \mathcal{B}^{\mathsf{PA-PUF}}_{n,n}$

C_{n-1}	...	C_0	$\Delta^{n-1}_{Pt_j Pt_k}(C^n)$
1	...	1	$\Delta^{n-1}_{Pt_{k-1} Pt_X}(C^n) < 0$ (or) ... (or) $\Delta^{n-1}_{Pt_{2k-2} Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) > 0$
⋮	⋮	⋮	⋮
−1	...	1	$\Delta^{n-1}_{Pt_1 Pt_X}(C^n) < 0$ (or) ... (or) $\Delta^{n-1}_{Pt_{k-2} Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) < 0$
−1	...	−1	$\Delta^{n-1}_{Pt_{k-1} Pt_X}(C^n) < 0$ (or) ... (or) $\Delta^{n-1}_{Pt_{2k-2} Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) > 0$

$\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$. Let us assume that $f_{n+1} = x_0 + x_2 + \cdots + x_n \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$. Here, there will be two cases depending on the parity of $n+1$.

Case I. Let $n+1$ be an even number, $n+1 = 2k$. If $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$, then the conditions in Table 6 must have a solution corresponding to the delay parameters.

Here, $\Delta^n_{Pt_j Pt_X}(C^{n+1})\big|_{(1,C_{n-1},\ldots,C_0)}$ can be expressed as $\Delta^{n-1}_{Pt_j Pt_X}(C^n)\big|_{(C_{n-1},\ldots,C_0)} + dl_n^{(Pt_j Pt_j)} - dl_n^{(Pt_X Pt_X)}$. Since $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$, the solution to the above system of equation exists, say such as $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)|_{(1,\ldots,1)} < dl_n^{(Pt_X Pt_X)} - dl_n^{(Pt_1 Pt_1)}$, ..., $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n)|_{(-1,\ldots,-1)} < dl_n^{Pt_X Pt_X} - dl_n^{(Pt_{2k-1} Pt_{2k-1})}$.

From the first 2^n conditions, we consider the inequalities of $\Delta^{n-1}_{Pt_1 Pt_X}(C^n), \ldots, \Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n)$. From Lemmas 1, and 2 we know that if we consider any two expressions among $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)$, ..., $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n)$, then there exists at least one delay parameter in each of these expressions, which is not present in the other expressions. We introduce a new variable corresponding to each subtraction between this independent delay parameter and the terms in the right hand side of the above inequalities $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)|_{(1,\ldots,1)} < dl_n^{(Pt_X Pt_X)} - dl_n^{(Pt_1 Pt_1)}$, ..., $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n)|_{(-1,\ldots,-1)} < dl_n^{Pt_X Pt_X} - dl_n^{(Pt_{2k-1} Pt_{2k-1})}$. For example, the expression $dl_{i-1}^{(Pt_j Pt_j)} - dl_n^{(Pt_j Pt_j)} + dl_n^{Pt_X Pt_X}$ will be replaced by $\widetilde{dl}_{i-1}^{(Pt_j Pt_j)}$, here $dl_{i-1}^{(Pt_j Pt_j)}$ is the independent delay parameter and $(dl_n^{(Pt_j Pt_j)} - dl_n^{(Pt_X Pt_X)})$ is the term in the right hand side of an inequality. Then using these newly introduced variables we form a new set of inequalities say Eq_1. This new system of inequalities Eq_1 is similar to the system of inequalities presented in Table 7.

Table 8 Conditions when $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$

C_n	...	C_0	$\Delta^n_{Pt_j Pt_k}(C^{n+1})$
1	...	1	$\Delta^n_{Pt_{k+1} Pt_X}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{Pt_{2k} Pt_X}(C^{n+1}) < 0$ (or) $\Delta^n_{Pt_{2k+1} Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1 Pt_2}(C^{n+1}) > 0$
⋮	⋮	⋮	⋮
-1	...	1	$\Delta^n_{Pt_1 Pt_X}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{Pt_k Pt_X}(C^{n+1}) < 0$ (or) $\Delta^n_{Pt_{2k+1} Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1 Pt_2}(C^{n+1}) < 0$
-1	...	-1	$\Delta^n_{Pt_{k+1} Pt_X}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{Pt_{2k} Pt_X}(C^{n+1}) < 0$ (or) $\Delta^n_{Pt_{2k+1} Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1 Pt_2}(C^{n+1}) > 0$

Table 9 Conditions when $f_n \in \mathcal{B}^{\mathsf{PA-PUF}}_{n,n}$

C_{n-1}	...	C_0	$\Delta^{n-1}_{Pt_j Pt_k}(C^n)$ Conditions
1	...	1	$\Delta^{n-1}_{Pt_{k+1} Pt_X}(C^n) < 0$ (or) ... (or) $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n) < 0$
⋮	⋮	⋮	⋮
-1	...	-1	$\Delta^{n-1}_{Pt_1 Pt_X}(C^n) < 0$ (or) ... (or) $\Delta^{n-1}_{Pt_k Pt_X}(C^n) < 0$

Now, the system of inequalities Eq_1 will have a solution as $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$ and Eq_1 is constructed from the inequalities of Table 6. As Eq_1 has a solution, the system presented in Table 7 will also have a solution as the structure of the equations are exactly the same. This proves the existence of a Boolean function $x_0 + x_2 + \cdots + x_{n-1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n,n}$, which is a contradiction to our assumption. Thus, $f_{n+1} = x_0 + x_2 + \cdots + x_n \notin \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$.

Case II. Let $n+1$ be an odd number, $n+1 = 2k+1$. If $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$, then the conditions in Table 8 must have a solution for the delay parameters.

Here $\Delta^n_{Pt_j Pt_X}(C^{n+1})\big|_{(1,C_{n-1},\ldots,C_0)}$ can be expressed as $\Delta^{n-1}_{Pt_j Pt_X}(C^n)\big|_{(C_{n-1},\ldots,C_0)} + dl_n^{(Pt_j Pt_j)} - dl_n^{(Pt_X Pt_X)}$. As $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$, the solution to the above system of inequalities exists. Let the solution of the above system of inequalities be of the form $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)|_{(1,\ldots,1)} < dl_n^{(Pt_X Pt_X)} - dl_n^{(Pt_1 Pt_1)}, \ldots, \Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n)|_{(-1,\ldots,-1)} < dl_n^{Pt_X Pt_X} - dl_n^{(Pt_{2k} Pt_{2k})}$.

From the first 2^n conditions, we consider the inequalities of $\Delta^{n-1}_{Pt_1 Pt_X}(C^n), \ldots, \Delta^{n-1}_{Pt_{2k} Pt_X}(C^n)$. From Lemmas 1 and 2, we know that there exists a delay parameter in the expression of $\Delta^{n-1}_{Pt_i Pt_X}(C^n)$ which is not present in the expression of $\Delta^{n-1}_{Pt_j Pt_X}(C^n)$, for $i \neq j$. We will consider these independent variables to introduce new variables in each inequality $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)|_{(1,\ldots,1)} < dl_n^{(Pt_X Pt_X)} - dl_n^{(Pt_1 Pt_1)}$, ..., $\Delta^{n-1}_{Pt_{2k-1} Pt_X}(C^n)|_{(-1,\ldots,-1)} < dl_n^{(Pt_X Pt_X)} - dl_n^{(Pt_{2k} Pt_{2k})}$. For example, $dl_{i-1}^{(Pt_j Pt_j)} - dl_n^{(Pt_j Pt_j)} + dl_n^{(Pt_X Pt_X)}$ will be replaced by the new variable $\widetilde{dl}_{i-1}^{(Pt_j Pt_j)}$. Let the new system of inequalities be denoted by Eq_2. The system of inequalities Eq_2 is similar to the system of inequalities presented in Table 9.

Table 10 Conditions when $f_3 = x_0 + x_2 \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$

C_2	C_1	C_0	$\Delta^2_{Pt_jPt_k}(C^3)$
1	1	1	$\Delta^2_{Pt_3Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4Pt_X}(C^3) < 0$
1	1	−1	$\Delta^2_{Pt_1Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_2Pt_X}(C^3) < 0$
1	−1	1	$\Delta^2_{Pt_3Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4Pt_X}(C^3) < 0$
1	−1	−1	$\Delta^2_{Pt_1Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_2Pt_X}(C^3) < 0$
−1	1	1	$\Delta^2_{Pt_1Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_2Pt_X}(C^3) < 0$
−1	1	−1	$\Delta^2_{Pt_3Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4Pt_X}(C^3) < 0$
−1	−1	1	$\Delta^2_{Pt_1Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_2Pt_X}(C^3) < 0$
−1	−1	−1	$\Delta^2_{Pt_3Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4Pt_X}(C^3) < 0$

Table 11 Conditions when $f_2 = x_0 \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$

C_1	C_0	$\Delta^1_{Pt_jPt_k}(C^2)$
1	1	$\Delta^1_{Pt_2X}(C^2) < 0$ (or) $\Delta^1_{Pt_3X}(C^2) < 0$ (&) $\Delta^1_{Pt_1Pt_2}(C^2) > 0$
1	−1	$\Delta^1_{Pt_1X}(C^2) < 0$ (or) $\Delta^1_{Pt_3X}(C^2) < 0$ (&) $\Delta^1_{Pt_1Pt_2}(C^2) < 0$
−1	1	$\Delta^1_{Pt_2X}(C^2) < 0$ (or) $\Delta^1_{Pt_3X}(C^2) < 0$ (&) $\Delta^1_{Pt_1Pt_2}(C^2) > 0$
−1	−1	$\Delta^1_{Pt_1X}(C^2) < 0$ (or) $\Delta^1_{Pt_3X}(C^2) < 0$ (&) $\Delta^1_{Pt_1Pt_2}(C^2) < 0$

As the solution of the inequalities presented in Table 8 exists and Eq_2 is constructed from the inequalities of Table 8 only, there will be a solution of the inequalities of Table 9. The existence of this solution ensures that $x_0+x_2+\cdots+x_{n-1} \in \mathcal{B}_{n,n}^{\mathsf{PA-PUF}}$, which is a contradiction to our assumption. Thus, $f_{n+1} = x_0 + x_2 + \cdots + x_n \notin \mathcal{B}_{n+1,n+1}^{\mathsf{PA-PUF}}$. □

In [8] it has been shown that the Boolean function $f_3 = x_0 \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$. We also observed similar result in our model of PA-PUF. Now we are interested in extending this result to $\mathcal{B}_{i+1,i}^{\mathsf{PA-PUF}}$.

Theorem 2 *The function* f_i *of the form* $f_i = \begin{cases} x_0 & i = 2 \\ x_0 + x_2 + \cdots + x_{i-1} & i \geq 3 \end{cases}$ *belongs to* $\mathcal{B}_{i+1,i}^{\mathsf{PA-PUF}}$.

Proof We prove that $f_i \in \mathcal{B}_{i+1,i}^{\mathsf{PA-PUF}}$ using mathematical induction. The case when $i = 2$ is already addressed in [8]. To prove the statement for $i = 3$, that is, $f_3 = x_0 + x_2 \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$, we construct a solution of the system of inequalities in Table 10 using the solution to the system of inequalities of Tables 11 and 12.

The set of inequalities involving $\Delta^2_{Pt_1Pt_X}(C^3)\big|_{(1,C_1,C_0)}$, $\Delta^2_{Pt_3Pt_X}(C^3)\big|_{(1,C_1,C_0)}$ presented in Table 10 can be expressed as $\Delta^1_{Pt_1Pt_X}(C^2)\big|_{(C_1,C_0)} + dl_2^{(Pt_1Pt_1)} - dl_2^{(Pt_XPt_X)}$ and $\Delta^1_{Pt_3Pt_X}(C^2)\big|_{(C_1,C_0)} + dl_2^{(Pt_3Pt_3)} - dl_2^{(Pt_XPt_X)}$ respectively. We use the inequalities of $\Delta^1_{Pt_1Pt_X}(C^2)$ and $\Delta^1_{Pt_3Pt_X}(C^2)$ presented in Tables 11 and 12 for $\Delta^2_{Pt_1Pt_X}(C^3)$, $\Delta^2_{Pt_3Pt_X}(C^3)$ presented in Table 10. For $C_2 = 1$ we take the inequalities from Table 11 and for $C_2 = -1$

Table 12 Conditions when $f_2 + 1 = x_0 + 1 \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$

C_1	C_0	$\Delta^1_{P_{t_j}P_{t_k}}(C^2)$
1	1	$\Delta^1_{P_{t_1}X}(C^2) < 0$ (or) $\Delta^1_{P_{t_3}X}(C^2) < 0$ (&) $\Delta^1_{P_{t_1}P_{t_2}}(C^2) < 0$
1	−1	$\Delta^1_{P_{t_2}X}(C^2) < 0$ (or) $\Delta^1_{P_{t_3}X}(C^2) < 0$ (&) $\Delta^1_{P_{t_1}P_{t_2}}(C^2) > 0$
−1	1	$\Delta^1_{P_{t_1}X}(C^2) < 0$ (or) $\Delta^1_{P_{t_3}X}(C^2) < 0$ (&) $\Delta^1_{P_{t_1}P_{t_2}}(C^2) < 0$
−1	−1	$\Delta^1_{P_{t_2}X}(C^2) < 0$ (or) $\Delta^1_{P_{t_3}X}(C^2) < 0$ (&) $\Delta^1_{P_{t_1}P_{t_2}}(C^2) > 0$

Table 13 Conditions when $f_{n+1} \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$

C_n	...	C_0	$\Delta^n_{P_{t_j}P_{t_k}}(C^{n+1})$
1	...	1	$\Delta^n_{P_{t_{k+1}}P_{t_X}}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{P_{t_{2k}}P_{t_X}}(C^{n+1}) < 0$
⋮	⋮	⋮	⋮
−1	...	−1	$\Delta^n_{P_{t_1}P_{t_X}}(C^{n+1}) < 0$ (or) ... (or) $\Delta^n_{P_{t_k}P_{t_X}}(C^{n+1}) < 0$

we take the inequalities from Table 12. We also choose the delay parameters $dl_2^{(P_{t_4}P_{t_4})}$ and $dl_2^{(P_{t_3}P_{t_4})}$ in such a way that we get $\delta_{P_{t_4}}(2) > \max\{\delta_{P_{t_1}}(2), \delta_{P_{t_2}}(2), \delta_{P_{t_3}}(2)\}$. This helps us achieve $\Delta^2_{P_{t_i}P_{t_4}}(C^3) < 0$, for $i = 1, 2, 3$. We also choose $dl_2^{(P_{t_1}P_{t_1})} \approx dl_2^{(P_{t_2}P_{t_2})} \approx dl_2^{(P_{t_3}P_{t_3})} \approx dl_2^{(P_{t_4}P_{t_4})}$ and $dl_2^{(P_{t_4}P_{t_1})} \approx dl_2^{(P_{t_1}P_{t_2})} \approx dl_2^{(P_{t_2}P_{t_3})} \approx dl_2^{(P_{t_1}P_{t_4})}$ so that the first four inequalities of Table 10 and the inequalities of Table 11 have the same inequality sign, as well as the last four inequalities of Table 10 and the inequalities of Table 12 have the same inequality sign.

The system of inequalities corresponding to Tables 11 and 12 has a solution corresponding to the involved delay parameters as f_2, $f_2 + 1 \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$. Using this, we are therefore able to form a solution corresponding to the involved delay parameters involved in the inequalities of Table 10. This shows that $f_3 = x_0 + x_2 \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$.

Now, we assume that our statement is true for $i = n$, that is, $f_n = x_0 + x_2 + \cdots + x_{n-1} \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$, and we will show that the statement is true for $i = n + 1$, that is, $f_{n+1} = x_0 + x_2 + \cdots + x_n \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$. To prove this we construct a solution corresponding to the system of inequalities when $f_{n+1} \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$ using the solution of the system of inequalities when $f_n \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$ and $f_n + 1 \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$. One can note that from the solutions of the delays corresponding to the system of equations related to $f_n \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$ we can easily produce the solutions corresponding to $f_n + 1$ by taking the negative of all the delays this gives guarantee that $f_n + 1 \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$. Here, there will be two cases depending upon the parity of $n + 1$.

Case I. Let $n + 1$ be an even number, $n + 1 = 2k$.

Here, $\Delta^n_{P_{t_j}P_{t_X}}(C^{n+1})\big|_{(1,C_{n-1},\ldots,C_0)}$ can be expressed as $\Delta^{n-1}_{P_{t_j}P_{t_X}}(C^n)\big|_{(C_{n-1},\ldots,C_0)} + dl_n^{(P_{t_j}P_{t_j})} - dl_n^{(P_{t_X}P_{t_X})}$. We note that the inequalities involving $\Delta^n_{P_{t_1}P_{t_X}}(C^{n+1})$, ..., $\Delta^n_{P_{t_{2k-2}}P_{t_X}}(C^{n+1})$ of Table 13 can be constructed using $\Delta^{n-1}_{P_{t_1}P_{t_X}}(C^n)$, ..., $\Delta^{n-1}_{P_{t_{2k-2}}P_{t_X}}(C^n)$ of Tables 14 and 15. For $C_n = 1$, that is, the first 2^n conditions, we consider $\Delta^{n-1}_{P_{t_1}P_{t_X}}(C^n), \ldots, \Delta^{n-1}_{P_{t_{2k-2}}P_{t_X}}(C^n)$ from Table 14. For $C_n = -1$, that is, the next

Table 14 Conditions when $f_n \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n}$

C_{n-1}	$\ldots$	C_0	$\Delta^{n-1}_{Pt_jPt_k}(C^n)$
1	$\ldots$	1	$\Delta^{n-1}_{Pt_{k-1}Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{2k-2}Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1Pt_2}(C^n) > 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\ldots$	1	$\Delta^{n-1}_{Pt_1Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{k-2}Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1Pt_2}(C^n) < 0$
-1	$\ldots$	-1	$\Delta^{n-1}_{Pt_{k-1}Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{2k-2}Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1Pt_2}(C^n) > 0$

Table 15 Conditions when $f_n + 1 \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n}$

C_{n-1}	$\ldots$	C_0	$\Delta^{n-1}_{Pt_jPt_k}(C^n)$
1	$\ldots$	1	$\Delta^{n-1}_{Pt_1Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{k-2}Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1Pt_2}(C^n) < 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\ldots$	1	$\Delta^{n-1}_{Pt_{k-1}Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{2k-2}Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1Pt_2}(C^n) > 0$
-1	$\ldots$	-1	$\Delta^{n-1}_{Pt_1Pt_X}(C^n) < 0$ (or) $\ldots$ (or) $\Delta^{n-1}_{Pt_{k-2}Pt_X}(C^n) < 0$ (or) $\Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) < 0$ (&) $\Delta^{n-1}_{Pt_1Pt_2}(C^n) < 0$

Table 16 Conditions when $f_{n+1} \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+2,n+1}$

C_n	$\ldots$	C_0	$\Delta^{n}_{Pt_jPt_k}(C^{n+1})$
1	$\ldots$	1	$\Delta^{n}_{Pt_{k+1}Pt_X}(C^{n+1}) < 0$ (or) $\ldots$ (or) $\Delta^{n}_{Pt_{2k}Pt_X}(C^{n+1}) < 0$ (or) $\Delta^{n}_{Pt_{2k+1}Pt_X}(C^{n+1}) < 0$ (&) $\Delta^{n}_{Pt_1Pt_2}(C^{n+1}) > 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\ldots$	1	$\Delta^{n}_{Pt_1Pt_X}(C^{n+1}) < 0$ (or) $\ldots$ (or) $\Delta^{n}_{Pt_kPt_X}(C^{n+1}) < 0$ (or) $\Delta^{n}_{Pt_{2k+1}Pt_X}(C^{n+1}) < 0$ (&) $\Delta^{n}_{Pt_1Pt_2}(C^{n+1}) < 0$
-1	$\ldots$	-1	$\Delta^{n}_{Pt_{k+1}Pt_X}(C^{n+1}) < 0$ (or) $\ldots$ (or) $\Delta^{n}_{Pt_{2k}Pt_X}(C^{n+1}) < 0$ (or) $\Delta^{n}_{Pt_{2k+1}Pt_X}(C^{n+1}) < 0$ (&) $\Delta^{n}_{Pt_1Pt_2}(C^{n+1}) > 0$

Table 17 Conditions when $f_n \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$

C_{n-1}	$\dots$	C_0	$\Delta_{Pt_j Pt_k}^{n-1}(C^n)$
1	$\dots$	1	$\Delta_{Pt_{k+1} Pt_X}^{n-1}(C^n) < 0$ (or) $\dots$ (or) $\Delta_{Pt_{2k} Pt_X}^{n-1}(C^n) < 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\dots$	-1	$\Delta_{Pt_1 Pt_X}^{n-1}(C^n) < 0$ (or) $\dots$ (or) $\Delta_{Pt_k Pt_X}^{n-1}(C^n) < 0$

Table 18 Conditions when $f_n + 1 \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$

C_{n-1}	$\dots$	C_0	$\Delta_{Pt_j Pt_k}^{n-1}(C^n)$
1	$\dots$	1	$\Delta_{Pt_{k+1} Pt_X}^{n-1}(C^n) < 0$ (or) $\dots$ (or) $\Delta_{Pt_{2k} Pt_X}^{n-1}(C^n) < 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\dots$	-1	$\Delta_{Pt_1 Pt_X}^{n-1}(C^n) < 0$ (or) $\dots$ (or) $\Delta_{Pt_k Pt_X}^{n-1}(C^n) < 0$

Table 19 Conditions for $f \in \mathcal{B}_{n+1,n+1}^{\mathsf{PA-PUF}}$

C_n	$\dots$	C_0	$\Delta_{Pt_j Pt_k}^{n}(C^{n+1})$
1	$\dots$	1	$\Delta_{Pt_1 Pt_X}^{n}(C^{n+1}) < 0$ (or) $\dots$ (or) $\Delta_{Pt_{n+1} Pt_X}^{n}(C^{n+1}) < 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\dots$	-1	$\Delta_{Pt_1 Pt_X}^{n}(C^{n+1}) < 0$ (or) $\dots$ (or) $\Delta_{Pt_{n+1} Pt_X}^{n}(C^{n+1}) < 0$

Table 20 Conditions for $f \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$

C_n	$\dots$	C_0	$\Delta_{Pt_j Pt_k}^{n}(C^{n+1})$
1	$\dots$	1	$\Delta_{Pt_1 Pt_X}^{n}(C^{n+1}) < 0$ (or) $\dots$ $\Delta_{Pt_{n+1} Pt_X}^{n}(C^{n+1}) < 0$ (or) $\Delta_{Pt_{n+2} Pt_X}^{n}(C^{n+1}) < 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
-1	$\dots$	-1	$\Delta_{Pt_1 Pt_X}^{n}(C^{n+1}) < 0$ (or) $\dots$ $\Delta_{Pt_{n+1} Pt_X}^{n}(C^{n+1}) < 0$ (or) $\Delta_{Pt_{n+2} Pt_X}^{n}(C^{n+1}) < 0$

2^n conditions, we consider $\Delta_{Pt_1 Pt_X}^{n-1}(C^n), \dots, \Delta_{Pt_{2k-2} Pt_X}^{n-1}(C^n)$ from Table 15. Further, we choose the delay parameters $dl_n^{(Pt_{2k} Pt_{2k})}$ and $dl_n^{(\sigma_{n+1}(Pt_{2k}) Pt_{2k})}$ in such a way that $\delta_{Pt_{2k}}(n) > \max\{\delta_{Pt_1}(n), \dots, \delta_{Pt_{2k-1}}(n)\}$ which makes $\Delta_{Pt_j Pt_{2k}}^{n}(C^{n+1}) < 0$. We also choose $dl_n^{(Pt_1 Pt_1)} \approx dl_n^{(Pt_2 Pt_2)} \approx \dots \approx dl_n^{(Pt_{n+1} Pt_{n+1})}$ and $dl_n^{(Pt_{n+1} Pt_1)} \approx dl_n^{(Pt_1 Pt_2)} \approx \dots \approx dl_n^{(Pt_1 Pt_{n+1})}$ so that we can carry forward the inequality signs of Tables 14 and 15 to Table 13.

The delay parameters involved in the inequalities of Tables 14 and 15 have solutions. Using these delay parameters we are able to form a solution corresponding to the delay parameters involved in the inequalities of Table 13. This shows that $f_{n+1} = x_0 + x_2 + \cdots + x_n \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$.

Table 21 Conditions for $f_1 = (0, 0, 0, 1) \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$

C_1	C_0	$\Delta^1_{Pt_j Pt_k}(C^2)$
1	1	$\Delta^1_{Pt_2 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) > 0$
1	−1	$\Delta^1_{Pt_2 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) > 0$
−1	1	$\Delta^1_{Pt_2 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) > 0$
−1	−1	$\Delta^1_{Pt_1 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) < 0$

Table 22 Conditions for $f_2 = (1, 0, 0, 0) \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$

C_1	C_0	$\Delta^1_{Pt_j Pt_k}(C^2)$
1	1	$\Delta^1_{Pt_1 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) < 0$
1	−1	$\Delta^1_{Pt_2 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) > 0$
−1	1	$\Delta^1_{Pt_2 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) > 0$
−1	−1	$\Delta^1_{Pt_2 Pt_X}(C^2) < 0$ (or) $\Delta^1_{Pt_3 Pt_X}(C^2) < 0$ & $\Delta^1_{Pt_1 Pt_2}(C^2) > 0$

Table 23 Conditions for $f_3 = f_1 \parallel f_2 = (0, 0, 0, 1, 1, 0, 0, 0)$ in $\mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$

C_2	C_1	C_0	$\Delta^2_{Pt_j Pt_k}(C^3)$
1	1	1	$\Delta^2_{Pt_3 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4 Pt_X}(C^3) < 0$
1	1	−1	$\Delta^2_{Pt_3 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4 Pt_X}(C^3) < 0$
1	−1	1	$\Delta^2_{Pt_3 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4 Pt_X}(C^3) < 0$
1	−1	−1	$\Delta^2_{Pt_1 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_2 Pt_X}(C^3) < 0$
−1	1	1	$\Delta^2_{Pt_1 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_2 Pt_X}(C^3) < 0$
−1	1	−1	$\Delta^2_{Pt_3 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4 Pt_X}(C^3) < 0$
−1	−1	1	$\Delta^2_{Pt_3 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4 Pt_X}(C^3) < 0$
−1	−1	−1	$\Delta^2_{Pt_3 Pt_X}(C^3) < 0$ (or) $\Delta^2_{Pt_4 Pt_X}(C^3) < 0$

Case II. Let $n + 1$ be an odd number, $n + 1 = 2k + 1$.

Here, $\Delta^n_{Pt_j Pt_X}(C^{n+1})\big|_{(1,C_{n-1},\ldots,C_0)}$ can be expressed as $\Delta^{n-1}_{Pt_j Pt_X}(C^n)\big|_{(C_{n-1},\ldots,C_0)} + dl_n^{(Pt_j Pt_j)} - dl_n^{(Pt_X Pt_X)}$. We note that the set of conditions involving $\Delta^n_{Pt_1 Pt_X}(C^{n+1})$, …, $\Delta^n_{Pt_{2k} Pt_X}(C^{n+1})$ of Table 16 can be constructed via $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)$, …, $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n)$ of Table 17. For $C_n = 1$, that is, the first 2^n conditions, we consider $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)$, …, $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n)$ from Table 17. For $C_n = -1$, that is, the next 2^n conditions, we $\Delta^{n-1}_{Pt_1 Pt_X}(C^n)$, …, $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n)$ from Table 18. We also choose the delay parameters $dl_n^{(Pt_{2k+1} Pt_{2k+1})}$, $dl_n^{(\sigma_{n+1}(Pt_{2k+1}) Pt_{2k+1})}$ in such a way that $\delta_{Pt_{2k+1}}(n) > \max\{\delta_{Pt_1}(n), \ldots, \delta_{Pt_{2k}}(n)\}$, which makes $\Delta^n_{Pt_j Pt_{2k+1}}(C^{n+1}) < 0$. We also choose $dl_n^{(Pt_1 Pt_1)} \approx dl_n^{(Pt_2 Pt_2)} \approx \ldots \approx dl_n^{(Pt_{n+1} Pt_{n+1})}$

Table 24 Expanded form of the conditions when $f_3 = (0, 0, 0, 1, 1, 0, 0, 0) \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$

C_2	C_1	C_0	$\Delta^1_{Pt_j Pt_k}(C^3)$
1	1	1	$\Delta^1_{Pt_3 Pt_X}(C^2) + dl_2^{(Pt_3 Pt_3)} - dl_2^{(Pt_X Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_4 Pt_X}(C^2) + dl_2^{(Pt_4 Pt_4)} - dl_2^{(Pt_X Pt_X)} < 0$
1	1	−1	$\Delta^1_{Pt_3 Pt_X}(C^2) + dl_2^{(Pt_3 Pt_3)} - dl_2^{(Pt_X Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_4 Pt_X}(C^2) + dl_2^{(Pt_4 Pt_4)} - dl_2^{(Pt_X Pt_X)} < 0$
1	−1	1	$\Delta^1_{Pt_3 Pt_X}(C^2) + dl_2^{(Pt_3 Pt_3)} - dl_2^{(Pt_X Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_4 Pt_X}(C^2) + dl_2^{(Pt_4 Pt_4)} - dl_2^{(Pt_X Pt_X)} < 0$
1	−1	−1	$\Delta^1_{Pt_1 Pt_X}(C^2) + dl_2^{(Pt_1 Pt_1)} - dl_2^{(Pt_X Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_2 Pt_X}(C^2) + dl_2^{(Pt_2 Pt_2)} - dl_2^{(Pt_X Pt_X)} < 0$
−1	1	1	$\Delta^1_{Pt_4 \sigma_4(Pt_X)}(C^2) + dl_2^{(Pt_4 Pt_1)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_1 \sigma_4(Pt_X)}(C^2) + dl_2^{(Pt_1 Pt_2)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$
−1	1	−1	$\Delta^1_{Pt_2 \sigma(Pt_X)}(C^2) + dl_2^{(Pt_2 Pt_3)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_3 \sigma_4(Pt_X)}(C^2) + dl_2^{(Pt_3 Pt_4)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$
−1	−1	1	$\Delta^1_{Pt_2 \sigma(Pt_X)}(C^2) + dl_2^{(Pt_2 Pt_3)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_3 \sigma_4(Pt_X)}(C^2) + dl_2^{(Pt_3 Pt_4)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$
−1	−1	−1	$\Delta^1_{Pt_2 \sigma(Pt_X)}(C^2) + dl_2^{(Pt_2 Pt_3)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$ (or)
			$\Delta^1_{Pt_3 \sigma_4(Pt_X)}(C^2) + dl_2^{(Pt_3 Pt_4)} - dl_2^{(\sigma_4(X) Pt_X)} < 0$

and $dl_n^{(Pt_{n+1} Pt_1)} \approx dl_n^{(Pt_1 Pt_2)} \approx \ldots \approx dl_n^{(Pt_1 Pt_{n+1})}$, so that we can carry forward the signs of the inequalities from Tables 17 and 18 to Table 16.

Using the solutions corresponding to the delay parameters involved in the inequalities of Tables 17 and 18 we could form a solution corresponding to the delay parameters involved in the inequalities of Table 16. Hence $f_{n+1} = x_0 + x_2 + \cdots + x_n \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$. □

Theorem 3 *The set of all Boolean functions generated by an $(n+1)$-path PA-PUF is a proper superset of the set of all Boolean functions generated by an n-path PA-PUF, that is, $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}} \supset \mathcal{B}_{n,n}^{\mathsf{PA-PUF}}$, for $n \geq 2$.*

Proof We will prove $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}} \supseteq \mathcal{B}_{n,n}^{\mathsf{PA-PUF}}$ by mathematical induction. For $i = 2$, the delay difference for a 2-path PA-PUF is

$$\Delta^1_{Pt_1 Pt_2}(C^2) = \left(\frac{C_1 + 1}{2}\right)\left(\Delta^0_{Pt_1 Pt_2}(C^1) + dl_1^{(Pt_1 Pt_1)} - dl_1^{(Pt_2 Pt_2)}\right) - \left(\frac{C_1 - 1}{2}\right)\left(\Delta^0_{Pt_2 Pt_1}(C^1) + dl_1^{(Pt_2 Pt_1)} - dl_1^{(Pt_1 Pt_2)}\right) \quad (16)$$

and the delay difference equation for a 3-path PA-PUF is

$$\Delta^1_{Pt_1 Pt_2}(C^2) = \left(\frac{C_1 + 1}{2}\right)\left(\Delta^0_{Pt_1 Pt_2}(C^1) + dl_1^{(Pt_1 Pt_1)} - dl_1^{(Pt_2 Pt_2)}\right)$$

Table 25 Expression of $\Delta^1_{Pt_1Pt_2}(C^2)$

C_1	C_0	$\Delta^1_{Pt_1Pt_2}(C^2)$
1	1	$\Delta^0_{Pt_1Pt_2}(C^1) + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)}$
1	−1	$\Delta^0_{Pt_1Pt_2}(C^1) + dl_1^{(Pt_1Pt_1)} - dl_1^{(Pt_2Pt_2)}$
−1	1	$\Delta^0_{Pt_1Pt_2}(C^1) + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)}$
−1	−1	$\Delta^0_{Pt_1Pt_2}(C^1) + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)}$

Table 26 $\Delta^0_{Pt_1Pt_2}(C^1)$ conditions for the existence of all Boolean functions in $\mathcal{B}^{\mathsf{PUF}}_{2,1}$ [16]

$f_1 = 1$		$f_2 = x_0 + 1$	
C_0	$\Delta^0_{Pt_1Pt_2}(C^1)$	C_0	$\Delta^0_{Pt_1Pt_2}(C^1)$
1	$\Delta^0_{Pt_1Pt_2}(C^1) < 0$	1	$\Delta^0_{Pt_1Pt_2}(C^1) > 0$
−1	$\Delta^0_{Pt_1Pt_2}(C^1) < 0$	−1	$\Delta^0_{Pt_1Pt_2}(C^1) < 0$
$f_3 = x_0$		$f_4 = 0$	
C_0	$\Delta^0_{Pt_1Pt_2}(C^1)$	C_0	$\Delta^0_{Pt_1Pt_2}(C^1)$
1	$\Delta^0_{Pt_1Pt_2}(C^1) < 0$	1	$\Delta^0_{Pt_1Pt_2}(C^1) > 0$
−1	$\Delta^0_{Pt_1Pt_2}(C^1) > 0$	−1	$\Delta^0_{Pt_1Pt_2}(C^1) > 0$

$$-\left(\frac{C_1-1}{2}\right)\left(\Delta^0_{Pt_3Pt_1}(C^1) + dl_1^{(Pt_3Pt_1)} - dl_1^{(Pt_1Pt_2)}\right). \tag{17}$$

We can clearly see that when $\Delta^0_{Pt_3Pt_1}(C^1) = \Delta^0_{Pt_2Pt_1}(C^1)$ and $dl_1^{(Pt_2Pt_1)} = dl_1^{(Pt_3Pt_1)}$, then Eqs. (16) and (17) are similar. In that case the delay difference equation of a 2-path PA-PUF is a particular case of a 3-path PA-PUF. Thus every Boolean function which is generated by a 2-path PA-PUF can be generated using a 3-path PA-PUF, implying that $\mathcal{B}^{\mathsf{PA-PUF}}_{3,2} \supseteq \mathcal{B}^{\mathsf{PA-PUF}}_{2,2}$.

Let us assume that the result is true for $i = n$, that is, $\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n} \supseteq \mathcal{B}^{\mathsf{PA-PUF}}_{n,n}$. We now need to show (for $i = n+1$) that $\mathcal{B}^{\mathsf{PA-PUF}}_{n+2,n+1} \supseteq \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$. If $f \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n+1}$ then the inequalities given in Table 19 will have a solution corresponding to the delay parameters.

Now we construct a system of inequalities using the those presented in Table 19. For example, we take $\Delta^n_{Pt_jPt_X}(C^{n+1}) < 0$ for any arbitrary $X \in \{1, 2, \ldots, n+1\}$. Then, we choose the delay parameters $dl_n^{(Pt_n+2Pt_n+2)}$ and $dl_n^{(\sigma_{n+2}(Pt_n+2)Pt_n+2)}$ such that $\delta_{Pt_{n+2}}(n) > \max(\delta_{Pt_1}(n), \ldots, \delta_{Pt_{n+1}}(n))$, which makes $\Delta^n_{Pt_jPt_{n+2}}(C^{n+1}) < 0$. This setup will force $\Delta^n_{Pt_jPt_X}(C^{n+1}) < 0$, where $X \in \{1, 2, \ldots, n+2\}$. Using this technique we can infer that the inequalities presented in Table 20 will have a solution corresponding to the delay parameters.

The inequalities of Table 20 correspond to the same Boolean function f. As the inequalities have a solution corresponding to the involved delay parameters, we can say that $f \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+2,n+1}$. This implies $\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n} \supseteq \mathcal{B}^{\mathsf{PA-PUF}}_{n,n}$. The strict inclusion is obtained by combining the results of Theorems 1 and 2, and so, we get $\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n} \supset \mathcal{B}^{\mathsf{PA-PUF}}_{n,n}$, which is our desired result. □

We now describe a technique starting with an example where $f_1, f_2 \in \mathcal{B}^{\mathsf{PA-PUF}}_{3,2}$, and we show the existence of a solution of the inequalities corresponding to $(f_1 \parallel f_2) \in \mathcal{B}^{\mathsf{PA-PUF}}_{4,3}$.

Table 27 Comparisons of different PUF designs

PUF Design	Uniqueness (%)	Reliability (%)	Hardware	Area
17-path PA-PUF	47.14	99.41	Spartan-7	113 slices per PUF
PA-PUF^{128} [8]	49.63	95	Artix-7	47 slices per PUF
SRAM PUF[6]	49.97	88	FPGA	4800 SRAM bits
Butterfly PUF^{64}[10]	50	94	Virtex-5	130 slices
Ring Oscillator PUF^{128}[22]	46.15	99.52	Virtex-4	1024 ring oscillators
APUF^{64}[7]	36.75	98.28	Virtex-5	129 slices

We let, for example, the functions $f_1 = (0,0,0,1)$, $f_2 = (1,0,0,0) \in \mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$. The inequalities corresponding to f_1, f_2 are described in Tables 21 and 22.

The parameter $\Delta^2_{Pt_jPt_k}(C^3)$ can be represented in terms of $\Delta^1_{Pt_jPt_k}(C^2)$ for $C_2 = 1$ and $\Delta^2_{Pt_jPt_k}(C^3)$ can be represented in terms of $\Delta^1_{\sigma_4(Pt_j)\sigma_4(Pt_k)}(C^2)$ for $C_2 = -1$. We rewrite the conditions of Table 23 in Table 24.

The inequalities in Table 24 can be constructed using the inequalities of $\Delta^1_{Pt_jPt_X}(C^2)$ presented in Tables 21 and 22. For $C_2 = 1$ we take the four conditions of $\Delta^1_{Pt_jPt_X}(C^2) < 0$ from Table 21, that is,

$$\Delta^1_{Pt_3Pt_X}(C^2) < 0, \quad \Delta^1_{Pt_3Pt_X}(C^2) < 0, \quad \Delta^1_{Pt_3Pt_X}(C^2) < 0, \quad \Delta^1_{Pt_1Pt_X}(C^2) < 0.$$

These four inequalities have a solution for the delay parameters, let the solution be $dl_0^{(Pt_3Pt_3)} < -dl_1^{(Pt_3Pt_3)} + dl_1^{(Pt_XPt_X)} + dl_0^{(Pt_XPt_X)}$, $dl_1^{(Pt_3Pt_3)} < -dl_0^{(Pt_2Pt_3)} + dl_1^{(Pt_XPt_X)} + dl_0^{(\sigma_3(Pt_X)Pt_X)}$, $dl_0^{(Pt_2Pt_3)} < -dl_1^{(Pt_3Pt_1)} + dl_1^{(\sigma_3(Pt_X)Pt_X)} + dl_0^{(\sigma_3^2(Pt_X)\sigma_3(Pt_X))}$, $dl_0^{(Pt_2Pt_2)} < dl_1^{(\sigma_3(Pt_X)Pt_X)} - dl_1^{(Pt_2Pt_3)} + dl_0^{(\sigma_3(Pt_X)\sigma_3(Pt_X))}$. For $C_2 = -1$ we take the four conditions of $\Delta^1_{Pt_jPt_X}(C^2)$ from Table 22, that is,

$$\Delta^1_{Pt_1Pt_X}(C^2) < 0, \quad \Delta^1_{Pt_3Pt_X}(C^2) < 0, \quad \Delta^1_{Pt_3Pt_X}(C^2) < 0, \quad \Delta^1_{Pt_3Pt_X}(C^2) < 0.$$

These four inequalities have a solution for the delay parameter. Let the solution be $dl_0^{(Pt_1Pt_1)} < dl_1^{(Pt_XPt_X)} + dl_0^{(Pt_XPt_X)} - dl_1^{(Pt_1Pt_1)}$, $dl_1^{(Pt_3Pt_3)} < -dl_0^{(Pt_2Pt_3)} + dl_1^{(Pt_XPt_X)} + dl_0^{(\sigma_3(Pt_X)Pt_X)}$, $dl_0^{(Pt_2Pt_2)} < -dl_1^{(Pt_2Pt_3)} + dl_1^{(\sigma_3(Pt_X)Pt_X)} + dl_0^{(\sigma_3(Pt_X)\sigma_3(Pt_X))}$, $dl_1^{(Pt_2Pt_3)} < dl_1^{(\sigma_3(Pt_X)Pt_X)} + dl_0^{(\sigma_3^2(Pt_X)\sigma_3(Pt_X))} - dl_0^{(Pt_1Pt_2)}$. We choose the delay parameters $dl_2^{(Pt_4Pt_4)}$ and $dl_2^{(Pt_3Pt_4)}$ such that $\delta_{Pt_4}(2) > \max\{\delta_{Pt_1}(2), \delta_{Pt_2}(2), \delta_{Pt_3}(2)\}$ to make $\Delta^2_{Pt_jPt_4}(C^3) < 0$, for all $j \neq 4$. To keep the first four inequalities in Table 24 with the same sign as in Table 21, we impose these conditions $dl_2^{(Pt_1Pt_1)} \approx dl_2^{(Pt_2Pt_2)} \approx dl_2^{(Pt_3Pt_3)} \approx dl_2^{(Pt_4Pt_4)}$. To keep the last four inequalities in Table 24 with the same sign as in Table 22, we impose these conditions $dl_2^{(Pt_4Pt_1)} \approx dl_2^{(Pt_1Pt_2)} \approx dl_2^{(Pt_2Pt_3)} \approx dl_2^{(Pt_1Pt_4)}$. Now we have a set of inequalities which is similar to the set of inequalities of Table 23. As the inequalities of Tables 21, 22 have solutions corresponding to the delay parameters, we can form the solution corresponding to the inequalities of Table 23 using the above mentioned technique. Hence $f_1 \parallel f_2 \in \mathcal{B}_{4,3}^{\mathsf{PA-PUF}}$. We will use this solution forming technique to prove our main result, Theorem 4.

Theorem 4 *For $n \geq 2$, we have $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}} = \mathcal{B}_n$.*

Proof We prove this statement using mathematical induction. For $n = 2$, we show $\mathcal{B}_{3,2}^{\mathsf{PA-PUF}} = \mathcal{B}_2$, that is, the system in Table 25 should have a solution for any combination of inequality signs (that is, for all 2^{2^2} possible inequalities).

From [16] we already know that $\mathcal{B}_1^{\mathsf{PUF}} = \mathcal{B}_1$. Thus the involved delay parameters have a solution corresponding to any combination of inequality signs corresponding to $\Delta^0_{Pt_1Pt_2}(C^1)$ (see Table 26).

We use the same values of the delay parameters obtained from Table 26 in the expression of $\Delta^1_{Pt_1Pt_2}(C^2)|_{(1,1)}$ and $\Delta^1_{Pt_1Pt_2}(C^2)|_{(1,-1)}$ from one set of $\Delta^0_{Pt_1Pt_2}(C^1)$ conditions and $\Delta^1_{Pt_1Pt_2}(C^2)|_{(-1,1)}$ and $\Delta^1_{Pt_1Pt_2}(C^2)|_{(-1,-1)}$ from the other set of $\Delta^0_{Pt_1Pt_2}(C^1)$ conditions. We choose $dl_1^{(Pt_1Pt_1)} \approx dl_1^{(Pt_2Pt_2)}$ and $dl_1^{(Pt_3Pt_1)} \approx dl_1^{(Pt_1Pt_2)}$ so that we can have the same inequality sign in the expression of Table 25 as in Table 26. This shows the existence of the solution corresponding to the delay parameters involved in Table 25 with any possible inequality sign. Using all possible combinations of $\Delta^0_{Pt_1Pt_2}(C^1)$ conditions we can construct all 2^{2^2} inequalities, which guarantees the existence of all 2^{2^2} Boolean functions in $\mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$. Hence, $\mathcal{B}_{3,2}^{\mathsf{PA-PUF}} = \mathcal{B}_2$.

Now, we assume that the statement is true for $i = n$, that is, $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}} = \mathcal{B}_n$ and we will show that the statement remains true for $i = n + 1$. To prove this we show that there exist a solution corresponding to the delay parameters corresponding to the concatenation of two systems of inequalities associated to two Boolean functions from $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$. This is similar to the above example, where we concatenated two systems of inequalities corresponding to two Boolean functions $\mathcal{B}_{2,1}^{\mathsf{PA-PUF}}$ to form a system of inequalities associated to one Boolean function in $\mathcal{B}_{3,2}^{\mathsf{PA-PUF}}$. We consider two cases based upon the parity of $n + 1$.

Case I. Let $n + 1 = 2k$. The conditions for the existence of a Boolean function in $\mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$ are $\Delta^n_{Pt_1Pt_X}(C^{n+1}) < 0$, $\Delta^n_{Pt_2Pt_X}(C^{n+1}) < 0, \ldots, \Delta^n_{Pt_{2k}Pt_X}(C^{n+1}) < 0$. This can also be expressed as $\Delta^{n-1}_{Pt_1Pt_X}(C^n) + dl_n^{(Pt_1Pt_1)} - dl_n^{(Pt_XPt_X)} < 0$, $\Delta^{n-1}_{Pt_2Pt_X}(C^n) + dl_n^{(Pt_2Pt_2)} - dl_n^{(Pt_XPt_X)} < 0, \ldots, \Delta^{n-1}_{Pt_{2k}Pt_X}(C^n) + dl_n^{(Pt_{2k}Pt_{2k})} - dl_n^{(Pt_XPt_X)} < 0$ when $C_n = 1$ and $\Delta^{n-1}_{Pt_nPt_X}(C^n) + dl_n^{(Pt_nPt_1)} - dl_n^{\sigma_{n+2}(Pt_X)(Pt_X)} < 0$, $\Delta^{n-1}_{Pt_1Pt_X}(C^n) + dl_n^{(Pt_1Pt_2)} - dl_n^{(\sigma_{n+2}(Pt_X)Pt_X)} < 0, \ldots, \Delta^{n-1}_{Pt_{2k-1}Pt_X}(C^n) + dl_n^{(Pt_{2k-1}Pt_{2k})} - dl_n^{(\sigma_{n+2}(Pt_X)Pt_X)} < 0$ when $C_n = -1$. There will be 2^{n+1} inequalities of this form. We can construct a system of inequalities, where the first 2^n numbers of $\Delta^{n-1}_{Pt_iPt_X}(C^n)$ are from the conditions of the existence of a Boolean function in $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$ and the next 2^n numbers of $\Delta^{n-1}_{Pt_iPt_X}(C^n)$ are from the conditions of the existence of another Boolean function in $\mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$. Then we choose the delay parameters $dl_n^{(Pt_{2k}Pt_{2k})}$ and $dl_n^{(\sigma_{n+2}(Pt_{2k})Pt_{2k})}$ such that $\delta_{Pt_{2k}}(n) > \max\{\delta_{Pt_1}(n), \ldots, \delta_{Pt_{2k-1}}(n)\}$ which makes $\Delta^n_{Pt_jPt_{2k}}(C^{n+1}) < 0$. We also choose $dl_n^{(Pt_1Pt_1)} \approx dl_n^{(Pt_2Pt_2)} \approx \ldots dl_n^{(Pt_{2k}Pt_{2k})}$ and $dl_n^{(Pt_1Pt_{2k})} \approx dl_n^{(Pt_1Pt_2)} \approx \ldots dl_n^{(Pt_{2k-1}Pt_{2k})}$ such that $\Delta^n_{Pt_iPt_X}(C^{n+1})$ has the same inequality sign as $\Delta^{n-1}_{Pt_iPt_X}(C^n)$. The existence of solutions corresponding to the delay parameters involved in $\Delta^n_{Pt_1Pt_X}(C^{n+1}) < 0$, $\Delta^n_{Pt_2Pt_X}(C^{n+1}) < 0, \ldots, \Delta^n_{Pt_{2k}Pt_X}(C^{n+1}) < 0$ guarantees that if $f_1, f_2 \in \mathcal{B}_{n+1,n}^{\mathsf{PA-PUF}}$ then $f_1 \parallel f_2 \in \mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$.

Case II. Let $n + 1 = 2k + 1$. The conditions for the existence of a Boolean function in $\mathcal{B}_{n+2,n+1}^{\mathsf{PA-PUF}}$ are $\Delta^n_{Pt_1Pt_X}(C^{n+1}) < 0$, $\Delta^n_{Pt_2Pt_X}(C^{n+1}) < 0, \ldots, \Delta^n_{Pt_{2k+1}Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1Pt_2}(C^{n+1}) < 0$, $\Delta^n_{Pt_{2k+1}Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1Pt_2}(C^{n+1}) > 0$. This can also be expressed as $\Delta^{n-1}_{Pt_1Pt_X}(C^n) + dl_n^{(Pt_1Pt_1)} - dl_n^{(Pt_XPt_X)} < 0$,

$\Delta^{n-1}_{Pt_2 Pt_X}(C^n) + dl_n^{(Pt_2 Pt_2)} - dl_n^{(Pt_X Pt_X)} < 0$, …, $\Delta^{n-1}_{Pt_{2k+1} Pt_X}(C^n) + dl_n^{(Pt_{2k+1} Pt_{2k+1})} - dl_n^{(Pt_X Pt_X)} < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) + dl_n^{(Pt_1 Pt_1)} - dl_n^{(Pt_2 Pt_2)} < 0$, $\Delta^{n-1}_{Pt_{2k+1} Pt_X}(C^n) + dl_{2k+1}^{(Pt_{2K+1} Pt_{2k+1})} - dl_n^{(Pt_X Pt_X)} < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) + dl_n^{(Pt_1 Pt_1)} - dl_n^{(Pt_2 Pt_2)} > 0$, when $C_n = 1$, and $\Delta^{n-1}_{Pt_{2k} Pt_X}(C^n) + dl_n^{(Pt_{2k} Pt_1)} - dl_n^{(\sigma_{n+1}(Pt_X) Pt_X)} < 0$, $\Delta^{n-1}_{Pt_1 Pt_X}(C^n) + dl_{2k}^{(Pt_1 Pt_2)} - dl_n^{(\sigma_{n+1}(Pt_X) Pt_X)} < 0$, …, $\Delta^{n-1}_{Pt_{2k+1} Pt_X}(C^n) + dl_n^{(Pt_{2k} Pt_{2k+1})} - dl_n^{(Pt_X \sigma_{n+1}(Pt_X))} < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) + dl_n^{(Pt_{2k+1} Pt_1)} - dl_n^{(Pt_1 Pt_2)} < 0$, $\Delta^{n-1}_{Pt_{2k+1} Pt_X}(C^n) + dl_n^{(Pt_{2k} Pt_{2k+1})} - dl_n^{(\sigma_{n+1}(Pt_X) Pt_X)} < 0$ (&) $\Delta^{n-1}_{Pt_1 Pt_2}(C^n) + dl_n^{(Pt_{2k+1} Pt_1)} - dl_n^{(Pt_1 Pt_2)} > 0$, when $C_n = -1$. There will be 2^{n+1} inequalities of this form. We can now construct a system of inequalities, where the first 2^n numbers of $\Delta^{n-1}_{Pt_i Pt_X}(C^n)$, $\Delta^{n-1}_{Pt_1 Pt_2}(C^n)$ are from the conditions of the existence of a Boolean function in $\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n}$ and the next 2^n numbers of $\Delta^{n-1}_{Pt_i Pt_X}(C^n)$, $\Delta^{n-1}_{Pt_1 Pt_2}(C^n)$ values are from the conditions of existence of a Boolean function in $\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n}$. Further, we choose the delay parameters $dl_n^{(Pt_{2k+1} Pt_{2k+1})}$ and $dl_n^{(\sigma_{n+1}(Pt_{2k+1}) Pt_{2k+1})}$ in such a way that we get $\delta_{Pt_{2k+1}}(n) > \max(\delta_{Pt_1}(n), \ldots, \delta_{Pt_{2k}}(n))$, which makes $\Delta^n_{Pt_j Pt_{2k+1}}(C^{n+1}) < 0$. We also choose $dl_n^{(Pt_1 Pt_1)} \approx dl_n^{(Pt_2 Pt_2)} \approx \ldots dl_n^{(Pt_{2k+1} Pt_{2k+1})}$ and $dl_n^{(Pt_{2k+1} Pt_1)} \approx dl_n^{(Pt_1 Pt_2)} \approx \ldots dl_n^{(Pt_{2k} Pt_{2k+1})}$ such that $\Delta^n_{Pt_i Pt_X}(C^{n+1})$ has the same inequality sign as $\Delta^{n-1}_{Pt_i Pt_X}(C^n)$. The existence of solutions corresponding to the delay parameters involved in $\Delta^n_{Pt_1 Pt_X}(C^{n+1}) < 0$, $\Delta^n_{Pt_2 Pt_X}(C^{n+1}) < 0$, …, $\Delta^n_{Pt_{2k+1} Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1 Pt_2}(C^{n+1}) < 0$, $\Delta^n_{Pt_{2k+1} Pt_X}(C^{n+1}) < 0$ (&) $\Delta^n_{Pt_1 Pt_2}(C^{n+1}) > 0$ guarantees that if $f_1, f_2 \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n}$ then $f_1 \parallel f_2 \in \mathcal{B}^{\mathsf{PA-PUF}}_{n+2,n+1}$.

The existence of solutions of delay parameters under Cases I and II implies that an $(n+1)$-length PA-PUF with $(n+2)$ paths will generate all possible Boolean functions of $(n+1)$-variable as $\mathcal{B}^{\mathsf{PA-PUF}}_{n+1,n} = \mathcal{B}_n$, that is, $\mathcal{B}^{\mathsf{PA-PUF}}_{n+2,n+1} = \mathcal{B}_{n+1}$. This proves our theorem. □

In the next section we will discuss the feasibility of the implementation of our proposed model of PA-PUF in hardware. We will also discuss the security of our PA-PUF against several machine learning attacks.

4 Hardware implementation and resistance towards machine learning attacks

In this section we consider a toy model of n length PA-PUF with $n+1$ paths and show the feasibility of implementation of our proposed PA-PUF in hardware. We consider 17-path PA-PUF with a 16 bit challenge input and a 1 bit response. The main purpose of this implementation was not to create a performance-optimized device, but rather to serve as a validation platform to confirm that the core principles of our model hold true in a physical system. We instantiated the fundamental architectural elements of our PA-PUF model with 17 paths and 16 switches in Verilog HDL. The design was purposefully simplified to eliminate confounding variables from intricate optimizations and enable a straightforward evaluation of the underlying physical phenomena. Each PA-PUF instance occupied 113 LUT slices in Xilinx Spartan 7 FPGA.

We used the method and framework introduced by Maiti et al. [14] to evaluate our PA-PUF. In this approach, uniqueness measures how different the responses are across different chips. This ensures that each device can be uniquely identified. The implemented PA-PUF achieved

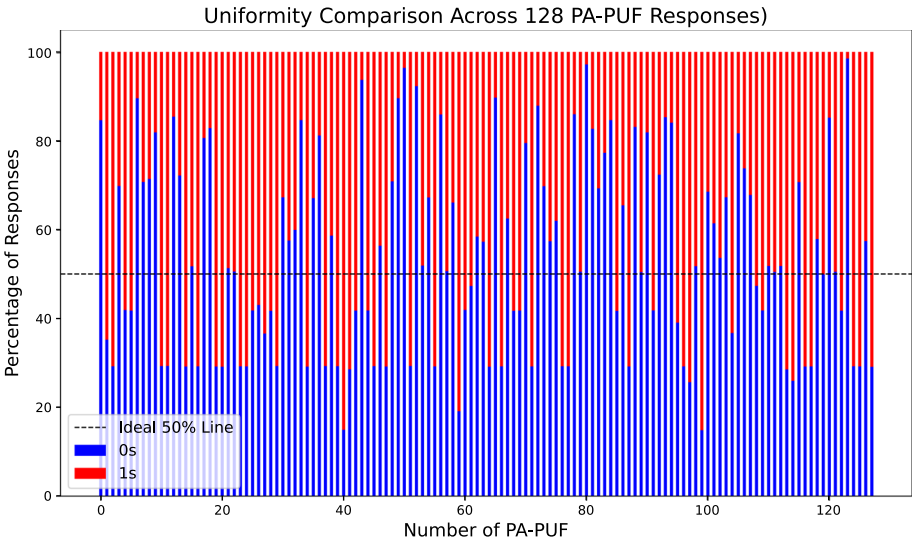

Fig. 6 Uniformity in 128 PA-PUF instances

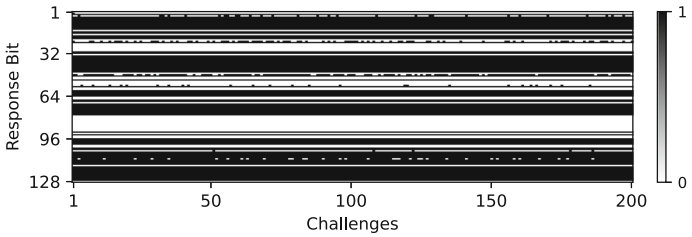

Fig. 7 Reliability in 128 PA-PUF instances

an average uniqueness of 47.14%, which is close to the theoretical ideal value. Uniformity assesses whether the response bits are evenly distributed between 0's and 1's, showing that there is no bias. The uniformity for our PA-PUF came out to be 45.95% measured on average with 128 PA-PUF instances, the oscillation in number of 0's and 1's as shown in Fig. 6 that our PA-PUF can generate different Boolean functions. However, we can see a few instances with ideal uniformity 50% that are suitable for practical applications.

The reliability checks how consistently a chip produces the same response under different environmental conditions, such as changes in temperature or voltage. For the implemented PA-PUF, an average reliability of 99.41% was achieved which can be observed in Fig. 7.Table 27 compares our 17-path PA-PUF with prior PUF designs.

We evaluated the resilience of the proposed 17-Path PA-PUF against Machine Learning (ML) based modeling attacks. Such attacks represent a significant threat, as they attempt to create a predictive software clone of the PUF from a set of observed challenge-response pairs. Although numerous studies have demonstrated the vulnerability of classical arbiter PUFs to these methods, our analysis sought to quantify the robustness of the PA-PUF architecture.

To do this, we subjected a large dataset of CRPs from the PA-PUF to several standard classifiers similar to the study on the 3-path PA-PUF [8] with the following configurations:

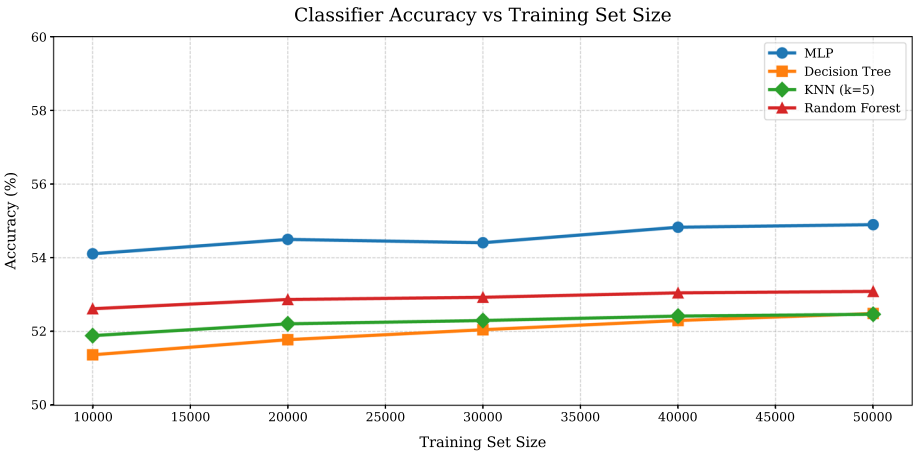

Fig. 8 Predictability percentage of PA-PUF several standard ML classifiers

1. Multi-Layer Perceptron (MLP): Using the ReLU activation function, Adam optimizer, and a learning rate of 0.001.
2. Decision Tree: Employing the Gini impurity criterion.
3. k-Nearest Neighbors (KNN): Configured with five neighbors.
4. Random Forest: Comprised of 100 decision trees.

The empirical results shown in Fig. 8 imply that, the highest prediction accuracy achieved by any of these models was approximately 54.92%. This low predictability indicates that our proposed model of PA-PUF exhibits a high resistance to these machine learning modeling techniques. We also believe that this simulation trend follows for any arbitrary $n+1$-paths PA-PUF with n switches due to its potentiality of generating all Boolean functions involving n-variables.

5 Conclusion

In this paper, we address the open problem raised by Kansal et al. [8]. We first generalize the classical PUF model with 2 paths to a PUF with an arbitrary number of paths called PA-PUF. Using this model of PA-PUF, we show that increasing the number of paths in PA-PUF increases the class of Boolean functions generated from the PA-PUF. We first show that $\mathcal{B}_{i+1,i} \supset \mathcal{B}_{i,i}$ for $i \geq 2$. In our main result, we show that we need $(n+1)$ paths in an n-length PA-PUF to generate the complete set of Boolean functions involving n variables. We also provide a hardware implementation of a particular instance of our proposed model of PA-PUF. From our simulation we notice that n-length PA-PUF with $n+1$ paths in every switch has good cryptographic properties.

Acknowledgements The authors sincerely thank the Editors, Reviewers for their time, insightful comments, and helpful recommendations, all of which have significantly enhanced the quality and readability of this work. Dibyendu Roy would like to acknowledge the grant from the project entitled "Design and Analysis of Physically Unclonable Functions" with project number 02011/18/2025/NBHM (R.P)/R&D II/12843 awarded by National Board for Higher Mathematics, Department of Atomic Energy, Govt. of India.

Author contributions All the authors have contributed equally.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Becker G.T.: The gap between promise and reality: on the insecurity of XOR arbiter PUFs. In: Güneysu T., Handschuh H.: (eds.) CHES 2015. LNCS, **9293**, pp. 535–555, Springer, Heidelberg (2015).
2. Chatterjee D., Pratihar K., Hazra A., Rührmair U., Mukhopadhyay D.: Systematically quantifying cryptanalytic nonlinearities in strong PUFs. IEEE Transactions on Information Forensics and Security (2023).
3. Chen Q., Csaba G., Ju X., Natarajan S. B., Lugli P., Stutzmann M., Schlichtmann U., Rührmair U.: Analog circuits for physical cryptography. Proc. 2009 12th International Symposium on Integrated Circuits, pp. 121–124, IEEE (2009).
4. Devadas S.: Physical unclonable functions and secure processors. In Clavier C., Gaj K.: (eds.) CHES 2009. LNCS, **5747**, p. 65, Springer, Heidelberg (2009).
5. Gassend B., Clarke D., Dijk M. V., Devadas S.: Silicon physical random functions. Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 148–160, ACM (2002).
6. Guajardo J., Kumar S.S., Schrijen G. J., Tuyls P.: FPGA intrinsic PUFs and their use for IP protection. Cryptographic hardware and embedded systems - CHES 2007, LNCS, **4727**, pp. 63–80, Springer (2007).
7. Hori Y., Takahiro Y., Toshihiro K., Akashi S.: Quantitative and statistical performance evaluation of arbiter physical unclonable functions on FPGAs. In 2010 International conference on reconfigurable computing and FPGAs, pp. 298–303. IEEE (2010).
8. Kansal M., Roy A., Roy D., Bodapati S., Chattopadhyay A.: Priority arbiter PUF: analysis. Discret. Appl. Math. **356**, 71–95 (2024).
9. Krishna A. R., Narasimhan S., Wang X., Bhunia S.: MECCA: A robust low-overhead PUF using embedded memory array. Cryptographic hardware and embedded systems - CHES 2011, LNCS, **6917**, pp. 407–420, Springer (2011).
10. Kumar S. S., Guajardo J., Maes R., Schrijen G. J., Tuyls P.: Extended abstract: The butterfly PUF protecting IP on every FPGA. IEEE International workshop on hardware-oriented security and trust, Anaheim, pp. 67–70, IEEE (2008).
11. Lee J.W., Lim D, Gassend B., Suh G.E., Dijk M.V., Devadas S.: A technique to build a secret key in integrated circuits for identification and authentication applications. Symposium on VLSI Circuits. Digest of Technical Papers, pp. 176–179, IEEE (2004).
12. Lim D., Lee J.W., Gassend B., Suh G.E., Dijk M.V., Devadas S.: Extracting secret keys from integrated circuits. IEEE Trans. Very Large Scale Integration (VLSI) Syst. **13**(10), 1200–1205 (2005).
13. Lofstrom K., Daasch W. R., Taylor D.: IC identification circuit using device mismatch. IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056), pp. 372–373, IEEE (2000).
14. Maiti A., Vikash G., Patrick S.: A systematic method to evaluate and compare the performance of physical unclonable functions. In Embedded systems design with FPGAs, pp. 245-267. Springer, New York (2012).
15. Mishra N., Pratihar K., Mandal S., Chakraborty A., Rührmair U., Mukhopadhyay D.: Calypso: an enhanced search optimization based framework to model delay-based PUFs. IACR Trans. Cryptogr. Hardware Embedded Syst. **2024**(1), 501–526 (2024).
16. Roy A., Roy D., Maitra S.: How do the arbiter PUFs sample the Boolean function class? In the proceeding of 28th Edition of Selected Areas in Cryptography (SAC 2021). Springer, LNCS (2021).
17. Rührmair U., Jaeger C., Hilgers C., Algasinger M., Csaba G., Stutzmann M.: Security applications of diodes with unique current-voltage characteristics (short paper). Financial cryptography and data security: 14th international conference, FC 2010, Tenerife, Canary Islands, January 25-28, 2010, Revised Selected Papers. **14**, pp. 328–335, Springer (2010).
18. Rührmair U., Sehnke F., Sölter J., Dror G., Devadas S., Schmidhuber J.: Modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM conference on computer and communications security, pp. 237–249. ACM (2010).
19. Siddhanti A. A., Bodapati S., Chattopadhyay A., Maitra S., Roy D., Stănică P.: Analysis of the strict avalanche criterion in variants of arbiter-based physically unclonable functions. Progress in Cryptology-INDOCRYPT 2019, LNCS, **11898**, pp. 556–577, Springer (2019).

20. Simons P., van der Sluis E., van der Leest V.: Buskeeper PUFs, a promising alternative to D Flip-Flop PUFs. In 2012 IEEE international symposium on hardware-oriented security and trust, pp. 7–12, IEEE (2012).
21. Singh S., Bodapati S., Patkar S., Leupers R., Chattopadhyay A., Merchant F.: PA-PUF: a novel priority arbiter PUF. IFIP/IEEE 30th international conference on very large scale integration (VLSI-SoC), Patras, Greece, pp. 1–6, IEEE (2022).
22. Suh G. E., Devadas S.: Physical unclonable functions for device authentication and secret key generation. 44th ACM/IEEE Design Automation Conference, San Diego, pp. 9–14 (2007).
23. Suzuki D., Shimizu K.: The glitch PUF: a new delay-PUF architecture exploiting glitch shapes. Cryptographic hardware and embedded systems - CHES 2010, LNCS, **6225**, pp. 366–382, Springer (2010).