DOI: 10.15439/2014F140
ACSIS, Vol. 2

A New Mode of Operation for Arbiter PUF to Improve Uniqueness on FPGA

Takanori Machida*, Dai Yamamoto†, Mitsugu Iwamoto* and Kazuo Sakiyama*
*The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
Email: {machida, mitsugu, sakiyama}@uec.ac.jp
†Fujitsu Laboratories Ltd.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki-shi, Kanagawa, 211-8588, Japan
Email: yamamoto.dai@jp.fujitsu.com

Abstract*—Arbiter-based Physically Unclonable Function (PUF) is one kind of the delay-based PUFs that use the time difference of two delay-line signals. One of the previous work suggests that Arbiter PUFs implemented on Xilinx Virtex-5 FPGAs generate responses with almost no difference, *i.e.* with low uniqueness. In order to overcome this problem, *Double Arbiter PUF* was proposed, which is based on a novel technique for generating responses with high uniqueness from duplicated Arbiter PUFs on FPGAs. It needs the same costs as 2-XOR Arbiter PUF that XORs outputs of two Arbiter PUFs. Double Arbiter PUF is different from 2-XOR Arbiter PUF in terms of *mode of operation for Arbiter PUF*: the wire assignment between an arbiter and output signals from the final selectors located just before the arbiter. In this paper, we evaluate these PUFs as for uniqueness, randomness, and steadiness. We consider finding a new mode of operation for Arbiter PUF that can be realized on FPGA. In order to improve the uniqueness of responses, we propose *3-1 Double Arbiter PUF* that has another duplicated Arbiter PUF, *i.e.* having 3 Arbiter PUFs and output 1-bit response. We compare 3-1 Double Arbiter PUF to 3-XOR Arbiter PUF according to the uniqueness, randomness, and steadiness, and show the difference between these PUFs by considering the mode of operation for Arbiter PUF. From our experimental results, the uniqueness of responses from 3-1 Double Arbiter PUF is approximately 50%, which is better than that from 3-XOR Arbiter PUF. We show that we can improve the uniqueness by using a new mode of operation for Arbiter PUF.

I. Introduction

RECENTLY, counterfeit products have been a problem in commercial market. The security for existing anti-counterfeit technologies relies on the technical difficulty to create a duplicate. However, future developments in counterfeit technologies might affect the technical difficulty. Physically Unclonable Function (PUF) [1] is being focused as a future solution [2].

PUF is a function in which an input (challenge) is related to one unique output (response) based on physical units such as semiconductor circuits. It is difficult to duplicate PUFs because the response values of PUFs depend on a physical variation. This difficulty to duplicate PUFs can be used device authentication against counterfeiting [3][4]. For example, a server as a verifier stores challenge–response pairs for a device as a prover. The verifier can authenticate the device by using the challenge–response pairs since they are unique for the PUF implemented in the device. PUFs are also used for a more secure method of storing secret keys than non-volatile memories. A secret key stored on internal memories will be revealed if an attacker can open the package of a device. In contrast, the secret key on PUFs cannot be read out accurately because physical variation and the values of responses have been changed once the package of the device is opened. Therefore, it is expected that PUFs are used for secure key generation [5][6].

PUFs are implemented not only on ASIC (Application Specific Integrated Circuit) [7] but also on FPGA (Field Programmable Gate Array) [8][9]. FPGA implementations have an advantage that their design and implementations are easy to change. Therefore, FPGAs are widely used in commercial products in the real world [10].

Some evaluation results on FPGAs of Arbiter PUF (APUF) [11], one of the delay-based PUFs, have been reported [12][13]. Previous work of [12][13] suggests that the APUFs implemented on Xilinx Virtex-5/Kintex-7 FPGAs generate responses with quite low uniqueness. The authors of [14] claim that one of the reasons for the low-unique responses obtained from APUFs on Virtex-5 FPGAs is based on the problem of SLICEs on the FPGAs. The problem is mentioned in general FPGAs. In a conventional APUF, a response is generated by comparing signals through two wires. The length of the two wires for any challenges in APUFs is expected to be equal. However, the layout of logic elements (*i.e.* SLICE) on FPGAs is completely fixed, so the length of wires among the logic elements cannot be controlled by designers. Because the difference between delay times arisen from physical variation is much smaller than that from the wire length, the responses obtained from different APUFs on different devices have small difference against a lot of challenges, *i.e.* low uniqueness.

In order to generate responses with high uniqueness on FPGAs, a novel technique that is called *Double Arbiter PUF* (DAPUF) [14] is proposed. The authors of [14] duplicate another APUF on neighboring SLICEs where the original APUF is implemented. They assume that a wire of duplicated APUF has almost the same length as the wire of the original APUF. 2-XOR APUF whose response is obtained by XORing 2-bit responses from two APUFs on the same FPGA are proposed in [3]. It has the same circuit costs as DAPUF, *i.e.*

978-83-60810-58-3/$25.00 © 2014, IEEE

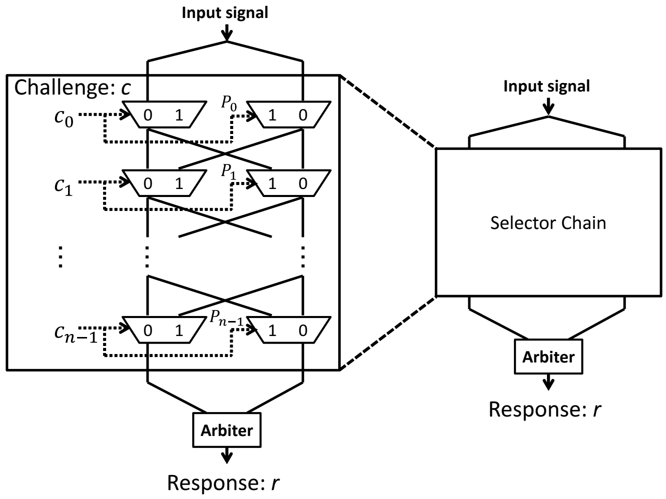


Fig. 1. Structure of conventional APUF

having two selector chains. In this paper, we call m-n APUF or DAPUF which has m selector chains and generates n-bit responses. For example, 2-XOR APUF is called *2-1 APUF*, and 2-bit output DAPUF is called *2-2 DAPUF*.

A. Our Contributions

In order to compare the structure of PUFs, we define a selector chain of conventional APUF as a building block as shown in Fig. 1. 2-2 DAPUF is different from 2-1 APUF in the wire assignment between the arbiter and output signals from the final selectors as shown in Figs. 2(a) and 3(b). Our two contributions in this paper are as follows:

- We introduce a new concept: *mode of operation for APUF* that is determined by a choice of the wire assignment. We compare PUFs that have two selector chains such as 2-2 DAPUF and 2-1 APUF and evaluate these PUFs on Virtex-5 FPGA regarding the uniqueness, randomness, and steadiness.
- We propose 3-1 DAPUF by using three selector chains, which is an improved version of 2-2 DAPUF. We compare it to conventional 3-XOR APUF, which have three selector chain. The evaluation results of these PUFs on Virtex-5 FPGA regarding the uniqueness show that 3-1 DAPUF generates responses with high uniqueness.

First, we evaluate four PUFs that have two selector chains. For a fair comparison, each PUF with the same-length response is compared. Therefore, 2-2 DAPUF is compared to *2-2 APUF*: two conventional APUFs as shown in Fig. 2(b). From our experimental results, we show that the uniqueness of responses from 2-2 DAPUF is higher than that from 2-2 APUF. We propose *2-1 DAPUF* by XORing 2-bit responses of 2-2 DAPUF as shown in Fig. 3(a), and compare it to 2-1 APUF. Our experimental results show that the uniqueness of responses from 2-1 DAPUF is approximately 41%, which is superior to 2-1 APUF.

One pair of the 2-2 DAPUFs has comparatively low uniqueness of responses because the proportion of 0s and 1s in responses (randomness) is still biased [14]. In order to eliminate the influence of the biased responses, we use another duplicated APUF, *i.e.* having three selector chains. In this paper, we propose *3-1 DAPUF* whose response is generated by XORing 6-bit responses of DAPUFs as shown in Fig. 4(a), for details to Sect. VI. Then, we compare 3-1 DAPUF to 3-XOR APUF that have three selector chains and generate 1-bit response. In this paper, we denote 3-XOR APUF as *3-1 APUF* as shown in Fig.4(b). Our experimental results show that the uniqueness of responses from 3-1 APUF is approximately 6%, which is still low. In contrast, the uniqueness of responses from 3-1 DAPUF is approximately 50%, which is much superior to that from 3-1 APUF.

We show that we can improve the uniqueness by using the new mode of operation for APUF and using responses obtained by XORing responses from more duplicated arbiter on Virtex-5 FPGAs.

B. Organization of This Paper

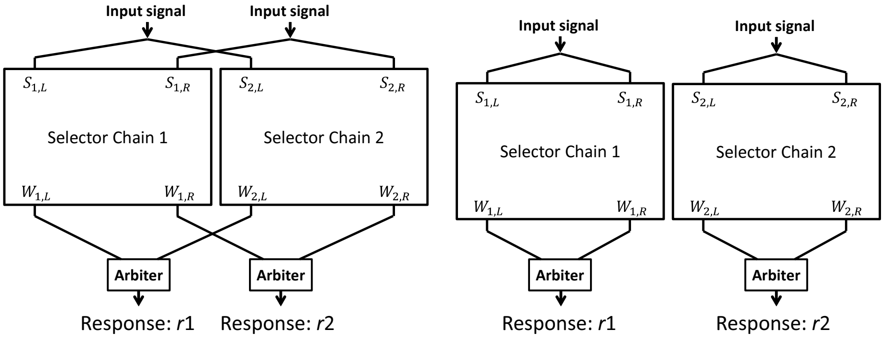
Organization of this paper is following. Section II gives previous work. Section III mentions the motivation of this work. Section IV shows experimental setup such as environment, and evaluation indicators. Moreover, we introduce the experimental results of conventional APUF evaluated by these indicators. Section V compares DAPUF to other APUFs which have two selector chains by using the indicators. Section VI proposes 3-1 DAPUF and compares it to 3-1 APUF, which have three selector chains by using the indicators. Finally, Sect. VII concludes this work.

II. Arbtier PUF

Arbiter PUF (APUF) is one of the delay-based PUFs that use the difference between delay times of two signals. APUF has left and right selector pairs connected in series as shown in Fig. 1. Each bit of n-bit challenge c corresponds to a selection input c_i to the selector pair P_i ($0 \leq i < n$). After the challenge is determined, an input signal is supplied to the first selector pair P_0 at the same timing. For the case of $c_i = 1$, the left (right) selector in P_i is cross-connected to the right (left) selector in P_{i+1}, respectively. For the case of $c_i = 0$, the left (right) selector in P_i is straightly connected to the left (right) selector in P_{i+1}. This means that an input signal reaches through various paths depending on the value of the challenge. A 1-bit response is determined by which signal reaches an arbiter faster than the other. The response is strongly affected by the propagation path of the input signal, *i.e.* the challenge.

In APUFs, it is possible to increase the number of challenge bits easily by using more selector pairs [3]. APUF with n selector pairs has 2^n challenges. It is known that APUF can be modeled and simulated by building software models and programs based on the relation between challenges and responses [11]. This modeling against APUFs makes it possible for an attacker to predict responses for almost all challenges [15].

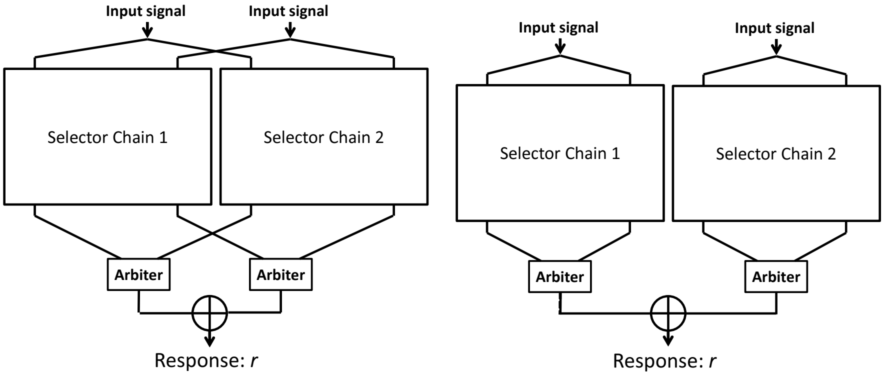
In order to prevent this modeling prediction, N-XOR APUFs have been proposed, where N-bit responses obtained from N APUFs are XORed into 1-bit response [3].



(a) 2-2 DAPUF (DAPUF)

(b) 2-2 APUF

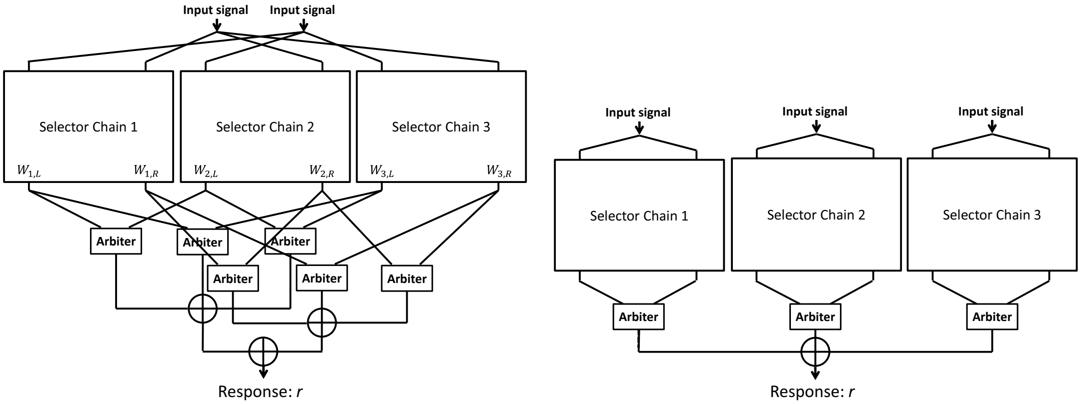
Fig. 2. Structure of 2-2 PUFs



(a) 2-1 DAPUF

(b) 2-1 APUF (2-XOR APUF)

Fig. 3. Structure of 2-1 PUFs



(a) 3-1 DAPUF

(b) 3-1 APUF

Fig. 4. Structure of 3-1 PUFs

III. Motivation

Previous work of [12][13] reports that APUFs implemented on Xilinx Virtex-5 FPGAs generate responses with low uniqueness. DAPUF has been proposed in order to generate responses with high uniqueness even on such FPGAs [14]. Although almost all of DAPUFs improve the uniqueness of responses, the authors of [14] clarify that one pair of the DAPUFs has comparatively low-unique responses, which should be solved.

In this paper, we divide APUF into three components as shown in Fig. 1.

- An input signal part

- A selector chain part (building block)
- An arbiter part

Two wires are input to the top of the building block, and two wires are output from the bottom of it as shown in Fig. 1. The two wires from the top connect the input wires to 1-bit selector pair P_0, and that from the bottom connect output wires from n-bit selector pair P_{n-1} to the arbiter. The input signal is provided into the selector pair P_0 at the same time through the two wires from the top. Depending on the challenge value, the right input signal reaches the arbiter as the left or right signals.

To divide APUF into three parts enables us to clarify the difference between 2-2 DAPUF and 2-2 APUF: assignment of the input signal part and the arbiter part to the selector chain part, *i.e.* the mode of operation for APUF. We evaluate both PUFs with the same circuit costs as for the uniqueness, randomness, and steadiness of the PUFs and make comparison of these.

IV. Preliminary for Our Experiment

A. Experimental Environment

In this paper, APUF was implemented on a Xilinx Virtex-5 FPGA (XC5VLX30) [16] on SASEBO-GII (Side-channel Attack Standard Evaluation Board) [17]. Xilinx ISE 13.2 and Xilinx PlanAhead 13.2 were used for logic synthesis and floorplanning, respectively. We designed the APUF with 64-bit challenges so that wire length difference between two paths could be minimized by using Static Timing Analysis (STA), according to [9]. The selector pair (P_i) is located on the SLICE pair (X14Y(76-i), X15Y(76-i)). An input signal is supplied from the register with the equal distances from 1-bit selector pair P_0. A response is generated from another register (*i.e.* the arbiter in Fig. 1) with the equal distance from selector pair P_{n-1}.

B. Evaluation Indicators

Several performance indicators of PUFs are introduced in [18].

1) Uniqueness: When the same challenge is given to different PUFs, the responses should be completely different from one another. We use the value of SC Inter-HD (Same-Challenge Inter-Hamming Distance) [13] divided by the response bit length as the indicator of uniqueness. SC Inter-HD is calculated as the average of the Hamming distances between two responses obtained from different two PUFs for the same challenge. If the value of the uniqueness is close to 50%, we regard the uniqueness of the responses to be high in this paper.

2) Randomness: The proportion of 0s and 1s in responses should be equal. In this paper, we define the randomness of responses as the average number of 1s in responses (for randomly chosen challenges) divided by the responses bit length. The randomness of responses is 50% ideally.

3) Steadiness: When the same challenges are given to a PUF for repeated measurements, all of the responses should be the same. We use the value of SC Intra-HD (Same-Challenge Intra-Hamming Distance) [13] divided by the response bit length as the indicator of steadiness. SC Intra-HD is calculated as the average of the Hamming distances between arbitrary two responses for the same challenge. If the steadiness is close to 0%, we regard the steadiness of responses to be ideal.

TABLE I. Uniqueness of Conventional 1-1 APUF

Pair of FPGAs	uniqueness[%]
A with B	4.72
B with C	4.96
C with A	4.44

TABLE II. Randomness and Steadiness of Conventional 1-1 APUF

FPGA	randomness[%]	steadiness[%]
A	53.81	0.76
B	56.53	0.83
C	54.00	0.45

C. Results of Conventional Arbiter PUFs on FPGAs

Previous work [12][13] shows that APUFs implemented on Xilinx Virtex-5 and Kintex-7 FPGAs generate responses with low uniqueness, experimentally. In this section, we implement APUFs on Virtex-5 FPGAs and evaluate these PUFs for preliminary experiments.

First, we evaluate the uniqueness of 5000-bit responses obtained from APUFs on FPGA-A, FPGA-B, and FPGA-C. Table I shows the uniqueness of responses obtained from conventional APUFs on Virtex-5 FPGAs. The uniqueness is less than 5%, while 50% ideally. This means that physical variation of each PUF cannot be extracted as the uniqueness of responses.

Second, we evaluate the randomness of 2^{16} responses. The results are shown in the left part of Table II. The randomness is around 50%, which is the ideal. However, the authors in [19] report that the most of responses from APUFs on Virtex-5 FPGAs become either 1 or 0 with particular challenges. They evaluated 2^{16} responses from 64-bit APUFs for the challenges where $c_0 = c_1 = \cdots = c_7 = 1$ are fixed and $c_8, c_9, \ldots, c_{63}$ are randomly chosen. Under the condition that the Hamming weight is odd, the proportion of 1s in responses is approximately 80%, and under the condition that the Hamming weight is even, the proportion of 1s in responses is approximately 30% [19]. If the difference between the delay times of the two signals in selector chains is critically large, the responses are determined by whether the signal having larger delay than the other reaches right or left wire input to the arbiter. The two signals of conventional APUFs are crossed when $c_i = 1$ $(0 < i < 64)$. Therefore, whether Hamming weight of c_i is odd or even determines whether the signal having larger delay is supplied to the right or left wire. Under the condition of randomly chosen challenge, the proportion of 1s and 0s in responses become approximately 50%. In the result, the randomness of the responses from conventional APUFs on FPGAs comes to 50% regardless of the proportion of 1s and 0s in responses for particular challenges.

Finally, we evaluate the steadiness of 128-bit responses. The results are shown in the right part of Table II. The steadiness is calculated with 128-bit response for fixed challenges for 128 repeated measurements. The challenges are randomly chosen. The steadiness is less than 1% among all pairs of FPGAs, which is nearly ideal as PUF. It shows that conventional APUFs generate the same responses for the same challenges. However, it is based on low uniqueness of the responses.

We reconfirm that the conventional APUFs on Virtex-5 FPGAs have low unique responses, which are not enough to perform as ideal PUFs.

V. DAPUFs *v.s.* APUFs of $m = 2$

A. *Modes of Arbiter Operation*

1) Double Arbiter PUF: It is discussed that the length of the two wires in APUF are not equal at all [19]. In [14], it is suggested that the reason why conventional APUFs on Virtex-5 FPGAs generate low-unique responses is the unequal length of the two wires. Since the difference between delay times arisen from physical variations is much smaller than that from the signal propagation on the wire, the physical variations of each PUF cannot be found in responses, *i.e.* the uniqueness of responses become low. In order to generate responses with high uniqueness, it is proposed that a novel technique called *Double Arbiter PUF* (2-2 DAPUF) [14]. 2-2 DAPUF is designed for the purpose of equalizing the length of the two wires. Figure 5 shows the floorplanning for Xilinx PlanAhead. As illustrated in Fig. 5, the length of wires (1) and (2) seems equal but different precisely. Therefore, the authors duplicate another APUF on neighboring SLICEs where the original APUF is implemented. The authors expect that wire (1) has the almost the same length as wire (3) because both cell-pairs (1a,1b) and (3a,3b) are symmetrically located on the neighboring SLICEs.

Figure 2(a) shows the mode of operation for APUF of 2-2 DAPUF. Let $S_{i,L}$ and $S_{i,R}$ be the left and right wires which are inputs to the first selector pairs P_0 in Selector Chain i $(1 \leq i \leq 2)$, respectively, as shown in Fig. 2(a). The signals on $S_{1,L}$ and $S_{2,L}$ are supplied to Selector Chain 1 and 2 at the same time. Let $W_{i,L}$ and $W_{i,R}$ be the left and right wires, respectively. They are outputs from the n-th selector pairs P_{n-1} in Selector Chain i $(1 \leq i \leq 2)$, respectively. The signal on $S_{1,L}$ reaches $W_{1,*}$ $(* \in \{L, R\})$ and the signal on $S_{2,L}$ reaches $W_{2,*}$ regardless of the value of the challenge. Similarly, the signals on $S_{1,R}$ and $S_{2,R}$ are supplied to the Selector Chain 1 and 2 at the same time. Two 1-bit responses $r1$ and $r2$ are generated from two pairs of wires ($W_{1,L}$, $W_{2,L}$) and ($W_{1,R}$, $W_{2,R}$), respectively. Therefore, the signals on $S_{1,L}$ and $S_{2,L}$ are not crossed even if $c_i = 1$ $(0 < i < 64)$. The signals on $S_{1,R}$ and $S_{2,R}$ are also not crossed even if $c_i = 1$.

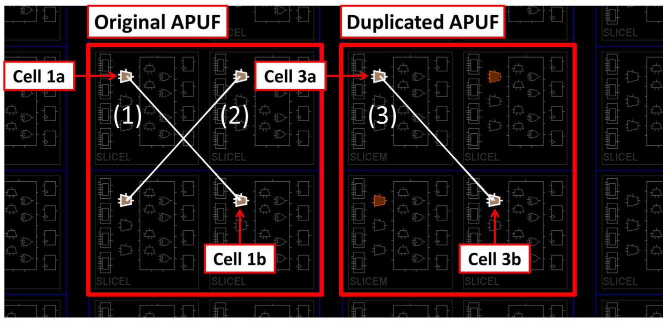


Fig. 5. Xilinx PlanAhead floorplanning

2-2 DAPUF generates 2-bit responses. We propose *2-1 DAPUF* that generates 1-bit responses obtained by XORing the 2-bit responses as shown in Fig. 3(a).

2) 2-XOR Arbiter PUF: We consider two PUFs that have the same circuit costs as DAPUFs of $m = 2$, *i.e.* having two selector chains. A straight forward example of APUFs of $m = 2$ is just two APUFs generating responses, *i.e.* *2-2 APUF* as shown in Fig. 2(b). The signals on $S_{1,L}$ and $S_{1,R}$ are supplied to Selector Chain 1 at the same time. It depends on the value of challenges whether signal on $S_{1,L}$ or $S_{1,R}$ reaches $W_{1,L}$ or $W_{1,R}$. Similarly, the signals on $S_{2,L}$ and $S_{2,R}$ are supplied to the Selector Chain 2 at the same time. Two 1-bit responses $r1$ and $r2$ are generated from two pairs of wires ($W_{1,L}$, $W_{1,R}$) and ($W_{2,L}$, $W_{2,R}$), respectively. Therefore, the signals on $S_{1,L}$ and $S_{1,R}$ are crossed when $c_i = 1$ $(0 < i < 64)$. The signals on $S_{2,L}$ and $S_{2,R}$ are also crossed when $c_i = 1$. This is different from 2-2 DAPUFs in which the two signals are not crossed. We compare 2-2 APUF to 2-2 DAPUF and discuss the uniqueness, randomness, and steadiness. The difference of the wire connections has influence on its results, as mentioned in next section.

These 2-bit responses can be XORed into 1-bit responses: 2-1 APUF (2-XOR APUF) as shown in Fig. 3(b). We compare 2-1 APUF to 2-1 DAPUF according to the uniqueness, randomness, and steadiness.

B. *Results*

The results of the uniqueness and randomness of 2-2 DAPUF are from [14].

First, we evaluate the uniqueness of 5000-bit responses obtained from DAPUFs and APUFs of $m = 2$. The results are shown in Table III. The uniqueness of responses from 2-2 DAPUFs introduced in [14] is higher than that from 2-2 APUFs. However, the uniqueness of responses $r1$ between FPGA-A and FPGA-B is approximately 9%, which is comparatively lower than that of others. The reason for this is discussed along with the results of the randomness, as mentioned in the next paragraph. The uniqueness of responses from 2-1 DAPUFs is approximately 42%, which is much higher than that from 2-1 APUFs.

Second, we evaluate the randomness of 2^{16} responses. The results are shown in Table IV. In the following, we discuss the reason why 2-1 APUFs have low randomness although conventional APUFs have high randomness. From Table III, two conventional APUFs on Virtex-5 FPGAs generate low-unique responses: one PUF generates the same responses as the other PUF for many challenges. Therefore, 2-1 APUFs whose 1-bit response is obtained by XORing the responses of the two PUFs have low randomness obviously because the response becomes 0 when the same values are XORed. However, it is worth mentioning that the high randomness value is just superficial as mentioned in Sect. IV C. The reason why one pair of the 2-2 DAPUFs has comparatively low uniqueness of responses can be explained by low randomness of responses $r1$ on FPGA-A and FPGA-B. Here, we discuss one of the reasons of this low randomness.

TABLE III. UNIQUENESS [%] OF APUFS AND DAPUFS OF $m = 2$

	2-2 APUF		2-2 DAPUF		2-1 APUF	2-1 DAPUF
Pair of FPGAs	$r1$	$r2$	$r1$ [14]	$r2$ [14]	r	r
A with B	4.72	4.40	8.76	37.62	4.96	41.36
B with C	4.96	5.94	61.26	56.64	5.62	49.70
C with A	4.44	5.58	66.90	36.98	5.58	48.06

TABLE IV. RANDOMNESS [%] OF APUFS AND DAPUFS OF $m = 2$

	2-2 APUF		2-2 DAPUFs		2-1 APUF	2-1 DAPUF
FPGA	$r1$	$r2$	$r1$ [14]	$r2$ [14]	r	r
A	53.81	56.92	1.72	54.20	6.32	55.19
B	56.53	56.25	7.68	25.62	4.72	31.40
C	54.00	54.04	68.50	80.22	4.93	50.63

TABLE V. STEADINESS [%] OF APUFS AND DAPUFS OF $m = 2$

	2-2 APUF		2-2 DAPUF		2-1 APUF	2-1 DAPUF
FPGA	$r1$	$r2$	$r1$	$r2$	r	r
A	0.76	0.67	0.67	7.11	1.43	7.79
B	0.83	0.73	4.70	6.52	1.36	11.22
C	0.45	0.08	2.96	7.24	0.52	10.05

2-2 APUFs

Even if the deterministic difference between delay times of the two signals is produced, the randomness seems high. Because it depends on Hamming weight of challenge whether the signal having lager delay than the other is supplied to left or right wire input to the arbiter.

2-2 DAPUFs

In contrast, if the deterministic difference between delay times of the two signals is produced, the randomness can become low. Because it does not depend on Hamming weight of challenge whether the signal having larger delay is supplied to left or right input to the arbiter. This is caused by the signals on $S_{1,L}$ and $S_{2,L}$ ($S_{1,R}$ and $S_{2,R}$) are not crossed, as mentioned above.

Finally, we evaluate the steadiness of 128-bit responses. The results are shown in Table V. Almost all of the implemented 2-1 DAPUFs generate responses with lower steadiness than 2-1 APUFs. We consider that the large difference of delay times arisen from the imbalance wire length, as mentioned in Sect. IV C, results in high steadiness, but in contrast, low uniqueness. There is a trade-off between steadiness and uniqueness.

VI. 3-1 DAPUF *v.s.* 3-1 APUF

A. Modes of Arbiter Operation

1) 3-1 Double Arbiter PUF: One pair of the 2-2 DAPUFs has comparatively low uniqueness of responses because they have still biased responses [14]. In order to eliminate the influence of the biased responses, we use the following technique. We duplicate another APUF, *i.e.* having three selector chains, and generates multiple responses. Even if each of these responses is biased, we can obtain a less-biased response by XORing these responses. Let $W_{i,L}$ and $W_{i,R}$ be the left and right wires which are outputs from the n-th selector pairs P_{n-1} in Selector Chain i ($1 \leq i \leq 3$), respectively. We use two wires chosen from three left wires: $W_{1,L}$, $W_{2,L}$, $W_{3,L}$ to generate three 1-bit responses as shown in Fig. 4(a). Similarly, we use two out of three right wires: $W_{1,R}$, $W_{2,R}$, $W_{3,R}$. Therefore, the left and right wires can generate six 1-bit responses in total. In this paper, we consider *3-1 DAPUF*: having three selector chains and generating a 1-bit response by XORing the six 1-bit responses.

2) 3-XOR Arbiter PUF: 3-1 APUF (3-XOR APUF) generates 1-bit responses obtained by XORing 3-bit responses from three conventional APUFs as shown in Fig. 4(b). The circuit costs of a 3-1 APUF are the same as that of a 3-1 DAPUF. They have three selector chains and generate 1-bit responses. We compare 3-1 DAPUF to 3-1 APUF according to the uniqueness, randomness, and steadiness.

B. Results

First, we evaluate the uniqueness of 5000-bit responses obtained from APUFs and DAPUFs of $m = 3$. The results are shown in Table VI. The uniqueness of responses from 3-1 DAPUFs is 50±1%, which is very close to the ideal results. In contrast, the uniqueness of responses from 3-1 APUFs is approximately 6%, which is much inferior to that from 3-1 DAPUFs. Further, 3-1 DAPUFs generate responses with high uniqueness among all pairs of FPGAs although one pair of the 2-2 DAPUFs has comparatively low uniqueness of responses. This means that we can eliminate the influence of the biased responses from the DAPUFs. The uniqueness of responses from 3-1 APUFs does not improve similarly to 2-1 APUFs. We consider that this is caused by the low uniqueness of each response from three conventional APUFs.

Second, we evaluate the randomness of 2^{16} responses. The results are shown in Table VII. The randomness of responses generated from 3-1 DAPUFs is around 50%, which is almost ideal. Further, the randomness of responses from 3-1 DAPUFs is more improved than that from 2-1 DAPUFs. The randomness value of responses from 3-1 APUFs seems high since the number of XORing responses are three (odd number).

Finally, we evaluate the steadiness of 128-bit responses. The results are shown in Table VIII. The steadiness of responses from 3-1 DAPUFs is approximately 12%, which is

TABLE VI. UNIQUENESS [%] OF APUF AND DAPUF OF $m = 3$

Pair of FPGAs	3-1 APUF	3-1 DAPUF
A with B	5.96	50.60
B with C	6.76	51.34
C with A	6.32	48.78

TABLE VII. RANDOMNESS [%] OF APUF AND DAPUF OF $m = 3$

FPGA	3-1 APUF	3-1 DAPUF
A	54.88	55.68
B	55.05	52.54
C	54.96	53.59

TABLE VIII. STEADINESS [%] OF APUF AND DAPUF OF $m = 3$

FPGA	3-1 APUF	3-1 DAPUF
A	1.43	14.11
B	1.36	10.93
C	0.74	10.35

inferior to that from 3-1 APUFs. We consider that one of the reasons is a trade-off between the steadiness and uniqueness.

We show the summary of the uniqueness, randomness, and steadiness for 1-1, 2-1, and 3-1 PUFs in Figs. 6, 7, and 8, respectively. The uniqueness of responses from APUFs and DAPUFs is improved with the increasing the number of selector chains. It is clear that the uniqueness of responses from DAPUFs using the new mode of operation is superior to APUFs. The randomness of responses from DAPUFs is also improved with that. That from only 2-1 APUFs having the even selector chains is lower than the other. However, the high randomness of 1-1 and 3-1 APUFs is just superficial as mentioned above section. In contrast, the steadiness of responses become low, *i.e.* the value of steadiness is high, with that.

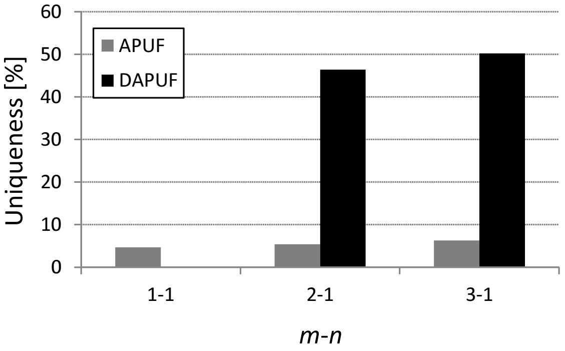


Fig. 6. Summary of uniqueness for m-n APUFs and DAPUFs

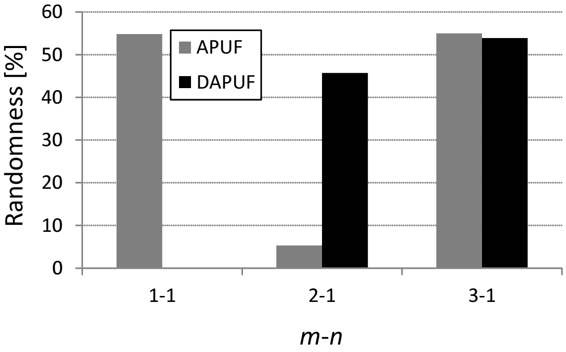


Fig. 7. Summary of randomness for m-n APUFs and DAPUFs

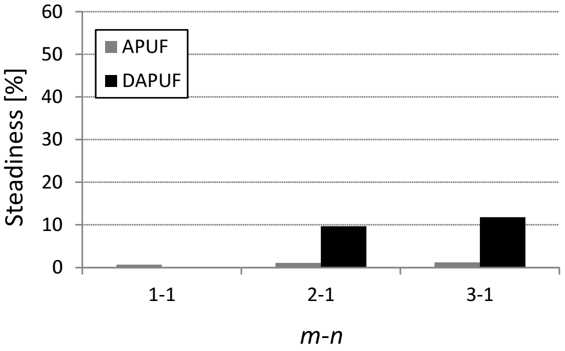


Fig. 8. Summary of steadiness for m-n APUFs and DAPUFs

We show that we can improve the uniqueness and randomness by using the new mode of operation for APUF and using responses obtained by XORing responses from more duplicated selector chains on Virtex-5 FPGAs.

VII. Conclusion and Future Work

2-2 DAPUF was proposed in order to generate responses with high uniqueness in previous work. In this paper, we introduced new concept: *mode of operation for APUF* that was determined by the connection method of the wires to arbiter. We compared DAPUFs and APUFs of $m = 2$ that have two selector chains such as 2-2 DAPUF and 2-1 APUF and evaluated these PUFs regarding the uniqueness, randomness, and steadiness. Further, we proposed 3-1 DAPUF by using three selector chains, which was improved version of DAPUF. We compare 3-1 DAPUF to 3-1 APUF, which have three selector chains, and evaluate these PUFs. From our experimental results, the uniqueness of responses from 3-1 DAPUFs was approximately 50%, which was much superior to that from 3-1 APUFs. On general FPGAs, we showed that we could improve the uniqueness and randomness by using the new mode of operation for APUF and using responses obtained by XORing responses from more duplicated selector chains.

The future work of this study is to implement *4-1* or *5-1 DAPUF* and to evaluate the responses from these PUFs according to the uniqueness, randomness, and steadiness. Further, we compare these PUFs to 2-1 and 3-1 DAPUFs by using the results.

References

[1] P. S. Ravikanth, "Physical one-way functions," Ph.D. dissertation, 2001, http://dx.doi.org/10.1126/science.1074376.

[2] P. Tuyls, B. Skoric, and T. Kevenaar, *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting.* Springer-Verlag New York, Inc., 2007, http://dx.doi.org/10.1007/978-1-84628-984-2.

[3] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Proceedings of DAC*, 2007, pp. 9–14, http://dx.doi.org/10.1145/1278480.1278484.

[4] B. Gassend, D. Lim, D. E. Clarke, M. van Dijk, and S. Devadas, "Identification and authentication of integrated circuits." *Concurrency and Computation: Practice and Experience*, pp. 1077–1098, 2004, http://dx.doi.org/10.1002/cpe.805.

[5] Z. S. Paral and S. Devadas, "Reliable and efficient PUF-based key generation using pattern matching." in *in Proceedings of HOST.* IEEE Computer Society, 2011, pp. 128–133, http://dx.doi.org/10.1109/HST.2011.5955010.

[6] H. Handschuh, G. J. Schrijen, and P. Tuyls, "Hardware intrinsic security from physically unclonable functions." in *Towards Hardware-Intrinsic Security*, 2010, pp. 39–53, http://dx.doi.org/10.1007/978-3-642-14452-3_2.

[7] R. Maes, "Physically Unclonable Functions - Constructions, Properties and Applications," in *Springer*, 2013, http://dx.doi.org/10.1007/978-3-642-41395-7.

[8] J. H. Anderson, "A PUF design for secure FPGA-based embedded systems," in *Proceedings of ASP-DAC*, 2010, pp. 1–6, http://dx.doi.org/10.1109/ASPDAC.2010.5419927.

[9] K. Seki, Y. Hori, and H. Imai, “Implementation and Evaluation of Physical Unclonable Function on SASEBO-GII (in Japanese),” in *Symposium record of SCIS*, 2010.

[10] S. Saifullah, A. Khawaja, Hamza, Arsalan, Maryam, and Anum, “Keyless car entry through face recognition using FPGA,” in *Proceedings of FITME*, 2010, pp. 224–227, http://dx.doi.org/10.1109/FITME.2010.5654862.

[11] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, “Extracting secret keys from integrated circuits,” *IEEE Trans. Very Large Scale Integr. Syst.*, pp. 1200–1205, 2005, http://dx.doi.org/10.1109/TVLSI.2005.859470.

[12] A. Maiti, V. Gunreddy, and P. Schaumont, “A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions,” in *Embedded Systems Design with FPGAs*, 2013, pp. 245–267, http://dx.doi.org/10.1007/978-1-4614-1362-2_11.

[13] Y. Hori, T. Katashita, and K. Kobara, “Performance Evaluation of Physical Unclonable Functions on Kintex-7 FPGA (in Japanese),” in *IEICE Technical report of RECONF*, 2013.

[14] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, “A Study on Uniqueness of Arbiter PUF Implemented on FPGA (in Japanese),” in *Symposium record of SCIS*, 2014.

[15] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, “Modeling Attacks on Physical Unclonable Functions,” in *Proceedings of CCS*, 2010, pp. 237–249, http://dx.doi.org/10.1145/1866307.1866335.

[16] XILINX, “Virtex-5 FPGA User Guide,” http//www.xilinx.com/support/documentation/user_guides/ug190.pdf.

[17] National Institute of Advanced Industrial Science and Technology, “Side-channel Attack Standard Evaluation Board (SASEBO),” http://www.risec.aist.go.jp/project/sasebo/.

[18] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, “Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs,” in *Proceedings of ReConFig*, 2010, pp. 298–303, http://dx.doi.org/10.1109/ReConFig.2010.24.

[19] T. Machida, T. Nakasone, and K. Sakiyama, “Evaluation Method for Arbiter PUF on FPGA and Its Vulnerability (in Japanese),” in *IEICE Technical report of ISEC*, 2013.