Implementation of Double Arbiter PUF and Its Performance Evaluation on FPGA[*]

Takanori Machida† Dai Yamamoto†† Mitsugu Iwamoto† Kazuo Sakiyama†
†The University of Electro-Communications ††Fujitsu Laboratories Ltd.

Abstract— Low uniqueness and vulnerability to machine-learning attacks are known as two major problems of Arbiter-Based Physically Unclonable Function (APUF) implemented on FPGAs. In this paper, we implement Double APUF (DAPUF) that duplicates the original APUF in order to overcome the problems. From the experimental results on Xilinx Virtex-5, we show that the uniqueness of DAPUF becomes almost ideal, and the prediction rate of the machine-learning attack decreases from 86% to 57%.

I. Introduction

Recently, fake IC (Integrated Circuit) chips are distributed to market. Not only authorized chip makers are damaged by the fake IC chips, but also the products embedding these chips may cause a serious accident. PUF (Physically Unclonable Function), one of the anti-counterfeit technologies, is being focused, which can be used for a secure device authentication.

In general, a server (verifier) authenticates a device (prover), by checking whether or not the server can obtain a correct response for a challenge sent from the device. PUF is a physical function that outputs a unique response for a challenge. Since the response is derived from physical variations of the PUF device, it is believed that making a clone is significantly difficult.

However, the layout in designing APUF is significantly difficult on both ASIC (application specific integrated circuit) and FPGA (Field Programmable Gate Array) because it is required that a pair of wires to be equally routed. Since the logic elements called SLICE on FPGA are fixed in position, the equal-length wiring is harder than the case for ASICs. In fact, previous work reports that APUF on Xilinx Virtex-5 and Kintex-7 FPGAs generate low-unique responses among devices [3] [4]. Our previous work also clarifies that the low-unique responses on Xilinx Virtex-5 FPGA is due to the difficulty of equal-length wiring on Xilinx Virtex-5 FPGA [2]. Accordingly, we introduce a new APUF called DAPUF that enhances the uniqueness of APUF [1]. However, there exist no results for prediction attacks on DAPUF. In the prediction attacks, an attacker tries to predict the responses from APUF by using a machine-learning model [5]. Therefore, in this paper, we aim at implementing a high-uniqueness PUFs on FPGAs that has a high-tolerance against machine-learning attacks.

Signal that changes from 0 to 1 is supplied to the input wires of APUF at the same timing, and the two signal transitions are competed. The structure of APUF consists of two selector chains and an arbiter[1] that determines which signal is faster than the other, as shown in the left part of Fig. 1 (denoted as conventional APUF). The sequence of inputs denoted by $c_0c_1 \cdots c_{n-1}$ is called challenge of APUF, where c_i selects the outputs of the $(i+1)$-th selector $(0 \leq i \leq n-1)$. When $c_i = 1$, the signal propagations of the two output signals of the $(i-1)$-th selectors are physically crossed and provided to the inputs of the i-th selectors. When $c_i = 0$, they are provided without crossing. Hence, the path of signals is determined by the challenge. If the length of these two wires does not equal, the delay time difference between the two signals could become larger as the signals go through selectors. As a result, the delay times arising from the physical variations of devices tend to become smaller than the delay time difference by the two signal transitions. In a worst case, the responses from any devices can become the same for a specific challenge.

Suppose that the delay time features of physically copied PUF modules are similar even on FPGA. Then, designing 2-1 DAPUF as shown in the middle part of Fig. 1, the delay of the signal from each duplicated APUF has a similar delay path, and we can escape the wiring problem. We also consider 3-1 DAPUF that has three selector chains, which is illustrated in the right part of Fig. 1.

II. Experiments and results

A. Experimental environment

We implement 2-1 DAPUF and 3-1 DAPUF on Xilinx Virtex-5 FPGAs, as well as the conventional APUF. The three FPGA devices are used for the experiments. Xilinx ISE 13.2 and PlanAhead 13.2 are used for the logic synthesis and the floorplanning, respectively. Our measurement system is shown in Fig. 2.

B. Evaluation metrics

Uniqueness When the same challenge is given to all devices, it is required for each device to output a unique

[*] Although this paper is based on [1, 2], we evaluate a resistance against machine-learning attacks as shown in Table I.

[1] We use an SR latch.

978-1-4799-7792-5/15/$31.00 ©2015 IEEE

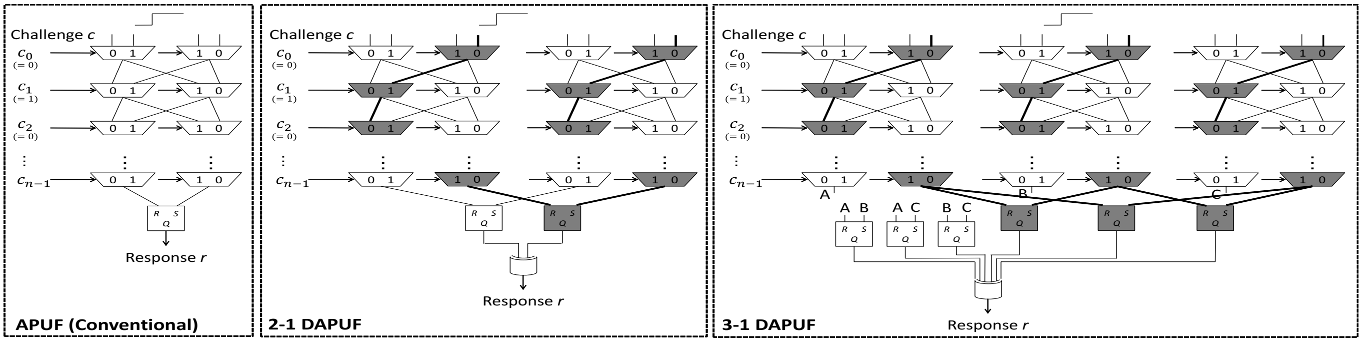

Fig. 1. Structures of APUF, 2-1 DAPUF and 3-1 DAPUF

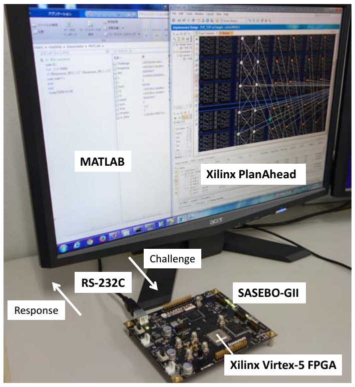

Fig. 2. Measurement system

response. When 5000-bit challenge is given to two devices, a pair of 5000-bit responses is generated. We measure the Hamming distance between the responses. In this paper, uniqueness is defined as the Hamming distance divided by the length of response, 5000. An ideal uniqueness is 50%, which means that the Hamming distance of responses from any PUFs is different in a half.

Prediction rate We evaluate a resistance of PUF against machine-learning attacks. Support Vector Machine (SVM) builds a model by using 1000 challenge–response pairs as training data, which are randomly chosen. SVM predicts the responses from the target PUF for randomly chosen 10000 challenges. The prediction rate is considered as an evaluation metrics of resistance against the machine-learning attacks. Instead of giving the challenges themselves to SVM, they are transformed according to [5]. An ideal prediction rate is 50%, which indicates that the prediction is impossible.

Cost The hardware cost of PUFs is defined as the number of the selector chains. Lower cost is obviously better.

III. Performance comparison

The evaluation results for the original APUF, 2-1 DAPUF, and 3-1 DAPUF are summarized in Table I. The uniqueness of the original APUF on FPGA is low as reported in [3] and [4]. In contrast, the uniqueness of 2-1 and 3-1 DAPUF are approximately 46.4% and 50.2%[2], respectively. Thus, the uniqueness is improved drastically.

TABLE I
Evaluation results of APUF, 2-1 DAPUF and 3-1 DAPUF

Indicators	Ideal	APUF	APUF	2-1	3-1
		[4]‡		DAPUF	DAPUF
Uniqueness [%]	50	3.6	4.7	**46.4**	**50.2**
Pred. rate [%]	50	N/A	86.3	69.0	**57.0**
Cost (ratio)	1	1	1	2	3

‡ These values are converted from the results on Xilinx Kintex-7 [4]

It can be said that the wiring problem is escaped by duplicating the selector chains. As for a resistance against the machine-learning attacks, 86% responses from APUF can be predicted. The prediction rate for 2-1 DAPUF is 69%, which is not enough but improved from APUF. In the case for 3-1 DAPUF, the prediction rate is 57%, which can be considered secure to some extent. It is considered as the secondary effect of the high uniqueness. On the other hand, the hardware cost is proportionally-increased. Actually, the numbers of occupied SLICEs in our floorplaning are 177, 303 and 436, respectively for APUF, 2-1 DAPUF, and 3-1 DAPUF.

IV. Summary and Future Work

We implemented 2-1 and 3-1 DAPUF on Xilinx Virtex-5 FPGA, and evaluated the uniqueness and resistance against the machine-learning attacks. From the experimental results, we confirmed that 2-1 and 3-1 DAPUF generated responses with higher uniqueness than that of the conventional APUF. The prediction rate of 3-1 DAPUF improved from 86% to 57%. The future work is to investigate the process tolerance. Further, we will implement 4-1 DAPUF in order to improve the performances.

References

[1] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, “A New Mode of Operation for Arbiter PUF to Improve Uniqueness on FPGA (in press),” in *Proceedings of FedCSIS*, 2014.

[2] 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.

[3] A. Maiti, V. Gunreddy, and P. Schaumont, “A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions,” in *Embedded Systems Design with FPGAs*, pp. 245–267, 2013.

[4] 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.

[5] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, “Modeling Attacks on Physical Unclonable Functions,” in *Proceedings of CCS*, pp. 237–249, 2010.

[2] The average of (41.3%, 49.7%, and 48.1%) and (50.6%, 51.3%, and 48.8%)