ADJUSTABLE ARBITER PHYSICAL UNCLONABLE FUNCTION WITH FLEXIBLE RESPONSE DISTRIBUTION

Jing Ye, Xiaowei Li*, Huawei Li, and Yu Hu*
[1] State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
*Corresponding Author's Email: yejing@ict.ac.cn

ABSTRACT

This paper proposes an adjustable arbiter PUF with flexible response distribution, which expands the application areas of PUFs. In comparison with existing works, it reduces the test time of reliable CRPs by about 3.3×10^4X, reduces the hardware area of ECC by at least 90%, and reduces the over-testing cases of reliable CRPs by 57%.

INTRODUCTION

The Physical Unclonable Function (PUF) is an emerging hardware security primitive [1]. It exploits the random process variations to produce particular responses for input challenges, which are called the Challenge-Response Pairs (CRPs). Even with the same design, different manufactured PUFs will have different CRPs, which are hard to predict before manufacturing, to control during manufacturing, and to physically clone after manufacturing. Due to these advantages, PUFs have broad application prospects in the field of hardware security [2~4].

In general, PUFs can be classified into two categories: the weak PUF such as the SRAM PUF [5] and the strong PUF. A typical strong PUF is the arbiter PUF [6]. It compares the delays of two paths to produce a response bit. Each path is consisted of path segments selected by a challenge.

Existing arbiter PUFs have several issues:

Issue I: Single arbiter PUF is vulnerable to machine learning based modeling attacks [7, 8], so different variants of arbiter PUFs have been proposed [9, 10]. XOR arbiter PUF logically XORs response bits of multiple arbiter PUFs as the final response bit. With sufficient number of arbiter PUFs employed, the XOR arbiter PUF is able to resist certain modeling attacks. However, the higher security, the lower reliability. Last year, [11] solved this issue by proposing a method to select reliable CRPs of a highly secure XOR arbiter PUF for usage. The issue is that each challenge is input to the PUF for 10^5 times to evaluate its reliability, which costs a long test time.

Issue II: For the applications such as [3], where selecting CRPs is not allowed, Error Correction Code (ECC) circuit is needed, which costs a large hardware area.

Issue III: Last year, [12] proposed an online reliability checker. However, due to process variations, over-testing may happen, where reliable CRPs are considered as unreliable ones. Moreover, when it is implemented in FPGA, this issue becomes more seriously due to the limitation of placement and routing.

This paper proposes an adjustable arbiter PUF with flexible response distribution. The major contributions include:

1) The test time of reliable CRPs is reduced by about **3.3×10^4X**, in comparison with [11].

2) The hardware area of ECC is reduced by at least **90%** for applications such as [3].

3) The over-testing cases of reliable CRPs are reduced by **57%**, in comparison with [12].

4) The distribution of response bits 0s and 1s is adjustable, which expands the application areas of PUFs. Examples of using it to construct a stealthier hardware Trojan and to protect scan test data are illustrated.

PROPOSED DESIGN

The proposed design is shown in Fig.1. Please understand that Fig.1 only shows the circuit of one adjustable arbiter PUF. The XOR arbiter PUF can still employ multiple adjustable arbiter PUFs to ensure the security. Three parts are included.

Path Segments Controlled by Challenges

In the first part, the path segments are controlled by the challenge bits c_1~c_n. Same as traditional arbiter PUFs, a transition is propagated from t to the flip-flop, and the flip-flop compares the delays of two paths to produce the response bit g. The path segments are designed with the same nominal delay, so the process variations determine the g.

Path Segments for Adjusting Response Distributions

In the second part, the path segments are controlled by the adjustment bits $d_{0,1}$~$d_{0,h}$ and $d_{1,1}$~$d_{1,h}$. If $d_{0,h}$ is 0, the transition goes from $v_{0,h}$ to $w_{0,h}$, otherwise it goes from $u_{0,h}$ to $w_{0,h}$ through an extra buffer. In this way, two functions can be realized:

1) Traditional PUFs produce response bits 0s and 1s with similar probability because the compared two paths have the same nominal delay. With the path segments

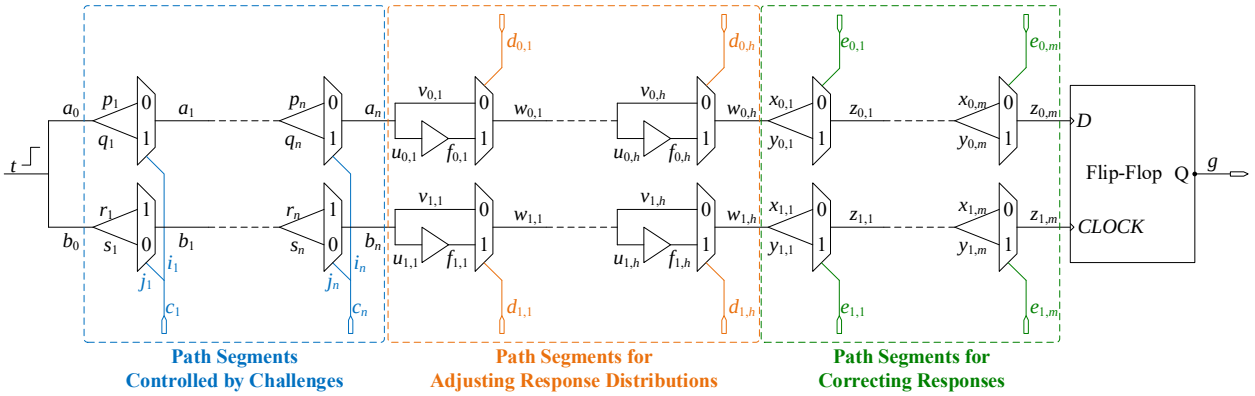

Figure 1: Adjustable Arbiter PUF

controlled by the adjustment bits, once the number of 1s among $d_{0,1}$~$d_{0,h}$ and the number of 1s among $d_{1,1}$~$d_{1,h}$ are different, the compared two paths will no more have the same nominal delay. Thus, the response bit will be bias to 0 or 1. By properly designing the buffers and controlling the adjustment bits, it is possible to make the PUF produce response bits 0s and 1s with any probabilities. In our implementation of the adjustable arbiter PUF in FPGA, a simulated annealing based heuristic algorithm is proposed to set the adjustment bits.

With this function, the application areas of PUFs can be expanded. For example, if sufficient buffers are added to the bottom path, then the PUF will rarely produce the response bit 1, and even for most such PUFs, no challenges can produce 1. If so, this PUF can be used as a trigger signal of a hardware Trojan. Such hardware Trojan will be much stealthier than traditional ones [13, 14]. This is because for traditional hardware Trojans, all the chips with the same hardware Trojan design have the same trigger condition, but for the PUF based hardware Trojan, only a few chips may contain the hardware Trojans which can really be triggered and their trigger conditions are different too. Considering that in 2016, million Samsung note 7 phones are recalled due to battery explosion of only a few number of phones, such hardware Trojans can also cause serious consequences.

On the other side, such bias PUF can also be used in a good way. For example, it can be used as a certification mechanism. Only if a user inputs right challenges to make certain number of bias PUFs output response bit 1s, the user is certified. This mechanism can be used to restrict unauthorized users to access critical data such as the scan test data, the details of which can be found in [15].

2) The delay difference of two paths can be measured. When adjustment bits are all 0s, assuming the delays of two paths are T_0 and T_1, then the response bit g depends on which one is larger. If T_0 is larger than T_1, then the g is 0. Next, we can set one adjustment bit of $d_{1,1}$~$d_{1,h}$ to 1, assuming the delay of the added extra buffer is T_b, then the g is still 0 if T_0 is larger than T_1+T_b. If the g becomes 1, then the delay difference of the two paths is less than T_b. Thus, with proper usage of adjustment bits, the delay difference can be well measured.

Since CRPs are unreliable when the delay difference of two paths is too small, with the capability of measuring the delay difference, at most testing a CRP three times with different adjustment bits is sufficient to identify whether it is reliable or not. In comparison with [11], where each CRP is tested for 10^5 times, the test time is significantly reduced. In comparison with [12], the proposed adjustable arbiter PUF has 2**h* adjustment bits to realize a more accurate measurement of the delay difference, so to reduce over-testing cases where a reliable CRP is identified as unreliable in [12].

Path Segments for Correcting Responses

In the third part, the path segments are controlled by the correction bits $e_{0,1}$~$e_{0,m}$ and $e_{1,1}$~$e_{1,m}$. They can replace the ECC bits. The correction bits of a challenge are generated by repeatedly using adjustment bits to evaluate the reliability of the response bit under the challenge and the tried correction bits. If a reliable response bit is identified, then the corresponding correction bits can correct the CRP. By using the path segments for correcting responses, far less hardware area is costed than using ECC circuits.

EXPERIMENTAL RESULTS

The proposed design is evaluated on two Xilinx Zynq7000 chips. For each chip, five adjustable arbiter PUFs are implemented in different places.

Firstly, the two basic metrics, uniformity and uniqueness, for evaluating a normal PUF are calculated. The uniformity of one PUF is the percentage of response bit 1s (or 0s) among all the CRPs. The idea uniformity is 50%. The uniqueness of two PUFs is the percentage of different (or same) response bits among all their CRPs. The idea uniqueness is also 50%. Since the path segments for adjusting response distributions and correcting

responses do not change the core of the PUF: process variations determine the CRPs, the features of traditional arbiter PUF are not destroyed, and the adjustable arbiter PUFs can still construct XOR arbiter PUF as [11] to ensure the security. In our experiments, we randomly choose 10^5 CRPs to evaluate the uniformity and the uniqueness. Their average values are 50.5% and 49.6%.

Secondly, we also randomly choose 10^5 CRPs for each PUF to evaluate the ability of adjusting response distributions. In Fig.2, the x-axis represents the target distribution, i.e. the percentage of response bit 1s among all the bits. The y-axis represents the distributions of 10 PUFs, using a simulated annealing based heuristic algorithm. It can be seen that the response distributions are successfully adjusted.

Thirdly, we evaluate the ability of selecting reliable CRPs and generating effective correction bits. For each CRP which is identified as a reliable one without or with correction bits, we test it under various work environments to check whether it is really reliable: temperature from -10ºC to 60ºC and supply voltage from 0.9v to 1.1v (normal value is 1v). As shown in Table I, the same as [11], all the CRPs which are identified reliable are really reliable, but we reduce the test time by about 3.3×10^4X. Meanwhile, without ECC circuits, we can still effectively correct CRPs. This at least save 90% hardware area. Finally, 57% more reliable CRPs are identified than [12].

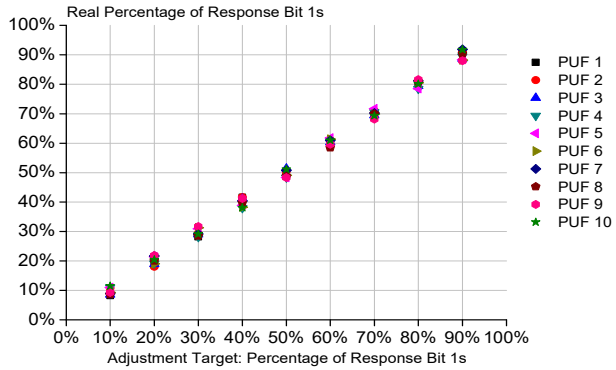

Figure 2: Response Distribution Adjustment Evaluation

TABLE I. RELIABILITY

	[11]	**This work**
Among CRPs which are identified reliable, the percentage of CRPs which are really reliable	100%	100%
Test times per CRP	10^5	3
Among CRPs with correction bits, the percentage of CRPs which are really reliable by using the correction bits		100%
Among CRPs which are identified unreliable by **[12]**, the percentage of CRPs which are identified reliable by us		57%

ACKNOWLEDGEMENTS

This paper is supported in part by National Natural Science Foundation of China (NSFC) under grant No. (61532017, 61704174, 61432017)

REFERENCES

[1] U. Ruhrmair, D. E. Holcomb, "PUFs at a Glance," DATE, 2014.

[2] J. Zhang, B. Qi, Z. Qin, G. Qu, "HCIC: Hardware-assisted Control-flow Integrity Checking", IoTJ, 2018.

[3] Q. Guo, J. Ye, B. Li, Y. Hu, X. Li, Y. Lan, G. Zhang, "PUFPass: A Password Management Mechanism Based on Software/Hardware Codesign," Integration, VLSI journal, 2018.

[4] Q. Guo, J. Ye, Y. Gong, Y. Hu, X. Li, "PUF based Pay-per-Device Scheme for IP Protection of CNN Model," ATS, 2018.

[5] J. Guajardo, S. S. Kumar, G.-J. Schrijen, P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," CHES, 2007.

[6] D. Lim, J. W. Lee; B. Gassend, G. E. Suh, M. van Dijk, S. Devadas, "Extracting Secret Keys from Integrated Circuits," TVLSI, 2005.

[7] J. Ye, Q. Guo, Y. Hu, H. Li, X. Li, "Modeling Attacks on Strong Physical Unclonable Functions," VTS, 2018.

[8] J. Ye, Y. Hu, X. Li, "Attack on Non-Linear Physical Unclonable Function," CCS, 2016.

[9] J. Ye, Y. Hu, X. Li, "RPUF: Physical Unclonable Function with Randomized Challenge to Resist Modeling Attack," AsianHOST, 2016.

[10] J. Ye, Y. Hu, X. Li, "OPUF: Obfuscation Logic Based Physical Unclonable Function," IOLTS, 2015.

[11] C. Zhou, K. K. Parhi, C. H. Kim, "Secure and Reliable XOR Arbiter PUF Design: An Experimental Study based on 1 Trillion Challenge Response Pair Measurements," DAC, 2017.

[12] J. Ye, Y. Hu, X. Li, "VPUF: Voter based Physical Unclonable Function with High Reliability and Modeling Attack Resistance," IOLTS, 2017.

[13] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, M. Tehranipoor. "Benchmarking of Hardware Trojans and Maliciously Affected Circuits," JHSS, 2017.

[14] J. Ye, Y. Hu, X. Li, "Hardware Trojan in FPGA CNN Accelerator," ATS, 2018.

[15] W. Li, J. Ye, X. Li, H. Li, Y. Hu, "Bias PUF based Secure Scan Chain Design," AsianHOST, 2018.