Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020)
IEEE Conference Record # 48766; IEEE Xplore ISBN: 978-1-7281-5371-1

FPGA based Hardware Security for Edge Devices in Internet of Things

Ms. Swati Kulkarni*, Dr. Vani R.M†, Dr. P.V. Hunagund‡
*IEEE Student Member
*‡Applied Electronics Department, †University Science Instrumentation Center
Gulbarga University, Gulbarga-585106 India
*swatikulkarni494@gmail.com, † vanirm12@rediffmail.com, ‡ prabhakar_hungund@yahoo.co.in

Abstract— **In Earlier days software security was restricted only for the military, defense, and banking applications. Due to rapid improvements in technology, in each sector, people are using smart electronics devices i.e. nothing but the Internet of Things (IoT). Day by day IOT applications are spreading widely into the market. In IoT applications, all smart objects and devices are connected through the internet. Internet is the most un-secure medium for data communication and it is very susceptible to attacks. In such cases, not only data security but hardware security is equally important to secure low power IoT devices. Hardware Security is quite an upcoming field in the electronics industry. It deals with security in Integrated Circuits (IC). Physically unclonable function (PUF) is an integral part of Hardware Security. It is working on the concept of process variation. In this paper, the most popular Ring Oscillator is implemented based PUF and Arbiter PUF types of PUF on the Zed board (ZYNQ Evaluation and Development Kit xc7z020clg484-1) FPGA using Xilinx Vivado 2016.1 Analysis of PUF parameter is not in the scope of this paper.**

Keywords- Hardware Security, PUF, RO-PUF, Arbiter PUF, Field Programmable Gate Array (FPGA)etc.

I. Introduction

PUF is a function based on certain physical characteristics of devices that occurred due to manufacturing or process variations. These process variations are occurred due to capacitance, threshold voltage or various other nano-scale factors [1]. These variations are so small that they cannot be measurable nor controllable. These internal process variations are unique to each IC so it is not possible to clone them [2].

There are several application areas where hardware security is an important aspect. The first one is in cryptography, to provide random and unique security key to encrypt and decrypt the data coming from any sensors to the IOT devices. There are different factors need to be considered while designing IOT applications. Security and Privacy are a major concern in IOT applications [3][9]. To achieve Data Security, Authentication and Confidentiality, cryptography techniques are used. Generally, in cryptographic applications, the secret key is stored in volatile or non-volatile memory. In normal cryptographic application one dedicated secrete key is stored in the volatile or non-volatile memory location and during encryption or decryption process key is retrieved from memory. It observed that attackers will attack the secrete key by doing a side-channel attack [4][17]. The entire system may get collapsed if the attacker got the key. Thus, storing the key in memory is not helpful. The output of the PUF system is a random bitstream. These random bits streams can be used as the secrete key instead of using the common dedicated key for cryptography. Every time PUF gives new random and unique key so no need to save the secrete key in memory [5].

Another important application of PUF is avoiding counterfeit and piracy of electronic devices/ ICs. According to a research institute for secure system Japan, duplicate electronic products have been increasing in all areas. These productions could damage the global economy of $1.7 trillion in 2015 [6]. Providing a unique identification ID to each product enables the supply chain to stop counterfeiting. PUF circuit extracts this unclonable fingerprint to generate a unique ID and can be used for device identification. PUF also can be used for securing Intellectual Property (IP). Even if PUF design has been cloned and the same PUF may be manufactured by counterfeit then the output of both designs will be different. However, each chip has a unique digital signature like a human fingerprint. Now in Xilinx Ultra-scale+ SOCs are also having inbuilt PUF for security purposes [7].

Fig1.1 Basic Block Diagram of CRP

The classification of PUF mainly depends on the number of Challenges. Challenges(C) are nothing but external input and responses(R) are output bit of PUF. C=2 n and R=2 m, where n-bit Challenges and m bit response bit. PUF is generally classified as strong and weak PUF. The number of challenges doesn't need to equal the number of responses. There is not a defined relationship between challenges and response but combinedly called them challenge-response pair (CRP) shown in fig 1.1. If the maximum number of CRPs then it is called as strong PUF whereas the minimum number of CRPs is called a weak PUF and don't want any predictable linking between C&R as they are independent. Two similar Challenges applied to two similar devices; Response should be different. A puf is embedded in ic chips that should be protected from cloning that's why the name is a physically unclonable function [9].

Organization of the paper follows; introduction, related work, proposed work, result analysis, PUF design issues, conclusion and references.

II. RELATED WORK

A. Ring Oscillator

Fig 2.1 Simple Ring Oscillator

Fig 2.1 shows the basic construction of a simple ring oscillator. One NAND gate and 4 inverters are connected serially. One of the inputs of the NAND gate is an enable input and another input is coming from second last NOT gate i.e. feedback input.

In the RO-PUF series of N, ring oscillators are connected to N bit multiplexers. Multiplexer will select randomly any two RO from the series. The output of Multiplexers is connected as a clock input to the two counters respectively. Finally, both the counters output will be compared. Counter overflows will be determined by the comparator. The output frequency of each RO in the series is different. The frequency of RO is depending upon process variations. Process variation contributes delays in ring oscillator design so the resultant output will be different [13].

$$freq\ of\ ring\ osr\ (f) = \frac{1}{2*n*t} \quad (1)$$

Where n: -no of stages of RO and

t: - propagation delay of each gate

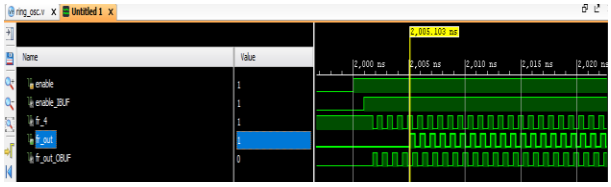

Fig 2.2 Simulation waveform of Ring Oscillator

. fig 2.2 shows the output waveform of ring oscillator It generates a square wave [12][27].

B. Arbiter

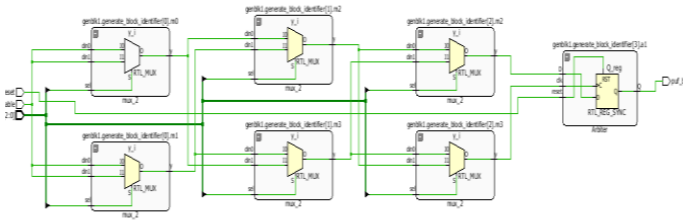

Fig 2.3 Typical Arbiter PUF RTL Schematic

Arbiter PUF is fundamentally based on switches. Nitin Pundir et.al describes arbiter PUF constructed using multiplexers and D-latch. Two Multiplexers are acting as switching elements. The same input is applied to both multiplexers and created two parallel paths.

Fig 2.3 shows typical Arbiter PUF connection arrangement Both paths are having different delays so it will take different time to reach the output. Last multiplexers of upper path's output are connected to D-input whereas lower path multiplexer output is fed as the clock input to D-FF and know that when only a clock edge is positive and D-input =1 then only D-out =1 otherwise D-out =0 [10][11].

III. PROPOSED WORK

A. Implementation of RO-PUF

RO-PUF is implemented. Fig 2.1 consists of 4 inverters and NAND gate. Consider fig 3.1 in this series of sixteen ring oscillator (RO) are connected parallelly. All the RO outputs connected to the two 16:1 multiplexer. In this case, the select line of multiplexers is nothing but challenges. Challenges will pick any two ROs from series. To pick challenges used 8-bit counter. First 4 bits i.e. MSB bits are connected to select line of the first multiplexer and lower 4 bits i.e. LSB is connected to select lower mux. This configuration provides 2^8= 256 different combinations of Challenges.

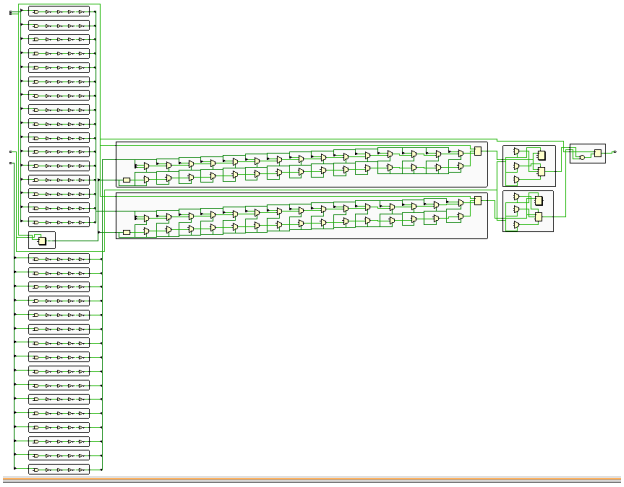

Fig 3.1 RTL schematic of RO-PUF

The outputs of multiplexers are fed to two counters. Counters are initialized to reset and stats incrementing on every positive edge of the clock. Multiplexer's output is clock input to the counter. RO selected by multiplexers produces different frequency so clock input of both counters will be different. Suppose counter one will reach its final value faster than counter two. The output of both the counters compared by the comparator. The comparator will decide which counter overflows first. The output of the comparator is nothing but PUF output i.e. response. the response is of one bit. Multiple instances of RO-PUF will be required to generate multiple response bits [26]. In this case, 8 instances of design is implemented.

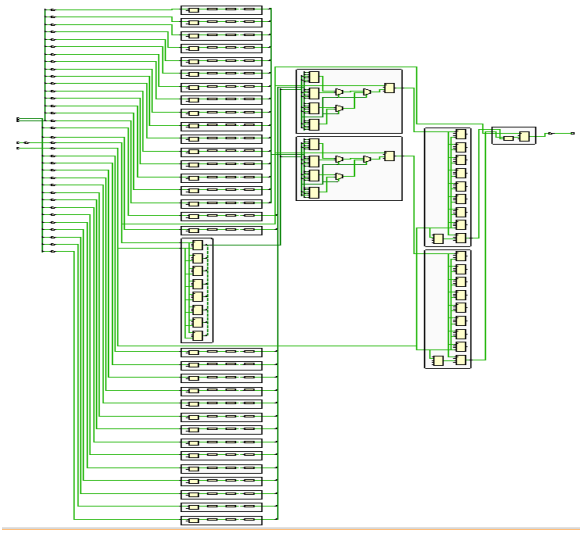

Fig 3.2 Synthesis schematic of RO-PUF

Fig 3.2 shows the synthesized schematic of RO-PUF for a single instance. The logic of RO-PUF is now converted into a gate-level netlist. All the logic gates are represented in terms of FPGA resources such as LUTs, Wide multiplexers, and SRL, etc. After synthesis, Implementation is performed.

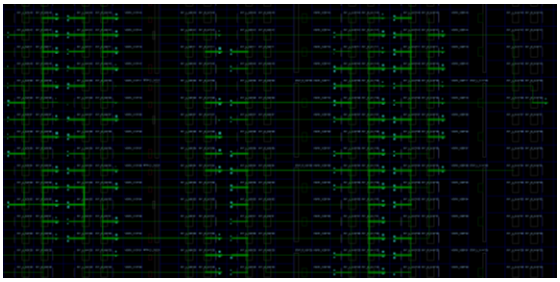

Fig 3.3 Implemented Device view of RO_PUF

Fig 3.3 shows how Placement and Routing of component is done for 8 instances.

B. Implementation of Arbiter PUF

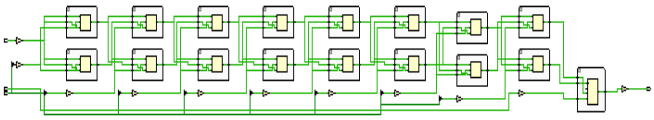

Fig 3.4 Synthesized schematic of Arbiter PUF

Fig 3.4 shows synthesized schematic of Arbiter PUF. 2:1 multiplexer would be primitive component of Arbiter.

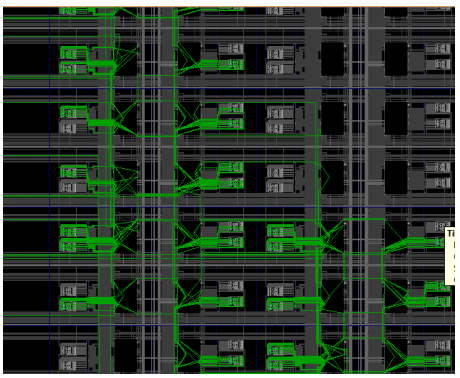

Fig 3.5 Implementation device view of Arbiter PUF

Placing and routing of Arbiter PUF is also done shown in fig 3.5. Like RO-PUF here also get 1 bit as response bit hence design is instantiated 8 times to get 8-bit response.

Result Analysis

A. *RO-PUF*

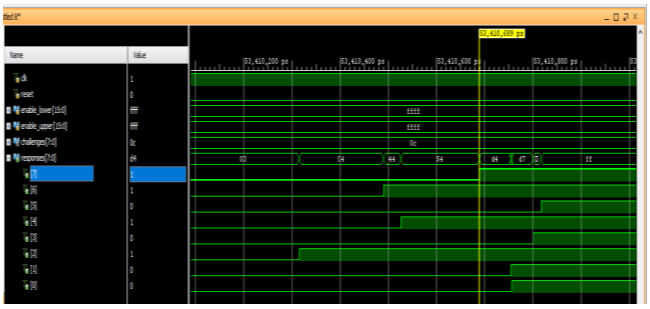

Fig 4.1 Post Implementation timing simulation of RO-PUF

Fig 4.1 shows the post-implementation timing simulation of RO-PUF. Actually, in Xilinx Vivado can perform 5 types of simulations apart from that it performed post-implementation timing simulation. PUF is a function of delay and in post-implementation timing simulation, the implementation engine of vivado will add all logic and net delays of components. The output of RO-PUF is nothing but the random bitstream.

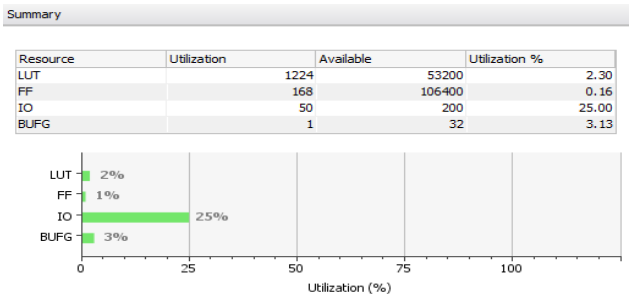

Fig 4.2 Resouce Utilization of RO-PUF

The design for 8-bit PUF response is implementd. Fig 4.2 shows Recourse Utilization summary for RO-PUF for 8 bits.

B. *Arbiter PUF*

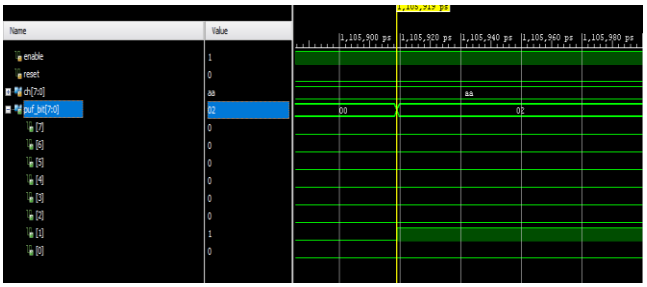

Fig 4.3 Post Implementation timing simulation of Arbiter-PUF

Fig 4.3 shows post implementation timing simulation of Arbiter PUF The output of Arbiter PUF is nothing but random bit stream.

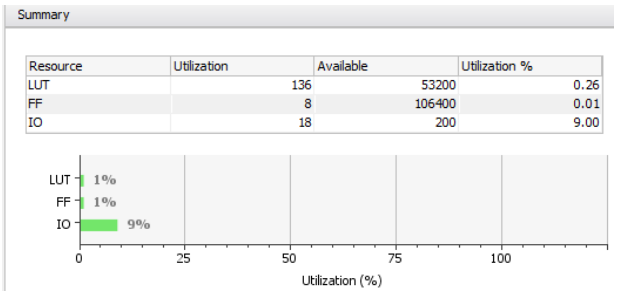

Fig 4.4 Utilization Report Arbiter PUF

Fig 4.4 shows utilization report of arbiter PUF. Here it is showing number of FPGA resources has been acquired by Arbiter PUF.

There are three more critical aspects of PUF Uniqueness, Reliability and unpredictability.

1. **Uniqueness:** How PUF will generate Unique response bit for different challenges i.e. nothing but unique property of PUF. When designed PUF then it is expected that it should provide a unique output bit every time. It is calculated using inter-chip variations. To verify this property better to implement using hard macros. The maximum inter-chip variation is the quality of good PUF [19][20].

2. **Reliability**: Reliability of PUF design is how many times the device will give the same output / Response for applying the same challenges. To verify this property, apply the same challenge multiple times and calculate intra-chip variation. Irrespective of temperature, voltage variations or aging.. The intra-chip variation shows the robustness of PUF. Irrespective of temperature or voltage variation for the same device and same challenge output/ response bit should be as close as possible and i.e. sign of good PUF [21][22].

3. **Unpredictability:** This is the most important property of PUF. For two close challenges output bit should be far different. An attacker should not be able to predict the response bit. To observe the above property Inter-chip and intra-chip variation analysis is done. Inter-chip means in the same device/FPGA PUF is implemented and Intra-chip means on different FPGA board PUF is implemented. In our case, 4 different ZedBoard is used on the same temperature and voltage condition [23][24].

IV. PUF DESIGN ISSUES

All the designs are implemented using Xilinx Vivado 2016.1 tool. Xilinx tool is designed in such a way that its synthesis engine will try to optimize your code [14]. While designing RO-PUF and Arbiter PUF, many components will

get optimized by tool and resultant will be 'Empty Netlist'. To avoid such a problem Xilinx suggested some synthesis technics in the Xilinx synthesis user guide [28]. Following technics were used in our design.

1. Disabled optimization property by making changes in the Synthesis setting -flatten_hierarchy made none, it will avoid optimization of the netlist.

2. Disabled Keep_equivalent _resistor, to avoid unnecessary delays in design.

3. Use attribute (* dont_touch = "yes" *) for every instance in RTL code to avoid optimization of instances[28].

4. While generating bitstream for RO-PUF, needed to specify combinatorial path constraints otherwise bitstream not generate and give DRC error.

5. It is suggested by many authors while implementing RO-PUF on FPGA that uses hard macro technique [12].

CONCLUSION

The purpose of this paper is to develop ideas about security and privacy for IoT applications. PUF provides hardware security with minimum hardware requirements. Designed RO-PUF and Arbiter PUF on Xilinx Vivado is desinged. the verified random and unique key gets generated for every different challenge.RO-PUF require more resources than Arbiter PUF for generating the same length of the key. FPGA gives optimized hardware. PUF builds trust in hardware. Designing energy-efficient PUF is a more interesting and important topic for future research.

REFERENCES

[1] V. P. Yanambaka, S. P. Mohanty and E. Kougianos, "Making Use of Manufacturing Process Variations: A Dopingless Transistor Based-PUF for Hardware-Assisted Security," in IEEE Transactions on Semiconductor Manufacturing, vol. 31, no. 2, pp. 285-294, May 2018

[2] G. Edward Suh, Srinivas Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in 44th ACM/IEEE Design Automation Conference,, San Diego, CA,, 2007.

[3] Kulkarni, S., Vani, R.M., Hunagund "Review on IoT based case study: applications and challenges," in International Conference on Intelligent Data Communication Technologies and Internet of Things,Springer, Cham, Coimbtoor,India, 2018.

[4] John Ross Wallrabenstein, "Practical and Secure IoT Device Authentication using Physical Unclonable Functions," in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), Vienna, 2016, pp. 99-106., 2016.

[5] Kulkarni, S., Vani, R.M., Hunagund, "A Study on Physical Unclonable Functions Based Security for Internet of Things Applications," Hemanth D., Shakya S., Baig Z. (eds) Intelligent Data Communication Technologies and Internet of Things. ICICI 2019., Vols. Lecture Notes on Data Engineering and Communications Technologies, vol 38. Springer, Cham, pp. 607-614, 2019.

[6] Shital Joshi ,Saraju P. Mohanty, Elias Kougianos, "Everything You Wanted to Know About PUFs"; in: IEEE Potentials Vol. 36 , Issue: 6 , Nov.-Dec. 2017 pp. 38-46

[7] Xilinx Zynq Ultra-Scale+ Device Technical Reference Manual UG1085 (v2.1) August 21, 2019

[8] Helena Handschuh, Geert-Jan Schrijen, and Pim Tuyls, "Hardware Intrinsic Security from Physically Unclonable Functions," in A.-R. Sadeghi, D. Naccache (eds.), Towards Hardware-Intrinsic Security, Information Security and Cryptography, DOI 10.1007/978-3-642-14452-3_2,, Verlag Berlin Heidelberg 2011, 2011.

[9] Cédric Marchand, Lilian Bossuet, Ugo Mureddu, Nathalie Bochard, Abdelkarim Cherkaoui, Viktor Fischer, "Implementation and characterization of a physical unclonable function for IoT: a case study with the TERO-PUF.," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE, 2018.

[10] Nitin Pundir, Fathi Amsaad, Muhtadi Choudhury, and Mohammed Niamat, "Novel Technique to Improve Strength of Weak Arbiter PUF," in IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS),, Boston, MA, 2017, pp. 1532-1535., 2017.

[11] Shahin Tajik, Enrico Dietz, Sven Frohmann, Jean-Pierre Seifert, Dmitry Nedospasov, Clemens Helfmeier, Christian Boit, Helmar Dittrich, "Physical Characterization of Arbiter PUFs," in Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[12] Charles Herder, Meng-Day (Mandel) Yu, Farinaz Koushanfar, and Srinivas Devadas, "Physical Unclonable Functions and Applications: A Tutorial," Proceedings of the IEEE, vol. Vol. 102, pp. 1126-1141, August 2014.

[13] Xin Xin, Jens-Peter Kaps, Kris Gaj, "A Configurable Ring-Oscillator-Based PUF for Xilinx FPGAs," in 4th Euromicro Conference on Digital System Design, Oulu, 2011.

[14] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto and Kazuo Sakiyama, "A New Mode of Operation for Arbiter PUF to Improve Uniqueness on FPGA," in Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 871–878, 2014.

[15] Mahin Anil Kumar and Ramesh Bhakthavatchalu, "FPGA based delay PUF Implementation for Security Applications," in IEEE International Conference on Technological Advancements in Power and Energy(TAP Energy) , Kollam, India, 2017.

[16] Muhammad Naveen Aman, Kee Chaing and Biplab Sikdar, "Physically Secure Mutual Authentication for IoT", IEEE 2017, pp. 310-317.

[17] Anuj P. Johnson, Rajat Subhra Chakroborty and Debdeep Mukhopadhyay, "A PUF- Enabled Secure Architecture for FPGA-Based IoT Applications", IEEE Transaction on multi- scale computing system, vol 1, No2, April-June 2015.

[18] Muhammad Arif Mughal, Xiong Luo, Zahid Mahmood and Ata Ullah, "Physical Unclonable Function Based Authentication Scheme for Smart Devices In Internet Of Things", 2018 IEEE international conference on smart Internet of things, IEEE 2018, pp. 160-165.

[19] Ulrich Rührmair, Ulf Schlichtmann, Wayne Burleson, "Special Session: How Secure Are PUFs Really? On The Reach and Limits Of Recent PUF Attacks", in Design, Automation & Test in Europe Conference & Exhibition (DATE), Year: 2014, pp. 1 – 4.

[20] N. Nalla Anandakumar ; Mohammad S. Hashmi ; Somitra Kumar Sanadhya, "Compact Implementations of FPGA-based PUFs with Enhanced Performance", in 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID) Year: 2017, pp. 161 – 166.

[21] G.Pocklassery ; Venkata K Kajuruli ; J. Plusquellic ; F. Saqib, "Physical Unclonable Functions And Dynamic Partial Reconfiguration For Security In Resource-Constrained Embedded Systems", in IEEE International Symposium on Hardware Oriented Security and Trust (HOST) Year: 2017,pp. 116 – 121

[22] Charles Herder ; Meng-Day Yu ; Farinaz Koushanfar ; Srinivas Devadas, "Physical Unclonable Functions and Applications: A Tutorial "; Proceedings of the IEEE Vol. 102 , Issue: 8 , Aug. 2014, pp. 1126 – 1141.

[23] D. P. Sahoo, R. S. Chakraborty and D. Mukhopadhyay, "Towards Ideal Arbiter PUF Design on Xilinx FPGA: A Practitioner's Perspective," 2015 Euromicro Conference on Digital System Design, Funchal, 2015, pp. 559-562.

[24] E. Dubrova, "A Reconfigurable Arbiter PUF with 4 x 4 Switch Blocks," 2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL), Linz, 2018, pp. 31-37.

[25] N. Sivasankari and A. Muthu kumar," Implementation Of A Hybrid Ring Oscillator Physical Unclonable Function" intact journal on microelectronics, JULY 2018, VOLUME: 04, ISSUE: 02,pp 602-607

[26] M. Patterson, J. Zambreno, C. Sabotta, S. Vyas, A. Mills, "Ring oscillator PUF design and results", 2011.

[27] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. In Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, pages 67 –70, June 2008.

[28] Mohammed Saeed Alkatheiri, Yu Zhuang, Mikhail Korobkov, and Abdur Rashid Sangi, "An Experimental Study of the State-of-the-Art PUFs Implemented on FPGAs," in IEEE Conference on Dependable and Secure Computing, Taipei, Taiwan, 2017.

[29] Xilinx Vivado Design Suite User Guide Synthesis UG901 (v2019.2) January 27, 2020