

Secure lightweight obfuscated delay-based physical unclonable function design on FPGA

Mohammad Haziq Ishak⁵, Mohd Syafiq Mispan^{1,3,4}, Wong Yan Chiew^{1,3,5}, Muhammad Raihaan Kamaruddin^{2,3,5}, Mikhail Aleksandrovich Korobkov⁶

¹Micro & Nano Electronics, Fakulti Kejuruteraan Elektronik & Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

²Machine Learning and Signal Processing, Fakulti Kejuruteraan Elektronik & Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

³Centre for Telecommunication Research & Innovation, Fakulti Kejuruteraan Elektronik & Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

⁴Fakulti Teknologi Kejuruteraan Elektrik & Elektronik, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

⁵Fakulti Kejuruteraan Elektronik & Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

⁶Siemens Digital Industries Software (EDA), Moscow, Russia

Article Info

Article history:

Received Oct 20, 2021

Revised Jan 13, 2022

Accepted Feb 3, 2022

Keywords:

Field-programmable gate array

Internet of things

Physical unclonable function

ABSTRACT

The internet of things (IoT) describes the network of physical objects equipped with sensors and other technologies to exchange data with other devices over the Internet. Due to its inherent flexibility, field-programmable gate array (FPGA) has become a viable platform for IoT development. However, various security threats such as FPGA bitstream cloning and intellectual property (IP) piracy have become a major concern for this device. Physical unclonable function (PUF) is a promising hardware fingerprinting technology to solve the above problems. Several PUFs have been proposed, including the implementation of reconfigurable-XOR PUF (R-XOR PUF) and multi-PUF (MPUF) on the FPGA. However, these proposed PUFs have drawbacks, such as high delay imbalances caused by routing constraints. Therefore, in this study, we explore relative placement method to implement the symmetric routing in the obfuscated delay-based PUF on the FPGA board. The delay analysis result proves that our method to implement the symmetric routing was successful. Therefore, our work has achieved good PUF quality with uniqueness of 48.75%, reliability of 99.99%, and uniformity of 52.5%. Moreover, by using the obfuscation method, which is an Arbiter-PUF combined with a random challenge permutation technique, we reduced the vulnerability of Arbiter-PUF against machine learning attacks to 44.50%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mohd Syafiq Mispan

Fakulti Teknologi Kejuruteraan Elektrik & Elektronik, Universiti Teknikal Malaysia Melaka

Street Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

Email: syafiq.mispan@utem.edu.my

1. INTRODUCTION

The internet of things (IoT) is an ecosystem of networked physical objects accessible via the Internet. It is an embedded technology that allows devices to communicate with each other [1]. It also allows a device to monitor and understand a scenario or environment without human assistance. With the increasing adoption of wireless fidelity (Wi-Fi) and four-generation (4G) long term evolution (LTE) wireless internet access, the evolution towards ubiquitous information and communication networks is already apparent [2]. The IoT can

also be described as a data exchange environment where devices are connected to wired and cellular networks [3]. IoT applications exist in various fields such as smart cities, healthcare, smart homes, and industrial security. For example, with an IoT-based remote monitoring system, it becomes easier to monitor the overall performance of the system through a web-based approach rather than using an on-site monitoring meter [4]. According to Bao *et al.* [5], the field-programmable gate array (FPGA) has proven to be a viable platform for IoT development due to its inherent flexibility and reconfigurability.

FPGA is an integrated circuit that combines internal hardware blocks with user-programmable interconnects to customize operation to a particular application. The interconnects can be easily reprogrammed, allowing an FPGA to adapt to design changes. FPGAs evolved from earlier devices such as programmable read-only memories (PROMs) and programmable logic devices (PLDs). While these devices could be programmed at the factory or in the field, they were based on fuse technology and could not be modified once programmed. In contrast, FPGAs store their configuration data in a reprogrammable medium such as static random access memory (SRAM) or flash memory. A typical FPGA design consists of thousands of basic elements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects known as a fabric that routes signals between the CLBs. The FPGA and external devices are interconnected via input/output (I/O) blocks. Like any other technology, FPGAs must be protected against several security threats, such as cloning of the FPGA bitstream or piracy of the core intellectual property (IP).

Physically unclonable function (PUF) was introduced to address these shortcomings by exploiting the inherent physical properties of a device. PUF is a function based on a physical system that is easy to evaluate. It is not clonable or reproducible on another copy of the same physical system, even if the functionality is known. These advantages of PUFs are due to the process variations in each chip caused by the manufacturing process. Several implementations of PUFs design for security applications in FPGAs include intellectual property protection, secure key generation, and cloud security [6]–[9]. However, the previous PUFs designed on FPGAs have drawbacks, such as delay imbalances caused by routing constraints. Therefore, in this study, we propose relative placement method to implement the symmetrical routing in the obfuscated delay-based PUF on the diligent Nexys-4 Artix-7 FPGA board. The obfuscated delay-based PUF is an Arbiter-PUF combined with a random challenge permutation technique to reduce the vulnerability of the Arbiter-PUF against machine learning attacks (ML-attack). We prove that using relative placement method to implement the delay-based PUF in FPGA can eliminate the biased response and achieved great PUF quality which include uniqueness, reliability, and uniformity. Moreover, we show that the random challenge permutation method can be implemented using routing obfuscation, hence achieve low area overhead.

This work begins with a detailed introduction to related work on PUFs in section 2. Subsequently, the PUF design and implementation are described in section 3. Followed by the method used to construct the architecture of the PUF in section 4. PUF performance analyses are presented in section 5. Conclusion are drawn in section 6.

2. RELATED WORK

Majzoobi *et al.* [10] proposed an Arbiter-PUF (APUF) structure for the FPGA platform based on programmable delay logic (PDL). The internal structure of the look up tables (LUTs) is used to construct a PDL. Two PDL are combined to form a path swapping switch. However, this technique introduces a bias at the beginning and end of the delay paths. Sahoo *et al.* [11] has proposed an improvement for the PDL technique. The authors use a PDL chain to construct the upper and lower delay paths independently by using the hard macro function of the Xilinx computer aided design (CAD) tool, and then instantiated the hard macro twice for the asymmetric implementation of the delay paths. In contrast, creating a hard macro of a long PDL chain is a time-consuming process that limits the flexibility of the design.

In another study, Habib *et al.* [12] implement a new ring oscillator (RO) design on FPGA. In this design, the internal variations of the FPGA LUTs are exploited to generate a PUF response. The proposed PUF design on FPGA achieves great uniqueness, reliability and uniformity but the PUF suffers from a continuous dynamic power dissipation due to the oscillation. Elsewhere, Dan *et al.* [13] implemented a reconfigurable XOR PUF (R-XOR PUF) on the FPGA. The R-XOR PUF consists of multiplexers and inverters. The response of R-XOR PUFs is generated by XORing the two responses. Nevertheless, this method provides a low evaluation metric in terms of uniqueness and reliability. In our study, we explore an efficient method to implement the symmetrical routing in obfuscated delay-based PUF on FPGA without compromising the PUF performance

such as uniqueness, reliability, and uniformity. Moreover, we show that the random challenge permutation in obfuscated delay-based PUF can be implemented simply by routing obfuscation which introduces low area overhead.

3. DESIGN OF PHYSICAL UNCLONABLE FUNCTION

The obfuscated delay-based PUF is an Arbiter-PUF combined with a random challenge permutation technique. Arbiter-PUF was first introduced by Lee *et al.* [14], which is based on race conditions in integrated circuits (ICs). Although it is considered as one of the strong PUFs, an adversary can easily predict the response of Arbiter-PUF by using a linear additive model using machine learning techniques. To overcome this problem, random challenge permutation technique was introduced by [15] to reduce the vulnerability of Arbiter-PUF against ML-attack.

A critical factor to be emphasized when implementing this PUF design in FPGA is symmetric routing. Based on Figure 1, the CLB element in Xilinx Artix-7 contains a pair of slices, and each slice consists of four LUT with 6 inputs. Relative placement is used to ensure balanced routing for the PUF and to prevent delay imbalance. In relative placement, each stage of the circuit component has been implemented in a slice with two LUTs. The LUTs for the upper path are located at the position of the upper LUTs in the slice, and the LUTs for the lower path are located at the position of the lower LUTs in the slice. The routing of the path starts from slices X0Y1 to X31Y1, where "X" followed by a number indicates the position of each slice in a pair as well as the column position of the slice, while a "Y" followed by a number indicates a row of slices. Tool command language (TCL) script is used to automate the process of relative placement method.

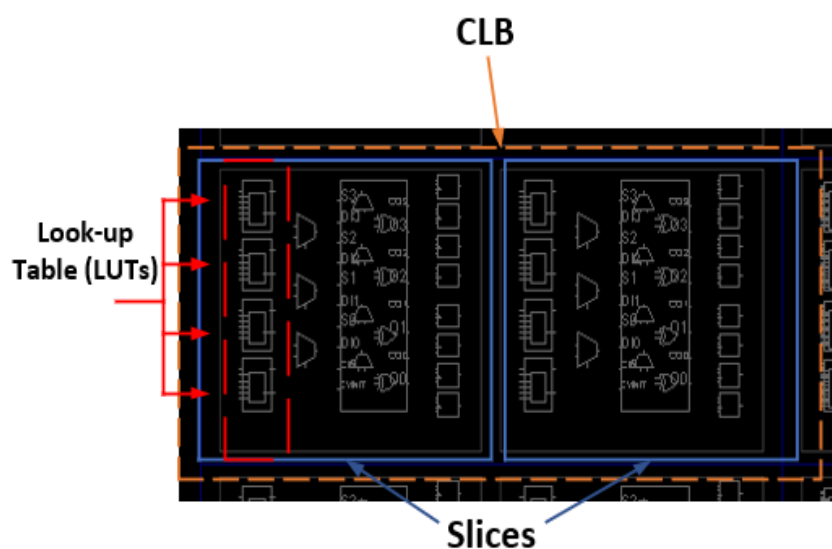


Figure 1. Xilinx Artix-7 fabric

Based on Figure 2, the obfuscated delay-based PUF was designed with 32 stages of switching components and a D flip-flop as the arbiter on Artix-7. A microblaze processor block diagram was designed to write and read the challenge-response pairs (CRPs). The architecture of the processor consists of general purpose input output (GPIO) 0 and 1. These GPIOs were used to send pulses, challenge bits and fetch the response from the PUF module. The Artix-7 board has a built-in 100 MHz oscillator that is used as a clock input to the FPGA and to communicate with the PC via a 115,200 bps universal asynchronous receiver transmitter (UART).

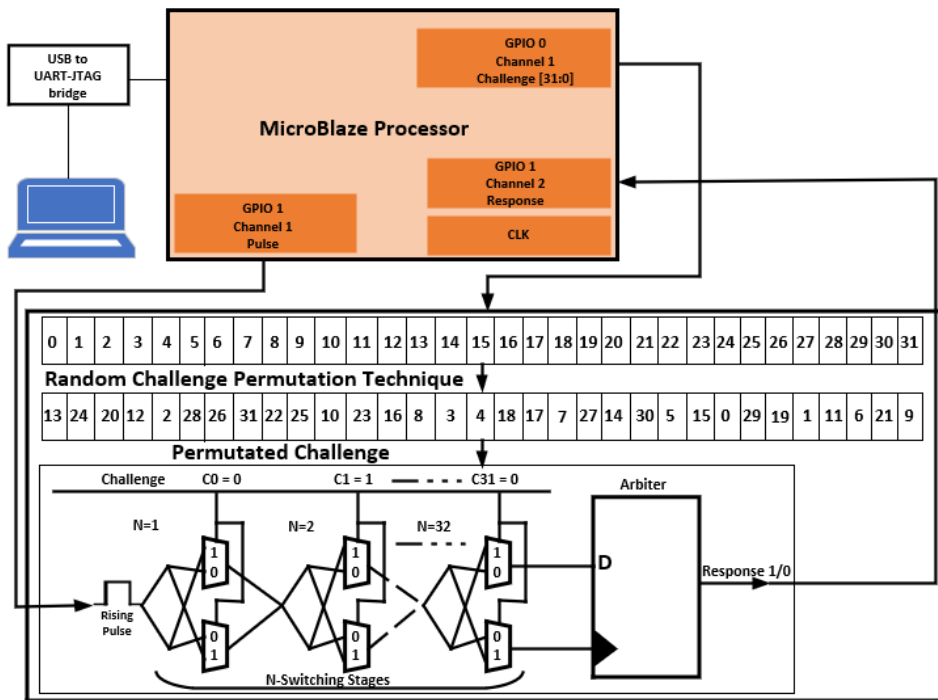


Figure 2. System functional block diagram

4. METHODOLOGY

Xilinx vivado 2018.3 is used to synthesize and implement the obfuscated delay-based PUF in digilent Nexys-4 Artix-7 FPGA board. For PUF performance evaluation, sufficient number of PUFs must be designed to obtain statistically significant results. For cost reasons and following the method described in [16], five obfuscated delay-based PUFs were implemented in an FPGA, each on a different set of slices. We assume that each PUF design represents an FPGA device. The routing of the path starts from slices X0Y150 to X31Y150 for PUF 1, X0Y151 to X31Y151 for PUF 2, X0Y152 to X31Y152 for PUF 3, X0Y153 to X31Y153 for PUF 4, and X0Y154 to X31Y154 for PUF 5. A 32-bit response is generated for each of the PUF instances. The placement constraints in vivado were achieved using TCL scripts. A microblaze soft processor core with PUF architectural design was designed and implemented for writing and reading the CRPs. Also, an integrated logic analyzer (ILA) was connected to MicroBlaze to display the discrete waveform of the pulse, the challenge bit, and the collected response in the hardware manager of Xilinx vivado. The MicroBlaze program was written in the C language in the Xilinx software development kit (SDK) for simulation purposes. The Tera term software is connected to the MicroBlaze processor through a serial data interface to collect CRPs. Subsequently, the collected CRPs are analyzed using MATLAB to evaluate the uniqueness, reliability and uniformity. For the ML-attack evaluation, 32000 CRPs are applied to the PUF, where each challenge generates one bit of response. Following the method described in [15], the artificial neural network (ANN) technique is used to evaluate the resiliency of the PUF against ML-attack.

5. SIMULATION RESULTS AND ANALYSIS

5.1. Physical unclonable function implementation

Figure 3 shows the implementation design for the switching component. As can be observed inside the switching component, there are two multiplexers implemented by LUTs. Each LUTs has three inputs, namely I0, I1, and I2. The input of the challenge bit (I1) controls the path inside the LUTs. When the challenge bit is '0', the pulse signal is routed through (I0). When the challenge bit is '1', the pulses cross and pass through (I2). Each switching component has its own challenge bit. In our work, there are 32 switching components, and their challenge bits are permuted randomly. The last switching component is connected to D flip-flop,

which is the arbiter block. The upper path of the last switching component is connected to the data input, and the lower path is connected to the clock of D flip-flop.

Figure 4 illustrates the MicroBlaze processor block diagram connection with its slave module. Each slave has its own address to communicate with the processor through the peripheral module. As can be observed, the Arbiter-PUF module has two inputs and one output, namely ipulse, ichallenge, and oresponse. The 32-bit challenge was generated by GPIO 0 and connected to the ichallenge of the PUF. The pulse from GPIO 1 (channel 1) is connected to ipulse of the PUF. Any response generated by the PUF is fetched from the oresponse output of the PUF through GPIO 1 (channel 2). This entire process was controlled by the UART module set up in the MicroBlaze processor to collect the CRPs.

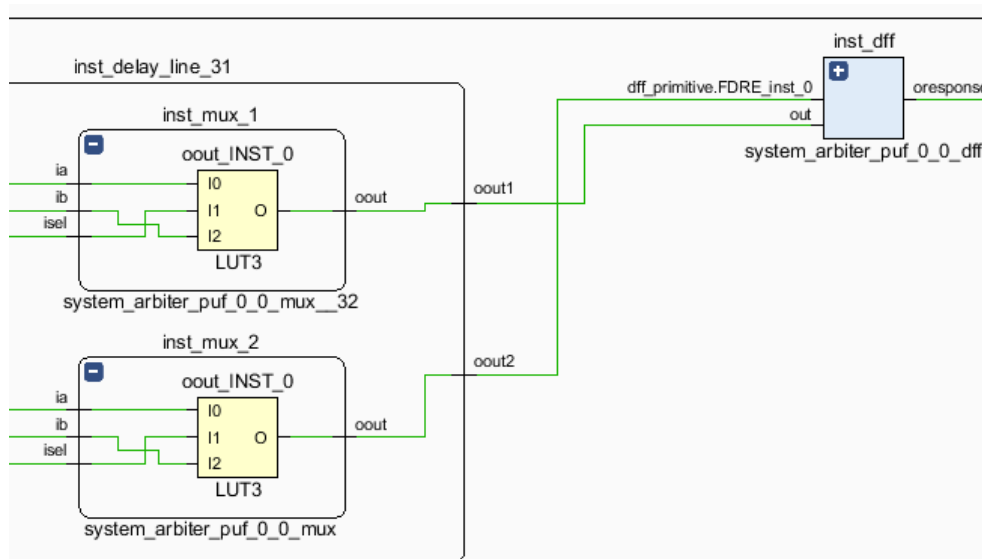


Figure 3. Switching component implementation

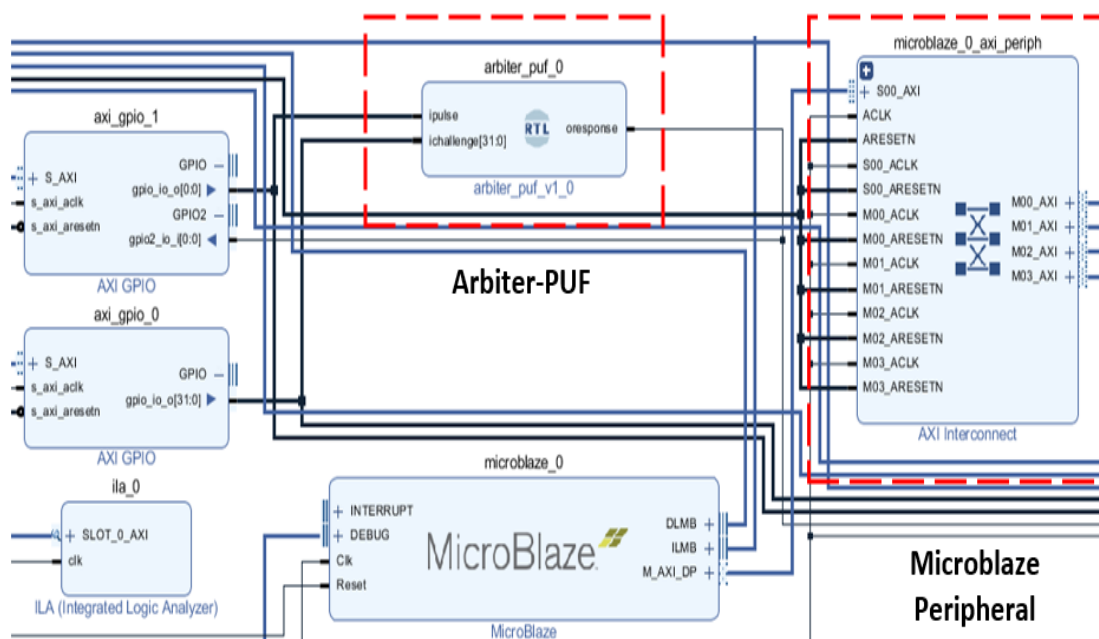


Figure 4. MicroBlaze processor block diagram

Figure 5 shows the complete balance routing map for a switching component. It can be seen that the routing of the upper and lower paths is symmetrically designed, which guarantees an un-biased PUF response and proves that our relative placement method for the PUF design is successfully created. Table 1 shows the total accumulated delay difference between the upper and lower paths for 5 different PUF placements in the FPGA. According to Pundir *et al.* [17], the A-PUF implementation on the FPGA is challenging because the two paths must be symmetric and similar, so the difference in the generated response is due to variations in the fabrication process rather than architectural delays in the design. It can be observed from Table 1, that for each PUF placement, the delay difference between the upper and lower paths is small and random, which proves that the implementation of the manual routing technique used in this study has been successfully created in the FPGA.

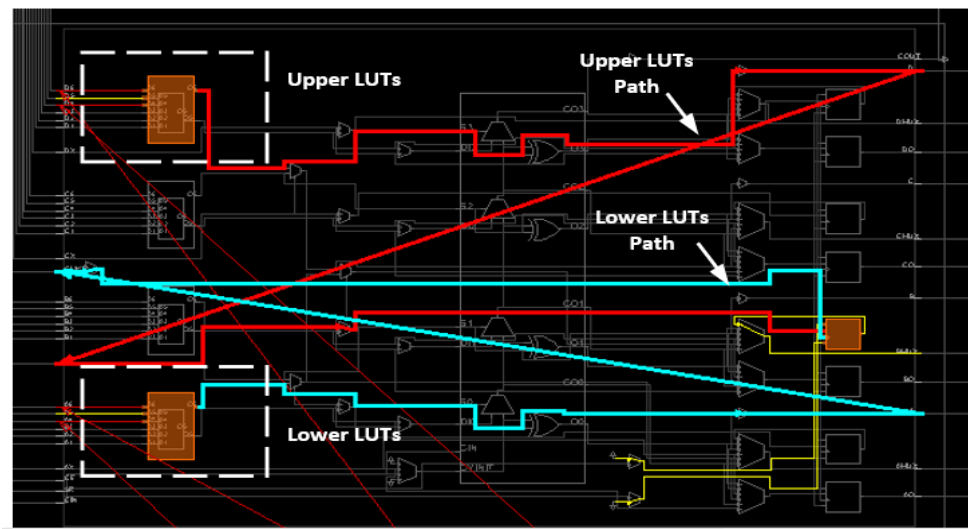


Figure 5. Upper and lower routing path

Table 1. Total accumulated delay for upper and lower paths

PUF	Upper (ps)	Lower (ps)
PUF 1	18601	18146
PUF 2	17906	17927
PUF 3	18162	18144
PUF 4	17719	17667
PUF 5	18124	18146

5.2. Performance metrics

A 32-bit response is generated for each PUF instances using the method described in section 4. The response is used to quantify the PUF performance using standard evaluation metrics. As described in [18], three metrics to evaluate the PUF performance are uniqueness, reliability, and uniformity. Uniqueness is the ability of a PUF to be uniquely distinguished from a group of PUFs of a similar type. Meanwhile, reliability defines the ability of the PUF to produce the same responses when applied with the same challenges over the temperature and supply voltage fluctuations. Uniformity defines the proportion of 0's and 1's in a PUF response. The number of 0's and 1's in a PUF response must be balanced, hence an ideal uniformity is distributed at 50%.

Table 2 compares the performance of the obfuscated delay-based PUF design with the previously proposed PUFs. As can be seen in Table 2, PCS-PUF has the highest uniqueness with 49.80% and the M-PUF technique has the lowest with 40.60%. Nevertheless, our approach also provides higher uniqueness which is 48.75%. For reliability, 10 sets of a hundred 32-bit responses of the obfuscated delay-based PUF are collected under a nominal condition of 1.1 V supply voltage and 24 °C. The first set of responses is used as a reference in which the other nine sets are compared. The results show that our design achieves close to an ideal value of 100% compared to the other PUFs. The response generated by the PUF is stable even though the CRPs

collection process was repeated 9 times. In terms of uniformity, RO-PUF has the highest uniformity with 50.13%, followed by PCS-PUF with 49.77%. The uniformity of our approach is 52.5%, which is slightly higher than the ideal value. Our work achieves efficient resource utilisation with 64 LUTs and 32 slices in hardware resource consumption. From Table 2, we can conclude that our approach has achieved good result in PUF quality metrics and the area overhead in the FPGA.

Table 2. A comparison of hardware resource consumption and metrics of different PUF designs

PUF design	Uniqueness (%)	Reliability (%)	Uniformity (%)	FPGA	Area overhead	Predictability (%)
PCS [19]	49.80	98.19	49.77	Spartan 3E	388 LUTs, 196 slices	-
RO-PUF [20]	-	-	-	Spartan-3E	-	85.20
PDL [11]	45.25	97.12	50.34	Spartan-3E	-	-
MISR [21]	49.30	99.80	-	Xilinx ZC706	380 LUTs, 128 FF	-
MPUF [22]	40.60	-	37.03	Xilinx Artix 7	-	-
Lattice PUF [23]	50.00	-	49.98	Spartan-6	-	50.00
APUF [24]	44.3	96.00	48.45	Spartan-6	234 slices	-
R-XOR [13]	40.00	-	-	Virtex 5	268 LUTs	-
Rec-DAPUF [25]	-	94.80	-	Xilinx Artix 7	64 LUTs, 55 slices	64.90
Our approach	48.75	99.99	52.5	Xilinx Artix 7	64 LUTs, 32 slices	44.50

5.3. Machine learning-attacks

Another critical criterion for a PUF is its resistance to ML-attack. In a previous study, Hospodar *et al.* [26] proves that the ANN technique is better in predicting the response of Arbiter-PUF and XOR Arbiter-PUF compares to the support vector machine (SVM). In our study, the neural network classifier was optimized by using the trial and error method [27]. The trial and error procedure has found that the best prediction accuracy can be achieved by using two hidden layers, five neurons per layer, and the “logsig” activation function. A resilient back propagation has been chosen for the training algorithm of the neural network classifier as it is fast and more accurate than the other training algorithms. 30,000 CRPs were used for the training data set and the remaining 2,000 CRPs were used as the testing data set.

Based on Table 2, it can be seen that lattice PUF method successfully reduces the predictability of response against ML-attack with an ideal value of 50%, followed by Rec-DAPUF and RO-PUF with 64.90% and 85.20%, respectively. In all the mentioned works, the same method was used to predict the response of Arbiter-PUF as in this study which is ANN. Although the resistance to ML-attack was reduced, the area overhead used in each method is high due to the complexity of the design. By implementing our design, the predictability can be reduced to 44.50%. Our random challenge permutation technique to reduce the susceptibility of an Arbiter-PUF against ML-attack consumes low area overhead as it can be implemented by routing obfuscation. Therefore, it is suitable for lightweight security devices.

6. CONCLUSION

In this paper, the obfuscated delay-based PUF has been implemented in diligent Nexys-4 Artix-7 FPGA board. The proposed PUF has been implemented by balancing and constraining the routing using TCL script. The delay analysis results show that the symmetric delay path was achieved by using manual routing for placing the switching components and the arbiter block on the FPGA. As for the evaluation of PUF performance metrics, the proposed PUF achieved uniqueness of 48.75%, reliability of 99.99%, and uniformity of 52.5%. These results show that the obfuscated delay-based PUF achieves good PUF quality. Moreover, we have also shown that the obfuscated delay-based PUF reduces the susceptibility of the conventional Arbiter-PUF to ML-attack from 98% to 44.50% without requiring additional complex circuitry on the FPGA, which would increase the area consumption. These findings indicate that our proposed PUF design is suitable for lightweight identification and authentication applications in resource-constrained IoT devices.

ACKNOWLEDGEMENT




The authors would like to thank Universiti Teknikal Malaysia Melaka and the Ministry of Higher Education Malaysia for the financial funding under Grant No. FRGS/1/2020/TK0/UTEM/02/56 for completing this project.

REFERENCES




- [1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, October 2010, doi: 10.1016/j.comnet.2010.05.010.
- [2] K. Witkowski, "Internet of Things, Big Data, Industry 4.0–Innovative Solutions in Logistics and Supply Chains Management," in *Procedia Engineerin*, vol. 182, pp. 763–769, 2017, doi: 10.1016/j.proeng.2017.03.197.
- [3] Xian-Yi Chen and Zhi-Gang Jin, "Research on Key Technology and Applications for Internet of Things," *Physics Procedia*, vol. 33, pp. 561–566, 2012, doi: 10.1016/j.phpro.2012.05.104.
- [4] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018, doi: 10.1016/j.jisa.2017.11.002.
- [5] S. Bao, H. Yan, Q. Chi, Z. Pang, and Y. Sun, "Transactions on Industrial Informatics A FPGA-Based Reconfigurable Data Acquisition System for Industrial Sensors," in *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1503–1512, Aug. 2017, doi: 10.1109/TII.2016.2641462.
- [6] N. A. Hazari, F. Alsulami, and M. Niamat, "FPGA IP Obfuscation Using Ring Oscillator Physical Unclonable Function," *NAECON 2018-IEEE National Aerospace and Electronics Conference*, 2018, pp. 105–108, doi: 10.1109/NAECON.2018.8556746.
- [7] J. Zhang, Y. Lin, Y. Lyu, and G. Qu, "A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-per- Device Licensing," in *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1137–1150, June 2015, doi: 10.1109/TIFS.2015.2400413.
- [8] S. Tian, A. Krzywosz, I. Giechaskiel, and J. Zefer, "Cloud FPGA Security with RO-Based Primitives," *2020 International Conference on Field-Programmable Technology (ICFPT)*, 2020, pp. 154–158, doi: 10.1109/ICFPT51103.2020.00029.
- [9] M. A. Usmani, S. Keshavarz, E. Matthews, L. Shannon, R. Tessier, and D. E. Holcomb, "Efficient PUF-Based Key Generation in FPGAs Using Per-Device Configuration," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 2, pp. 364–375, Feb. 2019, doi: 10.1109/TVLSI.2018.2877438.
- [10] M. Majzoobi, F. Koushanfar, and S. Devadas, "FPGA PUF using Programmable Delay Lines," *2010 IEEE International Workshop on Information Forensics and Security*, 2010, pp. 1–6, doi: 10.1109/WIFS.2010.5711471.
- [11] 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*, 2015, pp. 559–562, doi: 10.1109/DSD.2015.51.
- [12] B. Habib, K. Gaj, and J. P. Kaps, "FPGA PUF based on programmable LUT delays," *2013 Euromicro Conference on Digital System Design*, 2013, pp. 697–704, doi: 10.1109/DSD.2013.79.
- [13] F. Dan, et al., "A modeling attack resistant r-XOR apuf based on FPGA," *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, 2018, pp. 577–581, doi: 10.1109/SIPROCESS.2018.8600484.
- [14] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. V. Dijk, and S. Devadas, "A Technique to Build a Secret Key in Integrated Circuits for Identification and Authentication Applications," *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*, 2004, pp. 176–179, doi: 10.1109/VLSIC.2004.1346548.
- [15] M. S. Mispan, H. Su, M. Zwolinski, and B. Halak, "Cost-efficient design for modeling attacks resistant PUFs," *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 467–472, doi: 10.23919/DATE.2018.8342054.
- [16] A. Schaller, T. Arul, V. Van Der Leest, and S. Katzenbeisser, "Lightweight anti-counterfeiting solution for low-end commodity hardware using inherent PUFs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8564, pp. 83–100, 2014, doi: 10.1007/978-3-319-08593-7_6.
- [17] N. Pundir, F. Amsaad, M. Choudhury, and M. Niamat, "Novel technique to improve strength of weak arbiter PUF," *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 1532–1535, doi: 10.1109/MWSCAS.2017.8053227.
- [18] 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, doi: 10.1007/978-1-4614-1362-2_11.
- [19] M. H. Mahalat, S. Mandal, A. Mondal, and B. Sen, "An Efficient Implementation of Arbiter PUF on FPGA for IoT Application," *2019 32nd IEEE International System-on-Chip Conference (SOCC)*, 2019, pp. 324–329, doi: 10.1109/SOCC46988.2019.1570548268.
- [20] A. Oun and M. Niamat, "Defense Mechanism Vulnerability Analysis of Ring Oscillator PUFs against Neural Network Modeling Attacks using the Dragonfly Algorithm," *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 378–382, doi: 10.1109/EIT48999.2020.9208320.
- [21] S. S. Zalizvaka, A. A. Ivaniuk, and C. H. Chang, "Low-cost fortification of arbiter PUF against modeling attack," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4, doi: 10.1109/ISCAS.2017.8050671.
- [22] Q. Ma, C. Gu, N. Hanley, C. Wang, W. Liu, and M. O'Neill, "A machine learning attack resistant multi-PUF design on FPGA," *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018, pp. 97–104, doi: 10.1109/ASPDAC.2018.8297289.
- [23] Y. Wang, X. Xi, and M. Orshansky, "Lattice PUF: A Strong Physical Unclonable Function Provably Secure against Machine Learning Attacks," *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2020, pp. 273–283, doi: 10.1109/HOST45689.2020.9300270.
- [24] N. N. Anandakumar, M. S. Hashmi, and S. K. Sanadhya, "Compact Implementations of FPGA-based PUFs with Enhanced Performance," *2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID)*, 2017, pp. 161–166, doi: 10.1109/VLSID.2017.7.
- [25] N. Shah, D. Chatterjee, B. Sapui, D. Mukhopadhyay, and A. Basu, "Introducing Recurrence in Strong PUFs for Enhanced Machine Learning Attack Resistance," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 319–332, June 2021, doi: 10.1109/JETCAS.2021.3075767.
- [26] G. Hospodar, R. Maes, and I. Verbauwhede, "Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability," *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012, pp. 37–42, doi: 10.1109/WIFS.2012.6412622.
- [27] R. Sarkar, S. Julai, S. Hossain, W. T. Chong, and M. Rahman, "A comparative study of activation functions of NAR and NARX neural network for long-term wind speed forecasting in Malaysia," in *Mathematical Problems in Engineering*, vol. 2019, 2019, doi: 10.1155/2019/6403081.

BIOGRAPHIES OF AUTHORS






Mohammad Haziq Ishak    received B.Eng Electronics from Universiti Teknikal Malaysia Melaka, Malaysia in 2021. He is working toward the M.Sc degree in Electronics Engineering with the Universiti Teknikal Malaysia Melaka (UTeM). His M.Sc degree is on the Implementation of a new lightweight authentication scheme using physical unclonable function for FPGA-based IoT applications. He can be contacted at email: M022020030@student.utem.edu.my or haziqsepd@gmail.com.






Mohd Syafiq Mispan    received B.Eng Electrical (Electronics) and M.Eng Electrical (Computer and Microelectronic System) from Universiti Teknologi Malaysia, Malaysia in 2007 and 2010 respectively. He had experienced working in semiconductor industries from 2007 until 2014 before pursuing his Ph.D. degree. He obtained his Ph.D. degree in Electronics and Electrical Engineering from University of Southampton, United Kingdom in 2018. He is currently a senior lecturer in Fakulti Teknologi Kejuruteraan Elektrik dan Elektronik, Universiti Teknikal Malaysia Melaka. His current research interests include hardware security, CMOS reliability, VLSI design, and Electronic Systems Design. He can be contacted at email: syafiq.mispan@utem.edu.my.






Wong Yan Chiew    is from Micro and Nano Electronics (MiNE) research group, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Melaka. She is also an associate professor in Faculty of Electronics and Computer Engineering, UTeM and teaching integrated circuit design, artificial intelligence, microprocessor and computer engineering subjects. She obtained her B.Eng. and MEng (Master degree) in Universiti Teknologi Malaysia in 2005 and 2008. In 2014, she completed her PhD in Electronics Engineering, School of Engineering, The University of Edinburgh, United Kingdom. Her research work focuses on artificial algorithm, analog CMOS design, VLSI design, energy harvesting and power management system. She can be contacted at email: ycwong@utem.edu.my.



Muhammad Raihaan Kamaruddin    received the B.Eng (Electronics and Computer Systems) and M.Eng (Electronics and Information Science) degrees from Takushoku University, Japan. He is working toward the PhD degree in Electronics and Computer Engineering with the Universiti Teknikal Malaysia Melaka (UTeM). His PhD is on the Implementation of bio-inspired robotic navigation system using stochastic computing. He has working experience as lecturer in Universiti Teknikal Malaysia Melaka (UTeM) for 10 years (2010-present). His research interest includes machine learning, robotic and stochastic computing. He can be contacted at email: raihaan@utem.edu.my.



Mikhail Aleksandrovich Korobkov    received Ph.D in a technical science (Radio engineering, including systems and TV devices, Radar and radio navigation) from Ryazan State Radio Engineering University in 2016. He is currently a functional verification and emulation field application engineer at Siemens EDA. Areas of interests are FPGA, PUF, functional/formal verification for FPGA and ASIC designs. He can be contacted at email: korobkov.m.a@yandex.ru.