

# FPGA implementation of artificial neural network for PUF modeling

Mohd Syafiq Mispan<sup>1,2</sup>, Mohammad Haziq Ishak<sup>2</sup>, Aiman Zakwan Jidin<sup>1,2</sup>, Haslinah Mohd Nasir<sup>2</sup>

<sup>1</sup>Micro and Nano Electronics (MiNE) Research Group, Centre for Telecommunication Research and Innovation (CeTRI), Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia

<sup>2</sup>Faculty of Electronics and Computer Technology and Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia

## Article Info

### Article history:

Received Feb 26, 2024

Revised Jul 22, 2024

Accepted Aug 12, 2024

### Keywords:

Computational model

Hardware fingerprinting

Lightweight authentication

Machine learning

Physical unclonable function

## ABSTRACT

Field-programmable gate array (FPGA) is a prominent device in developing the internet of things (IoT) application since it offers parallel computation, power efficiency, and scalability. The identification and authentication of these FPGA-based IoT applications are crucial to secure the user-sensitive data transmitted over IoT networks. Physical unclonable function (PUF) technology provides a great capability to be used as device identification and authentication for FPGA-based IoT applications. Nevertheless, conventional PUF-based authentication suffers a huge overhead in storing the challenge-response pairs (CRPs) in the verifier's database. Therefore, in this paper, the FPGA implementation of the Arbiter-PUF model using an artificial neural network (ANN) is presented. The PUF model can generate the CRPs on-the-fly upon the authentication request (i.e., by a prover) to the verifier and eliminates huge storage of CRPs database in the verifier. The architecture of ANN (i.e., Arbiter-PUF model) is designed in Xilinx system generator and subsequently converted into intellectual property (IP). Further, the IP is programmed in Xilinx Artix-7 FPGA with other peripherals for CRPs generation and validation. The findings show that the Arbiter-PUF model implementation on FPGA using the ANN technique achieves approximately 98% accuracy. The model consumes 12,196 look-up tables (LUTs) and 67 mW power in FPGA.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Mohd Syafiq Mispan

Faculty of Electronics and Computer Technology and Engineering, Universiti Teknikal Malaysia Melaka

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

Email: syafiq.mispan@utem.edu.my

## 1. INTRODUCTION

Internet of things (IoT) enable the ubiquitous electronic devices in which these devices are connected via an internet network, and it is possible to exchange data among them. IoT implementation often requires specific and unique network requirements, which can be programmed or reprogrammed in the field of application with a cost and time-efficient manner. Field-programmable gate array (FPGA) is a foundation for building the next generation of IoT systems since it offers scalability, low latency, and low power [1]-[3]. FPGA can be programmed or reprogrammed according to the requirements of IoT applications. Examples of IoT applications include secure access, smart surveillance cameras, smart homes, and smart meters. All these applications require user-specific data to be processed. Hence, it is very crucial to enable device identification and authentication in IoT applications [4].

Physical unclonable function (PUF) is a technology that can be deployed in FPGA-based IoT applications for device identification and authentication. PUF provides root-of-trust from a hardware layer by exploiting the integrated circuit (IC) manufacturing intrinsic process variations [5]. PUF maps an input known as a challenge to generate a unique output known as a response. The mapping of the challenge and response pairs (CRPs) is unique for a group of similar types of PUFs (i.e., device-specific response). Hence, PUF provides a great and promising capability to be used for device identification and authentication application. Figure 1 depicts the PUF-based identification and authentication process, which consists of two phases; enrollment and authentication. During the enrollment phase, the CRPs of the prover are extracted and stored in the verifier database,  $d$ , in a trusted environment. In the field of application, the prover sends its response ( $\tilde{r}$ ) to the verifier and compares it against the response ( $r$ ) in the database. If both responses are matched, the prover is a genuine or valid device, otherwise, the prover is identified as a fake device.

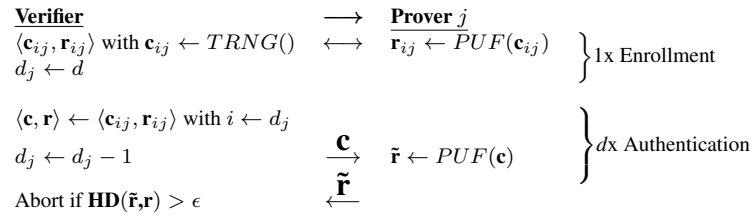


Figure 1. Identification and authentication process using PUF [6], [7]

PUF-based identification and authentication, as described above, has a major drawback of severe area overhead in the verifier database. The CRPs are not allowed to be reused to avoid on-path attack or man-in-the-middle attack [8]. Therefore, the verifier has to store an enormous amount of CRPs to authenticate the PUFs. Storing the PUF computational model is an alternative solution to overcome the severe area overhead in the verifier database [9]-[13]. Aghaie *et al.* [9] developed a technique to build the computational model of delay-based PUFs by using an internal delay sensor known as a time-to-digital converter (TDC) in FPGA. The sensor measures the delay of signals that pass through the switching components in delay-based PUF architecture. Subsequently, the measured delay is used to build the PUF computational model. Although the above method reduces the number of CRPs to build the PUF computational model, the sensors remain on-chip, hence exposing the device to be easily modeled by the adversaries. In other studies [10]-[13], the machine learning (ML) technique is used to model the PUF. Enormous CRPs are measured during the enrollment phase, and subsequently the PUF model is built using ML technique based on the extracted CRPs.

Elsewhere, Idris *et al.* [14] developed a lightweight authentication protocol that is built using the PUF model. The usage of the PUF model in the verifier database and its physical PUF in the prover device without any protection mechanism is insecure, as an adversary can perform a modeling attack by collecting the exposed CRPs. Hence, the protocol in [14] deploys secret pattern recognition to perform mutual authentication between the verifier and prover. In another study, Yue *et al.* [15] proposed an authentication scheme involving the sequence of dynamic random access memory (DRAM) power-up values and convolutional neural network (CNN). Power-up values in memory are random and exhibit device-specific features. CNN is deployed to model these unique features based on the DRAM power-up sequence that has been converted to a two-dimensional (2D) image structure. The proposed authentication scheme requires only the DRAM-PUF model (i.e., unique feature) in the database. Nevertheless, deploying deep learning architecture such as CNN in the proposed authentication scheme requires a huge area as deep learning typically consists of a significant number of layers and a complex computational matrix.

All of the above studies show that deploying the PUF model in the database of verifier is getting the attention of the PUF research community. Nevertheless, the chosen ML technique must be able to build the PUF model in a cost-efficient manner. Moreover, the previous studies only focusing on methodical approach (i.e., building protocol of using PUF model) and/or simulation-level analysis only. Therefore, this study focuses on a PUF computational model development in Xilinx Artix-7 FPGA board using an artificial neural network (ANN) to enable lightweight authentication protocol in FPGA-based IoT applications. The PUF model accuracy, area and power consumption are evaluated and discussed.

## 2. METHOD

$k$ -bit Arbiter-PUF [16], [17] is used as a case study for building the computational model of PUF in FPGA. Figure 2 illustrates the top-level architecture of  $k$ -bit Arbiter-PUF. Arbiter-PUF is chosen in our study as it has a lightweight architecture [18] and  $k$  value is set to 32 to provide considerably enough process variations for Arbiter-PUF implementation in FPGA [19]. There are three major design steps in the development of PUF modeling in FPGA. First, the physical Arbiter-PUF is implemented on FPGA following the methods as described in [20]. Subsequently, random and unique challenges were generated using 32-bit linear-feedback shift register (LFSR) with a primitive polynomial of  $x^{32} + x^{31} + x^{30} + x^{10} + 1$  and applied to the physical Arbiter-PUF to generate the 1-bit corresponding responses. In total, 20,200 CRPs are extracted from the physical Arbiter-PUF for building the PUF model.

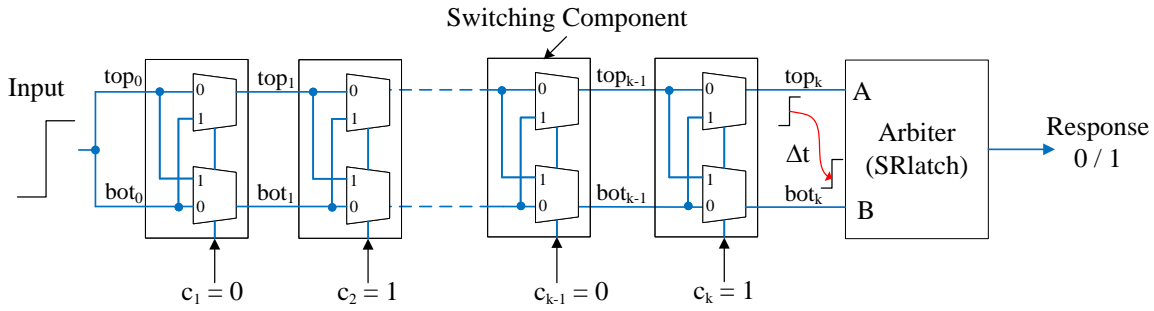


Figure 2.  $k$ -bit Arbiter-PUF architecture

Based on the extracted CRPs, the next step is to build the PUF model in a MATLAB using the ANN technique. A 3-layer of ANN architecture is used, which consists of one input layer, one hidden layer with five neurons, and one output layer. The number of neurons has been determined based on the rule of thumb described in [21]. Log-sigmoid function is used as an activation in the hidden layers, which is given as  $f(x) = \frac{1}{1+e^{-x}}$ . Following [22], a resilient backpropagation algorithm is used as the training algorithm as it provides fast convergence time and optimum prediction accuracy. 20,000 CRPs are used as a training data set, and the remaining 200 CRPs are used as a test data set. The weightage and bias values from the successful training of PUF modeling in MATLAB are extracted for the subsequent design steps.

The third design step is to implement the above ANN architecture (i.e., with the extracted weight and bias values) in Xilinx system generator. Xilinx system generator is a MATLAB Simulink add-on that enables the development of architecture-level FPGA designs using graphical block programming [23]. The design of the 32-bit Arbiter-PUF model in Xilinx system generator is subsequently converted into intellectual property (IP) core. Finally, the IP core, MicroBlaze core processor, and other peripherals are programmed into Xilinx Artix-7 FPGA using Xilinx Vivado Design Suite to validate the functionality of the Arbiter-PUF model as compared to the physical Arbiter-PUF.

## 3. RESULTS AND DISCUSSION

### 3.1. Artificial neural network architecture

As described in section 2, 32-bit Arbiter-PUF can be modeled by using ANN in which the ANN architecture consists of 3 layers which are input, hidden, and output layer. Figure 3 illustrates the top-level architecture of ANN in the Xilinx system generator environment. The number of input at the input layer is equivalent to  $k$ . The feature extraction is also implemented at the input layer to transform the inputs to parity vectors [24]. The transformed inputs are fed to the hidden layer for the subsequent process. In the hidden layer, it consists of five neurons. The extracted weightage and bias values from the ANN modeling in MATLAB are applied in the Xilinx system generator environment for the computational of the neuron's output. The computational process in each neuron can be represented as  $x = \sum_{i=1}^k w_i c_i + \theta$  where  $x$  is the neuron's output,  $c$  is the  $i$ -th transformed input,  $w$  is the weightage, and  $\theta$  is the bias value. Figure 4 depicts the partial computational block diagram to compute the neuron's output in the hidden layer. Block *CMult* is used to compute multiplication of  $wc$  and block *AddSub* is used to compute  $\Sigma$ .

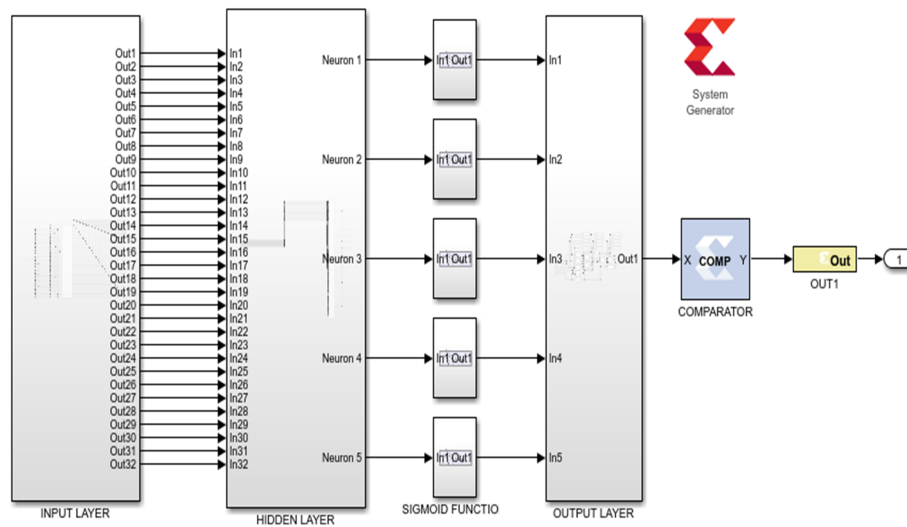


Figure 3. Top-level of ANN architecture for modeling the 32-bit Arbiter-PUF

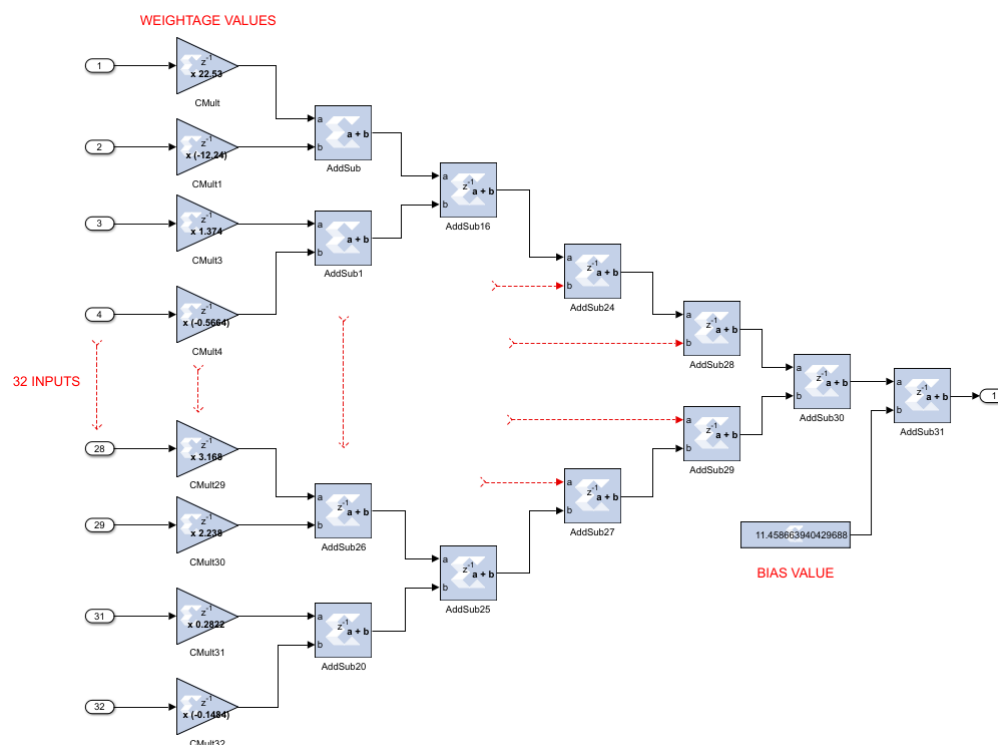


Figure 4. Partial computational block diagram of each neuron in the hidden layer

The output of each neuron,  $x$  is input to the log-sigmoid activation function, which is given as  $f(x) = \frac{1}{1+e^{-x}}$ . The log-sigmoid activation function bounds its output to the range of (0,1). According to Tisan *et al.* [25], a piecewise second-order approximation is used in our study to reduce the computational complexity. Figure 5 illustrates the implementation of the log-sigmoid activation function in Xilinx system generator. Meanwhile, Figure 6 depicts the graph comparison of an ideal log-sigmoid versus piecewise second-order approximation. As can be seen, the approximation technique requires a bigger  $x$  value to bounds its output to the range of (0,1).

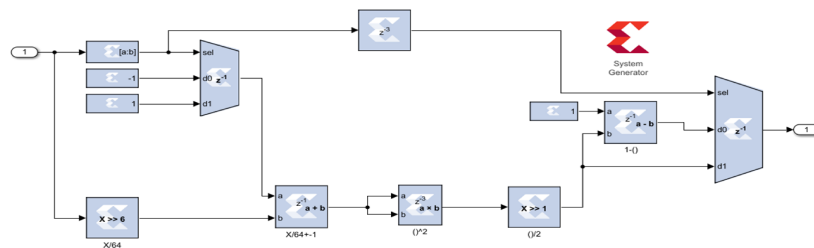


Figure 5. Computational block diagram of sigmoid activation function

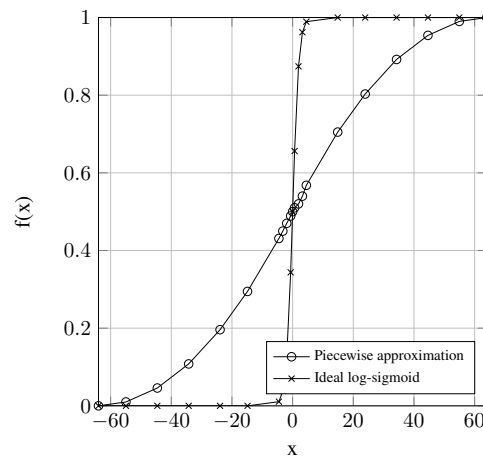


Figure 6. Comparison of an ideal and approximation log-sigmoid activation functions

Subsequently, the activated output is input to the third layer or output layer. Figure 7 illustrates the computational block diagram of an output layer. The computational process in the output layer can be represented as  $o = \sum_{j=1}^n w_j y_j + \theta$  where  $o$  is the output,  $y$  is equivalent to  $f(x)$  (i.e., the activated output),  $w$  is the weightage,  $\theta$  is the bias value, and  $n$  is the total number of neurons. The output layer performs the classification process to classify the response '0' and '1'. In the output layer, an additional block called a comparator is required to counteract the approximated computation of log-sigmoid functions. The design of 32-bit Arbiter-PUF model in Xilinx system generator which based on ANN architecture as discussed above is converted into an IP core. Subsequently, the IP core, MicroBlaze core processor, and other peripherals are programmed into Xilinx Artix-7 FPGA as illustrated in Figure 8 for CRPs collection.

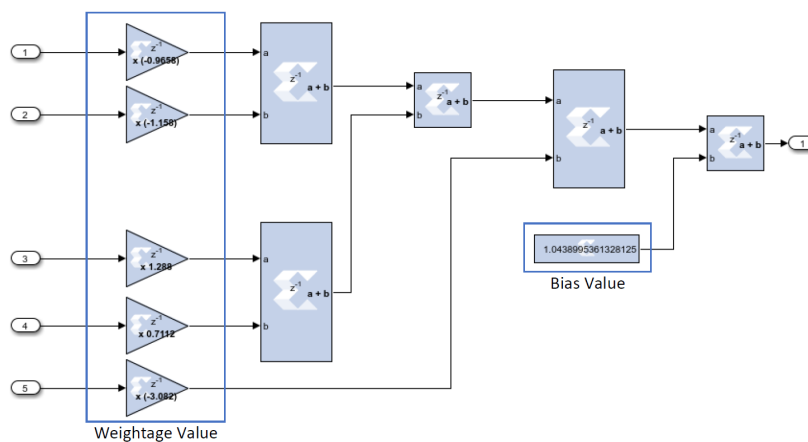


Figure 7. Computational block diagram of an output layer

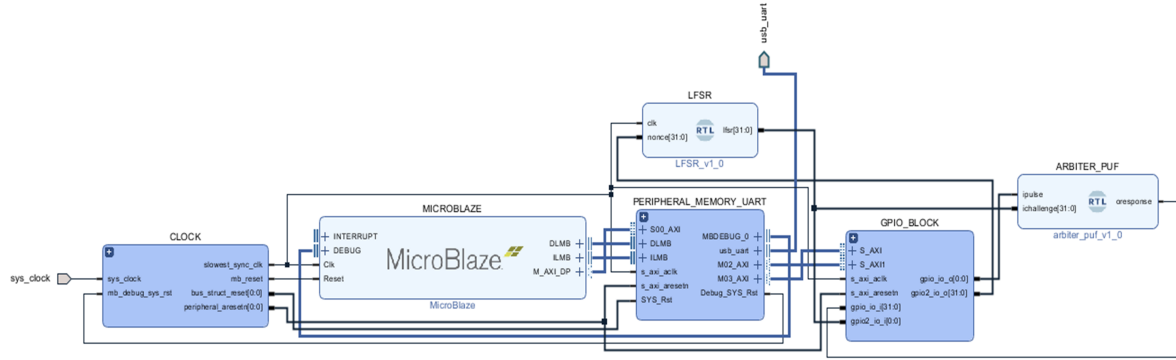


Figure 8. CRPs extraction using microblaze core processor

### 3.2. Modeling accuracy and area consumption

20,000 CRPs are collected from the Arbiter-PUF model using the MicroBlaze core processor as configured in Figure 8. These CRPs are compared against the measured CRPs of physical Arbiter-PUF. Figure 9 depicts the modeling accuracy of the Arbiter-PUF model. The number of CRPs is varied from 400 CRPs up to 20,000 CRPs. The results show that the Arbiter-PUF model achieves very high accuracy, approximately on average 98%. Table 1 lists the area and power consumption of the Arbiter-PUF model. The area and power consumption of the PUF model is higher than the physical PUF because of the complexity of ANN architecture as compared to Arbiter-PUF architecture (see Figure 2). Based on these findings, the Arbiter-PUF is suitable to be used in resource-constrained provers as it consumes insignificant area overhead and power. The corresponding PUF model can be configured in the verifier to perform the authentication process.

As discussed in section 1, storing the PUF model in the verifier significantly reduces the area overhead as compared to storing CRPs for each PUF-based device. All the previously proposed techniques of using PUF model [9]-[15] as discussed in section 1 are methodical approach (i.e., building protocol of using PUF model) and/or simulation-level analysis. Therefore, no comparison of the area overhead and power consumption can be made. The successful PUF model provides scalability in which an unlimited number of authentications can be performed by a prover as it is no longer limited by the number of CRPs stored in the verifier's database.

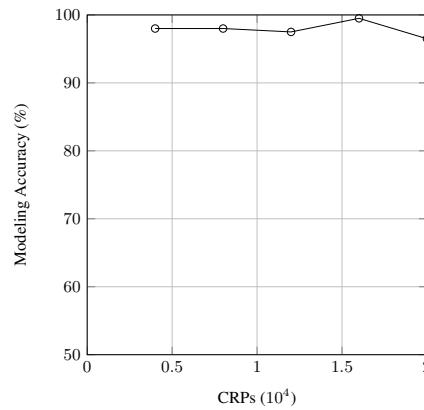


Figure 9. Modeling accuracy of 32-bit Arbiter-PUF modelled using ANN technique

Table 1. Area overhead and power consumption

Unit block	LUTs	Power consumption (mW)
Physical arbiter-PUF	32	< 1
Arbiter-PUF model	12196	67

#### 4. CONCLUSION

In this study, the 32-bit Arbiter PUF has been modeled using the ANN technique in MATLAB and subsequently, the model is implemented on FPGA using Xilinx system generator and Xilinx Vivado Design Suite. The FPGA implementation consumes 12196 LUTs, 67 mW power, and  $\approx 98\%$  accuracy. A successful implementation of the PUF model can replace the conventional CRPs database in the verifier. The verifier consists of the ANN architecture and a database of weightage and biases of provers. PUF model provides scalability in which an unlimited number of authentications can be performed by a prover. Future direction may focus on security enhancement of the verifier database to avoid adversaries' attacks on retrieving the weightage/biases information.

#### ACKNOWLEDGEMENT

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





#### REFERENCES

- [1] M. Elnawawy, A. Farhan, A. A. Nabulsi, A. R. Al-Ali, and A. Sagahyroon, "Role of FPGA in internet of things applications," in *IEEE International Symposium on Signal Processing and Information Technology*, pp. 1-6, 2019, doi: 10.1109/IS-SPIIT47144.2019.9001747.
- [2] A. Magyari and Y. Chen, "Review of state-of-the-art FPGA applications in IoT networks," *Sensors*, vol. 22, no. 19, pp. 1-18, 2022, doi: 10.3390/s22197496.
- [3] T. P. Dinh, C. Pham-Quoc, T. N. Thinh, B. K. D. Nguyen, and P. C. Kha, "A flexible and efficient FPGA-based random forest architecture for IoT applications," *Internet of Things*, vol. 22, pp. 1-14, 2023, doi: 10.1016/j.iot.2023.100813.
- [4] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkarnan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *International Conference for Internet Technology and Secured Transactions*, pp. 336-341, 2016, doi: 10.1109/IC-ITST.2015.7412116.
- [5] M. S. Mispan, B. Halak, and M. Zwolinski, "A survey on the susceptibility of PUFs to invasive, semi-invasive and non-invasive attacks: challenges and opportunities for future directions," *Journal of Circuits, Systems and Computers*, vol. 30, no. 11, pp. 1-37, 2021, doi: 10.1142/S0218126621300099.
- [6] G. E. Suh and S. Devadas, "Physical Unclonable Functions for device authentication and secret key generation," in *ACM/IEEE Design Automation Conference*, pp. 9-14, 2007, doi: 10.1145/1278480.1278484.
- [7] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," *ACM Computing Surveys*, vol. 48, no. 2, pp. 1-42, Nov. 2015, doi: 10.1145/2818186.
- [8] M. S. Mispan, A. Z. Jidin, M. R. Kamaruddin, and H. M. Nasir, "Proof of concept for lightweight PUF-based authentication protocol using NodeMCU ESP8266," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1392-1398, 2021, doi: 10.11591/ijeecs.v24.i3.pp1392-1398.
- [9] A. Aghaie, M. Ender, and A. Moradi, "PUFs physical learning: accelerating the enrollment via delay-based model extraction," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1621-1632, 2022, doi: 10.1109/TETC.2021.3115176.
- [10] A. Aghaie, M. Ender, and A. Moradi, "PUFs Physical Learning: Accelerating the Enrollment via Delay-Based Model Extraction," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1621-1632, Jul. 2022, doi: 10.1109/TETC.2021.3115176.
- [11] A. Ali-Pour, F. Afghah, D. Hely, V. Beroulle, and G. Di Natale, "Secure PUF-based Authentication and Key Exchange Protocol using Machine Learning," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*, IEEE, Jul. 2022, pp. 386-389, doi: 10.1109/ISVLSI54635.2022.00086.
- [12] M. S. E. Quadir and J. A. Chandy, "Embedded systems authentication and encryption using strong PUF modeling," in *IEEE International Conference on Consumer Electronics*, pp. 1-6, 2020, doi: 10.1109/ICCE46568.2020.9043104.
- [13] Y. Yilmaz, S. R. Gunn, and B. Halak, "Lightweight PUF-based authentication protocol for IoT devices," in *2018 IEEE 3rd International Verification and Security Workshop, IVSW 2018*, pp. 38-43, 2018, doi: 10.1109/IVSW.2018.8494884.
- [14] T. A. Idriss, H. A. Idriss, and M. A. Bayoumi, "A Lightweight PUF-Based Authentication Protocol Using Secret Pattern Recognition for Constrained IoT Devices," *IEEE Access*, vol. 9, pp. 80546-80558, 2021, doi: 10.1109/ACCESS.2021.3084903.
- [15] M. Yue, N. Karimian, W. Yan, N. A. Anagnostopoulos, and F. Tehranipoor, "DRAM-Based Authentication Using Deep Convolutional Neural Networks," *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 8-17, 2021, doi: 10.1109/MCE.2020.3002528.
- [16] D. Lim, "Extracting secret keys from integrated circuits," M.S. thesis, Massachusetts Institute of Technology, Cambridge, United States, 2004.
- [17] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," *IEEE Symposium on VLSI Circuits, Digest of Technical Papers*, pp. 176-179, 2004, doi: 10.1109/vlsic.2004.1346548.
- [18] A. Babaei and G. Schiele, "Physical unclonable functions in the internet of things: State of the art and open challenges," *Sensors (Switzerland)*, vol. 19, no. 14, pp. 1-18, Jul. 2019, doi: 10.3390/s19143208.
- [19] T. Xu, "Digital physical unclonable functions : architecture and applications," M.S. thesis, University of California, 2014.
- [20] M. H. Ishak, M. S. Mispan, W. Y. Chiew, M. R. Kamaruddin, and M. A. Korobkov, "Secure lightweight obfuscated delay-based physical unclonable function design on FPGA," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 2, pp. 1075-1083, Apr. 2022, doi: 10.11591/eei.v11i2.3265.
- [21] J. Heaton, *Introduction to neural networks for Java, 2nd Edition*, 2nd ed. Heaton Research, Inc., 2008.





- [22] J. Mathew, R. S. Chakraborty, D. P. Sahoo, Y. Yang, and D. K. Pradhan, "A novel memristor-based hardware security primitive," *ACM Transactions on Embedded Computing Systems*, vol. 14, no. 3, pp. 1–20, May 2015, doi: 10.1145/2736285.
- [23] Xilinx, *Vivado design suite tutorial - model-based DSP design using system generator*, 2020.
- [24] Y. Gao et al., "Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices," in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2016*, IEEE, Mar. 2016, pp. 1–6, doi: 10.1109/PERCOMW.2016.7457162.
- [25] A. Tisan, S. Oniga, D. MIC, and A. Buchman, "Digital implementation of the sigmoid function for FPGA circuits," *Acta Technica Napocensis*, vol. 50, no. 2, pp. 15–20, 2009.

## BIOGRAPHIES OF AUTHORS







**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 Faculty of Electronics and Computer Technology and Engineering, 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.







**Mohammad Haziq Ishak**     received B.Eng. electronics from Universiti Teknikal Malaysia Melaka, Malaysia in 2021. He is working toward two M.Sc. degree in electronics engineering with the Universiti Teknikal Malaysia Melaka (UTeM). His M.Sc. degree research is focusing on the implementation of a lightweight authentication scheme using physical unclonable function for FPGA-based IoT applications. He can be contacted at email: haziqsepd@gmail.com.



**Aiman Zakwan Jidin**     is currently a Ph.D. candidate at Universiti Malaysia Perlis, Malaysia. His research topic is focusing on optimizing memory testing algorithm efficiency for improving fault coverage. Previously, he obtained his M.Eng. in electronic and microelectronic system from ESIEE Engineering Paris, France in 2011, before working as FPGA IP Core Design Engineer at Altera Corporation Malaysia (now part of Intel). He is a full-time lecturer and researcher at Universiti Teknikal Malaysia Melaka (UTeM), in electronic and computer engineering. His research interests include DFT, VLSI, and FPGA system design. He can be contacted at email: aimanzakwan@utem.edu.my.



**Haslinah Mohd Nasir**     received her bachelor degree in electrical - electronic engineering (2008) from Universiti Teknologi Malaysia (UTM), M.Sc. (2016) and Ph.D. (2019) in electronic engineering from Universiti Teknikal Malaysia Melaka (UTeM). She had 5 years (2008–2013) experience working in industry and currently a lecturer in UTeM. Her research interest includes microelectronics, artificial intelligence, and biomedical. She can be contacted at email: haslinah@utem.edu.my.