A Machine Learning Attack Resistant Multi-PUF Design on FPGA

Qingqing Ma*, Chongyan Gu†, Neil Hanley†, Chenghua Wang*, Weiqiang Liu*, Máire O'Neill†

* College of Electronic and Information Engineering (CEIE)
Nanjing University of Aeronautics and Astronautics (NUAA)
China, 211106, e-mail: liuweiqiang@nuaa.edu.cn
† CSIT, Institute of Electronics, Communications & Information Technology (ECIT)
Queen's University Beflast, U.K., BT3 9DT
e-mail: cgu01, n.hanley, m.oneill@qub.ac.uk

Abstract— Current approaches for building physical unclonable function (PUF) designs resistant to machine learning attacks often suffer from large resource overhead and are typically difficult to implement on field programmable gate arrays (FPGAs). In this paper we propose a new arbiter-based multi-PUF (MPUF) design that utilises a Weak PUF to obfuscate the challenges to a Strong PUF and is harder to model than the conventional arbiter PUF using machine learning attacks. The proposed PUF design shows a greater resistance to attacks, which have been successfully applied to other Arbiter PUFs. A mathematical model is presented to analyse the complexity and obfuscation properties of the proposed PUF design. Moreover, we show that it is feasible to implement the proposed MPUF design on a Xilinx Artix-7 FPGA, and that it achieves a good uniqueness result of 40.60 % and uniformity of 37.03 %, which significantly improves over previous work into multi-PUF designs.

1 Introduction

PUFs are a promising lightweight security primitive which use manufacturing process variations to generate a unique digital fingerprint for an electronic device, *e.g.* application-specific integrated circuits (ASICs) or FPGAs. Since even the manufacturer cannot control these variations, PUFs are inherently difficult to clone, providing additional tamper-evident properties. Since no two PUFs are identical, the same $n-$bit input challenge generates a different $n-$bit response on different devices. Such a security primitive provides a number of advantages over current state-of-the-art alternatives and allows for higher security protocols and applications, *e.g.* key storage and device authentication.

PUF architectures can be broadly split between Weak PUF and Strong PUF types based on the size of their challenge-response pair (CRP) space which captures the information about the underlying variation. Weak PUFs have a limited CRP space, and in the extreme case only having a single output. Therefore, they are more suited to applications such as key storage or for seeding a pseudo random number generator (PRNG), where the response never leaves the chip and is only accessed as required. In contrast, Strong PUFs have a large number of possible CRPs, whereby a large number of random challenges will return a random response unique to the challenge, as well as the physical device. By design, this implies a much larger entropy pool. The Arbiter PUF [1] is one of the most widely studied Strong PUFs. However, it has been successfully broken by machine learning based modelling attacks by building up a linear additive delay model for each bit [2]. While some researchers have proposed modifications to im-

978-1-5090-0602-1/18/$31.00 ©2018 IEEE

prove its resistance to modelling attacks, *e.g.* XOR Arbiter PUF [3] and lightweight Arbiter PUF [4], these have also been broken with a sufficient number of CRPs [5, 6]. To counter this, non-linear Arbiter PUFs based on voltage transfer characteristic (VTC) [7] and current mirrors [8] have been proposed specifically to prevent such modelling attacks. However, to date, these approaches have only been simulated for ASICs and have not been proven in practice, and they are not suitable for FPGAs. While Arunkumar *et al.* [7] point out the properties that designers should consider when designing machine learning resistant PUF designs, a practical and feasible implementation strategy has still not been proven. Obfuscating CRPs with some random noise is an efficient method to make mathematical modelling more complex, such that it is difficult for modelling attacks to succeed, *e.g.* [9, 10]. The tradeoff is that these approaches generally require additional hardware resources. Siarhei *et al.* in [11] also proposed a new method to achieve increased reliability which has a small hardware and latency overhead, by removing "Weak" CRPs.

The concept of combining both Weak and Strong PUFs in a PUF design to improve the quality of the overall response has already been presented in previous work, e.g. [12, 13]. In [13], the authors proposed a composite PUF by using smaller PUFs as design building blocks to build a larger challenge-space PUF. However, it exhibits poor uniqueness results for both ring oscillator (RO) PUF and Arbiter PUF (APUF) based composite designs implemented on FPGAs, achieving a uniqueness of less than 10 % for the APUF (the ideal value for uniqueness is 50%). Moreover, none of the above multi-PUF proposals analysed their resistance to modelling attacks. To address the above limitations, we propose a new arbiter-based lightweight MPUF design which shows significantly improved resistance to modelling attacks. More specifically, our scientific research contribution can be summarised as follows:

- We propose a new lightweight PUF design, MPUF, which utilises Weak PUFs to obfuscate the challenges to a Strong PUF to enhance its security against modelling attacks.
- A mathematical model of the proposed MPUF design is presented and complexity analysis is provided. Compared to the conventional Arbiter PUF, the proposed MPUF design has a higher complexity and, hence, is more difficult to attack.
- A case study based on a combination of a weak PUF design [14] and the conventional Arbiter PUF is presented. Two different types of machine learning based modelling attacks, namely logistic regression (LR) and covariance matrix adaptation evolution strategies (CMA-ES), are utilised to investigate its resistance to modelling attacks. The experimental results show that the proposed MPUF achieves a 50% prediction rate using LR compared to 100% for the conventional Arbiter PUF. For a 32-bit challenge the CMA-ES attack also achieves a 100% prediction rate for the conventional Arbiter PUF, whereas for the MPUF design, even with a large sample set of 10,000 CRPs, the prediction rate is less than 80%.
- We validated the proposed MPUF architecture with 22 designs implemented on Xilinx Artix-7 FPGAs. An experimental evaluation of this design shows uniqueness and uniformity results of 40.60 % and 37.03 % respectively, which are the best results reported for a multi-puf design to date, to the best of the authors' knowledge.

The rest of this paper is organised as follows. We present the new lightweight MPUF design in Section 2. In Section 3 we present a mathematical model of the proposed MPUF design and compare it to a conventional Arbiter PUF design. The LR and CMA-ES modelling attacks of the proposed MPUF design are discussed in Section 4 and a performance evaluation is presented in Section 5. We conclude with a summary and discussion of our results and future work in Section 6.

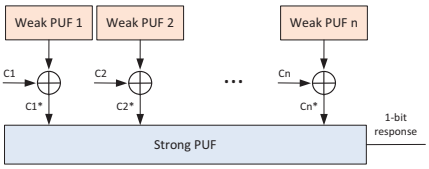

Figure 1: The architecture of proposed MPUF design.

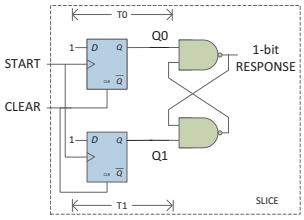

Figure 2: The PicoPUF design in [17].

2 Proposed MPUF Design

2.1 Architecture of proposed MPUF design

The proposed MPUF design comprises an array of n elementary 1-bit cells to generate an $n-$bit response. The design of a 1-bit response generation cell is shown in Figure 1, and is composed of a group of Weak PUFs and a Strong PUF. The outputs (responses) of the Weak PUFs are XORed with an $n-$bit challenge and the result then forms the challenge for the Strong PUF.

Hence, the reliability of the Weak PUF is critical to the performance of the overall MPUF design. Some types of Weak PUF designs have demonstrated high reliability with/without post-processing techniques, *e.g.* DRAM PUF [15], FPGA based PUF identification generator [14], *etc.*. In the next section, the previously proposed lightweight and reliable PUF ID generator design [14], which we will refer to henceforth as PicoPUF, is utilised to show the feasibility of the proposed MPUF. From [14] it is clear that through pre-processing, the PicoPUF design can achieve 100% reliability. Generally, any type of Strong PUF can be utilised to construct the proposed MPUF design, *e.g.* Arbiter PUF [1] or FF-APUF [16]. In this scheme, the challenge to the Strong PUF is completely obfuscated by the Weak PUF.

2.2 MPUF design with PicoPUF and an Arbiter PUF

The PicoPUF PUF design, [14, 17], demonstrates both high reliability and uniqueness, two important metrics in PUF designs. The design of a 1-bit response generation cell is designed to compactly fit into one FPGA slice, as schematically shown in Figure 2. It is very lightweight and can be flexibly placed anywhere in an FPGA. To generate a single bit response, two matched time delay paths, *T0* and *T1*, implemented by two D type flip flops are activated simultaneously by the rising edge of a *START* signal connected to their clock pins after first being reset by *CLEAR*; Due to underlying manufacturing process variation, the propagation times of the signals passing through the flip flops are different, thus creating a race condition between the two delay paths. Cross-couple NAND gates are used as an arbiter to decide which signal arrived first, and outputs the response as a binary value 0 or 1.

The APUF is one of the best studied Strong PUF designs, and an example is shown in Figure 3. To form a 1-bit APUF, two parallel cascaded $n-$stage multiplexer (MUX) chains and one flip flop are used. Two MUXs are constructed as either a cross- or straight- through connection based on the input challenge bit. After propagation through the final $n-$th stage, the two signals arrive at the cross-coupled NAND gates which determines which signal arrived first, and outputs a 1-bit response, either 0 or 1, accordingly.

The proposed 1-bit MPUF design, as shown in Figure 4, is composed of n PicoPUF designs and an $n-$

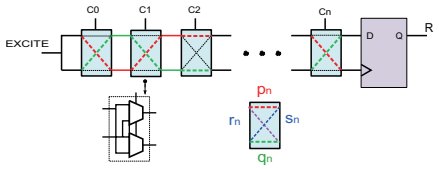

Figure 3: The Arbiter PUF design [1].

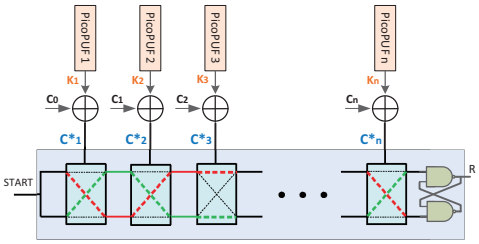

Figure 4: The MPUF design based on a PicoPUF and an Arbiter PUF.

stage APUF design. The response of the i^{th} PicoPUF is XORed with the challenge bit C_i to mask the original challenge bit and a new challenge bit C_i^* is generated. $C_1, C_2, ..., C_n$ is the challenge input into the MPUF and $C_1^*, C_2^*, ..., C_n^*$ is the challenge generated from the PicoPUFs and used by the APUF.

3 Mathematical Model

The proposed MPUF design is composed of the $n-$ PicoPUF designs [14] and an APUF design. Using both PicoPUF and APUF models, the proposed MPUF design model can be described as

$$\Delta^*(n) = \vec{P^*} \cdot \vec{d} \tag{1}$$

where, $\Delta^*(n)$, the delay difference of the two delay paths in the APUF in the proposed design, can be represented as the inner product of the parity vector $\vec{P^*}$ and the constant vector $\vec{d} = (\alpha_1, \alpha_2 + \beta_1, ..., \alpha_n + \beta_{(n-1)}, \beta_n)$. α_n and β_n can be calculated as shown in Equation 2 and Equation 3 [18].

$$\alpha_n = \frac{p_n - q_n - r_n + s_n}{2} \tag{2}$$

$$\beta_n = \frac{p_n - q_n + r_n - s_n}{2} \tag{3}$$

where, p_n, q_n, r_n and s_n are the notations of the delay segments in each stage of the conventional Arbiter PUF. The Parity check vector $\vec{P^*} = (p_0^*, p_1^*, ..., p_n^*)$ is defined as

$$p_k^* = \prod_{i=k+1}^{n} C_i \oplus K_i \tag{4}$$

where, K_i is the output of the i^{th} PicoPUF design, and C_i is the i^{th} bit of the input challenge.

Comparing the mathematical models of both the conventional APUF [18] by itself and the proposed MPUF designs, we can see that the proposed MPUF design demonstrates higher complexity than the conventional APUF since the actual input challenge to the APUF is obfuscated and masked.

4 Machine Learning Attacks on Proposed MPUF Design

As mentioned above machine learning based modelling attacks have been shown to expose the vulnerability of PUF circuits [2, 5, 6]. Although improvements have been suggested to increase their resistance to machine learning attacks, *e.g.* [3, 19, 4], these can still be successfully attacked using machine learning based techniques to build a model, *e.g.* support vector machine (SVM), LR, evolutionary strategies (ES), *etc.*. Currently, the reliability based CMA-ES attack, proposed by Becker [6], is the most effective modelling attack against PUFs. It uses repeated measurements for the same challenge to observe the reliability of the response bits and then feeds this observation into a fitness function to find the 'best-fit' delay parameters.

In this paper the two most widely used machine learning based algorithms, LR and CMA-ES, are used to evaluate the proposed MPUF design due to their effectiveness in modelling Arbiter based PUF architectures to date.

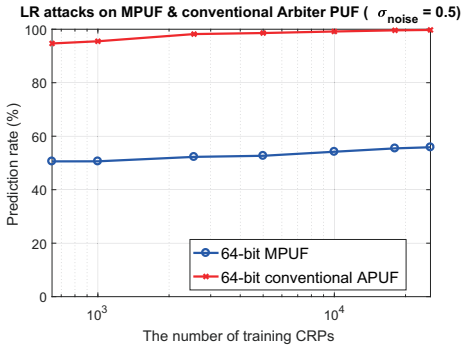

Figure 5: The prediction rates for conventional Arbiter PUF and proposed MPUF designs by LR attack.

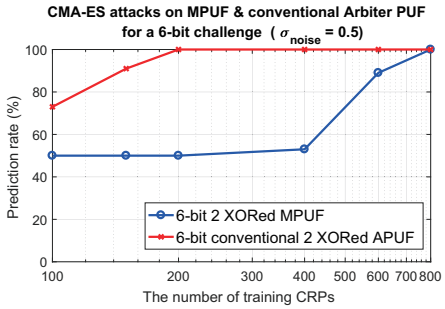

Figure 6: The prediction rates for conventional Arbiter PUF and proposed MPUF designs by CMA-ES attack.

4.1 LR Results

In this experiment, we use an open source implementation of LR with RProp programmed by Ulrich *et al.* [2] in Python, which are available from [20].

To predict the proposed MPUF design using a LR algorithm, a group of tests on different numbers of training samples is carried out. Figure 5 shows the prediction rates for both the conventional APUF and proposed MPUF designs with training sample sets of size 3000, 5000, 10000, and 20000, with the separate test sample data set the same size as the training one. It can be seen that the conventional APUF can be successfully predicted with high reliability and the proposed MPUF demonstrates good resistance to the LR machine learning attacks.

4.2 CMA-ES Results

For the reliability based CMA-ES algorithm, we follow the work by Becker [6] and the source code in Matlab is adopted from [21]. A Gaussian distribution is employed to generate and simulate a group of random numbers for the delay elements in the PicoPUF and the conventional Arbiter PUF. To model the impact of noise a variable is added to the delay difference of each APUF model with Gaussian distribution of $norm\ (0, \sigma_{noise})$. An n-XORed APUF(*i.e.* where n responses from n different APUFs are XORed) is evaluated. In this work, a 2-XORed APUF and a 2-XORed MPUF, XORed 2 responses from 2 different APUFs/MPUFs, are evaluated by the CMA-ES attack. In the modelling attacks, there is an exponential increase in the number of CRPs required with each additional XOR. Figure 6 shows the prediction rates for both the conventional 2-XORed APUF and proposed MPUF designs with training sample sets of size 100 to 800, and assuming a 6-bit challenge. It can be seen that the conventional APUF can be successfully predicted by using 200 training samples whereas at least 800 are needed to predict the proposed MPUF design.

Figure 7 depicts the prediction rates for both the conventional APUF and proposed MPUF designs with training sample sets of size 1000 to 10000, and assuming a 32-bit challenge. The number of training samples needed will exponentially grow by increasing the number of delay stages, *i.e.* the bit length of the challenge. For a 32-bit challenge, even with a large sample set of 10,000 CRPs the prediction rate is less than 80%. Therefore, it is clear that the proposed MPUF design will be will be significantly harder to attack than the conventional APUF for larger challenges.

5 Performance Evaluation

As the proposed MPUF design exhibits good resistance to LR machine learning attacks, it is also expected to achieve good PUF design metrics. In

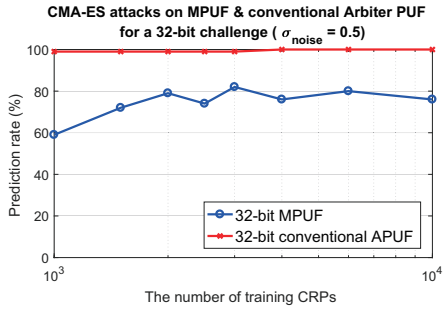

Figure 7: The prediction rates for conventional Arbiter PUF and proposed MPUF designs by CMA-ES attack.

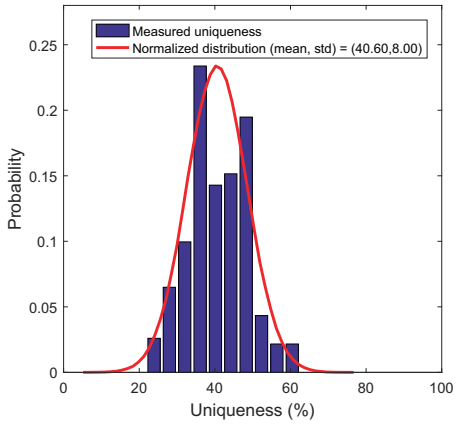

Figure 8: Uniqueness result for proposed MPUF design.

this paper, the key metrics of uniqueness and uniformity are evaluated. To evaluate these metrics, the proposed MPUF design is implemented on Digilent Nexys 4 boards that comprise Xilinx Artix-7 FPGAs. Two MPUF designs have been implemented on each of 11 FPGAs producing a total of 22 individual implementations for testing.

5.1 Uniqueness

Uniqueness measures inter-chip variation by evaluating how well a particular PUF circuit design can be differentiated between k different devices. Ideally, a PUF circuit is expected to produce an average inter-chip hamming distance (HD) of 50% by comparing the response from two devices supplied with the same challenge. The uniqueness, representing the average inter-chip HD, is defined as:

$$\text{Uniqueness} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\text{HD}(\text{R}_i, \text{R}_j)}{\text{n}} \times 100 \tag{5}$$

where, R_i and R_j represent N-bit responses from two PUF circuits Φ_i and Φ_j supplied with the same challenge C.

Figure 8 shows a histogram of the uniqueness result for the proposed MPUF design, which has an empirical mean of 40.6 % and a standard deviation (STD) of 8 %. This is equivalent to the uniqueness value of the flip flop based Arbiter PUF (FF-APUF) design [16]. Moreover, compared to the uniqueness results from 5.44 % to 10.82 %) achieved by previous work on multi-PUF [13], the proposed MPUF design demonstrates a significantly higher capability to differentiate between different devices.

5.2 Uniformity

Uniformity represents the proportion of zeros and ones in a PUF response. Ideally it should be 50%, *i.e.* half ones and half zeros in a response, making it difficult for an attacker to guess the response of a particular device. For device Φ_i and an N-bit response the Hamming Weight (HW) percentage is defined in Equation 6.

$$\text{HW}(\Phi_i) = \frac{1}{N} \sum_{j=1}^{N} R_{i,j} \times 100 \tag{6}$$

where, $R_{i,j}$ is the j-th bit of the response from the i-th device.

The uniformity result for the proposed MPUF design is shown in Figure 9 and has an empirical mean of 37.03 % and a standard deviation (STD) of 6.65 %. The result is equivalent to the uniformity result of [13]. In future work, these metrics will be evaluated over a larger sample set.

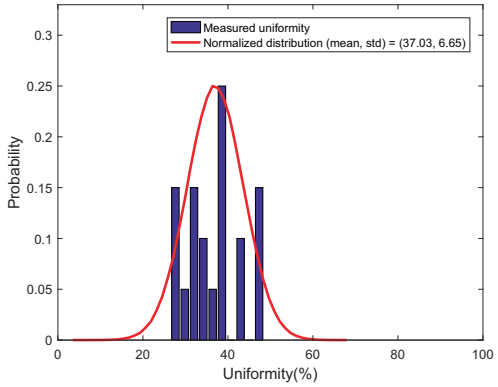

Figure 9: Uniformity result for proposed MPUF design.

6 Conclusion

In this paper, we propose a new MPUF design with resistance to machine learning attack by using a Weak PUF to obfuscate the challenge of a Strong PUF design. A case study is presented by constructing a MPUF using the PicoPUF [14] and a conventional APUF design. Two widely employed machine learning based attack techniques, LR and CMA-ES, are utilised to analyse the resistance of the proposed MPUF design to such attacks. The proposed MPUF design achieves a 50% prediction rate using the LR attack compared with 100% for the conventional Arbiter PUF design. Using CMA-ES, although the proposed design can be successfully predicted for a small challenge size, it is still significantly more resistant to such attacks than the APUF. However, when the challenge size is increased to just 32 bits the CMA-ES attack can only achieve a prediction rate of approximately 80% for the MPUF design, even with a large sample set of 10,000 CRPs, whereas it can still achieve a 100% prediction rate for the conventional Arbiter PUF. Two MPUF designs are also implemented on each of 11 Xilinx Artix-7 FPGAs. The uniqueness and uniformity metrics for the proposed MPUF design exhibit good results of 40.60 % and 37.03 % respectively. This significantly improves upon previous work into multi-PUF and illustrates the design's feasibility for implementation on FPGA.

Acknowledgements

This work is supported by grants from Natural Science Foundation of Jiangsu Province (BK20151477), and National Natural Science Foundation of China (61401197 and 61771239). This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2016-0-00399, Study on secure key hiding technology for IoT devices [KeyHAS Project]) and other project(s).

References

[1] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. 9th ACM Conference on Computer and Communications Security*, CCS '02, pp. 148–160, 2002.

[2] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proc. 17th ACM Conference on Computer and Communications Security, CCS*, pp. 237–249, 2010.

[3] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th Annual Design Automation Conference*, DAC '07, pp. 9–14, 2007.

[4] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure PUFs," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 670–673, 2008.

[5] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF modeling attacks on simulated and silicon data," *IACR Cryptology ePrint Archive*, vol. 2013, p. 112, 2013.

[6] G. T. Becker, *The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs*, pp. 535–555. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[7] A. Vijayakumar and S. Kundu, "A novel modeling attack resistant PUF design based on non-linear voltage transfer characteristics," in *Proc. Design, Automation And Test in Europe Conference And Exhibition (DATE)*, pp. 653–658, 2015.

[8] R. Kumar and W. Burleson, "On design of a highly secure PUF based on non-linear current mirrors," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust HOST'14, Arlington, VA, USA*, pp. 38–43, 2014.

[9] Y. Gao, G. Li, H. Ma, S. F. Al-Sarawi, O. Kavehei, D. Abbott, and D. C. Ranasinghe, "Obfuscated challenge-response: A secure lightweight authentication mechanism for puf-based pervasive devices," in *Proc. IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 1–6, March 2016.

[10] J. Ye, Y. Hu, and X. Li, "RPUF: Physical unclonable function with randomized challenge to resist modeling attack," in *Proc. IEEE Asian Hardware-Oriented Security and Trust (AsianHOST'16)*, pp. 1–6, Dec 2016.

[11] S. S. Zalivaka, A. A. Ivaniuk, and C.-H. Chang, "FPGA implementation of modeling attack resistant arbiter PUF with enhanced reliability," in *Proc. 18th International Symposium on Quality Electronic Design (ISQED'17)*, pp. 313–318, March 2017.

[12] S. T. C. Konigsmark, L. K. Hwang, D. Chen, and M. D. F. Wong, "System-of-pufs: Multilevel security for embedded systems," in *Proc. International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp. 1–10, Oct 2014.

[13] D. P. Sahoo, S. Saha, D. Mukhopadhyay, R. S. Chakraborty, and H. Kapoor, "Composite PUF: A new design paradigm for physically unclonable functions on FPGA," in *Proc. IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'14)*, pp. 50–55, May 2014.

[14] C. Gu, N. Hanley, and M. O'neill, "Improved reliability of FPGA-based PUF identification generator design," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 10, pp. 20:1–20:23, May 2017.

[15] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, "Dram-based intrinsic physically unclonable functions for system-level security and authentication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 1085–1097, March 2017.

[16] C. Gu, Y. Cui, N. Hanley, and M. O'neill, "Novel lightweight FF-APUF design for FPGA," in *Proc. 29th IEEE International System-on-Chip Conference, (SOCC'16)*, Sept. 2016.

[17] C. Gu, J. Murphy, and M. O'Neill, "A unique and robust single slice FPGA identification generator," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'14)*, pp. 1223–1226, June 2014.

[18] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, vol. 13, no. 10, pp. 1200–1205, 2005.

[19] 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," in *Proc. Symposium on VLSI Circuits*, pp. 176–179, 2004.

[20] "http://www.pcp.in.tum.de/code/lr.zip, 2010.."

[21] N. Hansen, "The CMA evolution strategy: a comparing review," *Towards a new evolutionary computation*, pp. 75–102, 2006.