Modelling Attack Analysis of Configurable Ring Oscillator (CRO) PUF Designs

Modelling Attack Analysis of Configurable Ring Oscillator (CRO) PUF Designs

Modelling Attack Analysis of Configurable Ring Oscillator (CRO) PUF Designs

Jack Miskelly*, Chongyan Gu*, Qingqing Ma†, Yijun Cui†, Weiqiang Liu†, Máire O'Neill*

***Abstract*— Physical Unclonable Functions (PUFs) have emerged as a lightweight security primitive for resource constrained devices. However, conventional delay-based Physical Unclonable Functions (PUFs) are vulnerable to machine learning (ML) based modelling attacks. Although ML resistant PUF designs have been proposed, they often suffer from large overheads and are difficult to implement on FPGA. Lightweight ML resistant FPGA compatible designs have been proposed which make use of combined multi-PUF designs, incorporating a set of weak PUFs to obscure the challenge to a strong PUF in order to increase the difficulty of model building. In such designs any unreliability in the main PUF is amplified by unreliability in the masking PUFs. For this reason strong PUFs suitable for FPGA that can achieve high reliability, such as the Configurable Ring Oscillator (CRO) PUF, are a promising option. In this paper a mathematical model of the CRO PUF is presented. We show that models of traditional CRO PUFs can be trained to above 99% prediction rate using the Linear Regression and CMA-ES strategies. A proposed multi-PUF design based on the previously proposed arbiter MPUF is evaluated with the same methods. It is shown that even with challenge obfuscation the CRO PUF can be predicted with greater than 90% accuracy. It is shown that with the addition of a second XORed PUF the ML resistance can be increased further with a maximum prediction rate of 86%.**

I. INTRODUCTION

Physical Unclonable Functions, or PUFs, are a category of hardware based security primitives which use manufacturer process variation to create a unique digital 'fingerprint' of a circuit or device. As the variations which form the fingerprint are below the tolerance of manufacturing processes even the manufacturer cannot intentionally replicate an instance of a PUF.

PUFs can be broadly divided into two categories based on the quantity of possible unique Challenge-Response Pairs (CRPs) possible in a given architecture. Weak PUFs have a relatively limited number of CRPs, and in some cases only a single CRP. Due to this small CRP set they are suitable only for limited applications such as key generation and as a component of Pseudo-Random Number Generators (PRNG). Examples of proposed weak PUF types are the SRAM PUF [1], the FPGA based PUF identification generator [2], and the DRAM PUF [3]. In contrast strong PUFs have a large set of CRPs, ideally increasing exponentially with PUF size, and can be used directly for authentication without additional cryptographic hardware. Examples of proposed strong PUFs include the arbiter PUF [6], ring-oscillator PUF [4], and configurable ring-oscillator PUF [5].

*CSIT, Institute of Electronics, Communications, and Information Technology (ECIT) Quen's University Belfast, UK, BT3 9DT jmiskelly08, cgu01, m.oneill @qub.ac.uk

†College of Electronic and Information Engineering (CEIE), Nanjing University of Aeronautics and Astronautics (NUAA) China, 211106 liuweiqiang@nuaa.edu.cn

A. MACHINE LEARNING ATTACKS ON PUFs

Early attempts to model PUFs focused on the physical reproduction of PUF instances with some success such as the cloning of the SRAM PUF by Helfmeier et al. [20]. In recent years a second approach based on digitally modeling PUFs using Machine Learning (ML) strategies has become increasingly prominent. The resultant model, trained with a small acquired subset of the PUF CRPs, can predict the response of the PUF to a given challenge - in effect digitally cloning it. The first ML attacks against PUFs were published by Ruhrmair et al. [7] in 2010, using a mathematical model of the well known arbiter PUF originally proposed in 2004 [8] and various ML algorithms to predict the response of arbiter PUFs.

Subsequent work [10] has demonstrated that a range of strong PUF architectures previously thought to be entirely immune to replication are in fact able to be modeled in this way. Multiple ML techniques have been demonstrated to be capable of PUF modeling, such as Support Vector Machine (SVM) [21], Logistic Regression (LR) [7], and Evolutionary Strategies such as the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [7].

B. BUILDING ML RESISTANT PUFs

Modifications to existing strong PUF architectures to impede ML attacks have been proposed. Early proposals such as the XOR arbiter [5] and lightweight arbiter [11] were successful only in raising the amount of training CRPs needed to model the system [9] [10].

As many of the ML attacks are based on the principle that most PUF responses are complex but fundamentally linear systems, and hence relatively easy to model, the use of components with non-linear, and hence harder to model responses, can confound ML attacks. There are several proposed architectures of this type including proposals based on voltage transfer characteristics [12], and feedthrough logic [22]. However, while these approaches have been successful they come with a fairly high design cost and are unsuitable for use in FPGAs or other applications where non-standard components are not viable.

Another proposed strategy is to obscure or mask the challenge to the PUF, for example by using a rotating shift register tied to a second, obscured, clock as in [23]. These

methods are of particular interest as they use only standard components and thus are implementable on FPGA.

The idea of using multiple PUFs in an architecture has been proposed as a means to improve PUF response such as in [14] and [15]. This concept has also been proposed as a means to impede ML attacks on arbiter PUFs [16]. However, there are several challenges to overcome to make such a design work practically. In some proposed multi-PUF (MPUF) architectures the overall uniqueness of the PUF is lower than it would be for the component PUFs individually as in [16]. Additionally when the input to the strong PUF is dependent on the output of a weak PUF or PUFs any error in the weak PUF(s) will amplify existing error in the strong PUF. For this reason the most promising candidate PUF architectures for use in multi PUF designs are those with strong uniqueness and reliability.

C. CONTRIBUTIONS

The contributions of this paper can be summarized as follows:

- An ML modeling resistant MPUF architecture based on the weak PUF architecture proposed in [2] (referred to from here on as the 'PicoPUF') and CRO PUF [5] is proposed. The CRO PUF is a desirable candidate for the basis of a multi-PUF design due to its high reliability, the key metric in multi-PUF architectures.
- A mathematical model of a CRO PUF is described for use in ML attacks.
- The proposed architecture is evaluated against two common ML attacks, the Covariance Matrix Adaptation Evolution Strategy and Logistic Regression. It is shown that the traditional CRO architecture is vulnerable to both these attacks.
- It is shown that the proposed CRO based MPUF design is resistant to the CMA-ES attack but can still be predicted with greater than 90% accuracy by the Logistic Regression attack.
- The same evaluation is performed using LR with multiple XORed CRO based MPUF, showing a maximum prediction rate of 86%.

II. MODELING OF CRO PUFS

A. CONFIGURABLE RING OSCILLATOR PUFS

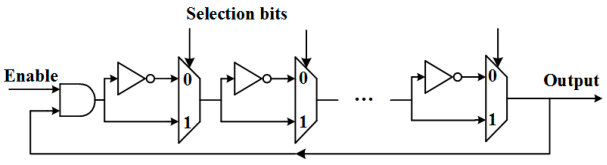

Fig. 1. Architecture of a configurable ring oscillator PUF [18]

A Ring Oscillator (RO) PUF uses rings of delay buffers to generate frequencies which vary based on the low level disorder of the individual delay components. In an RO PUF many of these rings are constructed and two rings, selected by the input challenge, are compared to generate a single output bit based on whether the frequency of the first or second oscillator is higher. The size of the CRP set is determined by how many rings there are in the PUF overall. The CRO PUF [5], shown in Figure 1, is a variant of the RO PUF where the multiple rings are replaced with two rings where each delay stage has two delay elements and a multiplexer to select between them. In a CRO PUF the challenge determines whether the upper or lower delay element will be used at each stage in effect constructing a unique RO. The output frequencies are then compared to derive the output as in the normal RO. The main advantage of CRO over RO is that it is much more efficient in terms of space and component usage. The RO based PUF architectures can achieve consistently high reliability compared to other strong PUF architectures [19].

B. A MATHEMATICAL MODEL OF CRO PUFS

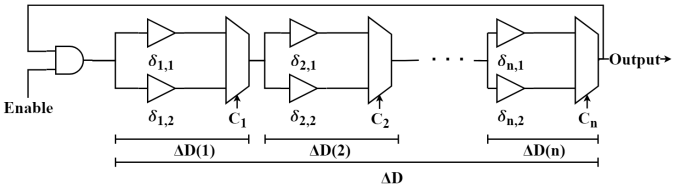

Fig. 2. CRO PUF diagram showing delay components of the upper CRO

Architecturally a CRO PUF is similar to the arbiter PUF for which a well defined mathematical model already exists. The derivation of a mathematical model of a CRO PUF is shown below, which can be used to perform ML modeling attacks of such PUFs.

Each stage of a CRO PUF consists of four delay components. An upper and lower delay path for each of the two CROs. The frequency is determined by the total delay ΔD, which is the sum of the delay components from whichever path, upper or lower, that is selected by the challenge bit at each stage.

$$\Delta D(i) = \Delta D_{upper}(i) - \Delta D_{lower}(i) \tag{1}$$

$$\Delta D_{upper}(i) = \frac{1-C_i}{2}\delta_{i1} + \frac{1+C_i}{2}\delta_{i2}$$

$$\Delta D_{lower}(i) = \frac{1-C_i}{2}\delta'_{i1} + \frac{1+C_i}{2}\delta'_{i2} \tag{2}$$

$$\Delta D(i) = \frac{1-C_i}{2}(\delta_{i1} - \delta'_{i1}) + \frac{1+C_i}{2}(\delta_{i2} - \delta'_{i2}) \tag{3}$$

$$Let \quad \delta_i^{\alpha} = \delta_{i1} - \delta'_{i1},$$

$$\delta_i^{\beta} = \delta_{i2} - \delta'_{i2} \tag{4}$$

Hence the overall delay from which the frequency is derived for an n-stage PUF can be described as a linear sum of vector dot products.

$$\Delta D = \sum_{i=0}^{n} \Delta D(i) = \overrightarrow{P_{\alpha}} \cdot \overrightarrow{W_{\alpha}} + \overrightarrow{P_{\beta}} \cdot \overrightarrow{W_{\beta}} \tag{5}$$

$$\overrightarrow{P_\alpha} = \left\{\frac{1-C_1}{2}, \frac{1-C_2}{2}, \ldots, \frac{1-C_n}{2}\right\},$$

$$\overrightarrow{P_\beta} = \left\{\frac{1+C_1}{2}, \frac{1+C_2}{2}, \ldots, \frac{1+C_n}{2}\right\},$$

$$\overrightarrow{W_\alpha} = \left\{\delta_1^\alpha, \delta_2^\alpha, \ldots, \delta_n^\alpha\right\},$$

$$\overrightarrow{W_\beta} = \left\{\delta_1^\beta, \delta_2^\beta, \ldots, \delta_n^\beta\right\} \quad (6)$$

The frequency of each PUF instance is a product of the overall delay ΔD and the number of clock cycles per measurement. The CRO PUF response is generated by the comparison of the response of two such instances to the same challenge, generating a binary 1 if the frequency of the upper CRO is higher and binary 0 if the frequency of the upper CRO is lower.

This model is used in section IV below to simulate a CRO PUF and to evaluate it against the CMA-ES and LR ML attacks.

III. Architecture of Proposed CRO Multi-PUF Design

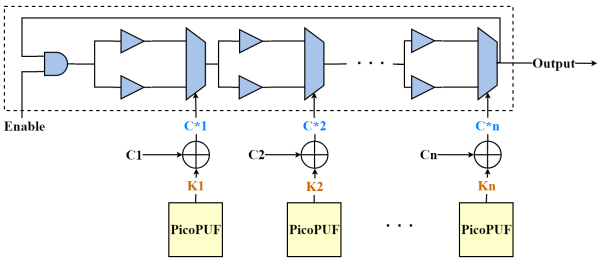

Fig. 3. Proposed CRO MPUF architecture

The proposed CRO MPUF consists of n 1-bit weak PUFs, each of which is XORed with a single bit of an n bit challenge to form the input challenge to the CRO PUF. As mentioned above in such a multi-PUF design any error in either PUF will amplify error in the other PUF therefore it is vital that the weak masking PUFs be highly reliable. High uniqueness is also desirable as low uniqueness will make it easier for the mask to be modeled. In order to be preferable to other methods of input masking such as shown by Cao et al. [23] the masking PUFs must use a minimum amount of resources and ideally require no post processing.

Due to these requirements the design used for the masking PUFs is the previously proposed PicoPUF, generating a binary 0 or 1 depending on whether the upper or lower path is faster. An individual PUF of this type can be implemented on just one FPGA slice and can achieve a reliability of approximately 100% [2]. The design of the masking PUFs is shown in Figure 4. As the PicoPUF is very lightweight preference can be given to the placement of the larger CRO PUF.

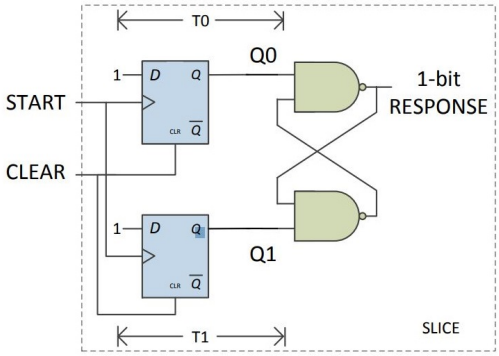

Fig. 4. PicoPUF architecture [2]

IV. ML Resistance of CRO MPUF

As mentioned above there is a large body of work demonstrating ML attacks against various PUF architectures and using a variety of ML strategies [7][10][21]. The proposed CRO MPUF has been evaluated against two of the most prominent techniques, Logistic Regression (LR) and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) following on from the work in [16], as shown in [16], can be confounded by input masking. They are therefore a good measure of the overall ML resistance of the proposed PUF.

A. CMA-ES ATTACK

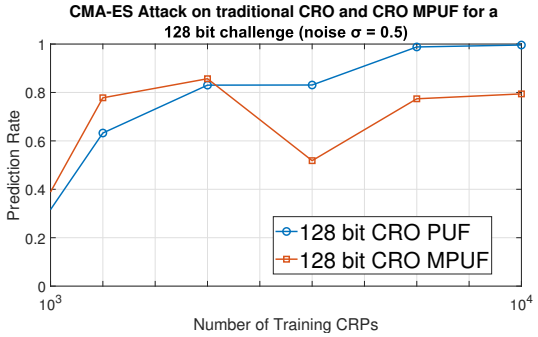

Fig. 5. Prediction rates for conventional CRO PUF and the proposed CRO MPUF with CMA-ES

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [9] is a reliability based evolutionary strategy algorithm. As with all evolutionary strategies it works by progressing through 'generations' in which a number of randomly varied versions of the model are generated and the 'fittest' of these permutations (as selected by the fitness function) forms the starting point of the next generation. Over multiple generations the model gradually fits to the parameters of the system being modeled.

In this work the source code used for the CMA-ES algorithm has been adapted from [17] and [16]. Each simulated delay element is modeled as a Gaussian distributed random

number. This is the case for the delay elements of the simulated masking PicoPUFs and the simulated CRO PUF. To model the impact of noise on the training of the ML model a noise variable is added to the frequency of each CRO PUF in the form of a Gaussian distribution of norm (0, σ_{noise}).

Figure 5 shows the prediction rates for a conventional CRO PUF and the proposed CRO MPUF with training sample sizes of 1000 to 10000 samples for a 128 stage PUF with $\sigma_{noise} = 0.5$. It can be seen that the conventional CRO PUF can be predicted with greater than 98% accuracy while the CRO MPUF prediction rate is less than 80% even with a relatively large 10000 training samples. Even with a large amount of training data the CMA-ES algorithm cannot predict the response of the CRO MPUF to a degree that would compromise the security of the PUF.

B. LR ATTACK

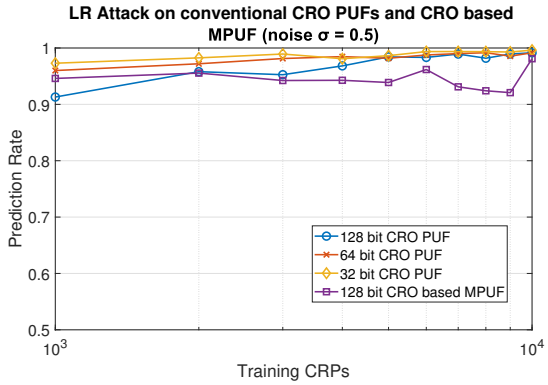

Fig. 6. Prediction rates for conventional CRO PUF and the proposed CRO MPUF with LR

Logistic Regression (LR) is an ML strategy in which the correlation between an independent and dependent variable of a known training set is used to construct a linear model of the system. The LR method has been successfully used in previous work to model PUFs such as the arbiter PUF [7], which like the CRO PUF has a linear additive mathematical model. The LR implementation used for this work is adapted from an open source implementation of LR with RProp programmed by Ulrich et al. [7] using Python. The original implementation can be found here [13].

Figure 6 shows the prediction rates for a conventional CRO PUF at sizes of 32, 64, and 128 bits compared to a 128 bit implementation of the proposed CRO based MPUF design, using the LR method for training sample sizes of 1000 to 10000 samples. The conventional CRO PUF design can be predicted with greater than 99% accuracy given 10000 training CRPs even for a relatively large 128 stage PUF. The prediction of the CRO based MPUF is less reliable, but interestingly is consistently greater than 90% and often greater than 95%. This high prediction rate implies that even with input masking the LR algorithm is able to derive a model of the more complex CRO MPUF architecture if supplied with enough training CRPs.

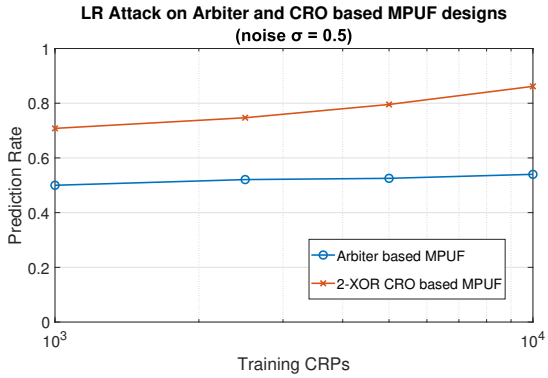

Fig. 7. LR prediction rates for a 2 XORed CRO based MPUF compared to the Arbiter based MPUF [16]

A common method of increasing the modeling complexity of a PUF design is to take two or more instances of the PUF and XOR the outputs against each other to form the final output response. An example of this method is the XOR arbiter PUF [5]. Figure 7 shows a comparison of the LR prediction rates of two XORed CRO based MPUFs to that of the previously proposed arbiter based MPUF [16]. The addition of the second XORed PUF instance lowers the prediction rate to approximately 86% even after 10000 training CRPs. This is sufficiently low as to prevent overall prediction of the PUF response, but is significantly higher than the arbiter based MPUF. Moreover, the 2 XORed design requires a doubling in the size of the PUF from the original proposed design.

C. PERFORMANCE EVALUATION

As shown in the sections above the proposed CRO based MPUF architecture is able to successfully confound the CMA-ES algorithm, reducing the prediction rate to less than 80% on average. However, the LR algorithm is still able to attain a prediction rate of greater than 90% even with input masking. The prediction rate can be reduced to below 90% with the addition of a second CRO MPUF instance to XOR the response bits with. This lowers the prediction rate to a degree that the LR attack cannot accurately predict the PUF response but at the cost of doubling the required resources of the implementation.

A possible cause of the vulnerability of the CRO PUF to LR even with input masking may be that unlike for arbiter PUFs each stage of the CRO delay is independent from the other stages. As the mask for each challenge bit is a constant value only a moderate, linearly increasing amount of additional complexity is added at each stage. While this increases the overall system complexity and hence the amount of training data required to produce an accurate LR model it does not fundamentally make the system any less linear. The addition of greatly increased complexity in

the form of a second XORed PUF instance is required to create a substantial level of ML resistance.

V. CONCLUSIONS

In this paper we propose a new multi-PUF design based on the previous work in [16] and the Configurable Ring Oscillator PUF, wherein the input challenge to the CRO PUF is obscured by an XOR mask of PicoPUF instances, with the aim of creating a high reliability modeling resistant strong PUF. The proposed architecture is evaluated against two machine learning based modeling techniques, the Covariance Matrix Adaptation Evolution Strategy and Logistic Regression, both of which have been used in previous works to attack the arbiter PUF. A linear mathematical model of the CRO PUF is described for use in machine learning attacks. It is shown that both CMA-ES and LR are capable of modeling conventional CRO PUFs at greater than 98% accuracy and that LR is capable of modeling the obscured CRO MPUF with greater than 90% accuracy under the same conditions. The addition of a second XORed CRO MPUF lowers the prediction rate to a more useful 86

The XORed version of the proposed design is sufficiently ML resistant to potentially be viable for practical use on FPGA. The additional resource usage of an individual instance of the proposed design is low due to the lightweight nature of the PicoPUF mask, however to be considered truly ML resistant the resource usage must be doubled by the introduction of the second XORed PUF instance. Nonetheless the proposed architecture may prove to be a better alternative to the previously proposed arbiter based MPUF if further work can show that the high reliability of the CRO PUF can improve on the reliability of the arbiter MPUF. The proposed design remains a promising option for an ML resistant FPGA based PUF.

ACKNOWLEDGMENTS

This work was partly supported by the Institute for Information & communications Technology Promotion(IITP) grant funded by the Korean government(MSIT) (No. 2016-0-00399, Study on secure key hiding technology for IoT devices [KeyHAS Project]), by the EPSRC (EP/N508664/-CSIT2), by Nature Science Foundation of Jiangsu Province (BK20151477) and by National Natural Science Foundation China (61771239).

REFERENCES

[1] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, FPGA Intrinsic PUFs and Their Use for IP Protection, in Cryptographic Hardware and Embedded Systems - CHES 2007, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 6380.

[2] C. Gu, N. Hanley, and M. Oneill, Improved reliability of FPGA-based PUF identification generator design, ACM Trans. Reconfigurable Technol. Syst., vol. 10, 2017, pp. 20:120:23.

[3] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication, IEEE Trans. Very Large Scale Integr. Syst., 2017.

[4] G. E. Suh and S. Devadas, Physical Unclonable Functions for Device Authentication and Secret Key Generation, in 44th ACM/IEEE Des. Autom. Conf., 2007, pp. 914.

[5] A. Maiti and P. Schaumont "Improved Ring oscillator PUF: An FPGA-friendly secure primitive" J. Cryptology, 2010, pp. 375-397.

[6] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, Silicon physical random functions, in Proc. the 9th ACM conf. on Comput. and commun. secur. - CCS 02, 2002, p. 148.

[7] U. Rhrmair, F. Sehnke, J. Slter, G. Dror, S. Devadas, and J. Schmidhuber, Modeling attacks on physical unclonable functions, in Proc. the 17th ACM Conf. Comput. Commun. Secur. - CCS 10, no. i, 2010, pp. 237.

[8] D. Lim, Extracting Secret Keys from Integrated Circuits, 2004.

[9] G. T. Becker, The Gap Between Promise and Reality: On the Insecurity of XOR Arbiter PUFs, pp. 535555. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[10] U. Ruhrmair, J. Solter, 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.

[11] M. Majzoobi, F. Koushanfar, and M. Potkonjak, Lightweight secure PUFs, in Proc. IEEE/ACM Int. Conf. on Comput.-Aided Des., 2008, pp. 670673.

[12] A. Vijayakumar and S. Kundu, A novel modeling attack resistant PUF design based on nonlinear voltage transfer characteristics, in Proc. Des., Automation And Test in Europe Conf. And Exhibition, 2015, pp. 653658.

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

[14] 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 Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2014, pp. 110.

[15] 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 Symp. on Hardware-Oriented Security and Trust (HOST14), 2014, pp. 5055.

[16] 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 Conf. (ASP-DAC), Jeju, 2018, pp. 97-104.

[17] N. Hansen, The CMA evolution strategy: a comparing review, Towards a new evolutionary computation, 2006, pp. 75102.

[18] Gao, M.; Lai, K.; Qu, G. A Highly Flexible Ring Oscillator PUF. In Proc. of the 51st Annual Design Automation Conference, 2014, pp. 16.

[19] R. Maes, V. Rozic, I. Verbauwhede, P. Koeberl, E. van der Sluis and V. van der Leest, "Experimental evaluation of Physically Unclonable Functions in 65 nm CMOS," 2012 Proc. of the ESSCIRC (ESSCIRC), 2012, pp. 486-489.

[20] C. Helfmeier, C. Boit, D. Nedospasov and J. P. Seifert, "Cloning Physically Unclonable Functions," 2013 IEEE International Symp. on Hardware-Oriented Security and Trust (HOST), 2013, pp. 1-6.

[21] 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.

[22] S. R. Sahoo, S. Kumar and K. Mahapatra, "A Modified Configurable RO PUF with Improved Security Metrics," 2015 IEEE International Symp. on Nanoelectronic and Information Systems, 2015, pp. 320-324.

[23] Y. Cao, X. Zhao, W. Ye, Q. Han and X. Pan, "A Compact and Low Power RO PUF with High Resilience to the EM Side-Channel Attack and the SVM Modelling Attack of Wireless Sensor Networks", Sensors, vol. 18, no. 2, p. 322, 2018.