IEICE Electronics Express, Vol.17, No.2, 1–6

LETTER

FPGA implementation of a challenge pre-processing structure arbiter PUF designed for machine learning attack resistance

Wei Ge[1a)], **Shenxin Hu**[1], **Jiquan Huang**[1], **Bo Liu**[1], **and Min Zhu**[2]

Abstract Utilizing the randomness caused by process variations in chip manufacturing, PUF can provide identification and verification by generating unique challenge-response pair. The output response of Arbiter PUF is due to path delay differences from different input challenge. However, due to the strong linear correlation between the response and challenge of the Arbiter PUF, the attacker can model the APUF through a machine learning algorithm. This paper proposes a challenge pre-processing structure arbiter PUF (CPP-APUF), which increases the unknowingness of the input challenge, and improves the APUF's ability to resist machine learning attacks. The 64-stage CPP-APUF is implemented based on FPGA, the machine learning algorithm is used to attack the CPP-APUF. The output response prediction accuracy is lower than 61.33%, which is effective against the modeling attack of machine learning. Finally, the challenge-response pair obtained from experimentally verifies the PUF characteristics.

Keywords: APUF, CRP, FPGA, attack resistance
Classification: Electron devices, circuits and modules

1. Introduction

PUF has received extensive attention in the fields of identification and verification, password storage and exchange, and has become a research hotspot in the field of information security [1, 2, 3]. The PUF relies on complex and uncontrollable changes in the chip manufacturing process to generate unique signatures. The PUF is considered to be a response to the input challenge, and the challenge corresponding to the response is called a Challenge-Response Pair (CRP) [4, 5, 6, 7]. PUF can be divided into two categories according to the number of CRPs: weak PUF and strong PUF [8]. Weak PUF usually has only one pair or a very small number of CRPs. On the other hand, the strong PUF has an exponential CRP corresponding to the PUF size, which can flexibly realize the security purpose of using only one specific CRP once.

As one of the typical strong PUF structures, Arbiter PUF (APUF) is the earliest proposed silicon PUF structure [9]. The APUF propagates the rising pulse through two identical delay paths and uses the challenge to select the path delay variation to generate the response [10, 11]. However, due to the strong linear correlation between the response and challenge of the Arbiter PUF, APUF's vulnerability in modeling attacks through machine learning (ML) algorithms and side channel attacks is well documented in the literature [12, 13, 14, 15, 16, 17, 18].

The difficulty of machine learning attacks is increased by disrupting the linear relationship of CPR by modifying the structure of traditional APUF. For example, G. Edward Suh et al. proposed the XOR APUF structure, and the final output was obtained by the XOR of several APUF responses [19]. B. Gassend et al. proposed the Feed Forward Arbiter PUF (FF APUF) protection structure, adding a pre-feedback structure based on the APUF, and the arbitration result of the previous stage is used as the selection signal of the latter stage arbiter unit [20]. Qingqing Ma et al. proposed a Multi-PUF protection structure to disturb the relationship of challenge-response pairs by XOR masking the original input challenge [21]. Siarhei S et al. proposed the Multiple Input Signature Register (MISR) protection structure and the T flip-flop XOR protection structure. The response of the current output depends not only on the current challenge but also on the previously input challenge [22, 23].

With the in-depth study of APUF modeling attacks, these APUF variant structures are still vulnerable to modeling attacks. In theory, neural network modeling method can successfully attack the protective structure by learning any nonlinear structure iteratively without analyzing the implementation details of the protective structure [14]. The Multi-APUF uses the value generated by the weak PUF to perform an XOR operation with the challenge. For the same Multi-APUF entity, the weak PUF is a single fixed value that can be cracked by modeling attacks. MISR-APUF uses a set of determined arithmetic units to process the challenge. Although a pre-configured unknown parameter participates in the operation, the attacker can calculate the classification of the intermediate results according to the known arithmetic unit, and then divide the challenge response into different groups according to the intermediate results, and finally model different APUF models according to different groups [12, 13, 14, 15, 16, 17, 18]. Therefore, in order to improve the ability of APUF to resist modeling attacks, this paper proposes a new challenge pre-processing APUF structure (CPP-APUF), the eigenvalues of the structure contain unpredictable unknown parameters and the challenge after processing varies with the original input challenge. Compared with the existing APUF protection structure, CPP-APUF can effectively resist the attack of neural network algorithm modeling.

[1]National ASIC System Engineering Technology Research Center, Southeast University, Nanjing 210096, P.R.C
[2]Wuxi Research Institute of Applied Technologies, Tsinghua University, Wuxi 214000, P.R.C
a) duiker@seu.edu.cn

DOI: 10.1587/elex.16.20190670
Received November 4, 2019
Accepted December 3, 2019
Publicized December 17, 2019
Copyedited January 25, 2020

The main contributions of this paper are as follows:
1. An effective challenge pre-processing structure APUF is proposed, which increases the unknowingness and uncertainty of the input challenge to improve the APUF's security against machine learning attacks.
2. Both the mathematical model and the design complexity analysis of the CPP-APUF are presented. The processing effect of the challenge pre-processing is analyzed.
3. Implement the modeling attack experiment for CPP-APUF on the FPGA platform. The modeling attack is carried out by four machine learning algorithms. The modeling accuracy of Linear Regression, Logic Regression, and SVM, is finally maintained at 54.00%, and the Back Propagation Neural Networks modeling algorithm only reaches 61.33%. The experimental result shows that compared with the existing APUF protection structure, the proposed CPP-APUF protection structure can effectively resist the modeling attack of neural network algorithm.

The rest of this article is organized as follows. In the second part, a new challenge preprocessing structure APUF (CPP-APUF) is proposed. In the third part, the CPP-APUF structure is implemented on the FPGA platform and the effects of resisting modeling attacks are discussed. Then, the PUF characteristics of the CPP-APUF are compared with the ideal PUF in the fourth part. Finally, we summarize and discuss our work in Section V.

2. Design of CPP-APUF

2.1 Structure of challenge pre-processing APUF

The principle of the CPP-APUF structure against the modeling attack is shown in Fig. 1. The CPP structure first pre-processes the original challenge of the input APUF, and then the processed challenge signal is sent to the APUF to generate an output response.

The CPP structure consists of an improved RS flip-flop, with the outputs of two adjacent RS flip-flops simultaneously acting as inputs to adjacent flip-flops. Compared to traditional RS flip-flops with only two input signals, the improved RS flip-flop contains four inputs, including the original input challenges C_i and $C_{(i+1)}$, and the aliasing signal $C'_{(i-1)}$ and $C'_{(i+2)}$. It is particularly noted that when the cascaded RS flip-flop is located at the edge of the structure, such as the AND gate corresponding to C_0 and C_m, the input confusion signals $C'_{(0-1)}$ and $C'_{(m+1)}$ use two C_0 and C_m signals are substituted. Compared with the truth table of the traditional RS flip-flop, the improved RS flip-flop has more input signal combination modes, including 16 input combinations corresponding to the 4 output combinations.

2.2 Mathematical model of CPP structure

The CPP-APUF proposed in this paper includes challenge pre-processing structure and the conventional APUF design. According to the mathematical model of APUF and the mathematical principle of the CPP structure, the model of CPP-APUF is described as Equation 1.

$$\Delta t_v = \vec{\gamma} \cdot \vec{\tau} \tag{1}$$

Where Δt_v is the delay difference of the upper and lower paths of the selected unit in the APUF circuit; $\vec{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_k)$ is the parity vector, which is defined by the Equation 2. K_i is the output of the i^{th} CPP design, and C_i is the i^{th} bit of the input challenge. Especially, the K_i and $K_{(i+1)}$ will not be "0" at the same time, and their output will have at least one "1". $\vec{\tau}$ is a constant vector, which is calculated by the combination of delay parameters $(\delta t_1^0, \delta t_1^1, \delta t_2^0, \delta t_2^1, \ldots, \delta t_k^0, \delta t_k^1)$ of each selected unit, and is defined by Equation 3.

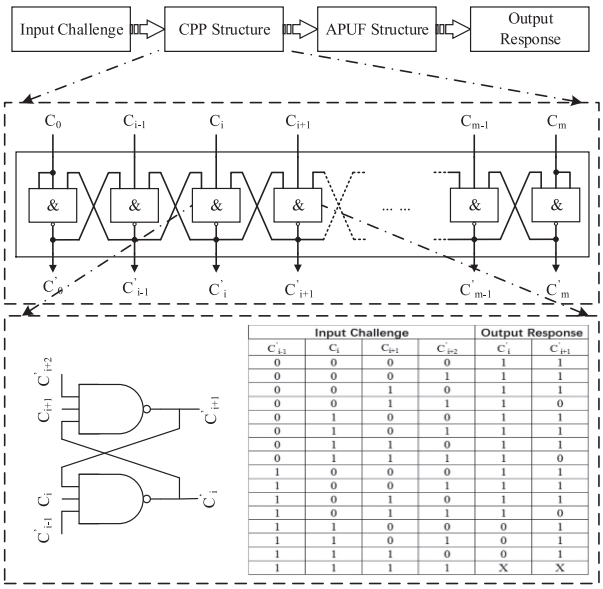

Fig. 1. Structure of challenge pre-processing APUF

$$\gamma_k = \prod_{i=k-1}^{n} (C_i \oplus K_i \oplus K_{i+2}), \quad K_i \cap K_{i+1} = 1 \tag{2}$$

$$\vec{\tau} = \frac{1}{2} \begin{pmatrix} \delta t_1^0 - \delta t_1^1 \\ \delta t_1^0 - \delta t_1^1 + \delta t_2^0 - \delta t_2^1 \\ \vdots \\ \delta t_{k-1}^0 - \delta t_{k-1}^1 + \delta t_k^0 - \delta t_k^1 \\ \delta t_k^0 - \delta t_k^1 \end{pmatrix} \tag{3}$$

Comparing the mathematical models with the conventional APUF [22], which response relies on the fixed internal delay parameters and input challenges. The proposed CPP-APUF design demonstrates higher complexity and unpredictability than the conventional APUF since the actual input challenge to the APUF is obfuscated and masked.

2.3 Effect analysis of CPP-APUF

For a conventional RS flip-flop, the output response signal is an indeterminate state "X" if and only if both the input challenge signals are "1" at the same time, and the value of the indeterminate state depends on the manufacturing process difference of the internal circuit of the RS flip-flop. In such a case, the output responses C'_i and $C'_{(i+1)}$ may only be one of "0, 1" or "1, 0". The challenge-response pair of the improved RS flip-flop complies with the characteristics of the weak PUF, the output response is determined by the circuit manufacturing process difference and only when the input challenge signals are simultaneous "1". Taking the 4-stage APUF as an example, for each 4-bit input challenge, the processed output response depends on the number of

consecutive “1” in the challenge data. In a specific scenario, when the input challenge is “0011”, the output response may be “1101” or “1110”.

The challenge pre-processing structure utilizes the weak PUF characteristics of the RS flip-flop structure to increases the unknown parameters. At the same time, change the linear relationship between the output response and the input challenge. Increasing the system entropy value through unpredictable variables can improve the difficulty of resisting modeling attacks.

3. FPGA implementation of CPP-APUF

Based on Altera FPGA platform, the logic circuit and data acquisition of CPP-APUF are realized [24, 25, 26, 27]. The system block diagram of this experiment is shown in Fig. 2. The PUF and control logic circuits are implemented on the FPGA platform. The module Challenge Receiver is used to receive a 64-bit random PUF challenge, and the control module Pulse Generator generates a pulse signal. The module UART is used for receiving challenge and transmitting PUF response, and the PUF circuit generates a response when the module Response Pack receives the response and packs it into 8-bit data, it sends the data to the PC through the module UART. The PC side sends the random challenge through the UART and receives the PUF response sent by the FPGA. Finally, the challenge response data is analyzed by the MATLAB program.

Based on the FPGA platform, this paper implements APUF with 32 cascaded challenge processing structure, and collects 170,000 stable CRPs, of which 150000 CRPs are used for training models, and 20,000 CRPs are used to detect the prediction accuracy of the model. The modeling results are shown in Fig. 3. Under the training model of more than 3000 CRPs, the modeling accuracy rates of the three modeling algorithms, Linear Regression, Logic Regression and SVM, are basically no longer changed, maintaining at around 54%, while the BPNN algorithm modeling accuracy is still rising high. Under the training of 10,000 CRPs, the BPNN algorithm has a modeling accuracy rate of 52.8%. Under the training of 50,000 CRPs, the modeling accuracy rate is 56.73%. Under the training of 100,000 CRPs, the modeling accuracy rate is as high as 59.32%, Under the training of 150,000 CRPs, the correct rate of modeling reached 61.33%, and then gradually stabilized, but it was significantly lower than the correct rate of 97.4% of the original APUF modeling model in the same case, indicating that the cascaded RS challenge processing structure has good capability to resist modeling attacks.

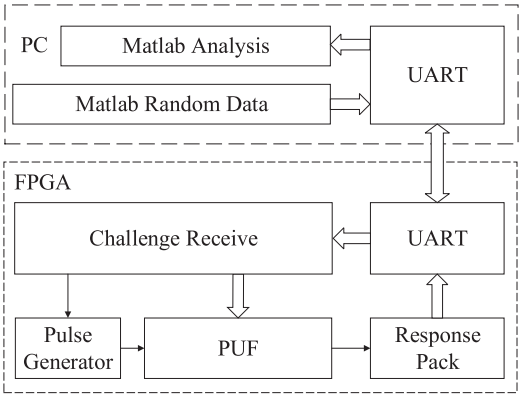

Fig. 2. Experimental system block diagram

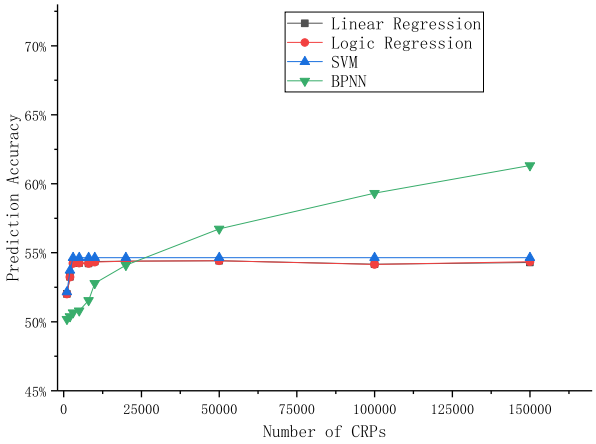

Fig. 3. Modeling attacks on CPP-APUF by different algorithms

The experiment compares the resist modeling attack effects of different APUF protection structures under BPNN algorithm, as shown in Fig. 4. The structure of 3XOR-APUF contains three APUF with shared challenges, and the final output response is obtained by the “XOR” of three APUF responses. The FF-APUF structure incorporates a pre-feedback structure on the basis of the APUF, and the pre-arbitration result serves as a selection signal for the post-stage switch unit. In this experiment, three feedback loops were adopted. The challenge of the Multi-PUF structure is “XOR” with a fixed random value based on the challenge of the APUF. The XOR-APUF and FF-APUF introduce finite random values through nonlinear protection structures such as “XOR” and feedback. And the Multi-PUF increases the difficulty of modeling attacks by “XOR” a fixed random value with the input challenges. However, CPP-APUF not only increased the random value of the protection structure by using the improved RS flip-flop, but also increased the difficulty of modeling attack with the random value changing with the input challenges. The original APUF structure can achieve a prediction accuracy of 97.60% under the training of only 5,000 CRPs. After training above 30,000 CRPs, the prediction accuracy exceeds 98.50%. The 3XOR-APUF structure contains three APUFs. With more than 30,000 CRPs, the prediction accuracy of the model is as high as 95.85% to 96.41%. For the FF-APUF structure, three feedback loops are used in this experiment. Under the training of more than 50,000 CRPs, the prediction accuracy rate can reach 90.78% to 93.56%. The challenge of the Multi-PUF structure is “XOR” with a fixed random value based on the challenge of the APUF. Adopting the modeling attack method of [3], the prediction accuracy can reach 95.73% under the training of 3000 CRPs. With more than 20,000 CRPs, the prediction accuracy can reach more than 98.10%. Under the training of 50,000 CRPs, the prediction accuracy of CPP-APUF is 56.73%. As the training number increases, the accuracy

rate does not change drastically. Even under the training of 150,000 CRPs, the prediction accuracy is only 61.33%.

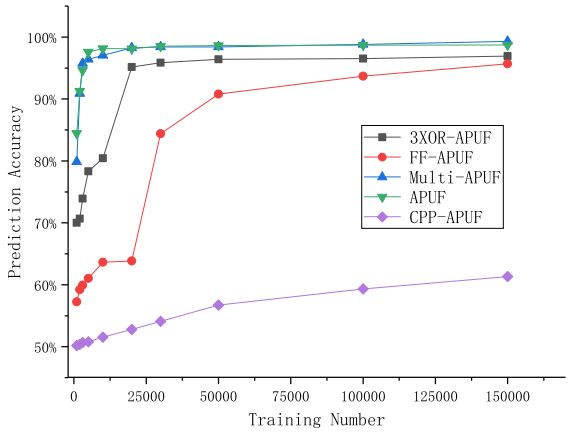

Fig. 4. BPNN algorithm modeling attack on different APUF protection structures

4. Characteristics of CPP-APUF

Although CPP-APUF shows good resistance to modeling attacks, it is still necessary to consider the impact of CPP structure on PUF characteristics. In this paper, 16 APUFs with CPP structure are implemented on the FPGA platform by QUARTUS II, and the uniqueness, stability, and uniformity of each CPP-APUF are tested [28, 29, 30].

4.1 Uniqueness of CPP-APUF

The uniqueness of PUF refers to the difference between the responses generated by multiple PUF entities with the same structure but independent of each other. Ideally, the value of uniqueness tends to be 50%. Uniqueness is an important characteristic for testing the success of PUF design, the unique calculation and evaluation formula is expressed by the following Equation 4. Where k indicates that a total of k PUF entities participate in the test, $HD(R_i, R_j)$ represents the inter-chip Hamming distance of the PUF entity, and n represents the bit width of the output response of each PUF entity.

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \quad (4)$$

The experimental response is collected, and the Hamming distance between each two PUF is calculated. The distribution map of Hamming distance is shown in Fig. 5. The uniqueness of CPP-APUF is calculated by Equation 4 to be 51.06%, which is close to the uniqueness of 50% of the ideal PUF requirement.

4.2 Stability of CPP-APUF

The stability of the PUF refers to the difference between the PUF output responses when the external factors change when the same PUF entity continuously inputs the same challenge. Ideally, the stability of a PUF entity tends to be 100%. The stability calculation formula of the PUF entity is as shown in Equation 5. Where m represents the number of measurements, $HD(R_g, R_t)$ represents the on-chip Hamming distance, R_g is a reference response measured under certain circumstances, R_t is the response obtained after inputting the same challenge for t times, m is the number of times the same challenge is input, n is the bit width of the PUF entity output response.

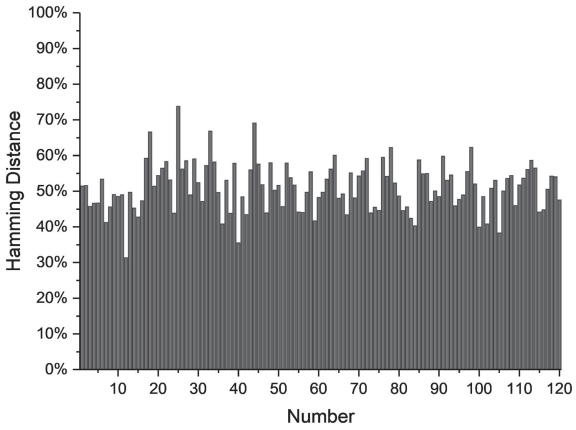

Fig. 5. Hamming Distance between 16 CPP-APUF

$$Reliability = 1 - \frac{1}{m} \sum_{t=1}^{m} \frac{HD(R_g, R_t)}{n} \times 100\% \quad (5)$$

The stability of the CPP-APUF response is shown in Fig. 6. The overall stability of the CPP-APUF output response is lower than that of the original APUF. The average value is 99.67%, which is 0.08% lower than the average of the stability of the original APUF.

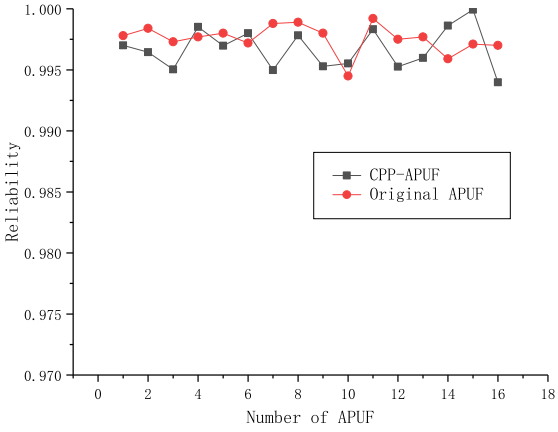

Fig. 6. Comparison of the stability of 16 CPP-APUF and original APUF

4.3 Uniformity of CPP-APUF

The uniformity of PUF refers to the probability that 1 and 0 appear in all responses generated by the same PUF entity. Ideally, the uniformity of the PUF tends to be 50%. The calculation of the uniformity is as shown in Equation 6. Where $r_{i,j}$ is the value of the j^{th} bit of the i^{th} response, and n is the bit width of an output response, Equation 6 calculates the proportion of "1" in the response.

$$Uniformity = \frac{1}{n} \sum_{j=1}^{n} r_{i,j} \times 100\% \quad (6)$$

The uniformity distribution map of 16 CPP-APUFs is shown in Fig. 7. The average of the uniformity of the 16

protective APUFs is 50.18%, which is close to the uniformity of 50% required by the ideal PUF.

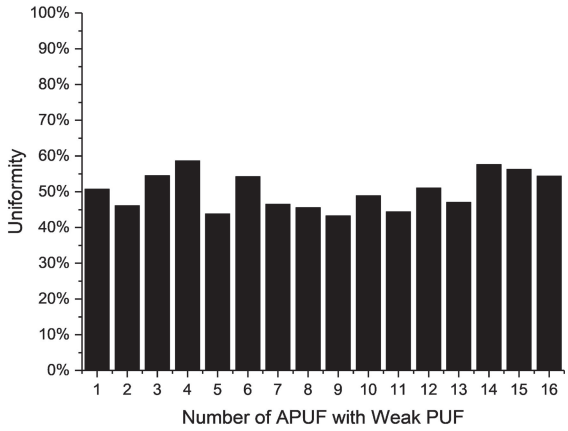

Fig. 7. Uniformity of 16 CPP-APUF

5. Summary and conclusions

In this paper, a new APUF with challenge pre-processing protection structure is proposed to resist the modeling attack. Based on the mathematical model, the design complexity analysis is given for the CPP-APUF. The APUF structure with challenge pre-processing is implemented on the FPGA platform. The experimental results show that the uniqueness and uniformity of the proposed CPP-APUF are close to that of the ideal PUF, and the stability is slightly lower than that of the original APUF structure. The machine learning algorithm is used for modeling attacks. The modeling accuracy of Linear Regression, Logic Regression and SVM is lower than 54.00%, and the BPNN modeling algorithm output response prediction accuracy is lower than 61.33%. None of them can accurately implement a predictive attack. It turns out that the CPP-APUF proposed in this paper has a better ability to resist machine learning attack than the protection structure of the existing literature.

Acknowledgments

We express thanks to Yang Jingjiang, Yang Jun and Liu Xinning for their useful and incisive comments. This work was supported by the National Key R&D Program of China (Grant No. 2018YFB2202102) and National Science and Technology Major Project (Grant No. 2017ZX01030101).

References

[1] R. S. Cannon, *et al.*: U.S. Patent App. 15/809,081 (2019).
[2] J. R. Booth, *et al.*: U.S. Patent 9,929,864 (2018).
[3] J. S. Kim, *et al.*: "The dram latency puf: Quickly evaluating physical unclonable functions by exploiting the latency-reliability tradeoff in modern commodity dram devices," 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA) (2018) 194 (DOI: 10.1109/HPCA.2018.00026).
[4] R. Maes, *et al.*: "Pufky: A fully functional puf-based cryptographic key generator," International Workshop on Cryptographic Hardware and Embedded Systems (2012) 302 (DOI: 10.1007/978-3-642-33027-8_18).
[5] T. Ziola, *et al.*: U.S. Patent 8,782,396 (2014).
[6] P. Tuyls and L. Batina: "Rfid-tags for anti-counterfeiting," Cryptographers' Track at the RSA Conference (2006) 115 (DOI: 10.1007/11605805_8).
[7] J. Guajardo, *et al.*: "Physical unclonable functions and public-key crypto for fpga ip protection," 2007 International Conference on Field Programmable Logic and Applications (2007) 189 (DOI: 10.1109/FPL.2007.4380646).
[8] P. Ravinkanth: "Physical one-way functions," Ph.D Thesis, MIT (2001).
[9] U. Rührmair and D. E. Holcomb: "Pufs at a glance," Proc. of the Conference on Design, Automation & Test in Europe. European Design and Automation Association (2014) 347 (DOI: 10.7873/DATE.2014.360).
[10] J. W. Lee, *et al.*: "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) 176 (DOI: 10.1109/VLSIC.2004.1346548).
[11] D. Lim, *et al.*: "Extracting secret keys from integrated circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **13** (2005) 1200 (DOI: 10.1109/TVLSI.2005.859470).
[12] U. Rührmair, *et al.*: "Puf modeling attacks on simulated and silicon data," IEEE Trans. Inf. Forensics Security **8** (2013) 1876 (DOI: 10.1109/TIFS.2013.2279798).
[13] U. Rührmair, *et al.*: "Modeling attacks on physical unclonable functions," Proc. of the 17th ACM conference on Computer and Communications Security (2010) 237 (DOI: 10.1145/1866307.1866335).
[14] A. Vijayakumar, *et al.*: "Machine learning resistant strong puf: Possible or a pipe dream?" 2016 IEEE international symposium on hardware oriented security and trust (HOST) (2016) 19 (DOI: 10.1109/HST.2016.7495550).
[15] J. Delvaux and I. Verbauwhede: "Side channel modeling attacks on 65 nm arbiter pufs exploiting cmos device noise," 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST) (2013) 137 (DOI: 10.1109/HST.2013.6581579).
[16] J. Delvaux and I. Verbauwhede: "Fault injection modeling attacks on 65 nm arbiter and ro sum pufs via environmental changes," IEEE Trans. Circuits Syst. I, Reg. Papers **61** (2014) 1701 (DOI: 10.1109/TCSI.2013.2290845).
[17] A. Mahmoud, *et al.*: "Combined modeling and side channel attacks on strong pufs," IACR Cryptology ePrint Archive 2013:632 (2013).
[18] J. Ye, *et al.*: "Vpuf: Voter based physical unclonable function with high reliability and modeling attack resistance," 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS) (2017) 74 (DOI: 10.1109/IOLTS.2017.8046200).
[19] X. Xu and W. Burleson: "Hybrid side-channel/machine-learning attacks on pufs: A new threat?," Proc. of the Conference on Design, Automation & Test in Europe. European Design and Automation Association (2014) 349 (DOI: 10.7873/DATE.2014.362).
[20] B. Gassend, *et al.*: "Identification and authentication of integrated circuits," Concurrency Computat.: Pract. Exper. **16** (2004) 1077 (DOI: 10.1002/cpe.805).
[21] Q. Ma, *et al.*: "A machine learning attack resistant multi-puf design on fpga," 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC) (2018) 97 (DOI: 10.1109/ASPDAC.2018.8297289).
[22] S. S. Zalivaka, *et al.*: "Low-cost fortification of arbiter puf against modeling attack," 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (2017) 1 (DOI: 10.1109/ISCAS.2017.8050671).
[23] S. S. Zalivaka, *et al.*: "Fpga implementation of modeling attack resistant arbiter puf with enhanced reliability," 2017 18th International Symposium on Quality Electronic Design (ISQED) (2017) 313 (DOI: 10.1109/ISQED.2017.7918334).
[24] S. Morozov, *et al.*: "An analysis of delay based puf implementations on fpga," International Symposium on Applied Reconfigurable Computing (2010) 382 (DOI: 10.1007/978-3-642-12133-3_37).

[25] H. Martin, *et al.*: "Total ionizing dose effects on a delay-based physical unclonable function implemented in fpgas," Electronics **7** (2018) 163 (DOI: 10.3390/electronics7090163).
[26] S. Hou, *et al.*: "A lightweight lfsr-based strong physical unclonable function design on fpga," IEEE Access **7** (2019) 64778 (DOI: 10.1109/ACCESS.2019.2917259).
[27] N. A. Hazari, *et al.*: "Fpga ip obfuscation using ring oscillator physical unclonable function," NAECON 2018-IEEE National Aerospace and Electronics Conference (2018) 105 (DOI: 10.1109/NAECON.2018.8556746).
[28] S. Srivathsa: Secure and energy efficient physical unclonable functions (2012).
[29] J. Kim, *et al.*: "A physical unclonable function with redox-based nanoionic resistive memory," IEEE Trans. Inf. Forensics Security **13** (2018) 437 (DOI: 10.1109/TIFS.2017.2756562).
[30] K. Yang, *et al.*: U.S. Patent 9,966,954 (2018).