DLA-PUF: deep learning attacks on hardware security primitives

Pugazhenthi, Anugayathiri, Karimian, Nima, Tehranipoor, Fatemeh

Anugayathiri Pugazhenthi, Nima Karimian, Fatemeh Tehranipoor, "DLA-PUF: deep learning attacks on hardware security primitives," Proc. SPIE 11009, Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019, 110090B (2 May 2019); doi: 10.1117/12.2519257

Event: SPIE Defense + Commercial Sensing, 2019, Baltimore, Maryland, United States

DLA-PUF: Deep Learning Attacks on Hardware Security Primitives

Anugayathiri Pugazhenthi[a], Nima Karimian[b], and Fatemeh Tehranipoor[a]

[a]San Francisco State University, 1600 Holloway Ave., San Francisco, USA
[b]San Jose State University, One Washington Sq., San Jose, USA

ABSTRACT

Physical Unclonable Functions (PUFs) act as functions encoded in hardware, which produce a unique output, being referred to as a response, for a specific input, being called a challenge. PUFs provide a varying level of security, and can, therefore, be used in different applications, depending on the number of their available input-output pairs, which are referred to as Challenge-Response Pairs (CRPs). For example, a PUF with only a single challenge-response pair can be used for identification, while a PUF with multiple CRPs can be used to provide multiple different session keys for authentication.[1,2] In the first case, the response needs to be secret, while, in the second one, responses can be also used without any secrecy, as long as the related CRPs are not used again. Generally, PUFs are vulnerable to modeling and machine learning attacks. In this paper we investigate and show the resiliency of DRAM-based PUFs against Machine Learning (Naive Bayes (NB), Logistic Regression (LR) and Support Vector machine (SVM)) and also Deep Learning (in particular convolutional neural network (CNN) attacks. We are the first to provide a detailed analysis of on-board DRAM startup values for the purpose of generating unique IDs and their vulnerabilities to attacks. We performed our experiments on the Digilent Atlys board (Xilinx Spartan 6 FPGA); using the on-board DRAM memories (MIRA P3R1GE3EGF G8E DDR2). Our results indicates that the 3 startup value-based DRAM PUFs (DRAM1, DRAM2, and DRAM3) are robust against machine learning attacks.

Keywords: DRAM PUF, Convolutional Neural Network (CNN), Machine Learning, Security Primitives

1. INTRODUCTION

Security and privacy of everyone's sensitive information in untrusted environment is one of the major issues recently. Securing side channel or any other hardware based attack through software patches are almost a temporary solution. Therefore, we need intrinsic hardware based secure solutions. One of the most researched such is Physical Unclonable Function(PUF). Since 2001, various types of PUFs have been proposed.[3] Among all, intrinsic PUF implementations, which do not require the addition of other hardware components, have been proposed as a lightweight and cost-efficient basis for security solutions. Memory components, such as Static Random Access Memories (SRAMs), Dynamic Random Access Memories (DRAMs) and Flash memories,[7,9,11] have proven to be particularly well-suited for the implementation of intrinsic hardware-based security primitives. DRAMs seem to have a number of additional advantages regarding their usage for the implementation of security primitives, such as their presence in most contemporary computer systems, their large storage size, and the easy way in which they can be accessed, even at run-time. In particular, their large storage size can guarantee both the existence of an adequate amount of entropy and that a part of them can be used, for a limited amount of

Further author information: (Send correspondence to Fatemeh Tehranipoor)
Fatemeh Tehranipoor: E-mail: tehranipoor@sfsu.edu

Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019
edited by Michael C. Dudzik, Jennifer C. Ricklin, Proc. of SPIE Vol. 11009, 110090B
© 2019 SPIE · CCC code: 0277-786X/19/$18 · doi: 10.1117/12.2519257

time, exclusively as dedicated security primitive, while the system is running. These advantages have led to an increased focus being placed on novel DRAM-based security primitives[12, 13] in recent publications regarding hardware-based security primitives.

However, recently these are PUFs also considered to be reverse engineered through machine learning techniques. We perform, to the best of our knowledge, the first security analysis of DRAM PUFs[15] against machine learning and deep learning attacks. To do that, we have tested 3 on-board DRAMs and collected the startup values[14] of them under various operating conditions (normal condition, temperature variation, voltage variation and aging). Our results show that DRAM PUFs are resistant to modeling attacks.

The rest of this paper is organized in the following way. Section 2 provides a review of related works and our specific contributions in this paper. We will describe DRAM based PUFs, experimental setup and data collection in Section 3. Section 4 shows our attack methodologies (machine learning and deep learning) in detail. We will demonstrate the results in Section 5. Finally, we will conclude in Section 6, following by future works.

2. RELATED WORK AND OUR CONTRIBUTIONS

There exists various attacks on strong and weak PUFs. In,[16] Ruhrmair et al. attacked arbiter PUFs, ring oscillator PUFs, XOR Arbiter PUFs based upon various machine learning techniques including logistic regression and evolution strategies. They have achieved a prediction rate of 99% to 99.9% through pseudo randomly generated CRP data and from an addictive delay model. In 2013, Mahmoud et al.[17] combined machine learning modeling attack (logistic regression) and side channel attack (power side channel) to improve the effectiveness of these kind of attack models. They applied their technique on XOR Arbiter PUFs, Arbiter PUFs and suggested a countermeasure to immune PUFs. Another work by Ruhrmair et al. in 2013[21] confirms the vulnerability of strong PUFs to modeling attacks with prediction rate of above 99%. In this work, given a set of challenge-response pairs (CRPs) of a Strong PUF, they constructed a computer algorithm which behaves indistinguishably from the original PUF on almost all CRPs. They applied various machine learning techniques to challenge-response sets from two sources Pseudo-random numeric simulations which used an addictive delay model with and without artificially injected errors and errors; and Silicon CRP data from FPGAs and ASICs. In 2012, Hospodar et al.[22] presented a work at WIFS and demonstrated the susceptibility of an actual 65nm CMOS arbiter PUF implementation to modeling attacks based on machine learning (ANN and SVM). With 1000 training CRPS they achieved the prediction accuracy of already ¿ 90%, and after 5,000 training CRPS the prediction is perfect up to the robustness of the PUF.[26] They also extended their work in 2014[24] by accelerating a little bit of the repeatability attacks by increasing the fraction of unstable Challenge Response Pairs. They were able to achieve a 2.4 times better attack speed than their previous model while still maintaining an excellent accuracy. In HOST 2013, Delvaux et al. demonstrated that 65nm CMOS arbiter PUFs can be modeled successfully, without even utilizing any ML algorithm. All strong PUF designs can be exploited through modeling using the side channel and information can be leaked through. They introduced a PUF repeatability model for predicting the responses by applying different challenges and also with the help of least mean square (LMS) attack method. Further they also discussed about a second attack method called Differential measurements method which is inferior to LMS method but that they can be used for other strong PUFs. The Modeling accuracy result that the authors got through LMS method by modeling the 65nm arbiter PUF exceeds 97%.[25]

In 2015, Sahoo et al.[27] proposed attacks on composite PUF using the challenge-response pairs (CRPs) from its field programmable gate array (FPGA) implementation, and attack on LSPUF is validated using the CRPs of both simulated and FPGA implemented lightweight secure PUF (LSPUF). In case of simulation, the modeling accuracy of virtual output bits is 99% with 25,000 CRPs, but in case of Artix-7 FPGA, accuracy lies in the range of 92.22% and 95.30% with same number of CRPs. Karakoyunlu et al. presented differential template attacks on PUFs. Their attack involves cloning the PUF device by recovering the fuzzy extractor (FE) input, they used the information leaked through a power side channel attack in the initial stages of error correction.[29] They reported two attacks: a simple power analysis (SPA) attack where they make use of conditional checks in a naive implementation to recover the PUF response by simply observing time shifts in the power consumption profile. In DATE 2014, Wayne Burleson et al. presented an overview and a survey on all the hybrid side channel and machine learning techniques that could possibly be used to attack Physical Unclonable Functions. They also provide both the challenges and opportunities that could be faced through attacking the PUFs. The authors compared different methodologies of implementing the side channel attacks on PUFs for better understanding. They categorized the side channel attacks into passive (power and timing side channel attack), Active attacks like for example fault injection, semi-invasive and hybrid attacks.[30] To the extent of our knowledge, all the above discussed researches in Section 2, or any other researches which has been done on PUF are not based on memory based PUF. Memory based Physical Unclonable Function implementations are considered as a more reliable option for the researchers who are working on building a more secure PUF. Since they are ubiquitous and do not require much additional circuitry, makes it easier to work with. Specifically, we have the following contributions in the paper:

(i). We have tested and collected data from 3 DRAMs based on their startup values.[19] The DRAM cells may flip under various operating conditions; therefore we gathered the data under normal condition, temperature variation,[20] voltage variation and aging[31] conditions.

(ii). We applied a couple of machine learning techniques (such as NB, LR, and SVM) to evaluate the vulnerability of DRAM based PUFs against these kinds of modeling attacks.

(iii). We also applied CNN to DRAM PUFs to see whether they are resilient against deep learning based attacks or not. Based on our results, DRAM PUFs are robust against such attack models.

3. PHYSICAL UNCLONABLE FUNCTIONS

The design of a strong silicon Physical Unclonable Function (PUF) with a rigorous security argument that is also lightweight and reliable has been the fundamental problem in PUF research, since its introduction in 2001.[3] Over the past years, there are various types of PUFs introduced,[8] each with its own application and security features. PUF implementations could provide multiple CRPs, forcing a distinction between PUFs with a single or very few CRPs, which are referred to as ”weak” PUFs, and PUFs with a large number of CRPs, which are called ”strong”. PUFs are great candidates to be used in internet of things (IoT) applications.[4–6, 18]

3.1 DRAM Startup Values based PUFs

Tehranipoor et al.[9, 10] demonstrated that DRAMs surprisingly have startup values - i.e. non-zero values when the DRAM is powered on. The reason for such behavior is understood by an examination of the typical DRAM cell structure. Figure 1 illustrates the architecture of a typical DRAM memory cell. One side of the capacitor is connected to a voltage of Vcc/2 , and the other end is initially left floating and then discharged to ground or charged to Vcc when written. The access transistor controls the charging of the capacitor. The difference between the two capacitor plates is read by the bit line to determine the read value. A voltage of Vcc generates the value one and a voltage of 0 gives a zero. At startup, however, before the memory cell is written, the bit line has a voltage of Vcc/2, creating unpredictable behavior in the DRAM read values. Thus, the non-zero values at startup. Tehranipoor et al. [teh17] were able to successfully use this behavior to construct a PUF. As mentioned earlier, we have tested 3 DRAMs based on their startup values under normal condition, high temperature, low temperature, low voltage, high voltage, and aging. We got 10 measurements under each condition from each

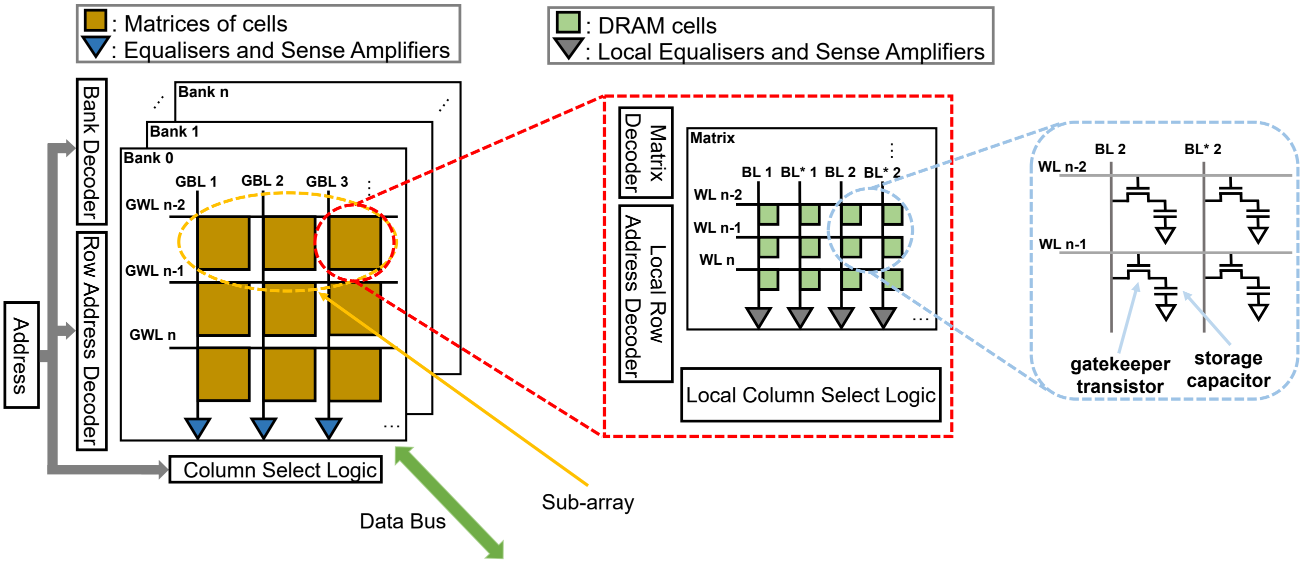

Figure 1. Internal DRAM organization. DRAM is organized in banks, each of which contains matrices of cells. Each cell consists of an access transistor and a storage capacitor.

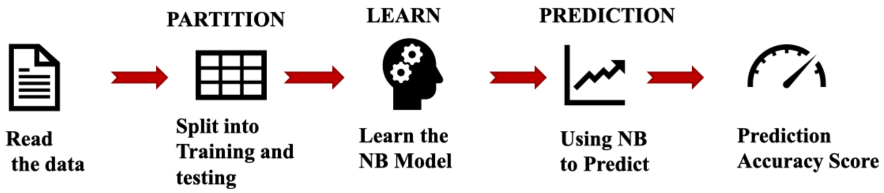

Figure 2. Steps involved in Naive Bayes (NB) model

DRAM memory. We then used these measurement for training and testing data in machine learning and deep learning techniques.

3.2 Experimental Setup and Data Collection

In this work, we attempted to do our experiment using the on-board MIRA P3R1GE3EGF G8E DDR2 on the Diligent Atlys board. (Xilinx Spartan 6 FPGA). The FPGA hardware was configured to transfer the memory data through a serial port at startup. A serial terminal was then used to record the bits and write them to a text file. We have collected the startup values of 3 on-board DRAMs under various experimental conditions (such as normal condition, high temperature, low temperature, high voltage, low voltage and aging).

4. ATTACK METHODOLOGIES

As diiscussed in the previous section, recent days researches show that some of the PUFs are vulnerable to attacks based on machine learning techniques. All the researches involve training and testing with randomly generated challenge response pairs. But our work is associated with both machine learning and deep learning techniques for trying to predict our DRAM power-up bits which will later be used for ID generation. Formerly in our research, We tried applying some of the supervised learning algorithms such as Naive Bayes, Logistic Regression etc., to our data, for predicting the raw startup values. The raw data implies the DRAM startup values that are taken under different conditions. We have m sets of n measurements of 1MB data for each condition from DRAM1, DRAM2 and DRAM3.

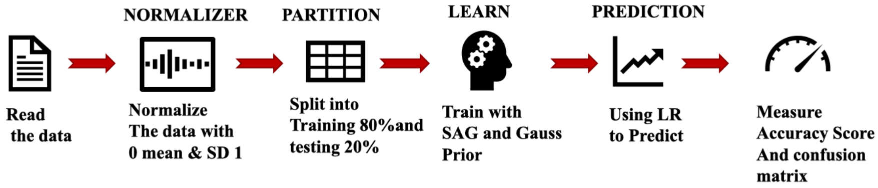
NORMALIZER
PARTITION
LEARN
PREDICTION
Read the data
Normalize The data with 0 mean & SD 1
Split into Training 80%and testing 20%
Train with SAG and Gauss Prior
Using LR to Predict
Measure Accuracy Score And confusion matrix

Figure 3. Steps involved in Logistic Regression (LR) model

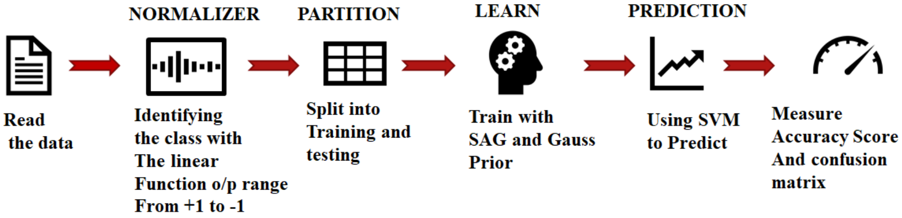
NORMALIZER
PARTITION
LEARN
PREDICTION
Read the data
Identifying the class with The linear Function o/p range From +1 to -1
Split into Training and testing
Train with SAG and Gauss Prior
Using SVM to Predict
Measure Accuracy Score And confusion matrix

Figure 4. Steps involved in Support Vector Machine (SVM) model

4.1 Employed Machine Learning Methods

4.1.1 Naive Bayes model(NB)

One of the classification algorithms which has been considered as computationally fast and reliable is the Naive Bayes classifier algorithm. Naive Bayes works on the basis of the Bayes theorem. Given that the probability of the event has already occurred, the theorem finds the probability of other events that will occur. It also calculates the probability of all the factors. The Naive indicates that this theorem assumes the attributes that need to be predicted are independent of the class, and also to each other, but this is a conditional Independence. Gaussian, Multinomial and Bernoulli are different types of naive Bayes classifier algorithm. Gaussian naive Bayes is used for Gaussian naive Bayes is a classifier algorithm that could be used on continuous data, it assumes that the input features or the values are normally distributed through a Gaussian process. While the discrete calculation is done by Multinomial naive Bayes, i.e. this algorithm involves predicting the number of times the outcome is observed over the number of trials. Lastly, Bernoulli is similar to Multinomial but this theorem takes only Boolean feature vectors. i.e. it could only tell you if the event is occurring or not. In our work, we used Gaussian naive Bayes for predicting the bits that are used for ID generation. We trained the model with n measurements of data and tried to predict the (n+1)th measurement of data and compared its accuracy with our original data. Figure 2 shows the basic steps involved in the NB prediction model.

4.1.2 Logistic Regression(LR)

A classification algorithm which has been commonly used to attack the PUFs Logistic Regression(LR). So we applied it to our data to evaluate the prediction rate. This model uses statistical analysis which gives an approximate calculation of an event that will occur for the previously described data. Instead of predicting the data through a straight line like linear regression, logistic regression will use a logarithmic function. This function finds coefficients using the testing data and then with that coefficients, the future results are measured through logic equation. Binary, multi-class and ordinal are the different types of logistic regression. Binary logistic regression is considered to be predicting whether the output is yes/no or pass/fail. Multi-class logistic regression implements one versus all classification algorithm, i.e. each of the input is mapped with each of the outputs. Through the previously tested data, the model can predict the input data. Likewise in our research, we used the model to predict the bits that we use for ID generation. In the end, we compare the results from the other machine learning techniques we used. Figure 3 shows the basic steps involved in LR learning Model.

4.1.3 Support Vector Machine (SVM)

One of the learning algorithms that could be used as both supervised and unsupervised learning is support vector network. So analyzing the data through all types regression, classification and clustering are possible through this. The algorithm works by constructing a decision boundary through the data for dividing into class and then measures the distance between the test data and plane for predicting the output. The number of features implies the number of dimensions of the hyperplane. SVM algorithm first discovers the support vectors points (SVP) that are close to hyperplane from both classes.[28] Then the marginal distance between the boundary and the SVP are computed. Maximizing the margin gives the optimal hyperplane. Support Vector Machine name implies that the optimal line is supported by the closest points which are represented through vectors. The perfect boundary between the possible outputs is found out through data transformation. Transformation of complex data is done through the kernel trick technique. G. Hospodar et al.,[22] implemented SVM and they achieved an accuracy of almost greater than 90 percent. Apart from that, this model have benefits of capturing more complex relationships between the data without much preprocessing on our own. In our work, we applied the SVM classification for predicting a new column of measurement. But the only problem we faced while training this model is the computational time was much longer than the other models. Figure 4 shows the basic steps involved in SVM learning Model.

4.2 Employed Deep Learning Technique

In this study, we propose a new modeling attack without modeling PUF using deep learning. The proposed method utilizes a raw DRAM initial value without artificial modeling. In our deep learning technique, an extension neural network with 5 layers, where the input layer receives a DRAM initial values which are the challenge of DRAM PUF and output layers returns responses which are the PUF ID. In other word, deep learning will be trained by having knowledge of the challenge-response pair of DRAM 1 in normal condition to predict the model. In testing, we are expecting to achieve a high success rate for other DRAMs. Based on the experimental results, we show that the chance of predicting the other DRAM responses is very low even though the model is trained very well in one DRAM challenge-response pairs.[23]

5. EXPERIMENTAL RESULTS

We used Anaconda environment an open-source of Python/R programming for machine learning applications and data science distribution. In this work, Python programming language is used. It has a collection of more than 1500 open source packages. We can develop AI and machine learning models rapidly directly from the platform. Anaconda can provide the tools, to collect data from files and databases. Within the anaconda, Jupyter notebook, the most popular web-based environment tool to use python in research and development. Scikit-learn is a machine learning library for python that could be used in the Jupyter notebook. Various classification, regression and clustering algorithms including SVM, K-means are featured within the library. We used this library for importing the machine learning models that we used. In our research, For loading the data in .mat format, SciPy library have been used. It also has tools like Matplotlib, pandas, etc. For concatenating the different measurements of data, the Pandas data frame is used in our research.

The result of our work is shown in Table 5. As can be seen, we have tested three DRAMs (DRAM1, DRAM2, and DRAM3) under a couple of machine learning based attacks. The results from gaussian naive bayes and bernoulli naive bayes indicate that naive bayes is unable to predict the startup values of DRAMs. It is very unlikely that someone could attack DRAM PUF using naive bayes techniques. Additionally, it is very hard to attack to the DRAM PUFs using logistic regression and support vector machine techniques either. As

	DRAM1				DRAM2				DRAM3			
	NC-HT	HT-LV	LT-AA	NC-HV	NC-AA	HT-HV	LT-LV	AA-HT	HV-AA	LV-HT	NC-LT	HT-HT
G-NB	10.1%	8.2%	17.8%	3.7%	18.2%	11.8%	15.8%	24.0%	21.6%	14.1%	38.2%	11.23%
B-NB	20.8%	13.1%	9.9%	23.3%	13.4%	13.2%	22.4%	12.8%	14.3%	24.7%	8.7%	25.5%
LR	13.2%	6.2%	38.0%	17.9%	35.2%	8.5%	41.5%	35.2%	30.26%	51.8%	16.7%	35.4%
SVM	15.3%	23.4%	7.2%	45.7%	16.2%	34.8%	40.7%	43.2%	19.6%	4.9%	31.3%	10.2%

Table 1. Accuracy result from G-NB (Gaussian Naive Bayes), B-NB (Bernoulli Naive Bayes), LR and SVM.

shown in Table 5, the accuracy of the results for these two methods are also very low. Note that the following abbreviation are: NC ”normal condition”; HT ”high temperature”; LT ”low temperature”; HV ”high voltage”; LV ”low voltage”; and AA ”Aging”.

6. CONCLUSION AND FUTURE WORKS

In this work, we investigate the vulnerability of DRAM PUFs against machine learning and deep learning based attacks. This is the first work showing that DRAM PUFs (in particular startup value based) are resistance to ML and DL attacks. Based on our result, it is very unlikely that an attacker could predict the keys generated from DRAM PUFs and also raw data using machine learning and deep learning techniques. In future, we intend to apply these machine learning techniques on the data collected from other types of DRAMs. we also would like to apply ML techniques on other types of PUFs in general and evaluate their vulnerabilities to these attacks.

REFERENCES

[1] Yan, Wei, Fatemeh Tehranipoor, and John A. Chandy. ”A novel way to authenticate untrusted integrated circuits.” In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, pp. 132-138. IEEE Press, 2015.

[2] Yan, Wei, Fatemeh Tehranipoor, and John A. Chandy. ”Puf-based fuzzy authentication without error correcting codes.” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 36, no. 9 (2017): 1445-1457.

[3] Gassend, Blaise, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. ”Silicon physical random functions.” In Proceedings of the 9th ACM conference on Computer and communications security, pp. 148-160. ACM, 2002.

[4] Hassan, Qusay F., and Sajjad A. Madani. Internet of things: Challenges, advances, and applications. Chapman and Hall/CRC, 2017.

[5] Chandy, John A., Jim Fahrny, Asad Haque, Paul A. Wortman, Nima Karimian, and Fatemeh Tehranipoor. ”Exploring Methods of Authentication for the Internet of Things.” In Internet of Things, pp. 71-90. Chapman and Hall/CRC, 2017.

[6] Karimian, Nima, Paul A. Wortman, and Fatemeh Tehranipoor. ”Evolving authentication design considerations for the internet of biometric things (IoBT).” In Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, p. 10. ACM, 2016.

[7] Holcomb, Daniel E., Wayne P. Burleson, and Kevin Fu. ”Power-up SRAM state as an identifying fingerprint and source of true random numbers.” IEEE Transactions on Computers 58, no. 9 (2009): 1198-1210.

[8] Yan, Wei, Chenglu Jin, Fatemeh Tehranipoor, and John A. Chandy. ”Phase calibrated ring oscillator PUF design and implementation on FPGAs.” In 2017 27th International Conference on Field Programmable Logic and Applications (FPL), pp. 1-8. IEEE, 2017.

[9] Tehranipoor, F.; Karimian, N.; Yan, W.; Chandy, J.A. DRAM-Based Intrinsic Physically Unclonable Functions for System-Level Security and Authentication. IEEE Trans. Very Large Scale Integr. Syst. 2017, 25, 1085-1097.

[10] Tehranipoor, Fatemeh, Nima Karimian, Kan Xiao, and John Chandy. "DRAM based intrinsic physical unclonable functions for system level security." In Proceedings of the 25th edition on Great Lakes Symposium on VLSI, pp. 15-20. ACM, 2015.

[11] Prabhu, P.; Akel, A.; Grupp, L.M.; Wing-Kei, S.Y.; Suh, G.E.; Kan, E.; Swanson, S. Extracting Device Fingerprints from Flash Memory by Exploiting Physical Variations. In International Conference on Trust and Trustworthy Computing; Springer: Berlin, Germany, 2011; pp. 188-201.

[12] Tehranipoor, Fatemeh, Wei Yan, and John A. Chandy. "Robust hardware true random number generators using DRAM remanence effects." In 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 79-84. IEEE, 2016.

[13] Anagnostopoulos, Nikolaos, Stefan Katzenbeisser, John Chandy, and Fatemeh Tehranipoor. "An Overview of DRAM-Based Security Primitives." Cryptography 2, no. 2 (2018): 7.

[14] Anagnostopoulos, Nikolaos Athanasios, Andre Schaller, Yufan Fan, Wenjie Xiong, Fatemeh Tehranipoor, Tolga Arul, Sebastian Gabmeyer, Jakub Szefer, John A. Chandy, and Stefan Katzenbeisser. "Insights into the Potential Usage of the Initial Values of DRAM Arrays of Commercial Off-The-Shelf Devices for Security Applications." Proceedings of the 26th Crypto-Day, Nuremberg, Germany (2017): 1-2.

[15] Wortman, Paul, Wei Yan, John Chandy, and Fatemeh Tehranipoor. "P2M-based security model: security enhancement using combined PUF and PRNG models for authenticating consumer electronic devices." IET Computers & Digital Techniques 12, no. 6 (2018): 289-296.

[16] Ruhrmair, Ulrich, Frank Sehnke, Jan Solter, Gideon Dror, Srinivas Devadas, and Jurgen Schmidhuber. "Modeling attacks on physical unclonable functions." In Proceedings of the 17th ACM conference on Computer and communications security, pp. 237-249. ACM, 2010.

[17] Mahmoud, Ahmed, Ulrich Ruhrmair, Mehrdad Majzoobi, and Farinaz Koushanfar. "Combined Modeling and Side Channel Attacks on Strong PUFs." IACR Cryptology ePrint Archive 2013 (2013): 632.

[18] Tehranipoor, Fatemeh, Nima Karimian, Paul A. Wortman, and John A. Chandy. "Low-cost authentication paradigm for consumer electronics within the internet of wearable fitness tracking applications." In 2018 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6. IEEE, 2018.

[19] Eckert, Charles, Fatemeh Tehranipoor, and John A. Chandy. "DRNG: DRAM-based random number generation using its startup value behavior." In 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1260-1263. IEEE, 2017.

[20] Anagnostopoulos, Nikolaos Athanasios, Tolga Arul, Yufan Fan, Christian Hatzfeld, Fatemeh Tehranipoor, and Stefan Katzenbeisser. "Addressing the Effects of Temperature Variations on Intrinsic Memory-Based Physical Unclonable Functions." crypto day matters 28 (2018).

[21] Ruhrmair, Ulrich, Jan Solter, Frank Sehnke, Xiaolin Xu, Ahmed Mahmoud, Vera Stoyanova, Gideon Dror, Jurgen Schmidhuber, Wayne Burleson, and Srinivas Devadas. "PUF modeling attacks on simulated and silicon data." IEEE Transactions on Information Forensics and Security 8, no. 11 (2013): 1876-1891.

[22] Hospodar, Gabriel, Roel Maes, and Ingrid Verbauwhede. "Machine learning attacks on 65nm Arbiter PUFs: Accurate modeling poses strict bounds on usability." In 2012 IEEE international workshop on Information forensics and security (WIFS), pp. 37-42. IEEE, 2012.

[23] Karimian, Nima, Fatemeh Tehranipoor, Nikolaos Anagnostopoulos, and Wei Yan. "DRAMNet: Authentication based on Physical Unique Features of DRAM Using Deep Convolutional Neural Networks." arXiv preprint arXiv:1902.09094 (2019).

[24] Delvaux, Jeroen, and Ingrid Verbauwhede. "Fault injection modeling attacks on 65 nm arbiter and RO sum PUFs via environmental changes." IEEE Transactions on Circuits and Systems I: Regular Papers 61, no. 6 (2014): 1701-1713.

[25] Delvaux, Jeroen, and Ingrid Verbauwhede. "Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise." In 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 137-142. IEEE, 2013.

[26] Anagnostopoulos, Nikolaos Athanasios, Tolga Arul, Yufan Fan, Christian Hatzfeld, Jan Lotichius, Ratika Sharma, Felipe Fernandes, Fatemeh Tehranipoor, and Stefan Katzenbeisser. "Securing IoT Devices Using Robust DRAM PUFs." In 2018 Global Information Infrastructure and Networking Symposium (GIIS), pp. 1-5. IEEE, 2018.

[27] Sahoo, Durga Prasad, Phuong Ha Nguyen, Debdeep Mukhopadhyay, and Rajat Subhra Chakraborty. "A case of lightweight PUF constructions: Cryptanalysis and machine learning attacks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 34, no. 8 (2015): 1334-1343.

[28] Karimian, Nima, Fatemeh Tehranipoor, Md Tauhidur Rahman, Shane Kelly, and Domenic Forte. "Genetic algorithm for hardware Trojan detection with ring oscillator network (RON)." In 2015 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-6. IEEE, 2015.

[29] Karakoyunlu, Deniz, and Berk Sunar. "Differential template attacks on PUF enabled cryptographic devices." In 2010 IEEE International Workshop on Information Forensics and Security, pp. 1-6. IEEE, 2010.

[30] Xu, Xiaolin, and Wayne Burleson. "Hybrid side-channel/machine-learning attacks on PUFs: A new threat?." In Proceedings of the conference on Design, Automation & Test in Europe, p. 349. European Design and Automation Association, 2014.

[31] Tehranipoor, Fatemeh, Nima Karimian, Wei Yan, and John A. Chandy. "Investigation of DRAM PUFs reliability under device accelerated aging effects." In 2017 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-4. IEEE, 2017.