University of California
Santa Barbara

On Design and Machine Learning Resiliency of Memristor- and eFlash-Memory-Based Strong Physical Unclonable Functions

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Shabnam Larimian

Committee in charge:

Professor Dmitri B. Strukov, Chair
Professor Tim Sherwood
Professor Li-C. Wang
Professor Çetin Kaya Koç

June 2021

The Dissertation of Shabnam Larimian is approved.

Professor Tim Sherwood

Professor Li-C. Wang

Professor Çetin Kaya Koç

Professor Dmitri B. Strukov, Committee Chair

April 2021

On Design and Machine Learning Resiliency of Memristor- and eFlash-Memory-Based Strong Physical Unclonable Functions

by

Shabnam Larimian

To my dear parents, Maryam and Javad, and my loving sister, Marjan, for their endless love, support, and encouragement.

To my beloved husband, Mahdi, who has been a constant source of love, help, support, and encouragement in my life and studies.

This journey would not be possible without them.

Acknowledgements

First, I would like to express my sincere gratitude to my advisor Professor Dmitri B. Strukov for his continuous support, enthusiasm, immense knowledge, and patience. This dissertation and all the research work I have conducted during my PhD journey would not have been possible without him. I am thankful to have such the great mentor and advisor during my graduate life.

I would like to express my gratitude to the members of my examination committee, Professor Tim Sherwood, Professor Li-C. Wang, and Professor Çetin Kaya Koç for their valuable and constructive comments and suggestions.

I would like to thank my colleagues, Dr. Mohammad Reza Mahmoodi, Dr. Farnood Merikh-Bayat, Dr. Michael Klachko, Dr. Hussein Nili, Dr. Adrien Vincent, and Dr. Ping-Lin Yang for sharing their knowledge and work. The conversations with you were always helpful to enrich my research. I should thank my colleague, Dr. Itir Akgun, and my friend, Sogol Khanof, for being great and caring friends during my PhD journey.

I would like to express my deepest gratitude to my dearest family: my parents, my husband, and my younger sister. They are my strongest support and always there for me giving me the strength to reach for the stars and chase my dreams.

At the end, I would like to thank the Department of Electrical and Computer Engineering at the University of California, Santa Barbara, for all of the teaching assistant opportunities, administrative work, and IT infrastructure. I would also like to thank computing resource support from the Center for Scientific Computing from the California NanoSystems Institute (CNSI) at the University of California, Santa Barbara.

Curriculum Vitæ

Shabnam Larimian

Education

2021	Ph.D. in Computer Engineering, University of California, Santa Barbara
2015	M.Sc. in Computer Engineering, University of California, Santa Barbara
2014	B.Sc. in Electrical Engineering, University of Tehran, Iran

Employment

2015 - 2021	Teaching Assistant, Department of Electrical and Computer Engineering, University of California, Santa Barbara
2019	Software Engineer Intern, Google
2018	Software Engineer Intern, Google
2017	Data Analyst Graduate Intern, Cadence Design Systems
2016	R&D Software Engineering Intern, Cadence Design Systems

Skills

- Python, Java, C/C++, SQLite, HTML, JavaScript, PHP, MySQL, Tcl, Perl, Verilog HDL, System Verilog
- MATLAB, Android, HSPICE, PSPICE, Cadence Virtuoso, XcitePI, ModelSim, NI Multisim, Xilinx Vivado, CodeVision AVR, Altera Quartus, Proteus, Arduino, Analog Discovery, Raphael, Filter Solution

Awards

2021	Outstanding teaching assistant award at University of California, Santa Barbara
2020	ECE dissertation fellowship award at University of California, Santa Barbara
2016 - 2020	Outstanding teaching assistant award of ECE department at University of California, Santa Barbara
2019	Grace Hopper Celebration scholarship award
2017	Nominee of UCSB outstanding teaching assistant award at University of California, Santa Barbara
2012	Certificate of appreciation for teaching in applied electronics workshop at University of Tehran, Iran

2008	Ranked 71st among 300,000 participants in the national university entrance exam in physics and mathematics, Iran
2007	Accepted in first step of mathematical Olympiad, Iran
2007	Accepted in first step of chemistry Olympiad, Iran
2007	First place of physics articles in student competition at University of Tehran, Iran

Publications

- **S. Larimian**, M. R. Mahmoodi, and D. B. Strukov, *Improving machine learning attack resiliency via conductance balancing in memristive strong PUFs*, in: SRC Techcon, Austin, TX, Sept. 2020, pp. 1-4.
- **S. Larimian**, M. R. Mahmoodi, and D. B. Strukov, *Lightweight integrated design of PUF and TRNG security primitives based on eFlash memory in 55-nm CMOS*, IEEE Transactions on Electron Devices, vol. 67 (4), pp. 1586-1592, 2020.
- M. R. Mahmoodi, Z. Fahimi, **S. Larimian**, H. Nili, H. Kim, and D. B. Strukov, *A strong physically unclonable function with* $> 2^{80}$ *CRPs and* $< 1.4\%$ *BER using passive ReRAM technology*, IEEE Solid-State Circuits Letters, vol. 3, pp. 182-185, 2020.
- M. R. Mahmoodi, H. Nili, Z. Fahimi, **S. Larimian**, H. Kim, D. B. Strukov, *Ultra-low power physical unclonable function with nonlinear fixed-resistance crossbar circuits*, in: Proc. IEDM'19, San Francisco, CA, Dec. 2019, pp. 30.1.1-30.1.4.
- M. R. Mahmoodi, H. Nili, **S. Larimian**, X. Guo, D. B. Strukov, *ChipSecure: A reconfigurable analog eFlash-based PUF with machine learning attack resiliency in 55nm CMOS*, in: Design Automation Conference (DAC), Las Vegas, NV, Jun. 2019, pp. 1-6.

Abstract

On Design and Machine Learning Resiliency of Memristor- and eFlash-Memory-Based Strong Physical Unclonable Functions

by

Shabnam Larimian

The emergence of the Internet of Things (IoT) has enabled an unprecedented expansion of interconnected networks and devices over which a huge amount of personal and/or sensitive data is carried. As a result, privacy and security issues are among the most significant challenges in designing IoT devices. These challenges can hardly be addressed using conventional cryptographic approaches because they rely on storing secret keys in memories, which not only are vulnerable to physical and side-channel attacks but also consume huge area and vast amounts of power.

Hardware-based security approaches such as physical unclonable functions (PUFs) have attracted considerable attention as replacements for conventional methods. PUFs are well suited to a wide spectrum of security applications including key generation and authentication because they generate secure keys on the fly (rather than explicitly storing any security-critical information). This is achieved by utilizing electronic devices that entail inherent sources of randomness, which in turn help create unique keys for different physical entities.

Recently, a variety of emerging nano-scale non-volatile memories are being explored for use in the design of PUFs including memristors and embedded flash (eFlash) memories. The highly non-linear current-voltage characteristics and the inherent process variations of these memory devices make them promising candidates for designing PUFs. Additionally, the ultra-low power consumption and low computation time of these de-

vices enable their use in applications with stringent requirements on energy efficiency and throughput.

This dissertation presents memristor- and eFash-memory-based PUF designs that show promising security characteristics such as near-to-ideal uniformity, diffuseness, robustness, and reliability. The robustness is verified by demonstrating the high output randomness with the test suits of the National Institute of Standards and Technology and by studying various machine learning attacks.

The specific contributions of this dissertation is that investigates several unexplored areas in crossbar-memory-design PUFs, e.g., finding optimal design for maximizing robustness characteristics, studying the impact of the capacity of machine learning models on robustness, and the impact of environmental change and thermal noise on reliability.

Contents

Chapter 1

Introduction

With the fast and continuing development of Internet of Things, smart devices and interconnected networks have become ubiquitous for everyday tasks. In many of those tasks, significant volume of personal and/or sensitive information is carried which raises security and privacy issues. The conventional cryptographic approaches can hardly address those challenges because they rely on storing secret keys in memories and assume that the keys are unknown to adversary. However, it is difficult to uphold this assumption because the memories are vulnerable to physical and side-channel attacks. Moreover, to store the secret keys, the memories consume huge area and power. As a result, securing a resource-constraint, integrated system is an ongoing challenging problem [1, 2, 3, 4].

As a replacement for conventional cryptographic approaches, hardware-based security approaches such as physical unclonable functions (PUFs) have attracted substantial attentions. PUFs utilize the inherent randomness in electronic devices to generate keys on the fly rather than storing them in non-volatile memories which makes PUFs well suited to variety of security applications such as key generation and authentication.

The conventional PUFs utilize uncontrollable process variation in conventional Complementary-Metal-Oxide-Semiconductor (CMOS) fabrication technology. Process variations in purely

CMOS analog circuits often limit computation accuracy and result in large performance overheads due to over-designing and calibration techniques. As a result of scaling down to nano region, the next generation of PUFs will be implemented using nano-electronic devices ([5]) such as memristors ([6]) and flash memories ([7]) whose highly non-linear current-voltage characteristics and the inherent process variations make them promising candidates for designing PUFs. Moreover, the mentioned nano-electronic devices are CMOS compatible and have ultra-low power consumption and low computation time [8, 9].

This thesis contains several contributions to the field of PUFs. We have proposed two techniques to boost memristor-based strong PUFs robustness against machine learning. Furthermore, we present a lightweight, integrated flash-memory-based design of PUF and true random number generator (TRNG) on a shared silicon which can be effectively used in mutual authentication applications. Below are the summary of the chapters.

Chapter 2. In this chapter, after defining PUF, we discuss its main types, applications, and cryptographic metrics. Then, we briefly discuss how TRNGs are associated with PUFs. Finally, we summarize the prior work on PUFs.

Chapter 3. Previous works have shown excellent prospects for implementing strong PUFs with memristive crossbar circuits. In this chapter, we propose two techniques for boosting the robustness of such PUFs to machine learning attacks. The general idea behind both proposals is to maximize the contribution of each crosspoint device to the PUF output to make the response less predictable. Specifically, we present results for choosing an optimal ratio of selected rows and columns and investigate in detail the improvements in robustness due to the balancing of device conductances in the crossbar array. The effectiveness of the proposed algorithm for conductance balancing is confirmed by modeling the response of two-sided PUF based on a 20×20 crossbar memristive circuit with a multilayer perceptron network. Then, we explore some open questions which require in-depth analysis. Specifically, we quantify the effect of device nonlinearity and device analog-tunability. We show that nonlinear, analog memristive PUFs outperform the PUFs that have either linear or digital devices. Finally, we explore the effect of stuck-at fault devices (non-ideal yield) on PUFs uniformity. Indeed, by modeling this hardware imperfection, we show that the proposed algorithm results in a more-robust PUF.

Chapter 4. In chis chapter, we present a lightweight, integrated design of flash-memory-based PUF and TRNG on a shared silicon. Specifically, the randomness in nonlinear I-V characteristics and temporal current fluctuations of embedded flash memories are exploited to generate static entropy (for PUF functionality) and dynamic entropy (for TRNG functionality). A time-multiplexed architecture is designed to enhance the security and expand the challenge-response pair space to 10^{211}. Experimental results demonstrate 50.3% average uniformity, 49.99% average diffuseness, and native $\leq 5\%$ bit error rate. Moreover, accelerated aging measurements is done for the designed PUF. The measurements indicate stable PUF response after 900 minutes of baking at 85°C. The analysis of the measured data also shows strong resilience against machine learning attacks and possibility for extremely energy efficient, 0.56 pJ/b operation.

Chapter 2

Preliminaries and Prior Work

Physically unclonable functions and true random number generators are two main cryptography primitives. The former is used to implement secure secret key generation and low-cost device authentication whereas the latter generate random numbers from a physical process.

This chapter presents a concise background on physically unclonable functions. After defining physically unclonable function, its main types, applications, and cryptographic metrics are discussed. Then, it is discussed how true random number generators are associated with physically unclonable functions. Finally, the prior work on physically unclonable functions are summarized.

2.1 Physically Unclonable Functions (PUF)

Nowadays, cryptographic keys are the foundation of secure cryptographic protocols in electronic systems and are typically stored in non-volatile memories (NVMs). Because the key is assigned by an outside source and stored in a NVM, it is vulnerable to be copied and it is not trivial to maintain its security without dedicated protection.

Physical unclonable function, or PUF, is a cost-effective alternative approach that does not have the mentioned issues. Indeed, PUF is a class of hardware security primitives that generates cryptographic keys by exploiting the inherent random variations introduced during manufacturing process. Because the generated key is internal and is not assigned by an outside source, it is infeasible to clone it and create an identical physical copy. This form of randomness is inexpensive to access and does not affect the original functionality of the devices [10, 11]. Additionally, by leveraging intrinsic or extrinsic randomness sources, PUF works as a one-way function that maps an input (challenge) to an output (response). The set of generated challenge-response pairs (CRPs) are then used in different applications. Below, PUF types (based on number of CRPs), applications, and metrics are discussed.

2.1.1 Types

PUFs are typically classified as weak and strong based on the size of CRPs. This usually corresponds to how the number of CRPs increases when device size increases (scaling rate). Weak PUF has a small access-restricted CRPs (due to linear or polynomial scaling rate) which means that the full set of CRPs can be read if an attacker holds physical possession of the device. While it is not possible to reproduce the physical PUF itself, the attacker can deduce the mapping function with the knowledge of observed CRPs. Strong PUF, on the other hand, has huge number of CRPs (due to exponential scaling rate) that prevents a full read-out of CRPs even if the attacker gains physical access for a considerable time. This makes PUF mapping function resilient to learn or reproduce [8, 12, 2].

2.1.2 Applications

The most fundamental security applications of Internet of Things are key generation and authentication. Key generation (and key storage) requires a random source to generate unique keys (and a protected memory to hide them from an attacker). Comparatively, authentication requires validity of the identifying information. Depending on the application, either one-way or mutual authentication should be implemented. A common authentication approach relies on the challenge-response protocols where the verifier provides a challenge and the prover provides the response to be authenticated. In this section, some of the PUF security requirements for each of the mentioned applications are reviewed.

Key Generation

Secure keys are typically generated by seeding pre-stored keys to pseudorandom number generators [13]. However, by using PUFs, the unique keys can be generated on the fly which eliminates the need for key storage. Because this application typically needs a limited capacity and CRP space, weak PUF is a promising candidate for it. Additionally, in this application, the generated keys should be reliable. Indeed, the PUF response should be reproducible across process, voltage, and temperature (PVT) variations. Because no error is tolerable in key generation applications, error correction codes and algorithms are often applied to improve the PUF reliability. Furthermore, the weak PUFs being considered in key generation application should have high throughput and low power and area overhead [8].

Authentication

For authentication applications, it is crucial for PUF to be unpredictable. Indeed, the PUF should be both physically and mathematically unclonable. For this purpose, the CRP space should be large enough to avoid man-in-the-middle attack by changing the challenge after each run. Additionally, the PUF circuit should be complex enough that it cannot be modeled or deduced even if an attacker has physical access to it and observe a certain number of CRPs. As a result of the mentioned requirements, strong PUFs are the best candidates for the authentication applications [8].

2.1.3 Quality Metrics

Assessing if a physical PUF behaves as a theoretically ideal PUF is not trivial. To assist the evaluation process, several metrics are suggested over the years [14, 15, 16, 1, 17, 18]. Here, the main and common metrics are discussed.

Fractional Hamming Weight or Uniformity

The Hamming weight (HW) of a vector is the number of non-zero elements in the vector. For a binary vector, HW is equal to the number of '1's in the string. Because the length of vector may vary, HW is normalized by dividing its value by the vector length. This is called fractional Hamming weight (FHW) or uniformity (UF) and is calculated as

$$FHW(R) = UF(R) = \frac{1}{|R|} HW(R) = \frac{1}{|R|} \sum_{i=0}^{|R|-1} (R_i)$$

where $|R|$ is the length of response vector. This metric is used to assess the PUF randomness by measuring the balance of its response vector and is usually reported in percentage. The ideal value for UF is 50% indicating a perfect balance between possible responses (same number of '0's and '1's in a response vector). UF value that is either much below or much above 50% can indicate that PUF responses are biased, and thus PUF might have non-ideal behavior.

Fractional Hamming Distance or Diffuseness

The Hamming distance (HD) between two equal-length vectors is defined as the number of positions at which the corresponding elements are different. Because the length of vector may vary, HD is normalized by dividing its value by the vector length. This is called fractional Hamming distance (FHD) or diffuseness (DF) and is calculated as

$$FHD(R_i, R_j) = DF(R_i, R_j) = \frac{1}{|R|} HD(R_i, R_j) = \frac{1}{|R|} \sum_{i=0}^{|R|-1} (R_i - R_j)$$

where R_i and R_j are two equal-length vectors of the same PUF under different challenges. This metric is used to assess PUF randomness by measuring dissimilarity among response vectors of the same PUF under different challenges and is usually reported in percentage. The ideal value of diffuseness is 50% which shows the complete dissimilarity between PUF response when different challenges are applied. This metric that represents self-dissimilarity has little meaning in certain cases such as a weak SRAM PUF which has a single CRP.

Uniqueness

Uniqueness (UQ) is another metric that assess PUF randomness. It measures the dissimilarity of responses of different PUFs to the same challenge. Uniqueness is defined as

$$UQ(R, \bar{R}) = \frac{1}{|R|} HD(R, \bar{R}) = \frac{1}{|R|} \sum_{i=0}^{|R|-1} (R - \bar{R})$$

where R and $\bar{R}$ are two equal-length response vectors of different PUF under the same challenge and HD is the Hamming distance of the two vectors. This metric is used to assess how different PUFs respond uniquely to the same challenge. UQ is usually reported in percentage and its ideal value is 50%. Any large deviations from this ideal value demonstrate correlation between different PUF instances.

Bit Error Rate

Bit error rate (BER) is used to measure PUF reliability and reproducibility. BER shows the difference between the same PUF response under the same challenge but different situations caused by variation in temperature, variation in voltage, or noise. BER is usually reported in percentage and its ideal value is 0% meaning that PUF always produce the same response for a given challenge. When BER is not close to 0%, excessive error correction is needed to reduce it, which is costly in terms of computation time, energy consumption, and memory usage.

Entropy

In information theory, entropy, a basic quantity associated to any random variable, is interpreted as the average level of information or uncertainty inherent in the variable's possible outcomes. Based on Shannon equation, entropy (H(X)) is defined as

$$H(X) = -\sum_{i=1}^{n} p_i log_2(p_i)$$

where $p_i = P(X = x_i)$ is a probability of a random variable X that can take on values x_1, x_2,. . . , x_n, $p_i \geq 0$, and $\sum_{i=1}^{n} p_i = 1$. The entropy is expressed in the number of bits that carry information. For example, in a truly random binary process ($p_i = 0.5$), the entropy is calculated as

$$H(X) = -(0.5log_2(0.5) + 0.5log_2(0.5)) = 1$$

The maximum entropy (1 in a binary process) indicates the maximum uncertainty of next response bit that is not observed yet [9, 19].

Correlation

Correlation is a metric that shows if the PUF output has any bias towards any of its inputs. Although many of the above metrics indirectly represents correlation, it can be directly calculated by the percentage of '0's or '1's in the output when one specific input changes and the other inputs stay the same. The ideal correlation value if 50% which means the output is 0 or 1 with equal probability regardless of the specific bit in challenge.

National Institute of Standards and Technology (NIST) Test Suite

Various statistical tests can be applied to a sequence to evaluate its randomness by assessing the presence or absence of a pattern which if detected would indicate that the sequence is not random. The National Institute of Standards and Technology (NIST) test suite ([19]) is one of the online public statistical packages consisting of sixteen tests that is developed to test the randomness of binary sequences by calculating P-value. P-value is the probability that a sequence less random than the tested sequence can be generated. The P-value of 1 indicates that the sequence is perfectly random while the P-value of 0 means that the sequence is completely non-random. A significance level can be chosen as a threshold for P-value so that any value higher/lower than that indicates that the sequence is random/non-random. Typically, the significance level is chosen in the range of $[0.001, 0.01]$. The significance level that is used in NIST test suite in this thesis is 0.01. This means that if the P-value of a sequence is greater/less than 0.01, the sequence passes/fails that test.

Predictability (Machine Learning Test)

An ideal PUF should be unpredictable to attackers. To perform predictability analyses, machine learning (ML) models are considered as they are currently the most effective attack form for strong PUFs [13]. The machine learning models would be trained on a subset of CRPs and then would be tested on a mutually exclusive subset of CRPs. The input and output of the machine learning models would be connected to the challenge and response of PUF, respectively. For a single-output PUF, different binary classifiers such as logistic regression (LR), support vector machine (SVM), and multilayer perceptron (MLP) can be used. In an ideal PUF, the accuracy of the ML test should be 50% which means that the attacker cannot model the PUF by accessing a subset of CRPs.

The LR algorithm uses logistic (sigmoid) function to find the relationship between input and output. The sigmoid function is an S-shaped curve that can take any real-valued number and map it to a value between 0 and 1. LR is mostly used for a linear-separable data. Therefore, it might not be the best ML algorithm for strong PUFs where the output is a nonlinear function of the inputs.

The SVM algorithm creates a hyperplane or line (decision boundary) which separates data into classes. It uses the kernel trick to find the best line separator (decision boundary that has same distance from the boundary point of both classes). In other words, SVM tries to reduce the error by finding the best margin (distance between the line and the support vectors) that separates the classes. Although SVM is a more powerful way of learning complex nonlinear functions (comparing to LR), it runs very slow on huge amount of data. As a result, for a PUF with huge exponential CRP space, SVM is not the most efficient ML algorithm.

The MLP algorithm is a class of feedforward neural network consists of at least three layers: an input layer, a hidden layer, and an output layer. Except from the input nodes, each node is a neuron that uses a nonlinear activation function (e.g. sigmoid). MLP uses backpropagation technique for training. Because MLP has multiple layers and utilizes nonlinear activation function, it can be used to distinguish data that is not linearly separable. That is why, MLP is a good ML algorithm candidate to asses the predictability of strong PUFs.

Based on the PUF design, the attacker can choose any ML algorithm to model PUF behaviour. An ideal PUF should be resilient to any type of ML algorithms. In other words, the trained ML model should ideally has 50% accuracy on the unseen data meaning that the attacker cannot model the relationship between the input and the output of PUF using ML algorithms.

2.2 True Random Number Generator (TRNG)

It is important to mention that the process variation explained in Section 2.1 can be a source of randomness for true random number generators (TRNG) as well. Indeed, process variation leads to two types of random behavior namely static and dynamic. While the static response behavior is used in PUF applications, the dynamic response behavior (which is mostly due to the very small differences in operating conditions and circuit noise) is used in TRNG applications. These two randomness sources can be combined in a circuit and provides both PUF and TRNG functionalities for cryptography applications. The unified design will be very efficient in terms of area, power, and energy [16].

In order to evaluate the statistical properties of TRNG output, different metrics need to be calculated and measured. In this section, three of well-known quality metrics are reviewed and defined which ensure the presence of cryptography quality randomness.

The first metric to evaluate the TRNG output sequence is NIST test suite that is explained in detail in Section 2.1.3. NIST test suite consists of sixteen tests to evaluate randomness of a binary sequence. The generated TRNG sequence should pass all of the designed test with a sufficient significance level, so that we can use it in cryptographic applications.

The second metric to evaluate the predictability of TRNG output sequence is ML algorithms. In TRNG applications, we want to make sure the output is not predictable given the previous part of the output sequence. Due to the importance of past output bits, history-based ML algorithms such as long short-term memory could be used. In these algorithms, the output sequence of TRNG sequence would be used for both input and output [20]. Specifically, N adjacent bits within response sequence are used as one input, whereas the immediate bit after the input bit sequence is used as the output. Ideally, the ML accuracy should be 50% for the TRNG output sequence meaning that the output is not predictable even if the past N bits are given.

The third metric that is used to evaluate TRNG output sequence is auto-correlation. Auto-correlation is a mathematical metric that identifies randomness of a sequence as well as independence of each bit of the sequence with respect to the previous bits. Auto-correlation ($R_{XX}[k]$) is calculated by the expectation between two sub-sequences of TRNG output sequence (x) that are separated by k-lag. Auto-correlation can be rewritten as

$$R_{XX}[k] = \frac{E[x_i - \mu, x_{i+k} - \mu]}{\sigma^2}$$

where $k \in [1-N, N-1]$ is the lagged interval, N is the length of sequence x, μ and σ are the mean and the standard deviation of x. In an ideal case, the auto-correlation of a random sequence has a zero auto-correlation for $k \neq 0$ and has maximum value of 1 when $k = 0$. This means that each sub-sequence has no correlation with the other lagged sub-sequences, resulting in a totally random sequence.

2.3 Prior Work

Pappu et al. [21] introduced an optical PUF where the output (response) is a function of the input laser location/polarization (challenge). The optical PUF requires large external measurement devices and is difficult to be integrated on resource-constrained hardware device. Additionally, its reliability is highly dependent on the accurate calibration of the input location.

Gassend et al. [22] then proposed the Arbiter PUF (APUF) which generates response based on the time difference between the two signal paths. The APUF consists of serially connected individual stages, where the path through each stage is determined by the input bit vector. Because the APUF is based on linear additive blocks, it is vulnerable to modeling attacks if an adversary gain access to the CRPs [23, 24]. To increase the complexity of such modeling attacks, some variants of the APUFs such as the XOR-APUFs ([23, 25]) and feed forward APUFs ([23, 26]) are proposed. To implement APUFs, latches are used which can cause meta-stable state, leading to poor reliability. To overcome the meta-stability issues, another type of delay-based PUFs, namely Ring Oscillator PUF (ROPUF) is proposed in [22, 25]. The ROPUF contains ring oscillators which are connected to two counters through two multiplexers. The select lines of the multiplexers become the challenge input to the PUF design. The counters count the number of oscillations. The comparison between the two counted number generate the output (either 0 or 1). Due to the manufacturing variations in the fabrication phase, the oscillation frequencies will not be the same for all ring oscillators, so the output bit will change each time that two ring oscillators are selected through the multiplexers. The proposed ROPUF design is then improved in [27] and [28]. [29] includes an overview of different ROPUFs.

In addition to the delay-based PUFs, there are other types of PUFs such as traditional memory-based PUFs (e.g. SRAM). As explained in [30, 31], a SRAM cell consists of six transistors whose initial random states determine the PUF output when power is applied to the SRAM cell. The overview of other conventional PUF architectures is studied in [1] and [14]. The mentioned conventional PUFs exploit uncontrollable process variation in conventional Complementary-Metal-Oxide-Semiconductor (CMOS) fabrication technology.

Although the conventional CMOS-based PUFs are well established in industry, the technological developments are particularly important for building PUFs. As a result, the next generation of PUFs will be implemented using emerging nano-electronic devises [5]. Nano-technologies such as memristors ([6]) and flash memories ([7]) provide new opportunities due to severe inherent randomness (as a result of scaling down to nano region), low-energy consumption, simple fabrication, and CMOS compatibility [8, 9].

The memristor- and flash-memory-based weak PUFs operate based on the intrinsic statistical (mostly in switching characteristics) properties of a single memory cell in an array of devices (Figure 2.1). Therefore, the CRP space is a linear function of the array size which makes the output predictable after observing a subset of the CRPs. As shown in Figure 2.1, with each CRP, a challenge is used as the address to select rows and columns. The response is then generated by comparing the resistance of two selected devices or by comparing the current of the selected device with a reference current [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43]. Table 2.1 provides the comparison between the experimental features of memristor- and flash-memory-based weak PUFs [8].

The memristor- and flash-memory- based strong PUFs operate based on the intrinsic

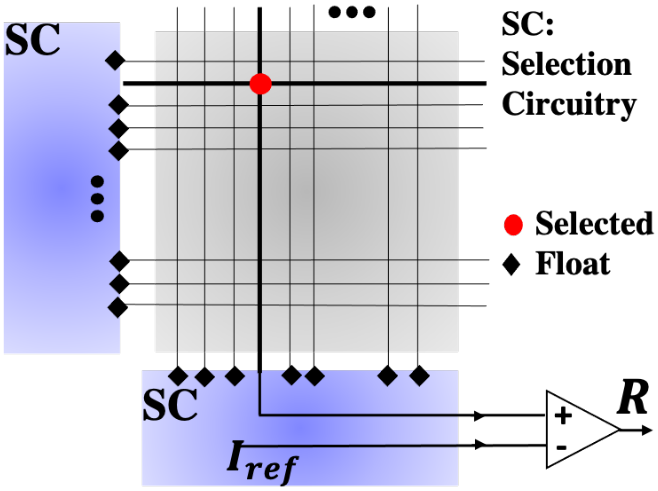


Figure 2.1: Basic architecture of NVM-based weak PUF.

statistical properties (mainly device-to-device variations) of multiple memory cells in an array or layers of arrays of devices. The basic idea is shown in Figure 2.2 where the current of two paths (which includes the sneak-path currents) are compared with each other to ensure that all the devices (selected, half-selected, and non-selected) are contributing to the response.

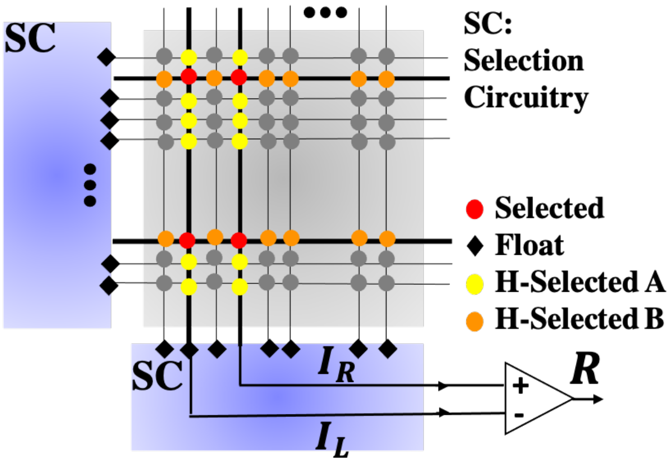


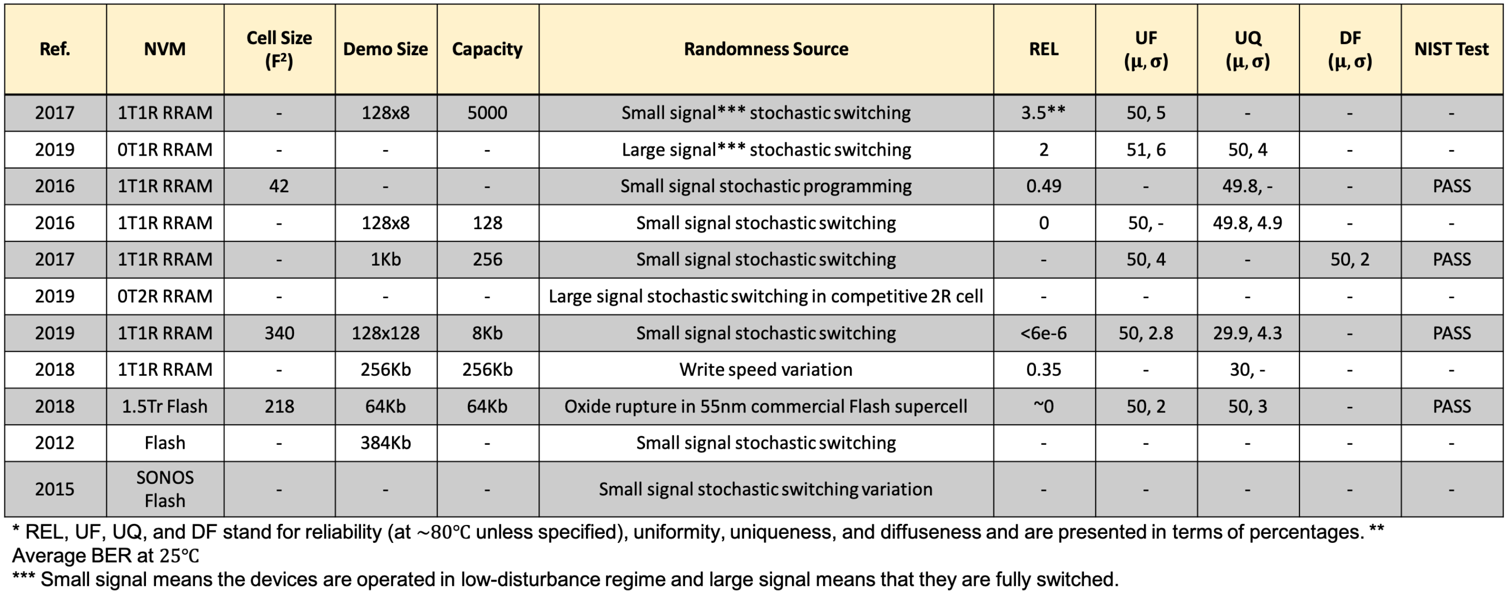
Figure 2.2: Basic architecture of NVM-based strong PUF.

The first demonstration of the NVM-based strong PUF is proposed in [20, 44] which utilizes the variations in the nonlinear, analog tunable I–V characteristics of passive memristors. The basic building block of the proposed PUF is a CMOS-compatible two-level stack of 10x10 memristor arrays. In each CRP, 5 rows and 2 columns are selected. Thus, the CRP space is $\binom{20}{5} \times \binom{10}{2} \approx 7 \times 10^7$. The experimental data for about 4×10^7 collected CRPs show 50% uniformity, 50% uniqueness, 50% diffuseness, and 1.5% reliability. Moreover, the responses pass NIST tests and show resiliency against a 30 × 250 × 250 × 1 MLP classifier.

The proposed PUF ([20, 44]) has been then extended in [45] using the same source of randomness. [45] consists of a 20 × 20 memristors whose conductance values are chosen from a Gaussian distribution. It is shown that the devices which have the tail values of the distribution could result in a week bias in the output. To overcome this issue, two auxiliary lines in the array are used to generate the response in two cycles. In the first cycle, the currents sensed at the auxiliary columns are compared, and then, it is XORed by the result of the generated bit in the next cycle to produce the final bit. This procedure is introduced as resistive-XOR PUF (RX-PUF). RX-PUF show 50.04% uniqueness, 50% diffuseness, and 4.1% reliability. In addition, the responses pass NIST tests. Moreover, the results show that RX-PUF has high resiliency against a 40 × 500 × 500 × 1 MLP classifier which is trained on a subset of observed CRPs ($\approx 120 \times 10^3$) and then tested on a mutually exclusive observed set (5×10^3). Table 2.2 provides the comparison the experimental features of memristor-based strong PUFs [8].

2.4 Summary

We started this chapter with a description of PUF, its main types, and applications. Next, we discussed the most important metrics by which the quality of PUFs is assessed. Then, we discussed the description of TRNG followed by the main metrics by which the randomness of TRNGs are evaluated. Finally, we provided a summary of prior work on PUFs.



Ref.	NVM	Cell Size (F^2)	Demo Size	Capacity	Randomness Source	REL	UF (μ, σ)	UQ (μ, σ)	DF (μ, σ)	NIST Test
2017	1T1R RRAM	-	128x8	5000	Small signal*** stochastic switching	3.5**	50, 5	-	-	-
2019	0T1R RRAM	-	-	-	Large signal*** stochastic switching	2	51, 6	50, 4	-	-
2016	1T1R RRAM	42	-	-	Small signal stochastic programming	0.49	-	49.8, -	-	PASS
2016	1T1R RRAM	-	128x8	128	Small signal stochastic switching	0	50, -	49.8, 4.9	-	-
2017	1T1R RRAM	-	1Kb	256	Small signal stochastic switching	-	50, 4	-	50, 2	PASS
2019	0T2R RRAM	-	-	-	Large signal stochastic switching in competitive 2R cell	-	-	-	-	-
2019	1T1R RRAM	340	128x128	8Kb	Small signal stochastic switching	<6e-6	50, 2.8	29.9, 4.3	-	PASS
2018	1T1R RRAM	-	256Kb	256Kb	Write speed variation	0.35	-	30, -	-	-
2018	1.5Tr Flash	218	64Kb	64Kb	Oxide rupture in 55nm commercial Flash supercell	~0	50, 2	50, 3	-	PASS
2012	Flash	-	384Kb	-	Small signal stochastic switching	-	-	-	-	-
2015	SONOS Flash	-	-	-	Small signal stochastic switching variation	-	-	-	-	-

* REL, UF, UQ, and DF stand for reliability (at ~80℃ unless specified), uniformity, uniqueness, and diffuseness and are presented in terms of percentages. ** Average BER at 25℃

*** Small signal means the devices are operated in low-disturbance regime and large signal means that they are fully switched.

Table 2.1: Comparison of experimentally demonstrated memristor- and Flash-memory-based weak PUFs.

Ref.	NVM	Cell Size (F^2)	Demo Size	Capacity	Randomness Source	REL*	UF* (μ, σ)	UQ* (μ, σ)	DF* (μ, σ)	ML** Test
2016	0T1R RRAM	4	12x12	~1000	Switching threshold variations	-	-	46, 3	-	YES
2018	0T1R RRAM	4	3D 2x10x10	~7x10^6	Variations in nonlinearity and tunability	1.5	50, -	50, 0.9	50, -	YES
2018	0T1R RRAM	4	20x20	~40x10^6	Variations in nonlinearity and tunability	4.1	50.04, -	-	50, 6	YES

* REL, UF, UQ, and DF stand for reliability (at ~80°C unless specified), uniformity, uniqueness, and diffuseness and are presented in terms of percentages.
** ML stands for machine learning.

Table 2.2: Comparison of experimentally demonstrated memristor-based strong PUFs.

Chapter 3

Improving Machine Learning Attack Resiliency via Conductance Balancing in Memristive Strong PUFs

As thoroughly explained in Chapter 2, among various emerging technology PUFs [8, 10], the implementations based on memristive crossbar circuits are especially promising due to their simple and low-cost fabrication process, small footprint, and CMOS integration compatibility [20]. Indeed, prior work has shown memristive strong PUFs with superior resiliency against most powerful modeling attacks as compared to CMOS PUFs [8]. Memristive PUFs utilize spatial device-to-device variations, e.g., in I-V nonlinearity [20, 34, 45], in an array of memory cells to generate random responses. Promising results were also reported for reliability and statistical properties of generated keys, as well as physical performance. The main goal of this study is to further improve the robustness of the memristive strong PUFs based on an architecture presented in [20], [45], [13].

In addition to the PUF robustness improvement, some of the open questions in designing the memristive strong PUFs are explored in this study. For example, the effect of device nonlinearity and leakage current are quantified. Additionally, the impacts of device analog-tunability, crossbar uniformity, and selection scheme are explored.

The rest of the word is organized as follows: Section 3.1 provides a brief overview of the memristive strong PUF circuits. Section 3.2 and 3.3 describe the memristors modeling approach and the evaluation metrics used in this study, respectively. Section 3.4 introduces PUF optimization methods and their results. Section 3.5 explores the hardware imperfection. Finally, Section 3.6 is dedicated to the discussion and summary of the work.

3.1 Background: Memristive Strong PUF

The focus of this study is on the strong PUF circuit (Figure 3.1), consisting of an M×M array of 0T1R memristive crossbar array and its peripheral circuitry (SCs) used for biasing specific rows and columns according to the applied challenge. The memristive PUF circuit can operate in either one-sided [20, 35] or two-sided [46] approaches. The one-sided scheme requires only half of the peripheral circuitry, e.g. left and bottom SCs shown with blue color. Specifically, one bit of the output (i.e. response) is generated when applying 2M bit input (challenge). The '1's in the first M bits of the input encode the positions of "selected" rows (out of M total). Similarly, the remaining M bits specify the position of "selected" columns. All selected rows are biased with a read voltage V_{read}, all selected columns are grounded, while all the remaining lines (rows or columns) are kept floating. The output bit is computed by comparing the total current flowing in the left half of the selected columns I_L to that of the right half I_R, i.e. output is 1 if $I_R > I_L$ and 0 otherwise.

With such a selection scheme, the devices can be classified according to the type of rows/columns they are connected to. The device is called "selected" when both its row and column are selected. The device is called "half-selected" when either its row (type B) or its column (type A) are selected, while the other electrode is floating. The device is called "non-selected" when both of its row and column are floating. Assuming that n rows and m columns are selected in the M x M crossbar array, the total number of the district CRPs is $\binom{M}{m} \times \binom{M}{n}$, which is a very large number even fore moderate M, n, and m values.

In the two-sided scheme, an output bit is generated in two phases. In the first phase, one intermediate output bit is generated as discussed for the one-sided approach. In the second phase, the input biases are applied to the columns, while currents are read from rows to generate another intermediate bit, i.e. effectively using the same single-sided design but with a rotated crossbar array by 90 degrees with respect to the peripheral circuitry. The output bit is then generated by XORing these two intermediate, independent bits. The two-sided PUF features a more uniform response, and hence more robust to the machine learning attacks [13], though at the cost of halved throughput and doubled the energy consumption.

To demonstrate the benefits of the two-sided approach over a one-sided one, the response uniformity and yield of 2K CRPs of 25 instances for a variety of PUF sizes are considered. The realistic values of target conductance distribution (mean of 8.3 μS and standard deviation of 2%) are considered for all cases. The simulation results are plotted in Figure 3.2. As shown in Figure 3.2a, the response uniformity of most of the two-sided PUFs are near-to-ideal (50%) whereas the response uniformity of most of the one-sided PUFs deviates from the ideal value. Additionally, Figure 3.2a shows that response uniformity improves when the size of the PUF crossbar increases. Furthermore, as shown in Figure 3.2b, the yield of the two-sided approach is more than two times

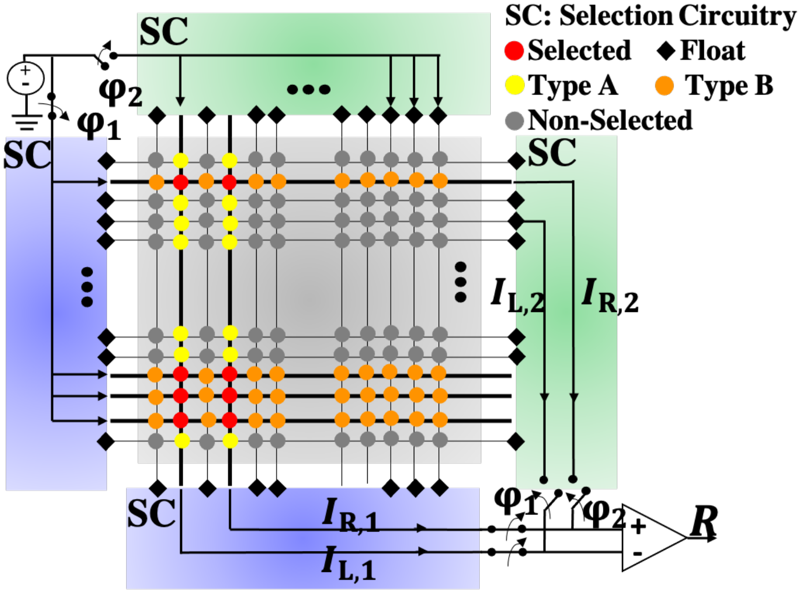


Figure 3.1: The top level architecture of the two-sided PUF design based on passively-integrated (0T1R) memristive crossbar circuit. The crosspoint devices are colored according to the types of crossbar electrodes that they are connected to.

higher than the yield of a one-sided one. The yield is calculated based on the number of cases that pass the NIST frequency tests. Additionally, Figure 3.2b shows that yield of the two-sided approach improves when the crossbar size increases. This trend is not clear for the one-sided approach which may be due to the lack of the number of tested instances.

Based on the results of this study, two-sided PUFs are considered for the rest of this study.

3.2 Modeling Approach

In this study, the PUF metrics were estimated by assuming ideal peripheral circuits and modeling the output currents of the crossbar circuit with the help of the SPICE tool (the simulation setup is briefly explained in Appendix A). To model the memristor static I-V characteristics, the nonlinear current via crosspoint device was approximated with a

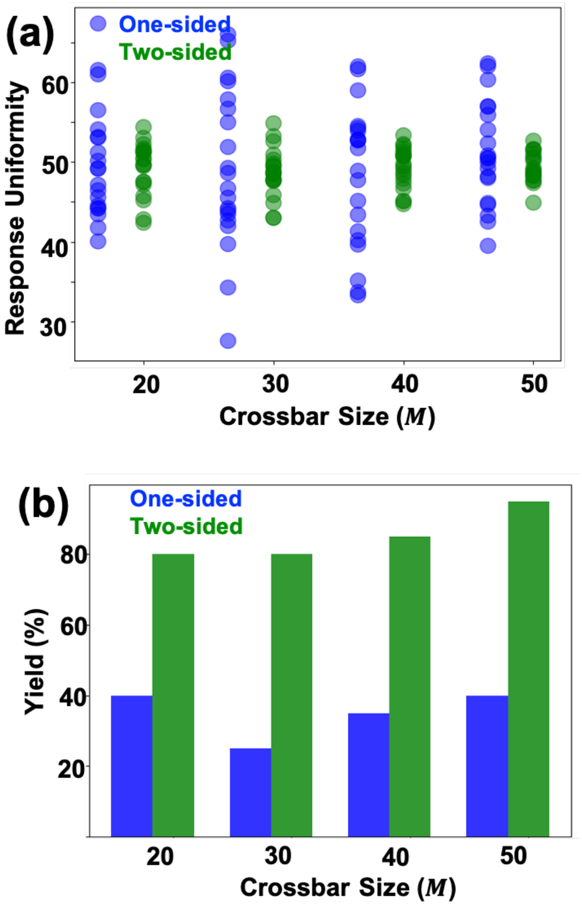


Figure 3.2: The effect of one-sided and two-sided approaches on (a) response uniformity and (b) yield. In both approaches, the response uniformity improves and the yield increases when PUF size increases.

generic expression

$$I = a\sinh(bV)$$

where a and b are constants that capture nonlinearity and device-to-device variations. These constants were randomly indirectly initialized for each device of the crossbar circuit. Specifically, to make the choice of the nonlinearity and variations more intuitive and representative of the real circuits, each device is characterized by the device current at "tuning" voltage V = 0.25 V, and the nonlinearity NL, which is defined as

$$NL = I(0.25V)/I(0.1V) \times 0.1V/0.25V$$

.

Unless otherwise specified, in all simulations in this study, NL is sampled from Gaussian distribution with an average of 1.5 and a specific standard deviation. Additionally, the device current at the tuning voltage is sampled from a Gaussian distribution with an average of 33.3 μA and a specific standard deviation σ. After a unique NL and current at 0.25 V has been assigned for each device, the constants a and b (and hence complete unique static I-V characteristics) are derived. Note that the described approach for choosing currents crudely corresponds to the uncertainty in the tuning process for configurable PUFs [20, 45] as well as representative of the variations in the crosspoint device conductances in the fixed-resistance PUFs based on the as-fabricated devices [46]. Also, note that the absolute values for the tuning currents are not important for this particular study due to the focus on the functional characteristics of the PUFs and the assumption of the ideal peripheral circuits.

3.3 Evaluation Metrics

To assess the performance of the PUF, we consider three main metrics e.g. uniformity (UF), NIST and predictability which are widely discussed in the literature [20, 17, 47]. Note that it is not feasible to measure bit error rate (BER) in this study because the temperature variation and conductance drift models are not available. Additionally, note that uniqueness is more relevant in evaluating experimental data. Furthermore note that diffuseness is a weaker PUF metric as it stays near to ideal value (50%) even if other metrics show low performance.

To perform predictability analyses, machine learning models are considered as they are currently the most effective attack form for strong PUFs [13]. Memristive crossbar PUF has a nonlinear input-output relationship, a huge CRP space, and a time-independent output response. As a result, multilayer perceptron (MLP) is chosen as an attack over logistic regression (used for linear-separable data), support vector machine (runs very slow for huge data), and recurrent neural network and long short-term memory (both use history of data). The studied MLP network consists of 2M inputs so that the challenge can be directly applied to the MLP input, and one output, corresponding to the PUF output, while the number of layers/neurons in the hidden layer(s) were varied in the simulations. A rectified linear (sigmoid) activation function was used for the hidden (output) layer neurons. The MLP classifier was trained and validated using the conventional backpropagation method on 80% of the simulated CRPs. The trained network is then used to predict PUF response on the remaining, mutually exclusive 20% of the CRPs.

3.4 PUF Optimization

This section first describes two proposed techniques for improving robustness against machine learning attacks. The common rationale for both techniques is that PUF robustness is increased when all crosspoint devices in the crossbar equally contribute to the output currents. The PUF response, in this case, would be a nontrivial function of the input, which depends on the unique I-V characteristics of all devices in the crossbar array. Furthermore, this section explores how the unique features of memristors namely device nonlinearity and analog tunability excel PUF security metrics. Moreover, this section studies the effect of crossbar size on PUF predictability.

3.4.1 Optimal Selection Ratio

This section proposes a technique to improve PUF robustness against machine learning attacks by maximizing the contribution of the current of all devices in I_L and I_R. The requirement for the balanced contribution can be simplified to having currents via selected devices similar to those via (type A) half-selected devices, given that the output current is the sum of these two parts. The circuit parameters for having similar currents can be found from the approximate equivalent circuit of the crossbar array (Figure 3.3a), which is derived assuming negligible line resistance and similar static I-V characteristics of all crosspoint devices. Using the approximate equivalent circuit, the selected current and the leakage current can be written as $s^2M^2a\sinh(bV_S)$ and $s(1-s)M^2a\sinh(bV_{HSA})$, respectively.

Our preliminary analysis for the considered average NL shows that selection ratios $n/M = 0.25$ and $m/M = 0.2$, i.e. $n = 8$ and $m = 6$ for $M = 32$, are close to the optimal values. To see if these selection ratios actually lead to the maximum contribution of all devices, a 32×32 crossbar with a fixed row selection ratio (0.25) and different column selection ratios (changes from 0.0625 to 0.5) are considered and 200K CRPs are collected for each of the PUFs. Then, the NIST test suite is conducted (Figure 3.3b) and MLP is applied to verify the predictability of the PUF output (Figure 3.3c). Although all column selection ratios pass the NIST test suite (P-values are greater than 0.01), the MLP accuracy is optimal when the column selection ratio is 0.2. In fact, with this selection ratio, the contribution of all devices is maximized resulting in more complex and less predictable PUF behavior. These optimum selection ratios are considered for the rest of the work.

3.4.2 Balancing Crossbar Array Conductances

The main focus of the work is on the technique of balancing the crossbar array conductances, which allows improving PUF robustness by optimizing crosspoint device conductances resulting in a uniform crossbar. Figure 3.4a presents a motivation for this technique. It shows that PUF becomes more predictable as the dispersion in the device conductances grow. This is explained by the fact that for larger σ, the output current is more likely dominated by only a few devices with larger conductance – a feature that apparently makes such PUF easy to model with MLP network.

Given the challenges in the accurate tuning of the memristors, especially in the passive crossbar circuits, the natural goal is to achieve better robustness for larger σ. The specific objective of this study is to find such optimal mapping of the devices with predetermined (fixed) I-V characteristics to the locations in the crossbar array circuit that would maximize the PUF robustness. A balancing heuristic algorithm is introduced to address this goal (Figure 3.5). The algorithm tries to balance the total device conductances (at tuning voltage 0.25 V) across rows and columns. The intuitive idea behind such an approach is that in the crossbar array with matched conductances along the rows and columns, the

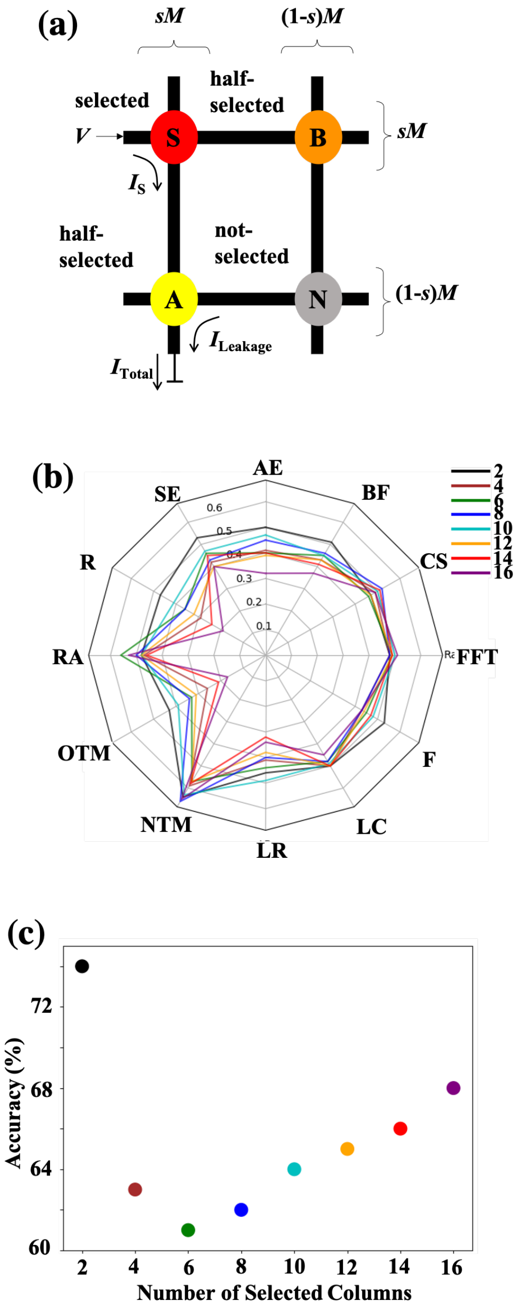


Figure 3.3: (a) simplified equivalent model of M x M crossbar circuit when read voltage V_{read} is applied to sM selected rows, the output currents are read from sM virtually grounded columns, while all the remaining lines are floated. (b) Effect of different column selection ratio in 32 × 32 crossbars (b) on NIST test suite and (c) on MLP prediction accuracy.

devices with larger conductances along the current path will be compensated with those with the smaller conductances, which would in turn help in making the output currents close to each other and ultimately reduce the bias in the PUF response.

Specifically, starting with random mapping, the algorithm tries to iteratively swap the locations of two randomly chosen memristors to minimize the cost function

$$\Gamma = \sum_i (\sum_j G_{ij} - MG_a)^2 + \sum_j (\sum_i G_{ij} - MG_a)^2$$

where G_{ij} is a conductance at 0.25V via device located in the i-th row and j-th column and $G_a = \sum_i \sum_j G_{ij} - (0.25V)/M^2$ is an average conductance of all devices in the simulated instance of the crossbar array. The first/second term in the cost function is a sum of squared differences between the conductance of the row/column and the global average value. The cost function optimizes PUF for both one-sided and two-sided architectures and is independent of the number of selected columns.

A simulated annealing approach was implemented so that a move is always accepted if the cost function is reduced, while it is accepted with a certain probability, determined by the change in the cost and the current annealing temperature, even if the cost is increased. The annealing parameters are chosen such that most of the memristors are swapped multiple times.

Figure 3.6a-c shows an example of applying an algorithm for a 10×10 crossbar array. The sum of the conductances across rows and columns has significant dispersion for the initial, random distribution of conductances (Figure 3.6a), while these sums become very close to each other after applying the algorithm (Figure 3.6b). Figure 3.6c shows how the value of the cost function reduces after each iteration. The algorithm effectiveness is investigated for different scenarios of the machine learning attacks (Figure 3.7). In the first study, the prediction accuracy of the MLP network was studied as a function of the

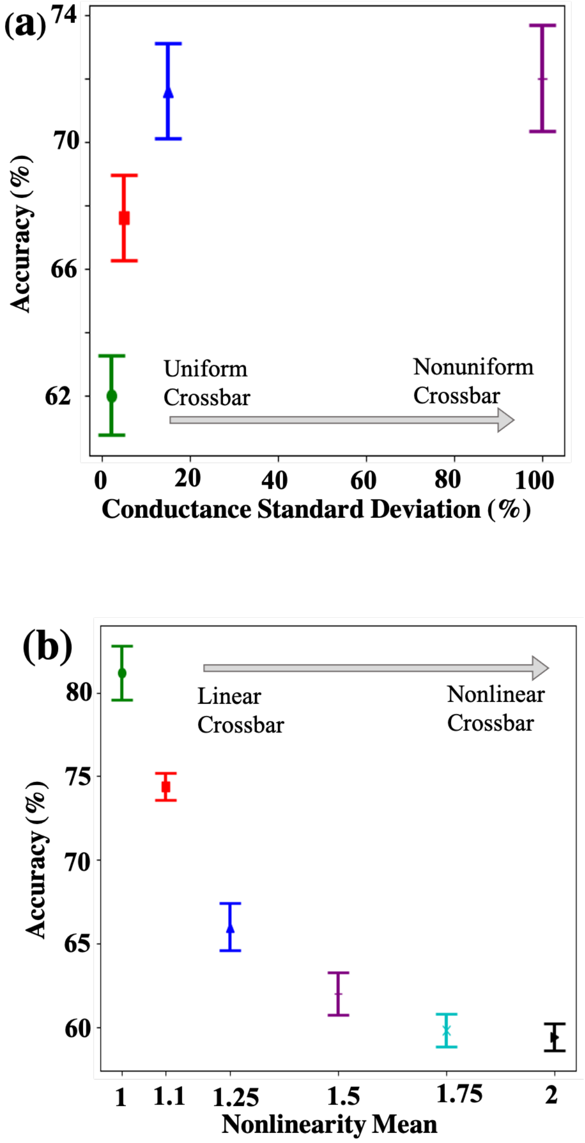
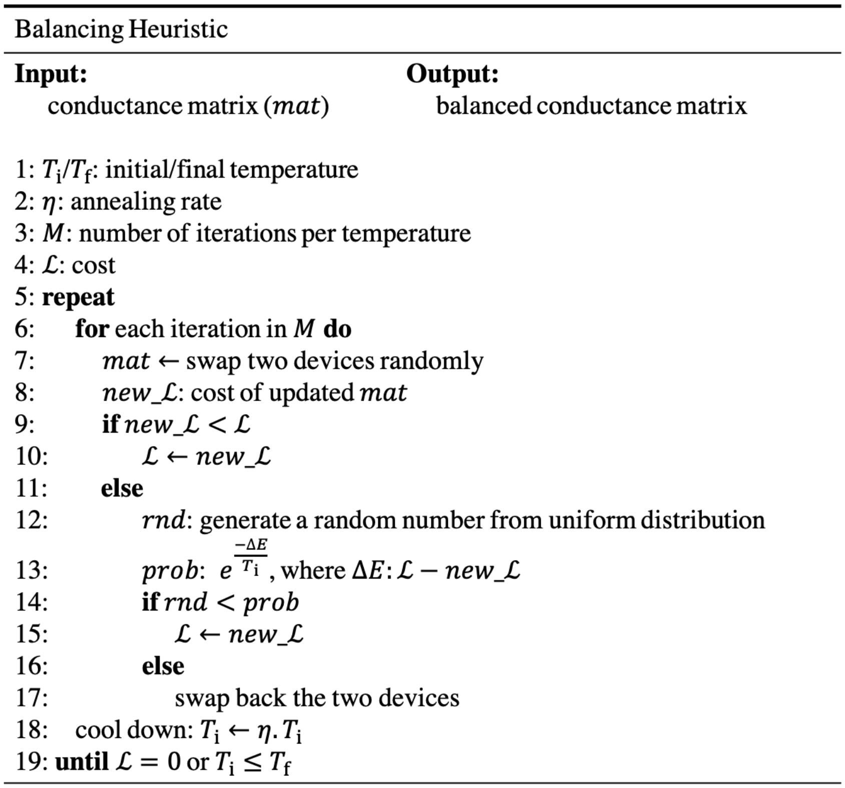


Figure 3.4: Prediction accuracy of two-sided PUF response modeled by 40-100-1 MLP network as a function of the (a) the crosspoint device conductance variations σ and (b) device nonlinearity mean value. The error bars represent standard deviation for the 5 simulated PUF instances, each with different device I-V characteristics. For all cases, $M = 20$, $n = 5$, $m = 4$, and $V_{bias} = 0.3$V. MLP accuracy decreases when the device nonlinearity mean value increases or when the conductance standard deviation decreases.



```
Balancing Heuristic
Input:                                  Output:
    conductance matrix ($mat$)              balanced conductance matrix

$T_\mathrm{i}/T_\mathrm{f}$: initial/final temperature
$\eta$: annealing rate
$M$: number of iterations per temperature
$\mathcal{L}$: cost
repeat
    for each iteration in $M$ do
        $mat \leftarrow$ swap two devices randomly
        $new\_\mathcal{L}$: cost of updated $mat$
        if $new\_\mathcal{L} < \mathcal{L}$
            $\mathcal{L} \leftarrow new\_\mathcal{L}$
        else
            $rnd$: generate a random number from uniform distribution
            $prob$: $e^{\frac{-\Delta E}{T_\mathrm{i}}}$, where $\Delta E$: $\mathcal{L} - new\_\mathcal{L}$
            if $rnd < prob$
                $\mathcal{L} \leftarrow new\_\mathcal{L}$
            else
                swap back the two devices
    cool down: $T_\mathrm{i} \leftarrow \eta . T_\mathrm{i}$
until $\mathcal{L} = 0$ or $T_\mathrm{i} \le T_\mathrm{f}$
```

Figure 3.5: Pseudo-code of the proposed heuristic algorithm for balancing conductances in the crossbar. The typical values for initial / final temperatures, annealing rate, and the number of iterations per temperature are 2 / $1e^{-9}$, 0.95, and 5000, respectively.

number of CRPs used in training for three cases of the crossbar conductances (Figure 3.7a). For a smaller number of CRPs, the accuracy is close to the ideal 50% when the device conductance distribution in the crossbar array is very tight. The accuracy is more than 60%, on average, for the naive (random) mapping with $\sigma = 25\%$, though the application of the algorithm allows reducing it to the ideal value. As expected, increasing the number of CRPs makes machine learning attacks more effective, though the prediction accuracy seems to saturate, which might be related to the limited capacity of the used MLP network.

The impact of the MLP capacity is further investigated by increasing the number of hidden layer neurons for the two-layer network (Figure 3.7b) and increasing the number of hidden layers while fixing the number of hidden layer neurons (Figure 3.7c). The accuracy first rapidly improves and then saturates in the former study case, which is likely due to the limited number of CRPs. Surprisingly, the accuracy is almost independent of the number of hidden layers for the latter. Finally, just like for the first experiment, the algorithm allows reducing prediction accuracy for the PUFs with $\sigma = 25\%$ device conductance distribution to that of naive one with $\sigma = 2\%$, which confirms the effectiveness of the algorithm.

It should be noted that the crossbar (either naive or balanced one) is secure against side-channel attacks because generating I_R/I_L does not reveal any extra information compared to I_R/I_L [47, 48, 49, 50, 51]. To study this claim, the power profile of 5K CRPs of 10 20×20 PUF instances are collected. Statistics show that 50% of the time response = 1 consumes more power than response = 0. This is because measuring none of the I_L and I_R has any power dominance comparing to the other one.

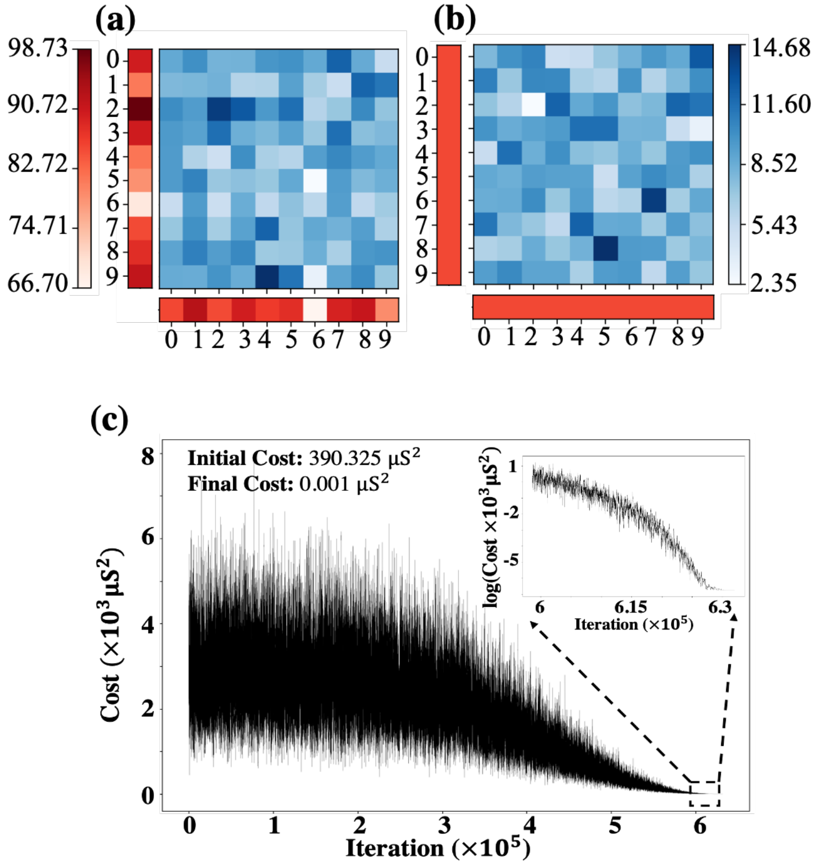


Figure 3.6: The example of applying balancing algorithm for a 10×10 crossbar circuit with $\sigma = 25\%$: (a) conductance heat-map before and (b) after applying the algorithm. The color bars at the edges of the arrays show the total conductances summed along the corresponding rows and columns. (c) The corresponding evolution of the cost function. All conductance maps are specified at 0.25V.

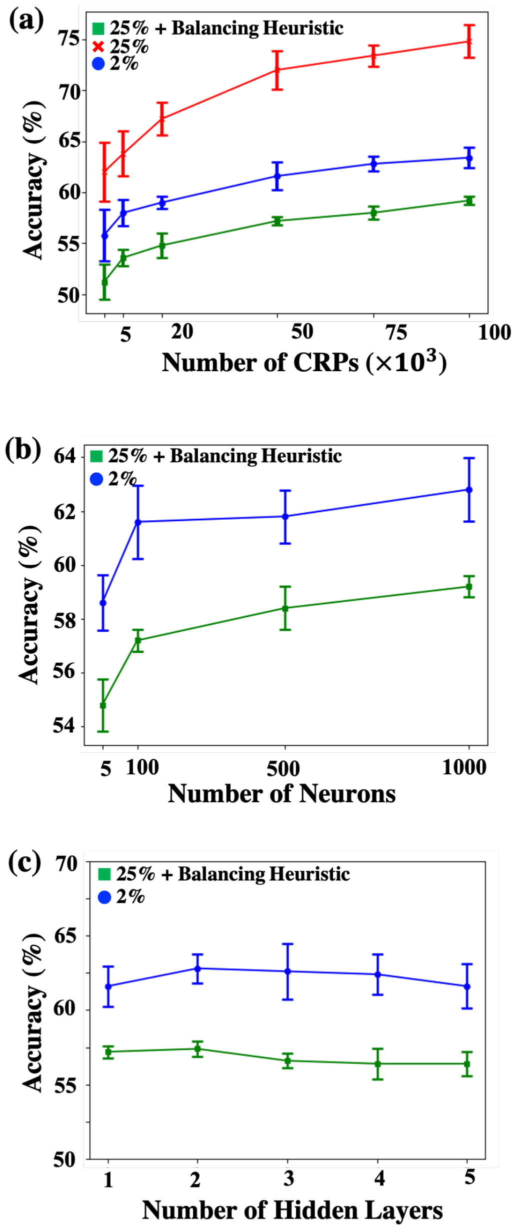


Figure 3.7: MLP prediction accuracy of the two-sided PUF (M = 20, n = 5, m = 4) response as a function of (a) the number of CRPs used in training 40-100-1 network, (b) the number of hidden layer neurons in two-layer network, and (c) the number of hidden layers in 40-100-...-100-1 network. There different scenarios for crossbar conductances scenarios were simulated: $sigma$ = 2% (blue circle symbols), $sigma$ = 25% without applying algorithm (red cross symbols), $sigma$ = 25% with applying algorithm (green square symbols). In panel b and c studies, 50K CRPs were used for training MLP network. The error bars represent standard deviation for the 5 simulated PUF instances, each with different device I-V characteristics.

3.4.3 Device Nonlinearity and Analog Tunability

The memristive PUFs have been widely studied in different literature [4, 8, 9], but they lack detailed studies on the effect of memristor nonlinearity and analog tunability on PUF security metrics. The former device feature leads to nonlinear PUF operation which makes the modeling attacks almost impossible. The latter device feature leads to tunable PUF which means the devices can be custom-tuned for a specific goal in the configuration phase. The analog-tunability of memristors results in a uniform crossbar which makes output random and independent of the input.

To study the effect of the device nonlinearity, 50K CRPs of 5 nonlinear and linear 20×20 PUF instances (one-sided approach) are simulated. Furthermore, to study the effect of device analog tunability, 50K CRPs of 5 analog and digital 20×20 PUF instances (one-sided approach) are simulated. Specifically, in analog crossbars, the target conductances are chosen as explained in Section 3.2 whereas in digital crossbars, the target conductances are chosen from two Gaussian distributions that are centered on ON- and OFF- conductance values (mean value of 8.3 μS and 1 μS, respectively). Based on the simulation results (Figure 3.8), nonlinear, analog-tunable, memristive crossbars outperform resistive crossbars and digital crossbars in UF (Figure 3.8a), NIST frequency test (Figure 3.8b), ML (Figure 3.8c), and entropy (Figure 3.8d).

Moreover, to evaluate the PUF robustness as a function of device nonlinearities, NL is swept from 1 to 2. As results are demonstrated in Figure 3.4b, when device nonlinearity increases, the PUF output will be a more complex function of all devices resulting in a less predictable PUF behavior.

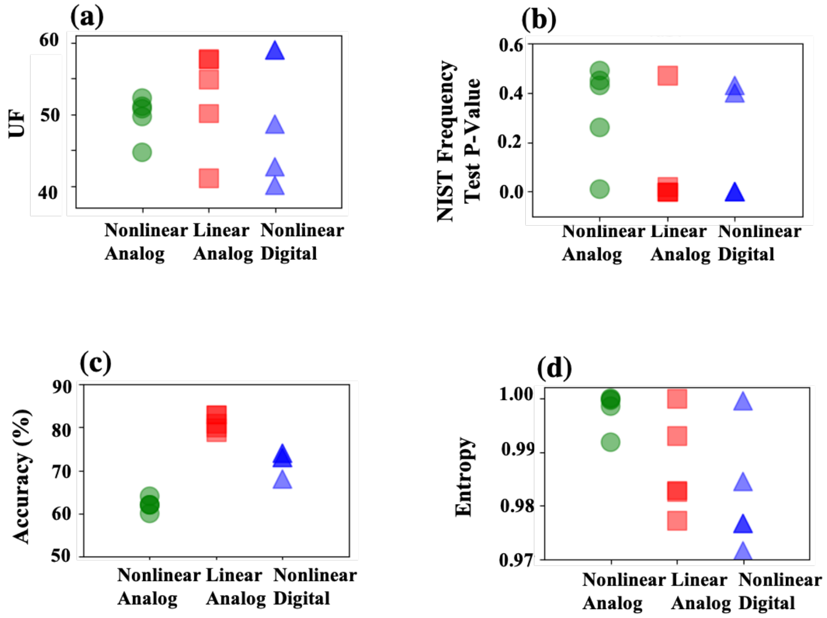


Figure 3.8: Effect of device nonlinearity and analog tunability on (a) UF, (b) NIST frequency test P-value, (c) MLP prediction accuracy, and (d) Entropy. Nonlinear analog crossbar outperforms linear and digital crossbars in all four mentioned metrics.

3.4.4 Crossbar Size

Another design variable that affects PUF predictability is crossbar size. In fact, when the crossbar size increases, the PUF complexity increases which results in a less predictable PUF behaviour. To study this claim, 1M CRPs for 20 × 20, 30 × 30, 40 × 40, and 50 × 50 PUF instances are collected. The realistic values of $\mu_G = 8.3$ μS with 25% variations in the target conductance distribution as well as balancing algorithm are considered for all cases. As shown in Figure 3.9, when the size increases, the MLP accuracy reduces. This is because when the crossbar size increases, the number of ML features increases quadratically and the number of CRPs increases exponentially.

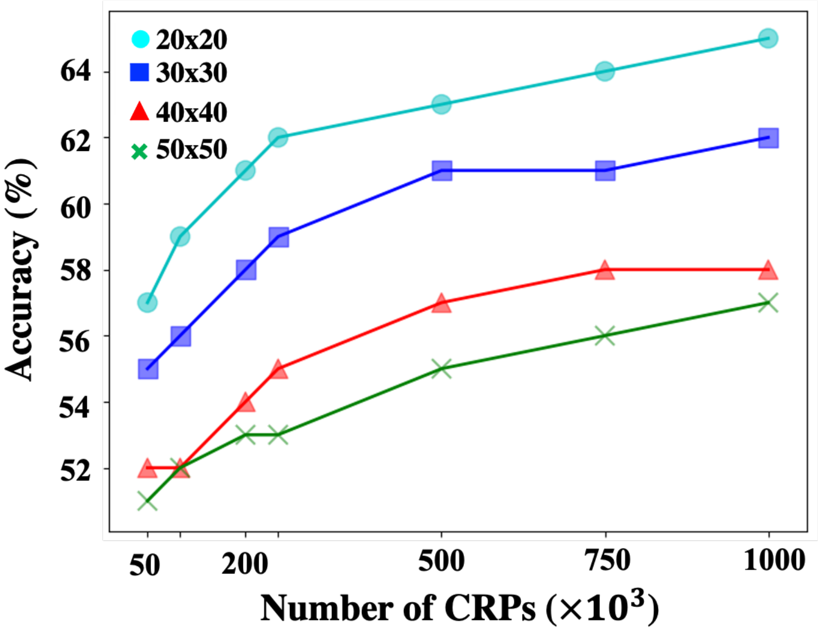


Figure 3.9: Effect of crossbar size on ML accuracy for different sizes of CRPs. When the crossbar size increases, the PUF complexity increases which results in a less lower ML accuracy.

3.5 Hardware Imperfection

Previous studies explored how IR-drop ([4]) affect PUF robustness. As explained in [4], when the wire resistance of interconnects are non-zero ideal, it causes IR drop along interconnects. In fact, the devices that are closer/further than the voltage source will have a lower/higher IR drop along the interconnects which results in insufficient voltage over crosspoint devices. As a result, some of the devices will have greater current reduction resulting in an undesired bias which reduces PUF reliability.

This study studies the effect of another hardware imperfection on PUF, namely non-ideal yield. When the yield is not 100%, some faulty devices exist that are stuck- at either ON or OFF state. When a device is stuck-at ON/OFF, its current is much higher/smaller than other devices. In this case, the column that consists of the stuck-at fault device becomes dominant resulting in a bias in the output which makes PUF unreliable. To simulate the effect of yield on PUF reliability, we collected 5K CRPs of 5 PUF instances for a variety of ReRAM-based PUF sizes which have different percentages of stuck-at ON (without loss of generality) devices. The realistic values of $\mu_G = 8.3$ μS with 2% (without balancing algorithm) and 25% (with balancing algorithm) variations in the target conductance distribution are considered for all cases.

The simulation results for PUF uniformity as a function of yield before and after applying the balancing heuristic are plotted in Figure 3.10a and Figure 3.10b, respectively. As shown in Figure 3.10a, initially, UF diverges from the ideal 50% when yield decreases. However, it converges back to the ideal value at some point. This is probably because the effect of some of the stuck-at ON devices is compensated with some other ones. Additionally, as shown in Figure 3.10b, the balancing heuristic improves PUF uniformity because it can map the devices such that devices with lower conductance values be in the same column as the stuck-at ON devices.

Furthermore, the simulation results for PUF uniformity as a function of crossbar size before and after applying the balancing heuristic are plotted in Figure 3.10c and Figure 3.10d, respectively. As shown in Figure 3.10c, the UF will be more resistant to hardware imperfection caused by yield when size increases. In fact, when the PUF size increases, the CRP space increases exponentially. Therefore, it will be less probable for the stuck-at faulty device to be selected. Additionally, as shown in Figure 3.10d, the balancing heuristic improves PUF uniformity because it can map the devices such that devices with lower conductance values be in the same column as the stuck-at ON devices.

3.6 Discussion and Summary

The simulation results confirm that the robustness to machine learning attacks of memristive PUF is significantly improved by balancing conductances in the crossbar

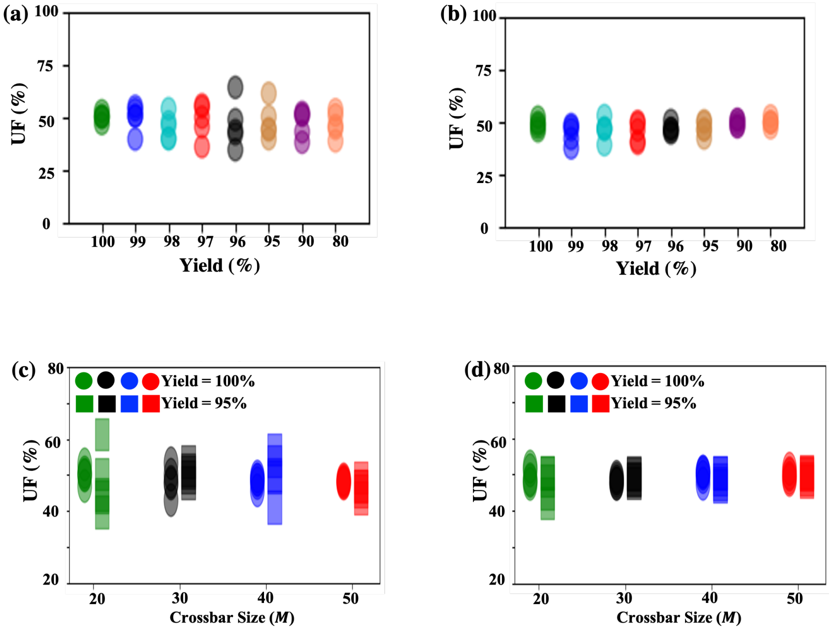


Figure 3.10: UF values of different 20 × 20 PUF instances when yield is not ideal in absence (a) and presence (b) of the balancing algorithm. UF values of different sizes when yield is not ideal in absence (a) and presence (b) of the balancing algorithm. In all panels, 5K CRPs were used for the 5 simulated PUF instances, each with different device I-V characteristics.

circuit, which can be achieved by either enforcing very similar conductances of all crossbar devices or by using the proposed heuristic algorithm. The latter can be helpful in the reconfigurable memristor PUFs [20, 38], for which the crosspoint device conductances can be tuned to certain optimal values. Additionally, when only crude tuning is possible, the proposed algorithm can be extended to dynamically update the optimal distribution of conductances based on the measured states of the already tuned devices. In addition, when stuck-at fault devices are present in a PUF, the proposed algorithm can map the devices so that the effect of the stuck-at fault devices become minimized. More balanced device conductances naturally lead to a narrower distribution of differential currents $I_L - I_R$. The impact of σ and conductance balancing is not just in the scaling of such distribution but also in making the responses less correlated, as confirmed by the MLP modeling. The tighter margins in reading differential currents, however, may degrade reliability. This issue is currently neglected in our study, in part due to the assumption of ideal peripheral circuitry. The investigation of the reliability/robustness trade-off is the next important immediate goals.

Chapter 4

Lightweight Integrated Design of PUF and TRNG Security Primitives Based on eFlash Memory in 55nm CMOS

A fundamental part of secure Internet of Things systems is authentication, which is to confirm the identity of a prover entity to a verifier. A mutual authentication is a common approach based on challenge–response protocol, which requires physically unclonable functions (PUFs) and true random number generators (TRNGs) as security primitives. The widely researched mutual authentication protocol is shown in Figure 4.1 [52, 53]. In this protocol, the server encrypts the PUF identifier (K_1) with the nonce generated by its local TRNG (T_1) and sends the encrypted data (E_1) to the device. On reception, the device decrypts the cipher-text by its own key (K_1) to extract the nonce. Then, it encrypts the key with a locally generated number (T_2 and $T_1 + T_2$) and transmits them (E_2 and E_3) to the server. At this stage, the server decrypts the pair and extracts the initialization nonce prior to granting the authentication. For encryption and decryption, an advanced encryption standard (AES) algorithm, based on the same number of PUF and TRNG bits, is utilized. As the data are encrypted, there is no access to raw data which reduce the scope of side-channel attacks [52, 53].

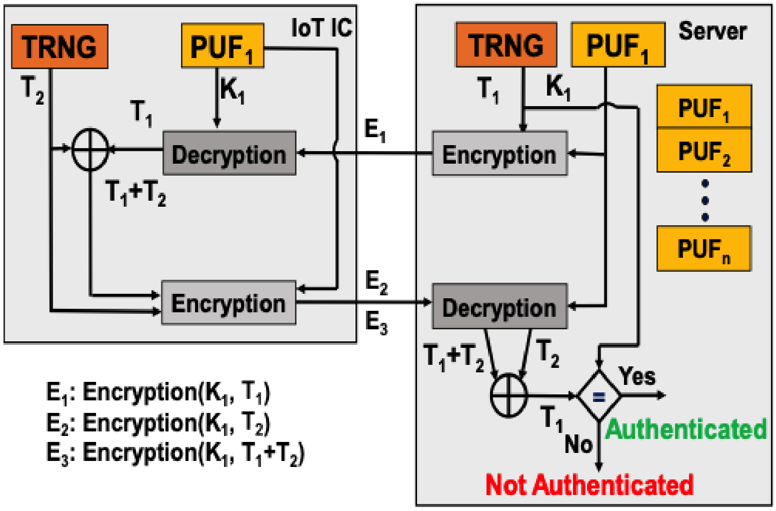


Figure 4.1: Privacy-preserving mutual authentication protocol with PUF and TRNG security primitives [52]

The main entropy generators in Internet of Things devices operating based on the mutual authentication are PUFs and TRNGs. The former generates a stable unique identifier by exploiting the underlying process variations in integrated circuit fabrication, whereas the latter is used to create a stochastic unbiased bitstream, which has no correlation with the circuit features. For edge devices, operating for years on batteries and harvested energy, low-power, low-cost, and secure implementation of entropy generator is a big challenge.

While most of the previous works focus either on PUF [54, 55, 56] or TRNG [57, 58, 59, 60], the very recent work [52] describes a unified approach of generating both PUF and TRNG from a common entropy source. Such approach improves the area utilization by 25% over standalone designs and achieves an excellent energy efficiency. This work introduces a low-power and secure design of integrated PUF and TRNG using analog-grade embedded flash memories. The presented work is the extension of our previous PUF design [61] and has the following key contributions.

1. Integration of TRNG design into the previously proposed PUF security primitive by modifying the architecture and reusing the same silicon circuits (which results in a low-power and dense architecture). We demonstrate that the designed TRNG satisfies the relevant cryptography features and tests.

2. Extensive aging measurements and verification of the behavior of the circuit under extreme temperature conditions, which is crucial for the long-term reliability.

3. Extended analysis of the PUF robustness against advanced machine learning techniques. The operating principle of the proposed unified entropy generator is discussed in Section 4.1, whereas the results for various metrics are provided in Section 4.2.

4.1 Unified Entropy Generator

In this work, commercial embedded NOR flash memories, with a compact $\sim 25F^2$ footprint, where F is the feature size, are used as the base of the integrated design of PUF and TRNG. In [61], we showed several rich sources of variations in this technology. Subthreshold slope (and leakage current) variations in devices biased in weak inversion are the major source of randomness harnessed in the present approach for building a complex one-way function. In addition, the analog-grade flash memories allow fine-tuning of their states using write–verify algorithm with high accuracy. The unpredictable programming error is the second source of randomness. By means of extensive experimental measurements on such analog-grade floating gate devices, we have reported process-induced variations in the static behavior of eFlash memories and preliminary results on designing a low-power PUF circuit [61]. (Note that implementation of a specific PUF instance involves one-time tuning of all cells, in a trusted environment, to some predetermined in advance desired values [62].)

4.1.1 Operation Principle

As shown in Figure 4.2a, the top-level architecture consists of two layers (L_1 and L_2) of 25-to-1 primitive blocks, one 40-bit hidden shift register (HSR), and two 2-to-1 XOR gates. The whole architecture is fed by 1010 challenge bits. Each C_1^1, . . . ,C_5^1, . . . ,C_1^8, . . . ,C_5^8 input has 25 challenge bits and is used to fed one of the primitive blocks of L_1. In addition, each C_6 and C_7 input has 5 bits and is used to fed one of the primitive blocks of L_2. The 1010 challenge bits are given in nine sequential cycles. Specifically, during each of the first eight cycles, 125 bits of challenge are applied to L_1, and the generated 5 bits are stored in HSR. At the end of eight cycles, 1000 bits of challenge are used to calculate 40 bits, which are then stored in HSR. In the ninth cycle, 10 remaining bits of the challenge along with 40 generated bits are applied simultaneously to L_2 followed by XOR gates to generate a single PUF bit (R_F) or a TRNG bit (T_F).

Figure 4.2b shows the structure of each primitive block, which consists of eFlash memory array, switching circuits, and a comparator. A 10 × 10 eFlash memory array is the main part of the primitive block, in which all the devices are operating in deep subthreshold. This is easily ensured during programming phase with the prior knowledge of applied biasing voltages. The operation is extremely nonlinear because for each flash memory cell, the current has an exponential dependence to the applied drain–source voltage in a weak inversion. In addition, all the flash memory cells contribute to the output either directly or indirectly through the circulation path of leakage current which adds to the nonlinearity of the operation. The idea is similar to the one in [20], in which sneak path currents are circulated in a crossbar of passively integrated memristors to build a compact security primitive. However, memristor fabrication technology is still in need of improvement to enable building practical security systems. Commercial flash memories, on the other hand, are already embedded in high-end CMOS process technologies and are excellent candidate for low-power operation.

Switching circuits are the other parts of the primitive block, which determine the selection and operations of eFlash memory array. Specifically, each primitive block is fed by a 25-bit challenge, of which 10 bits are used for row selection, 10 bits are used for column selection, and 5 bits are used for source line (SL) selection. When a row is selected, the associated control gate (CG) and word line (WL) are biased with $V_{CG,SEL}$ and $V_{WL,SEL}$, respectively. CGs of the devices in the non-selected rows are left floating, with their WLs connected to $V_{WL,UNSEL}$. When a column is selected, the corresponding bit-line (BL) is grounded. When a SL is selected, it is connected to the dynamic current comparator. The BLs for the non-selected columns and non-selected SLs are floated. With such biasing approach, the floating gate transistors in the memory array are categorized into four groups when a particular challenge is applied, namely: 1) selected, 2) half-selected type-A, 3) half selected type-B, and 4) non-selected devices (Figure 4.2b). In the selected and half-selected type-A devices, SLs and BLs act as drain and source, respectively. In the half-selected type-B devices, the current flows from BL to SL. However, for the non-selected devices, the current can unpredictably flow at either direction based on the selection scheme and current flowing in other devices. This selection scheme in conjunction with the nonlinear operation in weak inversion enables a circulation of leakage current through the floating devices biased in the very nonlinear regime, making the modeling attacks almost impossible. It should be noted that the selection of eFlash memories (e.g. switching circuits) are implemented using only one

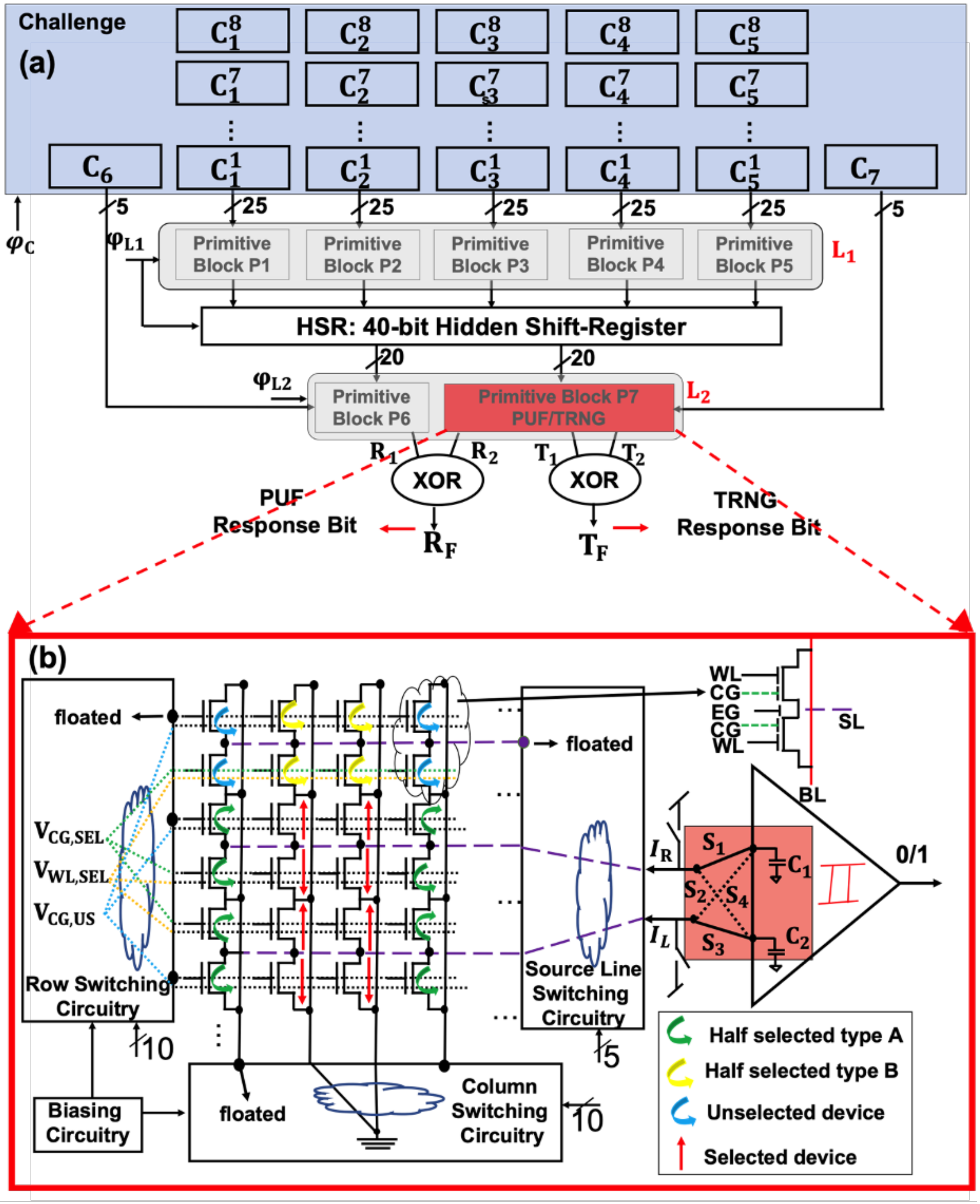


Figure 4.2: (a) The proposed time-multiplexed entropy generator and (b) primitive block topology. The top hierarchy in panel (a) consists of 7 primitive blocks, and is fed by a 1010 bits of a challenge. Each primitive blocks is supplied with 25 bits.

MOS transistor switch per line because '1's/'0's of an input bit-vector directly specify position of the selected/non-selected lines.

To compute the output bit in PUF mode, S_1 and S_3 are closed after the capacitors C_1 and C_2 are pre-charged to V_{BL}. Then, similar to [61], C_1 and C_2 are discharged by I_R and I_L, respectively. During the discharge period, the dynamic comparator compares the currents and produces the response bit.

In TRNG mode, we utilize the dynamic entropy source of flash memory cells, namely the current fluctuations due to thermal and low frequency (flicker and random telegraph noise). We exploit this feature by reading the current from the same SL in two consecutive cycles. Indeed, after C_1 and C_2 are pre-charged to V_{BL}, I_R discharges C_1 and C_2 in two consecutive cycles and so does I_L. Hence, two bits are generated in four cycles. In the top-level design, TRNG mode is only activated in P_7, and the outputs (two bits) are XORed to generate the final output bit (T_F). This means that while utilizing the same silicon for both PUF and TRNG, the throughput does not change. In fact, since all primitive blocks in L_1 are operating in parallel and each of them generates 8 bits per cycle (with a delay of t_d), the throughput is $\sim t_d/8$ for both PUF and TRNG security primitives.

4.1.2 Challenge–Response Pair Space

In a proposed time-multiplexed design, the throughput and energy efficiency are sacrificed to dramatically increase challenge–response pair (CRP) space, which, in turn, results in a very secure PUF. Indeed, in each primitive block of the design, five out of ten rows, five out of ten columns, and two out of five SLs are selected. This leads to $\#CRP_P = \binom{10}{5} \times \binom{10}{5} \times \binom{5}{2}$ possible selections per primitive block. Each primitive block of L_1 operates in eight cycles. Thus, for each primitive block, eight possible selections out of $\#CRP_P$ are chosen, which results in $\binom{\#CRP_P}{8}$ total number of combinations. The number of combinations is increased to $\binom{\#CRP_P}{8}^5$, since all five primitive blocks in L_1 equally contribute to the output and further multiplied by $\binom{5}{2}^2$, because 2 out of 5 bits of C_6 and C_7 are used for P_6 and P_7. Therefore, the maximum number of distinct CRPs is given by

$$\binom{\#CRP_P}{8}^5 \times \binom{5}{2}^2$$

Due to the exponential number of CRPs, it is impossible for adversaries to fully read out all of the CRPs even if they hold physical possession of the PUF.

4.2 Experimental Results

4.2.1 Design Prototype

We have fabricated 10 × 10 analog-grade flash memory circuits in standard Global Foundries (GF's) 55nm embedded CMOS process based on redesigned layout [62]. Each primitive block is individually programmed and measured on a custom-made printed circuit board using Keysight characterization tools. In fact, Keysight B1500A and B1530A tools and a custom made switch matrix were utilized for characterization, programming, and measurements. The fully integrated design occupies $1.3 \times 1.0mm^2$. It is dominated by low-voltage ($0.3mm^2$) and high-voltage ($0.1mm^2$) inputs/outputs and unused silicon ($\approx 0.9mm^2$). Active circuits, including programming circuitry ($4475\mu m^2$), flash memory array ($235\mu m^2$), registers ($19,250\mu m^2$),comparators ($150\mu m^2$), and logic ($110\mu m^2$) are very compact (total of $24,216\mu m^2$).

The tuning process is explained in detail in our previous work [61]. Figure 4.3a shows an example of a current map after tuning the array with 10% accuracy to the randomly generated distribution with $\mu = 500nA$ and $\sigma = 150nA$. Due to reconfigurability of our approach, a completely different map, i.e., a new fingerprint, is obtained after re-tuning the same physical array to a new distribution with $\mu = 7.5\mu A$ and $\sigma = 1.5\mu A$ (Figure 4.3b). Figure 4.3c and 4.3d shows, respectively, the measured read-out current distribution (I_R and I_L) and their difference for the PUF instance (with $V_{WL,SEL} = 1.25V$, $V_{WL,US} = 1.35V$, $V_{CG,SEL} = 0.3V$, and $V_{SL} = 0.1V$) corresponding to Figure 4.3a. Figure 4.3c shows the distribution of I_L and I_R for 3000 random CRPs, highlighting a very symmetric current distribution which is achieved due to the analog-grade reconfigurability of the devices. To obtain these results, the specific sets of rows and columns of P1 were selected based on the applied CRP in accordance with the described procedure in Section 4.1-B. The measured I_L and I_R correspond to the sum of the currents through all of the devices in the selected left and right columns, respectively. The similar shapes of distributions indicate that there is no explicit bias in the output. The corresponding uniformity is 52.6%, which is very close to the ideal, 50% value. The lack of bias is also confirmed by the data in Figure 4.4, which shows that the output response is balanced with respect to the selected line in the array, i.e. value of '1' at certain position in the challenge bit-vector. Moreover, we have measured the response uniformity of 12 different primitive block using 4 different silicon chips. For each primitive, we employed the same tuning procedure (Gaussian distribution with 10% targeted accuracy) but with different common-mode current, using mean values randomly picked from $200nA$ to $5\mu A$ range. Also, we have studied sensitivity of uniformity metric to biasing condition. This was done by selecting appropriate V_{WL} from 0.65 V to 1.35 V, V_{SL} from 0.1 V to 0.5 V, and V_{CG} from 0.1 V to 0.5 V to match the selected common-mode currents for each instance. For each block, 4K randomly selected challenges have been applied and the response was measured at room temperature. The experimental results (Figure 4.5) show again close to 50% uniformity for majority of the considered instances.

Furthermore, a StrongArm current comparator, with 10nA sensitivity in 55nm CMOS, is used to compare the current of SLs and then generate the output bits in each primitive block (similar to [45]). The main benefit of the current comparator is that the differential scheme is more noise immune and has a higher power supply rejection ratio in comparison with single-ended designs.

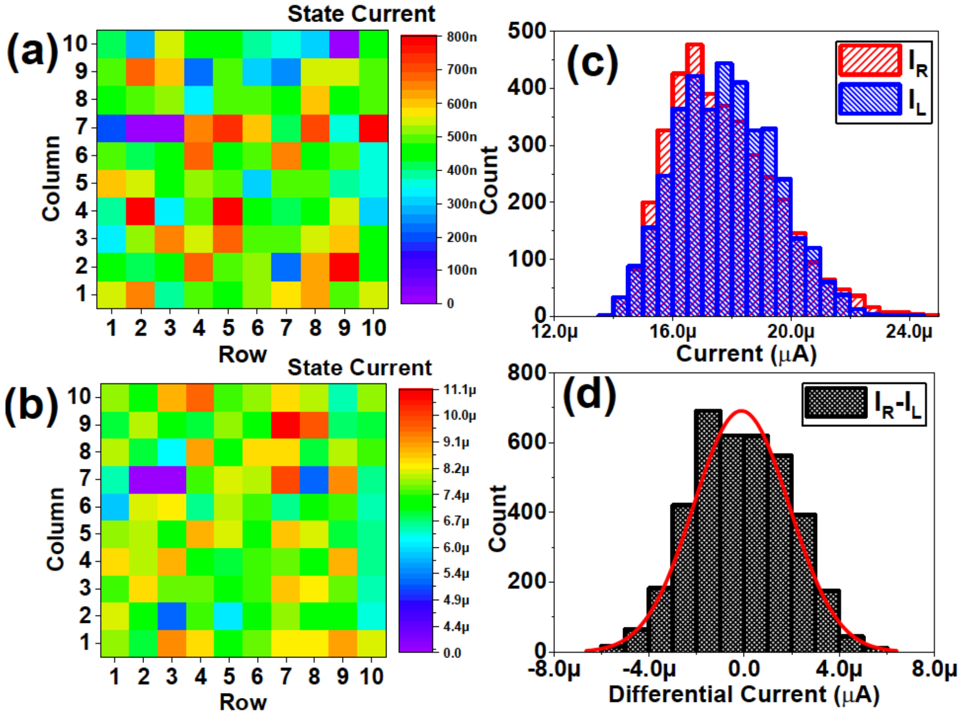


Figure 4.3: Measurements of one of the primitive blocks (P1) at room temperature: (a, b) two examples of resultant map of conductance states in 10 × 10 array of cells, (c) the distribution of read-out currents for 3000 cases, and (d) the corresponding distribution of differential current.

4.2.2 Functionality and Security Metrics

As TRNG security primitive is unified with the existing PUF security primitive, we need to make sure that PUF and TRNG output bits (R_F and T_F) are not correlated. For this purpose, the Pearson correlation coefficient between PUF and TRNG output bits is calculated. The Pearson correlation coefficient is a number between -1 and 1 that shows the extent to which two variables are linearly related. The ideal value for two

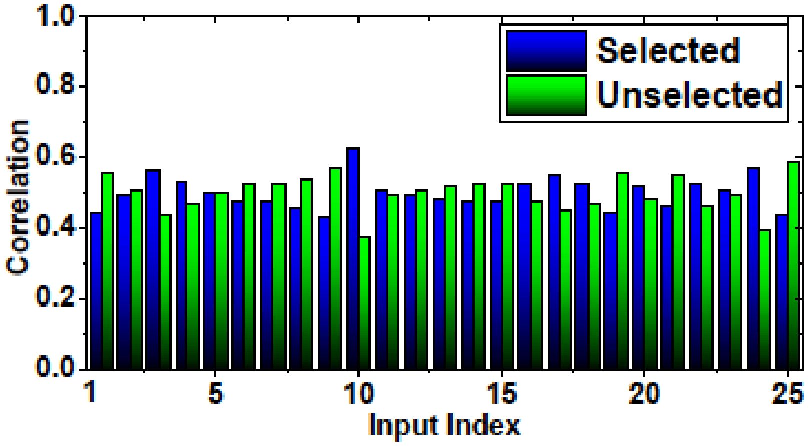


Figure 4.4: Measured correlation (fraction of '1's in the response when particular bit at the input is selected) based on 4K random challenge-response pairs.

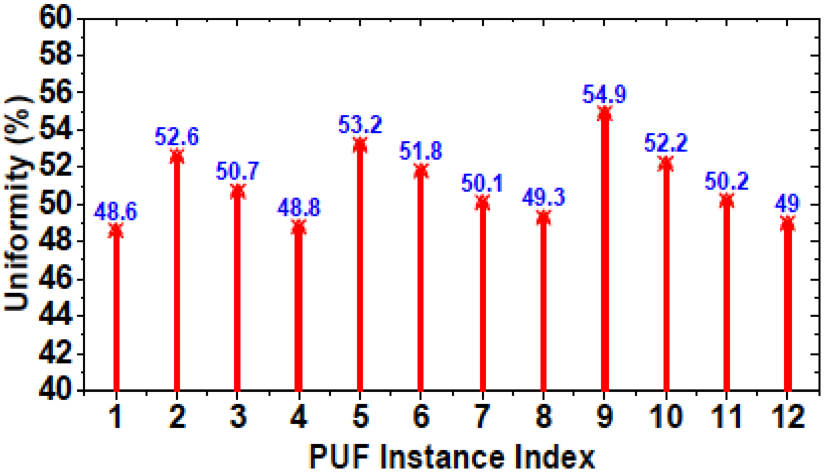


Figure 4.5: Response uniformity of 12 different primitive blocks obtained from reprogramming 4 different chips.

uncorrelated variables is 0. The Pearson correlation coefficient of 0.003 calculated based on 55K bits of PUF and TRNG indicates a negligible correlation in the responses.

In order to measure the uncertainty of PUF and TRNG output bits, Shannon entropy is calculated. The Shannon entropy for events with two possible outcomes is a number between 0 and 1. The Shannon entropy reaches its maximum when the outcome can be either of the two possible values with 0.5 probability. In the proposed architecture, Shannon entropy is 0.99958 and 0.99998 on the same data set for PUF and TRNG responses, respectively. Such almost ideal values of Shannon entropy show that the PUF and TRNG output bits have the maximum uncertainty.

To further study the cryptographic quality of PUF output bits, the uniformity of HSR bits and the fractional Hamming weight distributions of R_1, R_2, and R_F were calculated. Specifically, Figure 4.6a shows that the measured uniformity of HSR bits is near ideal for $P_{1,2,4}$ blocks, though there is visible bias in $P_{3,5}$ responses. Despite that, the differential current distribution of $P_{6,7}$ looks symmetrical as shown in Figure 4.6b for P_6. (Here, P_6 was tuned using 500 nA average state current and operated at $V_{WL,SEL} = 0.85V$, $V_{WL,US} = 0.9V$, $V_{CG,SEL} = 0.3V$, and $V_{SL} = 0.3V$.) Interestingly, the measured correlations (Figure 4.7), based on 100K challenge response pairs, are much weaker, as compared to those for single primitive block. The randomness in the output response is also highlighted by 2D visual representation of 1K randomly selected 128-bit keys (Figure 4.8). Furthermore, the fractional Hamming weight distributions of R_1, R_2, and R_F were calculated based on 5K randomly selected 64- and 128-bit responses. The normalized average Hamming weights were very close to ideal $\sim 50\%$ values (Figure 4.9a) [61]. Figure 4.9b shows near optimal results for diffuseness - the other important metric that evaluates the difference (Hamming distance) between unique keys generated by the same PUF under different challenges [63]. Similar to Hamming weight distribution, the average of the fractional Hamming distance of 1K randomly selected 64- and 128-bit keys is very close to ideal ∼ 50% value (Figure 4.9b).

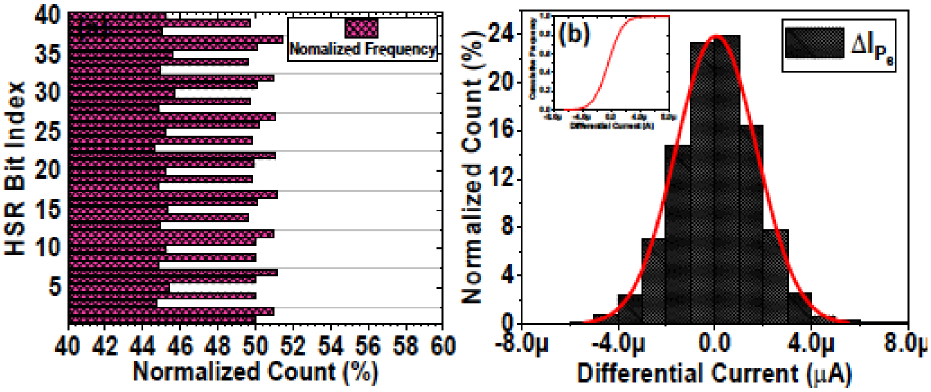


Figure 4.6: : (a) Normalized Hamming weight of HSR bits over 100K applied challenges. (b) Differential current distribution of P6, with the inset showing the corresponding CDF.

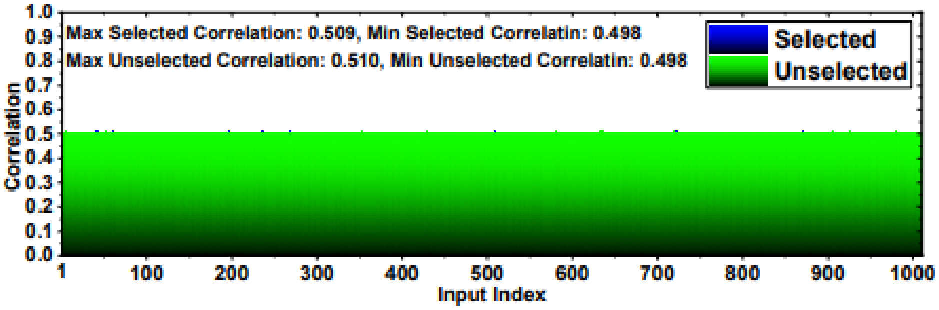


Figure 4.7: Measured correlation based on 100K random challenge-response pairs.

Figure 4.10 shows the probability of producing a ‘1’ response, i.e., the normalized number of ‘1’ s in the PUF outputs, as a function of the challenges for which the specified index of bit of the 1010-bit challenge is fixed to ‘1’, while allowing any values on the remaining bits of the challenge. As demonstrated, the response bit can be ‘0’ or ‘1’ with equal probability, irrespective of the specific bit of the input being set to ‘1’. This means that the PUF response bit is not correlated with any of the inputs. Hence, it is hard, if not impossible, for the attacker to extract any information from the system by only measuring or modeling the part of the system. It is worth mentioning that an undesired

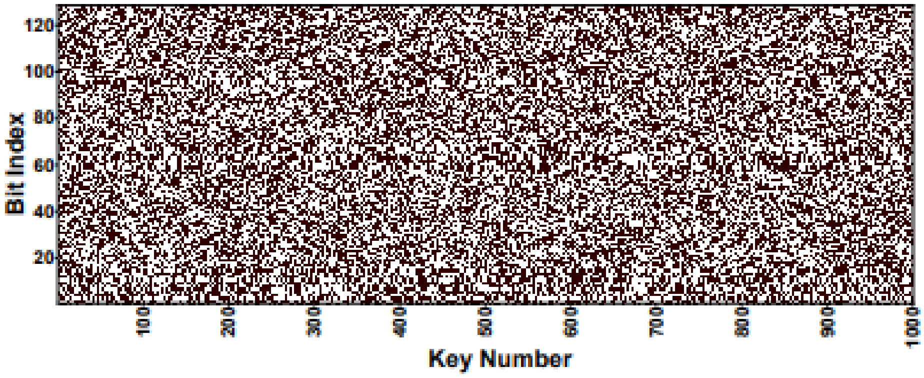


Figure 4.8: 2D representation of 1K 128-bit keys (black='1').

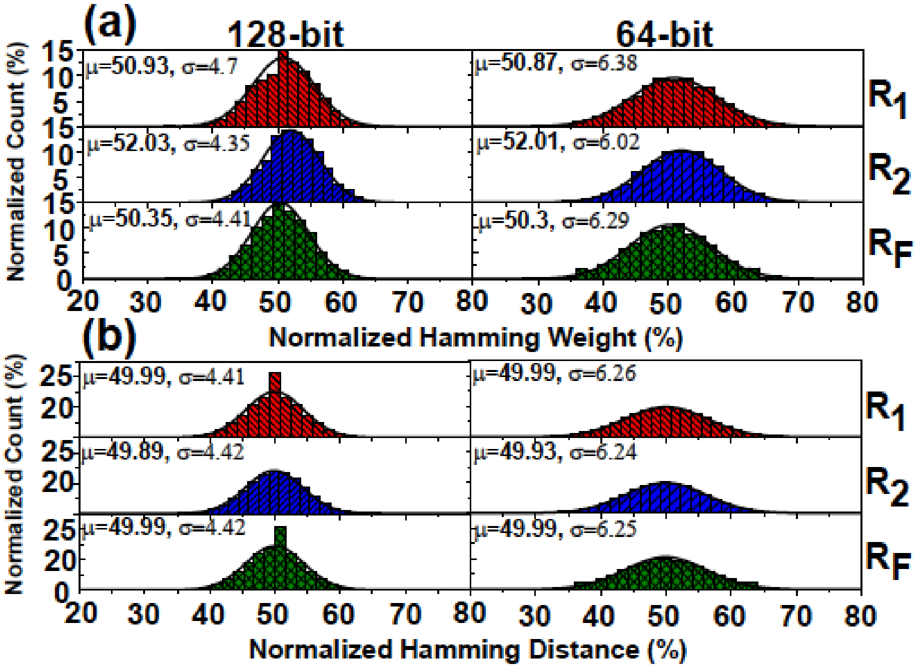


Figure 4.9: (a) Fractional Hamming weight and (b) fractional Hamming distance distribution of R1, R2, and RF. The results were computed based on (a) 5K and (b) 1K randomly generated 64-bit and 128-bit keys. All of the distributions have almost ideal mean value of 50%.

stuck-at-fault device in an eFlash array can bias the PUF response and make the circuit potentially vulnerable to probing attacks. However, the impact of such devices, if any, is significantly reduced by having eight 10 × 10 eFlash arrays, as compared to one large array.)

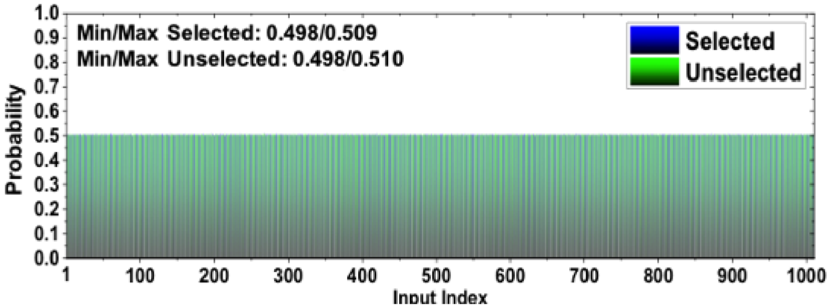


Figure 4.10: Output bit probability, based on 100K measured responses, for having a ‘1’ PUF response as a function of the specific set of challenges for which the specified index of the input bit is fixed to ‘1’. The results indicate that there is no bias in the output response.

To further study the cryptographic quality of TRNG output bits, the speckle pattern is represented along with the probability mass function of TRNG response bit, auto-correlation of TRNG response bits, and National Institute of Standards and Technology (NIST) test results. In particular, the randomness of TRNG is qualitatively demonstrated using a speckle pattern shown for 256 randomly selected 128-bit keys, with black and white pixels representing ‘1’ and ‘0’, respectively (Figure 4.11). The random distribution of black and white pixels is in correspondence with the cryptographic quality of TRNG output bits.

The probability mass function is calculated for 58K TRNG bits. The result shows that the TRNG output is ‘0’ or ‘1’ with a probability of 0.5. This uniform distribution of probability mass function demonstrates that TRNG output is an ideal random number [64]. The auto-correlation for TRNG outputs is calculated for the 10K-bit-long sequences within an experimentally measured stream of 55K bits. (The auto-correlation identifies

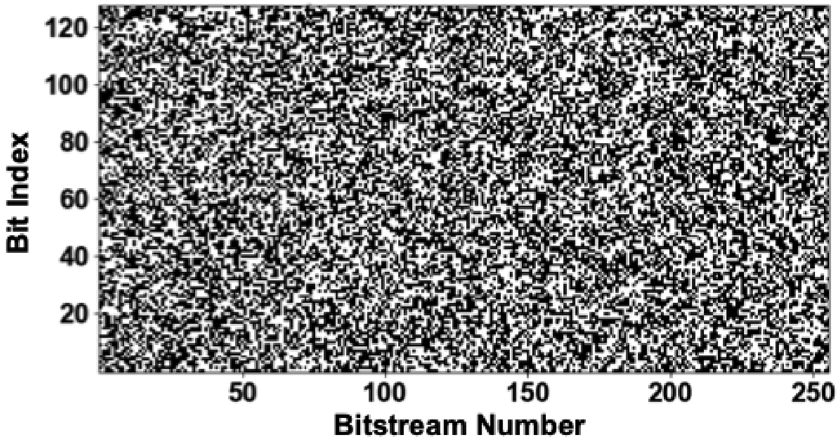


Figure 4.11: Speckle pattern of 256 128-bit keys (black = '1', white = '0') generated by the TRNG engine.

the non-randomness in a sequence and is calculated between a sequence and its lagged version [64]. In an ideal case, the auto-correlation of a random sequence should be zero for all nonzero lag values and should have a spike at a lag of 0.) The auto-correlation results again show high-quality, almost ideal randomness 4.12.

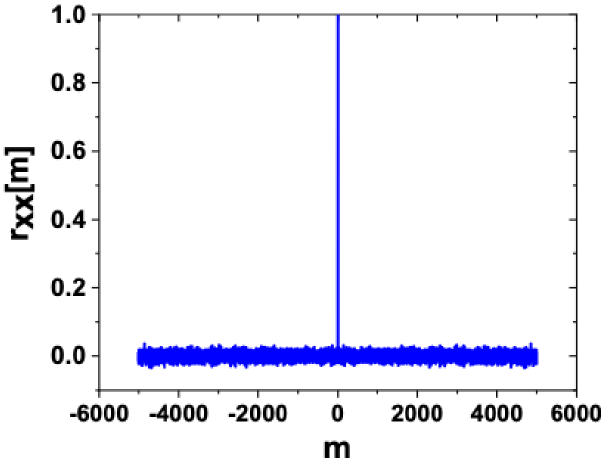
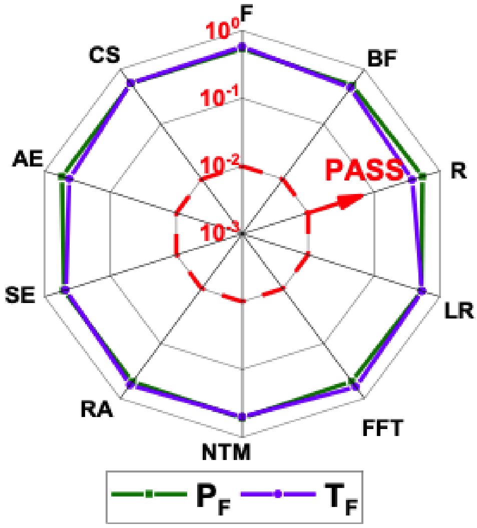


Figure 4.12: Auto-correlation of TRNG response and its lagged version.

To evaluate the statistical properties of the entropy generator, the NIST test suite is conducted over 175K measured PUF data and 56K data generated by the TRNG engine. As shown in Figure 4.13, both PUF and TRNG responses pass all relevant NIST tests with an average of 0.53 and 0.51 p-values, respectively. The results are very promising when compared with previous works [61].



NIST Test	PUF	TRNG
Frequency (F)	0.54	0.58
Block Frequency (BF)	0.51	0.46
Runs (R)	0.55	0.38
Longest Run (LR)	0.53	0.54
FFT	0.49	0.62
Non Overlapping Template (NTM) (m=5)	0.52	0.50
Rank (RA)	0.50	0.57
Serial (SER) (m=4)	0.51	0.48
Approximate Entropy (AE) (m=4)	0.55	0.42
Cumulative Sums (CS)	0.55	0.56

Figure 4.13: NIST randomness test results for both PUF (based on 87 2K-bit-long bitstreams) and TRNG (based on 28 2K-bit-long bitstreams). Average p-values and parameters used are shown in the table.

4.2.3 Reliability

In order to explore the reliability of the PUF security block in the designed architecture, an experiment was performed using 5 different block instances with specified current-mode currents. Each primitive block was characterized by measuring responses to 1K challenges at different ambient temperatures (with $\pm 5°C$ accuracy) and nominal voltage deviations. The dependence of BER on the utilized common-mode current is shown in Figure 4.14 ([61]). The results show that increasing temperature above the nominal $25°C$, at which devices were tuned, results in a semi-quadratic increase of BER, while the reliability is always improved by operating at higher bias currents. This is most likely due to the weaker temperature dependency at larger subthreshold currents. Indeed, the currents are almost independent of the temperature in strong inversion (which can be used to build a temperature insensitive current-reference [65]b). Therefore, there is a clear trade-off between power consumption and BER, and, e.g., the desired operating point could be determined based on the power budget and BER requirements of the PUF application. The same trend in BER is also observed with respect to the variations on the biased SL voltage (Figure 4.14b), though the dependence is weaker.

As results of Figure 4.14 showed, we can tune the devices to achieve a certain common-mode current and subsequently, a desirable BER. Based on Figure 4.14, the native BER of 5% can be achieved with $\approx 30\mu A$ SPICE simulations of the proposed design, including peripheral circuitry, show that the energy efficiency is 0.58 pJ/bit, with 88% / 12% contributed by array / comparators. Several previously proposed post-processing and error correction methods [66] can be utilized in our design to further improve reliability.

Furthermore, the aging measurements for a single PUF primitive block were performed by baking at $\sim$ 80°C. At every measurement step, after baking, the chip was cooled to 25°C to obtain 1kB of data. Figure 4.15 shows that the observed PUF response is stable and reliable in long term even after significant heating. The average uniformity remains very close to 50%, and error is below 3% after baking for a cumulative period of 15h. This stems from the differential nature of the circuit and optimized retention characteristics of analog-grade eFlash memories. Unlike the previous work [67], [15], the implemented design does not need an additional circuitry to maintain uniformity.

4.2.4 Machine Learning Modeling Attack

We have investigated modeling attacks based on several machine learning models, including multilayer perceptron (MLP), long short-term memory (LSTM), and typical online packages used in the previous works for studying PUF robustness. Specifically, in

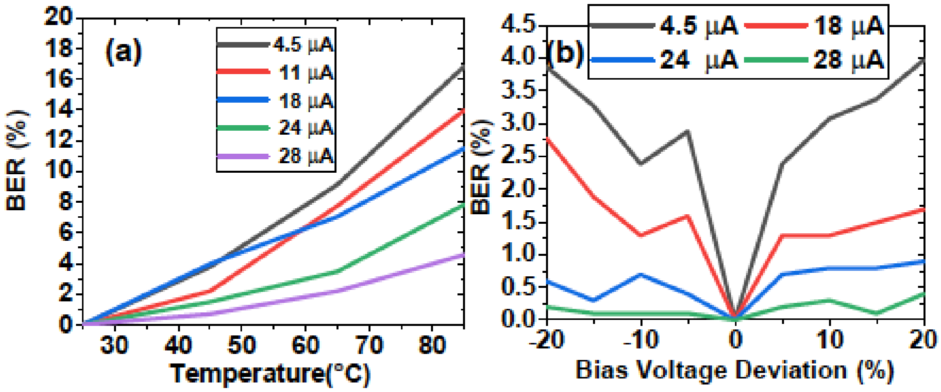


Figure 4.14: Measured BER as function of (a) temperature at nominal SL readout voltage for several common-mode readout currents and (b) bias voltage deviation for different common-mode currents at room temperature.

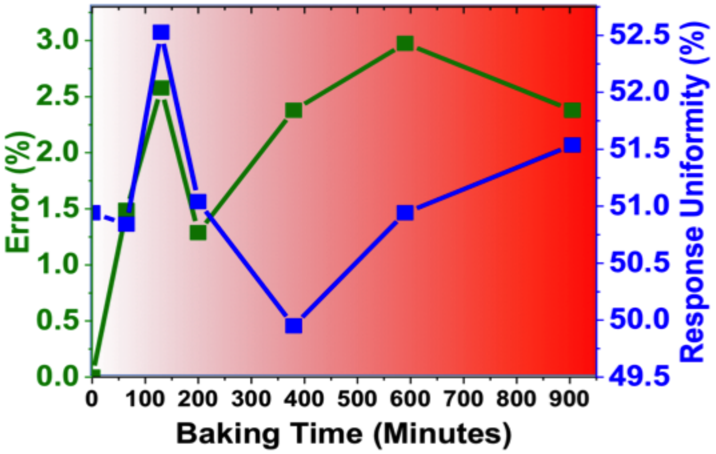


Figure 4.15: Accelerated aging test to verify the impact of the ambient temperature and the drift of the states over time on BER and the response uniformity. The results demonstrate the desired long-term reliability for PUF.

the first modeling attack study, a $1010 \times 100 \times 100 \times 100 \times 1$ MLP network with a rectified linear activation function in hidden layers and a sigmoid function in the output layer and RMSprop optimizer with a manually found 0.001 learning rate is used to model PUF response. The classifier was trained and validated with a specific subset of the observed CRPs and then tested on another mutually exclusive data. Specifically, the measured 160k of CRPs are divided into three groups—64% of the data are used for training, 16% for validation, and 20% for testing. Figure 4.16 shows the accuracy for predicting correct PUF response, when benchmarked on mutually exclusive (with training and validation) test data, as a function of the size of the data set utilized during training. The results indicate near ideal 50% accuracy even when using relatively large $>$ 100k measured training set.

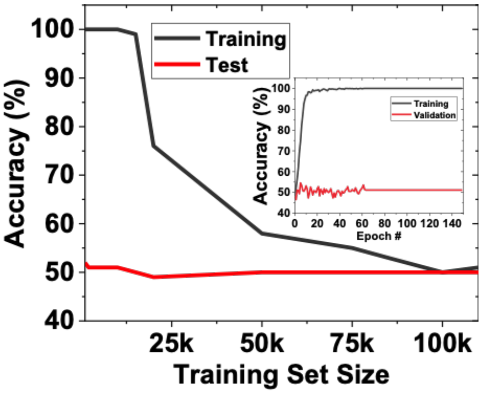


Figure 4.16: MLP modeling attack accuracy as a function of training data size. Training is performed with RMSdrop optimizer with the learning rate of 0.001. For all data points, the test set size is 20% of the data. The inset shows the prediction accuracy on the training and validation data sets over different 150 epochs.

In the second modeling study, we used the LIBLINEAR and LIBSVM open-source packages [64, 55] which support logistic regression and support vector machine algorithms. These open-source packages also resulted in close to 50% prediction accuracy when benchmarked on the test data.

Finally, an LSTM recurrent neural network was used to predict an output based on

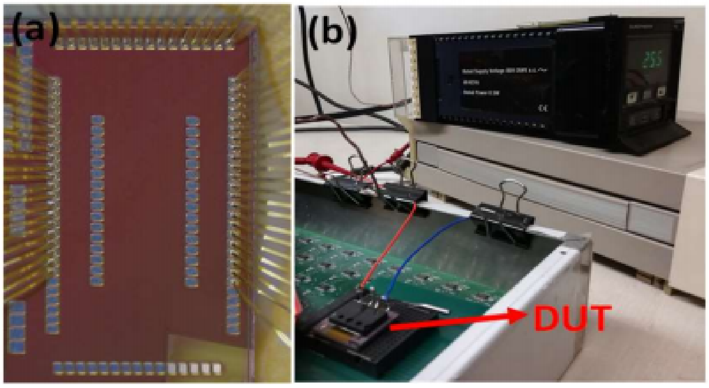


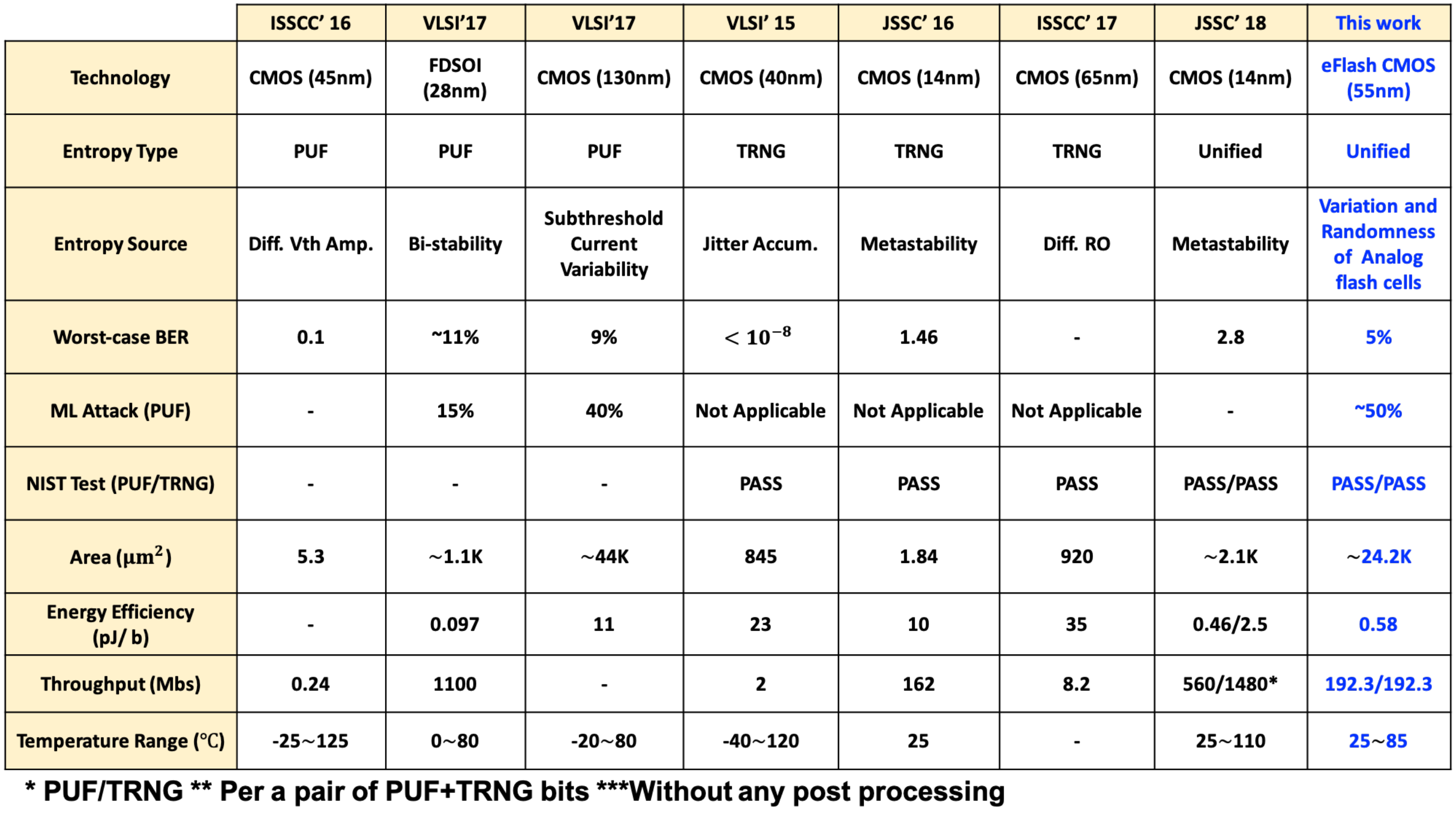
Figure 4.17: : (a) The die photo of a primitive block fabricated in GF's 55nm CMOS process and (b) the measurement setup.

the memory of past inputs. The implemented network has two LSTM layers, each with 128 units and rectified linear unit (ReLU) as activation function. The extracted features are then fed into two dense (fully connected) layers with sigmoid and softmax functions, respectively. The network uses an RMSprop optimizer for the binary classification task. To prepare LSTM inputs and outputs, we used a similar approach to that described in [68]. Specifically, N adjacent bits within response sequence are used as one input, whereas the immediate bit after the input bit sequence is used as the output. Then, the response sequence is shifted by 3-bit positions and is used as another input. Similarly, the immediate bit after the new N bits of the input is used as the new output. The shifting process continues until all of the LSTM input sequences and outputs are generated. After preparing the input sequences and outputs, the machine learning predictability of input sequences with length N = 32, 64, and 128 is studied for 10K-bit-long experimentally measured sequences. In all cases, the predictability is approximately 50%. This means that PUF response is either '0' or '1' with the probability of 50% regardless of the previous PUF response values. In other words, even if the history of PUF responses is given to attackers, they cannot predict the next PUF response.

4.3 Discussion and Summary

Table 4.1 shows the comparison between the presented design and state-of-the-art works. In addition to better quantative performance, the design is reconfigurable, and the features unified the implementation of PUF and TRNG, which is robust against machine learning attacks, thus offering additional functionalities compared to other works [54]–[59]. We also proposed blocking and time-multiplexed architecture to expand the PUF capacity and mitigate the impact of defective devices. The time-multiplexed design has slightly lower throughput when compared to another unified design [52], which was implemented in 14 nm.

In summary, we proposed and experimentally demonstrated PUF and TRNG, which are two fundamental hardware primitives, on a shared silicon (the die photo and the measurement setup is shown in 4.17). Since the same number of bits is generated for TRNG and PUF at one step, the proposed integrated design is, especially, suitable for AES algorithm used in privacy-preserving mutual authentication protocol. The design takes an advantage of intrinsic thermal and low-frequency noise of the circuit to generate both the static entropy and dynamic entropy. Experimental results demonstrate $\sim 10^{211}$ key space (in only $24,216\mu m^2$ because of using a time-multiplexing technique), 0.58pJ/b energy efficiency for $< 5\%$ controllable BER at $\sim$ 80°C, and 192.3Mbps throughput. The proposed design offers average uniformity of 50.3% average diffusivity of 49.99% for PUF. Both PUF and TRNG pass random NIST tests. We used several machine learning models to attack the PUF and showed that the system is resilient toward machine learning attacks. The important future work includes a detailed analysis and hardening against side channel and fault attacks, especially those related to the tuning circuitry. The other important future work includes finding the optimum distribution for cell currents, during tuning and under nominal biasing conditions, is an important future work.



	ISSCC' 16	VLSI'17	VLSI'17	VLSI' 15	JSSC' 16	ISSCC' 17	JSSC' 18	This work
Technology	CMOS (45nm)	FDSOI (28nm)	CMOS (130nm)	CMOS (40nm)	CMOS (14nm)	CMOS (65nm)	CMOS (14nm)	eFlash CMOS (55nm)
Entropy Type	PUF	PUF	PUF	TRNG	TRNG	TRNG	Unified	Unified
Entropy Source	Diff. Vth Amp.	Bi-stability	Subthreshold Current Variability	Jitter Accum.	Metastability	Diff. RO	Metastability	Variation and Randomness of Analog flash cells
Worst-case BER	0.1	~11%	9%	$< 10^{-8}$	1.46	-	2.8	5%
ML Attack (PUF)	-	15%	40%	Not Applicable	Not Applicable	Not Applicable	-	~50%
NIST Test (PUF/TRNG)	-	-	-	PASS	PASS	PASS	PASS/PASS	PASS/PASS
Area (μm^2)	5.3	~1.1K	~44K	845	1.84	920	~2.1K	~24.2K
Energy Efficiency (pJ/ b)	-	0.097	11	23	10	35	0.46/2.5	0.58
Throughput (Mbs)	0.24	1100	-	2	162	8.2	560/1480*	192.3/192.3
Temperature Range (°C)	-25~125	0~80	-20~80	-40~120	25	-	25~110	25~85

* PUF/TRNG ** Per a pair of PUF+TRNG bits ***Without any post processing

Table 4.1: Comparison with Previous Work.

Chapter 5

Conclusion and Future Opportunities

The ever-increasing presence of network-enabled devices in our daily lives requires cryptographic building blocks more than ever. Physically unclonable functions (PUFs) are a recently discovered class of cryptographic primitives that are suitable for a variety of security applications including including key generation and authentication. PUFs generate secure keys on the fly (rather than explicitly storing any security-critical information) by utilizing electronic devices that entail inherent sources of randomness.

Chapter 2 is reserved mainly for preliminaries information on PUFs. After defining physically unclonable function, we discuss the PUF main types, applications, and cryptographic metrics. Then, we discuss how TRNGs are associated with PUFs as well as main metrics by which the quality of a TRNG can be ascertained. Furthermore, we discuss the prior work on PUFs.

In Chapter 3, we propose two techniques to increase memristive strong PUF robustness against machine learning attacks. In both of the proposed techniques, we maximize the contribution of each crosspoint device to the PUF output. In the first approach, we choose an optimal ratio of selected rows and selected columns. In the second approach, we balance the device conductances in the crossbar array by either enforcing very similar conductances of all crossbar devices or by using the proposed balancing heuristic. The simulation results confirm that the robustness to machine learning attacks of memristive PUF is significantly improved by balancing conductances in the crossbar circuit. Then, we show that nonlinear analog tunable PUFs outperform the PUFs that have either linear or digital devices. Moreover, we investigate the effect of crossbar array size on the PUF robustness. Finally, we study the effect of stuck-at fault devices on PUFs. In fact, by modeling the hardware imperfection, we show that the balancing heuristic is effective to improve the effect of non-ideal yield.

In Chapter 4, we expand our previous work about flash-memory-based strong PUF by integrating the TRNG functionality on the same silicon. The proposed integrated design is especially suitable in privacy-preserving mutual authentication protocol because the same number of bits is generated for both TRNG and PUF. The design takes an advantage of intrinsic thermal noise and low-frequency noise of the circuit to generate the static entropy and the dynamic entropy, respectively. Experimental results demonstrate 10^{211} challenge-response pair space, 0.58 pJ/b energy efficiency for $< 5\%$ controllable BER at $\sim 80°$C, and 192.3 Mbps throughput. Moreover, the accelerated aging measurements indicate stable PUF response after 900 minutes of baking at 85°C. Both of the PUF and TRNG pass all relevant NIST randomness tests and are resilient against machine learning attacks.

Future Opportunities. Although a significant amount of research has been conducted on PUFs in the past few years, several open problems still exist. Here, we summarize the opportunities for PUF designs based on emerging nano-electronic devices.

1. Standard Models: There are no industry standard models for the memristors,

eFlash memories, and many other nano-electronic devices. Therefore, the simulation results might not reflect all of their features and we cannot fully compare the simulation results of different technologies. Furthermore, the lack of the temperature and other dependencies in the current device models make some reliability results speculative.

2. Complexity: In order to increase the PUF complexity, we can try different approaches that are not fully explored yet. For example, we can run several crossbars in parallel and then merge the output with XOR or majority voter. As an another example, we can use the cascade of PUFs to propagate the complexity from the first layer to the last one.

3. Hardware Imperfection Countermeasures: When PUFs are physically implemented, they suffer from different hardware imperfections such as line resistances. These problems can result in bias in the output which makes PUF predictable. Coming up with countermeasures that prevent such biases is crucial in designing robust PUFs.

4. Hybrid Attacks: So far, either the machine learning modeling attacks or side-channel attacks are studied in most of the articles. A detailed analysis on hybrid attacks which utilizes a combination of machine learning algorithms and PUF circuit characteristics (such as power consumption) is needed to further evaluate the robustness of different PUFs.

5. Design Trade-Offs: The investigation of the reliability/robustness/other metrics trade-off is an important research direction which can optimize PUF architecture based on a specific application.

6. Application Specification: Each PUF-based security application requires specific

features. An extensive industry-based survey on specifications of each application can be very informative in designing application-specific PUFs.

By conducting research on the mentioned areas, we can substantially improve the existing PUFs or coming up with new secure strong PUFs which are well-aligned with industry security applications.

Appendix A

Simulation Setup

To optimize and automate the simulation process for the work presented in Chapter 3, about 10000 lines of code is written. Here, we briefly review the main modules (Figure A.1) of the written software.

Module 1. In this module, challenge bits are generated based on the number of selected rows and columns.

Module 2. In this module, memristor parameters are generated based on specific distribution and device features. Then, the balancing algorithm is applied if the crossbar is needed to be balanced.

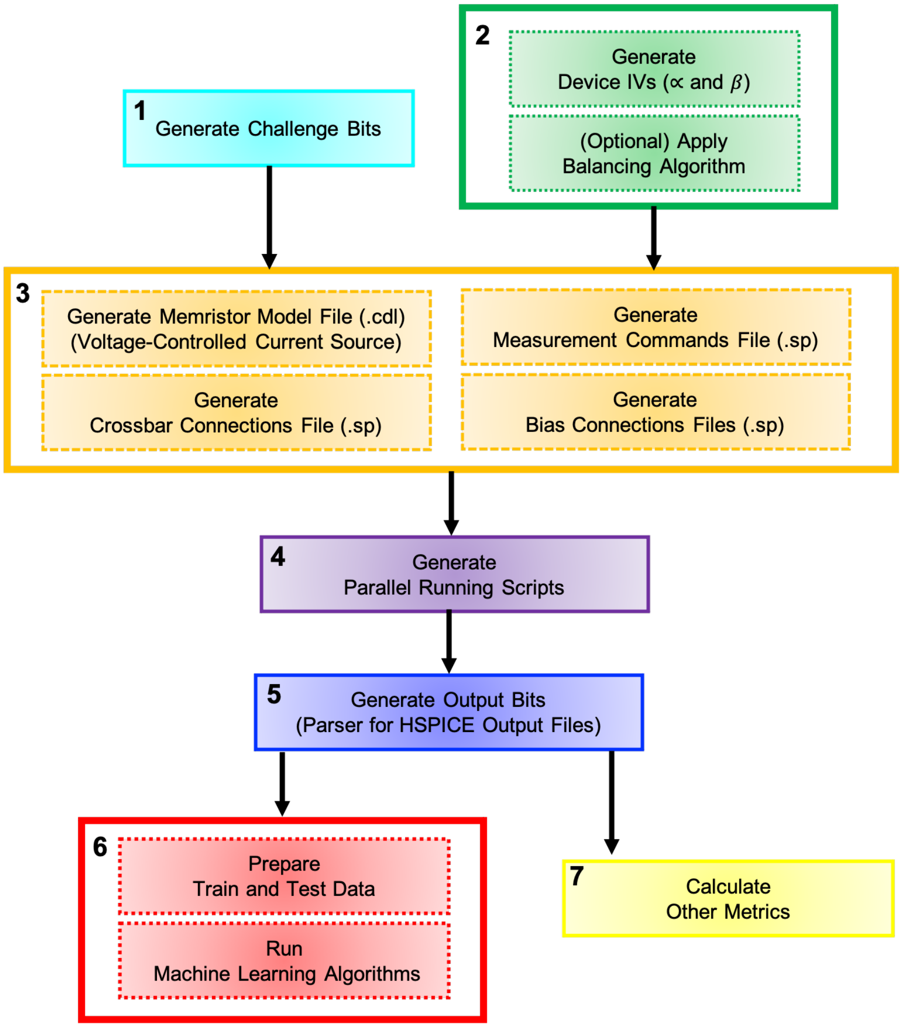
Module 3. In this module, HSPICE simulation files are generated which include the memristor model, crossbar connections, measurement commands, and the bias connections. While the first three files are fixed for a specific crossbar, the fourth file changes based on the input challenge.

Module 4. In this module, scripts are prepared, so that HSPICE files can be run in batch and in parallel. This step is substantially important for the simulation speedup specially when a huge number of CRPs and/or a big PUF is under study.

Module 5. In this module, the generated HSPICE output files are parsed and processed, so that the desired information (such as device current) is extracted. With this information, the output bit can be found.

Module 6. In this module, the CRPs are divided into train and test data sets. Then, the machine learning models are trained and tested.

Module 7. In this module, other PUF related metrics such as uniformity or correlation are calculated.



1
Generate Challenge Bits
2
Generate Device IVs (∝ and β)
(Optional) Apply Balancing Algorithm
3
Generate Memristor Model File (.cdl) (Voltage-Controlled Current Source)
Generate Measurement Commands File (.sp)
Generate Crossbar Connections File (.sp)
Generate Bias Connections Files (.sp)
4
Generate Parallel Running Scripts
5
Generate Output Bits (Parser for HSPICE Output Files)
6
Prepare Train and Test Data
Run Machine Learning Algorithms
7
Calculate Other Metrics

Figure A.1: Software Modules.

Bibliography

[1] C. Herder, M. M. Yu, F. Koushanfar, and S. Devadas, *Physical unclonable functions and applications: A tutorial*, *Proc. IEEE* **102** (2014), no. 8 1126–1141.

[2] U. Rührmair and D. E. Holcomb, *Pufs at a glance*, in *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014* (G. P. Fettweis and W. Nebel, eds.), pp. 1–6, European Design and Automation Association, 2014.

[3] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, *Modeling attacks on physical unclonable functions*, in *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 237–249, 2010.

[4] Y. Pang, B. Gao, B. Lin, H. Qian, and H. Wu, *Memristors for hardware security applications*, *Advanced Electronic Materials* **5** (2019), no. 9 1800872.

[5] M. Rostami, J. B. Wendt, M. Potkonjak, and F. Koushanfar, *Quo vadis, puf?: Trends and challenges of emerging physical-disorder based security*, in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2014.

[6] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, *The missing memristor found*, *nature* **453** (2008), no. 7191 80–83.

[7] D. C. Guterman, I. H. Rimawi, T.-L. Chiu, R. D. Halvorson, and D. McElroy, *An electrically alterable nonvolatile memory cell using a floating-gate structure*, *IEEE Transactions on Electron Devices* **26** (1979), no. 4 576–586.

[8] M. R. Mahmoodi, D. B. Strukov, and O. Kavehei, *Experimental demonstrations of security primitives with nonvolatile memories*, *IEEE Transactions on Electron Devices* **66** (2019), no. 12 5050–5059.

[9] Y. Gao, D. C. Ranasinghe, S. F. Al-Sarawi, O. Kavehei, and D. Abbott, *Emerging physical unclonable functions with nanotechnology*, *IEEE Access* **4** (2016) 61–80.

[10] Y. Gao, S. F. Al-Sarawi, and D. Abbott, *Physical unclonable functions*, *Nature Electronics* **3** (2020), no. 2 81–91.

[11] C. Herder, *Towards security without secrets.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2016.

[12] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, *A puf taxonomy*, *Applied Physics Reviews* **6** (2019), no. 1 011303.

[13] J. Kim, T. Ahmed, H. Nili, J. Yang, D. S. Jeong, P. Beckett, S. Sriram, D. C. Ranasinghe, and O. Kavehei, *A physical unclonable function with redox-based nanoionic resistive memory*, *IEEE Transactions on Information Forensics and Security* **13** (2017), no. 2 437–448.

[14] R. Maes, *Physically Unclonable Functions: Constructions, Properties and Applications (Fysisch onkloonbare functies: constructies, eigenschappen en toepassingen).* PhD thesis, Katholieke Universiteit Leuven, Belgium, 2012.

[15] K. Yang, D. T. Blaauw, and D. Sylvester, *Hardware designs for security in ultra-low-power iot systems: An overview and survey*, *IEEE Micro* **37** (2017), no. 6 72–89.

[16] A. V. Herrewege, *Lightweight PUF-based key and random number generation.* PhD thesis, Katholieke Universiteit Leuven, Belgium, 2015.

[17] A. Maiti, V. Gunreddy, and P. Schaumont, *A systematic method to evaluate and compare the performance of physical unclonable functions*, in *Embedded systems design with FPGAs*, pp. 245–267. Springer, 2013.

[18] M. Uddin, M. B. Majumder, K. Beckmann, H. Manem, Z. Alamgir, N. C. Cady, and G. S. Rose, *Design considerations for memristive crossbar physical unclonable functions*, *ACM Journal on Emerging Technologies in Computing Systems (JETC)* **14** (2017), no. 1 1–23.

[19] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, tech. rep., Booz-allen and hamilton inc mclean va, 2001.

[20] H. Nili, G. C. Adam, B. Hoskins, M. Prezioso, J. Kim, M. R. Mahmoodi, F. M. Bayat, O. Kavehei, and D. B. Strukov, *Hardware-intrinsic security primitives enabled by analogue state and nonlinear conductance variations in integrated memristors*, *Nature Electronics* **1** (2018), no. 3 197–202.

[21] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, *Physical one-way functions*, *Science* **297** (2002), no. 5589 2026–2030.

[22] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, *Silicon physical random functions*, in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 148–160, 2002.

[23] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, *Extracting secret keys from integrated circuits*, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **13** (2005), no. 10 1200–1205.

[24] U. Rührmair and M. van Dijk, *Pufs in security protocols: Attack models and security evaluations*, in *2013 IEEE symposium on security and privacy*, pp. 286–300, IEEE, 2013.

[25] G. E. Suh and S. Devadas, *Physical unclonable functions for device authentication and secret key generation*, in *2007 44th ACM/IEEE Design Automation Conference*, pp. 9–14, IEEE, 2007.

[26] B. Gassend, D. Lim, D. Clarke, M. Van Dijk, and S. Devadas, *Identification and authentication of integrated circuits*, *Concurrency and Computation: Practice and Experience* **16** (2004), no. 11 1077–1098.

[27] A. Maiti and P. Schaumont, *Improving the quality of a physical unclonable function using configurable ring oscillators*, in *2009 International Conference on Field Programmable Logic and Applications*, pp. 703–707, IEEE, 2009.

[28] M. Gao, K. Lai, and G. Qu, *A highly flexible ring oscillator puf*, in *Proceedings of the 51st Annual Design Automation Conference*, pp. 1–6, 2014.

[29] J.-L. Zhang, G. Qu, Y.-Q. Lv, and Q. Zhou, *A survey on silicon pufs and recent advances in ring oscillator pufs*, *Journal of computer science and technology* **29** (2014), no. 4 664–678.

[30] D. E. Holcomb, W. P. Burleson, and K. Fu, *Power-up sram state as an identifying fingerprint and source of true random numbers*, *IEEE Transactions on Computers* **58** (2008), no. 9 1198–1210.

[31] D. E. Holcomb, W. P. Burleson, K. Fu, *et. al.*, *Initial sram state as a fingerprint and source of true random numbers for rfid tags*, in *Proceedings of the Conference on RFID Security*, vol. 7, p. 01, 2007.

[32] Y. Pang, H. Wu, B. Gao, R. Liu, S. Wang, S. Yu, A. Chen, and H. Qian, *Design and optimization of strong physical unclonable function (puf) based on rram array*, in *2017 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, pp. 1–2, IEEE, 2017.

[33] G. S. Lee, G.-H. Kim, K. Kwak, D. S. Jeong, and H. Ju, *Enhanced reconfigurable physical unclonable function based on stochastic nature of multilevel cell rram*, *IEEE Transactions on Electron Devices* **66** (2019), no. 4 1717–1721.

[34] Y. Yoshimoto, Y. Katoh, S. Ogasahara, Z. Wei, and K. Kouno, *A reram-based physically unclonable function with bit error rate¡ 0.5% after 10 years at 125° c for 40nm embedded application*, in *2016 IEEE Symposium on VLSI Technology*, pp. 1–2, IEEE, 2016.

[35] R. Liu, H. Wu, Y. Pang, H. Qian, and S. Yu, *Experimental characterization of physical unclonable function based on 1 kb resistive random access memory arrays*, *IEEE Electron Device Letters* **36** (2015), no. 12 1380–1383.

[36] Y. Pang, H. Wu, B. Gao, D. Wu, A. Chen, and H. Qian, *A novel puf against machine learning attack: Implementation on a 16 mb rram chip*, in *2017 IEEE International Electron Devices Meeting (IEDM)*, pp. 12–2, IEEE, 2017.

[37] D. Arumí, Á. Gómez-Pau, S. Manich, R. Rodríguez-Montañés, M. B. González, and F. Campabadal, *Unpredictable bits generation based on rram parallel configuration*, *IEEE Electron Device Letters* **40** (2018), no. 2 341–344.

[38] Y. Pang, B. Gao, D. Wu, S. Yi, Q. Liu, W.-H. Chen, T.-W. Chang, W.-E. Lin, X. Sun, S. Yu, *et. al.*, *25.2 a reconfigurable rram physically unclonable function utilizing post-process randomness source with¡ 6× 10- 6 native bit error rate*, in *2019 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 402–404, IEEE, 2019.

[39] R. Liu, H. Wu, Y. Pang, H. Qian, and S. Yu, *A highly reliable and tamper-resistant rram puf: Design and experimental validation*, in *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 13–18, IEEE, 2016.

[40] J. Yang, X. Li, T. Wang, X. Xue, Z. Hong, Y. Wang, D. W. Zhang, and H. Lu, *A physically unclonable function with ber¡ 0.35% for secure chip authentication using write speed variation of rram*, in *2018 48th European Solid-State Device Research Conference (ESSDERC)*, pp. 54–57, IEEE, 2018.

[41] M.-Y. Wu, T.-H. Yang, L.-C. Chen, C.-C. Lin, H.-C. Hu, F.-Y. Su, C.-M. Wang, J. P.-H. Huang, H.-M. Chen, C. C.-H. Lu, *et. al.*, *A puf scheme using competing oxide rupture with bit error rate approaching zero*, in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 130–132, IEEE, 2018.

[42] Y. Wang, W.-k. Yu, S. Wu, G. Malysa, G. E. Suh, and E. C. Kan, *Flash memory for ubiquitous hardware security functions: True random number generation and device fingerprints*, in *2012 IEEE Symposium on Security and Privacy*, pp. 33–47, IEEE, 2012.

[43] M.-S. Kim, D.-I. Moon, S.-K. Yoo, S.-H. Lee, and Y.-K. Choi, *Investigation of physically unclonable functions using flash memory for integrated circuit authentication*, *IEEE Transactions on Nanotechnology* **14** (2015), no. 2 384–389.

[44] G. C. Adam, H. Nili, J. Kim, B. D. Hoskins, O. Kavehei, and D. B. Strukov, *Utilizing iv non-linearity and analog state variations in reram-based security primitives*, in *2017 47th European Solid-State Device Research Conference (ESSDERC)*, pp. 74–77, IEEE, 2017.

[45] M. R. Mahmoodi, H. Nili, and D. B. Strukov, *Rx-puf: Low power, dense, reliable, and resilient physically unclonable functions based on analog passive rram crossbar arrays*, in *2018 IEEE Symposium on VLSI Technology*, pp. 99–100, IEEE, 2018.

[46] M. R. Mahmoodi, H. Nili, Z. Fahimi, S. Larimian, H. Kim, and D. Strukov, *Ultra-low power physical unclonable function with nonlinear fixed-resistance crossbar circuits*, in *2019 IEEE International Electron Devices Meeting (IEDM)*, pp. 30.1.1–30.1.4, 2019.

[47] X. Xi, A. Aysu, and M. Orshansky, *Fresh re-keying with strong pufs: A new approach to side-channel security*, in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 118–125, IEEE, 2018.

[48] G. T. Becker, R. Kumar, *et. al.*, *Active and passive side-channel attacks on delay based puf designs.*, *IACR Cryptol. ePrint Arch.* **2014** (2014) 287.

[49] X. Xu and W. Burleson, *Hybrid side-channel/machine-learning attacks on pufs: A new threat?*, in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, IEEE, 2014.

[50] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, F. Koushanfar, and W. P. Burleson, *Power and timing side channels for pufs and their efficient exploitation.*, *IACR Cryptol. ePrint Arch.* **2013** (2013) 851.

[51] A. Mahmoud, U. Rührmair, M. Majzoobi, and F. Koushanfar, *Combined modeling and side channel attacks on strong pufs.*, *IACR Cryptol. ePrint Arch.* **2013** (2013) 632.

[52] S. Satpathy, S. K. Mathew, R. Kumar, V. B. Suresh, M. A. Anders, H. Kaul, A. Agarwal, S. Hsu, R. K. Krishnamurthy, and V. De, *An all-digital unified physically unclonable function and true random number generator featuring self-calibrating hierarchical von neumann extraction in 14-nm tri-gate CMOS*, *IEEE J. Solid State Circuits* **54** (2019), no. 4 1074–1085.

[53] W. Che, M. T. Martin, G. Pocklassery, V. K. Kajuluri, F. Saqib, and J. Plusquellic, *A privacy-preserving, mutual puf-based authentication protocol*, *Cryptogr.* **1** (2017), no. 1 3.

[54] B. Karpinskyy, Y. Lee, Y. Choi, Y. Kim, M. Noh, and S. Lee, *8.7 physically unclonable function for secure key generation with a key error rate of 2e-38 in*

45nm smart-card chips, in *2016 IEEE International Solid-State Circuits Conference, ISSCC 2016, San Francisco, CA, USA, January 31 - February 4, 2016*, pp. 158–160, IEEE, 2016.

[55] S. Jeloka, K. Yang, M. Orshansky, D. Sylvester, and D. Blaauw, *A sequence dependent challenge-response puf using 28nm sram 6t bit cell*, in *2017 Symposium on VLSI Circuits*, pp. C270–C271, 2017.

[56] X. Xi, H. Zhuang, N. Sun, and M. Orshansky, *Strong subthreshold current array puf with 265 challenge-response pairs resilient to machine learning attacks in 130nm cmos*, in *2017 Symposium on VLSI Circuits*, pp. C268–C269, 2017.

[57] K. Yang, D. T. Blaauw, and D. Sylvester, *A robust -40 to 120°c all-digital true random number generator in 40nm CMOS*, in *Symposium on VLSI Circuits, VLSIC 2015, Kyoto, Japan, June 17-19, 2015*, p. 248, IEEE, 2015.

[58] B. Ray and A. Milenković, *True random number generation using read noise of flash memory cells*, *IEEE Transactions on Electron Devices* **65** (2018), no. 3 963–969.

[59] E. Kim, M. Lee, and J. Kim, *8.2 8mb/s 28mb/mj robust true-random-number generator in 65nm CMOS based on differential ring oscillator with feedback resistors*, in *2017 IEEE International Solid-State Circuits Conference, ISSCC 2017, San Francisco, CA, USA, February 5-9, 2017*, pp. 144–145, IEEE, 2017.

[60] S. K. Mathew, D. Johnston, S. Satpathy, V. B. Suresh, P. Newman, M. A. Anders, H. Kaul, A. Agarwal, S. Hsu, G. K. Chen, and R. K. Krishnamurthy, *μrng: A 300-950 mv, 323 gbps/w all-digital full-entropy true random number generator in 14 nm finfet CMOS*, *IEEE J. Solid State Circuits* **51** (2016), no. 7 1695–1704.

[61] M. R. Mahmoodi, H. Nili, S. Larimian, X. Guo, and D. B. Strukov, *Chipsecure: A reconfigurable analog eflash-based PUF with machine learning attack resiliency in 55nm CMOS*, in *Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019, Las Vegas, NV, USA, June 02-06, 2019*, p. 137, ACM, 2019.

[62] X. Guo, F. M. Bayat, M. Prezioso, Y. Chen, B. Nguyen, N. Do, and D. B. Strukov, *Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm nor flash memory cells*, in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–4, 2017.

[63] Y. Hori, T. Yoshida, T. Katashita, and A. Satoh, *Quantitative and statistical performance evaluation of arbiter physical unclonable functions on fpgas*, in *2010 International Conference on Reconfigurable Computing and FPGAs*, pp. 298–303, IEEE, 2010.

[64] J. J. M. Chan, P. Thulasiraman, G. Thomas, and R. Thulasiram, *Ensuring quality of random numbers from trng: Design and evaluation of post-processing using genetic algorithm*, *Journal of Computer and Communications* **4** (2016), no. 4 73–92.

[65] A. Bendali and Y. Audet, *A 1-v cmos current reference with temperature and process compensation*, *IEEE Transactions on Circuits and Systems I: Regular Papers* **54** (2007), no. 7 1424–1429.

[66] S. K. Mathew, S. K. Satpathy, M. A. Anders, H. Kaul, S. K. Hsu, A. Agarwal, G. K. Chen, R. J. Parker, R. K. Krishnamurthy, and V. De, *16.2 a 0.19 pj/b pvt-variation-tolerant hybrid physically unclonable function circuit for 100% stable secure key generation in 22nm cmos*, in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 278–279, IEEE, 2014.

[67] D. E. Holcomb, W. P. Burleson, and K. Fu, *Power-up SRAM state as an identifying fingerprint and source of true random numbers*, *IEEE Trans. Computers* **58** (2009), no. 9 1198–1210.

[68] C. Chang and C. Lin, *LIBSVM: A library for support vector machines*, *ACM Trans. Intell. Syst. Technol.* **2** (2011), no. 3 27:1–27:27.