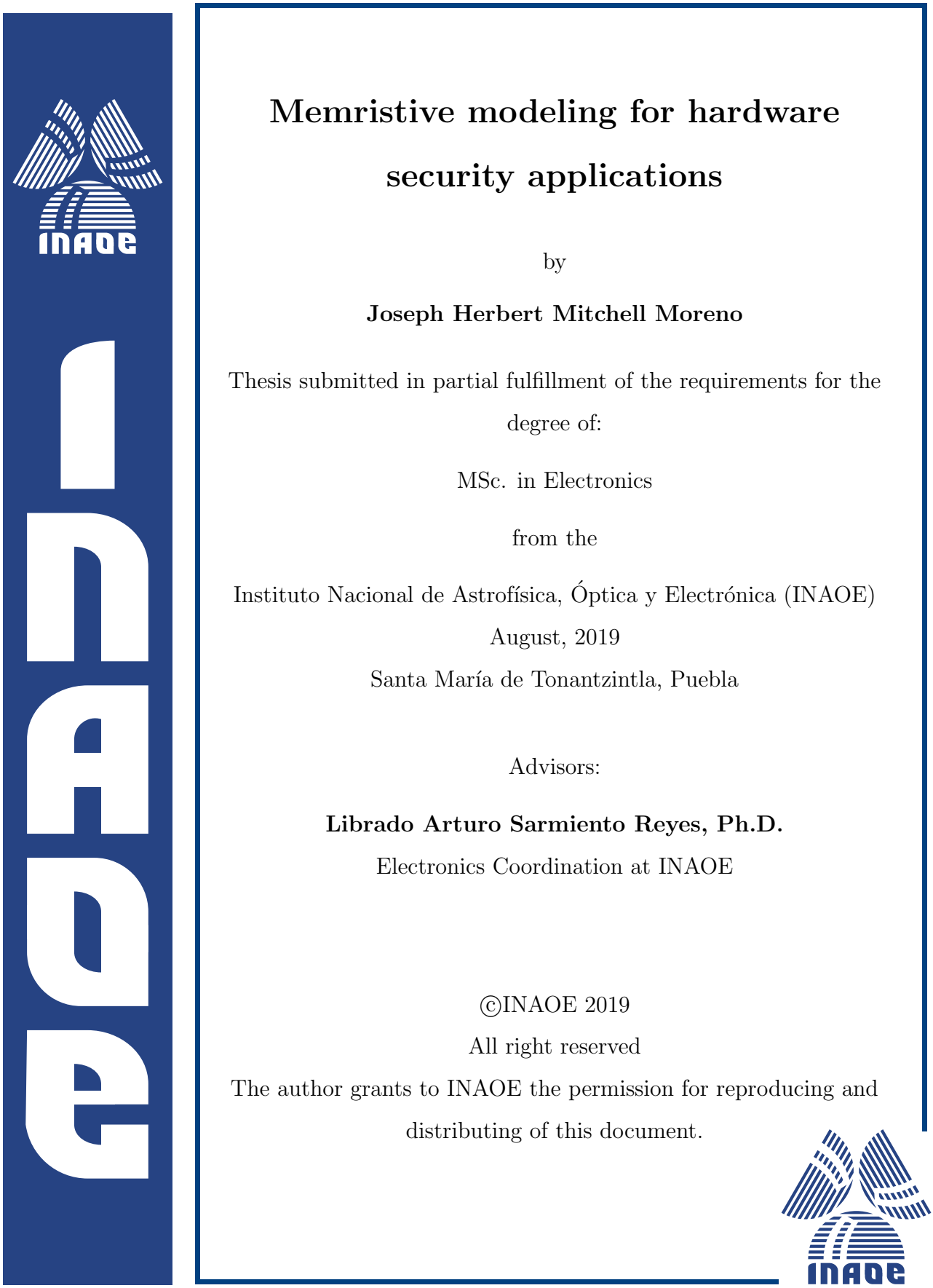


Memristive modeling for hardware security applications

by

Joseph Herbert Mitchell Moreno

Thesis submitted in partial fulfillment of the requirements for the degree of:

MSc. in Electronics

from the

Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)

August, 2019

Santa María de Tonantzintla, Puebla

Advisors:

Librado Arturo Sarmiento Reyes, Ph.D.

Electronics Coordination at INAOE

Acknowledgements

Quiero tomarme este espacio para agradecer a personas que considero fueron importantes durante la realización de esta tesis, pues de una u otra manera influenciaron en mi desarrollo a lo largo de estos dos años de maestría.

Al Dr. Arturo Sarmiento por la asesoría de esta tesis, por su ayuda, su constante apoyo y sus observaciones que siempre estuvieron direccionadas a hacer el mejor trabajo posible.

A mis sinodales, Dr. Roberto Murphy, Dr. Guillermo Espinosa y Dr. Luis Hernández por el tiempo dedicado en correcciones de la tesis y/o comentarios que sirvieron para mejorar la calidad del manuscrito.

A los profesores que me guiaron y me impartieron sus conocimientos, a todos mis amigos con los cuales compartí durante mi maestría, a mi novia Alejandra Chaparro que siempre me apoyó y me escuchó ante cualquier duda o adversidad, a mi familia que también siempre estuvo conmigo a pesar de la distancia dandome un apoyo enorme e incondicional. Y por último a todo el personal de dirección de formación académica porque siempre me instruyeron de forma correcta y amable para cualquier necesidad que tuve durante mi tiempo aquí.

A todos, muchas gracias.

Resumen

Con los recientes avances en el estudio del memristor como el cuarto elemento básico de la Teoría de Circuitos, se han abierto diversas posibilidades no sólo en los campos de fabricación y modelado del dispositivo, sino también en diversas aplicaciones de los llamados circuitos memristivos. En el grupo de trabajo de INAOE se han desarrollado diversos modelos analíticos que se enfocan a reproducir el comportamiento eléctrico del dispositivo a partir del mecanismo físico que describe el fenómeno de conmutación del mismo.

En este trabajo, se ha aplicado un modelo controlado por carga al desarrollo de circuitos para "Hardware Security" (HS*) bajo la premisa de que los parámetros del memristor permiten entonar sus características eléctricas y por tanto la respuesta de las celdas básicas en circuitería de HS. El enfoque ha sido incluir el memristor en funciones físicas no clonables (PUFs por sus siglas en inglés) basadas en osciladores de anillo. De allí, que el memristor actúa como el elemento principal que define la frecuencia de oscilación y por ende la dinámica completa del PUF.

La calidad de la respuesta del PUF se determina por medio de diversas métricas, como son: la unicidad, la uniformidad y el grado de enmascaramiento de bits. En este trabajo se han realizado diversos análisis que demuestran que los PUFs memristivos poseen métricas excelentes bajo diversas condiciones de complejidad de los sistemas.

Abstract

With the recent advances in the study of the memristor as the fourth fundamental element in Circuit Theory, a wide amount of opportunities have emerged not only in the fields of manufacturing and modeling of the device, but also in various applications of so-called memristive circuits. Several analytical models have been developed at INAOE's work group that focus on reproducing the electrical behaviour of the device based on the physical mechanism that describes its switching phenomena.

In this work, a charge-controlled model has been applied to the development of circuits for "Hardware Security" (HS*) under the premise that the parameters of the memristor allow to establish their electrical characteristics and therefore the response of the basic of circuit cells for HS. The approach focuses on including the memristor in physical unclonable functions (PUFs*) based on ring oscillators. Hence, the memristor acts as the main element that defines the oscillation frequency and therefore the complete dynamics of the PUF.

The performance of the PUF response is determined by using various metrics, such as: uniqueness, uniformity and bit aliasing. In this work, several analyzes have been carried out in order to demonstrate that the memristive PUFs have excellent metrics under diverse conditions of complexity of the systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the recent advances in the study of the memristor as the fourth fundamental element in circuit theory, a range of opportunities has emerged and the applicability of memristors has gone hand in hand with the development of mathematical models that can explain their behavior. Recently in the CAD group at INAOE a charge-controlled memristor model has been developed [7], this model represents a great advance since it is generated from the solution of the nonlinear-drift differential equation that models the device. Unlike other approaches made by the group, the charge-controlled model was not generated by a sinusoidal excitation source and as well it is bounded (avoiding reaching negative resistance values), which allows it to be used even when the circuit does not even have a sinusoidal excitement. The model is obtained by homotopy methods, it is possible to use it in DC and switching conditions [7].

Currently, security is a concurrent issue in all fields in a person's life. When we change residence one of the most important factors to take into account is the feeling of safety that we have in the new neighborhood and whether our property is safe enough to avoid any kind of assault. Based on the premise that safety is paramount

to every human being; science and industry have managed to create security schemes; for instance the locks on the doors that provide access to the key holder for that specific lock or the passwords that are used to protect a Wi-Fi key. Although, both types of schemes provide security, the key is physical, while passwords are not; an adversary who has access to the key of the door may violate security if he obtains a functional copy, in the case of a laptop, obtaining the password by any means indicates that the security system has been violated.

The needs of a society have evolved over time and it has become notorious that all tasks such as communications, security, business, financial transactions, identification and so on are much more efficient with the use of technology; even more now that we live in the digital age and the internet of things (IoT). In many applications it is really important to provide an environment of total safety when a direct communication between sender and recipient is established. Suppose you have an installed surveillance camera on which you constantly monitor how things are going at home, and only you want to access the data it transmits; most likely using a key you will access that information, but how does the camera really know that it is you who is trying to access it?
Nowadays a great majority of forms of authentication execute software, usually security depending on the computational capacity of an intruder; one possibility that resonates is to implement these authentications through hardware.

It is commonly seen that in security applications, secret keys are usually stored in memories (volatile or non-volatile), however, many reported examples have shown that they are vulnerable to different types of attacks [8]. Therefore, given an adversary with unlimited computational capabilities, the system must still remain secure, for these reasons it is necessary to implement new security strategies that are more reliable and more difficult to break.

In this work we propose physical unclonable functions (PUFs*)as an alternative

to solve issues previously reported in the field of device authentication, generation of random keys and the generation of random numbers.

1.1 Motivation

This research work is motivated in the exploration of alternatives for hardware security applications; the idea of implementing hardware with its own capabilities to generate unique identity in the devices is a strong and innovative concept that seems very promising and worth exploring in detail. The main objective is to focus this work on relating the memristor as an alternative, since it has several interesting features that can be used to generate systems dependent on the implicit properties at the physical level in obtaining secret keys or seeds of random number generators [9].

Despite not being the first work that aims to use memristors in applications of this type, this work tries to investigate the feasibility of using the memristor model that has been developed at INAOE for hardware security applications.

Hardware security has opportunities in many areas such as authentication and cryptography. Mainly security applications that include electronic devices such as communications, surveillance, banking and health.

1.2 Objective

The main objective of this work is to study the feasibility of using analytical memristor models in hardware security schemes.

1.3 Hypothesis

There are several hardware security schemes that consist of oscillator-based physical unclonable functions. In these circuits, the key working factor is the oscillating frequency. Since a memristor can be regarded as a time-dependent resistor, this work is based on the idea of establishing the oscillation frequency with a memristor and controlling its value with the memristor parameters.

1.4 Methodology

The methodology can be recast in the following steps.

- Achieve a conceptual classification of the state of the art.
- Define a security primitive to focus this work, electronic physical unclonable function are emphasized as the main alternative.
- Introduce the memristor as a circuit element to analyze unique properties that make it viable to be used as a complement to physical unclonable functions.
- Select an electric circuit type physical unclonable function that operates with memristors and use it as a security primitive.
- Subject the system to variability in the internal parameters that define the memristor model, to quantify the effects it has on performance.
- Analyze the effect of the memristor parameters in the overall system performance.

In Chapter 2 physical unclonable functions will be introduced and the main concepts necessary to give a global ideal of the topic, with examples and applications.

In Chapter 3 the memristor is formally introduced as a circuit element. In Chapter 4 we propose the use of memristors to make security schemes that implement physical unclonable functions. Later on, in Chapter 5, results for the proposed system are presented and finally in Chapter 6 we have the conclusions and future work.

Chapter 2

Fundamentals of physical unclonable functions

The silicon-based physical unclonable function (PUF*) was first proposed as a random physical function in 2002 at about the same time that an optical PUF was first introduced under the notion of a physical unidirectional function. The term PUF is now used to refer to several physical topologies developed to take advantage of innate physics in process variations for many different applications. [10]. The appearance of PUFs goes back to the search for new security primitives that leverage of the inherent and unique characteristics of physical objects. The concept of -unclonability- was exploited to somehow produce a signal from which no one has absolute control, thus making any input-output relationship of this object unique, on the other hand the non-clonability of the intrinsic properties of the object makes it impossible to replicate. Physical unclonable functions are an emerging as promising solutions to establish confidence in an integrated system, the main contribution of this type of systems is that they derive from physical properties that are used when required, with low overhead costs in energy and area.

Since then, many opportunities have appeared in order to obtain physical functions that come from different natures, we will explore PUFs taxonomy and we will try to define as much as possible every concept related to these system in this chapter.

2.1 Physical unclonable functions

The first thing to talk about when it is about physical unclonable functions is the concept of "individual" as cases are seen in regular life, like "fingerprints", "wrinkles in the skin of an old man", or "lines of a zebra"; all these are examples of unique characteristics that show up the particularities from different subjects. As it is well known, "the wrinkles, fingerprints or lines of the zebra" are unique, and it is impossible (naturally talking) to find exactly the same pattern on another individual. Expanding this kind of example we could imagine that these natural differences are also possible to occur in other fields, such as physics.

Now, extending the previous paragraph to our topic of interest, a physical unclonable function (PUF*) is an entity that is embodied in a physical structure as a chip or an integrated circuit that exploits the intrinsic complexity and irreproducibility of physical systems to generate secret information; these structures generate an specific response for an specific input, the response depends strongly on properties of the device, these properties are directly related to the unique internal structure of the PUF [11] [12] [2]. The main idea of a PUF is to avoid storing any information, instead the data in a certain way compose the hardware that produces the keys that in principle were not stored. Many of the current PUF designs focus on exploiting the process variations in CMOS (complementary metal oxide semiconductor) technology, normally these variations are not re-creatable and are not under control of the manufacturer, the only way to obtain the same signal for the object more than once is obtaining in advance the input that is producing that response [13].

The operation of the PUF is basically as follows:
When an input (challenge) is sent to the system or chip, a response is produced, the response is defined by the physical function that composes the PUF, this answer is unique to every chip. Given the same challenge, different PUF instances based on the same design will respond differently, such as it is shown in fig 2.1.

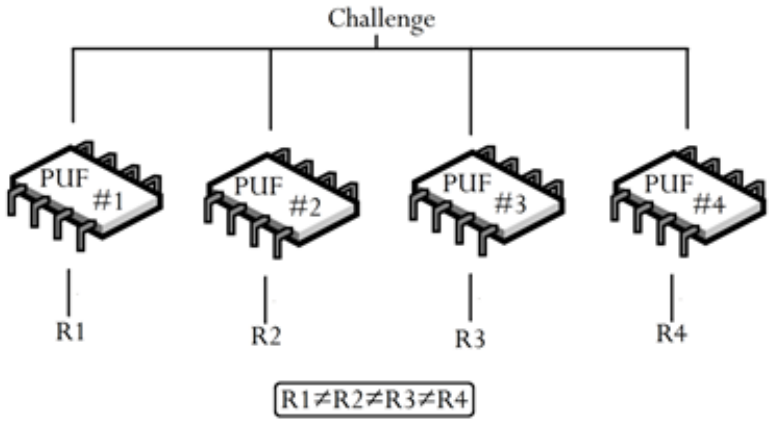


Figure 2.1: Behavior of different PUF instances for a single challenge

A challenge and its associated response, are commonly called as a challenge response pair (CRP*). A set of all possible CRPs in the instance is the identity of the PUF itself. Treating a PUF as a black box it is observed that it is not possible to obtain the same response applying different challenges, or even applying the same challenge to another PUF with the same characteristics, see Fig 2.2. Whether different challenges enter the same system, the output is different; on the other hand different systems that contain the same challenge do not generate the same result either. This is what it is expected to happen in optimal PUF designs, the only way to obtain a key more than once is using the same CRP twice or more.

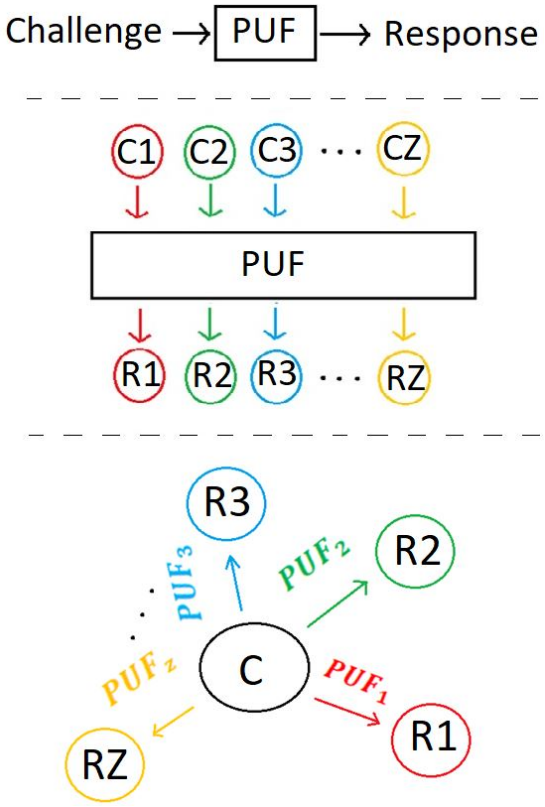


Figure 2.2: PUF seen as a black box, the response is unique for every PUF instance [1]

The behavior of the output that comes from the system is not always ideal, the physical function controlled by the physical conditions itself, could make it possible to obtain different outputs given the same challenge, or even it could produce a single output for different challenges. These behaviors are directly related to the random variations on the physical properties of a PUF. Other important reasons, are the environmental conditions that can affect the functionality of the system [2].

Different types of PUFs have been proposed so far, which base the generation of their output signal in the use of different technologies (although always leveraging process variations in the manufacture of that operating mechanism). Some examples of those mechanisms can be found in [2], however; in this chapter we will make a brief summary of some different PUFs based on electronics.

2.2 PUF Taxonomy

There has been more than a decade of intensive study on PUFs since the concept was first introduced in [14]. Among many PUF that have been proposed, silicon PUFs are the most interesting in terms of fabrication cost and readiness to be integrated to computing and communication devices. There are three major silicon PUFs natures defined by the physical features that generate their physical function, analog electronic PUFs, memory-based PUFs, and delay-based PUFs.

In this Chapter, we give a brief introduction to some of the PUFs outlined in fig 2.3.

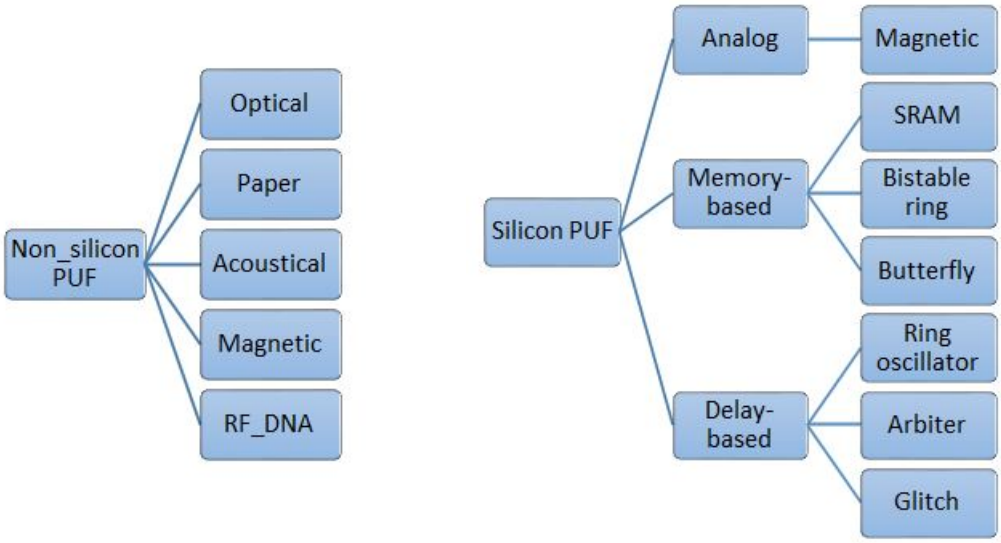


Figure 2.3: PUF Taxonomy with few examples of each

Non- Silicon PUFs: Are common designs that, although they obtain their unique function by physical phenomena, these are not based on process variations. The vast majority are not related to electronics.

Silicon PUFs: Use the uncontrollable manufacturing variations to generate a unique signature for each IC. According to the different source of variation, silicon PUFs can be categorized as:

- Analog electronic
- Memory based
- Delay-Based

There is a wide variety of PUFs designs so far, in this chapter we will show some examples, so only a few of them will be to introduced, in order to show the main idea about how different PUF technologies operate.

2.3 Characteristics of PUFs

If properties on a finger print were not unrepeatable, non-transferable or unclonable, surely they would not be registered by governments to identify people entering or leaving a country. In the same way, whether technologies do not differentiate one fingerprint from another, they will not be effective as identification methodology. With all this mentioned, we can agree that the advantages of these non-split phenomena can be exploited if and only if certain conditions that guarantee the differentiation of one subject from another are accomplished.
In the same way, physical functions that can not be cloned must fullfil conditions to be cataloged as:

Evaluable: A PUF is evaluable when, given a single random challenge the evaluation of the output is easy to measure and reliable. Any CRP must be evaluable with no risk in the result of the output.

Trustworthy: The CRP relation must be strong enough to be repeated with a really low error percentage to be still considered as the output for that challenge without confusing it with the output for another challenge.

Unclonable: A PUF is considered unclonable when the CRP relation is different for every PUF instance. It is impossible to obtain R from C without the physical presence of the PUF. In other words, given a PUF, it is not possible for an adversary to build another PUF that provides the same responses to every possible challenge.

Unpredictable: The response R given a challenge C is random and unpredictable, but it should remain the same for the same challenge over multiple observations. There must be an impossible mathematical equation to predict a response given a challenge, the only way of predicting the output for a challenge has to be achieved by previous proves.

Realizable: This property means that a PUF instance must be possible to obtain given its physical properties, most all types of PUFs are based on physical phenomena impossible to duplicate, however, a PUF could not be realizable when those phenomenous are not unclonable and random. The behavior on the CRP characteristic must be random in every PUF instance.

One way: Given only y and the corresponding PUF instance, it is not possible to find x such that $PUF(x) = y$

Tamper evident: Altering the physical entity embedding PUF transform PUF to PUF' such that with high probablility $\forall\ x \in C\ PUF(x) \neq PUF'(x)$

2.4 Classifications of PUF

As the size of CRP space has a direct implication on PUFs applications and their threat model, the already existing PUFs are often dichotomized into weak and strong [15].

2.4.1 Strong

A strong PUF is characterized by a huge number of CRPs, which grows exponentially with the number of symmetric component blocks used to create the PUF. This property makes it unfeasible to exhaustively evaluate all the challenges within a realistic time. Owing to its physical obfuscated structure and complex combinations of component mismatches, the probability of correctly predicting the response to a randomly chosen unknown challenge is very low even with the knowledge of many other CRPs [15].

A strong PUF must fulfill that:

- It has to be impossible to duplicate physically.
- It must supports a large number of CRPs, an adversary cannot establish a successful attack within a realistic time.
- An adversary must be unable to predict a single response for a given challenge even though knowing the response for other challenges.

An example of a strong PUF is the arbiter PUF.

2.4.2 Weak

Weak PUFs have limited CRPs. More specifically, the number of CRPs slighty increases with the number of basic cells or symmetric component blocks to form a PUF.
The CRPs of a weak PUF with finite physical size can be exhaustively measured within reasonable time. Because of that, the peripheral interface of a weak PUF

must be protected by integrating it with a fuzzy extractor (FE) to restrict direct access to the original response generated internally by the native PUF [15].
A weak PUF must fulfill that:

- It has to be impossible to duplicate physically.
- Its number of CRPs is limited with a direct dependency on the number of its challenge bits.

From the application perspective, weak PUFs are normally used in cryptographic key storage/generation [16], [17], [18], device identification [19], and brand name and IP protection [20].
Examples of weak PUF are SRAM, butterfly and coating PUF.

These types of systems, whether strong or weak, can also be:

2.4.3 Reconfigurable

This PUF can change its response given the same challenge by itself. Instead of exhibit a static behavior on its CPRs, it could be modified as a regular part of its functionality. The ability of reconfigure itself is desirable for a wide number of applications [12]. A reconfigurable PUF can be updated in such a way to alter the CRPs configuration fulfilling that:

- It must be impossible to duplicate physically.
- The CRPs of an rPUF are unpredictable after reconfiguration even if the CRPs of an rPUF before reconfiguration are known.
- The security properties of the rPUF are preserved after reconfiguration.
- Its reconfiguration is uncontrollable.

2.4.4 Erasable

This PUF variable works similar to a reconfigurable PUF, although it needs other complementary features, such as making the PUF erasable. It must have a form of non-volatile state to enable this erasure operation [21].

2.4.5 Public

The definition of a Public PUF (PPUF) is a multiple-input multiple-output system that is much faster to execute on the physical device than it is to simulate by several orders of magnitude [8].
In particular, the secrets of PUF and PPUF are different. Secrets of a PUF rely on the unpredictability of its responses for a given challenge. The model of the PUF that mathematically impersonates the physical function of the PUF must be kept safe. On the other hand, the PPUF hardware contains no secrets, since the PPUF model is known to every party including the verifier, prober and also the adversary. As long as the the model storage is safe against manipulation or rewriting, the authentication capacity is derived only from the difference in computational time between the hardware-based PPUF and its model, and the impossibility of the physical PPUF.

2.5 Types of PUFs implementations

Now that we have seen how PUF are classified according to the amount of CPRs achieved and their funcionality, we are going to introduce some different PUF technologies; leveraging the opportunity of making physical functions, many scientists have explored on different fields and they have achieved several types of PUF im-

plementations.

PUF technologies have not been classified yet according to appropriate standards; they are categorized as electronic and non-electronic, according to the way of obtaining its physical function. In this section some PUF schemes will be presented, as well as a table that compiles a large part of technologies reported to date.

2.5.1 SRAM

The SRAM PUF consists of a large number of SRAM memory units, they are basically a closed loop formed by two inverters, as shown in Fig 2.4. The circuit has two possible states, 0 and 1. Any voltage change presented in the transistors due to variations in the manufacturing process, will cause the cell to store the logical value of 1 or 0, caused by the amplifying and effect of each inverter acting on the output of the other inverter [22] thus creating the physical function of the PUF.

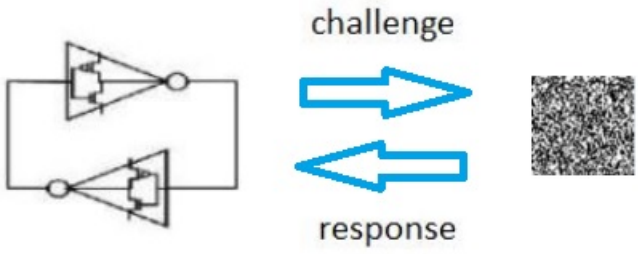


Figure 2.4: Logical circuit of an SRAM (PUF) cell

The right size on fig 2.4 represents a whole SRAM cell, the black pixels represents 1 logic states while white pixels represents 0 logic.

2.5.2 Arbiter

Arbiter-based PUFs exploit the statistical delay variation of wires and transistors across integrated circuits in manufacturing processes to build unclonable secret keys. This circuit was first proposed by [23] [12].

The circuit mainly consists of two routes that are manufactured of the same length in the layout, a signal is sent at the same time in each route, the challenge (X) is a signal of N bits where each bit configures a multiplexer which will indicate if the routes are interlaced or not generating random delays (created by the process variations inherent in the manufacturing process of CMOS technology). To evaluate the output to an input, the output latch (D) decides which path is faster, the generated bit is 1 if the data in D is faster, otherwise the output bit is 0.
This circuit generates a bit in the output, however there are methods to generate bitstrings at the output from the same circuit. An effective method is to use the circuit k times with k different input vectors obtained from a random number generator and for each run use a vector to configure the MUXs.
Because the PUF circuit is rather simple, attackers can try to construct a precise timing model and learn the parameters from many input-output pairs [12]. To prevent these model-building attacks, the PUF circuit output can be obfuscated by XOR'ing multiple outputs or a PUF output can be used as one of the MUX control signals. Note that the model building attack is irrelevant for the cryptographic key generation where the PUF output is never directly exposed.

2.5.3 Ring oscillator PUF

The ring oscillator circuit works generating oscillatory signals at different frequencies, which are used to generate bit strings. The ring oscillator PUF was first introduced by Suh and Devadas [2]. It is an analog-digital system that is able to detect small

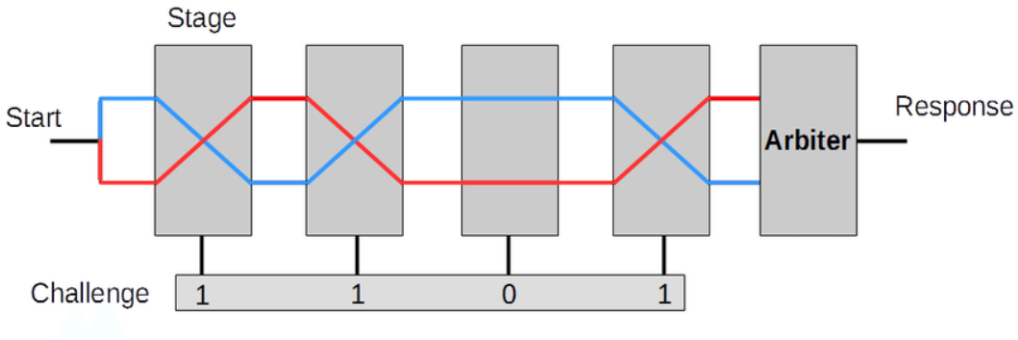


Figure 2.5: Arbiter PUF circuit implementation [2]

variations in frequency of different oscillators and generate binary outputs with high randomness and reliability. A challenge generates a unique bit string that is difficult to clone by another PUF with the same characteristics or even that has gone through the same manufacturing process. The system in a global way can be evidenced in Fig 2.6.

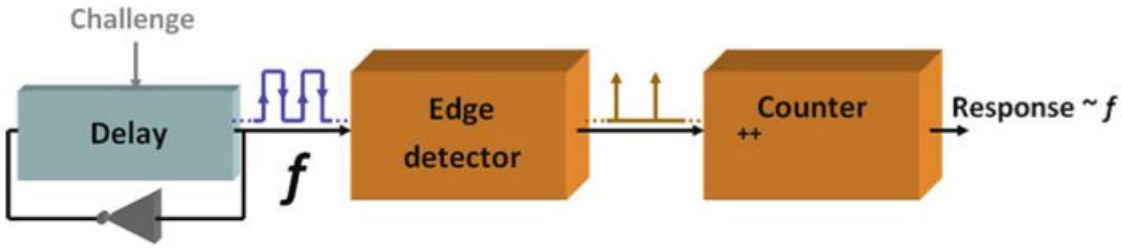


Figure 2.6: A ring oscillator PUF [3]

The following table is presented in order to demonstrate the amount of different opportunities for hardware PUF implementations.

Butterfly	SRAM	Bi-stable ring	Acoustical
Flip flop	Arbiter	Power distribution	Paper
LC	Glitch	Optical	Magnetic
Coating	Ring oscillator	CD	Memristive

Table 2.1: Examples of PUF technologies reported

2.6 Applications of PUFs

Physical unclonable functions are still growing their field of application, at this point of the text we will give the reader a wide view of how many opportunities exist for implementing it (Fig 2.7). It is suppose that a global knowledge of PUF is already built on reader's mind.

However, it has been seen that PUFs are commonly used as hardware security primitives for security applications, such as authentication and key extraction [24], IP protection [25] and integrated circuit (IC) obfuscation [26].

PUF do not store any information, instead they create the information with high reliability as soon as they are energized, reducing the probability of being the target for attackers, and consuming less energy than other solutions used for these purposes. In order to give a brief explanation only a few of them will be mentioned on this work.

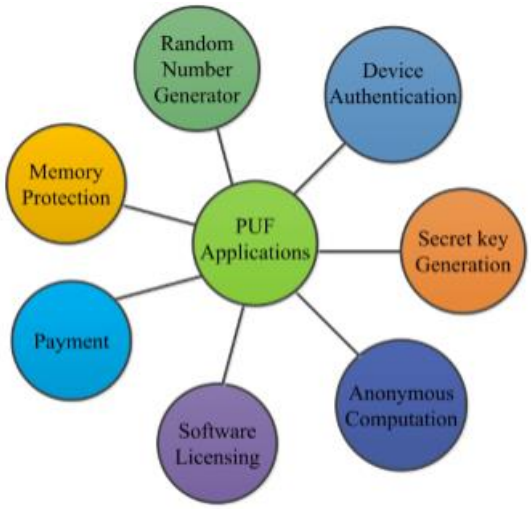


Figure 2.7: Applications and opportunities for PUFs on different fields taken from [4]

2.6.1 Authentication

Identification is an inherent characteristic of the PUF, since just as a person can be identified by their fingerprints, with a PUF any device can be identified based on its unique physical properties. As already explained above, output errors can occur due to physical conditions, that is, the output of the PUF of a device may not always be the same. However, this does not affect the identification since the outputs of the PUF in the same device will be very similar and at the same time different from the outputs of other devices.

In the operation of a RO-PUF, there are usually two phases, enrollment phase and evaluation phase. During enrollment, responses are generated by applying different challenges, and the CRPs are stored in a secure database for subsequent use. In the evaluation phase, the RO-PUF is supplied with the challenge of a random CRP from the secure database to regenerate a response.

During the identification process, the PUF generates an output that is compared with the outputs of PUFs from other devices stored in a database. If the output is close enough to any of the stored outputs and at the same time is sufficiently different from the rest, it can be concluded that the identification process is a success. Fig 2.8 shows a basic outline of this process. [2]

2.6.2 Generation of random numbers

Another application for these systems is the generation of random numbers with low cost and large entropy. The random sequence is obtained by introducing random and unpredictable challenges to generate equally unpredictable responses, or failing that, when you have a challenge which is affected by variations in your environment (high variability), it will always produce unpredictable outputs.

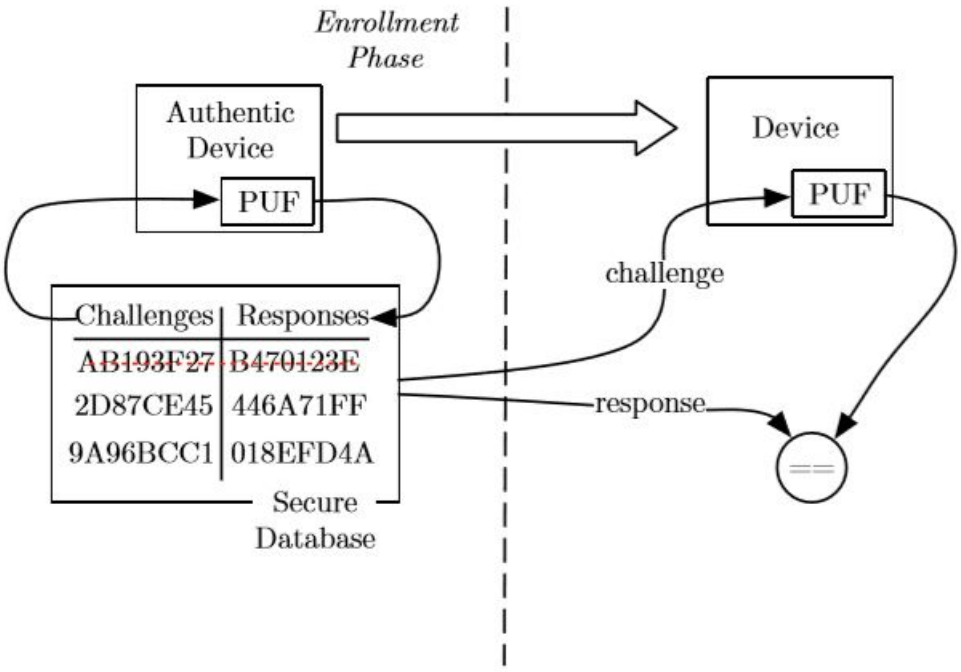


Figure 2.8: Overview of PUF-based authentication [2]

2.6.3 Secret key generation

PUFs have interesting properties in the generation and storage of secret keys. As the keys are generated from the instability of the physical properties that govern the manufacturing processes, intermediate steps are not required for the programming of the keys, thus simplifying the distribution process of the keys; for this reason, a non-volatile memory is not required to store keys. Implementing PUFs for generating cryptographic keys provides extra security against probing attacks and side-channel attacks since the keys are only in the hardware when it is powered [3].

Chapter 3

Memristor the missing circuit element

3.1 Conceptualization of the memristor

The study of electrical circuits for several decades was focused on only 3 basic elements, the resistor, capacitor and inductor; however, the classic approach in circuit theory could not explain some phenomena of a purely electric nature that were declared as anomalies. In 1971, Professor Leon O. Chua defined a theory in which the existence of direct relationships between the variables not yet related, electric charge and flux linkage was demonstrated [27]. The possibility of a fourth basic element of the circuit existed and this would close the link between the fundamental electrical variables, that element was assigned the name of memristor. It was until 4 decades later at the Hewlett-Packard laboratories that the physical device was created, demonstrating that the theory published by Chua was a reality. Furthermore in 1976 Chua and Kang extended the analysis, which discover the existence of memristive systems and demonstrated that diverse systems like thermistors, Josephson junctions and ionic transport in neurons [27] [28] [29] were special cases of memristive systems.

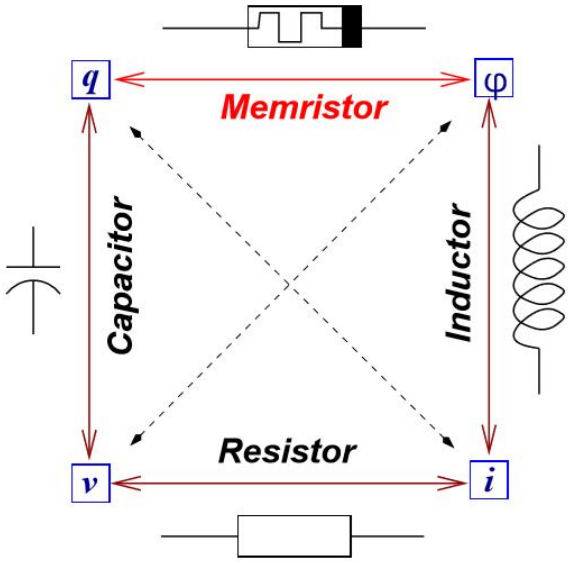


Figure 3.1: Electrical relationships between the all basic circuit elements

These advances caught the attention of the scientific community and many investigations around the device have emerged; the structural and operating behavior of the memristor as a real element had to be understood in order to obtain models according to its operation and reproduce it massively. To date, applications of memristor theory have been reported in fields such as biophysics, programmable logic, signal processing, neural networks, control systems, reconfigurable computing, hardware security, etc.

3.1.1 Memristor

The memristor (memory resistor) is a circuit element defined under an argument of physical symmetry, created for the first time as a semiconductor device in 2008, it establishes a direct relation between flux leakage and electric charge, see Fig 3.1.

This element is basically a resistance that changes its value depending on the magnitude and direction of the current passing through it, and also has the ability

to retain its last resistance value when the current flowing through it is interrupted.

In Fig 3.1 there are 6 possible relationships between the 4 fundamental variables, of which 3 of them correspond to the direct relationships that are obtained by resistor, capacitor and inductor. The mathematical relationships between these variables are:

$$v = R * i \tag{3.1}$$

$$q = C * v \tag{3.2}$$

$$i = L * \phi \tag{3.3}$$

Reviewing the mathematical relationships for electric charge (q) and flux linkage (ϕ) represented by the dotted lines, we have the following integral relations:

$$i(t) = \frac{dq(t)}{dt}; q(t) = \int_{-\infty}^{t} i(\tau)\ d\tau \tag{3.4}$$

$$v(t) = \frac{d\phi(t)}{dt}; \phi(t) = \int_{-\infty}^{t} v(\tau)\ d\tau \tag{3.5}$$

Then by an argument of symmetry one might think that the red line in fig 3.1 should represent a direct relation between charge and flux, an element that relates them must exist. This missing relationship was proposed by Chua and is defined as:

$$d\phi = M(q)dq; v(t) = M(q(t))i(t) \tag{3.6}$$

From the previous equation it is observed that M(q) has ohms as units and its value is a function of time (presenting memory effects), thus fulfilling what was

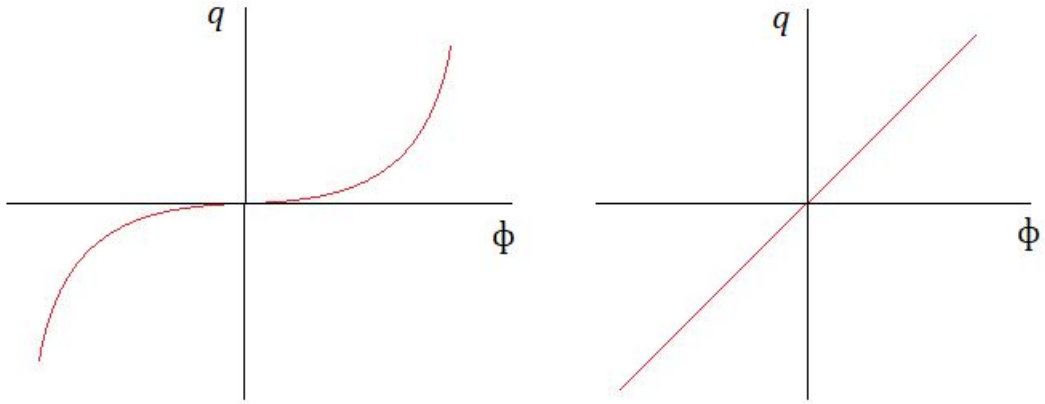


Figure 3.2: Memristor charge(q)-flux(φ) characteristic

expected to be defined as a memristor.

The parameter M(q) was defined as memristance with units of (Ω) and analogously to a resistor, there is an inverse relationship that in this case is known as memconductance and is defined as:

$$dq = W(\phi)d\phi; i(t) = W(\phi(t))v(t) \tag{3.7}$$

3.1.2 Memristor's fingerprints

Any semiconductor device exhibits particular behaviors that allow it to be identified as such. A device is considered a memristor when it complies with the following fingerprints.

1. **Monotonically increasing charge(q)-flux(ϕ) characteristic:**
   The characteristic curve of the memristor is a relation between the variables load and flow, this characteristic must always be monotonically increasing. The shape of the curve can vary if the memristor behaves as a time-varying

linear resistance or time-invariant linear resistance, see Fig 3.3

2. **Pinched hysteresis loop:**

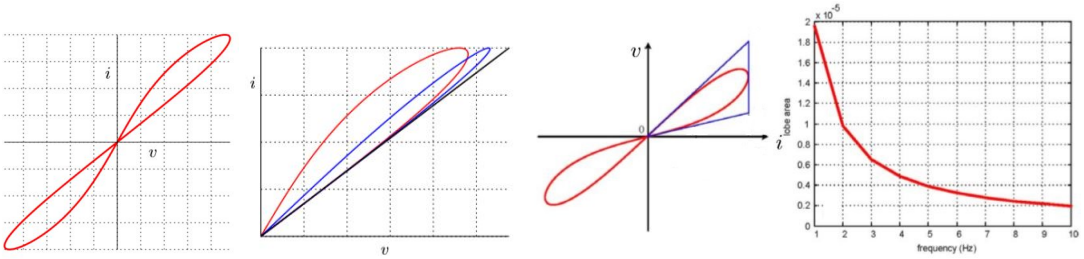
   The voltage-current relation for a sinusiodal excitation source presents a cycle of hysteresis for all time t, with the particularity that it must always cross by the origin of the Cartesian plane; that is, the element does not have energy storage properties.

3. **Reduction of the lobe area as the frequency increases:**

   The area of the hysteresis loop in the voltage-current curve decreases monotonically when the frequency of the sinusoidal excitation increases. Thus tending to behave as a linear resistance.

4. **Constant memristance at infinite frequency:**

   The hysteresis loop decreases when the frequency increases, tending to behave exactly equal to the curve of a linear resistor, it means, the memristance loses its dynamic behaviour.



(a) (b) (c)

Figure 3.3: (a) Hysteresis loop (voltage-current) (b) Memristor hysteresis loop turning into a resistor's characteristic (c) Reduction of hysteresis area lobe

3.1.3 HP memristor

Important events took place in the appearance of the memristor as a circuit element, which were mentioned previously in this chapter, Chua's theory and the creation of

the device.

37 years after the first public report on the existence of another circuit element, in the laboratories of HP a device was manufactured that, under certain controversies in the scientific community, was known as a memristor [30]. A crossbar structure composed of a layer of titanium oxide (TiO_2) formed by two layers with different concentration of oxygen vacancies (insulation layer TiO_2 of 2:1 ratio and conductive layer TiO_{2-x} with 5% of vacancies that result in greater conductivity) located between two platinum electrodes (Pt) Fig 3.4. A voltage applied to the device relocates the oxygen vacancies between the layers of TiO_2 by changing the conductivity of the device, when suspending any kind of excitation towards the device, the oxygen vacancies remain in their last location, fulfilling the memory property of the memristor.

The total length of the device and the doped area are denoted as Δ and w respectively, both zones have different conductivity as a function of the number of oxygen vacancies, the resistance of the region of TiO_2 (not doped) is Ron (resistance at ON state) on the other hand the resistance at region of TiO_{2-x} (doped) is $Roff$ (resistance in the OFF state). We have the equivalent of total resistance as the sum in series of both regions defined by the expression

$$R_{total} = R_{on}\frac{w}{\Delta} + R_{off}(1 - \frac{w}{\Delta}) \tag{3.8}$$

The length w can be normalized by taking $x = w/\Delta$, x corresponds to the state variable of the memristor and can take values from 0-1 (note that this variable represents the position of the dividing line between the Ron state and $Roff$), the variable x is controlled by the current that passes through the device $i(t)$.

Equation (3.9) using the linear drift model defines the relationship between the

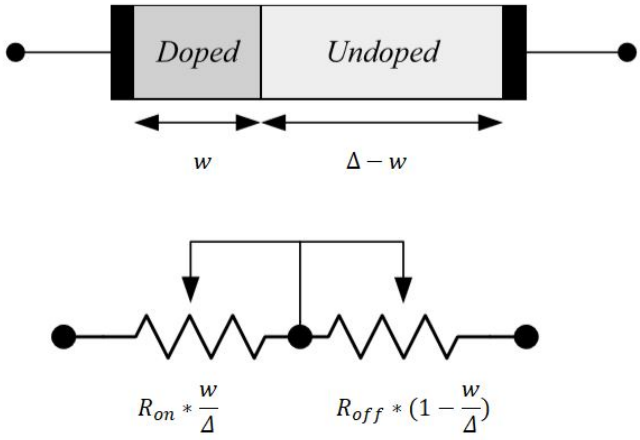


Figure 3.4: HP device

change of the state variable and the current

$$\frac{dx(t)}{dt} = \eta \frac{u_v R_{on}}{\Delta^2} i(t) \tag{3.9}$$

Where u_v is the mobility of the charges in the doped region and is measured in $\frac{m^2}{\Delta}$ and η indicates the direction of displacement of $x(t)$. As well, we can define a new variable $\kappa = \frac{u_v Ron}{\Delta^2}$, in order to have a simpler equation. In fact, η defines whether the movement of the interface compresses (η = -1) or widens (η = 1) the doped region under a positive polarization.

Considering the effects of non-linearities due to the high electric fields of the memristor, it is convenient to use the non-linear drift mechanism that introduces a window function $fw(x(t))$ to the expression eq (3.9) to obtain :

$$\frac{dx(t)}{dt} = \eta \kappa i(t) fw(x(t)) \tag{3.10}$$

This window function must fulfill certain characteristics; must be a limited func-

tion in domain and range to values between 0 and 1; also at the edges it must show a bottleneck such that $fw(0) = fw(1) = 0$ to guarantee no drift at the boundaries. These properties guarantee that the difference between this model and the linear-drift model vanishes in the bulk of the memristor as $w \to \Delta/2$

A window function is required in order to include non-linear traction phenomena in the differential equation that models the behavior of the memristor, Eq (3.10)

From the solution $x(t)$ the memristance is defined as the sum of the series resistances of the two regions of the memristor, such as:

$$M(t) = R_{on}x(t) + R_{off}(1 - x(t)) \tag{3.11}$$

Chapter 4

Memristor as a security primitive

Physical functions have been presented as a fascinating opportunity to create new security primitives; since the publication of the first type of PUF, the growth of different technologies based on this concept has increased drastically [31]. The first PUFs reported in the literature were not electronic, previously the idea of security based on inherent, non-clonable and unrepeatable disorders of physical objects emerged as a research topic in order to mitigate hacking threats in security systems [32].

As process variations become more prevalent due to technology scaling into the nanometer regime, novel nanoelectronic technologies such as memristors become viable options for improved security in emerging integrated circuits. With the growing development of technologies that cause process variations it is expected that new generations of PUFs will be implemented using nano technologies.

In this work, memristors will be included with special interest among other nanotechnologies that are of interest for science because they exhibit higher levels of randomness (thickness, cross-sectional area or doping profile) as a consequence of scaling down the nano scale. These devices are manufacturable, cheap and compat-

ible with the fabrication processes of CMOS technology. Memristors in particular have great properties to be used as security primitives: cycle to cycle (C2C) variations, can be implemented in crossbar architectures, bidirectionality, non-volatility, small footprint, low energy consumption, among others [1].

A general classification of PUF nanotechnologies is shown in fig 4.1. there are several PUFs designs that can be implemented in more than one nanodevice.

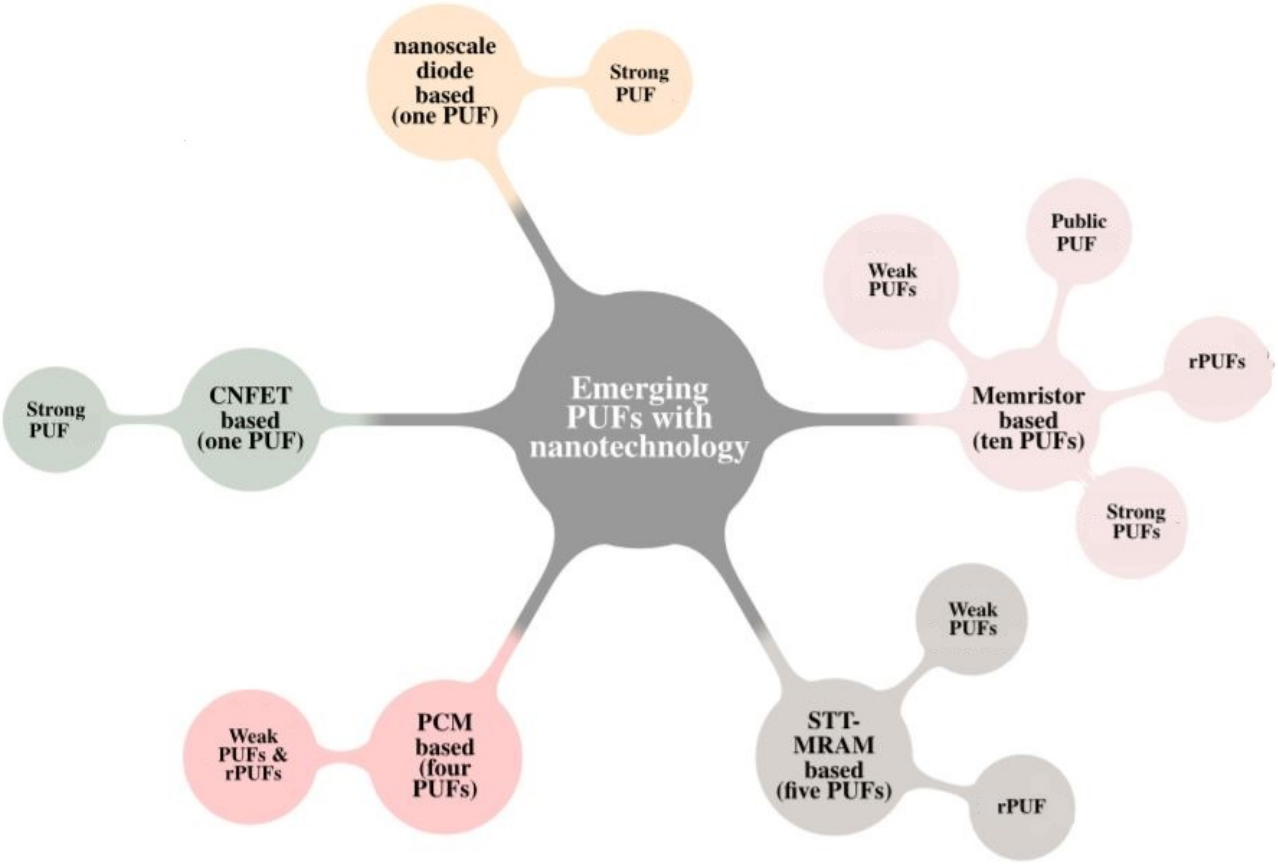


Figure 4.1: Emerging PUFs with nanotechnology. A general classification according to the technology employed [1]

4.1 Memristor and physical unclonable functions

This work uses memristors on security applications in order to obtain random and safe keys, leveraging the fact that memristors are able to vary its memristance as a function of their internal parameters. Memristors are as well highly sensitive devices to process variations.

Some arguments that justify the use of memristors in security applications that use PUFs are listed below:

- The memristor system status variable $w(t)$ or the corresponding effective resistance $M(w, i)$ provides an ideal situation for building memristor PUFs.
- The manufacture of this device is compatible with the manufacturing processes of CMOS technology.
- They present a phenomena known as cycle-to-cycle variations, in which its resistance is not known with total accuracy after a switchover.
- Non-linear and bidirectional input response.
- Non-volatility

The research work on these devices is just being focused on hardware security, many of their advantages have already have been mentioned. On the other hand being realistic, it must be emphasized that the memristor is a device that still requires a lot of research work regarding modeling and manufacturing. Despite these drawbacks, it is clear that it is worth taking a look at any promise that the theory can offer.

4.1.1 Memristive PUF

An example of a memristive PUF is presented in Fig 4.2. This circuit creates its physical function by leveraging the variations in the writing time to generate a bit of information, with the resistive switching properties of this element [33], the memristor is switched between its high and low resistance states and thus the output is generated. This circuit proposal has also been used to generate bitstrings of n bits at the output [5], and the memristor fulfills the same function of producing the physical function of the PUF.

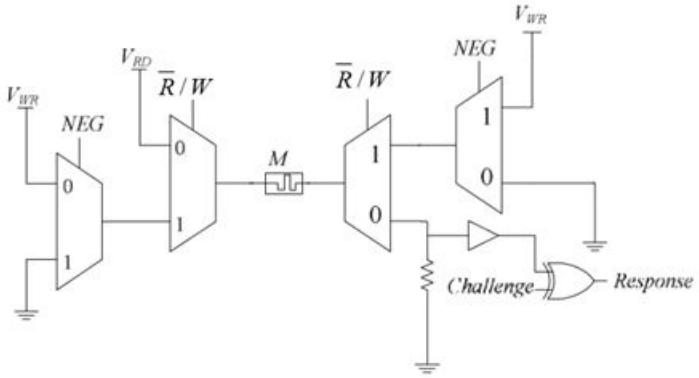


Figure 4.2: A 1-bit memristive memory-based PUF cell [5]

4.1.2 Mr PUF (Nanocrossbar structure):

Another newfangled alternative of using memristors in security primitives with PUFs is proposed in [6]; it leverages the large amount of information available in the nanocrossbar array along with the resistance variations of memristors.

The mrPUF architecture shown in Fig 4.3 consists of two key components: a nanocrossbar MxN matrix and a two 5-stage mirror-controlled current oscillator (CM-RO) that are reconfigured with the nanocrossbar and consequently, they result in a significant reduction in area; the individual variations in the resistance of

memristors in the nanocrossbar matrix are the ones that really make mrPUF work, while the CM-RO having an odd number of inverters translates the analog resistance variations of an individual memristor into frequency to digitize those analog variations.

The challenge bits are used to provide the address bits for both the multiplexers and the decoder, so that a selected memristor controls the reference current in the current mirrors, which results in a current starved ring oscillator structure. In this way we have that the oscillation frequency is dependent on this current which, is a direct function of the memristor value.

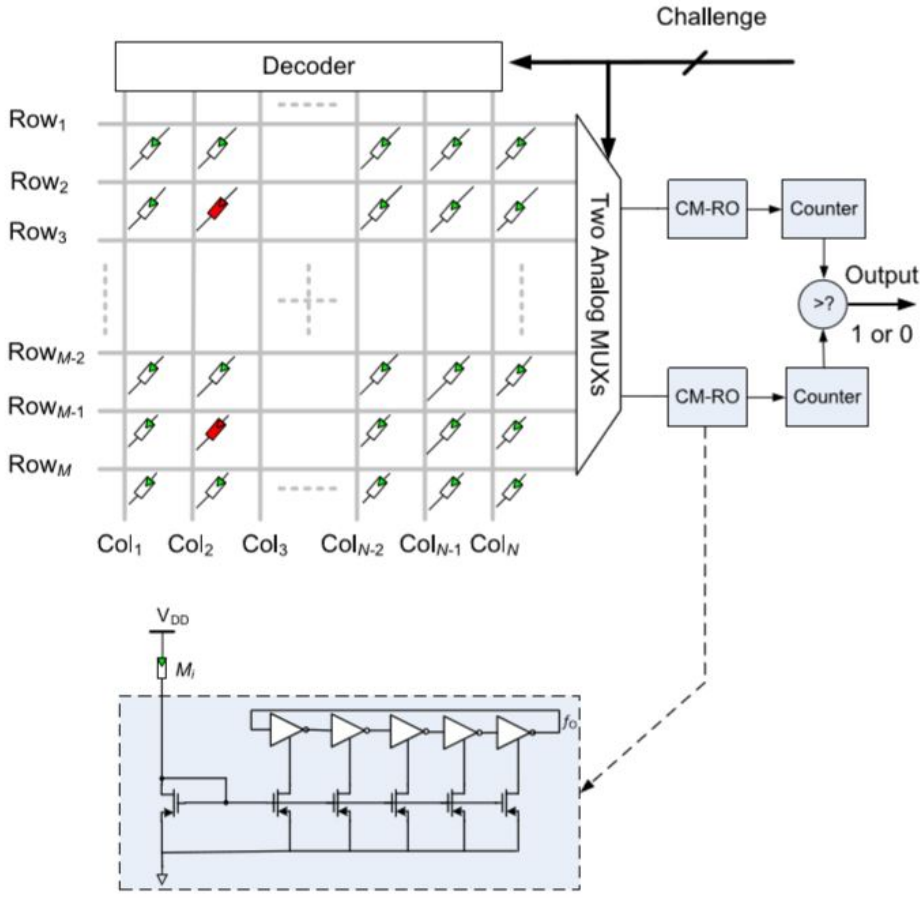


Figure 4.3: Nanocrossbar array structure using memristor for swicthing [6]

4.2 Memristor charge controlled model

Although the memristor model reported in [34] fulfills all the fingerprints of the device, it is very limited and presents many irregularities for its use in several applications. The main drawbacks of this model are, on the one hand, that it was generated from a source of sinusoidal excitation being only usable with this type of signals and, even more importantly, this model is not limited, which causes the generation of extremely large or negative memristance values.

A model controlled by electric charge is defined in [7] in order to solve the drawbacks mentioned above, where basically the current is integrated to obtain the electric charge and thus defines the behavior of the memristor.

Modifying Eq (3.10) and Eq (3.11) and expressing them as electric charge dependent, the current can be expressed as $i(t) = dq/dt$, which yields

$$\frac{dx(q)}{dq} = \eta\kappa f w(x(q)) \tag{4.1}$$

$$M(q) = R_{on}x(q) + R_{off}(1 - x(q)) \tag{4.2}$$

With a function defined in terms of the electric charge the expressions for the dynamics of the memristor eq (4.1) and the memristience eq (4.2) are observed from eq (4.1) that a window function is required. There are several versions of window functions commonly used, however for this model Joglekar's window was used [35].

$$f_w = 1 - (2x - 1)^{2k} \tag{4.3}$$

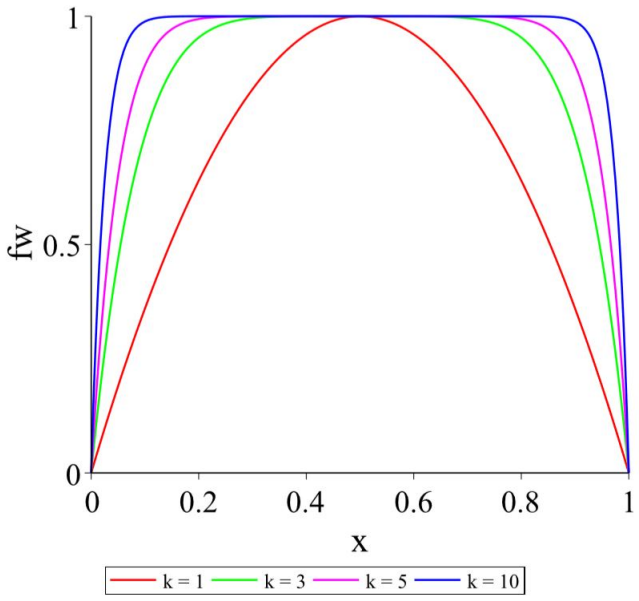


Figure 4.4: Joglekar's window function

Where k controls the linearity level of the function, (x and linearity have a directly proportional relationship), the function is also symmetric in both directions.

Plugging equation (4.3) in (4.1) we get the definitive expression that must be solved for the memristor

$$\frac{dx(q)}{dq} = \eta\kappa(1 - (2x(q) - 1)^{2k}) \tag{4.4}$$

This equation is solved by using homotopy methods [7]. From these methods of solving it should be noted that the equation of memristance is different depending on the value of k and the order n for homotopy.

Employing a value of $k = 1$ in the window function and solving the homotopic order 1 expression, an equation controlled by the electric charge that represents the memristance is obtained [7]. However, there are a couple of solutions because η takes values of +1 and -1 depending on charge displacement direction.

$$M(q) = \begin{cases} \overbrace{(R_{off} - R_{on})(X_o - 1)[(X_o - 2)e^{4\kappa q} - (X_o - 1)e^{8\kappa q}] + R_{on}}^{\eta_+} & q \leq 0 \\ (R_{off} - R_{on})X_o[X_o e^{-8\kappa q} - (X_o + 1)e^{-4\kappa q}] + R_{off} & q > 0 \\ \overbrace{(R_{off} - R_{on})X_o[X_o e^{8\kappa q} - (X_o + 1)e^{4\kappa q}] + R_{off}}^{\eta_-} & q \leq 0 \\ (R_{off} - R_{on})(X_o - 1)[(X_o - 2)e^{-4\kappa q} - (X_o - 1)e^{-8\kappa q}] + R_{on} & q > 0 \end{cases} \tag{4.5}$$

Using mathematical operators to condense equation 4.5 into a simpler expression, we define θ and Λ as shown in fig 4.5. The θ operator describes the direction of the drift η while the Λ operator is used to describe the sign of the electric charge q.

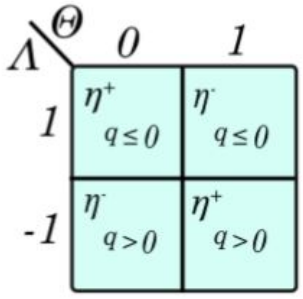


Figure 4.5: Applied operators on memristance equation [7]

$$\begin{aligned} M(q) = [\theta] * [(R_{off} - R_{on})(X_o - 1)[(X_o - 2)e^{\Lambda 4\kappa q} - (X_o - 1)e^{\Lambda 8\kappa q}] + R_{on}] + \\ [-\theta + 1] * [(R_{off} - R_{on})X_o[X_o e^{\Lambda 8\kappa q} - (X_o + 1)e^{\Lambda 4\kappa q}] + R_{off}] \end{aligned} \tag{4.6}$$

X_o is the initial condition of the state variable $x = \frac{w}{\Delta}$.

The expression R_{init} for the memristance of the device determines the memristance value when operating at high frequencies; this value is given by equation 4.7

$$R_{init} = X_0 R_{on} + (1 - X_0) R_{off} \tag{4.7}$$

4.3 Memristive ring oscillator PUF implemented

The concept of oscillator and ring oscillator will be introduced to understand the operation of the ring oscillator PUF, since in particular this circuit will be implemented in this work.

4.3.1 Oscillator

An oscillator is an electric circuit that generates a periodic signal at the output without having any periodic input, it requires a negative feedback loop (Fig 4.6) that under certain conditions might cause an oscillatory response in the circuit. The transfer function of a negatively feedback system is defined as:

$$\frac{V_{out}}{V_{in}} = \frac{H(s)}{1 + H(s)} \tag{4.8}$$

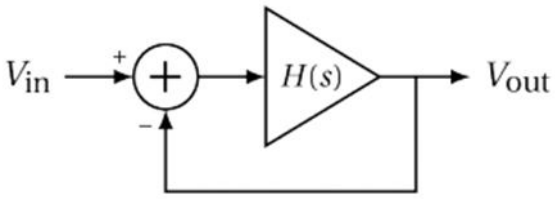


Figure 4.6: Negative feedback system

If the open-loop gain H(s) has enough phase shift to make the closed-loop gain Vout/Vin tend to infinity, the circuit will amplify its own noise components indefi-

nitely, which results in oscillation, fig 4.5.

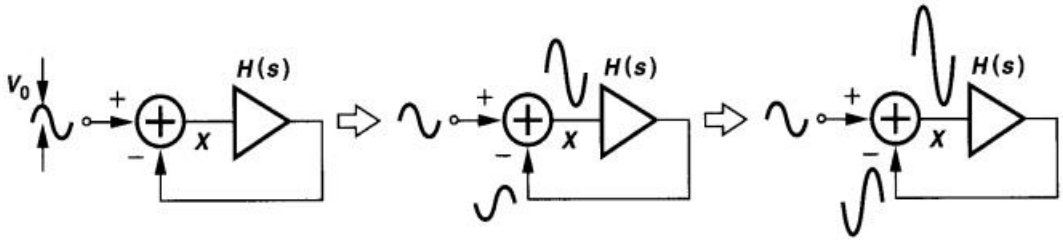


Figure 4.7: Oscillatory process

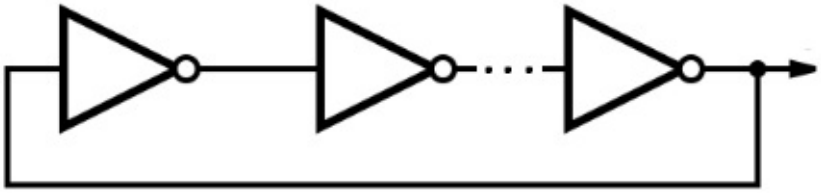
This characteristic behavior of an oscillator is due to a pair of conditions known as the Barkhausen oscillation criterion, it establish that there must be a phase shift of π between the input and output signals of the amplifier, in addition to having a gain loop of at least 1 to generate an instability in the circuit that only stops due to non-linearity phenomena implicit in the system. Such that:

$$\mid H(s = jw_o) \mid \geq 1 \tag{4.9}$$

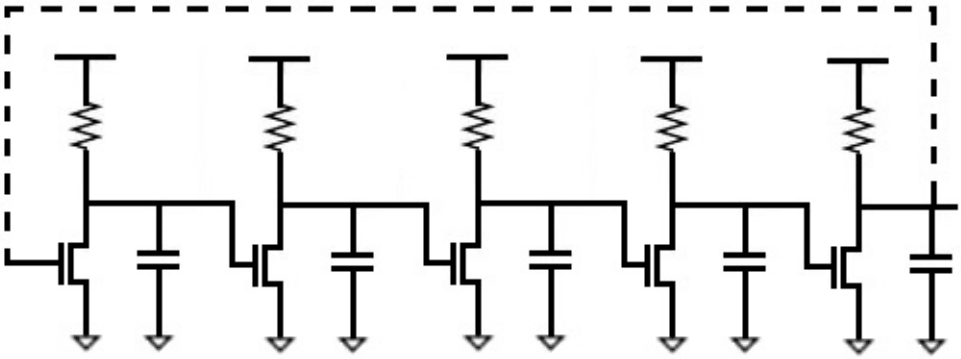
$$\measuredangle H(s = jw_o) = 180^\circ \tag{4.10}$$

4.3.2 Ring oscillator

A ring oscillator consists of an odd number greater than 1 of gain stages connected in cascade that fulfill the Barkhausen criteria mentioned in the previous subsection. Each inverter contributes to the delay of the signal. Adding more inverters increases the total delay and decreases the frequency of the oscillator. According to [36], a greater number of stages generates more robustness to process variations; this is due to the greater number of stages between which these variations are divided or compensated; on the other hand, more stages increase the power consumption of the circuit.



(a)



(b)

Figure 4.8: Five stage ring oscillator with common source

A 5 stage ring oscillator has been implemented for the realization of this work, whereas a memristor is basically a reconfigurable resistor with memory whose resistance depends on the voltage in its terminals over time. We decided to replace the resistors of the ring oscillator by memristors in order to take advantage of their compatibility with CMOS manufacturing processes [37] save area (it is smaller than a resistance), generate more randomness in the frequencies due to several parameters that affect the initial memristance, and observe what interesting behavior appears in the system. On the other hand, for the ring oscillator, common cascade source stages were implemented with the desire to observe the behavior of the memristor in this application.

4.3.3 Memristive ring oscillator PUF

Keeping in mind how a ring oscillator circuit operates, we can introduce formally the ring oscillator PUF.

This PUF is a mixed mode circuit that produces an output bit for a given challenge from the comparison of the frequencies between two oscillators, as indicated by Fig 4.9.

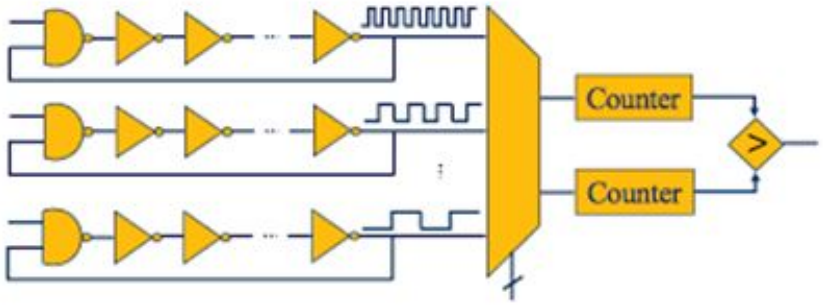


Figure 4.9: Ring oscillator PUF scheme

A multiplexer from N to 2 receives the challenge and selects 2 oscillators that will generate a bit at the output, two counters are used to count the number of peaks in a time window large enough to capture an imbalance in the frequencies of the oscillators. Subsequently a comparator receives the signal from each of the counters and determines which oscillator has a higher frequency; if counter 1 indicates to be greater than counter 2, a high state (1) is obtained at the output and low state (0) is obtained in all other cases. A bitstring of k bits at the output can also be obtained from this system by repeating the process k times for different pairs of oscillators, each cycle resulting in a bit. The methodology of generating the pairs of oscillators can vary and change the performance of the PUF. In Chapter 5, two different methodologies will be applied to obtain the output.

The left part of fig 4.10 indicates the change that is proposed to the ring oscillator

circuit, the right side of the figure shows the complete scheme of the ring oscillator PUF, including the challenge and the output. The performance of the circuit is somehow affected by the memristances of each oscillator, so it is necessary to make a sensitivity analysis to the memristance defined by equation 4.7 for each parameter.

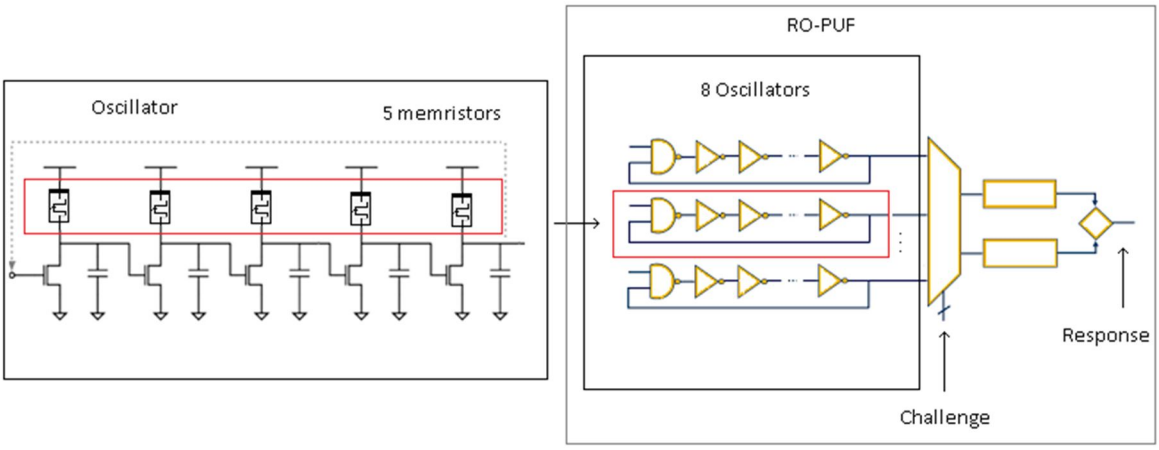


Figure 4.10: Memristive ring oscillator

4.4 Sensitivity to Rinit

As it has been mentioned so far, several ring oscillator circuits are required in the PUF to obtain the physical function leveraged to generate our security keys. An analysis of the variability is required to know how sensitive our oscillators are with respect to the memristance, since it is the variability of the system that defines for which application the PUF is promising.

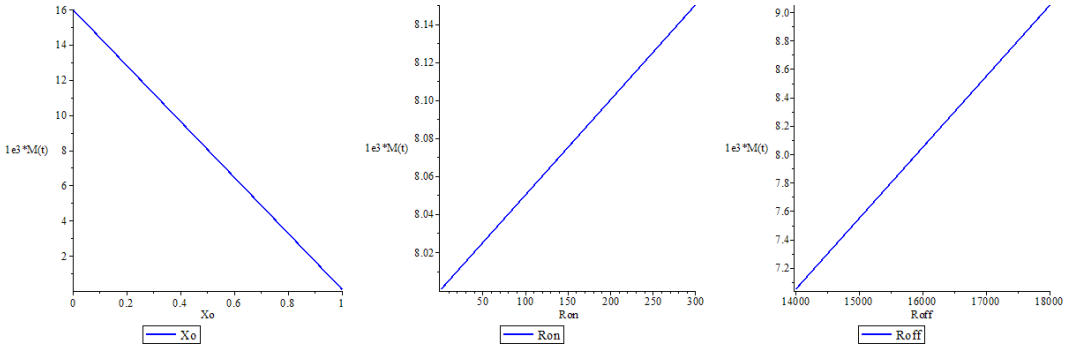
As the resistances that it commonly contains are substituted by memristors, which when operating at high frequencies present a behavior similar to a linear resistor, whose value is set by R_{init}, which represents the memristance at frequency $\omega = \infty$ or the memristance value at $t = 0$.

The memristance of each oscillator array depends on first order on process variations mainly produced by mismatch manufacturing process, in addition to being slightly dependent on the frequency of oscillation (the dynamics of the device are reduced as the frequency increases).
The oscillators are organized in a random way and for the parameters of equation (4.7) the nominal ones are established.

Xo	Roff	Ron
0.5	16kΩ	100Ω

Table 4.1: Nominal parameters for *Rinit*

These figures 4.11 show the resistive behavior of the memristor in high frequency, the controlled by charge model works correctly for this application.
For sensitivity, it can be concluded that the parameters of the memristor have a purely linear influence on the memristance; which means linear variations similarly for the oscillator circuit.



(a) (b) (c)

Figure 4.11: Linear dependence between memristance and parameters (a) Xo (b) Ron (c) Roff

Since it is known so far that the oscillatory behavior is strongly determined by the value taken by the memristors, we want to observe what is the proportionality relation between the parameters of the memristor and the resulting frequencies; for purposes of simplicity only the results obtained in changes of X_0 are provided, since this parameter represents the greatest variability.

Results of the oscillatory output for 3 different values of X_0 are shown in Fig 4.12; there is an inverse proportionality relation between the resistance and the frequency.

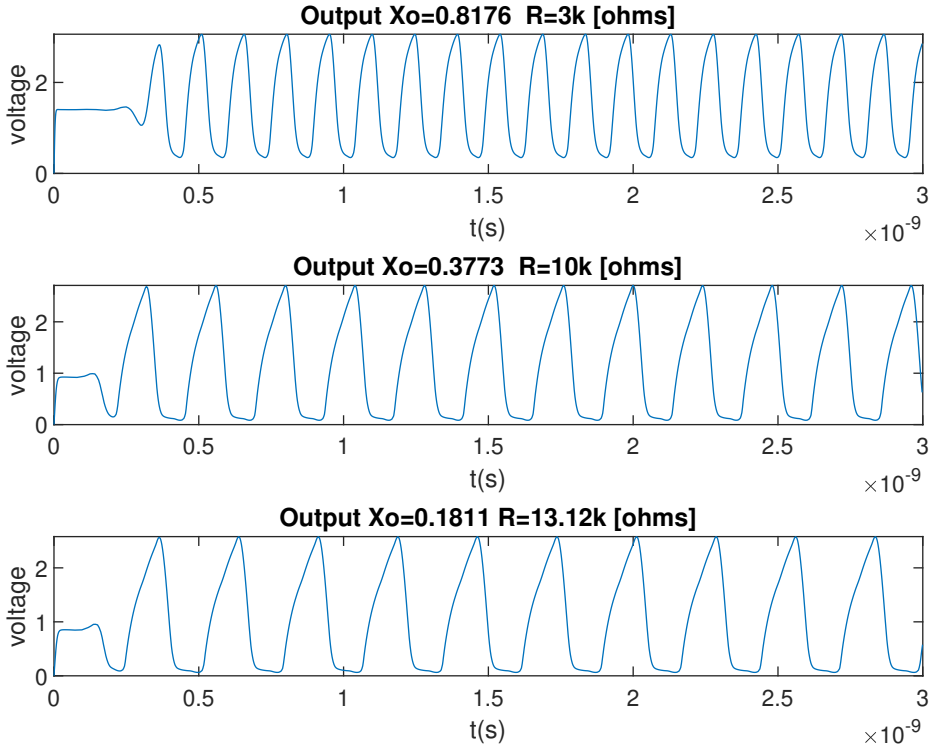


Figure 4.12: Ring oscillator output with different values of memristor parameter Xo

Chapter 5

Simulation and analysis of results

5.1 Performance metrics

The quality of a PUF is determined by metrics which can be used to verify the use of the PUF to a specific application. Since different types of applications have different sets of requirements, not all of these metrics are of equal importance. A taxonomy of such key metrics is shown in Fig 5.1.

Along with these metrics, design in terms of area, power consumption, design complexity, cost and delay always play a key role and should be considered. Similarly, metrics like false positive rate and false negative rate are also important in PUFs for the identification of a particular chip. The false positive rate is the probability of identifying any given chip as some other chip whereas the false negative rate is the probability that a correct chip is identified as an incorrect chip. This information can be obtained from inter-chip and intra-chip variations. These probabilities should be very small (ideally zero).

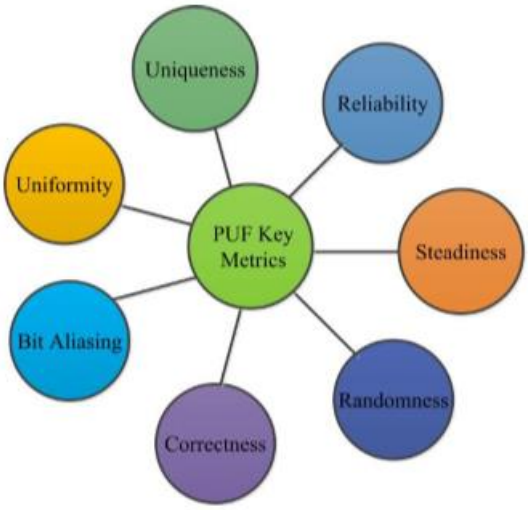


Figure 5.1: Key metrics for determining PUF performance [4]

The performance of a PUF can not be easily measured; despite being implemented as a common electric circuit, it is measured differently compared to other widely known electrical systems. The response of interest in an ring oscillator PUF is not a voltage or a current, it is rather coming from the voltage of two oscillators selected to generate a bitstring, which is the response of interest.

The quality of the response in a PUF must be measured according to how it is adapted to make the system less prone to attacks from third parties. In order to find a quantitative performance indicator of the system quality, many different researchers have proposed metrics, but nevertheless this work will only apply the following three measures widely used on hardware security applications.

Uniformity:

Measures the proportion of 1 and 0 bits of a response bitstring. For truly random PUF responses, this proportion must be 50 percent. Uniformity of an n-bit PUF is defined as its Hamming weight percentage.

$$(Uniformity)_i = \frac{1}{n}\sum_{l=1}^{n} r_{i,l} * 100 \tag{5.1}$$

Where $r_{i,l}$ is the $l-th$ binary bit of an $n-bit$ response for a chip i. The avergae uniformity value is obtained by averaging overall the uniformity values for all PUFs.

Uniqueness:

Represents the ability of a PUF (chip) to be distinguished from a group of chips of the same type. The Hamming distance between a pair of PUFs is used to evaluate uniqueness. If two different P_i and P_j chips have R_i and R_j responses, respectively, to a C challenge.

$$(Uniqueness) = \frac{2}{k(k-1)} \sum_{i=1}^{(k-1)} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} * 100 \tag{5.2}$$

The ideal value for this metric is 50 percent.

Bit-aliasing:

It allows to know whether different chips may produce identical PUF responses which is an undesirable outcome. The bit-aliasing of the $l-th$ bit of the PUF is calculated as the percentage of Hamming weight for $l-th$ bit of PUF across k devices. Its ideal value is 50 percent.

$$(Bit-aliasing)_l = \frac{1}{k}\sum_{i=1}^{k} r_{i,l} * 100 \tag{5.3}$$

Where $r_{i,l}$ is the $l-th$ binary bit of an $n-bit$ response for a chip i. The average bit-aliasing value is obtained by averaging overall the bit-aliasing values for all PUFs.

5.2 Scheme of bits generation

As mentioned in Chapter 2, a ring oscillator PUF can generate bitstrings from different mapping algorithms, the multiplexers act in a predetermined way by the designer of the system, there always being a direct relationship between the challenge and the output.
Three PUFs were made by means of different mapping algorithms, two using the combinatorial of all the possible pairs of oscillators that compose the bench of oscillators of the PUF and the last grouping by pairs of oscillators without the reuse of these. To validate the performance of the PUF we resorted to the definitions given in the previous section. Our objective is to compare the performance of each system against the three defined parameters.

Xo is varied randomly to simulate the effect of variations in memristance produced in manufacturing. For R_{on} and R_{off} similar results would be obtained with the particularity that such parameters would affect the system in lesser proportion.

5.2.1 By combinatory

- **8 oscillators system**

In this first case of study a bench of $n = 8$ oscillators is prepared, it contains 5 inverter stages common source with memristors, the combinatorial $\binom{n}{2} = \frac{(n)(n-1)}{2} = k$ generates 28 output bits produced by a challenge of the same number of bits. Each bit of the challenge corresponds to a combination of 2 oscillators, if one bit in the challenge is 1 the pairing is made by connecting the oscillator RO_1 in the channel 1 and RO_2 in the other channel , if the bit in question is 0 it is taken in the opposite way. The PUF is capable of generating $2^{\binom{n}{2}}$ bitstrings at the output, where each is produced by each of the challenges of $\binom{n}{2}$ bits.

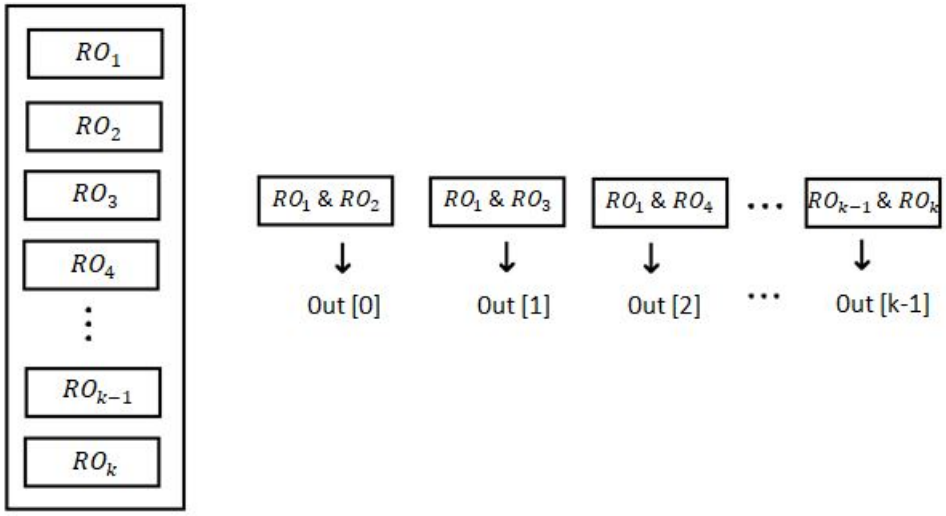


Figure 5.2: Pairing mapping by combinatorial

Results are shown in fig 5.3, 5.4 and 5.5, for this case only 5 PUF instances were prepared.

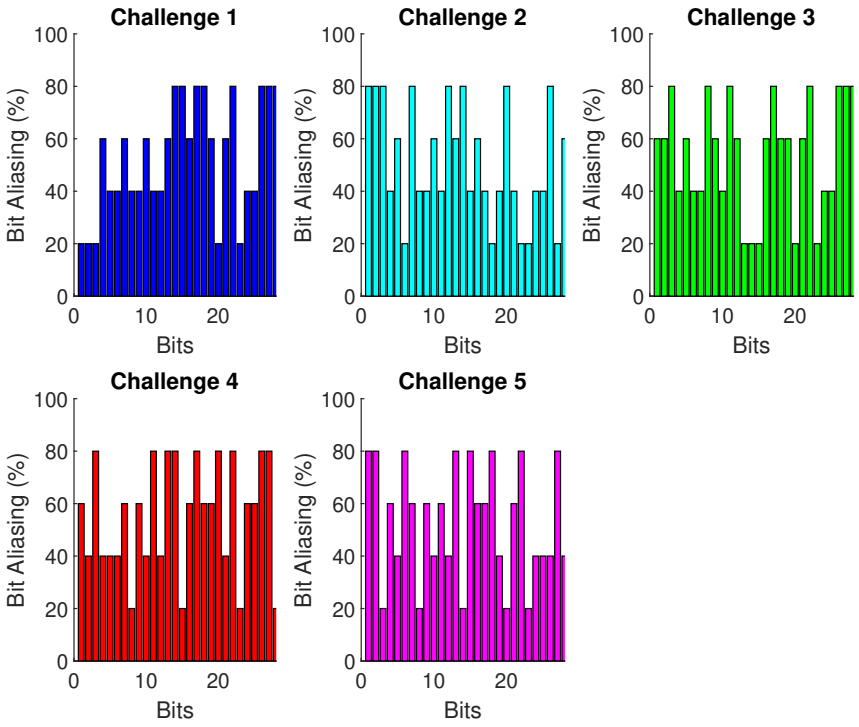


Figure 5.3: Bit Aliasing for different challenges applied (n=8)

If we take a look at Figs 5.3, 5.4 and 5.5 and taking into account that from the definitions mentioned in the previous section, uniformity is the only parameter that is measured against the same instance of PUF in order to rate a PUF as good or bad, while uniqueness and bit aliasing give an estimate about how the PUF works with respect to others of the same type. It is seen from fig. 5.3 that the percentage of aliasing bit is visually close to 50%, this is done by averaging all the percentages of each of the output bits for different challenges.

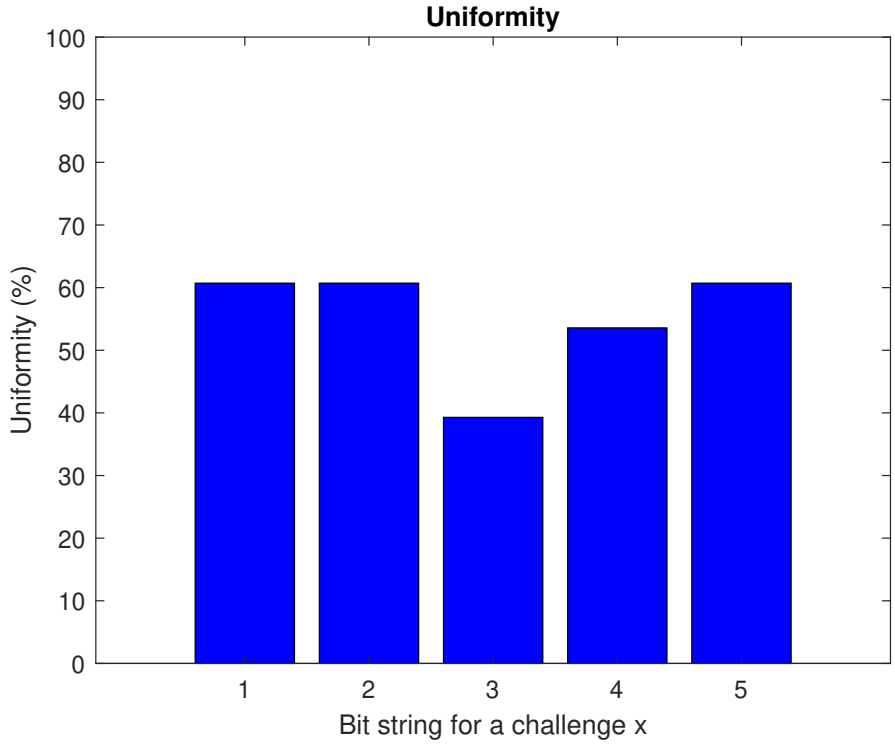


Figure 5.4: PUF mean uniformity for different challenges applied (n=8)

Uniformity is represented in Fig. 5.4, as in the previous case, an average value close to the ideal could be estimated; on the other hand it is important to clarify that a uniformity of 50% for all bits would lower the utility of the PUF since it would limit the possible valid combinations of an output. The important thing with these systems is to have values surrounding a value of 50% which means that the bitstings for different challenges are conformed by almost the same amount of 1 or 0.

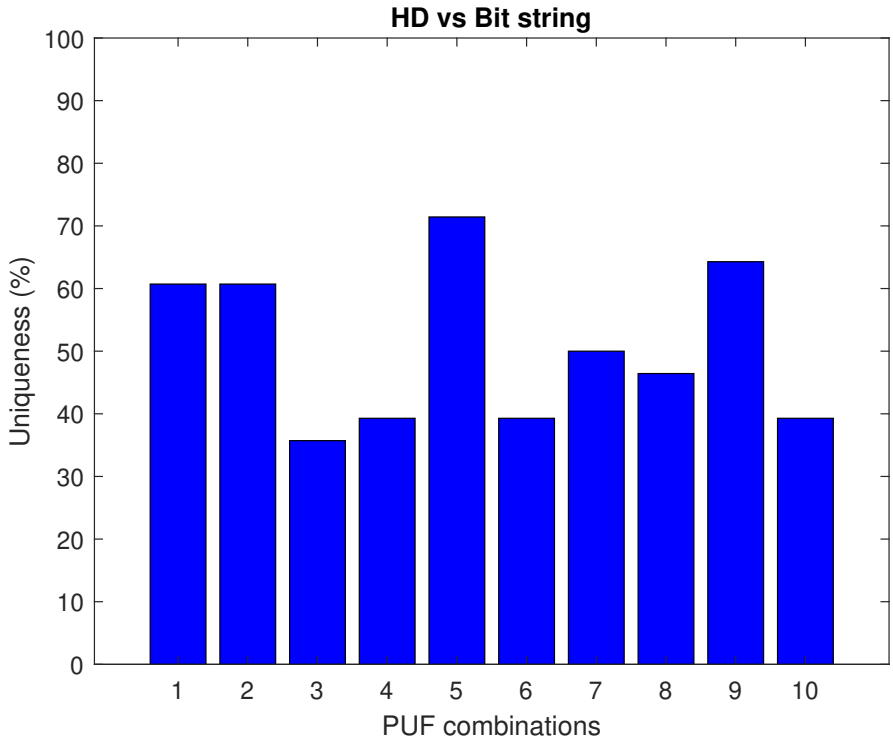


Figure 5.5: Mean uniqueness for different PUF instances (n=8)

In orded to see the uniqueness of the system, the hamming distance is plotted among all possible combinations of PUF instances, and for each of these combinations the percentage of uniqueness is calculated.

- **11 oscillators system**

An alternative system with 11 oscillators is implemented, to analyze the effect on the metrics when considering a bit larger system, it is important to take into account that the bitstring at the output also increases, which means greater security in the output bitstring. This scheme generates 55 bits, 27 bits more than the 8 oscillator scheme, a larger amount of bits can be achieved whether the system contains a higher number of oscillators; however several bits have a low entropy. A bit with low entropy is a non totally secure bit of the output. See fig 5.6, 5.7 and 5.8.

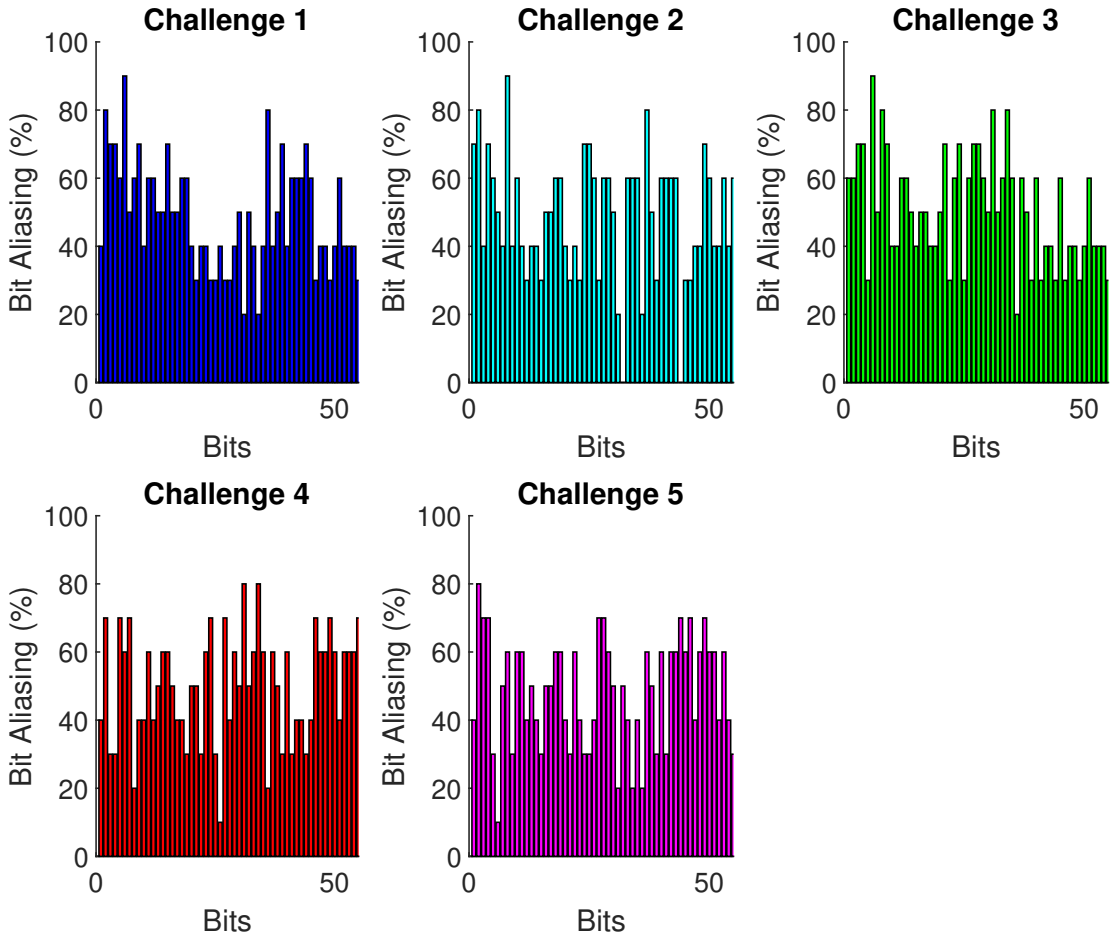


Figure 5.6: Bit Aliasing for different challenges applied (n=11)

As in the case where $n = 8$, the percentage of bit aliasing; see Fig 5.6 of each bit of the output is averaged, taking into account that for this scheme, the amount of bits increases up to $k = 55$ with an increase of only 3 oscillators .

With regard to uniformity Fig 5.7, it is observed that the percentage for each of the challenges applied remains quite uniform and close to 50 %, which is positive, however it should be clarified that this result comes from statistics; applying a large number of challenges to obtain the metrics is necessary, and most likely some challenge choices will increase the scattering of the bars on the graph.

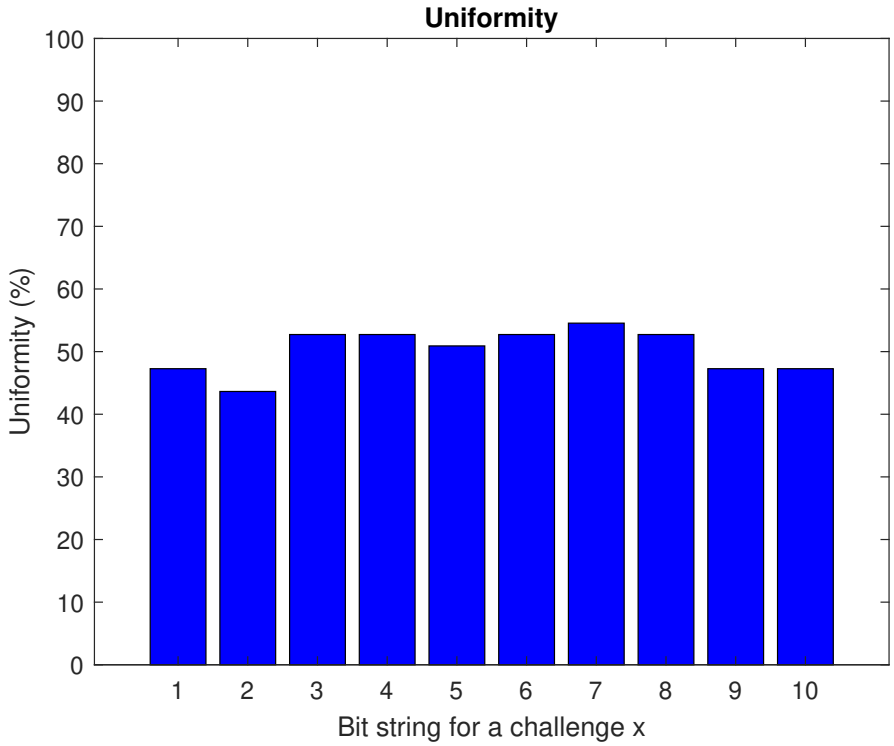


Figure 5.7: PUF mean uniformity for different challenges applied (n=11)

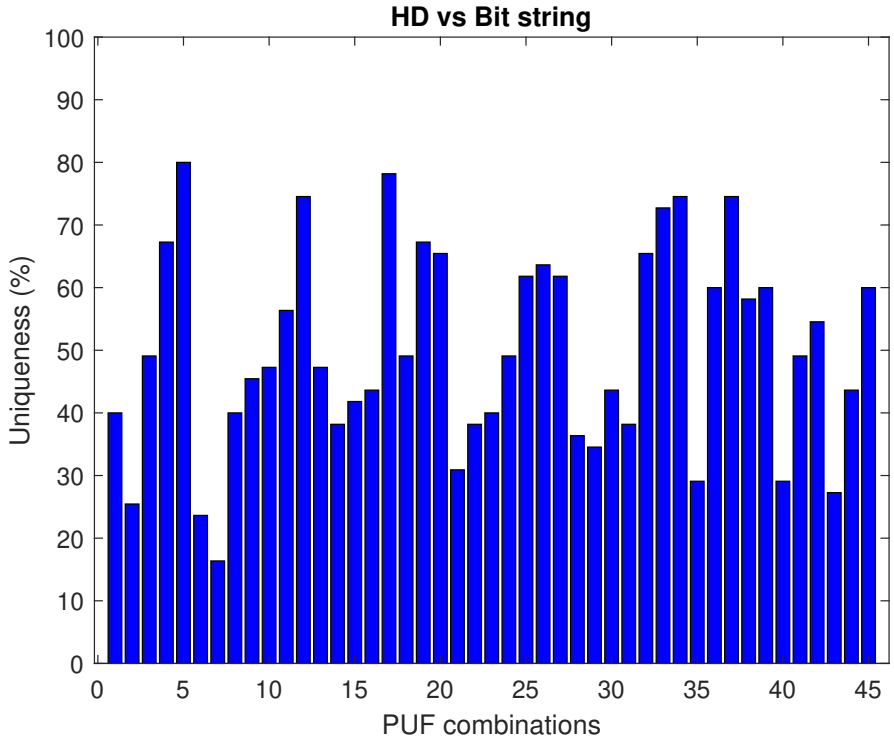


Figure 5.8: Mean uniqueness for different PUF instances (n=11)

For the uniqueness percentage Fig 5.8 it should be noted that the possible combinations between the 10 instances of PUF that were used with $n = 11$ to calculate the performance metrics make 45 combinations, which allows to find a better estimate than in the case of $n = 8$ for which only had 10 combinations. From here the importance of expanding the analyzes in this type of systems, in order to avoid systematic results that are subject to the random test challenges used.

In the case of the applied combinatorial method, of the 28 and 55 bits for each output only $log_2 n!$ are reliable bits that contribute to the entropy of the system (always dependent on the mapping algorithm used).
Since many of these used pairs of oscillators generate correlated bits, it would be very easy to know that if the frequency $f_a > f_b$ and $f_b > f_c$ then the frequency $f_a > f_c$, this correlation is unwanted.
This system is classified as weak PUF, although it generates an exponential number of CRPs $2^{\binom{n}{2}}$ large number of them are not reliable.

5.2.2 By pairs

A bench of n=50 different ring oscillators has been implemented to improve the entropy of the system at the cost of reducing the number of bits in the output. The challenge contains the same number of bits as the output, each of these bits represents the direction in which the pair of oscillators is taken to generate the corresponding bit. When using this mapping scheme it is evident that the keys generated are smaller, although in exchange for this there are highly secure bits, since there would be no correlation between each PUF run that can be discovered by any attacker. This algorithm generates an entropy of $\frac{n}{2} = k$. Fig 5.10, 5.11 and 5.12.

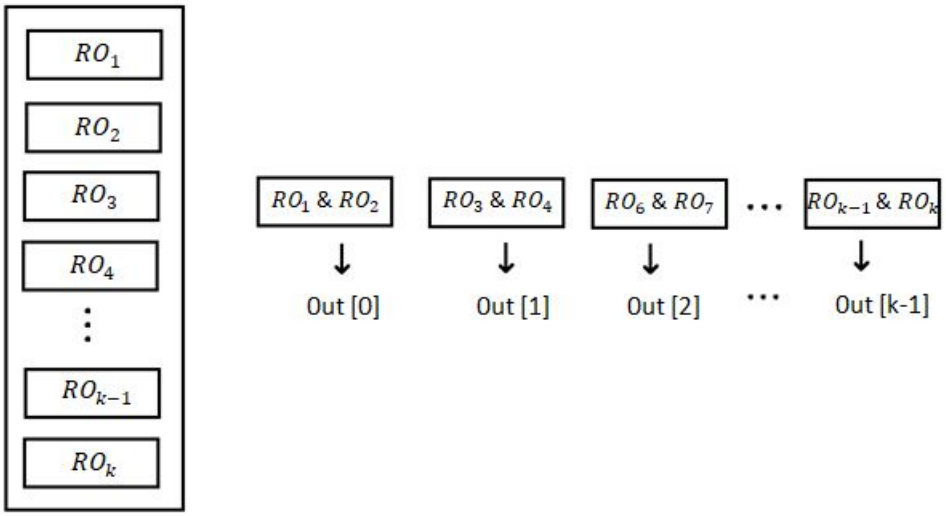


Figure 5.9: Pairing mapping by no repeatable pairs

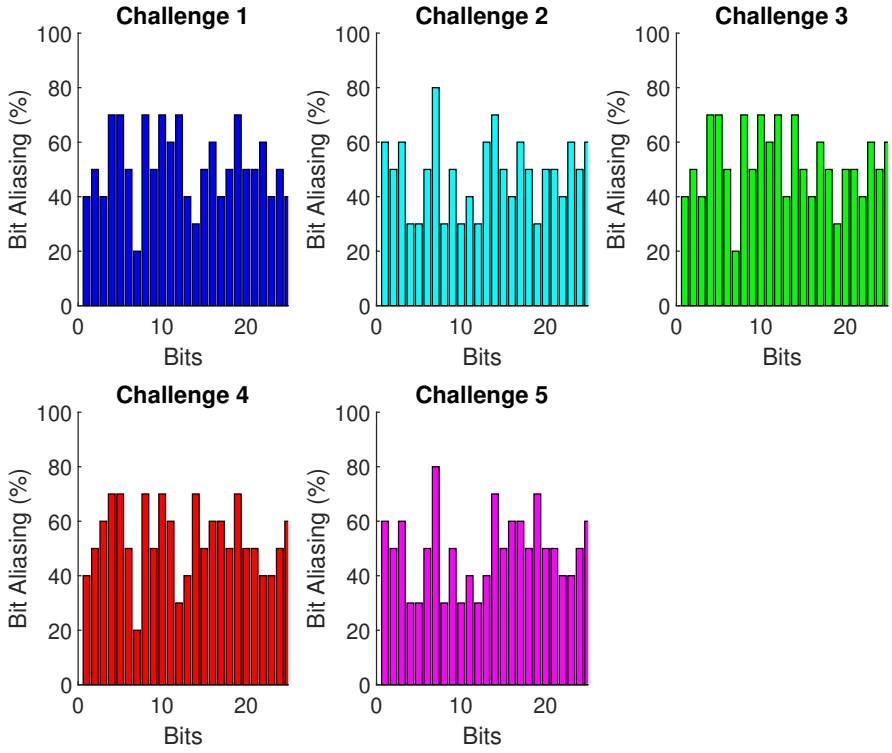


Figure 5.10: Bit Aliasing for different challenges applied (n=50)

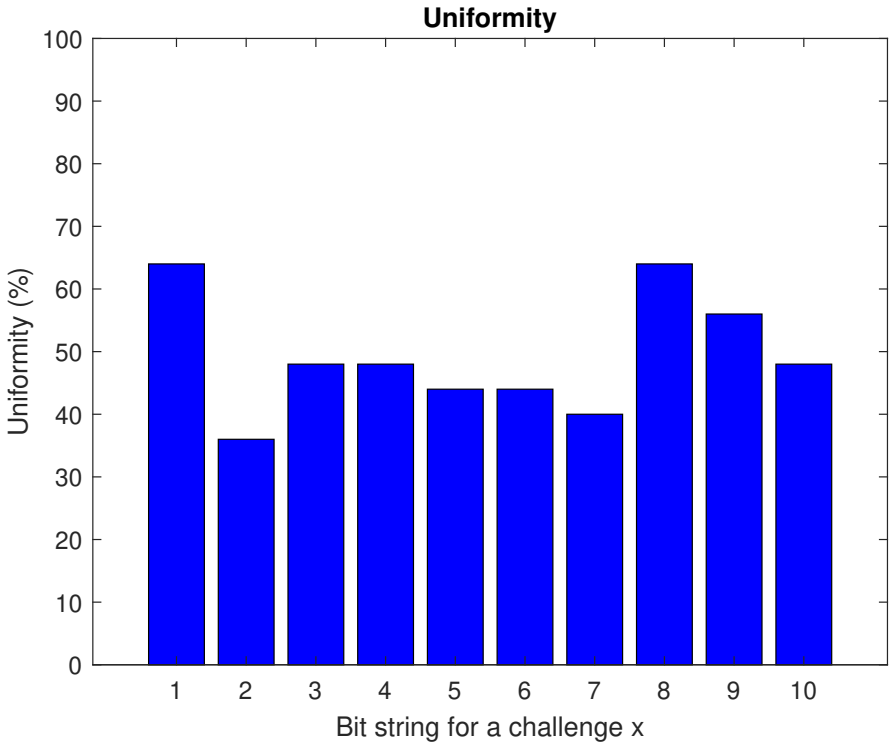


Figure 5.11: PUF mean uniformity for different challenges applied (n=25)

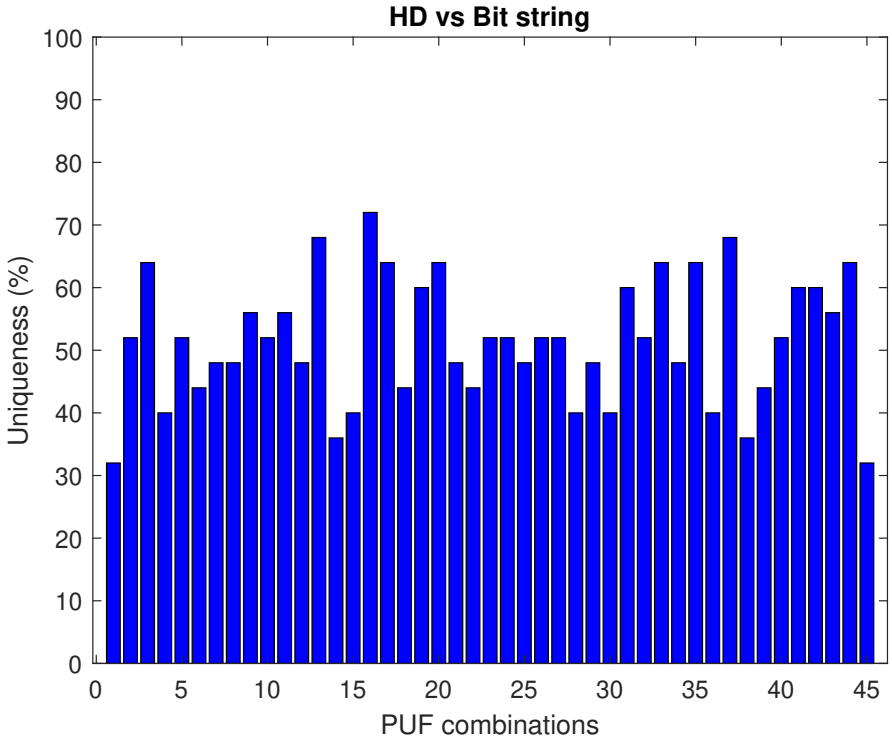


Figure 5.12: Mean uniqueness for different PUF instances (n=25)

A compilation of the obtained metrics is presented, which indicates that the values are quite close to the ideal.

	n	Uniformity%	BitAliasing%	Uniqueness%	MeanBit
n(n-1)/2	8	55	52.8571	50.7143	14.2
n(n-1)/2	11	50.1818	48.9091	50.0606	27.5333
n/2	50	51.60	49.20	51.4667	12.8667
Ideal	-	50	50	50	-

Table 5.1: Table of metrics for all the different algorithm of pairing

Environmental variations such as temperature, noise and polarization voltages are not taken into account throughout this simulation although it is known that they could have an effect that alters the PUF metrics, making it less reliable.

With no improvements in the capacity of CRPs a ring oscillator will be a weak PUF, however some works of researchers have focused their efforts on increasing the reliable CRPs generated from this configuration PUF [38].

5.3 Variability analysis to the system

The physical function from the system has been produced by the combination of different oscillators, as it has been mentioned previously, the resulting frecuency of the oscillators is random, for that reason the bitstring on the output is random as well. In order to study the PUF behavior to variations of the nominal value of any parameter, a test methodology has been proposed to quantify this variation in the performance metrics.

This methodology consists of several tests of variability in a single ring oscillator of the system. That is, each one has 5 memristors, it is subjected to variations in

some of the parameters (which directly affect the memristance in high frequencies) (4.7), the performance measures are calculated for each one of the values taken by the parameter in question, and finally a statistical study is made in order to find the amount of affectation that this variability had in the operation of the PUF. The test wanted to carry out will allow us to have a brief control over which PUF and even more, which oscillator will have affected its memristance. A flow chart shows up the methodology implemented is shown below.

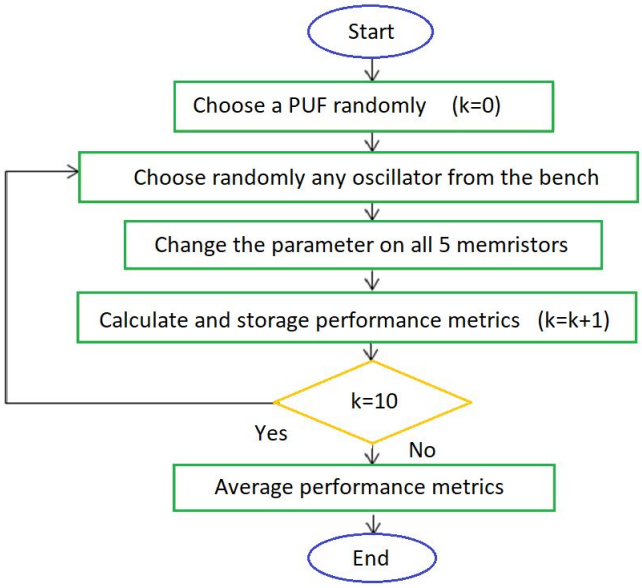


Figure 5.13: Variability methodology implemented

The oscillator extracted from the PUF is selected regardless of its nominal value parameters, the new value is assigned along the range of possible values for Xo, taking into account that it is required to explore which region of Xo can generate more randomness on the PUF.

We decided to consider affectations defined by an absolute uniform distribution in parameter Xo for the combinatorial PUF of $n = 8$, where Xo is the center value and it is varied by $+/-$ 0.1; 10 different values are used (one at the time) to apply the previously mentioned methodology.

1. X=AUNIF(0.5 , 0.1)

	Uniqueness (%)	Uniformity (%)	Bit-aliasing (%)
X - 0.08692	49,28571	48,57143	48,57143
X - 0.08460	49,28571	48,57143	48,57143
X - 0.06577	49,28571	48,57143	48,57143
X - 0.03719	49,28571	48,57143	48,57143
X + 0.1103	49,28571	47,8571	47,8572
X + 0.1145	49,28571	47,8571	47,8572
X + 0.5439	50	48,57143	48,57143
X + 0.7907	50	48,57143	48,57143
X + 0.9064	50	48,57143	48,57143
X + 0.9641	50	48,57143	48,57143

Table 5.2: metrics for the Xo parameter sweep centered at 0.5

2. X=AUNIF(0.8 , 0.1)

	Uniqueness (%)	Uniformity (%)	Bit-aliasing (%)
X - 0.08692	50,71429	47,14286	47,14286
X - 0.08460	50,71429	47,14286	47,14286
X - 0.06577	50,71429	47,14286	47,14286
X - 0.03719	50,71429	47,85714	47,85714
X + 0.1103	50,71429	47,85714	47,85714
X + 0.1145	50,71429	47,85714	47,85714
X + 0.5439	50,71429	47,85714	47,85714
X + 0.7907	49,28571	47,14286	47,14286
X + 0.9064	49,28571	47,14286	47,14286
X + 0.9641	49,28571	47,14286	47,14286

Table 5.3: metrics for the Xo parameter sweep centered at 0.8

3. X=AUNIF(0.2 , 0.1)

	Uniqueness (%)	Uniformity (%)	Bit-aliasing (%)
X - 0.08692	50	49,28571	49,28571
X - 0.08460	50	49,28571	49,28571
X - 0.06577	50	49,28571	49,28571
X - 0.03719	50	49,28571	49,28571
X + 0.1103	50	49,28571	49,28571
X + 0.1145	50	49,28571	49,28571
X + 0.5439	49,28571	48,57143	48,57143
X + 0.7907	49,28571	48,57143	48,57143
X + 0.9064	49,28572	48,57143	48,57143
X + 0.9641	50	49,28571	49,28571

Table 5.4: metrics for the Xo parameter sweep centered at 0.2

The results show a brief change in the behavior of the system, this effect is due to the internal configuration of each PUF. Each PUF contains 8 oscillators, the location of each of these directly affects the key generated.

If a challenge determines the selection of 2 oscillators to generate a bit, the alteration of the memristance in for instance oscillator placed at position 2, can cause a 0 or a 1 to be generated with greater probability. However, despite affecting the operation of the PUF, this effect is not large enough to make the system unusable.

Now, if for instance, the oscillator chosen to be swept is the oscillator 5 and its arrangement of memristances generated an oscillation of frequency f5 and the other 7 oscillators had a frequency f1, f2, ..., f7 respectively. If the variability in f5 (due to the changed parameter) is not large enough to cause f5 to exceed or be exceeded in frequency by another oscillator; no change in the performance of the PUF will

be presented, even the key generated by that PUF for a given challenge will not be altered at all.

On the other hand, if the variation of the parameter in oscillator 7 is significant enough to -clutter- the initial oscillator array, the PUF will be altered so that its measurements could vary and its response as well. Different tests were performed with a single oscillator subject to variability and their changes in Xo did not turn out to be an object of high sensitivity in the system. So it can be concluded that this PUF is strongly robust to this type of variation, however it is important to notice as well that a direct relationship to these effects of variability is due to the range of frecuencies selected for the oscillators. Nonetheless, these affectations, despite of not reducing the performance metrics, could have quite strong effects in applications such as dentification of integrated circuits, since when the set of CRPs of the system is altered, the fingerprints of these would be affected, ruining the data-base used for authentication.

5.4 Entropy variation due to temperature

It has already been mentioned up to this point that a ring oscillator PUF is robust under certain conditions to variations in parameters of memristance; however in many works it has been reported that process, voltage and temperature variations can strongly affect their applicability. The analog part of the system corresponds to the oscillators that change their frequency with these variations.
These variations could affect the analog part of the system in a different proportion, reducing its entropy and again strongly affecting the system response.

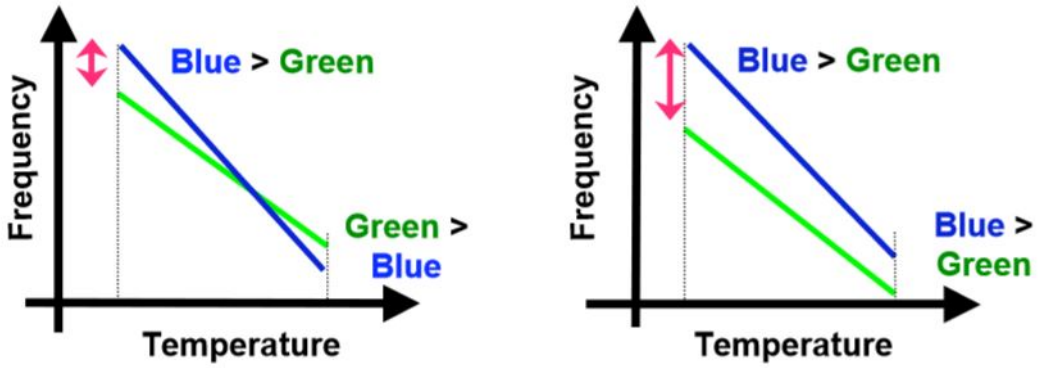


Figure 5.14: The effect of temperature on frequency, oscillators may flip when temperatures changes

This is not always undesirable, the random number generators can be implemented if a PUF with high variability is designed, if the bitstrings have a very very low entropy, the system would throw different output no matter what the challenge is. However, security applications require systems that are not greatly affected by these types of effects.

A study of variations of temperature and voltage for our system have been carried out in order to analyze if, under simulation conditions, the entropy of the system is affected. The following table shows the response of a PUF to the same challenge for different temperatures (only 20 bits are shown for aesthetic reasons in the document). This result allows us to see that for frequencies at which the oscillators operate, the considerations that were taken into temperature were not large enough to change the bitstring; which is good, since it allows concluding that once the PUF system counters and comparators are well designed, this type of alterations will not occur. Similar results are obtained for variations in voltage VDD, no entropy affectations are observed, probably because by using Spice simulators we change all the circuit operation at the same time while if it is implemented these variations occur while it is working, allowing to observe the changing bit error rate and the reduction of reliability.

PUF 100^oC	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	0	0
PUF 20^oC	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	0	0

Table 5.5: Response bitstring for different conditions of temperature

In order to also visually contrast the influence among environmental changes and frecuency, the fast fourier transform (FFT) is used, thus obtaining an idea of how variations in Xo produce changes in the frequency.

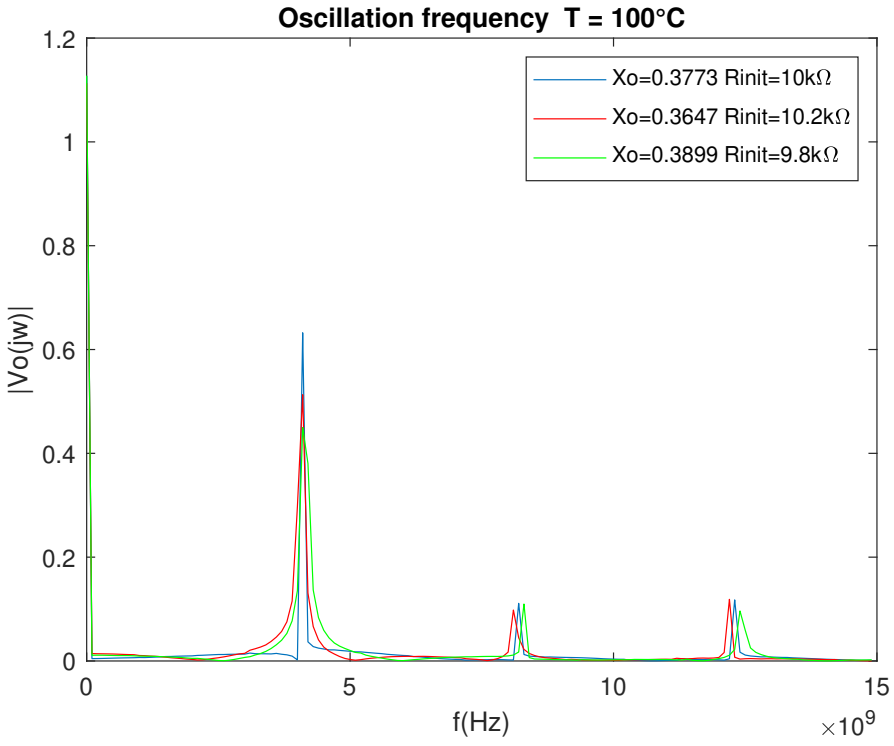


Figure 5.15: FFT for temperature of 100^oC

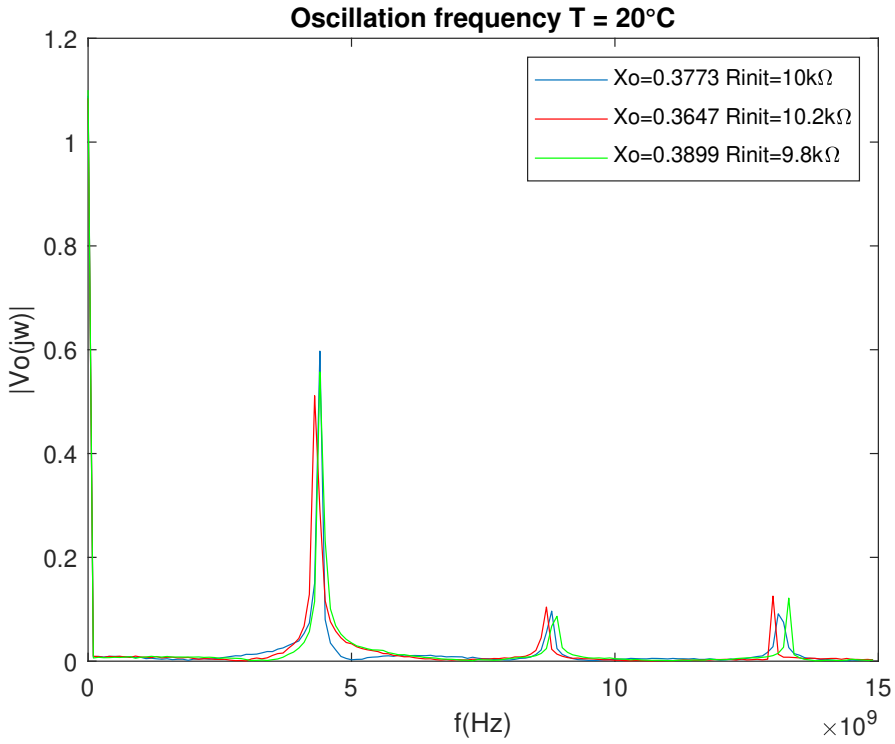


Figure 5.16: FFT for temperature of 20^{o}C

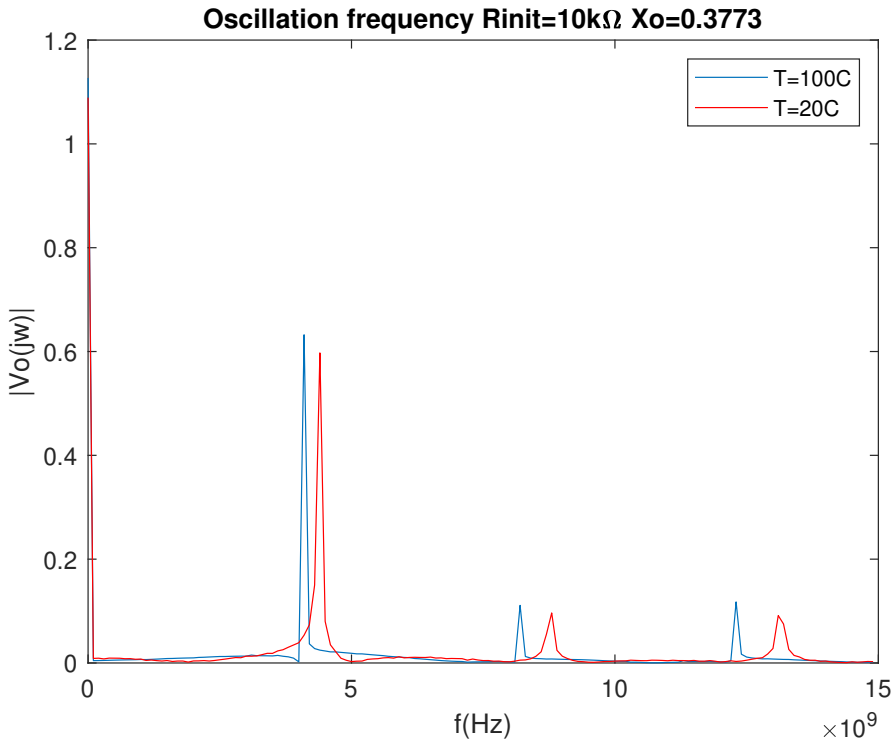


Figure 5.17: FFT for a single oscillator at difference frequencies

We intuitively suppose variations of 1.25% in memristance by changing the Xo value, 3 different values are given and the results shows up that even though small differences are notorius, we did not notice any flip that violates the tendence obtained in fig 4.12. These variations are evidenced more strongly in fig 5.17.

So we conclude from this that physical implementation of these kind of system is necessary in order to observe exactly how realiable the system is working under real conditions that most likely have this type environmental variations.

Chapter 6

Conclusions and future work

6.1 Conclusions

Memristors have unique properties (resistive switching) that make them strongly appropriate for electronic security applications, it is shown that they can be used in other working conditions without exploiting this property and still being applicable in electronic PUF circuits.

The feasibility of using functional or mathematical analytical models of memristors in hardware security schemes was demonstrated. The models were tested on a PUF based on ring oscillators that showed really good metrics close to the ideal value. Special emphasis was made in the analysis of the effects of the modification of memristor parameters on the performance metrics or on the overall behavior of the hardware security system. In addition to this, the effect of the temperature and the voltage on the resonance frequency of the oscillator in the ring is highlighted.

The functioning of the memristive PUF is based on the X_o that defines the value of the memrisance at high frequency.

6.2 Future work

- Incorporating the reported memristor models in other PUF schemes.
- Improve the memristor model used, including temperature as a parameter.
- Study of the feasibility of implementing a PUF topology in CMOS technology.

Bibliography

[1] Yansong Gao, Damith C Ranasinghe, Said F Al-Sarawi, Omid Kavehei, and Derek Abbott. Emerging physical unclonable functions with nanotechnology. *IEEE access*, 4:61–80, 2016.

[2] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference*, pages 9–14. IEEE, 2007.

[3] Roel Maes and Ingrid Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. In *Towards Hardware-Intrinsic Security*, pages 3–37. Springer, 2010.

[4] Shital Joshi, Saraju P Mohanty, and Elias Kougianos. Everything you wanted to know about pufs. *IEEE Potentials*, 36(6):38–46, 2017.

[5] Garrett S Rose, Nathan McDonald, Lok-Kwong Yan, and Bryant Wysocki. A write-time based memristive puf for hardware security applications. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 830–833. IEEE, 2013.

[6] Yansong Gao, Damith C Ranasinghe, Said F Al-Sarawi, Omid Kavehei, and Derek Abbott. mrpuf: A novel memristive device based physical unclonable function. In *International Conference on Applied Cryptography and Network Security*, pages 595–615. Springer, 2015.

[7] Arturo Sarmiento-Reyes, Juan Manuel Ugalde Franco, Yojanes Rodríguez-Velásquez, and JL Fernando Palomeque Loyo. Development of an operator-based fully analytical charge-controlled memristor model. In *2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–4. IEEE, 2018.

[8] Gabriel Hospodar, Roel Maes, and Ingrid Verbauwhede. Machine learning attacks on 65nm arbiter pufs: Accurate modeling poses strict bounds on usability. In *2012 IEEE international workshop on Information forensics and security (WIFS)*, pages 37–42. IEEE, 2012.

[9] Md Tanvir Arafin, Carson Dunbar, Gang Qu, N McDonald, and L Yan. A survey on memristor modeling and security applications. In *Sixteenth International Symposium on Quality Electronic Design*, pages 440–447. IEEE, 2015.

[10] Chip-Hong Chang, Yue Zheng, and Le Zhang. A retrospective and a look forward: Fifteen years of physical unclonable function advancement. *IEEE Circuits and Systems Magazine*, 17(3):32–62, 2017.

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

[12] Daihyun Lim, Jae W Lee, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, 2005.

[13] Mehrdad Majzoobi, Farinaz Koushanfar, and Miodrag Potkonjak. Lightweight secure pufs. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, pages 670–673. IEEE Press, 2008.

[14] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[15] Ulrich Rührmair and Daniel E Holcomb. Pufs at a glance. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 347. European Design and Automation Association, 2014.

[16] Meng-Day Yu, Richard Sowell, Alok Singh, David M'Raïhi, and Srinivas Devadas. Performance metrics and empirical results of a puf cryptographic key generation asic. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*, pages 108–115. IEEE, 2012.

[17] Le Zhang, Zhi Hui Kong, Chip-Hong Chang, Alessandro Cabrini, and Guido Torelli. Exploiting process variations and programming sensitivity of phase change memory for reconfigurable physical unclonable functions. *IEEE Transactions on Information Forensics and Security*, 9(6):921–932, 2014.

[18] Le Zhang, Zhi Hui Kong, and Chip-Hong Chang. Pckgen: A phase change memory based cryptographic key generator. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pages 1444–1447. IEEE, 2013.

[19] Keith Lofstrom, W Robert Daasch, and Donald Taylor. Ic identification circuit using device mismatch. In *2000 IEEE International Solid-State Circuits Conference. Digest of Technical Papers (Cat. No. 00CH37056)*, pages 372–373. IEEE, 2000.

[20] Sandeep S Kumar, Jorge Guajardo, Roel Maes, Geert-Jan Schrijen, and Pim Tuyls. The butterfly puf protecting ip on every fpga. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 67–70. IEEE, 2008.

[21] Yansong Gao, Chenglu Jin, Jeeson Kim, Hussein Nili, Xiaolin Xu, Wayne Burleson, Omid Kavehei, Marten van Dijk, Damith Chinthana Ranasinghe,

and Ulrich Rührmair. Efficient erasable pufs from programmable logic and memristors. *IACR Cryptology ePrint Archive*, 2018:358, 2018.

[22] Jorge Guajardo, Sandeep S Kumar, Geert-Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In *International workshop on cryptographic hardware and embedded systems*, pages 63–80. Springer, 2007.

[23] Jae W Lee, Daihyun Lim, Blaise Gassend, G Edward Suh, Marten Van Dijk, and Srinivas Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*, pages 176–179. IEEE, 2004.

[24] Ulrich Rührmair, Srinivas Devadas, and Farinaz Koushanfar. Security based on physical unclonability and disorder. In *Introduction to Hardware Security and Trust*, pages 65–102. Springer, 2012.

[25] Jiliang Zhang, Yaping Lin, Yongqiang Lyu, and Gang Qu. A puf-fsm binding scheme for fpga ip protection and pay-per-device licensing. *IEEE Transactions on Information Forensics and Security*, 10(6):1137–1150, 2015.

[26] Jiliang Zhang. A practical logic obfuscation technique for hardware security. *IEEE Transactions on very large scale integration (VLSI) systems*, 24(3):1193–1197, 2015.

[27] Leon Chua. Memristor-the missing circuit element. *IEEE Transactions on circuit theory*, 18(5):507–519, 1971.

[28] Leon O Chua and Sung Mo Kang. Memristive devices and systems. *Proceedings of the IEEE*, 64(2):209–223, 1976.

[29] Leon Chua. Device modeling via nonlinear circuit elements. *IEEE Transactions on Circuits and Systems*, 27(11):1014–1044, 1980.

[30] C Garling. Wonks question hp's claim to computer-memory missing link. *Wired. com, retrieved*, pages 09–23, 2012.

[31] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. A systematic method to evaluate and compare the performance of physical unclonable functions. In *Embedded systems design with FPGAs*, pages 245–267. Springer, 2013.

[32] Masoud Rostami, James B Wendt, Miodrag Potkonjak, and Farinaz Koushanfar. Quo vadis, puf?: trends and challenges of emerging physical-disorder based security. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014.

[33] Jesús Jiménez León. *Modelado comportamental de dispositivos basados en Ag/TiO2/ITO con efecto memristivo.* PhD thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2017.

[34] Arturo Sarmiento-Reyes, Luis Hernández-Martínez, Héctor Vázquez-Leal, Carlos Hernández-Mejía, and Gerardo Ulises Diaz Arango. A fully symbolic homotopy-based memristor model for applications to circuit simulation. *Analog Integrated Circuits and Signal Processing*, 85(1):65–80, 2015.

[35] Yogesh N Joglekar and Stephen J Wolf. The elusive memristor: properties of basic electrical circuits. *European Journal of Physics*, 30(4):661, 2009.

[36] M Salim-Maza. *Generación y Distribución de Señal de Reloj para Sistemas en Chip utilizando Anillos Interconectados Acoplados.* PhD thesis, Ph. D. dissertation, Instituto Nacional de Astrofísica, Óptica, y ..., 2005.

[37] Julius Teo Han Loong, Noor Alia Nor Hashim, Muhammad Saiful Hamid, and Fazrena Azlee Hamid. Performance analysis of cmos-memristor hybrid ring oscillator physically unclonable function (ro-puf). In *2016 IEEE International Conference on Semiconductor Electronics (ICSE)*, pages 304–307. IEEE, 2016.

[38] Mahshid Delavar, Sattar Mirzakuchaki, and Javad Mohajeri. A ring oscillator-based puf with enhanced challenge-response pairs. *Canadian Journal of Electrical and Computer Engineering*, 39(2):174–180, 2016.