0.3pJ/bit Machine Learning Resistant Strong PUF Using Subthreshold Voltage Divider Array

by

Abilash Venkatesh

05-10-2019

A thesis submitted to the
Faculty of the Graduate School of the
State University of New York at Buffalo
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical Engineering

Acknowledgements

This work wouldn't have been possible without the untiring encouragement, guidance, and support from my advisor Dr. Arindam Sanyal to whom I extend my deepest gratitude. Also, I would like to thank my committee member Dr. Jun Choi for his valuable inputs and suggestions. I thank everyone from Sanyal's Analog/Mixed Signal VLSI research group for patiently bearing all the troubles that I have caused and constantly supporting me throughout the work. I thank my family for their love and their efforts to keep me focused on completing my thesis. I thank my friends for their encouragement and well wishes.

Contents

List of Figures

List of Tables

Abstract

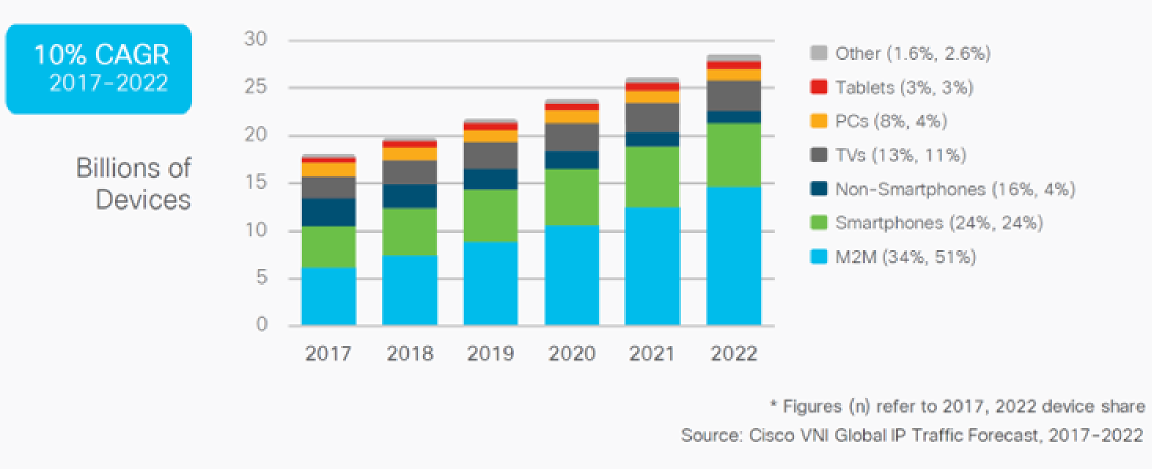
This thesis presents a novel architecture of subthreshold voltage divider based strong physical unclonable function (PUF). The PUF derives its uniqueness from random mismatch in threshold voltage in an inverter with gate and drain shorted and biased in subthreshold region. The nonlinear current-voltage relationship in subthreshold region also makes the proposed PUF resistant to machine learning (ML) based attacks. Prediction accuracy of PUF response with logistic regression, support vector machine (SVM) and random forest (RF) is close to 51%. A prototype PUF fabricated in 65nm consumes only 0.3pJ/bit, and achieves the best combination of energy efficiency and resistance to ML attacks. The measured inter and intra hamming distance (HD) for the PUF are 0.5026 and 0.0466 respectively.

Chapter 1

Introduction

1.1 Motivation

According to the white paper release from Cisco on Cisco Visual Networking Index (VNI) forecast (February, 2019) [1], The global IP traffic is expected to rise nearly to triple from 2017 to 2022. Also, Total Internet traffic has experienced dramatic growth in the past two decades, this gives rise to use of more connected devises. More than 20 years ago, in 1992, global Internet networks carried approximately 100 GB of traffic per day and ten years later, in 2002, global Internet traffic amounted to 100 Gigabytes per second (GB/second) and recently in 2017, global Internet traffic reached more than 45,000 GB/second.



* Figures (n) refer to 2017, 2022 device share

Source: Cisco VNI Global IP Traffic Forecast, 2017-2022

Figure 1.1: Cisco VNI Global IP Traffic Forecast, 2017-2022 [1]

Globally, devices and connections are growing faster (10% Compound Annual Growth Rate (CAGR)) than both the population (which is the rate of 1.0% CAGR) and Internet users (at the rate of 7% CAGR) shown in Fig.1.1. This trend is accelerating the increase in the average number of devices and connections per household and per capita around the world. Each year, various new devices in different form factors with increased capabilities and intelligence are introduced and adopted in the market. This growth tread is proliferating and never ceasing to saturation. A growing number of Machine-to-Machine (M2M) applications, such as smart meters, video surveillance, healthcare monitoring, transportation, and package or asset tracking, are contributing in a major way to the growth of devices and connections. According to the report By 2022, M2M connections will be 51% of the total devices and connections [1]. M2M connections will be the fastest-growing category, growing nearly 2.4-fold during the forecast period, at 19% CAGR, to 14.6 billion connections by 2022.

This develops the need to provide high speed data transfer to multiple devises without compromising on maximum security. As noted before, the increase of Machine-to-Machine interactions posses a new challenge in cyber-security. It is critical to make sure all the transferred data is secure since most of them contain vital information. There is increasing concern over the mode of securing these devises because of it's vulnerability to attacks. Traditionally the data from these secure devices are stored in Non-Volatile memory and they are processed through encryption algorithms. It is critical to note that there are many side channel attacks to extract this information since the data is physically stored. The next challenge would be to process all the data with the minimum power consumption. With the increases in M2M devices in future years, it is knowledgeable to enforce a method that delivers high efficiency. Thus, there is a need to develop a security device which can provide maximum security without the dependent on external memory source and also consume low power. If this system is made on-chip with minimum silicon footprint that consumes less power that would make this an ideal candidate. That introduces us to the family of PUFs. Si physical unclonable function (PUF) are lightweight hardware primitives that leverage random variations in CMOS integrated circuits to generate a unique key that can be used for authentication protocols or chip identification. They are proven to provide reliable security, consuming low power and occupying minimal Si foot print. Compared to standard cryptography algorithms like SHA or AES, Si PUFs provide on-chip security at a small fraction of power, thus making Si PUFs attractive candidate for secure internet-of-things (IoT) applications.

The first strong Si-PUF is an arbiter PUF [5] which uses variation in delay between two nominally identical paths to generate a 1-bit response. This gave rise to a novel hardware architecture which was soon gained popularity and was improvised into many architectures. Variants on arbiter PUF include using XOR-ing of arbiter PUF outputs [10] and using feed-forward paths [11] to inject non-linearity. The designed was made more non-linear to make sure they are more unpredictable thus ensuring more security. The security of PUF was developed on the base their core properties: *un-clonable* and *un-predictable*. It was stated that the PUF's function is very unique to its own and it is very difficult to derive a mathematical model [5]. PUFs increase physical security by producing volatile secrets that exist in a digital form only when the chip is powered on. In order to figure out the secret, the adversary has to mount an attack when the IC is running. An invasive attack must measure accurately the PUFs response without changing the properties or tampering the chip, which is very difficult, thus making it highly secure. The recent advancement in the of computer's processing power mathematical calculations as become quite faster [5]. This developed to new approach towards modelling the PUF using the computational power and advanced algorithms.

Even though the PUF was initially assumed to be highly secure and non-reproducible, In recent years [7, 12] have shown that most existing PUF models can be broken and PUF response predicted with high accuracy (90 ~ 99%) through the use of advanced machine learning (ML) models such as logistic regression or support-vector machine (SVM). Various modelling attacks were performed in early stages with minimal success but with the recent advancement in machine learning algorithms, modelling attacks on PUF have become more viable. The introduction of ML attacks pose a serious threat to security provided by PUFs. Recently there are few works published in ML-resistant PUF: strong PUF uses subthreshold current array [8] which has a strong non-linearity arising out of MOSFET subthreshold operation leading to ML prediction accuracy of 60%. The SRAM array of [9] consuming low power but exploiting the non-linearity developed form accessing the SRAM word line and read line, ML algorithms has ML prediction accuracy of only 89.4%. The work in [13] uses strong non-linearity of convergence time in a bi-stable ring arising out of variations in threshold voltages to limit prediction accuracy of ML attacks to 50%. The current array PUF though gives a very high ML immunity consumes more power compared to SRAM PUF. Thus our work is framed in such a way to establish a very high ML modelling attack-immunity while consuming low power compared to the existing PUF architectures.

In this work, we propose a subthreshold voltage-divider array based strong PUF which achieves simultaneous low energy consumption and high resistance to ML attacks.

1.2 Thesis Organization

The remaining part of thesis is organized as follows. An introduction about PUF including various PUF's architectures and its brief characterization is provided in chapter-2. In chapter-3 a detailed view on Strong PUF is provided, along with three more architectures and explaining its non-linearity. Chapter-4 introduces the modelling attacks made on PUF and gives a brief explanation about two important Machine learning algorithms that are used to model the PUFs. Chapter-5 explains two non-linearity PUF designs that are proven to be immune against machine learning modelling attacks. The proposed voltage divider array PUF is explained in chapter-6. The measurement results and analysis including comparison with other designs are established in chapter-7. Chapter-8 provides two more variants to the proposed architecture that brings about more non-linearity to the design and it's analysis are discussed. Finally, Chapter-9 briefs the conclusion for the thesis.

Chapter 2

Physically uncloanable functions

2.1 Introduction

The number of networked smart devices, programs, and information is constantly increasing which leads to an equally growing demand to ensure the security and reliability of these units. As they are pervasive in our daily lives, this issue has become a significant societal challenge since those data that are transmitted every second act as a digital copy of the individual [2]. One central task lies in realizing secure and reliable identification, authentication, and integrity checking of these systems. Traditional security methods based on secret digital keys often do not provide adequate solutions for this purpose. One major point of vulnerability relates to their hardware implementations and key storage: A whole host of attacks for extracting, estimating, or cloning secret keys that are stored digitally in nonvolatile memory have been developed and reported over the past several years [2]. The situation is especially problematic for embedded and mobile low power devices with a small form factor, where the adversaries can often gain full and direct access to the device. For many FPGA-based re-configurable devices, which are increasingly growing in market share, the permanent storage of secret keys can be a problem: Integrating secure nonvolatile memory (NVM) on FPGAs incurs additional costs and fabrication overhead and, thus, it is often not included. Therefore, keys have to either be stored in external memory, where they are highly vulnerable, or an additional back-up battery to power on-chip volatile storage must be used, which increases cost and system complexity. [2]

Over recent years, this approach has taken interest, which is based on the inherent, hard-to-forge and unique disorder of physical objects. It constitutes a promising alternative which can address the standing challenges of classical security that were described earlier. Two major classes of disorder-based security systems that have been proposed are Unique Objects (UNOs) and Physical Unclonable Functions (PUFs). A Unique Object is a physical system that, upon measurement by an external apparatus, exhibits a small, fixed set of inimitable analog properties that are not similar to any other objects. It shall be impossible to intentionally fabricate a second object with the same properties, even if the properties and exact structure of the original object are known. Such properties can be referred to as the fingerprint of a unique object for obvious reasons.We discuss several media that exhibit such unique disorder, including paper, fibers, magnetic disks, radio-wave scatterers, and optical tokens. [2]

PUFs are the second important class of disordered systems that can be employed for reliable identification, authentication, key storage, and other security tasks. The term and acronym PUF for denomination of this class first appeared in [14]. In a nutshell, a PUF is a disordered physical system S that, when interrogated by a challenge (or input, stimulus) denoted by C_i, generates a unique device response (or output) denoted by RC_i. This response shall depend on the applied challenge and on the specific disorder and device structure of the PUF. The unclonability requirement in the PUF definition is that it should be intractable for an adversary with physical access to create a physical or software clone of a PUF. Both the challenge-response pairs of PUFs and the fingerprints of Unique Objects have the purpose of uniquely identifying any device with high probability. In order to realize this in practice, we need stable repeated measurements, and must be able to cope with noise and varying operational conditions.

Two important metrics that are typically applied to categorize the uniqueness and robustness of PUF responses are inter- and intra- device distances. *Inter*-device distance is often quantified as the average Hamming distance between the responses to the same challenge obtained from two different PUFs, or the average distance between the fingerprints of two unique objects measured in the same conditions. *Intra*-device distance is the average Hamming distance between the responses to the same challenge applied at different times and environmental conditions to the same PUF. Ideal PUFs should lead to large inter-device (50%) and small intra-device (0%) distances. Another key requirement for PUFs is the entropy of the resulting responses or fingerprints. The entropy quantifies the number of independent IDs that can be generated by the same device architecture. Fig.2.1 shows various categories on how the the classification is made for security based on physical disorder given by [2].

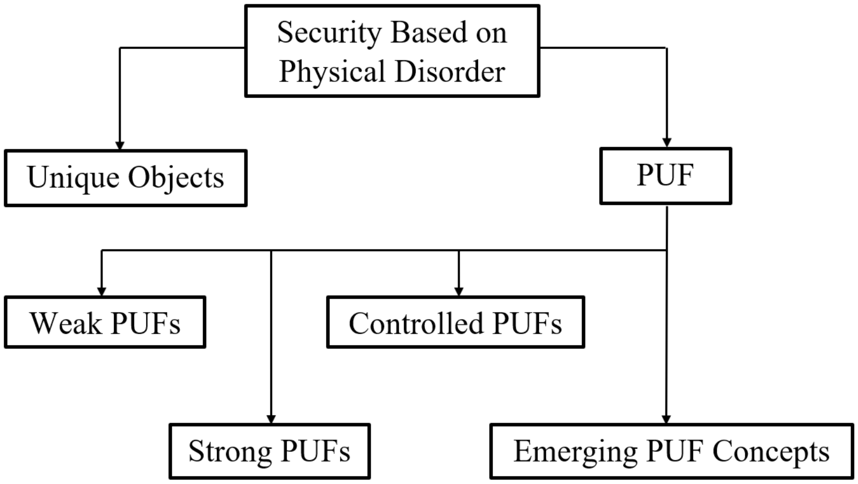


Figure 2.1: Categories of security provided based on physical disorder [2]

2.2 Weak PUF

One class of Physical Unclonable Functions based on inherent device variations are Weak PUFs [3]. They exploit the disordered, unique, internal structure of the underlying fabric as a nonvolatile memory for storing the secret keys [15]. In an ideal case, the volatile keys generated by Weak PUFs upon power-up cannot be determined by external and invasive attacks due to construction or tamper-proof properties of the pertinent structure. Weak PUFs are also known under the name of Physically Obfuscated Keys (POKs). The Weak PUF has limited number of Challenge-response pairs (CRPs) in contrast to the Strong PUFs. [2]

1.*Challenge-Response Pairs*: A Weak PUF can be interrogated by one (or a very small number of) fixed challenge(s) C_i, upon which it generates response(s) RC_i that depends on its internal physical disorder.

2.*Key Derivation*: The response(s) RC_i from a Weak PUF is (are) exploited by the device for deriving a standard digital key that can be used for security applications.

3.*Practicality and operability*: The generated response RC_i should be sufficiently stable and robust to environmental conditions and multiple readings.

2.2.1 ICID PUFs

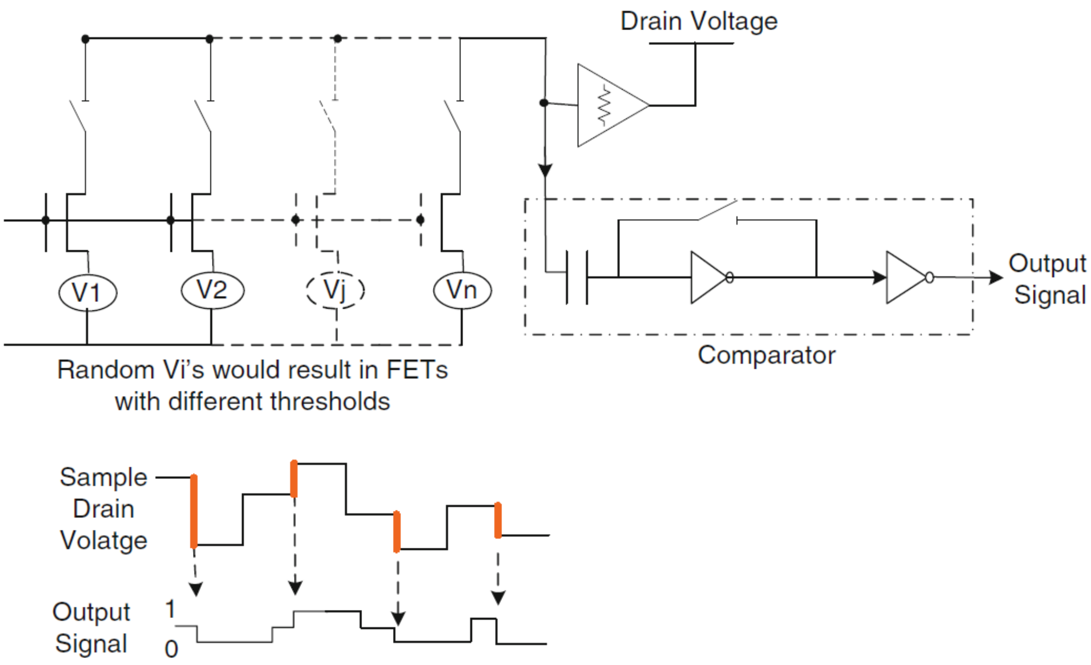


Figure 2.2: Array of ICID transistors producing a sequential random voltage proposed in [3]

ICID is the first proposed and designed circuit structure for generating a Weak PUF (or random chip ID) based on process variations [3]. They devised an array of addressable MOSFETs (shown in Fig.2.2), with common gate and source and sequentially selected drains driving a resistive load. Because of device threshold voltage mismatches (resulting from process variation) the drain currents are randomly different. Therefore, at each die, a unique sequence of random voltages would be generated at the load. ICID exploits these unique sequences of random but repeatable voltages to construct unique identification. In $0.35um$ technology, the authors reported about 10% false positive and false negative results for repeating random bits on their test circuits. Identification capability can be improved by increasing the bit length.

2.2.2 Physically Obfuscated Keys

Under the name of a Physically Obfuscated Key (POK), Gassend proposed a type of Weak PUF that was built from the first integrated Strong PUF [cite]. The POK/Weak PUF would only utilize one (or a small subset) of all possible challenges for a Strong PUF. This allows using them exactly as a digital key that is more resistant to physical attack, because it extracts its information from a complex physical system. [16]

2.3 Strong PUF

Immediately after the introduction of Weak PUFs or POKs, a second class of PUFs was put forward [14]. They have later often been referred to as Strong PUFs. In a nutshell, a Strong PUF is a disordered physical system with a very complex inputoutput behavior that depends on its disorder. The system must allow very many possible inputs or challenges, and must react with outputs or responses that are a function of the applied challenge and of the specific disorder present in the system. The input/output behavior should be so complex that it cannot be imitated numerically or by any other device. More specifically, a Strong PUF is a disordered physical system S with the following features: [2]
1.*Challenge-Response Pairs*: The Strong PUF can be interrogated by challenges C_i, upon which it generates a response RC_i that depends on its internal physical disorder and the incident challenge. The number of CRPs must be very large; often (but not always) it is exponential with respect to some system parameter, for example with respect to the number of components used for building the PUF.
2.*Practicality and operability*: The CRPs should be sufficiently stable and robust to environmental conditions and multiple readings.
3.*Access mode*: Any entity that has access to the Strong PUF can apply multiple challenges

to it and can read out the corresponding responses. There is no protected, controlled or restricted access to the PUFs challenges and responses.

4.*Security*: Without physically possessing a Strong PUF, neither an adversary nor the PUFs manufacturer can correctly predict the response to a randomly chosen challenge with a high probability. This shall hold even if both parties had access to the Strong PUF at an earlier time for a significant period, and could make any reasonable physical measurements on the PUF, including (but not limited to) determination of many CRPs.

More architectures from Strong PUF design will be discussed in chapter-3.

2.4 Controlled PUF

Let us start by specifying the notion of a Controlled PUF: A Controlled Physical Unclonable Function (CPUF) is a PUF that has been bound with an algorithm in such a way that it can only be accessed through a specific Application Programming Interface (API). The main problem with (uncontrolled) Strong PUFs is that anybody can query the PUF for the response to any challenge. To engage in cryptography with a PUF device, a user who knows a CRP has to use the fact that only he and the device know the response to the users challenge. But to exploit that fact, the user has to tell the device his challenge so that it can get the response. The challenge has to be told in the clear because there is no key yet. Thus a man in the middle can hear the challenge, get the response from the PUF device and use it to spoof the PUF device. Clearly, the problem in this attack is that the adversary can freely query the PUF to get the response to the users challenge. By using a CPUF in which access to the PUF is restricted by a control algorithm, this attack can be prevented. The API through which the PUF is accessed should prevent the man-in-the-middle attack we have described without imposing unnecessary limitations on applications.

While the details of various CPUF APIs are beyond the scope of this thesis, useful APIs have been developed [4] that satisfy the following properties: [2]

1.*Access Control*: Anybodywho knows a CRP that nobody else knows, can interact with the CPUF device to obtain an arbitrary number of other CRPs that nobody else knows. Thus users are not limited to using a small number of digital outputs from the PUF. Moreover, if one of these new CRPs was revealed to an adversary, transactions that use the other CRPs are not compromised. This is analogous to key management schemes that use session keys derived from a master key.

2.*Secret Sharing*: Anybody can use a CRP that only they knowto establish a shared secret with the PUF device. Having a shared secret with the PUF device enables a wide variety of standard cryptographic primitives to be used.

3.*Control Algorithm*: The control algorithm is deterministic. Because hardware random number generators are sensitive and prone to attack, being able to avoid them is advantageous.

4.*Cryptographic Primitive* The only cryptographic primitive that needs to be built into the control algorithm is a collision resistant hash function. All other cryptographic primitives can be updated during the lifetime of the CPUF device.

By selecting an appropriate API, a CPUF device can be resistant to protocol attacks. With careful design, Optical and Silicon PUFs can be made in such a way that the chip containing the control logic is physically embedded within the PUF: the chip can be embedded within the bubble-containing medium of an Optical PUF, or the delay wires of a Silicon PUF can form a cage on the top chip layer. This embedding should make probing of the control logic considerably more difficult, as an invasive attacker will have to access the wires to be probed without changing the response of the surrounding PUF medium.

The PUF and its control logic have complementary roles. The PUF protects the control logic from invasive attacks, while the control logic protects the PUF from protocol attacks. This synergy makes a CPUF far more secure than either the PUF or the control logic taken independently. Figure 2.3 demonstrates an example architecture of how a controlled PUF can be used for improving a PUF. A random hash function is placed before the PUF to prevent the adversary from doing a PUF chosen challenge attack. So a model-building adversary is prevented from selecting challenges that allow him to extract the PUF parameters. To ensure response consistency, an Error Correcting Code (ECC) is used. An output random hash function is used to decorrelate the response from the actual physical measurements, and therefore rendering a model-building adversarys task even harder.

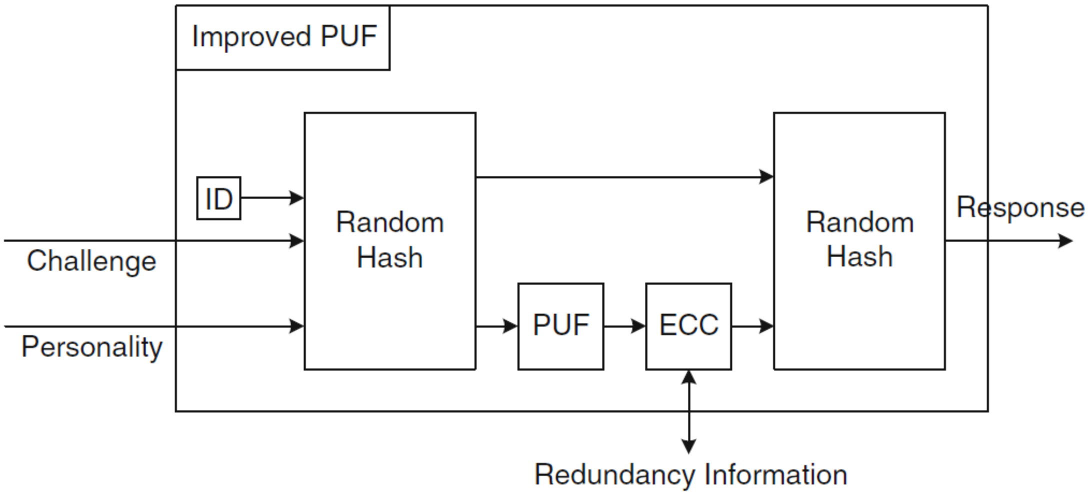


Figure 2.3: Controlled PUF architecture [4]

2.5 Emerging PUF Concepts

There are a number of new concepts that have emerged in the area of PUFs, and the pace of innovation is rapid. They cannot be categorized in the above mentioned groups. Two such interesting designs (concepts) are discussed here.

2.5.1 Quantum Readout PUFs

[17] proposed modifying the challenge-response mechanism of a PUF with quantum states, called a Quantum Readout PUF. The properties of the quantum states prevent an adversary from intercepting the challenges and responses without modifying them. Thus, there is no need for a trusted location for bootstrapping. However, no proof-of-concept implementation or practical architecture for this structure has been proposed to date. Finally, interfacing the quantum readout device to the regular PUF is likely a challenge.

2.5.2 SHIC PUFs

Super-High Information Content, abbreviated SHIC PUFs [18]. SHIC PUFs are Strong PUFs whose large number of CRPs are pairwise independent in an information-theoretic sense. Unlike other Strong PUFs, this allows them to become independent of computational assumptions in their security. The price they pay is a relatively large area consumption and slow read-out speed on the order of 102 to 104 bits per second. SHIC PUFs are unlikely to be used in low-cost commercial applications in the near future, since there are other, more favorable solutions to this end. But they represent an intriguing theoretical tool, since they are a variant of Strong PUFs with information-theoretic security.

Chapter 3

Strong PUF

In 2002, Pappu [19] suggested an optical system as the historically first PUF. It consists of a laser beam, which is directed at a transparent scattering token comprising of many randomly distributed scatterers. The laser light is scattered multiple times in the token and interferes constructively and destructively with itself. This leads to an interference pattern of bright and dark spots on a subsequently placed CCD. This pattern sensitively depends not only on the location of the scatterers in the token but also on the angle and point of incidence of the laser light (and on other parameters of the setup). The angle and point of incidence of the laser beam are usually regarded as the challenge of this PUF, while the interference pattern (or a suitably chosen image transformation of it) is interpreted as its response. This optical Strong PUF offers high internal complexity and security. On the downside, it cannot be integrated easily into an electronic microsystem, and requires an external, precise readout apparatus. Relatively soon afterward, integrated, electrical candidates for Strong PUFs have been suggested. One important example is the so-called Arbiter PUF [5], which exploits the natural variations in the runtime delays of integrated circuits.

Strong PUFs offer an enormous number of CRPs, often scaling exponentially with the required IC area. Despite their small response space, mostly $n = 1$, architectures are typically able to provide a large challenge space, for example, $m = 128$. Figure 3.1 shows total number of input challenge bits and the maximum possible challenge developed . Therefore, they might greatly exceed the need for secret key generation and have been promoted

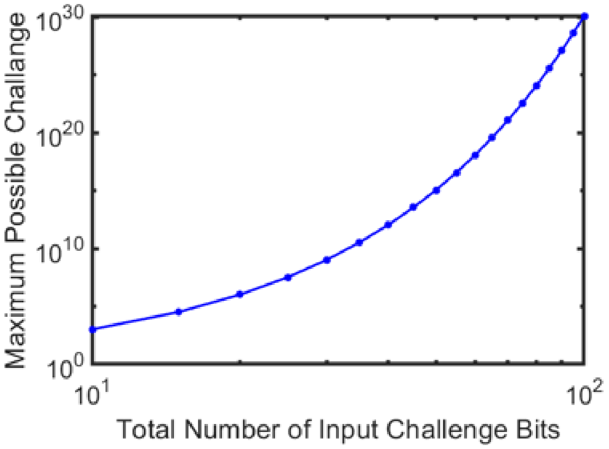


Figure 3.1: Exponential availability of CRPs in Strong PUF

primarily as lightweight authentication primitives. The most famous example is the arbiter PUF [5]. However, correlations among CRPs are severe, as these are not only spatial in nature but also induced by the functional behavior. Therefore, unprotected exposure to the PUF might enable so-called modeling attacks.

3.1 Arbiter PUF

Almost simultaneously to optical PUFs, the first integrated electrical Strong PUFs including Arbiter PUFs were put forward in [5]. Unlike optical PUFs, silicon PUFs do not require external measurement equipment. They are based on the runtime delay variations in electrical circuits. In one implementation, an electrical signal is split into two parallel signals, which race against each other through a sequence of k electrical components, for example, k multiplexers. This architecture is shown in Fig. 4.10. As shown in the figure, the challenges are applied to the selectors of the multiplexers. The exact signal paths are determined by these challenge bits $b_1, \ldots.., b_k$ applied at the multiplexers. At the end of the k components, an arbiter element decides which of the two signals arrived first and correspondingly outputs a zero or a one, which is regarded as the systems response.

It was clear from the beginning that these first electrical candidates were prone to modeling attacks as mentioned in [7]. Attacks using machine learning algorithms have been carried out. In these attacks, the adversary collects many challenge-response pairs

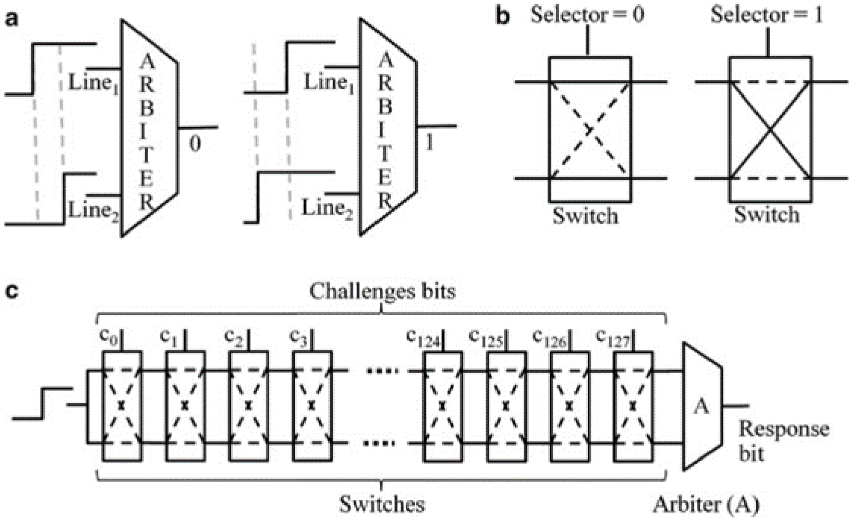


Figure 3.2: (a) Demonstration of an arbiters operation: the relative time of signal arrival at Line1 and Line2 would determine the value of the output bit; (b) Demonstration of a selectors operation: the selector bit would decide if the top and bottom lines continue in the same order, or they switch places; (c) An arbiter PUF with 128 challenge bits $c_0, ..., c_{127}$ applied as the selectors to the switches. The switch selectors dynamically configure two parallel paths with random delay differences that would form the response generated by the arbiter [5])

(CRPs), and uses them to derive the runtime delays occurring in the sub-components of the electrical circuit. Once they are known, simple simulation and prediction of the PUF becomes possible, breaking its security. More details about the modeling will be explained in chapter-4. One reason why these attacks worked so well lies in the fact that plain Arbiter PUFs have relatively simple linear models, in which the delay of each of the two signals can be approximated as the linear sum of the delays in the sub-components. Later, architectures were developed to introduce more non-linearity into the design.

3.2 XOR Arbiter PUF and Feed Forward Arbiter PUF

The earlier issues naturally led to the introduction of nonlinear electrical PUFs, for example, XOR arbiter PUFs, Lightweight Secure PUFs, and Feedforward Arbiter PUFs [10], [5], [11]. In an XOR arbiter PUF, multiple arbiter outputs are XORed to form a response. In Fig.3.3, an example is shown where two arbiter outputs are XORed. In the Feedforward Arbiter

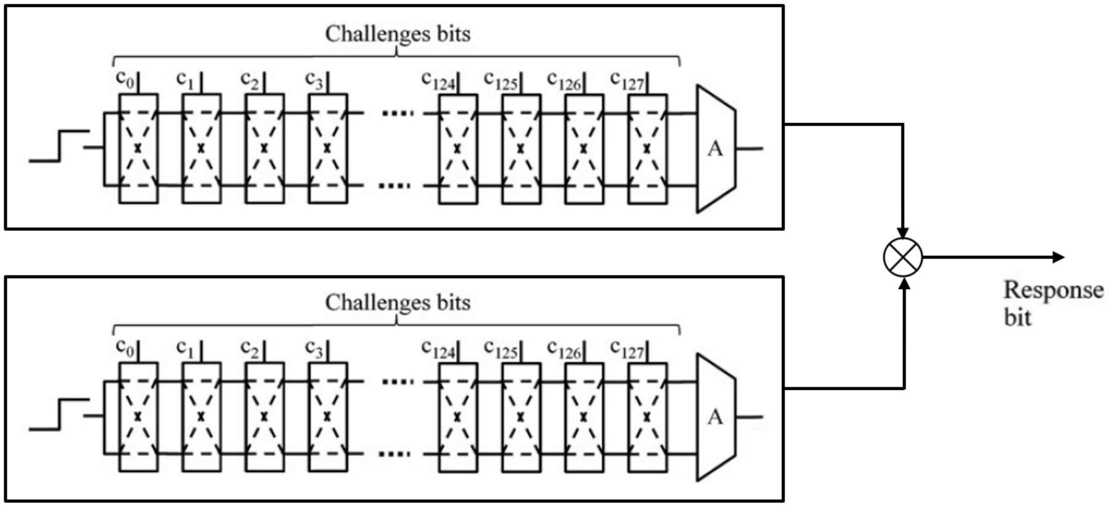

Challenges bits
c_0 c_1 c_2 c_3 .... c_{124} c_{125} c_{126} c_{127}
A
Challenges bits
c_0 c_1 c_2 c_3 .... c_{124} c_{125} c_{126} c_{127}
A
Response bit

Figure 3.3: XOR Arbiter PUF

PUF, the output of intermediate multiplexer(s) on the signal paths are input to so called Feedforward arbiter(s). The Feedforward arbiter output is then fed to the input of another multiplexer forward on the signal path. In Fig.3.4, an example of a Feedforward arbiter structure is shown. All of the aforementioned structures employ the basic Arbiter PUF architecture, but refine its architecture by introducing additional, nonlinearities. These structures showed a significantly higher resilience against machine learning attacks, but still could be attacked up to a certain level of size and complexity [20]. Arbiter PUFs and their variants have been shown to have small and stable integrated electrical implementations and have been commercialized [21].

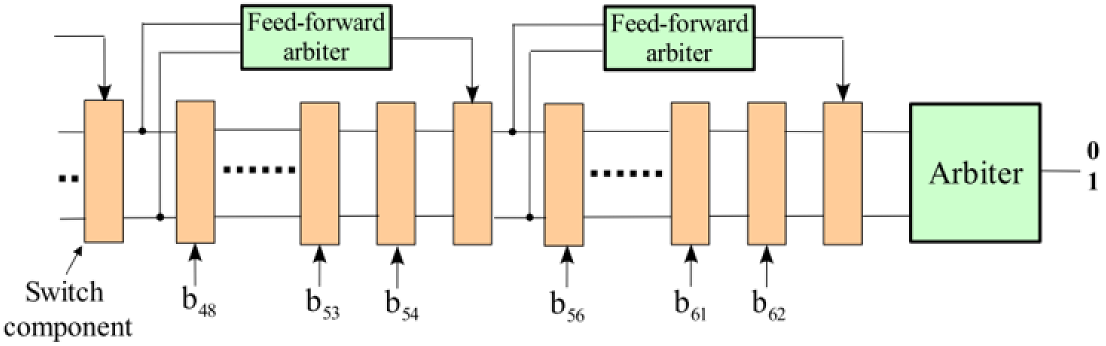


Figure 3.4: Feedforward Arbiter PUF

The Feedforward PUF shows more immunity against modelling attacks when compared to XOR PUF due to its non-linear delay path [7]. We can notice that many switch components are being activated using the response that is developed form the function of exciting delay lines through the Feed Forward Arbiters.

Chapter 4

Machine Learning Modelling Attacks

The promise of using PUF for authentication due to its unique properties of *unclonability* and *unpredictability* is debatable in recent times due to various modeling attacks based on machine learning (ML) algorithms. While ML algorithms are not new, the recent advances in computing power has enabled mounting of complicated ML attacks on PUF cells. For a given number of CRPs the attack is successful when the PUF's complex functions are digitally cloned, providing high accurate predictions for the response developed through ML algorithms from unknown challenges. [12] shows that use of support-vector machine (SVM) attack on the well-known arbiter PUF [5] can predict PUF response with an accuracy >90% and the prediction accuracy improves as the attacker has access to more CRPs. [7] shows successful modeling attacks (prediction accuracy of 99%) on various PUF architectures using logistic regression (LR) algorithm. [20] shows arbiter PUF and 2-XOR arbiter PUF are broken through SVM model with accuracy >95% and >80% respectively. [22,23] show an accuracy of >95% and >97% for prediction when ML modeling attacks based on evolution strategies (ES) were made against current-based PUFs and arbiter-PUFs respectively. [24, 25] employs artificial neural network (ANN) based ML modeling attack on feed forward PUFs and 64bit/128bit XOR PUFs resulting in prediction accuracy of >84% and >98% respectively. These cases show that PUFs can be broken through modeling attacks using

ML algorithms.

1) *Strong PUFs*: Strong PUFs are PUFs with very many possible challenges and a complex input-output relation. They are the PUF class for which our modeling attacks have been designed originally, and to which they are best applicable. The reason is that Strong PUFs usually have no protection mechanisms that restrict Eve applying challenges or in reading out their responses. Their responses are usually not postprocessed on chip in a protected environment. Most electrical Strong PUFs further operate at frequencies of a few MHz. Therefore even short physical access periods enable Eve to read-out and collect many CRPs. Another potential CRP source is simple protocol eavesdropping, for example on standard Strong PUF-based identification protocols, where the CRPs are sent in the clear. Please note that both eavesdropping on responses as well as physical access to the PUF is part of the established, general attack-model for PUFs. Once a predictive model for a Strong PUF has been derived, the two main security features of a Strong PUF no longer hold: The PUF is no longer unpredictable for parties that are not in physical possession of the PUF; and the physical unclonability of the PUF is overcome by the fact that the digital simulation algorithm can be cloned and distributed arbitrarily. Any Strong PUF protocol which is built on these two features is then no longer secure. [7]

2) *Controlled PUFs*: Controlled PUFs are a second PUF type, which consists of an underlying Strong PUF with a surrounding control logic. The challenge-response interface of the Strong PUF is not directly accessible, but is protected by the logic. Any challenges applied to the Controlled PUF are preprocessed by the logic before they are input to the Strong PUF, and any responses of the Strong PUF are postprocessed by the logic before they are being output by the Controlled PUF. Both the pre- and postprocessing step can add significantly to the security of a Controlled PUF. For any adversary that is restricted to noninvasive CRP measurement, Controlled PUFs successfully disable modeling attacks if the control logic uses a secure one-way hash over the outputs of the underlying Strong PUF.We note that this requires internal error correction of the Strong PUF outputs inside the Controlled PUF, since they are inherently noisy. Furthermore, it introduces a new, additional presumption, namely the security of the applied one-way hash function. Successful application of the techniques to a Controlled PUF only becomes possible if data

can be extracted from the internal probe, digital response signals of the underlying Strong PUF on their way to the control logic. Even though this is a significant assumption, probing digital signals is still easier than measuring continuous analog parameters within the underlying Strong PUF, for example determining its delay values. Note again that physical access to the PUF is part of the natural attack model on PUFs, as mentioned above. If a Controlled PUF has been modeled, the same effects for protocols resting on their unpredictability and physical unclonability apply that is explained above.

3) *Weak PUFs*: Weak PUFs (or POKs) are PUFs with few, fixed challenges, in the extreme case with just one challenge. It is usually assumed that their response(s) remain inside the PUF-carrying hardware, for example for the derivation of a secret key, and are not easily accessible for external parties. Weak PUFs are the PUF class that is the least susceptible to the presented modeling attacks.

Support Vector Machine (SVM) and Logistic Regression are the two main algorithms that are used to model the PUF.

4.1 Support Vector Machine and Logistic Regression

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis [6]. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall [6]. Fig.4.1a shows two different datasets, and a linear SVM model classifies them in a hyper plane as seen in that figure. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. Here we have implemented Radial base function to classify more non-linear dataset as shown in Fig.4.1b.

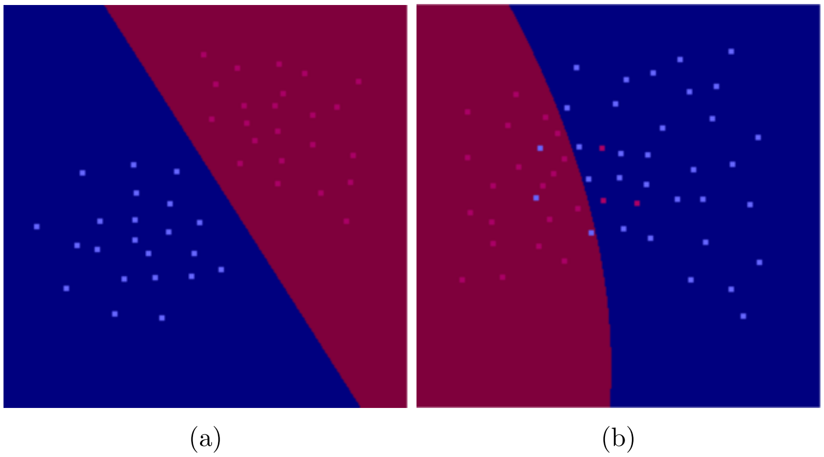


Figure 4.1: Support Vector Machine with (a) linear and (b) non-linear (RBF-kernal) classification [6]

Logistic regression (LR) is the other important method employed to attack the PUF, LR is appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis [26]. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labeled ”0” and ”1”. In the logistic model, the log-odds (the logarithm of the odds) for the value labeled ”1” is a linear combination of one or more independent variables (”predictors”); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled ”1” can vary between 0 (certainly the value ”0”) and 1 (certainly the value ”1”), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The Logistic function is shown in Fig.4.2. The binary logistic regression model has extensions to more than two levels of the dependent variable: categorical outputs with more than two values are modeled by multinomial logistic regression, and if the multiple categories are ordered, by ordinal logistic regression, for example the proportional odds ordinal logistic model. The model itself simply models probability of output in terms of input, and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other; this is a common way to make a binary classifier.

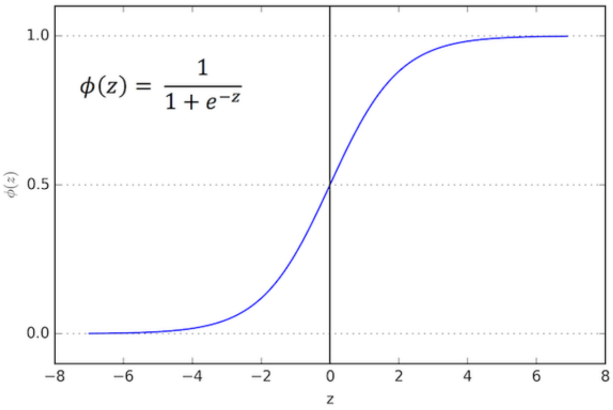


Figure 4.2: Function used for logistic regression

We used ML algorithms from [6] for SVM and [26] for LR to establish attacks on the PUF designs explained in chapter-7

4.2 Machine Learning Modelling Attack on PUF

1) *Arbiter PUFs*: Arbiter PUFs (Arb-PUFs) were first introduced in [5]. It has become standard to describe the functionality of Arbiter PUFs via an additive linear delay model . The overall delays of the signals are modeled as the sum of the delays in the stages. In this model, one can express the final delay difference δ between the upper and the lower path in a k-bit Arb-PUF as $\delta = \omega\theta$, where ω and θ are of $k + 1$ dimension. The parameter vector ω encodes the delays for the subcomponents in the Arbiter PUF stages, whereas the feature vector θ is solely a function of the applied k-bit challenge C_i. The output t of an Arbiter PUF is then determined by the sign of the final delay difference δ. The author made the technical convention of saying that $t = -1$ when the Arbiter PUF output is actually 0, and

$t = 1$ when the Arbiter PUF output is 1: [7]

$$t \quad = \quad sgn(\delta) = sgn(\omega\theta) \tag{4.1}$$

Equation 4.1 shows that the vector ω via $\omega\theta = 0$ determines a separating hyperplane in the space of all feature vectors θ. Any challenges C_i that have their feature vector located on the one side of that plane give response $t = -1$, those with feature vectors on the other side $t = 1$. Determination of this hyperplane allows prediction of the PUF. [7]

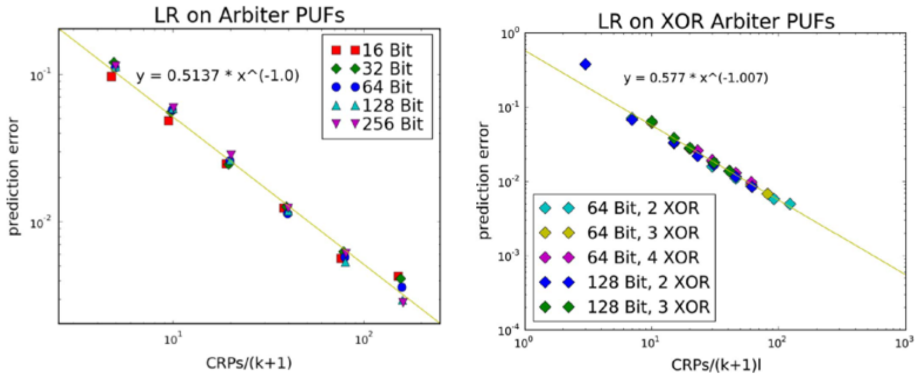
2) *XOR Arbiter PUFs*: One possibility to strengthen the resilience of arbiter architectures against machine learning, which has been suggested in [cite], is to employ l individual Arb-PUFs in parallel, each with k stages (i.e., each with bitlength k). The same challenge C_i is applied to all of them, and their individual outputs t_i are XORed in order to produce a global response t_{XOR}.We denote such an architecture as l-XOR Arbiter PUF (with the 1-XOR Arbiter PUF being identical to the standard Arbiter PUF). A formal model for XOR Arbiter PUFs can be derived as follows. Making the convention $t_i \in (-1, 1)$ as done earlier, it holds that $t_{XOR} = \prod_{i=1}^{l} t_i$. This leads with equation 4.1 to a parametric model of an l-XOR Arbiter PUF, where ω and θ denote the parameter and feature vector, respectively, for the i-th Arbiter PUF: [7]

$$t_{XOR} \quad = \quad \prod_{i=1}^{l} sgn(\omega\theta) = sgn \exp\left(\prod_{i=1}^{l} \omega\theta\right) \tag{4.2}$$

$$t_{XOR} \quad = \quad sgn(\oplus_{i=1}^{l} \omega \oplus_{i=1}^{l} \theta) = sgn \exp\left(\omega_{XOR}\theta_{XOR}\right) \tag{4.3}$$

While 4.2 gives a nonlinear decision boundary with $l(k+1)$ parameters, 4.3 defines a linear decision boundary by a separating hyperplane ω_{XOR} which is of dimension $(k+1)^l$.

Using this the authors have modelled the PUF by extracting few CRPs and then allowing ML algorithms to predict the possible CRPs which is not been exposed to the algorithms. We can note that as the prediction increases the chances of predicting the outcome of the PUF for any input challenge becomes easy. Thus the PUF is digitally cloned and is no more unpredictable.



(a) (b)

Figure 4.3: Machine Learning attack on (a) Arbiter PUF and (b) XOR-Arbiter PUF [7] NOTE: Prediction Error = 1 - Prediction Accuracy

Fig.4.3a and Fig.4.3b shows that Arbiter and XOR Arbiter PUFs output is predictable at almost 99% accuracy after the CRP is almost around 10000 bits. With high speed digital processing device at existence it will take very less time to extract this information on side channel attacks. Table 4.1 shows that all the Strong PUFs that have been discussed are prone to the Machine Learning modelling attack.

Table 4.1: Machine learning modelling attacks on strong PUF [7]

PUF Type	**No of XORs / FF-Loops**	**ML Method**	**Bit Length**	**CRP Source**	**CRPs ($\times 10^3$)**	**Prediction Rate**
Arbiter PUF	–	LR	128	ASIC	6.5	99%
XOR Arbiter PUF	5	LR	128	ASIC	78	99%
FF Arbiter PUF	8	ES	128	Simulation	50	99%

Chapter 5

Machine Learning Attack Resistive PUF

Chapter-4 leaves an understanding that most of the PUFs are predictable with the prediction accuracy close to 99%, thus bringing in the need to develop ML modelling attack immune PUFs. Two such recent works are discussed here.

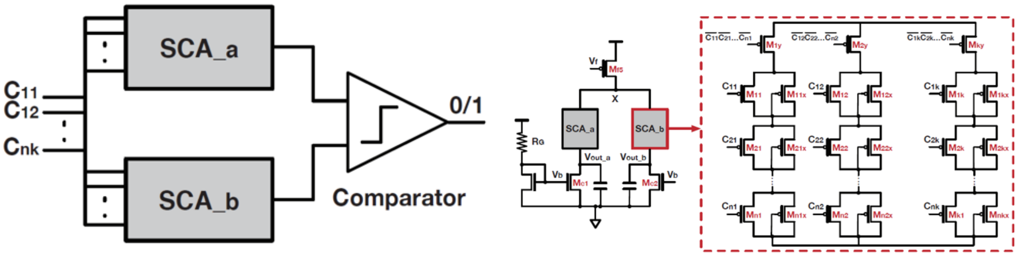
5.1 SCA PUF

The key for engineering a secure silicon PUF is identifying an output function that would be nonlinear in random variables. [8] introduces a highly unpredictable PUF that uses the strongly non-linear I-V terminal dependencies to generate PUF responses. Its central feature is that it moves away from the delay/digital implementation paradigm towards the current/analog one, thereby realizing the necessary degree of nonlinearity over a space of permutations. Because it doesnt rely on digital techniques for injecting the nonlinearity, it does not compromise the stability in the output response to environmental variations [8].

The output function should ideally have two properties: (1) be nonlinear in random parameters, and (2) introduce the coupling effect in which two or more random variables interact in producing the output. Both of these properties are enabled if the binary output is produced by comparing two voltages produced by a suitably arranged network of FETs operating in subthreshold region. The equation relating the subthreshold current to FET terminal voltages given in equation 5.1

$$I_{ds} = I_s 10^{\left(\frac{V_{gs}-V_{th}+\lambda V_{ds}+\gamma V_{bs}}{S}\right)}\left(1-10^{\frac{-nV_{ds}}{S}}\right) \tag{5.1}$$

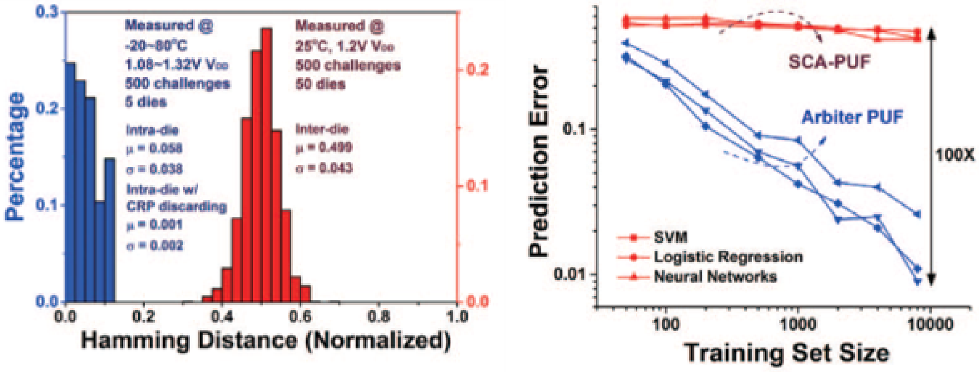
where I_{ds} is the drain-to-source subthreshold current, I_s is the nominal current, V_{gs} is the gate-to-source voltage, V_{th} is the transistor threshold voltage, V_{ds} is the drain-to-source voltage, V_{bs} is the body-to-source voltage, and λ, γ, and n are the coefficients of drain-induced barrier lowering and body bias, and the subthreshold coefficient respectively. Crucially, the current is exponentially dependent on the threshold voltage V_{th}. This is important because V_{th} exhibits large and spatially-uncorrelated variability due to random dopant fluctuation (RDF). In nanometer scale CMOS devices, RDF is very significant and grows with transistor scaling. Equation 5.1 also captures the impact of physical mechanisms of drain-induced barrier lowering and of body effect which lead to a dependence of V_{th} on V_{ds} and V_{bs}. In the second part of the equation, we use a linear expansion of V_{th} in terms of V_{ds} and V_{bs} to enable closed-form analysis.



(a) (b)

Figure 5.1: SCA PUF (a) Architecture and (b) Array schematics of the PUF [8]

Figure 5.1a depicts the overall architecture of the SCA (Sub-threshold Current Array) PUF. The PUF is implemented as a two-dimensional transistor array with all devices subject to stochastic variability operating in subthreshold region. Each PUF consists of two nominally identical arrays. The array schematic is shown in Figure 5.1b. The array is composed of k columns and n rows of a unit cell. We use the term stochastic transistor to refer to a device with high amount of threshold voltage variability. The unit cell consists of a stochastic subthreshold nFET, which is a transistor with a highly variable threshold



(a) (b)

Figure 5.2: SCA PUF (a) Inter-PUF HD and (b) Machine learning immunity [8]

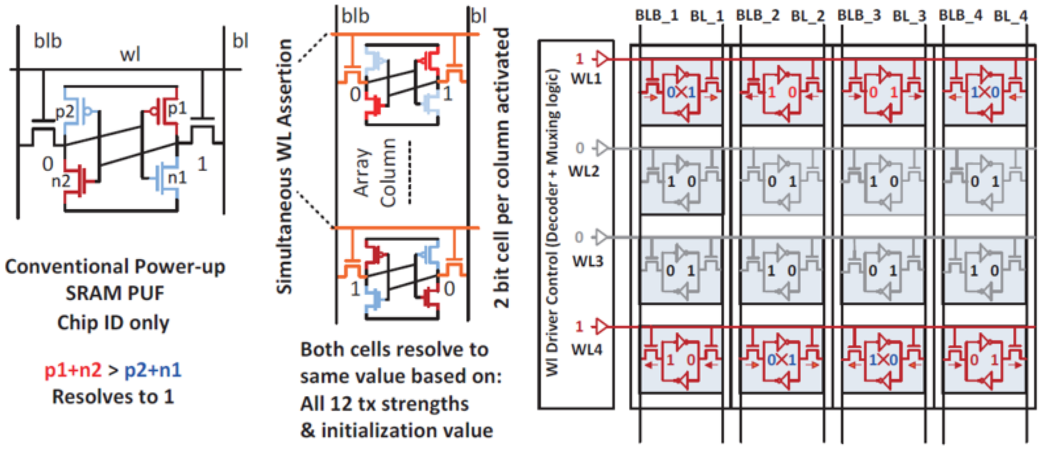
voltage that always operates in the subthreshold region. A non-stochastic switch transistor is arranged in parallel to the stochastic FET. The non-stochastic transistor M_0 acts as a load device and operates in the subthreshold region (its gate terminal is tied to ground). At the bottom of each column of cells is a footer transistor M_{iy} controlled by the challenges $C_{i1}...C_{in}$. Its role is to ensure that there is never a low-impedance path to ground from V_{out}. Both array blocks are driven with the same set of control inputs and thus in the absence of variability produce identical voltages. The randomness of transistor threshold voltages leads to the differences in two output voltages. The binary response is generated by comparing the output voltages produced by the two arrays via a comparator. The size of the CRP set is 2^{kn}, making it a strong PUF.

Inter die hamming distance and hamming weight showing the PUFs uniqueness and randomness, is taken across 50 dies with 500 challenges. For this design, the average normalized inter-die HD is 0.499 with standard deviation of 0.043 and average hamming weight is 0.528 (standard deviation = 0.109) shown in Figure.5.2a. The Intra-die HD used as a measure of bit error rate (BER) and temporal stability, is measured across 5 dies with 500 challenges, across the temperatures of -20°C to 80°C and voltages of 1.08-1.32V and its average is found to be 0.058 with standard deviation of 0.038. The worst case BER or 9% is reduced using dynamic thresholding technique to 2.6% with 42% loss in CRPs in the worst case. The number of usable CRPs are almost $3.7x10^{19}$ and overall energy consumed by this design is 11pJ/bit. The non-linearity extracted form sub-threshold operation of the SCA-

PUF makes it immune to the machine learning modelling attacks shown in Figure.5.2a. When compared to the Arbiter PUF at 10000 CRPs the machine learning prediction is around 60%.

5.2 SRAM PUF

Figure 5.3a shows a conventional SRAM PUF cell, which is usually based on the start-up value at power-up. The start-up value is determined by the relative strength of the two inverters in the cross-couple. The challenge-response space of a conventional SRAM PUF is equal to number of bits in the memory, and it can only be used as a chip ID. The proposed PUF is also an SRAM-based design, but is independent of the power-on state. The basic concept is to connect any two bit cells in the SRAM with complementary data initialization by simultaneously asserting their word-lines. The value they resolve to depends on the relative strength of all the 12 transistors of the two bit-cells, and their initial value. To illustrate, Figure 5.3b shows a small array with checkerboard initialization. Word lines $WL1$ and $WL4$, are asserted and the bit cells in the two rows with opposite states fight over the bit lines in each column, and resolve to a single value. Consequently, the challenge-response space is increased as we can choose any two rows from the array. For n rows, we have n^2 choices of pair-wise row selection.

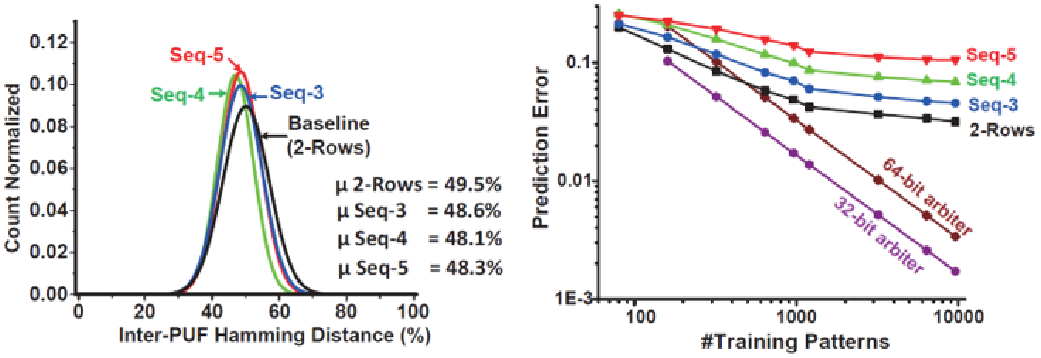


(a) (b)

Figure 5.3: SRAM PUF (a) Basic architecture and (b) Small array [9]

A sequence of initialization is made in such a way that two rows are turned on simultaneously. This results in a two differently ordered sequence that results in a final value in row 1 or in row 3. Therefore, both row selection and the order (permutations) in which the selected rows are sequentially connected determine the response of the PUF, making it more difficult to learn by ML algorithms.

The design is fabricated in 28nm technology and the average normalized inter-die HD is 0.481 to 0.495 shown in Figure.5.4a. The Intra-die HD is measured across the temperatures of 0°C to 80°C and voltages of 0.5-0.9V and its average is found to be 0.058. The worst case BER of 3.17%. The number of usable CRPs are almost $1.17x10^{11}$ and overall energy consumed by this design is only 97fJ/bit. Fig5.4b shows that the design is ML modelling attack resistant, yet gives a higher prediction error of around 89.1%



(a)

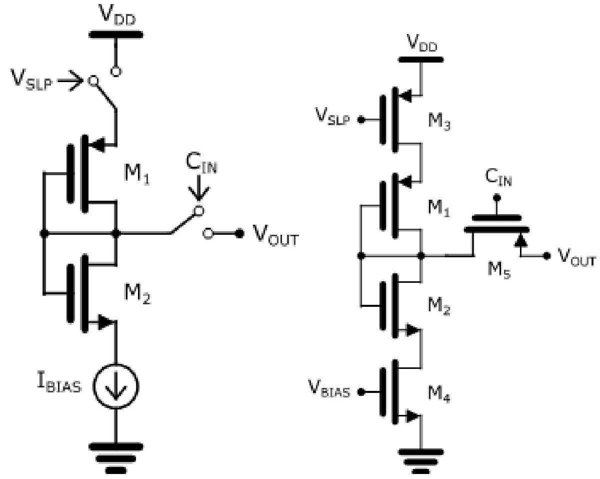
(b)

Figure 5.4: SRAM PUF (a) Inter-PUF Hamming Distance and (b) Machine learning immunity

Chapter 6

Proposed Architecture

6.1 Voltage Divider Array Strong PUF Circuit



(a) (b)

Figure 6.1: Voltage divider array strong PUF's (a) Unit cell and (b) schematics

It is established that in order to make the PUF resist the machine learning modelling-attacks, one needs to design a non-linear system. Non-linear system can tend to be less reliable but once the system is also made reliable, the overall hardware circuit becomes more secure. We know from equation 5.1 that at Subthreshold condition the current of the MOSFETs act more non-linear compared to the drain currents when they are in saturation condition. This idea can be extracted to construct a non-linear, yet a highly reliable design. The proposed PUF is constructed in $65nm$ technology. The fundamental block of this architecture is the Unit-PUF cell shown in Figure 6.1a. A unit PUF cell comprises an inverter (Transistor M_1 and M_2 with gate and drain shorted and an NMOS tail current source biased in subthreshold. Along with the inverter topology the unit PUF cell is also provided a sleep switch M_3 that disconnects the PUF from the power source and the bias current is fed in through the M_4 transistor shown in 6.1b. The challenges C_i are provided to the transistor M_5 which connects the Unit-PUF cell to the output V_{out}.

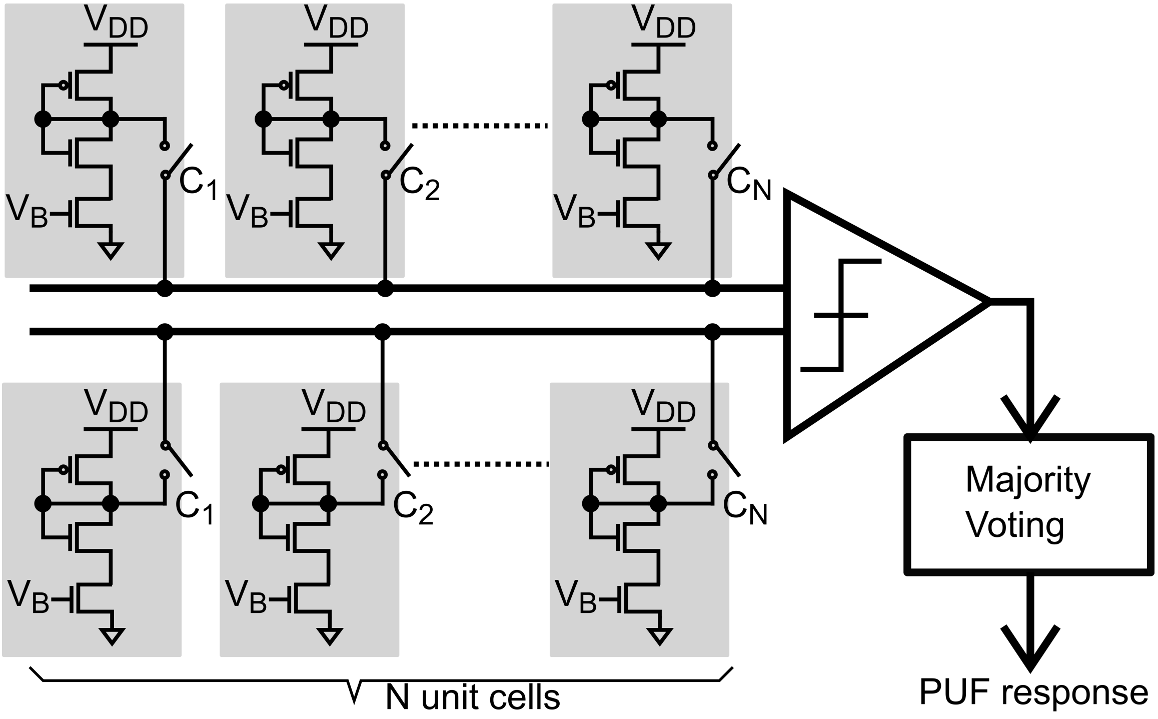


Figure 6.2: Proposed Architecture

A 1-b PUF output is obtained by comparing the drain voltage of two such unit PUF cells. To form a strong PUF, we use two arrays of N 'nominally identical' unit PUF cells shown in Figure 6.2.The challenge inputs, C_1 through C_N, determine which of the N unit PUF cells in both arrays are connected to the differential inputs of a comparator. For this design, we have 60 unit PUF cells in each array corresponding to 2^{60} possible CRPs. This makes this design a Strong PUF, providing to use almost equal to $1.1529e+18$ combinations. The PUF array used in this design has an intrinsic advantage over the current-array PUF of [8] in that the comparator input common-mode voltage does not vary significantly with the challenge pattern. Variation of comparator input common-mode voltage results in challenge pattern dependent offset and is problematic for comparator offset cancellation.

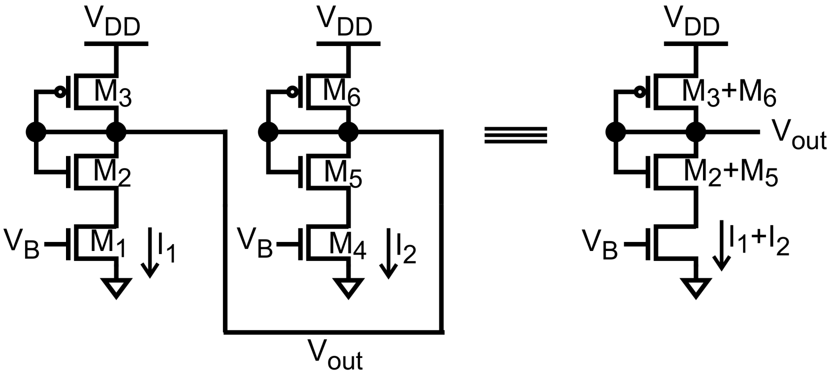


Figure 6.3: Non-linearity in PUF array with 2 unit cells

The output of the PUF is extracted from the comparator which undergoes majority voting that improves the stability of the system. Fig. 6.6c shows how nonlinearity is introduced in output voltage of the proposed PUF using 2 unit cells. The currents I_1 and I_2 through the two unit PUF cells using equation 5.1 can be written as

$$I_1 = I_s \exp\left(\frac{V_B - V_{th1}}{\eta V_T}\right);\ I_2 = I_s \exp\left(\frac{V_B - V_{th4}}{\eta V_T}\right) \tag{6.1}$$

where V_{th1} and V_{th4} denotes threshold voltages of M_1 and M_4 respectively, V_T is thermal voltage kT/q, V_B is biasing voltage for NMOS tail current source and it is assumed that V_{ds} of M_1 and M_4 is greater than 100mV. When the two unit PUF cells are connected together, the output voltage V_{out} can be expressed as

$$I_1 + I_2 \quad = \quad I_{s1} \exp\left(\frac{V_{DD} - V_{out} - |V_{thp}|}{\eta V_T}\right) \tag{6.2}$$

where $|V_{thp}|$ is the threshold voltage of PMOS transistor.

$$V_{out} = V_{DD} - |V_{thp}| - \ln\left(\frac{I_s}{I_{s1}}\right) - \ln\left[\exp\left(\frac{V_B - V_{th1}}{\eta V_T}\right) + \exp\left(\frac{V_B - V_{th4}}{\eta V_T}\right)\right] \tag{6.3}$$

It can be seen that 6.3 is a transcendental equation and V_{out} is nonlinear in terms of threshold voltages of the NMOS tail current sources. Second-order effects such as drain induced barrier lowering further increase the coupling between threshold voltage and V_{out}. Since threshold voltage exhibits large intrinsic variation due to random dopant fluctuation, voltage output of the proposed PUF is expected to show large nonlinear variation which cannot be modeled easily through ML attacks.

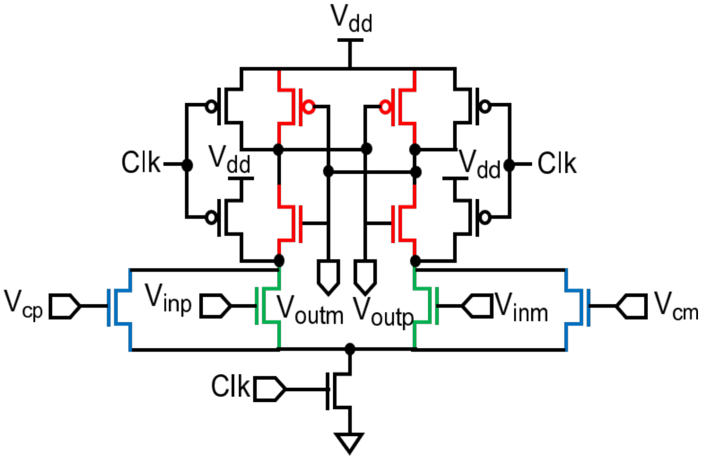
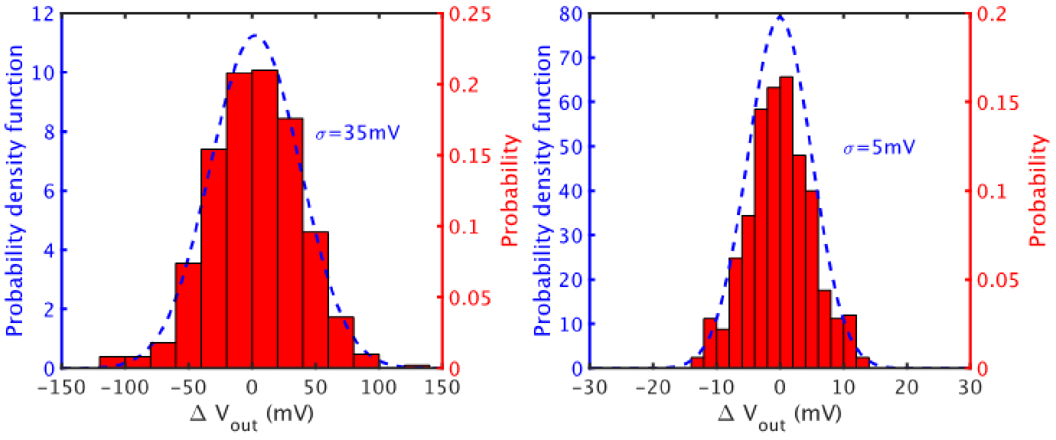


Figure 6.4: Strong arm latch comparator

Fig. 6.4 shows schematic of the comparator used in this design. A strong-arm latch is used as comparator.The comparator has two auxiliary input transistors for offset calibration shown in blue. The auxiliary transistors are controlled by the voltages V_{cp} and V_{cm} which are used to tune the comparator offset [8]. During offset calibration phase, the comparator inputs are shorted, the comparator is fired multiple times and the distribution of ‘1’ in the comparator output is observed. If the comparator has an offset, its output will have unequal distribution of ‘0’ and ‘1’. The voltages V_{cp} and V_{cm} are used to bias the auxiliary input transistors to ensure the comparator has approximately equal distribution of ‘0’ and ‘1’. The transistors in green are the amplifiers and the ones that are in red are the latches. For this design, comparator offset calibration is done in the foreground at nominal conditions of 0.9V power supply and at room temperature.

6.2 Choice of Operating Voltage

In order to ensure the reliable performance of the PUF, the noise of the PUF should be at the minimal. The comparator is the dominant noise source and thus following analyisis were made to reduce it's effect. Fig. 6.5 shows the distribution of PUF differential output voltage ΔV_{out} for two extreme cases: (a) when only 1 challenge input is '1' and (b) when all challenge inputs are '1'. The distributions are extracted from 500 monte-carlo runs. When only 1 challenge input is '1', ΔV_{out} has a large spread and a standard deviation, σ_{mis}, of 35mV. When all challenge inputs are '1', σ_{mis} reduces to 5mV.

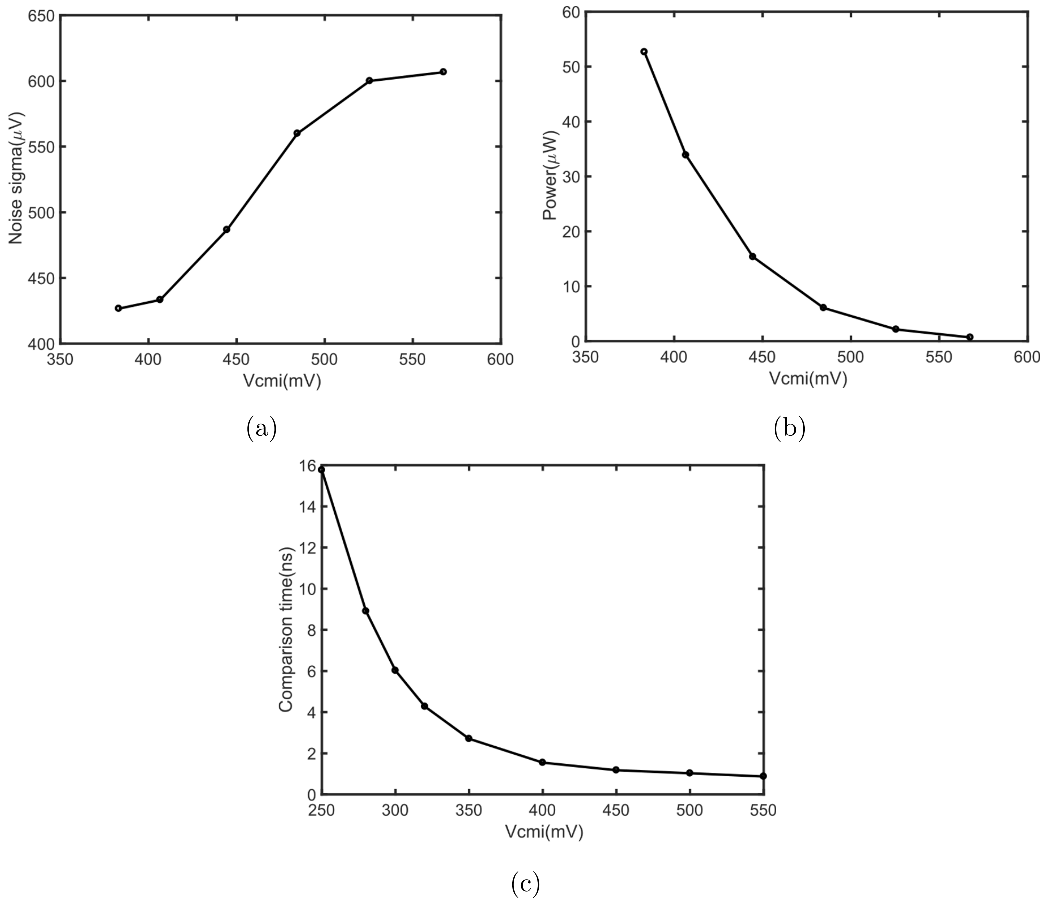


(a) (b)

Figure 6.5: Distribution of PUF differential output voltage for (a) when only 1 challenge input is '1' (b) when all challenge inputs are '1'

Now, the comparator is to be besigned such that the comparator noise is much smaller than the worst-case σ_{mis} of 5mV for the PUF to have high native stability. There is a trade-off between power and comparator noise which can be optimized by tuning the biasing voltage, V_B. As V_B is reduced, the current through unit PUF cells reduces, which reduces total power, but the comparator input common-mode voltage increases, which increases comparator noise. Also, the spped of convertion is reduced. Figure 6.6 gives the characteristics of chosing V_{cmi} that will be impacted by the choice of the V_B.

Assuming the comparator has a noise standard deviation of σ_n, the native stability



(a)

(b)

(c)

Figure 6.6: (a) V_{cmi} vs Noise sigma (b) V_{cmi} vs Power consumption (c) V_{cmi} vs Speed

of the PUF can be written as

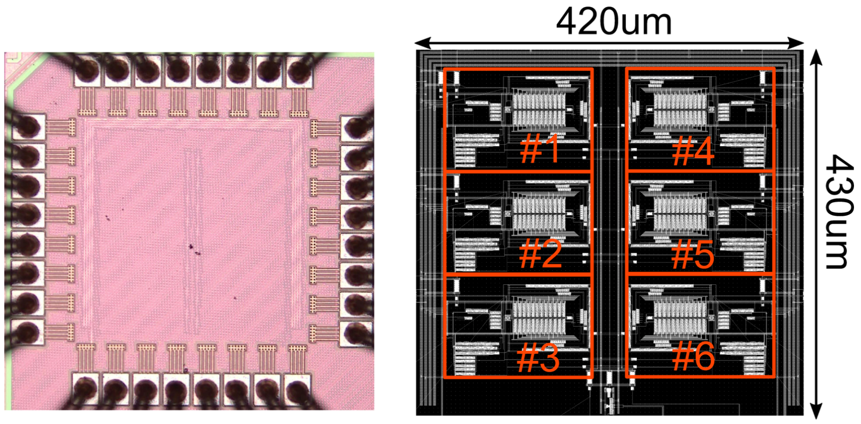
$$\text{stability} = 1 - \text{erf}\left(\frac{\sigma_n}{\sqrt{2}\sigma_{mis}}\right) \tag{6.4}$$

For this design, V_B is set to 200mV such that the comparator noise has a standard deviation of 350μV which corresponds to a native stability of 94.4%. In order to improve PUF native stability, we perform temporal majority voting of the comparator output. Application of majority voting of 7 reduces σ_n to 132μV which improves PUF native stability to 97.9%. We use a 3-bit counter which counts up every-time the comparator output is '1'. The MSB of the counter is used as 1-b PUF output. The counter is reset every 8th cycle of the comparator clock.

Chapter 7

Measurement Result and Analysis

7.1 Laboratory Setup



(a)

(b)

Figure 7.1: (a) Die photo and (b) Layout of proposed PUF

A test chip is fabricated in 65nm CMOS process. The die micro-photograph and layout are shown in Fig. 7.1. Each test chip contains 6 PUFs. Each PUF has an area of 110μm×170μm with the core (PUF array+comparator) occupying 40μm×70μm. Each blocks shown in Figure 7.1b contains the proposed PUF architecture, LFSR to provide test input challenges and a clock generator circuit to vary the throughput of the PUF.

Figure 7.2 shows the PCB set-up used to measure the data from the PUF. The inputs are Voltage source V_DD and Clock source F_clk. The output is extracted form the

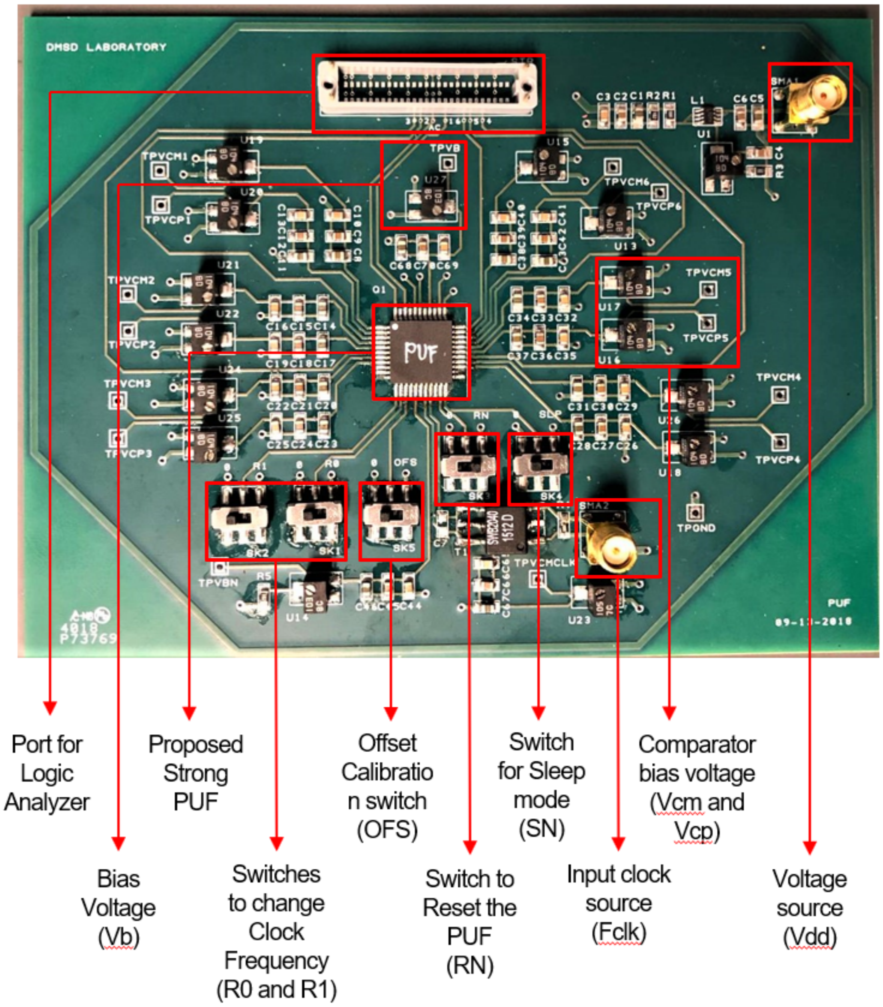

DMSD LABORATORY
PUF
Port for Logic Analyzer
Proposed Strong PUF
Offset Calibration switch (OFS)
Switch for Sleep mode (SN)
Comparator bias voltage (Vcm and Vcp)
Bias Voltage (Vb)
Switches to change Clock Frequency (R0 and R1)
Switch to Reset the PUF (RN)
Input clock source (Fclk)
Voltage source (Vdd)

Figure 7.2: PCB board for the proposed strong PUF

logic analyzer that gives the data of 6 PUF's output and the clock developed from the clock generator. Using the trimmers the voltage nodes of bias voltage V_b, common mode voltage of comparators (V_cm and V_cp) are defined. The switches R_0 and R_1 determine the frequency of operation. Switch S_N activates the sleep mode and turns off the PUF. Switch OFS is turned to make offset calibration of the comparator. Table.7.1 shows the operating

Table 7.1: Selected operating parameters

Operating Parameters	**VDD**	**Fclk**	**R0**	**R1**	**Vb**	**Vcm**	**Vcp**	**OFS**	**SN**	**RN**
Values	$900mV$	$100MHz$	$900mV$	$0V$	$100mV$	$401mV$	$398mV$	$0mV$	$0mV$	$900mV$

values used for one measurement.

7.2 Hamming Distance

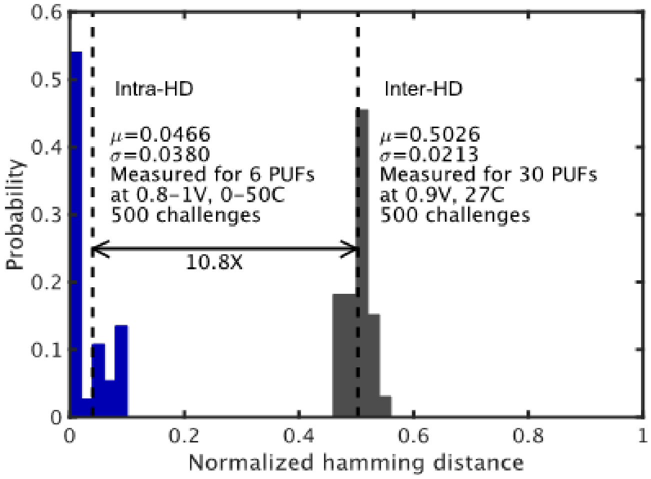


Figure 7.3: Measured normalized intra and inter-HD

Fig. 7.3 shows the measured normalized intra and inter-HD of the PUF. The ideal PUF should have the Intra-HD value to 0.5 and the Inter-HD value should be 0. Intra-HD is measured for 6 PUFs over a supply range of 0.8V-1V and temperature range of 0-50°C for 500 challenges. The measured intra-HD is 0.0466 with a standard deviation of 0.038. Inter-HD is measured for 30 PUFs at 0.9V supply and 27°C for 500 challenges. The measured inter-HD is 0.5026 with a standard deviation of 0.0213. The ratio between inter-HD to intra-HD is 10.8 which indicates that the proposed PUF has a high uniqueness.

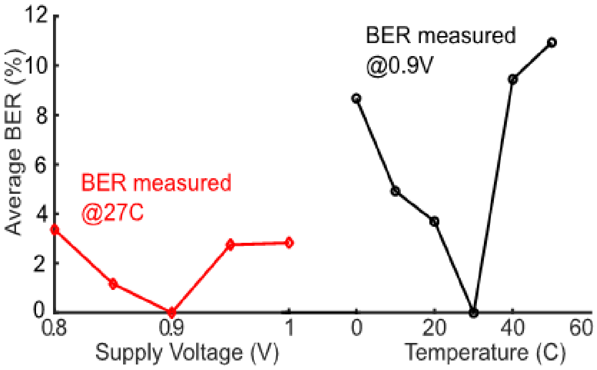


Figure 7.4: Average BER across supply voltage and temperature

Fig. 7.4 shows the measured average BER across supply voltage and temperature. The BER with variation in supply voltage and temperature is measured with respect to PUF output at 0.9V supply and 27°C temperature. Temperature variation affects BER more than supply voltage variation. The worst case BER is 10.9%.

7.3 Non Linearity and Randomness Analysis

In order to test the randomness of our PUF, we used NIST randomness tests on 30 PUF devices from 5 different test chips. For each PUF, we recorded the response for 16 different times. Thus, the NIST tests are performed on 480 PUF response streams. Fig. 7.5 shows graphically the results of the NIST tests. For 12/15 NIST tests, the minimum pass rate was greater than 0.95. The pass rate for 3 tests, DFT, overlapping template and approximate entropy, was between 0.85-0.9. The NIST test results indicate good randomness of the proposed PUF.

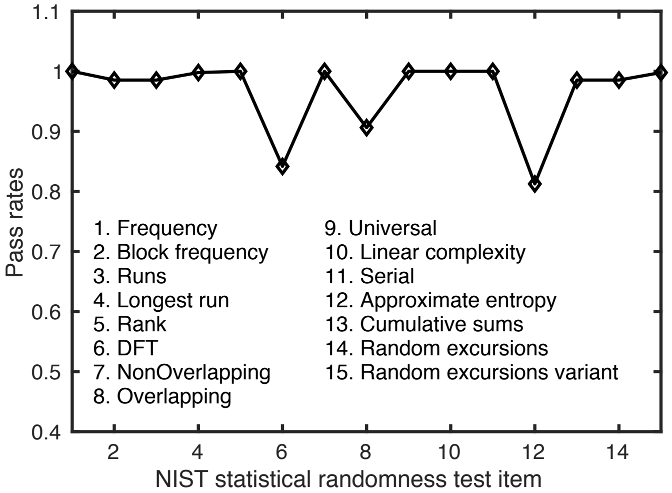


Figure 7.5: NIST randomness test results

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components.

Here the input of the PUF is expressed into a 2-D figure shown in Figure.7.6 where

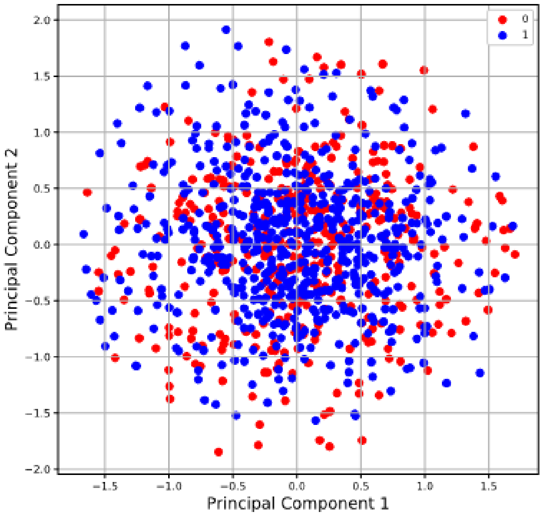


Figure 7.6: Principal component analysis

the binary output is plotted over the 2-D figure. The figure shows us that the proposed PUF's output is randomly arranged making it a non-linear model.

7.4 Machine Learning Immunity

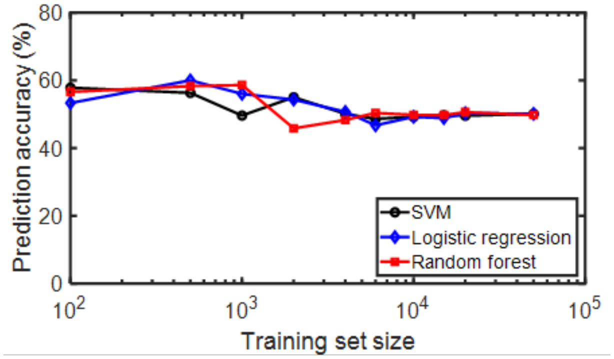


Figure 7.7: Machine learning modelling attack immunity

In order to test the susceptibility of the proposed PUF to machine learning attacks, we used three different machine learning algorithms: support vector machine with nonlinear radial bias function (RBF) kernel, logistic regression, and Random forest. Fig. 7.7 shows the results of machine learning attacks on our PUF. For small training set sizes, the prediction accuracy rate for the proposed PUF and SCA is close to 50% (Figure.5.2b), while the prediction accuracy rate for arbiter PUF is close to 70%. As the training set size is increased, the prediction accuracy rate for our PUF does not change significantly and remains close to 50%. The SCA PUF also exhibits similar performance and as training set size increases, the prediction accuracy rate for SCA PUF is 60%. On the other hand, the arbiter PUF has a prediction accuracy rate of 70% at small training set sizes which reduces quickly to 99.9% as the training set size is increased to 8000. Thus, use of subthreshold nonlinearity makes the proposed PUF and SCA PUF robust against the three different machine learning attacks we tried, while the arbiter PUF can be easily modeled with training set sizes > 2000.

7.5 Figure of Merit

Each chip contains 6 PUF and each PUF consumes 3.8μW power from 0.9V supply with a throughput of 12.5M samples/s. Out of the 3.8μW power, the comparator consumes 1.2μW power while the PUF array consumes 2.6μW power. Table 7.2 compares our work with state-of-the-art strong PUFs. Compared to existing work, our PUF achieves simultaneous high energy efficiency and strong resistance to ML attacks. While the proposed PUF has similar resistance to ML attacks as [8], energy efficiency of the proposed PUF is 36× better than [8]. The SRAM PUF of [9] has 3× better energy efficiency than the proposed PUF, but is 5× more susceptible to ML attacks than the proposed PUF. We propose a figure-of-merit (FoM) in order to quantitatively compare different PUFs by taking into account both energy efficiency and resistance to ML attacks. We define the FoM as the ratio of energy/bit and prediction error. Low energy consumption and high prediction error reduces the FoM. It can be seen from Table 7.2 that the proposed PUF has the best FoM which is a factor of 1.5× better than the current state-of-the-art.

Table 7.2: Comparison with state-of-the-art strong PUFs

	This **Work**	[8] VLSI'17	[10] ACM'07	[27] ISSCC'17	[5] VLSI'04	[9] VLSI'17
Technology (nm)	**65**	130	90	40	180	28
Type of PUF	**Voltage array**	Current array	Ring oscillator		Arbiter	SRAM
Possible CRPs	**1.15 x $\mathbf{10^{18}}$**	$\approx 3.7 \times 10^{19}$	523776	$\approx 5.5 \times 10^{28}$	$\approx 1.4 \times 10^{20}$	1.17×10^{11}
ML prediction error (10^4 CRPs)	**49%**	40%	1%	–	1%	10.6%
Worst-case BER	**10.9%**	9% (0.4%*)	0.48%	9%	4.8%	3.17%
Energy/bit (pJ/bit)	**0.3**	11	–	17.75	–	0.097
Voltage range (V)	**0.8-1**	1.08-1.32	1.08-1.2	0.7-1.2	1.75-1.85	0.5-0.9
Temperature range (°C)	**0-to-50**	−20-to-80	20-to-120	−25-to-125	20-to-70	0-to-80
Inter-HD	**0.5026**	0.499	0.4615	0.5007	0.4	0.481-0.495
Intra-HD	**0.0466**	0.058	0.0048	0.0101	0.0357	0.0317
FoM	**0.61**	27.5	–	–	–	0.91

*FoM = 100×Energy/bit/prediction error(%)

Chapter 8

Other Architectures

8.1 Cascaded Strong PUF

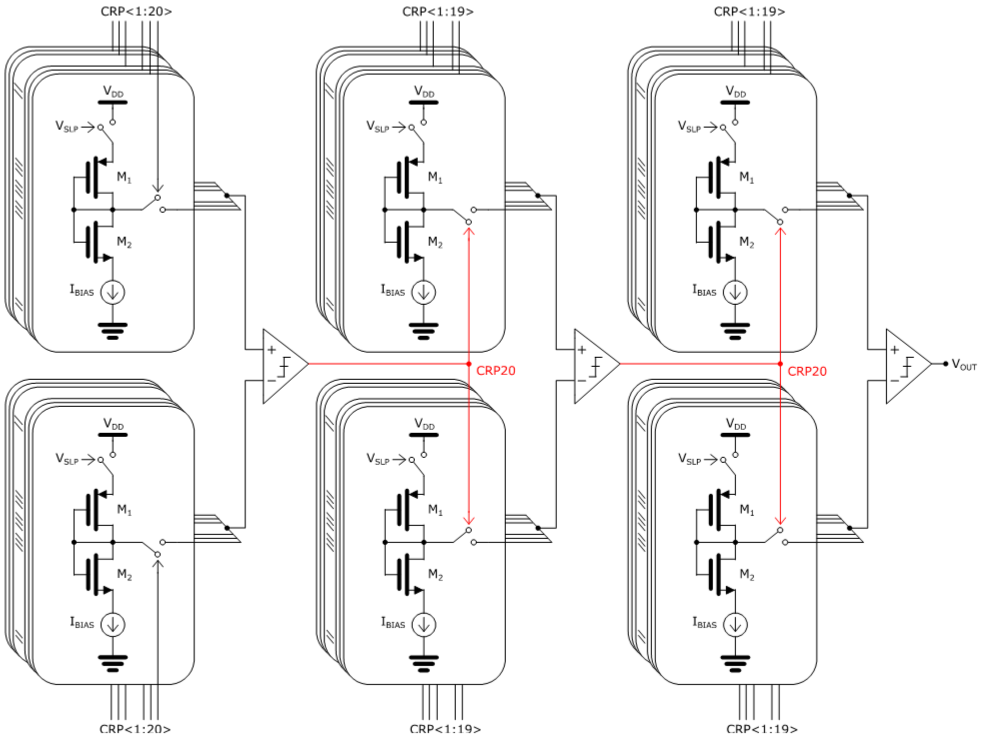
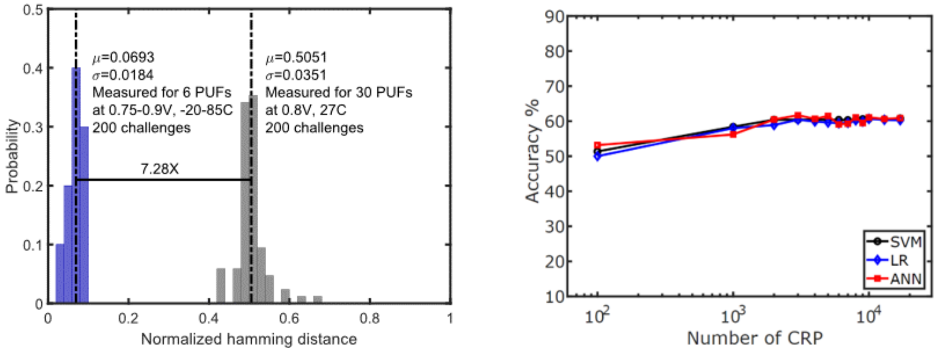


Figure 8.1: Cascaded strong PUF schematics

[28] proposes a architecture to cascade the individual Strong PUFs such that the response of one PUF is provided at the channel to the next PUF. If each PUF is constructed

as a function, then each function will be partially depended on the previous function thus making the design highly non-linear. This non-linear function leads to reduction in prediciton rate by machine learning algorithams.

We constructed a similar cascaded PUF shown in Fig.8.1. The proposed strong PUF is used as the single block and they are cascaded. Each stage of the cascade is formed by connecting an array of 20 unit PUFs differentially connected to the two inputs of a comparator. Three such stages are cascaded to form the overall strong PUF with 2^{58} challenge inputs, with the first stage accepting 2^{20} external challenges and the other two stages accepting 2^{19} external challenges. The comparator output of the first stage provides the 20^{th} challenge input to the second stage and comparator output of the second stage provides the 20^{th} challenge input to the third stage.



(a) (b)

Figure 8.2: Cascaded Strong PUF (a) Normalized intra and inter-HD and (b) Machine learning modelling attack immunity

Monte carlo simulations were run on Cadence to analyis the performance of the proposed cascaded PUF design. It was run for the temperature variation of -20 to 85°C and the voltage variation of 0.75 to 0.9V and 200 CRPs were recorded for Intra-HD, while for Inter-HD 30 PUFs were measured at 0.8V and 27°C temperature recording 200 CRPs. Figure.8.2a shows that Hamming distance have deteriorated a little. Inter HD is 0.5051 and the Intra-HD is 0.0693, the results in a BER of 14.75%. But the Machine learning prediction rate is around to 60% making the design secure. But the BER is quite high that might affect the reliability of the PUF

8.2 Controlled Cascaded Strong PUF

Chapter-2.4 shows that incorporating a authentication block around the PUF provides more security. It becomes difficult to model these PUFs because of limited accessibility as the authentication block provides added security by encryption and restricting the attackers from knowing the actual CRP to the PUF. Figure.8.3 shows an example schematic proposed, that makes use of cascaded PUF with and the authentication block to make the overall PUF highly secure. Selected challenges C_i under goes simple XOR operation with the user defined bits UID_i that acts as the validation code. And the CRPs are masked by MX_i which can also be altered. The masked challenge now reaches the actual cascaded PUF.

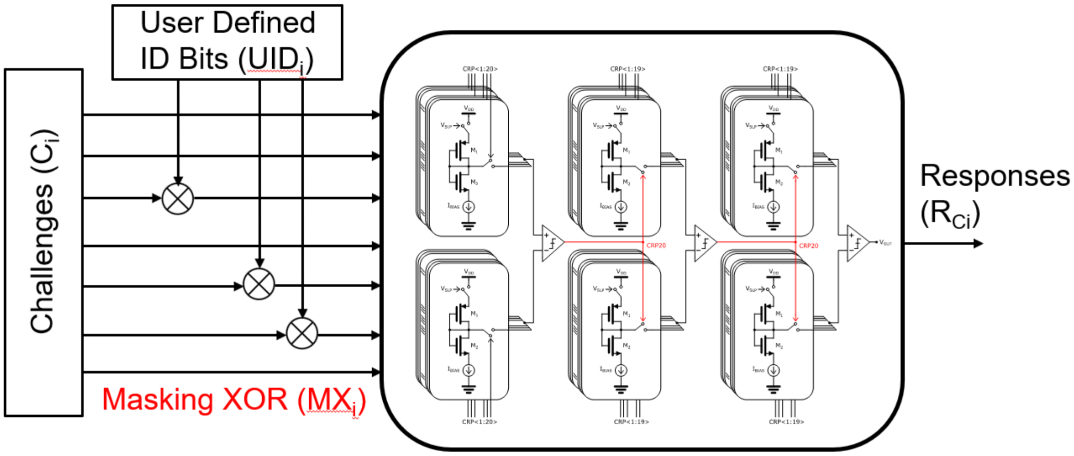
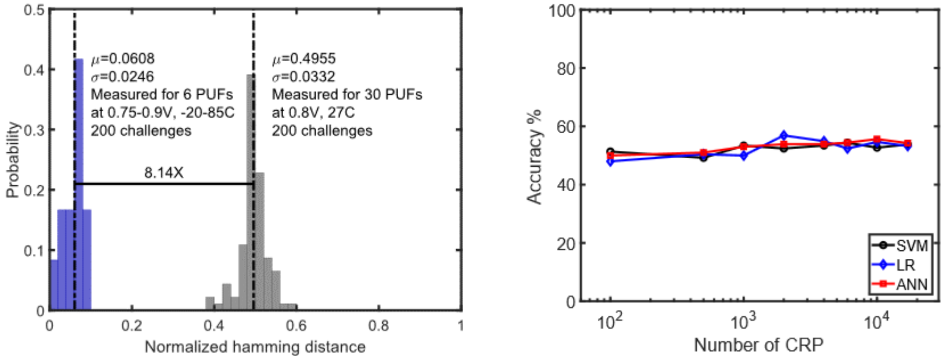


Figure 8.3: Controlled Cascaded Strong PUF schematics

The attacker is assumed to have knowledge about the actual challenge and its response. Even when training the CRP along with the User defined bits the prediction accuracy rate was around 53% (Fig.8.4b) that is favourably less compared to cascaded PUF.

Figure.8.4a shows that the Hamming distance did very much with the cascaded PUF design. This architecture was run under the same conditions and the Inter-HD is 0.4955 while the Intra-HD is 0.0332.

Table.8.1 gives the comparison among the proposed architectures. Though the FoM of Controlled Cascaded Strong PUF seems good, the BER is at 16.13 reducing the



(a) (b)

Figure 8.4: Controlled Cascaded Strong PUF (a) Normalized intra and inter-HD and (b) Machine learning modelling attack immunity

reliability. While the Voltage array PUF has the high FoM and got better BER.

Table 8.1: Comparison with proposed strong PUF architectures

	Voltage Array PUF	**Cascaded Strong PUF**	**Controlled Strong PUF**
Technology (nm)	65	65	65
Possible CRPs	1.15×10^{18}	1.15×10^{18}	1.15×10^{18}
ML prediction error (10^4 CRPs)	49%	40%	47%
Worst-case BER	10.9%	14.75%	16.13%
Energy/bit (pJ/bit)	0.3	0.43	0.432
Voltage range (V)	0.8-1	0.75-0.9	0.75-0.9
Temperature range (°C)	0-to-50	−20-to-85	−20-to-85
Inter-HD	0.5026	0.5051	0.4955
Intra-HD	0.0466	0.0693	0.0332
FoM	0.61	1.07	0.92

Chapter 9

Conclusion

The dependence on electronic devices has proliferated in almost everyday activities making them easy targets and threatening the security and privacy of an individual or a group. The traditional practice of using a secret binary key stored in non-volatile memory (NVM) for authentication is less secure against hardware or software based attacks. In contrast to NVMs, a physical unclonable function (PUF) does not store a physical key but rather derives its unique signature from random variations. This provides a promising advantage over traditional method of providing security to low power devices. A subthreshold voltage-divider array based strong PUF is proposed in this work. Voltage output of the proposed PUF has a strong nonlinear dependence on threshold voltage which results in robustness against ML based modeling attacks. A 65nm prototype consumes only 0.3pJ/bit and has prediction accuracy of 51% with three different ML algorithms. The ratio between Inter to Intra HD being 10.8 shows good reliability of PUF against voltage and temperature fluctuations. Design variants such as cascaded PUF and controlled cascaded PUFs were also proposed to increase the non-linearity at the expense of reliability. Though the alternate design architectures provided good ML-modelling attack resistance (60% and 53%), they provided worse BER (14.75% and 16.13%). Future work can be explored more on Controlled - Cascaded PUF architecture to make it more reliable and thus developing a highly secure authentication hardware device operating in very low power

Bibliography

[1] Cisco - White Paper (2017-2022). In *Cisco Visual Networking Index: Forecast and Trends*, 2019.

[2] Ulrich Rührmair, Srinivas Devadas, and Farinaz Koushanfar. Security based on physical unclonability and disorder. In *Introduction to Hardware Security and Trust ISBN: 9781441980793*, 2012.

[3] K. Lofstrom, W. R. Daasch, and D. 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, Feb 2000.

[4] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Controlled physical random functions. In *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, pages 149–160, Dec 2002.

[5] 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 *IEEE Symposium on VLSI Circuits*, pages 176–179, 2004.

[6] Chang, Chih-Chung, Lin, and Chih-Jen. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[7] Ulrich Rührmair, Jan Sölter, Frank Sehnke, Xiaolin Xu, Ahmed Mahmoud, Vera Stoyanova, Gideon Dror, Jrgen Schmidhuber, Wayne Burleson, and Srinivas Devadas. PUF

modeling attacks on simulated and silicon data. In *IEEE transactions on information forensics and security*, volume 8, pages 1876–1891, 2013.

[8] Xiaodan Xi, Haoyu Zhuang, Nan Sun, and Michael Orshansky. Strong subthreshold current array PUF with 2^{65} challenge-response pairs resilient to machine learning attacks in 130nm CMOS. In *IEEE Symposium on VLSI Circuits*, pages C268–C269, 2017.

[9] S. Jeloka, K. Yang, M. Orshansky, D. Sylvester, and D. Blaauw. A sequence dependent challenge-response PUF using 28nm SRAM 6T bit cell. In *IEEE Symposium on VLSI Circuits*, pages C270–C271, 2017.

[10] G Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual design automation conference*, pages 9–14. ACM, 2007.

[11] Blaise Gassend, Daihyun Lim, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Identification and authentication of integrated circuits. *Concurrency and Computation: Practice and Experience*, 16(11):1077–1098, 2004.

[12] Siarhei S. Zalivaka, Alexander A. Ivaniuk, and Chip-Hong Chang. Low-cost fortification of arbiter PUF against modeling attack. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017.

[13] Yuki Tanaka, Song Bian, Masayuki Hiromoto, and Takashi Sato. Coin flipping PUF: a novel PUF with improved resistance against machine learning attacks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):602–606, 2018.

[14] 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, 2002.

[15] 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.

[16] Gassend B. Physical random functions. *Masters thesis, Massachusetts Institute of Technology*, 2003.

[17] Boris Škorić. Quantum readout of physical unclonable functions. In Daniel J. Bernstein and Tanja Lange, editors, *Progress in Cryptology – AFRICACRYPT 2010*, pages 369–386, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[18] U. Rhrmair, C. Jaeger, M. Bator, M. Stutzmann, P. Lugli, and G. Csaba. Applications of high-capacity crossbar memories in cryptography. *IEEE Transactions on Nanotechnology*, 10(3):489–498, May 2011.

[19] Taylor J Gershenfeld N Pappu R, Recht B. Physical one-way functions. *Journal of Machine Learning Research*, 297:20262030, 2002.

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

[21] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal. Design and implementation of puf-based "unclonable" rfid ics for anti-counterfeiting and security applications. In *2008 IEEE International Conference on RFID*, pages 58–64, April 2008.

[22] Raghavan Kumar and Wayne Burleson. Hybrid modeling attacks on current-based PUFs. In *IEEE 32nd International Conference on Computer Design (ICCD)*, pages 493 – 496, 2014.

[23] Georg T. Becker. On the Pitfalls of Using Arbiter-PUFs as Building Blocks. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 34, pages 1295 – 1307, 2015.

[24] Ahmad O. Aseeri, Yu Zhuang, and Mohammed Saeed Alkatheiri. A Machine Learning-Based Security Vulnerability Study on XOR PUFs for Resource-Constraint Internet of

Things. In *IEEE International Congress on Internet of Things (ICIOT)*, pages 49 – 56, 2018.

[25] Mohammed Saeed Alkatheiri and Yu Zhuang. Towards fast and accurate machine learning attacks of feed-forward arbiter PUFs. In *IEEE Conference on Dependable and Secure Computing*, pages 181 – 187, 2017.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[27] Kaiyuan Yang, Qing Dong, David Blaauw, and Dennis Sylvester. A physically unclonable function with BER< 10e- 8 for robust chip authentication using oscillator collapse in 40nm CMOS. In *IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 1–3, 2015.

[28] Arunkumar Vijayakumar, Vinay C Patil, Charles B Prado, and Sandip Kundu. Machine learning resistant strong PUF: Possible or a pipe dream? In *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 19–24, 2016.