MACHINE LEARNING BASED ATTACK ANALYSIS ON PUFs

A thesis submitted in partial fulfilment of the requirements

for the degree of Master of Technology

in VLSI Design

by

Lokesh Kumar

(2017PEV5124)

Under the supervision of

Prof. Vineet Sahula

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR

June 2019

Certificate

Department of Electronics & Communication Engineering

Malaviya National Institute of Technology, Jaipur

This is to certify that the Dissertation Report on " **MACHINE LEARNING BASED ATTACK ANALYSIS ON PUFs**" by **Lokesh Kumar** is bonafide work completed under my supervision, hence approved for submission in partial fulfilment for the Master of Technology in VLSI Design, Malaviya National Institute of Technology, Jaipur during academic session 2018-2019 for the full time post graduation program of session 2017-2019. The work has been approved after plagiarism check as per institute rule.

Prof. Vineet Sahula

Professor

Department of Electronics & Communication Engineering

Malaviya National Institute of Technology, Jaipur

June 2019

Abstract

In this thesis, we demonstrate how numerically modeling attacks can break several Physical Unclonable Functions (PUFs) suggested. Because of fixed set of a PUF's challenge-response pairs (CRPs), our attacks build a computer algorithm that acts indistinguishably on nearly all CRPs from the initial PUF. Subsequently, this algorithm may impersonate the PUF and may be cloned and distributed arbitrarily.

Attacks based on Machine Learning (ML) are the most appropriate and helpful type of assault for so-called Strong Physical Unclonable Functions (Strong PUFs). In this job, we provide a summary of this technique: we address the main circumstances under which it is suitable; the ML algorithms used in this context; the recent and most advanced results on simulated and; the correct interpretation of real results; and possible future directions for studies. The subject of machine learning today is helpful, and exciting, "How secure are PUFs?"At the 2014 date.

Declaration

I declare that,

1. The work contained in this dissertation is original and has been done by me under the guidance of my supervisor.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have followed the guidelines provided by the Institute in preparing the dissertation.
4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Lokesh Kumar
(2017PEV5124)

Acknowledgements

I would like to take this opportunity to express my deep sense of gratitude and respect towards my Supervisor (Guide), **Dr. Vineet Sahula**, Professor, Department of Electronics & Communication Engineering, Malaviya National Institute of Technology, Jaipur.

I am very much indebted to him for the generosity, expertise and guidance, I have received from him while working on this project and throughout my studies. Without his support,encouragement and timely guidance, the completion of my project would have seemed a far-fetched dream.He always helped me to feel motivated throughout the research work. In this respect, I find myself lucky to have him as my Project Guide. He has guided me not only with the subject matter, but also taught me the proper style and techniques of working.

I would like to thank **Dr. D.Boolchandani**, HOD, Department of Electronics & Communication Engineering for his co-operation and help rendered in numerous ways for the successful completion of this work.

I take this opportunity to express my regards and obligation to my Family whose support and encouragement, I can never forget in my life. Also express my gratitude to all other faculty members in the department.

I am thankful to all those who have supported me directly or indirectly during the dissertation work.Above all, I thank Almighty who bestowed his blessings upon us.

Lokesh Kumar
(2017PEV5124)

Contents

List of Figures

List of Tables

Abbreviations

PUF	**P**hysical **U**nclonable **F**unction
CRPs	**C**hallenge **R**esponse **P**airs
ML	**M**achine **L**earning
SVM	**S**upport **V**ector **M**achine
LR	**L**ogistic **R**egression
ES	**E**volution **S**trategies
LW	**L**ight **W**eight
RO	**R**ing **O**scillator
Arb	**A**rbiter
ε	**E**rror **R**ate

Symbols

$\prod$	Multiplication
$\bigotimes$	Xored
Δ	Delay

Dedicated to My Family, Teacher and Friends and Spacial Thanks to Mr. Vineet Sahula

Chapter 1

Introduction

1.1 Introduction of Physical Unclonable Function

Storing digital data in a device in a way that is protect to physical attack is difficult and high cost. A PUF is random order system that can be generate random response R from different challenge pairs. This randomness can not be cloned because it is manufacturing variation and environment variation this is different for different PUF . Physical Unclonable Function is very complex and disorder due to this it is avoided some of the shortcoming correlated with digital keys.

1.1.1 Types of PUFs

There are the many type of PUFs. Every PUFs has own properties , application and security aspects. Three are many types but we are discussing major PUFs like strong , weak and controlled.

1.1.1.1 Strong PUFs

Strong PUFs are a random physical systems along with complexed challenge-response pairs and other possible potential inputs. Their armament features consist of :(1) It is tedious to clone a Strong PUF, that is, to create the other skeleton which carries on comparative from the bona fide PUF in its testing sets. This condition will be held notwithstanding for the

bona fide creator of the PUF. (2) A total forecast/supposition of all testing and training sets (CRPs) inside a determined period (a some days or v weeks) Must be unbelievable, despite of whether One can publicly challenge the PUF and have limitless access to its responses. This property usually is met by the enormous number of potential problems and the restricted read-out speed of a Strong PUFs (3)It should be difficult to physically predict the responses of strong PUFs , despite of whether many various CRPs are known.

1.1.1.2 Controlled PUFs

This kind of PUFs is likely same to Strong PUF as its configuration, but additional control line that controlled the PUF. The control line restrict challenges to apply freely to the PUFs, and block straightforward read-out of there responses to the third party.This rationale can be utilized to counter the demonstrating assaults. In any case, on the off chance that the yields of the controlled PUF can be legitimately inspected, at that point, it might be conceivable to assault in Strong PUF which break its security.

1.1.1.3 Weak PUFs

Weak PUFs, for long last, may not have many difficulties — For the extreme case only one, fixed inquiry. The response RCi is use to infer a standard unknown key, which is along these lines managed by the implanting a model in the another way, for example as a unknown contribution for some crypt scheme. Opposite to Strong PUFs, the results of a Weak PUFs are not planned to be offered legitimately for the outside world.

This is an unusual type of random keys generating PUFs. Their preferred position is that they would be more enthusiastically to study out intrusively than non-unstable memory similar to EEPROM. Regular precedents incorporate the SRAM PUF, Butterfly PUF, and Coating PUF. Coordinated Strong PUFs has been proposed to construct Weak PUFs or Physically Jumbled Keys (POKs), in the case just a little subset of every single imaginable test is utilized.

1.1.2 Motivation

Electronic devices are playing an essential role in everyone's daily life. These are easy target for third party, which cause a overprotection and isolation issues. Standard data protection deliver several measurements against mentioned problems, but all are get hold

on the concept of a secret binary key. Classical cryptography proposes that the tools can hold a section of knowledge which is and which will remain, unknown to the opponent. Unfortunately, maintaining this requirement in practice could be stimulating: physical attacks such as invasive, semi-invasive, or side-channel attacks on the other side, just as programming things such as attacks, diseases, can prompt main introduction and complete breaks in safety. The way that the gadgets ought to be reasonable, portable, and cross-connected irritates the issue. The portrayed circumstance was one inspiration that prompted the advancement of PUFs. A PUF is a (somewhat) scattered physical structure S that can be tested with alleged outer upgrades or difficulties C_i, after that it responds with comparing reactions named R_{C_i}. In opposition to standard advanced frameworks, a PUF's responses will rely upon the nano scale auxiliary issue present in the PUF. This issue can't be cloned or duplicated precisely, not even by its unique producer, and is one of a kind to each PUF. Expecting the solidness of the PUF's reactions, any PUF S henceforth executes an individual capacity FS that maps move Ci to responses R_{C_i} of the PUF

Because of its unpredictable and orderless structure, a PUFs can keep away from a portion of the inadequacies related to advanced keys. It is typically harder to peruse out, foresee, or determine its reactions than to acquire the estimations of computerized keys put away in unstable memory. This reality has been abused for different PUF-based security conventions — unmistakable models including for ID and confirmation, vital trade or advanced rights the executive's purposes.

1.1.3 Scope of Work

Our assaults are focused mainly on the CRP side. Then they require a measure of CRPs that becomes just directly or logarithmic straightly In the applicable basic parameters of the assaulted PUFs, for example, their quantities of stages, XORs, feed-forward circles, or on the other hand, ring oscillators. The calculation times expected to infer the models (i.e., to prepare the utilized ML calculations) are low-degree polynomial, with one exemption: The calculation times for assaulting XOR Arbiter and Lightweight Reliable PUFs are super-polynomial in the number of the XORs. The precariousness of these PUFs likewise increments exponentially in their number of XORs, whence this parameter can't be brought freely up in handy applications. Act that as it may, the count of stages in the PUFs can be built without a considerable impact on insecurity.

For Strong PUFs, which consists of Arbiter PUFs, XOR Arbiter PUFs, Feed-Forward Arbiter PUFs, Lightweight Secure PUFs, and Ring Oscillator PUFs, we portray efficient

demonstration attacks. For PUFs of up to a specified size and unpredictability, then the advances function; our prototype model forecast rates regularly exceed the observed or determined the soundness of the distinct PUFs in these ranges.

So we can crack the type one PUFs safety that is based on broken PUFs in this job. The adds protocols, like those regarded in, to recognize, authenticate, exchange key, or handle digital freedoms. Under the situations of assumptions and attack outlined in Section 3.1, Our results also limit the use of broken Strong PUF architectures in Controlled PUFs and as Weak PUFs if we suppose digital values can be considered.

1.1.4 Thesis contribution

The thesis starts with chapter 1 that brief about Physical unclonable function and its type. This chapter also contains the motivation, scope of work, and our contribution toward attacks.

Chapter 2 is a review of previous approaches. We have discussed all the literature related to attack on PUFs, which includes the attack on various modelling attacks on PUFs likes XOR Arbiter PUF,Lightweight Secure PUF, Feed Foreword Arbiter PUF and Ring Oscillator PUF.

Chapter 3 covers the Methodology section, which includes how to attack.

This thesis is focused on machine learning attacks. We describe the technique of our Support Vector Machine tries in Section 3. We present our outcomes for different Strong PUF applicants. Our attacks are most probably on the CRP side. They need some amount of CRPs that grows strictly linearly or logarithmic linearly in the appropriate structural parameters of the attacked PUFs.

Chapter 2

Literature Review

2.1 Attack on PUFs

Displaying assaults on PUFs assume that a foe Eve's got one way or the other, gathered a subset everything being equal of the PUF, and attempts to get a numerical model from this data set, i.e., a PC calculation which effectively prophesies the PUF's reactions to subjective difficulties with high likelihood. On the off chance that useful, this breaks the security of the PUF and of any conventions based on it. It is known from before work that AI (ML) strategies are a characteristic and useful asset. How the desirable CRPs can be gathered relies upon the kind of PUF enduring an onslaught.

2.1.1 XOR Arbiter PUF

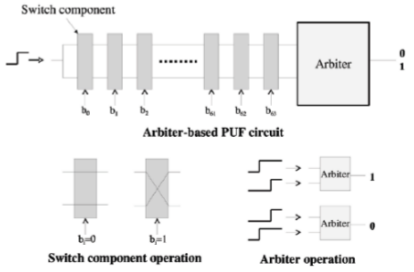

FIGURE 2.1: Arbiter PUF

One plausibility to fortify the flexibility of referee structures against AI, which has been recommended in [7], is to utilize l person Arb-PUFs in parallel, with k stages each. A

similar test C is related to every one of them, and their own output t_i are XORed in order to yield a global responses t_{xor}.We denote such a model as l-XOR Arb-PUF. The following can be taken from a formal model for the XOR Arb-PUF. It maintains the protocol ti 1, 1 as before.

$$t_{XOR} = \prod_{i=1}^{l} sgn(\overrightarrow{w_i^T}) = sgn(\coprod_{i=1}^{l} \overrightarrow{w_i^T}) \tag{2.1}$$

$$= sgn(\bigotimes_{i=1}^{l} \overrightarrow{w_i^T} \bigotimes_{i=1}^{l} \overrightarrow{\varphi_i}) = sgn(\overrightarrow{w_{XOR}^T \varphi_{XOR}}) \tag{2.2}$$

Where equation 2.1 gives a non-linear option limits and equation 2.2 gives liner decision boundary.

In the application of ES and SVMs to XOR Arbiter PUFs , we were able to break little occurrence , LR on Arbiter PUFs

ML Algorithm	Bits	Prediction Rate	CRPs	Time to taken
LR	64	95%	640	0.01 sec
		99%	2,555	0.13 sec
		99.9%	18,050	0.60 sec
LR	128	95%	1,350	0.06 sec
		99%	5,570	0.51 sec
		99.9%	39,200	2.10 sec

TABLE 2.1: Prediction Rate on Arb PUFs with 64 and 128 bits

2.1.2 Lightweight Secure PUF

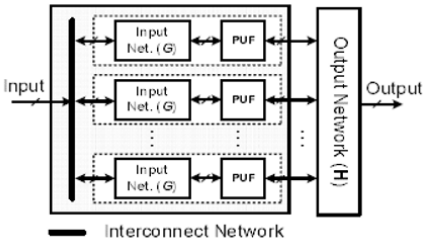

FIGURE 2.2: Lightweight PUF

Another kind of PUF, which is known as Lightweight Secure PUF or Lightweight PUF for short, has been discussed in [9]. It is like the XOR Arb-PUF of the final area. At its heart are l person standard Arb-PUFs masterminded in parallel, each has k stages, which

produce singular reactions/yields r_1, . . . , r_l. These individual yields are XORed to create multi-bit yield o_1, ..., o_m of the Lightweight PUF, as indicated by the recipe

$$o_j = \bigotimes_{i=1,.....,x} r_{(j+s+i)} modl \tag{2.3}$$

where j = 1,.....,m

Consequently, the qualities for m (the quantity of yield bits of the LW PUF), x (the quantity of qualities rj that impact each single yield bit) and s (the roundabout move in picking the x esteems rj) are variable plan parameters. Another distinction to the XOR Arb-PUFs lies in the l inputs

$$C_1 = b_1^1 .....b_k^1, C_2 = b_1^2 .....b_k^2, ...C_l = b_1^1 .....b_k^l \tag{2.4}$$

which are connected to the l singular Arb-PUFs. In spite of XOR Arb-PUFs, it doesn't hold that $C_1 = C_2 =$ . . . $= C_l =$ C, in any case, a progressively confused information mapping that infers the person inputs Ci from the worldwide info C is connected. This info mapping establishes the most critical distinction between the LW PUF and the XOR Arb PUF. We allude the peruser for further subtleties. So as to foresee the entire yield of LW PUF, one can apply comparative models and ML procedures as in the last area to foresee its single yield bits o_j . While the likelihood to foresee the full yield obviously diminishes exponentially in the misclassification rate of a solitary piece, the strength of the full yield of the LW PUF too diminishes exponentially in similar parameters. It accordingly appears to be reasonable for assault it in the depicted way; regardless, our outcomes challenge the bit security of the LW PUF.

Bit Size	Prediction Rate	Number of XORs	CRPs	Time
64	99%	3	6,000	8.9 sec
		4	12,000	1:28 hrs
		5	300,000	13:06 hrs
128	99%	3	15,000	40 sec
		4	500,000	59:42 min
		5	100,000	267 days

TABLE 2.2: LR on LW PUFs. Prediction rate

2.1.3 Feed Foreword Arbiter PUF

Feed Forward Arbiter PUFs (FF Arb-PUFs) were presented in and next discussion. A portion of the multiplexers are not exchanged in reliance of an outer test bit, however

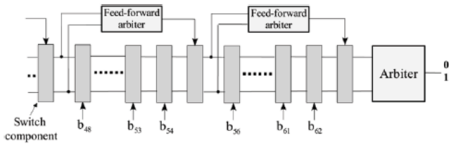

FIGURE 2.3: Feed foreword PUF

Bit Length	FF-loops	Prediction Rate	CRPs	Training Time	
64	6	97.72%	50,000	07:51 min	
	7	99.38%	50,000	47:07 min	
	8	99.50%	50,000	47:07 min	
128	6	99.11%	50,000	3:15 hrs	
	7	97.43%	50,000	3:15 hrs	
	8	98.97%	50,000	3:15 hrs	

TABLE 2.3: Prediction Rate on FF Arbiter PUFs.

as a capacity of the postpone contrasts amassed in before parts of the circuit. Extra referee parts assess these postponement contrasts, and their yield bit is bolstered into said multiplexers in a "feed-forward circle" (FF-circle). The quantity of circles as well as the beginning and end purpose of the FF-circles are variable structure parameters. It would be ideal if you note that a FF Arb-PUF with k-bit difficulties C = b_1.....b_k and l circles have s = k + l stages which contains multiplexers. Machine Learning Result The depicted reliance makes characteristic design models of FF Arb-PUFs no longer differentiable.This way, FF Arb-PUFs can't be assaulted conventionally with ML strategies that require straightly distinguishable of differentiable models (like SVMs or LR), even though such models can be found in unique cases, for instance for limited quantities of non-covering circles, and so on.

We tried different things with LR and SVMs on FF Arb-PUFs, utilizing various models and information portrayals, however could just break different cases with little quantities of non-covering FF circles, for example, l = 1, 2. This is in concurrence with prior outcomes revealed in [16].

The utilization of ES at long last enabled us to handle considerably more complex FF-structures with up to 8 FF-circles. All circles have equivalent length, and are disseminated routinely over the PUF, with covering begin and endpoints of progressive circles. Table 6 demonstrates the outcomes we acquired. If you don't mind note for contact that in-silicon executions of 64-bit FF Arb-PUFs with 6 FF-circles are known to have a natural solidness of 89.96%.

2.1.4 Ring Oscillator PUF

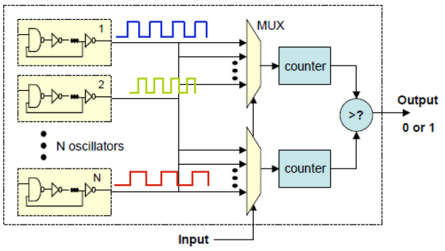

FIGURE 2.4: RO PUF

Ring Oscillator PUFs (RO-PUFs) were examined in [7].They depend because of manufacture minor departure from the recurrence of a few, indistinguishably structured R oscillators.While portrays the utilization of RO PUFs with regards to Controlled PUFs and constrained check verification, it merits dissecting them as strong PUFs competitor. A RO-PUFs comprises of k such oscillators, every one of which has its own, novel recurrence brought about by fabricating resistances. The contribution of a RO-PUFs comprises of a tuple (i, j), which chooses two of thek oscillators. Their frequencies are analysed, and the yield of the RO-PUF is "0" if the previous wavers quicker than the last mentioned, and "1" else. A ring oscillator can be demonstrated in a direct style by a tuple of frequencies (f_1, . . . , f_k). Its yield on input (I, j) is "0" if $f_i > f_j$, and "1" else. There are a few techniques to assault a RO-PUF. The most direct endeavour is a basic perused out all things considered. This is simple, since there are simply k(k− 1)/2 = O(k^2) CRPs of intrigue. On the off chance that Eve can pick the CRPs adaptively, she can utilize a standard arranging calculation to sort the RO-PUF's frequencies (f_1, . . . , f_k) in climbing request. This methodology hence enables her to foresee all yields with 100without knowing the precise frequencies fi themselves. The time and CRP complexities of the particular arranging calculations are notable ; for instance, there are a few calculations with normal and even most pessimistic scenario CRP unpredictability of N_{CRP} = O(k · log k). Their running occasions are too low-degree polynomial. The most intriguing case for our examinations is when Eve can't adaptively pick the CRPs she gets, yet at the same time needs to accomplish ideal expectation rates. This case happens by and by at whatever point Eve gets her CRPs from convention listening in, for instance. We completed examinations for this case, wherein we connected Quick Sort (QS) to arbitrarily drawn CRPs. The outcomes are appeared Table 8. The evaluated required number of CRPs is shown by :

$$N_{CRP} = \frac{k(k-1)(1-2\varepsilon)}{2+\varepsilon(k-1)} \tag{2.5}$$

what's more, the preparation times are polynomial low-degree. Eqn. 2.5 measures restricted check confirmation capacities of ROPUFs.

Chapter 3

METHODOLOGY

3.1 The Procedure of PUF Modelling and its Main Challenges

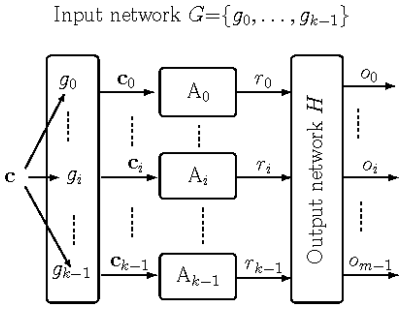

FIGURE 3.1: Capturing the Response data

We will presently talk about the essential procedure of AI Based displaying and its chief difficulties in more unique detail. The demonstrating method mainly is a 2 advance strategy. Its starting step comprises of setting up an inside,parametric prototype from the PUF. This needs you to find a capacity F which effectively depicts the PUF's test reaction conduct (input or yield conduct). F should take as info I a test C_i which is connected to the PUF, and (2) values that depict the inward, one of a kind, making assistant parameters of the puf. The last are usually given by some many dimension parameter vector w with qualities in the reals or rationals. F at that point will yield the right comparing reactions R_{c_i} of PUFs on test C_i, i.e., $F(W, C_i) = R_{C_i}$. The parametric model F is employed for PUF learning in a time step along with a fairly selected ML calculation. The calculation takes as info an enormous arrangement of CRPs of the PUF,the purpor-ted

"preparing set". Its will likely locate a solid vector W that prompts a decent forecast nature of the PUF. This quality is assessed on a moment, free CRP set, that CRP is called "test set".One imperative viewpoint in this setting is that the vector W determined by the ML calculation doesn't have to be equivalent to the "genuine" parameter esteems w of the considered, phy-sical PUF occurrence. Contingent upon the PUF design also, the model F, a wide range of vectors can prompt a relatively pro-portionate yield conduct, and recognizing one of them may be adequate for a decent forecast quality.

What are the difficult challenges in this procedure? The difficult task comprises of finding an appropriate model of the con-sidered PUF at all. Given some learning about the physical systems basic the PUF (e.g., about the utilized coordinated cir-cuits or on the other hand optical frameworks), in any case, a parametric model generally can be grown decently effectively. Finding a computationally productive model can be more enthusiastically. Keeping that in mind, the basic instruments in the PUF may should be improved or appropriately nearly.

A moment issue is to recognize a fit ML calculation that performs effectively on the considered PUF. In the perfect case, the presentation ought to be polynomial in some framework parameter (commonly in the PUF's test bitlength) and in the sought forecast quality. In any case, since PUFs can't be scaled uncertainly because of expense and security imperatives, too mellow exponential development rates might be adequate in certain circumstances, as long as the calculation still can break for all intents and purposes applicable PUF sizes. One surely understood the case for this impact is the demonstrating assaults on XOR Arbiter PUFs: The down to earth dependability of these PUFs diminishes Their amount of XORs is exponential. Current assaults have a multifaceted nature that increments exponentially in a similar parameter, yet at the same time reach nece-ssarily relevant size and multifaceted nature levels .

3.2 Employed Machine Learning Algorithm

3.2.1 Support Vector Machine

"Support Vector Machine" (SVM) is a calculation of administered AI that can be used for both arrangement and relapse problems. In any event, it is used in arrangement problems for the highest portion.In this calculation, we plot each data as a point in n-dimensional space (n are several highlights we have) with each element being estimated as a particular structure. We conduct order at that stage by discovering the hyperplane that significantly

distinguishes the two classes (take a gander in the picture below). The objective of the assistance vector machine estimation is to find a hyperplane in N-dimensional space(N—the number of highlights) that especially arranges the data centers. SVM recognized that your information sources are numeric as a matter of course. You may need to undercover all information sources off happening to parallel sham variables (one variable for each class). For example, there are diverse useful hyperplanes that could be picked to isolate the two kinds of information, either 1 or 0. For instance, we will likely discover a plane with the most exceptional edge, the most outrageous partition behind the two classes between information motivations. Improving the edge partition makes it easier to work with future data centers. Hyperplanes are restraints of choice that assist to master data centers. The data centers can be attributed to separate classes around dropping on either side of the hyperplane. Also, the hyperplane element relies on the number of characteristics. The hyperplane is just a line in case the quantity of information characteristics is 2, by then. If the information characteristics are 3, the hyperplane will move to a two-dimensional plane.It ends up being hard to imagine when the number of characteristics exceeds 3. SVM is an ML technique that is prepared to bring from a lot of preparation models into a double order model. Known preparatory models are mapped into a higher-dimensional space in the learning phase to loosen the order assignment.The learning calculation tries to find a reasonable isolating hyperplane that allows you to deal directly with problems that are not directly divisible in the first information space. The isolating hyperplane ought to have the most significant conceivable separation between information vectors having a place with various classes, and the contributions with negligible separation to the isolating hyperplane are called bolster vectors. The isolating hyperplane is worked with the assistance of two parallel supporting hyperplanes through the relating bolster vectors. The separation between the supporting hyperplanes is known as the edge. The essential thought of structure a conventional SVM is to augment the side while limiting the grouping mistake. These clashing objectives are exchanged off by a regularization parameter. Prepared SVMs depend intensely on the self inward result of the mapping capacity, called portion, assessed individually on the help vectors and on the test to be grouped. Three usually utilized pieces portrayed beneath.

3.2.1.1 Linear Kernel SVM

The linear kernel equation is shown below:

$$K(x, x_i) = sum(x * x_i) \tag{3.1}$$

The kernel characterizes the proximity or a measure of separation between current data and vectors of assistance. The spot element is the metric of proximity used for linear SVM or a straight part because segregation is a straight blend of information sources. For example, a Polynomial Kernel and a Radial Kernel can be used to change the info space into higher measurements. Using progressively complex parts is appealing as it allows lines to isolate bent or much more mind-boggling. Like classes can prompt gradually accurate classifiers.

3.2.1.2 Polynomial Kernel SVM

The Polynomial kernel equation is shown below.

$$K(x, x_i) = 1 + sum(x * x_i)^d \tag{3.2}$$

Where the polynomial order has to be hand-designated to the algorithm of machine learning, it's the same as the linear kernel when d=1. In this kernel enables input room for curved lines.

3.2.1.3 Radial Kernel SVM

The Radial kernel equation is shown below.

$$K(x, x_i) = e^{(-gamma * sum(x - x_i^2))} \tag{3.3}$$

Where gamma is a learning algorithm parameter, an attractive gamma default is 0.1, where gamma is between 0 and 1. This bit is a tremendous neighborhood and, like shutting polygons in two-dimensional space, can create complex areas within the scope of the element.

3.2.2 Logistic Regression

Strategic Regression is a well-researched managed AI system, which has been portrayed, for a model, in [18]. And its used to PUFs with single-piece yields, each test C = $b_1....b_k$ is doled out a likelihood $p(c, t \mid \overrightarrow{w})$ that it creates a yield in between negative one to positive one (for specialized reasons, one makes the show that t 1, 1 rather than 0, 1). In this way, the vector w encodes the relevant internal parameters of the individual PUF for

particular runtime delays to occur. The probability is provided by the calculated sigmoid following a vector-parameterized ability f(w) w as

$$p(c, \mid w) = \sigma(ft) = (1 + e^{-ft})^{-1} \tag{3.4}$$

. In this way, f decides through f = 0 a choice limit of equivalent yield probabilities. For a set of CRPs M, the limit is located by selecting the parameter vector w so that the probability of viewing this set is maximum, the negative likelihood of logging is small separately:

$$w = argmin_w(M, w) = \sum_{(C,t)\varepsilon M}^{n} -lm(\sigma(tf(w, C))) \tag{3.5}$$

As there is no logical answer to deciding the idealThe w parameter vector must be developed iteratively, e.g., by using gradient data.

$$\bigtriangledown\, l(M, w) = \sum_{(C,t)\epsilon w}^{n} t(\sigma(tf(w, C)) - 1) \bigtriangledown\, f(w, C) \tag{3.6}$$

From the unique advancement techniques which we tried in our ML tests (standard slope drop, iterative re-weighted least squares, RProp[20][21]), RProp inclination plummet performed best. Need not be (roughly) straightly detachable in highlight space, as is required for fruitful utilization of SVMs, yet only differentiable. In our ML tests, we utilized the usage of LR with RProp modified in our gathering, which has been put online under[20]. The emphasis goes on until we achieve a union point, i.e. until the average prediction value of two sequential squares of 5 back to back cycles is discovered no longer increases just because. In the event that the came to execution on the preparing set after installation is not sufficient, the operation starts again. On the test set, the expectation error is calculated after the addition to a reasonable agreement on the training set. The whole process is like the preparation of an Artificial Neural System (ANN)[18]. The PUF model requires time postponements after an ANN's loads after the coordinate. Like ANNs, we discovered that RProp makes a large difference in the LR's intermingling speed and reliability (with RProp only a few XOR-PUFs were learning). Even with RProp, however, the defer set may end up in a location of the search room where no (nearly least) accommodating inclination information are available. In such a situation, we encounter the above-mentioned situation of uniting on an agreement not sufficiently accurate and need to restart the operation.

3.2.3 Evolution Strategies

Evolution Strategies (ES) [21, 22] have a place with an ML sub-field known as populace based heuristics. They are propelled by the developmental adjustment of people to certain ecological conditions.One person in the masses is given for our situation by a solid instantiation of time delays in execution of PUF, i.e. by a particular instantiation of the w vector that appears in Equation 3.1 Moreover 3.2. The individual's natural well-being is dictated by how well he (re-)creates the objective PUF's correct CRPs on a particular set of CRPs. Evolution Strategies is going through a few cycles of development or alleged ages. The test reaction conduct of the population's best people is closer and closer to the objective PUF with an increasing number of generations. ES is a random order method that needs neither a (approximately) straightforwardly distinguishable problem (such as supporting vector machines) nor a differentiable architecture; a parameterizable model does the trick. Since all known electrical PUFs can be parameterized efficiently, ES is a very suitable method of attack. We used an in-house use of ES from our PyBrain AI library[23]. Throughout this paper, the meta-parameters in all ES uses are (6,36)-certainty and a $T = \frac{1}{\sqrt{n}}$. We used a scheme called Lazy Evaluation (LE) other than sometimes. LE suggests that not all of the preparation set's CRPs are used to survey the biological health of an individual; instead, only an erratically selected subset is used for assessment. If LE has been used.

3.3 Challenge-Response Pairs Generation

Hypothetically we have 2^N reactions for n bit difficulties if PUF is 16 bits than all-out CRP is 2^{16}. Given a solid PUF-engineering that ought to be checked, the challenge-reaction sets so we utilized in our ML investigations was created pseudo-arbitrarily in the accompanying design: (1) The postpone estimations of the given PUF were chosen any value as indicated by a standard ordinary circulation. (2) A lot of difficulties was chosen consistently at irregular from all potential problems/inputs. (3) The comparing reactions were marked by utilization of the defers chose in step (1), and by use of a directly added substance postpone model [12]. The created CRPs are like this utilized for preparing and testing the ML calculation. 2/3 CRPs have been used to prepare the set, the rest have been used as test set. A set amount of CRPs were used for testing XOR and Lightweight PUFs. We will utilize the accompanying definitions all through the work: NTrSet is the quantity of CRPs in the preparation set. N_{Teset} is the quantity of CRPs in the test set. N_{CRP} is the

absolute number of utilized CRPs,

$$N_{CRP} = N_{Trset} + N_{Teset} \tag{3.7}$$

The forecast mistake is the proportion of off base reactions of the prepared ML calculation when assessed on the test set. It is the number of inaccurate reactions of the prepared model on the test set separated by the number of CRPs in the test set.

3.4 Attack on Physical Unclonable Functions

To analyse attacks on Arbiter PUFs We utilize a prophet access model. A foe that needs to play out an assault on a PUF speaks with a prophet. Through the prophet, the enemy can acquire any number of authentic PUF reactions to (adaptively) picked difficulties. Notice that responses to rising to problems don't have dependably a similar worth since responses can be loud,. Note that we do not consider attacks where the attacker has physical knowledge of the PUF as a very specific assumptions would have to be made. Any attacker can be understood as a probabilistic PAC learning An algorithm that returns a model of the PUF, as first used in a study by Ganji et al. [GTS16]. The probabilistic algorithm has two properties $0 \leq \delta \geq 1$ and $0 \leq \varepsilon \geq 1$. We call δ the confidence parameter and ε the accuracy parameter of the algorithm. It outputs with probability $1 - \delta$ a model of the PUF that has error rateε. We call an attack successful if $\delta < \frac{1}{2} - c$ for a constant c and if $\varepsilon < \frac{1}{2} - c^{'}$ for a constant $c^{'}$. It is sufficient ford to have a constant distance from $\frac{1}{2}$ obtain a reasonable result, as we can improve the value of δ with s repetitions of the algorithm to $\delta^{'} < 1 - \frac{1}{2^{cs}}$. The accuracy is less easy to be improved. Schapire [Sch90] presented the first provable polynomial time algorithm that boosts the accuracy.

3.4.1 Modelling attacks using Support Vector Machine classifiers

Support Vector Machine classifiers have been utilized rather than straight programming. SVM classifiers locate a most significant edge hyperplane that isolates the 0 and 1 reactions. A preparation test comprising of a set number of CRPs is utilized to assemble the model. This model is used to anticipate future effects on an arbitrarily chosen set of difficulties. As appeared, SVM assaults can accomplish high forecast exactness more noteworthy than 90% with a preparation test set of around 2/3rd of total CRPs. While breaking down security, a unique viewpoint that should be examined is the similitude among protection and unwavering quality. In solicitation to bargain the security of a PUF, the forecast

precision accomplished through displaying assaults must be at any rate higher than the devotion performed by the PUF. Even though SVM assaults can perform forecast rates of 90%, it might at present demonstrate deficient in mimicking a PUF example. To expand further, the verification procedure of a PUF circuit with an unwavering quality of 95% will expect in any event 94 reactions out of 100 to be right. Consequently, a forecast rate of 90% isn't adequate since a product the confirmation procedure would dismiss PUF model, which gets just 90 out of 100 CRPs right. Adding non-linearity to PUF circuits can break this straight added substance postpone model and foil SVM assaults. Feed forward PUFs, XOR referee PUFs, and light weight secure PUFs have been proposed to oppose such demonstrating assaults. In this work, we are utilizing straight piece Support Vector Machine. The object of the speck is known as the piece and can be recomposed as:

$$K(x, x_i) = sum(x * x_i) \tag{3.8}$$

The piece characterizes the similarity or a measure of separation between current data and vectors of assistance. The spot element is the metric of proximity used by SVM or a straight piece because divorce is a mix of data sources through a combination. Different bits can be used,Like a Radial Kernel and a Polynomial Kernel, to alter the info room into more meaningful measurements. It is alluring to utilize progressively complex portions asIt allows lines to isolate bent or progressively mind-boggling courses. This thus can prompt increasingly exact classifiers.Scikit-learn is Python's free AI library. It features various figurings like assistance vector machine, random forests, and n-neighbours, and it in like manner supports Python numerical and legitimate libraries like NumPy and SciPy. In this library, we will make sense of how to code python and implement Machine Learning by the help of the scikit-learn library, which was created to functioning AI in Python more comfortable and progressively dependable. We will import the informational collection utilizing pandas, investigate the information using pandas strategies like head(), tail(), types(), and after that take a stab at using plotting methods from Seaborn to picture our information. At that point, we'll jump into scikit-learn and use preprocessing.LabelEncoder() in scikit-figure out how to process the information, and train test split() to part the informational collection into test and train tests. We will likewise utilize a cheat sheet to enable us to choose which calculations to use for the informational index. At long last, we will utilize three distinct considerations (Naive-Bayes, LinearSVC, K-Neighbors Classifier) to make forecasts and analyse their presentation using techniques like precision score() given by the scikit-learn library. We will likewise picture the presentation score of various models utilizing scikit-learn and Yellowbrick representation. To capitalize on this post, you ought

to most likely as of now be alright with: pandas essentials Seaborn and matplotlib nuts and bolts

Chapter 4

Analysis of Results and Discussion

4.1 Results Analysis

We had the choice to break small cases when using Support Vector Machine Algorithm to XOR Arbiter-PUFs, e.g. XOR Arbiter with 2 and 3 XORs and 64 steps. The other two systems were outstripped by LR. Instead of choosing the straight decision limit, the observation is necessary.I used the SVM to crack the PUF effectively and also compare the results with another paper[1] which is used Linear Regression method.

4.1.1 Prediction Rate

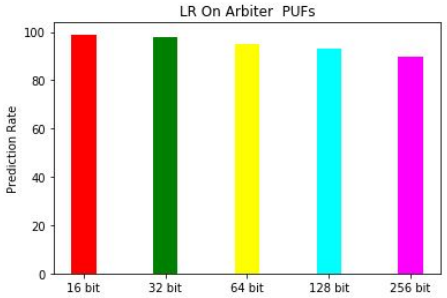

FIGURE 4.1: Prediction Rate

4.1.2 Prediction Error Rate

We assessed the exhibition on blunder caused CRPs as for SVMs and ES Arb PUFs.

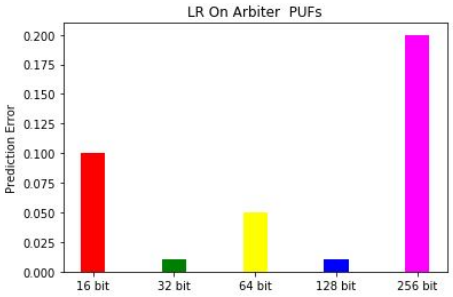

FIGURE 4.2: Prediction Error Rate

4.1.3 Iteration

Iteration in computing is the method that marks a specified amount of repetitions from a block of statements within a computer program. Whene number of iteration increases than the accuracy is also increase.

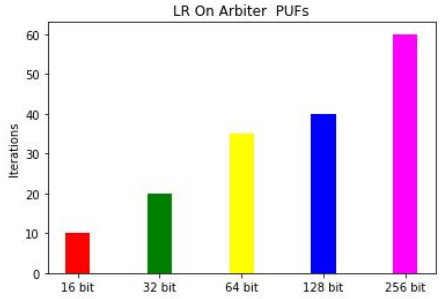

FIGURE 4.3: Iteration

4.2 Discussion

Two clear, yet one-sided investigations of the current modelling attack would be the accompanying: 1. All Strong PUFs are hazardous. 2. The long haul security of Strong electrical PUFs can be recouped inconsequentially, for instance, by expanding the PUF's Bit length. The two perspectives are shortsighted and increasingly included. The present assaults are undoubtedly satisfactory to break a few postponements based PUF usage. In any way, there are a few different cases of how PUF planners may most likely battle back in future structures.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The strikes presented in this work through cutting edge executions additionally, new ML systems. An introduction relationship between's our results and earlier systems that used SVMs even, for all intents and purposes indistinguishable methodologies [18], [19] avows the considerable Impact of the choice of the right ML-figuring. Another, abstractly fresh path is to combine displaying strikes with additional data captured directly from physical PUF estimates. For instance, utilizing the proportional test to distinct occurrences shows a reaction bit's noise level. It involves decisions to distinguish in the PUF about the all-out assessment of the last runtime. Such side channel data can enhance performance and combine ML system prices. Some fundamental moves toward this end were produced uniquely beginning in a few works late.

5.2 Future Work

The next years/Future will observe some challenge between code creators and code breakers in the territory of Strong PUF. Like the plan of old-style crypto natives, for instance, stream figures, this procedure can be required to combine sooner or later to arrangements that are strong against The known assaults. Some first endeavors into this heading have as of now been made in [20], [21] [20]yet their ML versatility has not been dissected fully in writing For PUF originators, it might premium explore a portion of the ideas that we referenced previously. One essential objective will be the development of Strong down to earth

PUFs that still have high demonstrating versatility. Note in this setting any non-linearity in the PUF configuration builds its ML-resilience yet. Improving is a helpful starting point for PUF breakers.The strikes presented in this work through cutting edge executions additionally, new ML systems. An introduction relationship between's our results and earlier methods that used SVMs even, for all intents and purposes indistinguishable methodologies [21], [22] avows the considerable Impact of the choice of the right ML-figuring.Another, abstractly fresh path is to combine displaying strikes with additional data obtained from direct estimates of physical PUF or side channels. For instance, applying the proportional test to various occurrences shows a reaction bit's noise level. It involves decisions about the last runtime's all-out estimate to distinguish in the PUF. Such side channel data can enhance performance and combine ML systems prices. Some fundamental moves towards this end were produced uniquely beginning in a few works late.

Bibliography

[1] J. Ye, Q. Guo, Y. Hu, H. Li and X. Li, "Modeling attacks on strong physical unclonable functions strengthened by random number and weak PUF," 2018 IEEE 36th VLSI Test Symposium (VTS), San Francisco, CA, 2018, pp. 1-6

[2] J. Guajardo, S. S. Kumar, G.-J. Schrijen, P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection", CHES, pp. 63-80, 2007.

[3] L. Zhang, C.-H. Chang, Z. H. Kong, C. Q. Liu, "Statistical Analysis and Design of 6T SRAM Cell for Physical Unclonable Function with Dual Application Modes", ISCAS, pp. 1410-1413, 2015.

[4] P. Prabhu, A. Akel, L. Grupp, W. K. Yu, G. Suh, E. Kan, S. Swanson, "Extracting Device Fingerprints from Flash Memory by Exploiting Physical Variations", ICTTC, pp. 188-201, 2011.

[5] S. Rosenblatt, S. Chellappa, A. Cestero, N. Robson, T. Kirihata, S. S. Iyer, "A Self-Authenticating Chip Architecture Using an Intrinsic Fingerprint of Embedded DRAM", IEEE JSSC, vol. 48, no. 11, pp. 2934, 2013.

[6] P. Koeberl, U. Kocabas et al., Memristor PUFs: A New Generation of Memory based Physical Unclonable Functions, 2013.

[7] D. Nedospasov, J.-P. Seifert, C. Helfmeier, C. Boit, "Invasive PUF Analysis", FDTC, pp. 30-38, 2013.

[8] C. Helfmeier, C. Boit, D. Nedospasov, J.-P. Seifert, "Cloning Physical Unclonable Function", HOST, 2013.

[9] D. Nedospasov, J.-P. Seifert, C. Helfmeier, C. Boit, "Invasive PUF Analysis", FDTC, pp. 30-38, 2013.

[10] A. Roelke, M. R. Stan, "Attacking an SRAM-Based PUF through Wearout", ISVLSI, pp. 206-211, 2016.

[11] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, S. Devadas, "Extracting Secret Keys from Integrated Circuits", IEEE TVLSI, vol. 13, no. 10, pp. 1200-1205, 2005.

[12] U. Ruhrmair, F. Sehnke, J. Solter, G. Dror, S. Devadas, J. Schmidhuber, "Modeling Attacks on Phyiscal Unclonable Functions", CCS, 2010.

[13] U. Ruhrmair, J. Solter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, S. Devadas, "PUF Modeling Attacks on Simulated and Silicon Data", IEEE TIFS, vol. 8, no. 11, 2013.

[14] A. Vijayakumar, S. Kundu, A Novel Modeling Attack Resistant PUF Design based on Non-Linear Voltage Transfer Characteristics, pp. 653-658, 2015.

[15] R. Kumar, W. Burleson, "On Design of a Highly Secure PUF Based on Non-Linear Current Mirrors", HOST, pp. 38-43, 2014.

[16] A. Vijayakumar, V. C. Patil, C. B. Prado, S. Kundu, "Machine Learning Resistant Strong PUF: Possible or a Pipe Dream", HOST., pp. 19-24, 2016.

[17] Q. Guo, J. Ye, Y. Gong, Y. Hu, X. Li, Efficient Attack on Non-Linear Current Mirror PUF with Genetic Algorithm, pp. 49-54, 2016.

[18] U. Ruhrmair, D. E. Holcomb, PUFs at a Glance, DESIGN, AUTOMATION AND TEST IN EUROPE 2014.

[19] M.-D. Yu, D. MRaihi, R. Sowell, S. Devadas: Lightweight and Secure PUF Key Storage Using Limits of Machine Learning. CHES 2011

[20] M. Majzoobi, M. Rostami, F. Koushanfar, D.S. Wallach, and S. Devadas: Slender PUF Protocol: A Lightweight, Robust, and Secure Authentication by Substring Matching. IEEE SP Workshops, 2012.

[21] Q. Chen, G. Csaba, P. Lugli, U. Schlichtmann, U. Ruhrmair. TheBistable Ring PUF: A New Architecture for Strong Physical Unclonable Functions. HOST 2011.

[22] D. Lim. Extracting Secret Keys from Integrated Circuits. Msc thesis, MIT, 2004.

[23] Erdinc O zturk, Ghaith Hammouri, and Berk Sunar. Towards robust low cost authentication for pervasive devices. IEEE PerCom, 2008.