PUF Modeling Attack Literature Review

Dazhi Li
dept. ECE
Rutgers University
New Brunswick, the United States
dl939@scarletmail.rutgers.edu

***Abstract*—This report is mainly discussing about PUF (Physical Unclonable Function) modeling attack strategies. An overview about methods from traditional machine learning to deep learning which can be applied to compromise the Strong PUFs.**

***Index Terms*—machine learning, deep learning, PUFs, security, hardware**

I. Introduction

As IoT (Internet of things) is becoming popular nowadays, IoT security is much more important for users' security. PUF (Physical Unclonable Function) is a typical hardware for hardware authentication. Many people call it hardware fingerprints as no PUF is the same due to the manufacture difference. However, some research found that adversary could use machine learning to generate a software based PUF model which break the law that any PUF is impossible to clone. Here, I am going to lead you to review ”PUF Modeling Attacks Intro”[1], ”Modeling Attacks on Physical Unclonable Functions”[2] and ”A Fast Deep Learning Method for Security Vulnerability Study of XOR PUFs”[3]. These papers is going to give you a brief idea about PUF modeling attack and then we are going to talk about some technique details of the machine learning modeling attack as a traditional method. Then, we will see how technique improves in modeling attack with fast deep learning method. All these papers are chosen for leading the readers to understand the modeling attack ideas and improvements these years.

II. Types of PUF And Modeling Attack

What are PUF modeling attacks? Under which conditions and to which PUF types are they applicable? In general, modeling attacks on PUFs presume that an adversary collected a subset of CRPs (Challenge Response Pairs) of the PUF. He/She then tries to derive a numerical model from this CRP data, i.e., a computer algorithm which correctly predicts the PUF's responses to arbitrary challenges with high probability. But not all the PUFs can be attacked through this way. Here we are going to discuss about different types of PUFs and how immune they are to modeling attack.

- Strong PUFs: Strong are the PUF class for which modeling attacks have been designed originally, and to which they are best applicable. As there is no protection mechanisms to restrict the adversary to get CRPs from them. Typical strong PUFs includes Arbiter PUF, XOR PUF etc.
- Weak PUFs: Weak PUF may have very few challenges. Their response are used to derive a standard key. So the responses of a weak PUF are never meant to be given directly to the outsiede world. In other words, weak PUFs are the PUF class that is the least susceptible to modeling attacks. Weak PUF could be compromised by the modeling attack if a strong PUF, embedded in some hardware system, is used to derive the physically obfuscated key[2].
- Controlled PUFs: A controlled PUF uses a strong PUF as a building block, but adds control logic(like hash algorithms) that surrounds the PUF. The logic prevents challenges from being applied freely to the PUF, and hinders direct read-out of its responses. This logic can be used to thwart modeling attacks. However, if the outputs of the embedded Strong PUF can be directly probed, then it may be possible to model the Strong PUF and break the Controlled PUF protocol.

III. Overview Through Machine Learning Methods

At the very beginning, I would like to mention that all these papers are focusing on breaking the strong PUF's security as other types of PUF are immune to modeling attack unless attacker could directly access the strong PUF inside them. In this section, we are going to talk about [1] and [2] as the traditional modeling attack method. We will discuss how they apply and evaluate their machine learning methods.

A. Problem statement

The problem they are addressing is finding a way to compromise the strong PUF to show that PUF design is not perfect. Although it is not able to physically produce two PUF which are the same. People could use machine learning algorithms to copy a software based model. This will help PUF designers better understand the adversary's method to reinforce the PUF's security.

B. Challenges

The major challenge these authors faced is finding a suitable machine learning algorithm to simulate the delay based PUF's

Identify applicable funding agency here. If none, delete this.

weights. And due to the diversity of strong PUF, one algorithm is not suitable for all the kinds of srong PUFs. They also need to evaluate on certain PUF, which methods is better applicable.

C. Methodology

There are two methodology these authors had applied to storng PUFs.

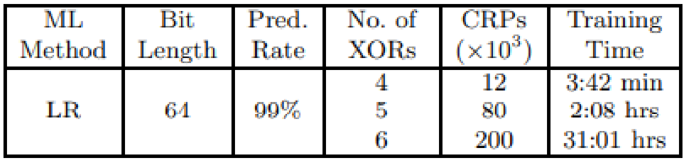
- Logistic Regression
  Logistic Regression (LR) is a well-investigated supervised machine learning framework that some people are very familiar. We assume that a delay based strong PUF is getting the response *RC* from the function

$$F(C, w) = RC \tag{1}$$

  Here the *C* means the challenge. We assume that the *w* is the weights of the model which stands for the delay of the strong PUF. Logistic Regression is a method to finding the correct *w* to build a software based model which could achieve the function *F*. The whole process is similar to training an Artificial Neural Network (ANN). The model of the PUF resembles the network with the runtime delays resembling the weights of an ANN. Similar to ANNs, they found that RProp makes a very big difference in convergence speed and stability of the LR (several XOR-PUFs were only learnable with RProp
- Evolution Strategies
  Evolution Strategies (ES) belong to an ML subfield known as population-based heuristics. They are inspired by the evolutionary adaptation of a population of individuals to certain environmental conditions. In our PUF modeling attack case, they use a lot of individual models to check how fitness they have to fit the environment. Here the environment means the CRPs. If they fit the CRPs well, the model could be saved to the next iteration until they find a good one that achieve their target accuracy. ES runs through several evolutionary cycles or so-called generations.

D. Evaluation metrics and result

The evaluation matrics here is the prediction accuracy, time they trained and CRPs they used. These three metrics is checking the how similar the software based model is to the original silicon based PUF, how fast they can compromise certain PUF and how smaller dataset they need to feed their model. To connect these two paper to the later deep learning method one, I just post one important table which could be compared to the deep learning one. Other results will be shown in words.



ML Method	Bit Length	Pred. Rate	No. of XORs	CRPs ($\times 10^3$)	Training Time
LR	64	99%	4 5 6	12 80 200	3:42 min 2:08 hrs 31:01 hrs

Fig. 1. LR on different XOR Arbiter PUFs [2]

As the Fig.1 shows that LR method's performance on N-XOR arbiter PUF. We can easily find that the amount of CRPs needed is huge and the training time is exponentially increasing while the XOR gates is increasing linearly. Also, these authors did research on other kinds of strong PUFs. It shows that LR method is good at Lightweight secure PUFs too. It could compromise a lightweight secure PUF with 3 XOR gates in 8.9 seconds with a high accuracy of 99%. The CPRs needed is only 6k although the bitlength of the arbiter PUF is 64. When the XOR gates increase, the training time increase exponentially too. However, LR is not the perfect method that could be applied to every kind of PUFs. When the authors is trying to break through the Feed Forward PUFs(FF PUF), ES shows better result than LR especially when the number of FF-loops is over 3, i.e LR could only break FF PUF with FF-loops less than 3. These authors could compromise a 6 FF-loops PUF in 8 minutes with an accuracy of 97.72%. The number of CRPs needed is 50k. But when the FF-loop increase, the training time of ES does not change a lot especially when the FF-loops is over 7. As a result, machine learning method shows great threats to strong PUFs but needs some time.

IV. Overview Through Deep Learning Methods

A. Problem statement

As previous research has shown that breaking through a large XOR arbiter PUF is time consuming. And for a super large XOR PUF the time is unacceptable. So these authors[3] come up with a new idea that using deep learning method to simulate the N XOR PUF and compromise it in accepted time. The biggest challenge of their work is building a suitable neural network and achieving extremely short training time.

B. Methodology

Their methodology is building a neural network based on the XOR gate size. That means the size of neural network is gong to change to adapt to different size of N XOR PUFs. The architecture of the neural network is shown in Fig 2.

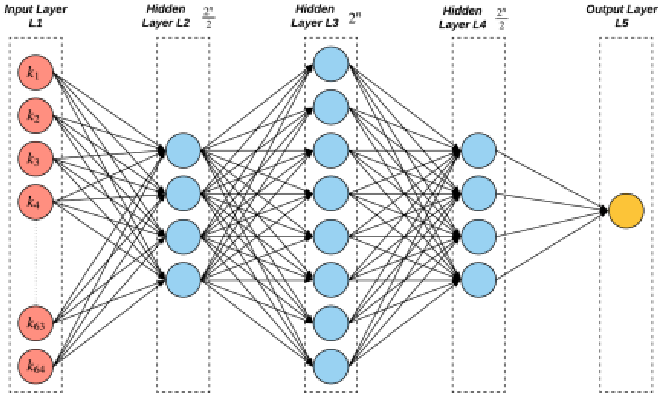


Fig. 2. Neural network architecture for different N XOR gates [3]

To better understand the neural network, let me use an example to explain it. If we are using a 5 XOR arbiter PUF, the value of n is 5 in the architecure. The input layer is 64 depending on the bitlength of arbiter PUF. So the number of neurons at hidden layer one and three is 2^{5-1} which is 16. The number of neuron at hidden layer two is 2^5 which is 32. Moreover, to improve the performance of deep learning, they utilize tangent activation function every hidden layer. And the output activation function is sigmoid which will give us a result between 0 and 1. The optimizer they are using is Adam optimizer. The loss function is BCE loss. By feeding the adaptive neural network, they could break most XOR arbiter PUF.

C. *Evaluation metrics and result*

They use the same evaluation matrix as what [1][2] did. This is convenient for readers to compare their models to each other. As a result, their deep learning methods shows fast learning speed than all the previous PUF modeling attack methods. Their best results is breaking through the 9 XOR arbiter PUF in 9.12 minutes with an accuracy of 98%. The traning set is 4.2M CRPs. While others[4] who tried to break through the 9 XOR PUF spending 25 days to achieve the same accuracy with 350M CRPs. One thing we can notice that this is a paper[3] published in 2020 which is 10 years later than [1]. The table below shows a comparison result for the 5 and 6 XOR PUF as [1] did not research on larger XOR arbiter PUFs.

TABLE I
COMPARISON BETWEEN LOGISTIC REGRESSION AND DEEP LEARNING METHODS ON DIFFERENT SIZE OF N XOR PUFS

Bit-len	XORs	Method	Tr.CRPs	Best Tst. Acc.	Tr.Time
64 Bit	5-XOR	[1]	80k	99%	2.08hrs
		[3]	42k	99%	0.17min
	6-XOR	[1]	200k	99%	31.01hrs
		[3]	255k	99%	2.04min

V. DISCUSSION AND CONCLUSION

a) Modeling attack: As what we have reviewed, modeling attack is becoming a major threats to PUFs. Traditional machine learning attack give us a sign that storng PUFs could be modeled by software. Latest deep learning attack shows us adding more XOR gates is no longer safe for XOR PUFs. The training time could be no longer than 10 minutes to break through a 9-XOR PUF and the structure could still be improved to get better performance. Hence, new security enhancement design for PUF should be developed in the future to face the modeling attack challenge.

b) Ways to defend: The reason for people to do research on PUF modeling attack is improving the security strength of it. As many people are trying to figure a way out against malicious purpose modeling attack. [5] proposed a way that by injecting some poison CRPs in the normal CRPs to obfucate the machine learning algorithms. These inverted CRPs will reduce the machine learning accuracy but has no influence to normal usage. On my perspective, people could try weak PUF strategies like reducing the number of CRPs or no accessing to the CRPs dataset. Anyway, making defense from the CRPs is a good strategy for security designers to further improve PUF's security. Also, adding some non-linear algorithms in the PUF design may also work against the modeling attack like how people did in controlled PUFs.

REFERENCES

[1] U. Rührmair and J. Sölter, ”PUF modeling attacks: An introduction and overview,” 2014 Design, Automation Test in Europe Conference Exhibition (DATE), 2014, pp. 1-6, doi: 10.7873/DATE.2014.361.

[2] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. 2010. Modeling attacks on physical unclonable functions. In Proceedings of the 17th ACM conference on Computer and communications security (CCS '10). Association for Computing Machinery, New York, NY, USA, 237–249. DOI:https://doi.org/10.1145/1866307.1866335

[3] Mursi, K.T.; Thapaliya, B.; Zhuang, Y.; Aseeri, A.O.; Alkatheiri, M.S. A Fast Deep Learning Method for Security Vulnerability Study of XOR PUFs. Electronics 2020, 9, 1715. https://doi.org/10.3390/electronics9101715

[4] Tobisch, J.; Becker, G.T. On the scaling of machine learning attacks on PUFs with application to noise bifurcation. In International Workshop on Radio Frequency Identification: Security and Privacy Issues; Springer: Berlin/Heidelberg, Germany, 2015; pp. 17–31.

[5] S. -J. Wang, Y. -S. Chen and K. S. -M. Li, ”Modeling Attack Resistant PUFs Based on Adversarial Attack against Machine Learning,” in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, doi: 10.1109/JETCAS.2021.3062413.