University of Sheffield

Reconfigurable Security for IoT Application in Handling Machine-Learning/Modelling Attacks

Cheng-Wei, Tsao

Supervisor: Dr Prosanta Gope

A report submitted in fulfilment of the requirements
for the degree of BSc in Computer Science

in the

Department of Computer Science

March 20, 2022

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Cheng-Wei, Tsao

Signature: Cheng-Wei, Tsao

Date: November 15, 2021

Abstract

This report investigates and provides a clear introduction on PUF(physical unclonable function), the aim of the project and current progress. Overall, there are thoroughly description on PUF, reconfigurability framework and concepts corresponding to specific machine learnings for modeling attack on PUF.

The aim of the project is to propose a novel, suitable machine learning to model PUF(physical unclonable function) and then design a reconfigurability framework to fight against such attack. The PUF structure is similar to a road network so machine learning related to ETA(estimated time arrival) problem is strongly considered.

The achievements to date are having a robust understanding of PUF, reconfigurability property, and an attempt to implement reinforcement-learning and graph attention neural network as modeling attack. SARSALambda Q learning has been tested on PUF but only around 60% accuracy has been achieved. Graph attention neural network is implemented as well, but will need further investigation to adapt to dynamic input.

Contents

1 Introduction **1**

1.0.1 Conventional protection of devices 1

1.0.2 Advantage of PUF compare to conventional protection on devices . . 1

1.1 Aims and Objectives of the project 2

1.2 Overview of the Report 2

2 Literature Survey **3**

2.1 The PUF concept 3

2.2 Weak and strong PUF 4

2.2.1 Weak PUF 4

2.2.2 Strong PUF 4

2.3 Application of PUF: Authentication 5

2.3.1 Authentication process: Enrollment stage 5

2.3.2 Authentication process: Authentication stage 6

2.4 Different type of PUF: Arbiter PUF and XOR arbiter PUF 6

2.4.1 Idea of arbiter PUF 6

2.4.2 Idea of XOR arbiter PUF 7

2.5 Potential threat on PUF: Modeling attack 8

2.6 Countermeasure for modeling attack: Reconfigurability 9

2.6.1 Basic concept of a type of reconfigurability framework: OPUF 9

2.6.2 DPUF: An example of OPUF 10

2.7 Summary 10

3 Analysis **12**

3.1 Project Aims and Evaluation 12

3.1.1 Modeling attack 12

3.1.2 Reconfigurability framework 13

3.1.2.1 Add intentional noise 13

3.1.2.2 OPUF 14

4 Progress **15**

4.1 Project progress 15

4.1.1 SARSALambda Q learning 15

4.1.1.1 Example of SARSALambda Q learning 15

4.1.1.2 Evaluation of SARSALambda Q learning 16

4.1.2 Graph attention neural network 17

4.1.2.1 Example of Graph attention neural network 17

4.1.2.2 Aggregate operation of Graph attention neural network . . . 18

4.1.2.3 Evaluation of Graph attention neural network 18

5 Conclusions 19

5.1 Summary of the project and project Plan 19

5.1.1 Short term plan 19

5.1.2 Long term plan 19

List of Figures

2.1 Different PUF that generate a different response when input same challenge . 3

2.2 Attacker can perform same behaviour as Weak PUF when having full access to CRPs and not under secure environment 4

2.3 The attacker eavesdropped CRP that has been used can not successfully validate in next evaluation for strong PUF 5

2.4 Enrollment stage in PUF authentication 6

2.5 Authentication stage in PUF authentication 6

2.6 Arbiter PUF structure 7

2.7 XOR arbiter PUF structure 8

2.8 Concept of forward unpredictability and backward unpredictability 9

2.9 Reconfigurability framework of DPUF 10

3.1 Add noise to PUF's response during authentication phase [4] 13

4.1 SARSALambda Q learning example environment 16

4.2 GAT aggregation on PUF structure 17

5.1 Gantt chart for project's plan 20

List of Tables

4.1 Q table for example environment 16

Chapter 1

Introduction

1.0.1 Conventional protection of devices

In the rapid development of the information era, many daily events are achieved by a variety of electronic devices such as computers or phones. Those electronic devices highly rely on integrated circuits(ICs) to perform specific events. For example, bank transactions can be done by different devices, the process contains personal data, and includes the usage of sensitive information. Therefore, information security like authentication, protecting confidential data has become important in nowadays society. In order to increase security's robustness, a range of ways has been proposed. One conventional way is by storing secret keys in non-volatile memory to encrypt sensitive data with it and using asymmetric cryptography to authenticate the device [9]. However, the implementation process of cryptography is expensive, especially on the resource-constraint device, and the device is still vulnerable to invasion attacks. Ideally, devices should be able to handle challenging problems corresponding to energy consumption, computational power and the ability to fight against cyber attacks.

1.0.2 Advantage of PUF compare to conventional protection on devices

PUF(physical unclonable function) has the ability to deal with the challenges mentioned at section 1.0.1. It does not store secrets in non-volatile memory, instead, the volatile secret is derived from devices' physical characteristics [9]. This is based on the inevitable random variation in the ICs manufacturing process, which leads to the fact that no two ICs have exact same physical characteristics [9]. For example, each ICs has unique delay sequences in the transistors and wires. With this property, PUF does not require lots of computational power and is cost-effective because no need to implement the cryptographic operation, which works particularly well on resources-constraint devices such as RFID. Also, the attacker needs to perform an attack when the device is on, which significantly increase the difficulty. As for an invasion attack, the attacker needs to have the exact information of its unique physical characteristics to successfully derive secrets [9]. Overall, PUF provides another interesting way of reinforcing security.

1.1 Aims and Objectives of the project

The objectives are split into two parts for this project. In the first stage, propose a novel machine learning to model different PUF(physical unclonable function) behaviour, so predict the response from a given PUF when given challenge bits. For example, considered the simplest PUF which is arbiter PUF, its operation to create a response is to input a challenge bits(binary), and two signals will go through the multiplexers in the PUF structure depending on the value of it. Consequently response a binary bit that will indicate which signal is faster. Therefore, the machine learning for modeling will be related to the ETA(estimated time arrival) problem since the structure of PUF is similar to a road network. For instance, travelling through each multiplexer is similar to travelling through each road segment, and both of them have delays to affect the time of arrival. Overall the first stage is to design a modeling attack considering these concepts. In the second stage, design a reconfigurability framework to fight against such modeling. In detail, evaluate the modeling attack by inserting noise in PUF or experiment on OPUF(one-time-PUF) which contain a reconfiguration process that can alleviate modeling attack.

1.2 Overview of the Report

The remaining of the paper will organize as follow. Chapter 2 provide a literature survey of the concept of PUF, including PUF's properties, detailed circuit structure and authentication process, existing modeling attack with experiment results, and reconfigurability framework. Chapter 3 describe the aim and objectives for the project, gives in-depth analyzes of how the project will be evaluated, the tests and experiments that support this. Chapter 4 demonstrate the current progress along with an explanation of the project. Chapter 5 provide brief summary of the main achievements with a well-organized future plan for the project.

Chapter 2

Literature Survey

2.1 The PUF concept

The simplest sentence to describe PUF is "A PUF is an object's fingerprint" [5]. The fingerprint can represent a specific human in the world, such as the PUF can represent an object. The fingerprint is inherently created when people were born, and so does PUF, which inherently exists in an object according to unique manufacturing random variation [5]. With the representation and inherent property, the PUF is said to be unclonable since it is impossible to control and predict a human's fingerprint. This is an important concept for PUF.

This intrinsic property can be extracted from a chip that has PUF circuit that existed inside [1]. The way PUF works is by entering a certain length of bits(so-called challenge) into the PUF, and it will generate another specific length of bits(so-called response). According to the property of PUF that was discussed in Chapter 1, it is impossible to find two different PUF that will produce the same response when entering the same challenge(See Figure 2.1).

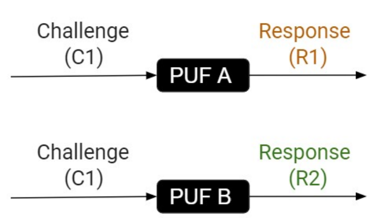

Figure 2.1: Different PUF that generate a different response when input same challenge

2.2 Weak and strong PUF

PUF can be classified into two categories, weak and strong PUF according to the strength of PUF. The strength of PUF indicates the number of challenge-response pairs(so-called CRPs) that can be generated from the PUF [6]. The higher numbers of CRPs can a PUF generate, the better strength it has. Generally, if increasing the size of the PUF leads to a linear increase in the number of CRPs, it is considered weak PUF. On the other hand, if increasing the size of the PUF leads to an exponential increase in the number of CRPs, it is considered strong PUF.

2.2.1 Weak PUF

For the weak PUF, it represents the PUF that has a smaller set of CRPs. While it is impossible to create a clone of PUF, but with a small set of CRPs, this will allow an attacker to record all the CRPs when the attacker has physical access to PUF [6]. With the knowledge of CRPs, attackers can easily provide the corresponding response to challenge as like they have a clone(See Figure 2.2). The weak PUF can be used for authentication and key storage. However, since weak PUF's CRPs can be fully accessed, ensuring having a secure environment and whether the original PUF is being evaluated is relatively important [6].

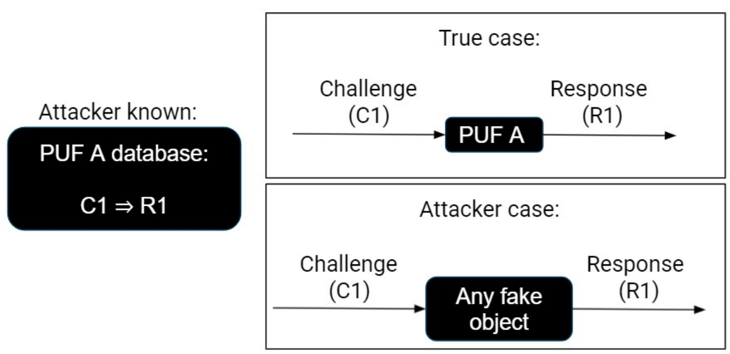

Figure 2.2: Attacker can perform same behaviour as Weak PUF when having full access to CRPs and not under secure environment

2.2.2 Strong PUF

For strong PUF, means the number of CRPs is significantly large that even attackers get access, having throughout knowledge of CRPs is impossible. While the number of CRPs is so large, and the CRP is randomly selected in usage, the probability that the attacker has knowledge about the CRP currently using is small [6]. In addition, each CRPs that is used once will be discarded(See Figure 2.3) so even if attackers recorded certain CRPs, also called eavesdropped, they will not be able to put them into use. The strong PUF can also be used for authentication but do not need to protect CRPs as serious as weak PUF.

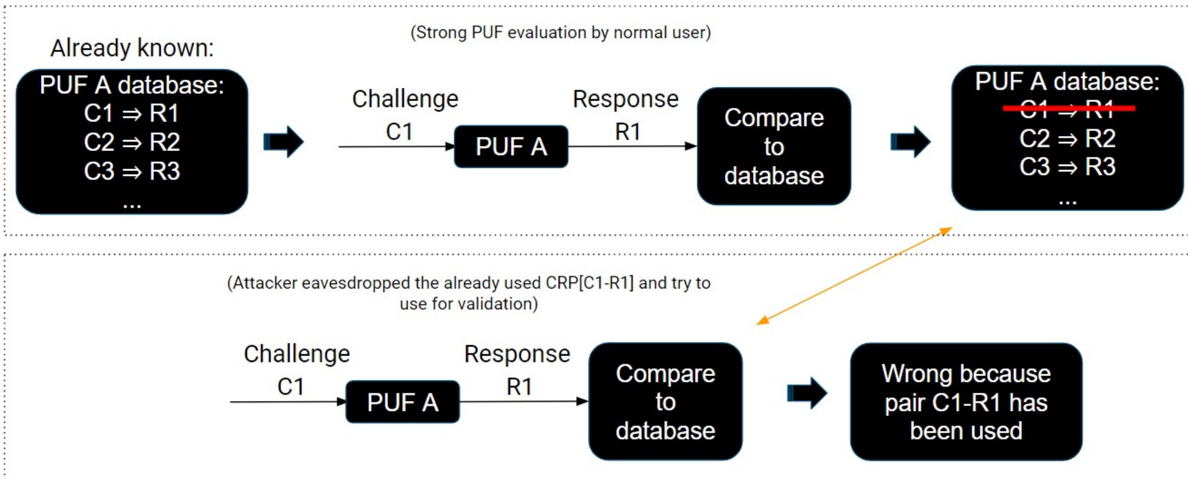

Figure 2.3: The attacker eavesdropped CRP that has been used can not successfully validate in next evaluation for strong PUF

2.3 Application of PUF: Authentication

One of the applications of PUF is authentication. As discussed in introduction of Chapter 1, PUF does not require huge computational power and are cost-effective, so it is suitable for many devices, especially the resources-constraint devices. The PUF's authentication included two stages, the enrollment and the authentication stage.

2.3.1 Authentication process: Enrollment stage

In the enrollment stage, the company possess the PUF, so the company can connect the server to PUF and send lots of challenges along with recording the CRPs into the database [1] (See Figure 2.4).

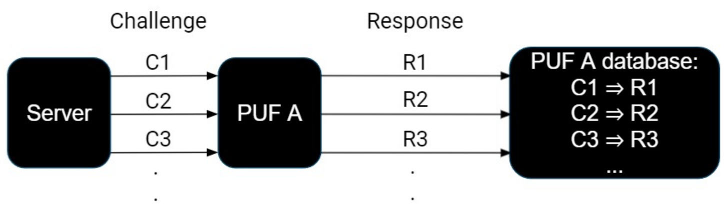

Figure 2.4: Enrollment stage in PUF authentication

2.3.2 Authentication process: Authentication stage

After recording all the CRPs, the company can now implement PUF on electronic devices. In the authentication stage, the server sent an arbitrary challenge to the devices that contain PUF while the device will return response. Afterwards, the server compares the response from the device with the database, if the challenge and response pair exists in the database, the device is valid [1] (See Figure 2.5). A life example will be a banking card.

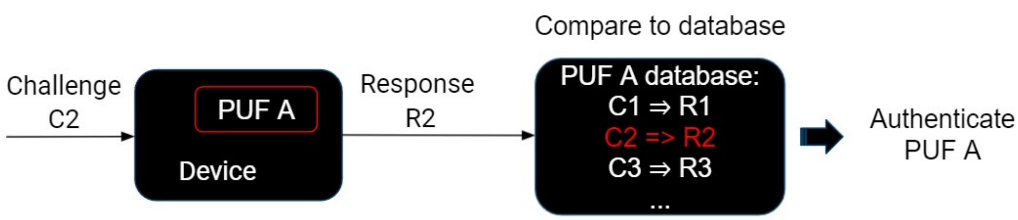

Figure 2.5: Authentication stage in PUF authentication

2.4 Different type of PUF: Arbiter PUF and XOR arbiter PUF

There are many different types of PUF such as arbiter PUF, ring oscillator PUF, lightweight PUF, etc. In this report, arbiter PUF and its mutation will be introduced in detail.

2.4.1 Idea of arbiter PUF

The general idea of the arbiter PUF is comparing the transition speed for two electrical signals in the PUF's structure(See Figure 2.6). The arbiter PUF's structure contains a number of multiplexers and an arbiter(mostly D flipflop), and two multiplexers will be combined into a switching box [9]. Look at Figure 2.6, when entering a challenge bit, apply each challenge bit to a switching box, bit 1 indicate the upper and lower signal will switch while bit 0 indicate the two signals remain unchanged in each switching box. This will eventually form paths for the signals(red and green line). Then the signals start transferring, the time arrived at the arbiter for two signals is different since each multiplexer and wire has a unique delay. The arbiter will determine which path is faster and based on that response a binary bit if the upper path is faster, the response is 1, otherwise, the response is 0.

The arbiter at the end is always fair, which will not favour any one of the paths. Even there exist bias, a simple solution of adding a delay in the structure can solve the problem. For example, if the arbiter favor the lower path, by adding a delay to the upper path, it can have a head start.

By looking at Figure 2.6, it is clear that the CRPs will be exponential. Assuming there are n switching boxes, two possible cases in each switching box, so the number of CRPs is 2^n, which indicate the arbiter PUF is a strong PUF. Arbiter PUF can also return a longer response by inputting K different challenges and getting a K bits response.

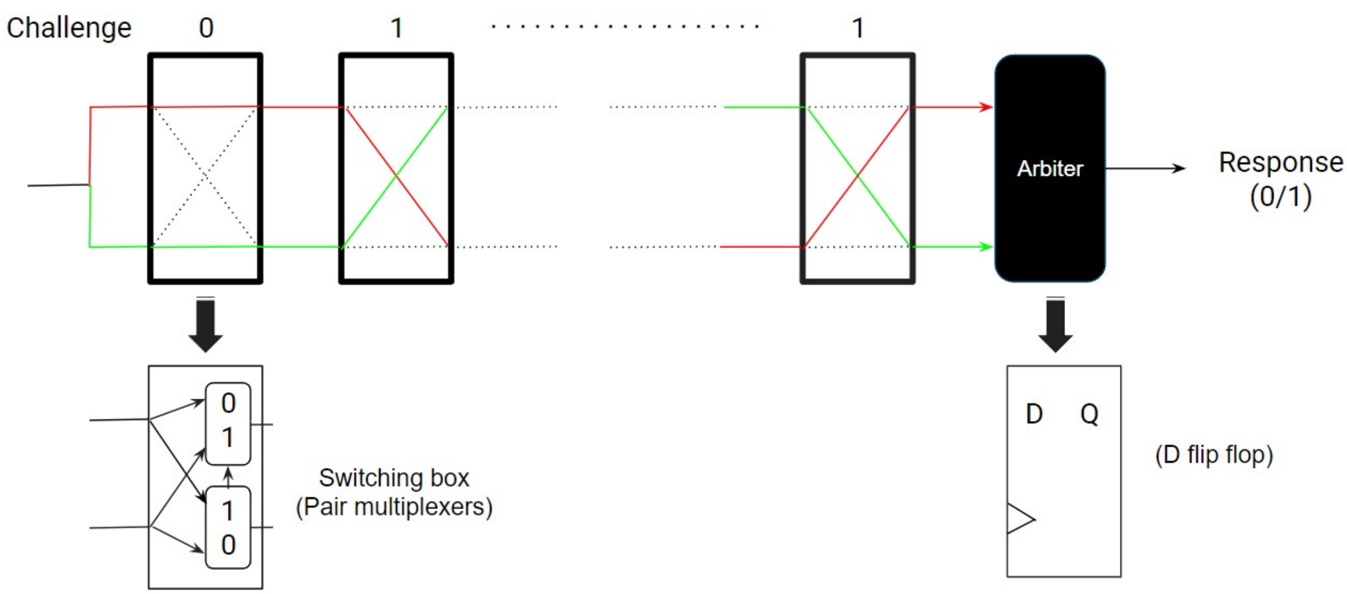

Figure 2.6: Arbiter PUF structure

2.4.2 Idea of XOR arbiter PUF

The normal arbiter PUF is vulnerable to modeling attack, the purpose of XOR arbiter PUF is to increase the robustness of arbiter PUF. The basic concept is to integrate multiple parallel arbiter PUF, given the same challenge to each arbiter PUF and XORed each response to produce the final response(See Figure 2.7) [8]. According to simulation, book [5] provides the

XOR arbiter PUF will have nonlinearity property that makes it harder to model.

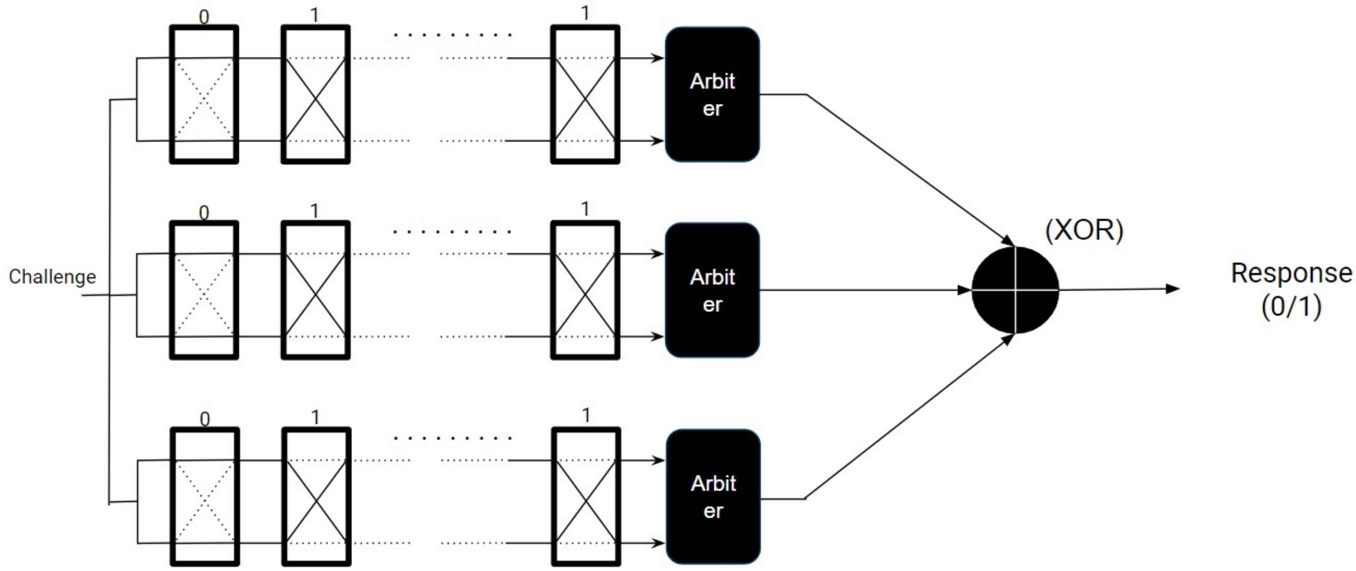

Figure 2.7: XOR arbiter PUF structure

2.5 Potential threat on PUF: Modeling attack

Many different threats can perform on devices, such as eavesdropped, gaining access to the memory that stores secret keys. For PUF, the main threat is that attackers can use techniques like machine learning to simulate the behaviour of CRPs(so-called modeling), which means even without the devices, attackers can still respond correctly when a challenge is provided. Take arbiter puf as an example, assume an arbiter PUF with i switching box, and the challenge that applies to each switching box is c[i]. The two signals travel through the path determined by challenge, and arrive at the arbiter at different time because of the delay in each component. The final response depends on the sign of final delay difference Δc:

$$r = \begin{cases} 1, & \text{if } \Delta c < 0. \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

which the delay difference is calculated by subtracting the upper path with the lower path's delay. The final delay difference Δc can represent as $w^T \Phi$, where w^T is a weight vector that represents delays for the components in PUF, and Φ is i bits challenges that was applied [8]. $w^T \Phi = 0$ will provide a hyperplane that separates the space for Φ, one side of the hyperplane are predicted as having response 1, the other side of the hyperplane are predicted as having response -1. In conclusion, the correct hyperplane indicates a good prediction of PUF. Machine learning such as logistic regression can play the role well. The modeling result for an arbiter PUF with 64 switching boxes, by using the logistic regression can get a good performance of 99.9% in a very short time with 18050 training CRPs [7].

2.6 Countermeasure for modeling attack: Reconfigurability

2.6.1 Basic concept of a type of reconfigurability framework: OPUF

In order to alleviate the problem that PUF is vulnerable to modeling attack, reconfiguration property embedded on PUF has been proposed. For example, the one-time-PUF(so called OPUF), the general idea is that its configuration alters after every authentication session, which means CRPs behaviour of the PUF is changed and invalidate the modeling attack. This is based on the forward unpredictability and backward unpredictability properties that OPUF possess [3]. Figure 2.8 shows the concept of forward unpredictability and backward unpredictability. The orange line represents the forward unpredictability, which will ensure the responses collected before the reconfiguration process is invalid afterwards. As for the blue line, which represent backward unpredictability, it will ensure the pattern observed by an attacker using modeling attack on PUF' is invalid for predicting the PUF before reconfiguration process [3]. In this case, assume an attacker is performing a modeling attack on the PUF, and required a certain amount of CRPs to gain proper prediction. However, the OPUF's structure keeps changing every execution, which means the CRPs behaviour is changing, so the modeling attack can't keep up or do not have time to collect enough CRPs [3].

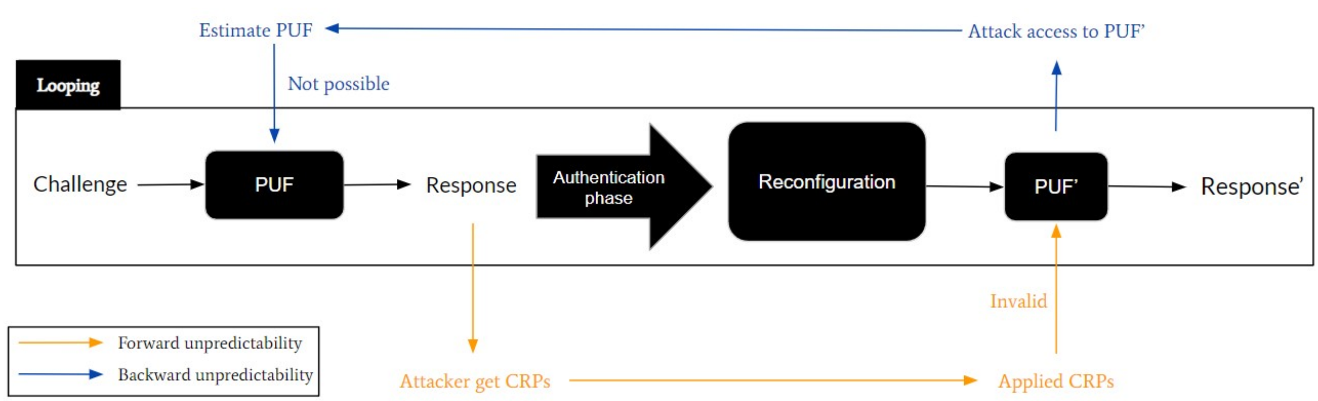

Figure 2.8: Concept of forward unpredictability and backward unpredictability

2.6.2 DPUF: An example of OPUF

DPUF can be considered as an example for creating the OPUF, it is built up with bit cells that contain capacitors and transistors, each cell stores information of values 0 and 1 [3]. However, these components will leak electrical signals every period of time, which means the behaviour might be eavesdropped. Therefore performing the reconfiguration process every period of time is effective. The reconfigurability can be shown in Figure 2.9, by varying parameters such as refresh-pause interval and the memory block, where the former can cause random bit flip in the cells while the latter store the final response in a random memory block which will alter every period of time, PUF can present unpredictable behaviour that prevents from attacker perform modeling attack [3].

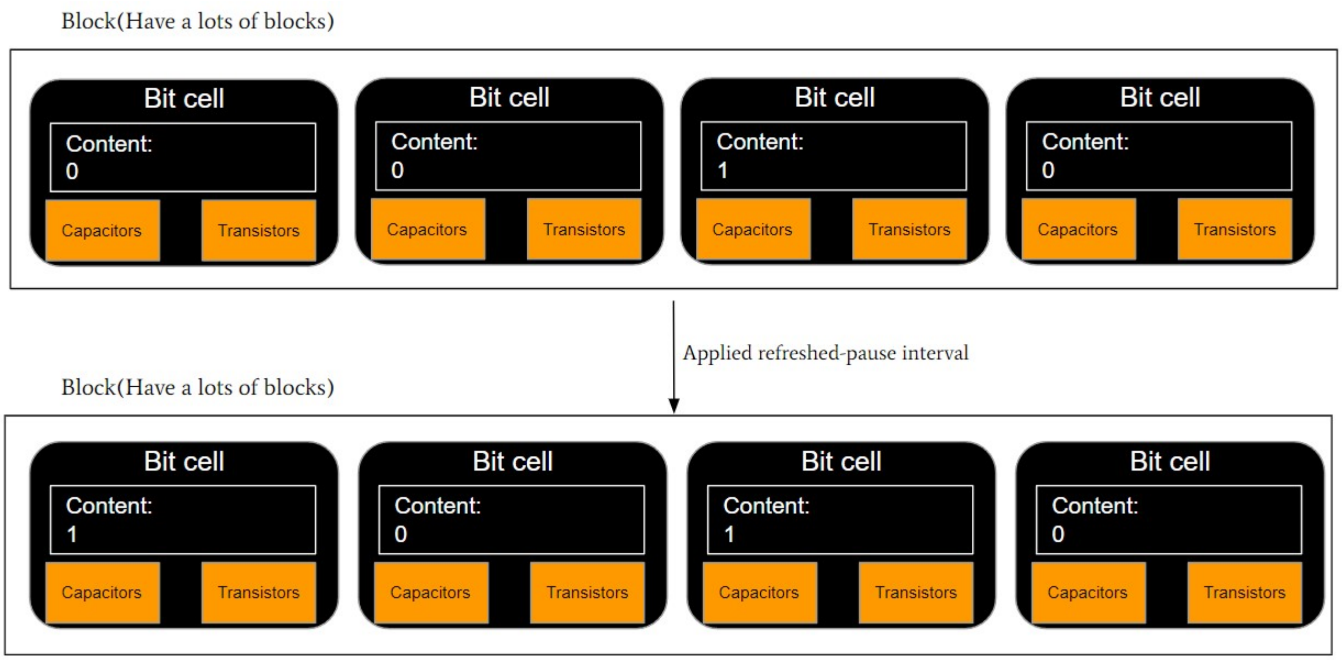

Figure 2.9: Reconfigurability framework of DPUF

2.7 Summary

In summary, chapter 2 describe the important background that need to understand for the project. This include showing PUF as a novel approach to increase the security robustness of electrical devices by making use of its unique physical characteristics and how it can be used for devices authentication. Also, the working process and concepts for arbiter PUF and XOR arbiter PUF is introduced in detail. Last, the largest potential threat on PUF is called the modeling attack and reconfigurability as the countermeasure are analyzed in detail. This chapter should provide basic concepts to understand project objectives.

Chapter 3

Analysis

Overall, a total of two objectives is required for the project. In the first part, propose a novel machine learning to modeling different PUF(physical unclonable function) behaviour, in particular arbiter PUF and XOR arbiter PUF. In the second part, consider reconfigurability like adding noise, implement protection related to concept of one-time-PUF to resilient against the modeling attack proposed in part one.

3.1 Project Aims and Evaluation

3.1.1 Modeling attack

The aim for part one is that the machine learning for modeling should be related to ETA(estimated time arrival) problem or the one people haven't used for modeling PUF. Moreover, reconfigurability needs to be considered when proposing the modeling attack while knowing the PUF can escape from it by changing CRPs behaviours in part two. Therefore, machine learning should ideally adapt to the changing behaviour of CRPs or the PUF. For example, unsupervised learning can be a good way since it can deal with unseen data. To evaluate the work, an overall accuracy of the modeling are expected to be around 90% and take short time like the logistic regression in paper [7]. In addition, the modeling attack should apply to a range of arbiter PUF and compare the experiment result.

3.1.2 Reconfigurability framework

3.1.2.1 Add intentional noise

The aim for part two is to propose a reconfigurability framework for the modeling attack in part one. Two different reconfigurability will be implemented. First, add intentional noise to PUF's response during the authentication phase to disturb the modeling attack [4]. The basic idea is shown in Figure 3.1, assuming the database has stored PUF's CRPs and provided a challenge c to PUF in the authentication phase. The PUF return a response r', and add noise e with r'. Then e and r' compose helper data h and send back to the verifier, where $h = r' \oplus e$. The verifier can reproduce $r' \oplus e$ with r and h if r and $r' \oplus e$ is similar. Last, the verifier send a arbitrary number x to user, both verifier and user create their own hashtag and compare to check if authenticate. In this case, because the attacker cannot accesses to r, they can only model based on c and $r' \oplus e$, the modeling might not be accurate, tha only thing attacker can do is try to build model that with high prediction accuracy based on $r' \oplus e$ [4]. To evaluate the work of adding intentional noise, the modeling attack proposed in part one should have a reasonable accuracy drop(3-4% according to [4]) and the authentication will still work though noise is added.

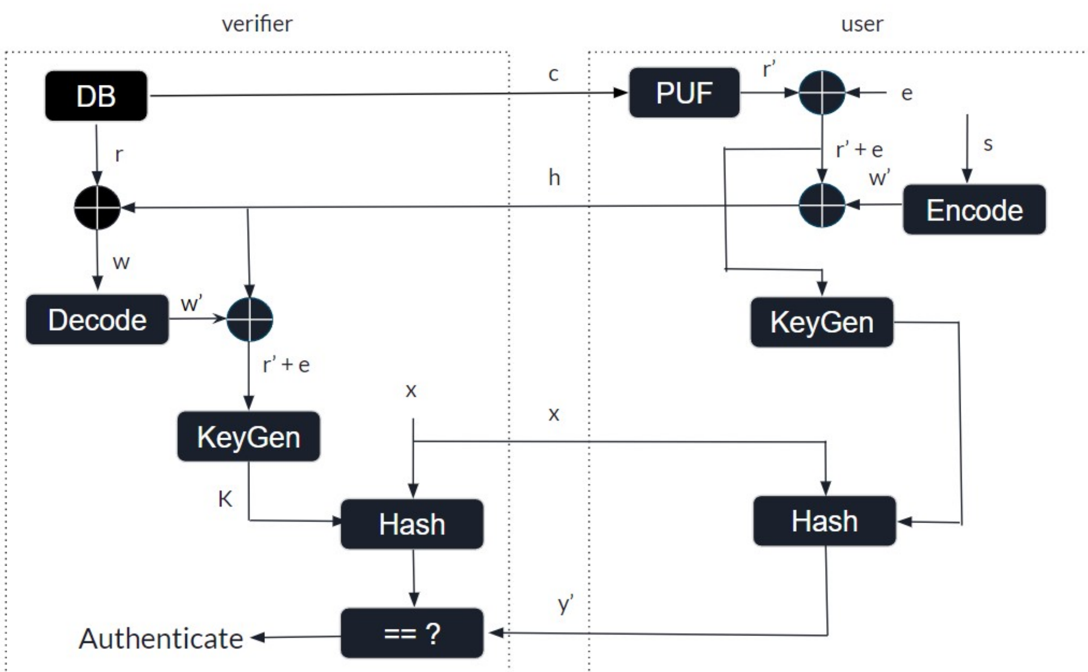

Figure 3.1: Add noise to PUF's response during authentication phase [4]

3.1.2.2 OPUF

Second, implement protection which refers to OPUF to change the behaviour of CRPs and prevent modeling attacks. With control of parameters like refreshed-pause timing and allocating memory block, the project will examine the result of the protection. For example, evaluate the accuracy and time consumed when applying the modeling attack in part one.

Chapter 4

Progress

4.1 Project progress

The project has two parts, currently, the progress is at the point of discovering and implementing machine learning for modeling the arbiter PUF. Methods such as reinforcement learning and graph attention neural network(so-called GAT) were used.

4.1.1 SARSALambda Q learning

The type of reinforcement learning that was implemented is the SARSALambda Q learning [11]. The general idea is that an agent will explore the environment with a final goal by randomly choosing actions space and recording the reward for each action for future evaluation.

4.1.1.1 Example of SARSALambda Q learning

Assuming Figure 4.1 is the PUF environment, a red and green circle is the starting point for top path and bottom path, each black rectangle represents a multiplexer with unique delay and the yellow circle is the goal. There are three actions: going up, going down and going straight. The reward of the multiplexer is determined by the delay, the bigger the delay, the smaller the reward, vice versa. In the training phase, the agent will travel through different combinations of multiplexers(different challenge) and construct a Q table(See Table 4.1) that stored the reward for the action of each node. The final goal for the agent is to find the route with the lowest delay. After the training, when selecting a CRPs, and inputting the challenge to the agent, ideally, the agent can find the lowest delay route, and use the constructed Q table to calculate, compare two paths' reward and reply the correct response. For example, assume a challenge 00001 has response 1(which means the bottom path is faster), the reward calculation is the following:

$$Challenge(00001) = \begin{cases} 0.082 + 0.008 + 0.041 + 0.002 + 0.07 = 0.203, & \text{Top path: 1,3,5,7,10.} \\ 0.022 + 0.415 + 0.222 + 0.555 + 0 = 1.214, & \text{Bottom path: 2,4,6,8,9.} \end{cases} \tag{4.1}$$

$$0.203 < 1.214, \text{ return response: } 1. \tag{4.2}$$

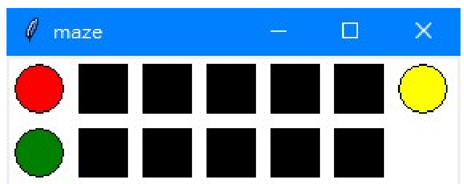

Figure 4.1: SARSALambda Q learning example environment

Multiplexer	go up	go down	go straight
1	0.000	0.028	0.082
2	0.094	0.000	0.022
3	0.000	0.357	0.008
4	0.009	0.000	0.415
5	0.000	0.181	0.041
6	0.042	0.000	0.222
7	0.000	0.641	0.002
8	0.003	0.000	0.555
9	0.000	0.070	0.000
10	0.000	0.000	0.131

Table 4.1: Q table for example environment

4.1.1.2 Evaluation of SARSALambda Q learning

In general, the route with lower delay will have a higher reward, on the other hand, the route with higher delay will have a lower reward. If the agent can respond correct response in high probability, the arbiter PUF's CRPs pattern can say to be successfully modelled. However, the accuracy for this modeling is around 60% - 69%, which is not a satisfying result. The problem can be the following: not providing enough features, the exploration does not cover every possible route, etc.

4.1.2 Graph attention neural network

As for the GAT [2], the basic idea of the GAT is that each node aggregate the neighbours' features based on adjacency matrix, adjacency attention value then update each node's features. The updated features will insert into the neural network and perform the classifying task. In order to reach high accuracy, the main task for the GAT is to update the adjacency attention matrix to reasonable value.

4.1.2.1 Example of Graph attention neural network

For a detail example, look at Figure 4.2, which is a arbiter PUF structure. Assume each node in Figure 4.2 represent a multiplexer, and has node feature $f1, f2, f3, f4$. If input a challenge 00, a one way relation is defined(observe green line): $m3 \to m1$, $m4 \to m2$, including self-loop, and the adjacency matrix can be constructed(See 4.3).

$$\begin{array}{cc} \begin{array}{cccc} M1 & M2 & M3 & M4 \end{array} & \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} M1 \\ M2 \\ M3 \\ M4 \end{array} \end{array} \tag{4.3}$$

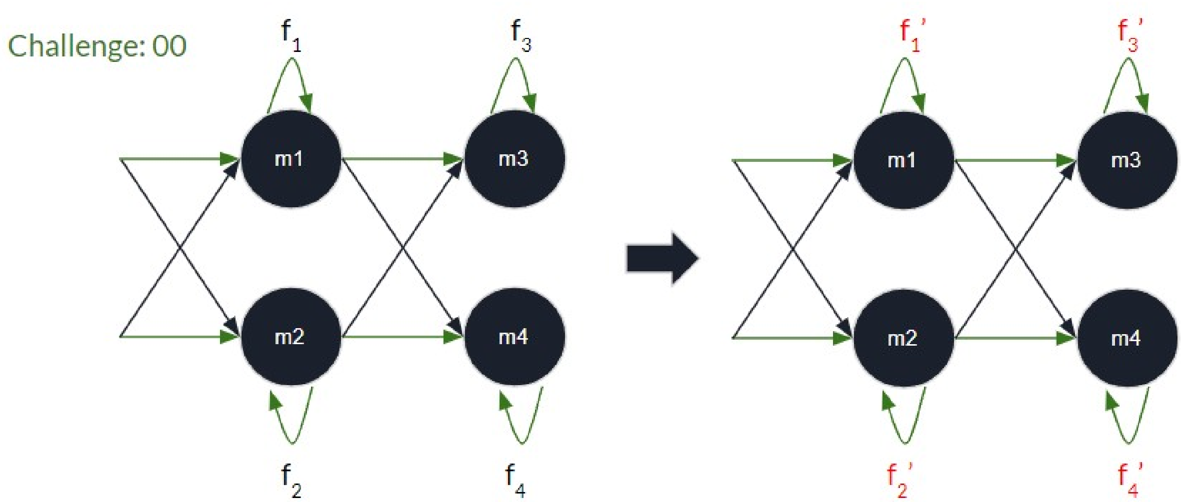

Figure 4.2: GAT aggregation on PUF structure

4.1.2.2 Aggregate operation of Graph attention neural network

In article [2], there are four steps to aggregate the features of each node:

The first step, add a weight matrix $\mathcal{W}$ to gain enough expressive power to transform the features into higher-level features [2]:

$$n_i = \mathcal{W} f_i \tag{4.4}$$

The second step, calculate a un-normalized attention value between every two nodes(i and j), but only consider nodes that is neighbor of i($j \in N_i$), where N_i is the neighbor of node i that can be observed in adjacency matrix(See 4.3). First, concatenates the features of the two nodes(symbol $\|$ represent concatenation), then perform a dot product with a trainable weight vector a, and applies a LeakyReLU at last [2]:

$$c_{ij} = LeakyReLU(\overrightarrow{a}^T(n_i \| n_j)) \tag{4.5}$$

The third step, apply Softmax function to normalize the attention value [2]:

$$\alpha_{ij} = \frac{exp(c_{ij})}{\sum_{k \in N_i} exp(c_{ik})} \tag{4.6}$$

The fourth step, aggregate the updated features from neighbors to current node, which consider the attention value [2]:

$$f_i^{'} = \sigma(\sum_{j \in N_i} \alpha_{ij} n_j) \tag{4.7}$$

4.1.2.3 Evaluation of Graph attention neural network

After updating every node's features, concatenates the node features of top path($f_1^{'} + f_3^{'}$)and bottom path($f_2^{'} + f_4^{'}$), and insert into a classify layer. Next, compare the output with ground fact (response of the challenge) and update the trainable parameter to increase the modeling accuracy. Ideally, the adjacency matrix will keep updating according to different input challenges, and the GAT will be able to learn the CRPs pattern after looking at many CRPs. However, there is doubt that whether the GAT support dynamically changing adjacency matrix, so this idea is not completely implemented yet.

Chapter 5

Conclusions

5.1 Summary of the project and project Plan

Currently, the project is still at the stage of designing modeling attack, which is the first part of objectives mentioned in Chapter 1. As stated in Chapter 4, reinforcement learning and GAT still suffer from implementation problems. The overall project plan is display in Figure 5.1.

5.1.1 Short term plan

For the short term goal, in the next few weeks till 12/22's meeting, the main task is to successfully implement modeling attack mentioned in Chapter 4 or other methods that can achieve prediction accuracy higher than 80%. The XGBoost was a new idea to try if both the previous machine learning fail to implement or model the arbiter PUF. However, more research on XGBoost was required. In addition, investigate more about pypuf, and try fully understand the paper [10], then prepare a presentation for 12/22's meeting.

5.1.2 Long term plan

In the long term, after the modeling attack is implemented, start on designing a reconfigurability framework to prevent the attack. The framework that was mentioned in Chapter 2 and Chapter 3 will be considered. The framework is expected to be completely implemented at the end of March 2022 so there will have enough time for testing and writing dissertation.

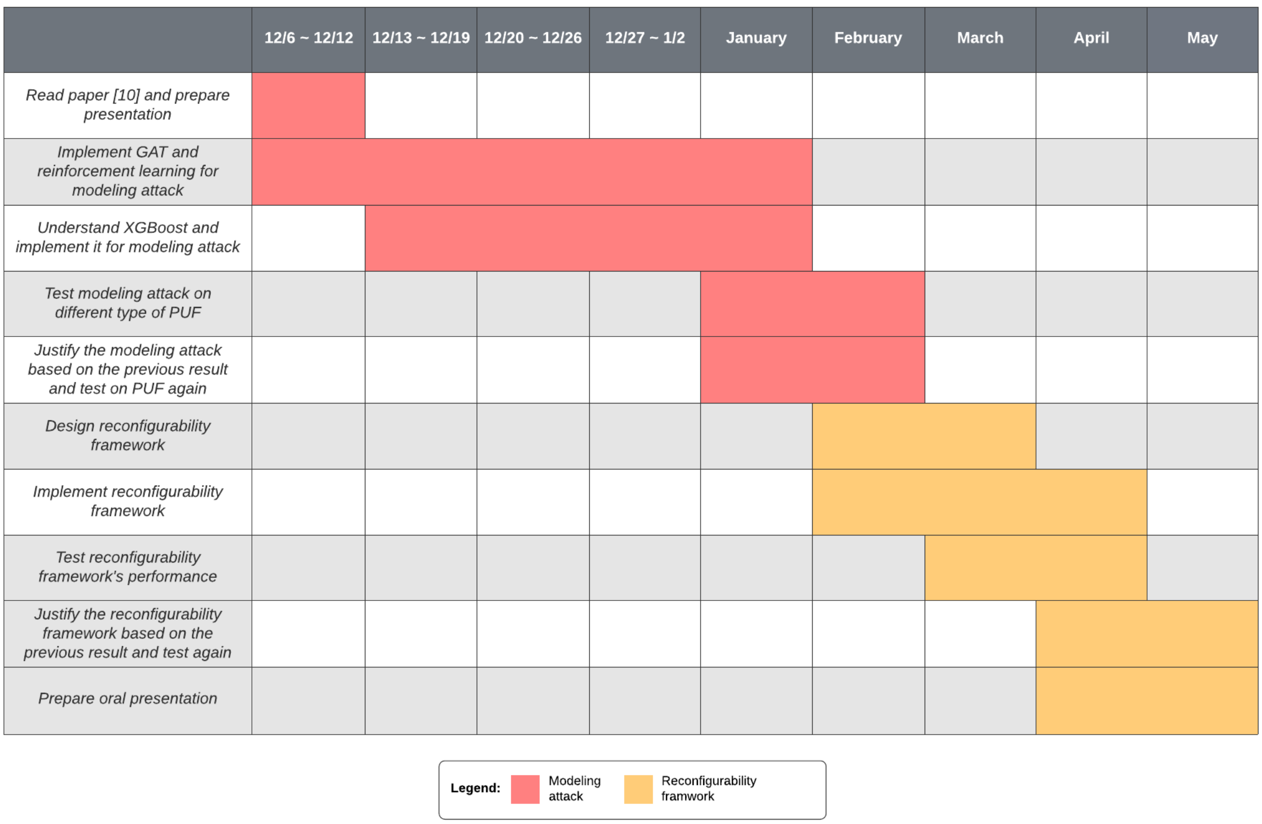

Figure 5.1: Gantt chart for project's plan

Bibliography

[1] Babaei, A., and Schiele, G. Physical unclonable functions in the internet of things: State of the art and open challenges. *Sensors*, 14 (2019).

[2] Dagar, A. Understanding graph attention networks (gat).

[3] Gope, P., Sikdar, B., and Millwood, O. A scalable protocol level approach to prevent machine learning attacks on puf-based authentication mechanisms for internet-of-medical-things. *IEEE Transactions on Industrial Informatics* (2021), 1–1.

[4] Lanet, J.-L., and Toma, C. *Innovative Security Solutions for Information Technology and Communications.* Springer, Cham, 2018.

[5] Maes, R. *Physically Unclonable Functions.* Springer, Berlin, Heidelberg, 2013.

[6] McGrath, T., Bagci, I. E., Wang, Z. M., Roedig, U., and Young, R. J. A puf taxonomy. *Applied Physics Reviews 6*, 12 (February 2019).

[7] Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., and Schmidhuber, J. Modeling attacks on physical unclonable functions. 237–249.

[8] Santikellur, P., Bhattacharyay, A., and Chakraborty, R. S. Deep learning based model building attacks on arbiter puf compositions. *IACR Cryptol. ePrint Arch. 2019* (2019), 566.

[9] Suh, G. E., and Devadas, S. Physical unclonable functions for device authentication and secret key generation. 9–14.

[10] Yao, W., Zhengtai, C., Shanshan, S., Binwei, S., Zhi, Q., Wenbing, F., and Juin, L. A lightweight authentication protocol against modeling attacks based on a novel lfsr-apuf and a private cover function. *IEEE Internet of Things Journal* (2021).

[11] Zhou, M. Reinforcement learning methods and tutorials., 2018.