Dynamic Adaptive Mutation Based Genetic Programming for Ring Oscillator PUF

Taichi Umeda, Yusuke Nozaki and Masaya Yoshikawa
Dept. of Information Engineering
Meijo University
Nagoya, Japan
e-mail: 183426003@ccalumni.meijo-u.ac.jp

***Abstract*—Recently, the damage caused by semiconductor counterfeit has become a serious problem in our lives. Physical Unclonable Function (PUF) has attracted attention as a countermeasure method. In the countermeasures, a ring oscillator (RO) PUF is one of the most popular PUFs. Regarding tamper resistance of RO PUFs, Genetic Programming (GP) based attacks have been proposed. However, the GP based attacks only apply the basic genetic strategy. To evaluate tamper resistance of RO PUF accurately, improvement of GP based attack for RO PUF is important. Therefore, this study proposes an accurate attack which is based on GP using dynamic adaptive mutation.**

Keywords-physical unclonable function; genetic programming; hardware security

I. Introduction

With the advancement of reverse engineering techniques, semiconductor counterfeiting has become a serious problem in recent years. This counterfeiting issue results in not only financial damage to companies, such as decreases in the brand image, but also human life hazards. Moreover, the Semiconductor Industry Association (SIA) estimates that counterfeit parts cost U.S. semiconductor companies more than $7.5 billion per year in lost revenue [1].

Therefore, Physical Unclonable Function (PUF) as forgery prevention technology has attracted attention. PUF is a system that gives input (challenge) and outputs (response) a value. The response is generated by utilizing uncontrollable random physical features at the manufacturing electronic parts. Therefore, reproducing PUF is very difficult. Moreover, several PUFs have been proposed [2]-[4]. Among them, Ring Oscillator (RO) PUF [2] is known as one of the most popular PUFs.

On the other hand, modeling attacks for PUF have been reported [5]-[7]. Among them, Genetic Programming (GP) based modeling attack for RO PUF has been reported [5][6]. In previous studies [5][6], a model is generated by only training Challenge and Response Pairs (CRPs). By using generated model, the unknown response is predicted from a new challenge.

To evaluate tamper resistance of RO PUF, improving efficiency of GP based attack is very important. Therefore, this study proposes a new GP based attack for RO PUF. To improve the attack actually, the proposed applies dynamic adaptive mutation.

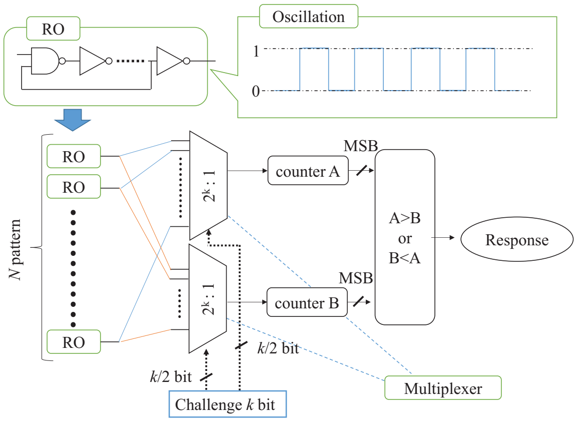

Figure 1. Outline of the RO PUF.

II. Preliminaries

A. Ring Oscillator PUF

Fig.1 shows outline of RO PUF. As shown in this figure, RO consists of a NAND and several NOTs. RO has a little difference of oscillation frequency by device characteristics. In the RO PUF, this difference is used as unique ID.

First, any number of ROs is set up. Then, outputs of two ROs are selected by the challenge k bit. At this time, oscillations of selected two ROs are input to counter A and B. Next, MSBs of counters are sent to arbiter circuit. Here, counting speed is affected by a difference of oscillation frequency. Therefore, if counter A is faster than counter B, output response is 1. If that is later than one, output response is 0. The unique ID of a device is generated using these responses.

B. Genetic Programming

Genetic programming (GP) is a population based optimization method. It is inspired by biological evolution mechanism. In GP, each individual represents a solution, and superior individuals have a better chance of survival. Fig.2 shows the outline of GP. In GP, each individual is expressed

978-0-7695-6528-6/18/$31.00 ©2018 IEEE
DOI 10.1109/ICCIA.2018.00047

as a tree structure. New individuals are generated by crossover processing and mutation processing.

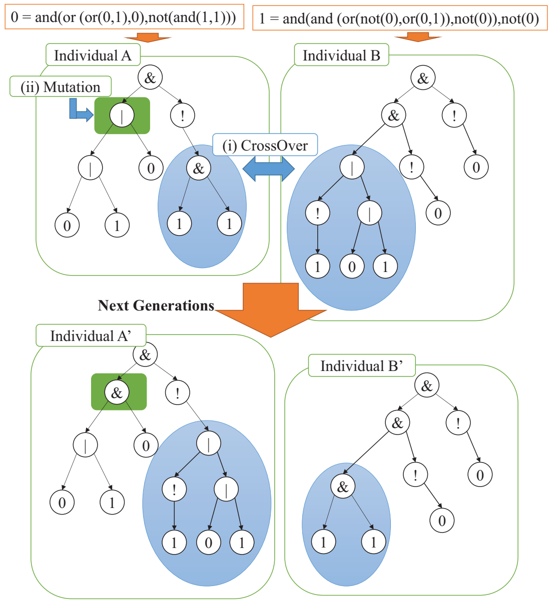

Figure 2. Example of individual, crossover and mutation.

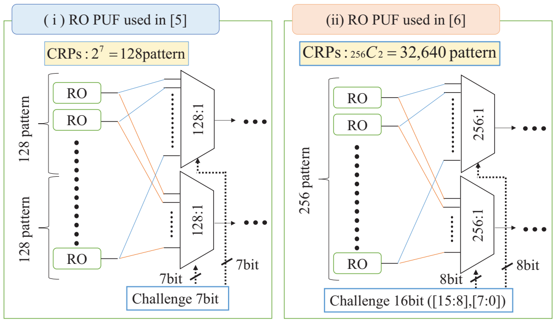

Figure 3. Outline of ROPUFs used in previous studies [6], [7].

As shown in Fig.2 (i), crossover processing replaces tree structures. Then, mutation processing changes an operator, as shown in Fig.2 (ii).

Next, the selecting method for individuals is explained. There are several selecting methods such as elitist model, tournament selection model, and roulette selection model. The elitist model selects one of the best individual in one generation. The stored best individual is took over to the next generation. Therefore, an effective solution is stably selected for every generation because of these procedures.

C. Related workS

In previous studies, the methods of GP based attack for RO PUF have been proposed [5], [6]. Fig.3 shows outline of RO PUFs used in papers [5], [6]. Here, the configurations are same as Fig. 1 after the multiplexers. In paper [5], modeling attack for RO PUF using 7-bit challenge shown in Fig. 3 (i). Since challenge size is 7 bit, CRPs are 128 (2^7=128) which is very small. Therefore, paper [6] proposed a method of GP based modeling attack for the general RO PUFs shown in Fig. 3 (ii). In paper [6], GP based attack using static mutation rate has been performed.

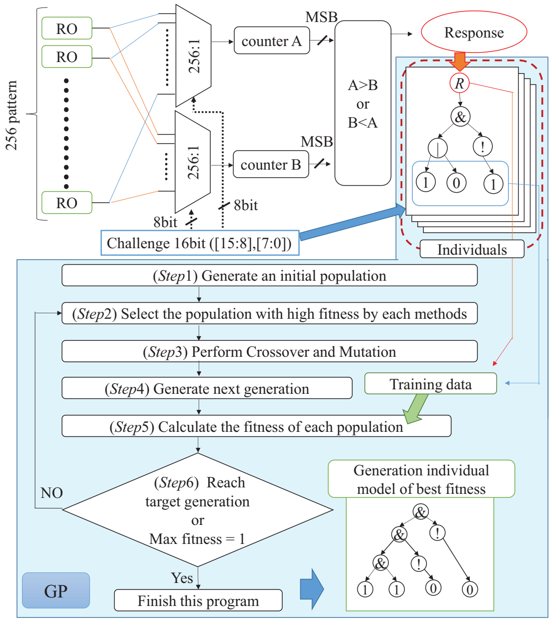
256 pattern
RO
RO
RO
256:1
256:1
counter A
counter B
MSB
MSB
A>B
or
B<A
8bit
8bit
Challenge 16bit ([15:8],[7:0])
Response
R
Individuals
(Step1) Generate an initial population
(Step2) Select the population with high fitness by each methods
(Step3) Perform Crossover and Mutation
(Step4) Generate next generation
Training data
(Step5) Calculate the fitness of each population
NO
(Step6) Reach
target generation
or
Max fitness = 1
Yes
Generation individual
model of best fitness
GP
Finish this program

Figure 4. Flowchart of GP based attack for ROPUF

III. Proposed Method

A. Based Algorithm

First, the based algorithm [5][6] of the proposed method is explained. This study generates a model of RO PUF by GP for the general RO PUF. Fig.4 shows outline of GP based attack for RO PUF. As shown in Fig.4, the individuals are generated using CRPs of the general RO PUF as training data. The individuals consist of operator and challenge bit. The operator consists of AND, OR, NOT. Moreover, in this study, crossover replaces randomly selected operator, mutation inverts randomly selected operator.

Blue area in Fig.4 shows the flow of the based algorithm. As shown in Fig.4, the based algorithm performs the following 6-step procedure.

*Step*1: Generate an initial population.

*Step*2: Select the population with high fitness by each methods.

Step3: Perform Crossover and Mutation.

Step4: Generate next generation.

Step5: Calculate the fitness of each population. Here, the fitness is calculated by applying training data to Individuals.

Step6: Decide the next step by the result.

Finally, a RO PUF model is generated by the individual of the best fitness.

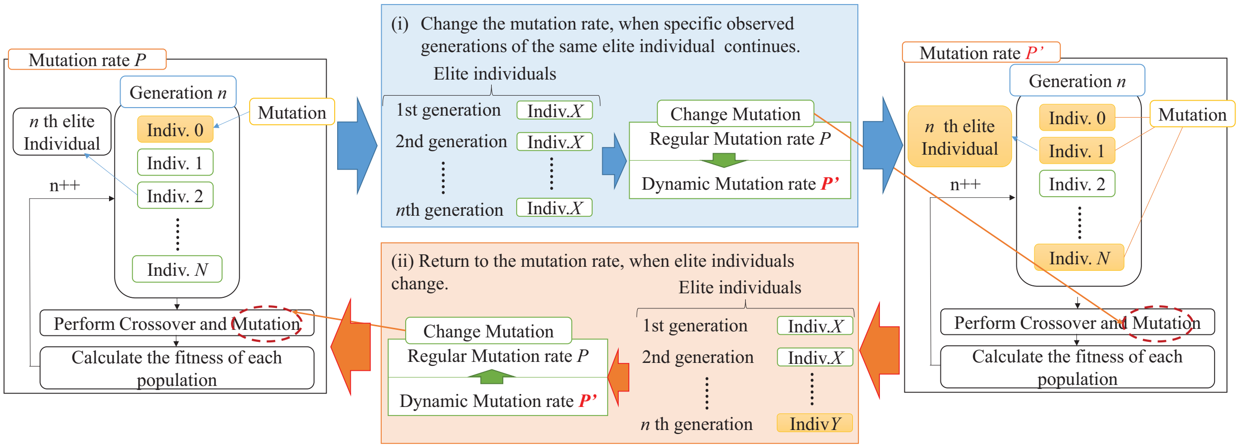
Mutation rate P
Generation n
n th elite Individual
Indiv. 0
Indiv. 1
Indiv. 2
Indiv. N
Mutation
n++
Perform Crossover and Mutation
Calculate the fitness of each population
(i) Change the mutation rate, when specific observed generations of the same elite individual continues.
Elite individuals
1st generation
2nd generation
nth generation
Indiv.X
Change Mutation
Regular Mutation rate P
Dynamic Mutation rate P'
(ii) Return to the mutation rate, when elite individuals change.
Elite individuals
1st generation
2nd generation
n th generation
Indiv.X
Indiv$Y$
Change Mutation
Regular Mutation rate P
Dynamic Mutation rate P'
Mutation rate P'
Generation n
n th elite Individual
Indiv. 0
Indiv. 1
Indiv. 2
Indiv. N
Mutation
n++
Perform Crossover and Mutation
Calculate the fitness of each population

Figure 5. Outline of the proposed dynamic adaptive mutation method

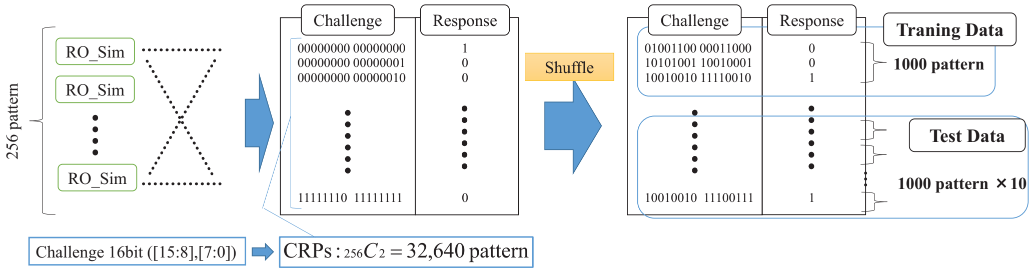

Figure 6. Generation method of training data and test data

B. *Applying Dynamic Mutation Rate for GP Based Attack*

Generally, in the field of GP, in order to improve the analytical accuracy, the method of changing the dynamic mutation rate have been reported [8]-[10]. These methods consider several parameters: the generation number, population number, and search situation. Therefore, to improve the attack efficiency, this study applies dynamic adaptive mutation to GP based attack for RO PUF.

Next, Fig.5 shows the outline of the proposed dynamic adaptive mutation method. Here, this study applies the elitist model. In the elitist model, the same model is generated occasionally for several generations in a row, this means that the solution converges. Therefore, the proposed method escapes from localized solution by changing the mutation rate to increase randomness (see Fig.5 (i)). Then, when a new elite individual is generated, the mutation rate is returned, as shown in Fig.5 (ii).

From the processing, the fitness improves than general static mutation because of the search range is expanded.

IV. Experiments

A. *Experimental Environment*

This study evaluated the proposed method in a simulator. Table.1 shows experimental environment. In the experiment, 1000 pieces of training data of CRPs were acquired. Moreover, 1000 pieces of test data of CRPs were acquired ten times. Fig.6 shows generation method of training data and test data. As shown in Fig.6 CRPs of 256 RO and 16 bit challenge RO PUF is $_{256}C_2 = 32640$ pattern. All patterns were generated and arranged in the list. Next, the list was shuffled to arrange without bias. Finally, 1000 pieces of training data were acquired in order from the top of the list of CRPs. Moreover, 1000 pieces of test data were acquired in order ten times from the bottom of the list of CRPs. From these procedures, training data and test data was elected without overlap.

In experiments, the average of response prediction rate was evaluated using best fitness model.

TABLE I. Experimental Condition

PC	Lenovo E440
CPU	Intel corei5
Memory	4GB
PUF	Ring Oscillator PUF
Ring Oscillator	256
Challenge bit (k)	16
Programing language	C++
Generation (n)	200
Population (N)	1000
Reproduction Scheme	Elite Model
Observed generation number	10
Crossover rate	0.8
Regular Mutation rate (P)	0.05
Dynamic Mutation rate (P')	0.4
Rand Seed	1-10

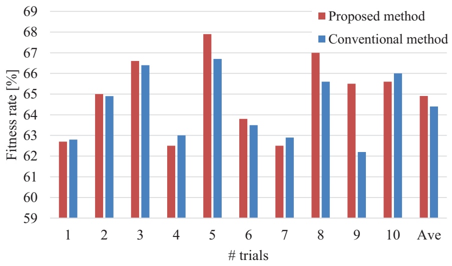

Figure 7. Fitness rate of RO PUF models against training data

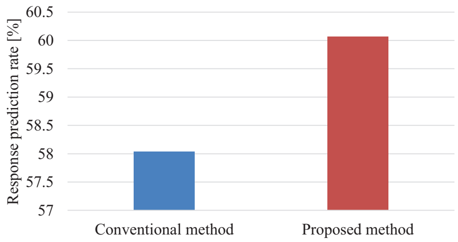

Figure 8. Experimental results

B. Experimental Result

Fig7 shows the modeling results. Fig.8 shows the attack results using best fitness model. In both figures, the red line and the blue line indicate the proposed method and the conventional method respectively. As shown in Fig.8, almost fitness data of the proposed method has higher fitness than the conventional method. Especially, in Fig.7, the best fitness can be found in the 5th trial.

Finally, Fig. 9 shows a generated model structure of the best fitness individual in the proposed method

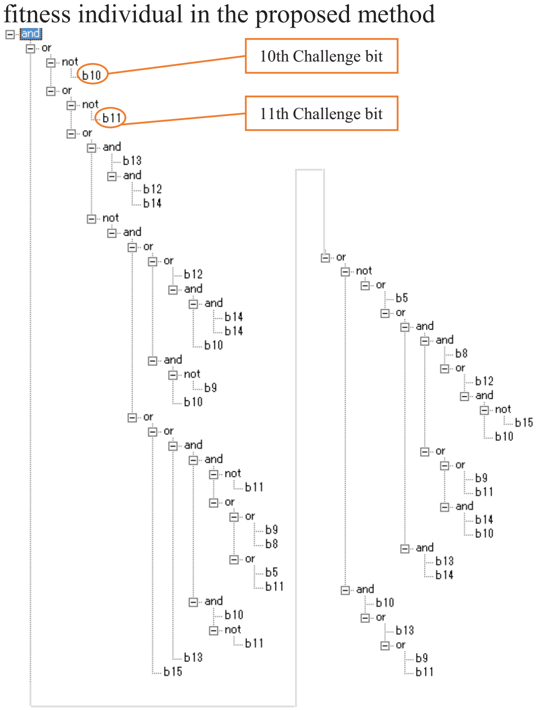

Figure 9. Model of best fitness individual.

V. Conclusion

This study proposed dynamic adaptive mutation based GP for RO PUF. The proposed method improved the prediction performance in convince of the conventional one. Future works include proposing of improving method of first-half challenge bit usage.

References

[1] Semiconductor Industry Association, ”Detecting and Removing Counterfeit Semiconductors in the U.S. Supply Chain, “https://www.semiconductors.org/clientuploads/directory/DocumentSIA/Anti%20Counterfeiting%20Task%20Force/ACTF%20Whitepaper%20Counterfeit%20One%20Pager%20Final.pdf”.

[2] G.E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” Proc. Design Automation Conference 2007 (DAC’07), pp. 9-14, 2007.

[3] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls, “The butterfly PUF protecting IP on every FPGA,” Proc. Hardware-Oriented Security and Trust 2008, pp.67-70, 2008.

[4] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls, “FPGA intrinsic PUFs and their use for IP protection,” Proc. Workshop on Cryptographic Hardware and Embedded Systems, pp. 63-80, 2007.

[5] I. Saha, R.R. Jeldi, and R.S. Chakraborty, “Model Building Attacks on Physically Unclonable Functions using Genetic Programming,” Proc. IEEE Int. Symposium on Hardware-Oriented Security and Trust (HOST 2013), pp.41-44, 2013.

[6] Y. Nozaki and M. Yoshikawa, “Genetic Programming Based Attack for RO PUF,”Proc. the 2018 RISP Int. Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP’18), pp.447-450, 2018.

[7] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, “Modeling attacks on physical unclonable functions,” Proc. 17th ACM Conf. Comput. Commun. Security, pp. 237-249, 2010.

[8] S. Marsili. Libelli, and P.Alba, “Adaptive Mutation in Genetic Algorithms,” Soft Computing, vol. 4, no. 2, pp. 76-80, Springer-Verlag, 2000.

[9] D. Thierens, “Adaptive mutation rate control schemes in genetic algorithms,” Proc. IEEE Int. Conf. Evolutionary Computation, pp. 980-985, 2002.

[10] M. Srinivas and L. Patnaik, “Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms,” IEEE Trans. systems, Man and Cybernetics, vol. 24, no. 4, pp. 656-667, 1994.