Hardware Demonstration of Mitiagating Power Analysis Attack Using Chaos Gates

Md Badruddoja Majumder, Md Sakib Hasan, Mesbah Uddin, and Garrett S. Rose
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, Tennessee 37996 USA
Email: {mmajumde, mhasan4 muddin6, garose}@utk.edu

***Abstract*—Chaos computing is a potential computing paradigm that is worth exploring as a means for mitigating security vulnerabilities in computation. Side channel attack is one of the most common vulnerabilities in hardware security. Power consumption, electromagnetic emanation, timing information are some of the common forms of side channel that adversaries exploit for gaining confidential information from a computing device. In this paper, we propose a hardware demonstration of chaos gate based mitigation technique for reverse engineering the opcodes of some instructions running on an Arithmetic Logic Unit (ALU). The mitigation technique involves using unique configurations of chaos gates for building different instances of ALU and thus preventing the attacker to gain information from one device by profiling another.**

***Index Terms*—chaos computing, arithmetic logic unit, side channel, power analysis attack, reverse engineering, instruction classification, FPGA**

I. Research Description

In traditional computing system, instances of a particular machine are all alike in terms of their physical implementations. This opens up the possibility of using characteristics of a particular computing machine to gain secret informations from another machine. In many cases information from a computing device leaks through side channel characteristics such as power consumption, electromagnetic emanation, timing to name the most common forms. As an example, power consumption or electromagnetic emanation can be leveraged to reverse engineer the opcode of each instruction running from a particular program. [1], [2]. Reverse engineering the opcodes is the major part of reverse engineering a software. In such reverse engineering attacks, an adversary uses a computing machine that she has access to and makes profile of each instruction based on its side channel leakage characteristics. Accessibility implies the ability to run any programs on the machine and observe its side channel characteristics. Collected instruction profiles are used to train a classifier and attack another machine to reverse engineer the instructions. We propose to mitigate this problem using chaos gates for building computing systems. A chaos gate is built using chaotic functions such as logistic map, tent map, chua's oscillator [3], [4]. A chaos gate can be configured to perform a large number of logic operation and the same logic operation can be performed using lots of different configurations. A computing system can be implemented using chaos gates where each instance uses a particular configuration for realizing its basic building blocks. The most important part of such implementation is that all these different implementations have different side channel leakage characteristics. An adversary therefore cannot use the instruction profiles collected from an accessible machine to attack another machine where she can only observe side channel characteristics.

II. Research Focus for the demo

This hardware demonstration targets at showing the prospect of using chaos gates in mitigating security vulnerabilities of computation. Specifically, we focus here on building different instances of an ALU using chaos gates where configurations of chaos gates varies from instance to instance while providing similar functionality with unique side channel leakage characteristics (power traces). In power analysis assisted instruction reverse engineering attack, usually an adversary uses a machine to profile each instruction and apply these profiles to reverse engineer the programs running on another machine. We propose here that, in chaos based implementation this does not work anymore since each machine is unique in terms of its side channel power characteristics. The main goal of the demo is to show that instruction profiling from one chaotic ALU can classify the instructions running on another ALU only with a very low accuracy as compared to the traditional case. In traditional scenario, each machines are alike and power profiles collected from one machine can be successfully be used to classify instructions running on another machine with higher accuracy.

III. Hardware setup

To demonstrate the idea of mitigating side channel power assisted instruction reverse engineering attack using chaos gates, we build a hardware set up comprising a FPGA board, oscilloscope and a laptop. In the FPGA we build a simple 16 bit arithmetic logic unit (ALU) that can perform certain logical (AND, OR, XOR), arithmetic (ADD, SUB) and shift/rotate (SHF, ROT) instructions. Logic gates used in the ALU is realized using a well-known chaotic function, logistic map shown in Eq. 1 .

$$x_{n+1} = rx_n(1 - x_n) \tag{1}$$

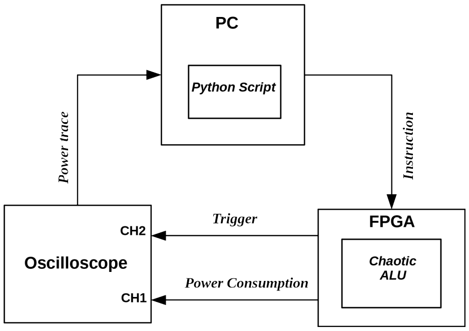


Fig. 1: Block diagram of Hardware setup for demonstrating power analysis attack mitigation on an ALU using chaos gates

Logistic map exhibits chaotic behavior when the parameter r lies between 3.5 and 4. Power consumption corresponding to each instruction is recorded using an oscilloscope which can be analyzed later for instruction profiling and classification. A python script facilitates all communication among PC, FPGA and oscilloscope. Fig. 1 shows the block diagram of the whole setup. The real set up that was used in preparing for the demo is shown in Fig. 2. For profiling each instruction based on power traces, random operands are sent from PC to the FPGA. A trigger signal is also sent which tells the FPGA and oscilloscope to start the operation and record the power trace, respectively. PC collects the power trace recorded by the oscilloscope and save it in csv files. A large number of power traces for each instructions will be used for training the classifier. For demonstrating the attack in traditional scenario, we classify random instructions using the profiling information collected from the same machine and measure the classification accuracy. This is analogous to the traditional case where each machine are similar in terms of their side channel behavior. On the other hand, to demonstrate the mitigation technique, we classify random instructions running on an ALU using the profiling information collected from another ALU where the chaos gates used in two ALUs use different configurations.

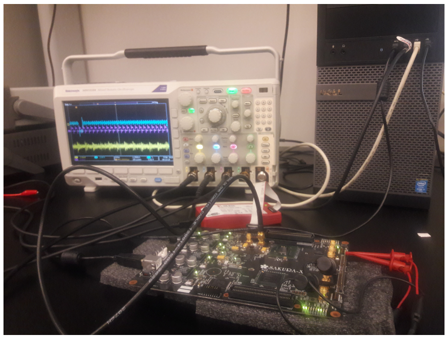
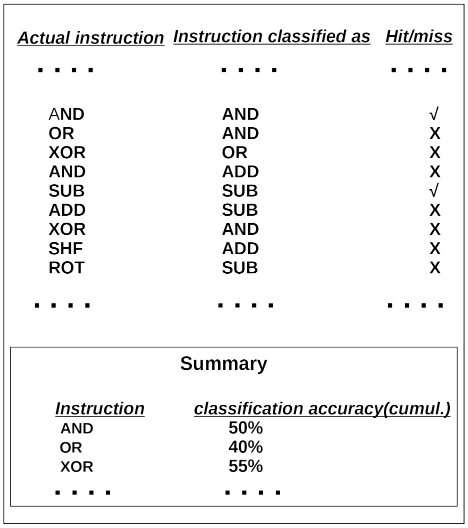


Fig. 2: Real hardware setup used for the demonstration.

IV. Observables

A graphical interface will dynamically show the classification accuracy of each ALU instruction running on the FPGA.



Actual instruction	*Instruction classified as*	*Hit/miss*
....	....	....
AND	AND	√
OR	AND	X
XOR	OR	X
AND	ADD	X
SUB	SUB	√
ADD	SUB	X
XOR	AND	X
SHF	ADD	X
ROT	SUB	X
....	....	....

Summary

Instruction	*classification accuracy(cumul.)*
AND	50%
OR	40%
XOR	55%
....	....

Fig. 3: Sample result display showing the instruction classification accuracy in runtime and also the cumulative accuracy.

The graphical interface will tell what instruction is actually running and what it is classified as. Based on the classification result, a hit or 'miss' will be displayed. A table will also be displayed that will show the cumulative classification accuracy of each instructions which is the percentage of hit to total occurrences of an instruction. Test programs consisting of supported ALU instructions will be executed in a random order. Training data bases will be made beforehand as it involves a large amount of data collection which will not be possible in the demonstration. However, the data collection process will be shown and explained at the beginning of the main demonstration. A sample result display for the proposed demonstration is shown in Fig. 3.

V. Conclusion

Chaos computing can be a potential solution to existing side channel vulnerabilities in computation. Proposed hardware demonstration shows the realization of an ALU with chaos gates and its effect on mitigating side channel based reverse engineering attacks. Such demonstration will inspire the exploration of chaos gates in mitigating more hardware security research.

References

[1] D. Vermoen, M. Witteman, and G. Gaydadjiev, "Reverse engineering java card applets using power analysis," *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems*, pp. 138–149, 2007.

[2] M. Msgna, K. Markantonakis, and K. Mayes, "Precise instruction-level side channel profiling of embedded processors," in *International Conference on Information Security Practice and Experience*. Springer, 2014, pp. 129–143.

[3] W. L. Ditto, K. Murali, and S. Sinha, "Chaos computing: ideas and implementations," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1865, pp. 653–664, 2008.

[4] L. O. Chua and G.-N. Lin, "Canonical realization of chua's circuit family," *IEEE transactions on Circuits and Systems*, vol. 37, no. 7, pp. 885–902, 1990.