Development of Diagnosis-based Hardware Trojan Tolerate System

Takuro Kasai
Hirosaki University
h21ms404@hirosaki-u.ac.jp

Masashi Imai
Hirosaki University
miyabi@hirosaki-u.ac.jp

Abstract – Hardware Trojan threats caused by adversaries have become one of serious issues. It has been recognized that it is significantly difficult to detect all the hardware Trojans in field. In this paper, a diagnosis-based hardware Trojan tolerate system with deep learning scheme is introduced. Several collection methods of dynamic information in order to judge whether a target behavior is normal or abnormal are explained and some evaluation results are shown.

I. Introduction

With the advance of VLSI fabrication technology, the number of transistors in a VLSI chip has been significantly increased. As a result, wide variety of multi-processor system-on-a-chip (MPSoCs) which integrate multiple processor cores, digital signal processing cores, memory cores, and various intellectual processor (IP) cores in a chip can be developed and commercially used. When an MPSoC is developed, designers may use the third party IPs and outsourcing developers in order to decrease turn-around-time to the market. In addition, fabless companies which do not have their own manufacturing facilities are increasing. Consequently, hardware Trojan threats caused by adversaries and untrusted foundries have become one of the serious issues [1,2].

Hardware Trojan means a malicious modification of the target integrated circuit. Hardware Trojan insertions can be arbitrarily adopted by adversaries at any design phase. Several detection and counter methods have been published and developed [2]. However, it has been recognized that it is significantly difficult to detect all the hardware Trojan threats since the relationship between the correct designers and the adversaries is similar to a cat-and-mouse game. We think that one of effective counter methods is a heuristic method using several hardware diagnosis information, software log information, and side channel information based on a deep learning scheme. In this paper, we propose a diagnosis-based hardware Trojan tolerate system based on the deep learning for certain configuration machines. By applying this method to computers with various machines, we will also explain the hardware Trojan tolerate system developed on machines with various configurations, and several collection methods of learning data we developed are also explained and several evaluation results are shown.

II. Hardware diagnosis information

Recently, several diagnostic information of a processor chip through a diagnosis tool provided by the processor vendor can be obtained [3]. Among the diagnostic information, static information makes no sense to detect an abnormal dynamic behavior. Thus, it is first needed to decide whether the diagnostic information is dynamic or static. The Intel Processor Diagnostic Tool (Intel PDT) is one of these diagnosis software which have the following functions and provide their results;

- Verifying processor core functions
- Confirming brand ID
- Checking the operating frequency
- Running load tests
- Testing processor functionality

In this research, the following two servers are used for evaluation with the Intel PDT in order to simply decide the dynamic information by comparing their results in which the obtained values due to their structure differences are omitted.

1. Processor: Intel® Xeon® W-2123, Memory: 32GB
2. Processor: Intel® Core™ i7-7700K, Memory: 16GB

As a result, it is recognized that the following items are different between the above two servers.

- Cache size
- Memory size
- Existence and type of modules and EISA (Extended instruction set architecture)
- Expected frequency
- Measured frequency
- Prime number generation test
- FLOPS test

Second, we use the WinSAT which is a standard evaluation tool as a changeable load in order to decide whether the above items are dynamic or static. Consequently, it is confirmed that the following three items are dynamically changed according to the applied load.

- Measured frequency
- Prime number generation test
- FLOPS test

However, it is also confirmed that the Intel PDT requires a long time to get a set of the above items. Thus, in this research, we develop in-house tools in order to obtain almost the same information.

It is needed to collect a large number and wide variety of normal data for machine learning in order to detect an abnormal behavior. In a Linux environment which includes WSL (Windows Subsystem for Linux), it is known that "`sar`" command can be used to get the values of cumulative activity counters in the operating system. Thus, we try to use the command to obtain the dynamic information. Several OS information files such as "`/proc/cpuinfo`" and "`/proc/meminfo`" are also referred as static and dynamic information of the current status. The above two test programs are also developed. The prime number generation test measures time that takes to generate up to 1000 prime numbers. The FLOPS test measures the number of floating-point operations performed per second. Each program is independently launched and the target values are obtained. The interval time of the consecutive launch of each program is specified as 60 seconds. Note that there are two methods to keep the program running at a regular time as follows;

- Use “sleep” and loop
- Use crontab or task scheduler

In the former method, the interval time between the executed programs can keep the specified value. However, the execution time of the own program is not taken into consideration. Thus, if a regular time is needed, a mechanism is required to parallelly execute both the program and “sleep” command so that they are started at the same time. “xargs” command can be used in order to simply solve it. In the latter method, they can specify the time of the server as a trigger, so the problem that occurs in the former method is solved. However, a new problem arises that the regular time cannot be set to less than 1 minute. Therefore, the former method is applied when the execution time is less than 1 minute. Otherwise, the latter method is applied.

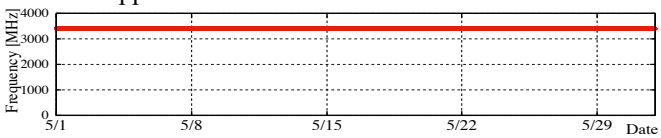

Fig. 1. Operating frequency

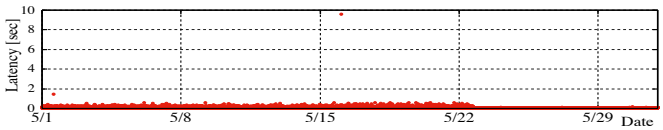

Fig. 2. Prime number generation test.

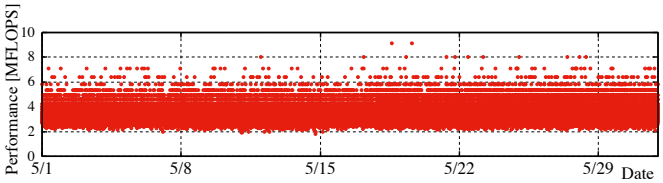

Fig. 3. FLOPS test.

Figure 1, 2, and 3 show the operating frequency, the execution time of the prime number generation test, and the evaluation results of the FLOPS test, respectively. Those data are collected from May 1st to May 31st. The horizontal axis represents measurement date and time.

In Fig. 1, the vertical axis represents operating frequency[MHz]. As shown in Fig. 1, no change can be observed although these numerical values are logically dynamic items. It can be considered that the continuous operations supposed in this evaluation may inhibit the dynamic voltage and frequency scaling (DVFS). Thus, it is required to collect the values under an assumption that the interval time is randomly selected considering the DVFS. In addition, “Intel Turbo Boost” and other factors should be also considered since they may affect the frequency characteristics.

In Fig. 2, the vertical axis represents execution latency[sec] to generate 1000 prime numbers. From Fig. 2, it can be observed that the latencies are usually varied within a certain range and its average value is 0.064[sec] and the standard deviation is 0.047[sec]. However, it can be also observed that they may become outlier values such as 9.6 on May 16th while no abnormal behavior is performed. Figure 3 shows the evaluation results of the FLOPS test. The vertical axis represents the calculation performance[MFLOPS]. It can be observed that the values are also varied within a certain range and its average value is 3.5[MFLOPS] and the standard deviation is 0.63[MFLOPS]. No extreme outlier value is observed in this evaluation. These collected data are used for AI learning.

III. Power consumption measurement

It can be considered that side channel information is also used to detect abnormal behaviors. One of considerable side channel information is power consumption. In this research, an automated measurement system of power consumption is developed in order to collect a large number of learning data. We introduce a measurement product which can check the power consumption of the target server whose supply voltage is 100[V] through a smartphone application. Normally, the value of the power consumption must be visually checked through the application which is installed in a tablet. Thus, macro instructions are used to automatically upload Excel-format power consumption data from the tablet to a data collection server. Figure 4 shows an example of power consumption measurement.

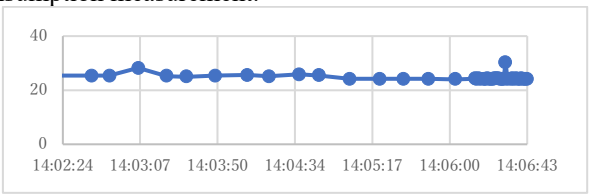

Fig. 4. Power consumption measurement.

However, in this evaluation, it is observed that the developed measurement system may cause several blanks for a certain period of time. Since there is no problem with the environment and used method, it is considered to be a problem with the product specifications. Currently, this product is used since there is no other alternative measurement instrument. If continuous data is required, another measurement method is reconsidered.

IV. Conclusion

This paper has proposed a diagnosis-based hardware Trojan tolerate system based on the deep learning scheme for certain configuration machines, and several methods and tools to collect data for AI learning. It is our future work to learn the AI and try to detect abnormal behaviors with the AI. It is also in the scope of our future work to improve the normal behavior model in which various loads and time to run are randomly selected.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Numbers JP20K11805 and JP21H04868.

References

[1] Mohammad Tehranipoor et al., “A survey of hardware trojan taxonomy and detection,” *IEEE Design & Test of Computers*, 27(1):10–25, 2010.
[2] Trust-hub. https://www.trust-hub.org/.
[3] Intel Processor Diagnostic Tool, https://www.intel.co.jp/content/www/jp/ja/support/articles/0000005567/processors.html