Impact of Cross-standard Cell Libraries on Machine Learning based Hardware Trojan Detection

Shang-Wen Chen, Jian-Wei Liao, Chia-Wei Tien and Jung-Hsin Hsiao
Cybersecurity Technology Institute, Institute for Information Industry, Taipei, Taiwan R.O.C.

Keywords: Hardware Trojan, Hardware Security, Machine Learning.

Abstract: Hardware Trojans (HTs) have become a new threat that owns a huge possibility to widespread into over the world because of its unique characteristic. Hardware Trojan is dependent on the invaded hardware, and if the invading success such that the devices which use the invaded hardware will spread to customers of hardware vendors all over the world. Thus, how to detect HT exists in our devices or not becomes an important issue. There are already some researches to try to solve this problem and acquire good results. The common premise of these researches is that the adopted standard cell library in model and testing set is the same. However, there is no good performance to detect HT with machine learning in reality under the above premise. The possible thinking is that adopted standard cell libraries of model and testing set are different in real case and it cause the bad result of machine learning. We experiment and verify this view. That is, we prove that the impact of cross-standard cell library on machine learning in hardware Trojan detection exists.

1 INTRODUCTION

There has a new category of hardware attack for integrated circuits (ICs) called hardware Trojan (HT) that appears in recent years. HTs have been known that they can trigger several serious attacks, though the number of reported HT is much less than software Trojans until now. However, the effects of HT that have already proved by the occurred events and the amount of HT will increase as time goes on. Therefore, the threat of HT is also rising with the time.

According to the related researches (R.S. Chakraborty et al., 2013) (Mukhopadhyay & Chakraborty, 2011), HT owns several properties. These properties contain many aspects such as disabling or altering the functionality of the IC, decreasing in reliability and expected lifetime of industry control system (ICS) (TrendMicro, n.d.), leaking sensitive user information through convert communication channels and bypassing the software security facilities and spy the users and so on. There are some cases that are caused by the front properties. For example, a Syrian radar failed to warn of an incoming air strike in 2007. The occurred reason is doubted as the potential backdoor built into system's chips (Mitra et al., 2015). Moreover, in 2014, the New York Times (David,E Sanger & Thom Shanker, 2014) reported that there is a program of US National Security Agency (NSA) called Quantum program which plans to implant HT circuitry into USB communication protocol or USB port. Besides, there are the other reports (Markoff, 2009) (Ellis, 2012) can display the influence of HT.

So far, there are some researches (Agrawal et al., 2007) (Danesh et al., 2014) (S. Jha & S. K. Jha, 2008) (Chakraborty et al., 2009) (Alkabani & Koushanfar, 2009) (Hasegawa et al., 2016) (Iwase et al., 2015) that try to solve the HT problem. They tried several different static methods such as IC fingerprints (Agrawal et al., 2007), side-channel analysis (Danesh et al., 2014) (Alkabani & Koushanfar, 2009), logic testing (S. Jha & S. K. Jha, 2008) (Chakraborty et al., 2009) and the other static analysis (Hasegawa et al., 2016) (Iwase et al., 2015) and acquired good results. There are brief introductions about the above papers in Table 1. Besides, the premise of these researches that use machine learning is that the used standard cell library of these gate-level netlists is the same one. However, there has a large difference between the premise and the reality. This could lead that the real performance of methods decreases in reality, and we try to prove it.

Chen, S., Liao, J., Tien, C. and Hsiao, J.
Impact of Cross-standard Cell Libraries on Machine Learning based Hardware Trojan Detection.
DOI: 10.5220/0010858800003120
In *Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP 2022)*, pages 420-425
ISBN: 978-989-758-553-1; ISSN: 2184-4356

Table 1: Related researches about solving HT problem.

Paper	Method	Description
(Agrawal et al., 2007)	IC fingerprints	1. Using noise modeling to construct a set of fingerprints for an IC family 2. The fingerprints utilizes side-channel information such as power, temperature, and electromagnetic (EM) profiles
(Danesh et al., 2014)	Side-channel analysis	Exploiting the special power characteristics of differential cascade voltage switch logic (DCVSL) to detect HTs at runtime
(Alkabani & Koushanfar, 2009)	Side-channel analysis	1. New Trojan detection method based on nonintrusive external IC quiescent current measurements 2. Using consistency, which is a new self-defined metric, and properties of function to detect Trojans
(S. Jha & S. K. Jha, 2008)	Logic testing	1. A randomization based technique to verify whether a manufactured chip is infected by Trojan 2. If infected, then this result and its fingerprint input pattern will be reported
(Chakraborty et al., 2009)	Logic testing	1. A test pattern generation technique based on multiple excitation of rare logic conditions at internal nodes 2. Increasing triggered and detected probability of Trojans and the sensitivity of Trojan detection

The contribution of this study to literture is two-fold. First, we propose an new idea about that there could be a bias of result of machine learning in Hardware Trojan detection between real case and related researches because of different premises. Second, we experiment and acquire the result of at least 10% decreasing of machine learning detection in Recall and F1-score. Thus, the correctness of our idea is verified. This provides an important new premise for next researchers.

The rest of the paper is organized as follows: Section 2 will describe the detail of problem encountered in reality and the difficulty in cross-library machine learning detection. Section 3 introduces our proposed method. Section 4 displays our experiment result of cross-standard cell library machine learning detection to prove the influence of different premises in reality. Section 5 describes our conclusion and future work.

2 BACKGROUND

In this chapter, we describe that the composition of hardware in cell's view and the reason of why the commonly used features are the parameters generated by standard cell library. Most important of all, the difficulty of cross-library hardware Trojan detection in machine learning.

Hardware Trojan is a kind of malware that launches its attack through hardware. Each hardware owns control chips which are composed of many cells. Each cell owns many control parameters like leakage power, area, footprint, details values of each pin, timing and so on. As the other perspective, a cell can be represented as a parameters pair. That is, the hardware can be viewed as the collection of parameters pairs.

These values of parameter in each cell is decided by the using standard cell library adopted by hardware, and the decision of adopted category of standard cell library is judged by hardware vendor. There are many different categories of standard cell library, and hardware vendors select their adopted standard cell library which is suitable for the working environment of vendors. For security, hardware vendors will not leak their using standard cell library and related information about their products. Thus, the parameters pairs of hardware are not only useful data but also the common used features that can acquire easily.

As the front mentioned, vendors will not reveal their adopted standard cell libraries for security. Besides, there exist many different standard cell libraries corresponding to their own specific situations. That is, it is almost impossible to encounter that there are two different companies which use the same standard cell library because the specific environment of each company is different.

This fact indicates that the premise, which we mentioned in last chapter, is almost wrong in reality. However, there almost has not any research to describe how to detect hardware Trojan in hardware which uses different standard cell libraries.

If we want to use machine learning to detect hardware Trojan in hardware with different standard cell libraries, there exist some foreseeable issues in it. First, the parameters pairs of cells in hardware are the common features and the values of parameter are decided by adopted standard cell library. Because every company uses different standard cell library, and different libraries are independent with each other. However, the basic hardware cells are the same. That is, even if the used standard cell libraries are different, there should be some relations between different parameters pairs generated by different standard cell libraries. However, we cannot confirm the relationship between parameters pairs that are decided by different standard cell libraries. There is no such definition to quantify this relationship. Second, if we train a model with the parameters' pairs decided by A standard cell library, and then use the parameters pairs which is generated by B standard cell library as testing set. Although the basic hardware cells are the same, after the values of parameters are processed by differently independent standard cell libraries. The performance of model is decided by how much relationship remains between these parameters pairs.

To evaluate how a model performance will be affected by parameters pairs generated by different cell libraries, we make an experiment of machine learning detection though using different standard cell libraries. Besides, this condition of experiment is closed to real situation. That is, the result of experiment owns reliability.

3 PROPOSED METHOD

In this chapter, we will introduce the process of our proposed method. Moreover, we also explain the meaning of parameter entered into program.

In our proposed method, the execution process can divide into two stages. The first stage is pre-processing stage, and the second stage is processing stage. The former will generate training set and testing set for machine learning according to input parameters. The latter will use the output of the first stage to train a model and output the result of hardware Trojan detection. Figure 1 shows the process diagram of our experiment and the detailed descriptions of this diagram will be state as follows.

In the first stage, we have to execute pre-processing twice to acquire the necessary training set and testing set used in machine learning. Then we use training set to generate a model that will be used in next stage. In each pre-processing, we have to provide six parameters to program. The parameters are circuit name, standard cell library, mapping on/off, mapping mode, filter threshold one and filter threshold zero. The detailed descriptions are described in Table 2. It is worth to notice that only the value of mapping on/off is changed between first and second pre-processing and the others are the same. After execution pre-processing twice, we can acquire training set and testing set which are used to generate the model of machine learning and test the performance of it. Moreover, the algorithms we used in model are Support Vector Machine (SVM) (Noble, 2006) and random forest (RF) (Breiman, 2001). After generating model and testing, we can acquire several evaluation metrics like accuracy, precision, recall, F1-score and etc.

In the second stage, we will firstly repeat the same process of first stage from beginning to generating training set and testing set. In this part, the only difference is that the used standard cell library of parameters is different from the one used in first stage. Then we can acquire new training set and testing set which are generated by new standard cell library. We use this new testing set to test the model generated in first stage and acquire the new outputs of evaluation metrics mentioned in last paragraph.

4 EXPERIMENTS

In this chapter, we will describe our experiment to display the comparison of the results of machine learning with different standard cell libraries.

We make an experiment to prove that the features compiled by different standard cell libraries will affect the performance of machine learning detection. We describe the dataset used in this experiment at first. In total, we collected 199 different netlists from various sources including public and private ones. However, there is a problem of non-disclosure agreement if we used the netlists collected from private source. Thus, we selected the 88 netlists collected from public source, Trust-Hub (Trust-Hub.org, n.d) (Salmani et al., 2013) (Shakya et al., 2017). Moreover, we collected 144 different standard cell libraries from private source and randomly chose two libraries as the compiler of training set and testing set.

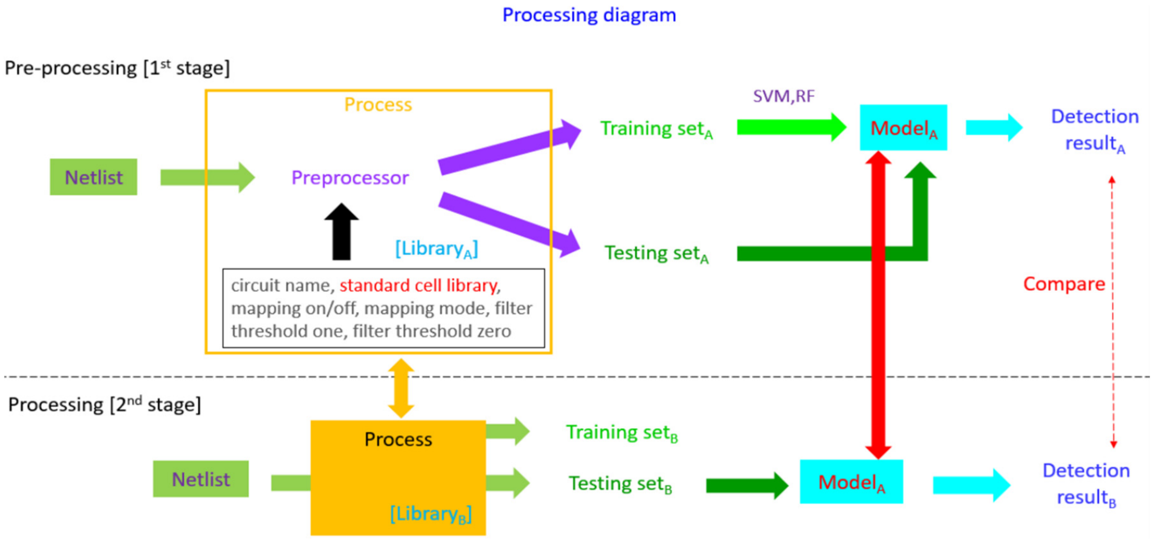


Figure 1: A processing diagram of experiment.

Table 2: The description of parameter that used in first stage of execution process.

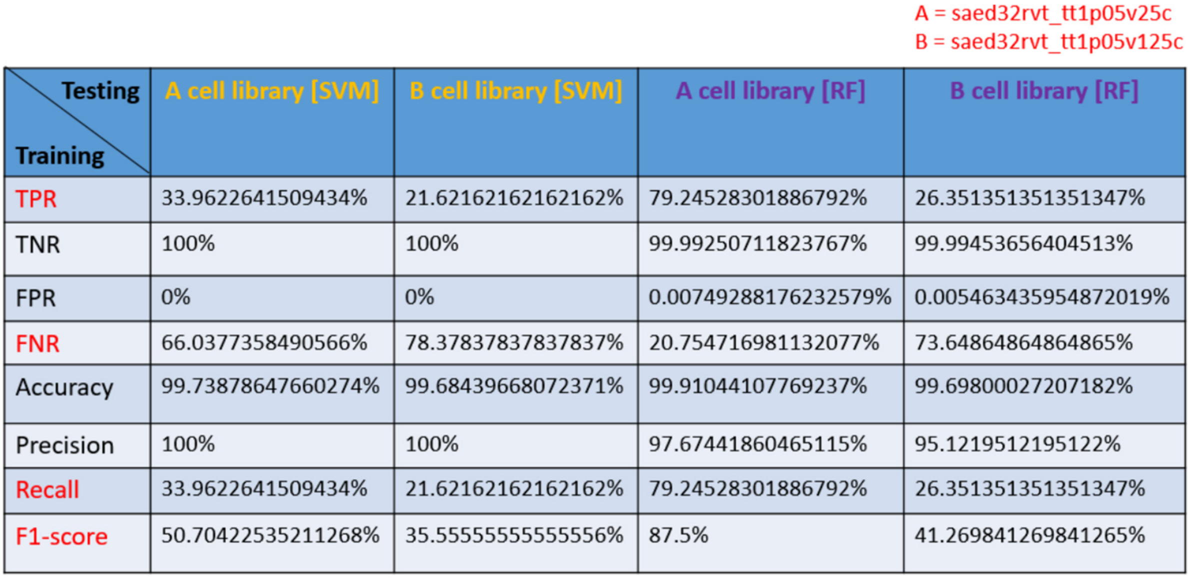
Parameter name	Description
circuit name	Name of gate-level netlist waiting to execute pre-processing
standard cell library	The path of standard cell library used in composition stage
mapping on/off	Decision of adding time order information into path features or not in this execution
mapping mode	Decision of allowing or disallowing that path exists error or not when adding time order information
filter threshold one	Threshold of percentage of signal which indicates 1 in path filter
filter threshold zero	Threshold of percentage of signal which indicates 0 in path filter

For comparing the results of machine learning with different libraries in the same gate-level netlist, we compared the outputs of evaluation metrics of first stage and second stage. In the first stage, we choose one of our collected standard cell libraries called "saed32rvt_tt1p05v25c." Besides, we adopt the other standard cell library called " saed32rvt_tt1p05v125c" in the second stage. The comparison result displays in Figure 2. In Figure 2, we can observe that no matter what algorithm the model uses, the result of testing set generated by "saed32rvt_tt1p05v125c" library shows a large difference in several metrics such as TNR, FNR, recall and F1-score. There exist at least 12% decrease in the TPR, recall and F1-score metric when model is used to detect the features generated by the second standard cell library. On the other hand, there exist at least 12% increase in the FPR metric when model is used to detect the features generated by the second standard cell library.

Based on the experiment result, we can confirm that there exists a great difference to the detection result of hardware Trojan in machine learning when adopting different cell libraries so that the detection of cross-standard cell library hardware Trojan is a great challenge to machine learning.

5 CONCLUSIONS

In this paper, we introduce a new category of threat called hardware Trojan and its possible serious effect to all over the world. Although there have been existed several researches to discuss this issue and acquire some good results. However, they focused on detecting hardware Trojan that used the same standard cell library. On the other hand, we research on the problem of the detection of cross-standard cell library hardware Trojan that is the common case in reality. This is a new research field but less people to



A = saed32rvt_tt1p05v25c
B = saed32rvt_tt1p05v125c

Testing Training	A cell library [SVM]	B cell library [SVM]	A cell library [RF]	B cell library [RF]
TPR	33.9622641509434%	21.62162162162162%	79.24528301886792%	26.351351351351347%
TNR	100%	100%	99.99250711823767%	99.99453656404513%
FPR	0%	0%	0.007492881762325 79%	0.0054634359548720 19%
FNR	66.0377358490566%	78.37837837837837%	20.754716981132077%	73.64864864864865%
Accuracy	99.73878647660274%	99.68439668072371%	99.91044107769237%	99.69800027207182%
Precision	100%	100%	97.67441860465115%	95.1219512195122%
Recall	33.9622641509434%	21.62162162162162%	79.24528301886792%	26.351351351351347%
F1-score	50.70422535211268%	35.55555555555556%	87.5%	41.269841269841265%

Figure 2: Comparison of testing results between different libraries.

research it because of its difficulty. We hope this study can be the prior knowledge of the follow-up investigators.

REFERENCES

R.S. Chakraborty, I.Saha, A.Palchaudhuri, G.K.Naik, (2013). “Hardware Trojan insertion by direct modification of FPGA configuration bitstream”, IEEEDes. Test30 (2), pp. 45–54.

D. Mukhopadhyay and R. S. Chakraborty, (2011). ‘‘Testability of cryptographic hardware and detection of hardware Trojans,’’ in Proc. IEEE Asian Test Symp. (ATS’11), pp. 517–524.

S. Mitra, H.S.P.Wong, S.Wong, (2015) “The Trojan-proofchip”, Spectr.IEEE52 (2), pp. 46–51

D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, (2007). “Trojan Detection using IC Fingerprinting,” in Security and Privacy, SP ’07. IEEE Symposium on, pp. 296–310

W. Danesh, J. Dofe and Q. Yu, (2014) "Efficient hardware Trojan detection with differential cascade voltage switch logic", *Proc. VLSI Des.*, pp. 1-10.

S. Jha and S. K. Jha., (2008). “Randomization Based Probabilistic Approach to Detect Trojan Circuits”, in Proc. IEEE High Assurance Systems Engineering Symposium – HASE, pp. 117–124

R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia., (2009). “MERO: A Statistical Approach for Hardware Trojan Detection”, in Proc. Cryptographic Hardware and Embedded Systems – CHES, volume 5747, pp. 396–410

Y. Alkabani and F. Koushanfar., (2009). “Consistency-based Characterization for IC Trojan Detection”, in Proc. IEEE International Conference on Computer-Aided Design – ICCAD, pp. 123–127.

K. Hasegawa, M. Oya, M. Yanagisawa, and N. Togawa, (2016) “Hardware Trojans classification for gate-level netlists based on machine learning,” in Proc. IEEE Symposium on On-Line Testing and Robust System Design (IOLTS), pp. 203–206

T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, (2015) “Detection technique for hardware Trojans using machine learning in frequency domain,” in 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), pp. 185–186

David,E Sanger, Thom Shanker, (2014, January 14) “N.S.A. Devises Radio Pathway Into Computers” http://www.nytimes.com/2014/01/15/us/nsa-effort-pries-open-computers-not-connected-to-internet.html

TrendMicro. (n.d.). Industrial Control System. Retrieved October 10, 2021, from https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system

J. Markoff, (2009, October 26). “Old Trick Threatens the Newest Weapons,” http://www.nytimes.com/2009/10/27/science/27trojan.html?pagewanted=all& r=1&

J. Ellis, (2012, February 27). “Trojan integrated circuits,” http://chipsecurity.org/2012/02/trojan-circuit/.

Noble, W. S., (2006). “What is a support vector machine?” *Nature Biotech*, 24 (12), 1565–1567

Breiman L, (2001). Random forests. Machine Learning, 45(1): 5–32

Trust-Hub.org. (n.d.). Chip-level Trojan Benchmarks. Retrieved from https://www.trust-hub.org/#/benchmarks/chip-level-trojan

H. Salmani, M. Tehranipoor, and R. Karri, (2013). "On Design vulnerability analysis and trust benchmark development," in IEEE Int. Conference on Computer Design (ICCD)

B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, (2017), “Benchmarking of Hardware

Trojans and Maliciously Affected Circuits," Journal of Hardware and Systems Security (HaSS)