Hardware Trojan Detection Combine with Machine Learning: an SVM-based Detection Approach

Taifeng Hu[1], Liji Wu[1*], Xiangmin Zhang[1], Yanzhao Yin[1], Yijun Yang[1,2]
[1] Institute of Microelectronics, Tsinghua University, Beijing, China
[2] Department of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China
{htf18, lijiwu, zhxm, yinyz17}@mails.tsinghua.edu.cn; yangyijun@link.cuhk.edu.hk

***Abstract*—With the application of integrated circuits (ICs) appears in all aspects of life, whether an IC is security and reliable has caused increasing worry which is of significant necessity. An attacker can achieve the malicious purpose by adding or removing some modules, so called hardware Trojans (HTs). In this paper, we use side-channel analysis (SCA) and support vector machine (SVM) classifier to determine whether there is a Trojan in the circuit. We use SAKURA-G circuit board with Xilinx SPARTAN-6 to complete our experiment. Results show that the Trojan detection rate is up to 93% and the classification accuracy is up to 91.8475%.**

Keywords—hardware trojan; trojan detection; side-channel analysis (SCA); support vector machine (SVM)

I. INTRODUCTION

Integrated circuits (ICs) have become indispensable part of lives, we can see them everywhere in daily life, such as cell phone, computer and so on. At the same time, ICs are often used to control and store important information in the above devices, if this important information are maliciously used by others, the loss will be huge. So, the security of ICs is critically important. Among many security threats, Hardware Trojan (HT) is a typical example. Hardware Trojan can be added at any step of the IC production, once HT is inserted into ICs, the ICs will have great potential risk. In recently years, many different Trojan detection methods have been researched.

The common Trojan detection methods can be divided into three part: (1) Reverse Engineering (RE); (2) Logic testing; (3) Side-Channel Analysis (SCA). In [1], the authors conducted a survey on reverse engineering to assess the feasibility of different RE ways. The authors in [2] proposed a new holistic framework named HAL to automates reverse engineering tasks, their Trojan detection is efficient by use HAL. However, most of REs are destructive, the cost of testing can be very high. Logic testing require a specialized test bench. In addition, it will take a lot of time to detect the Trojan. In comparison, side-channel analysis is non-destructive and efficient. Many kinds of data can be used as side-channel analysis, such as power, electromagnetic and path delay. The method developed in [3] use path delay as a side-channel signal to detect Hardware Trojan, but the increased delays are too small to detect at real time. In order to solve this problem, the authors in [4] use a relative time delays to find the hardware Trojan and achieved good results.

The authors in [5] used machine learning for detection, the method is for gate-level netlists, the true positive rate is 100%, but the process of extracting netlists is difficult. Trojan detection method by using SVM was mentioned in [6,7].

In this paper, we use machine learning methods (cross-validation and SVM) for Trojan detection. Firstly, we introduce some characteristics of hardware and the broadly classification of Trojans. Then, we set AES-128 as background circuit and insert the Trojan which was designed in [8]. The entire circuit is performed successfully on SAKURA-G circuit board with Xilinx SPARTAN-6. After that, we detect hardware Trojan using the method which is proposed by us. Finally, we envision future research direction.

The rest of this paper is organized as follows: Section II introduces some knowledge of hardware Trojan and SVM. Section III describes the specific steps of our experiments and analyzes the results. Section IV concludes this paper and gives future direction.

II. PRELIMINARIES

A. Characteristics of Hardware Trojans

Hardware Trojans have three key characteristics: malicious intention, evasion of detection, and rarity of activation[9]. To be specific, if a hardware Trojan is inserted into a circuit, it will be difficult to be detected due to its stealthies, and the Trojan may cause circuit function failure under certain conditions or steal important information, such as private keys. Trigger and payload are two main components for hardware Trojans. Fig.1 shows the typical structure of a hardware Trojan which is inserted into the circuit[10]. The trigger always monitors specific signals (such as reset signal) in the circuit and activities when an expected event occurs, which is typically derived from the original circuit, when the trigger signal is sent to the payload, then it will deliver the malicious behavior to the circuit[10].

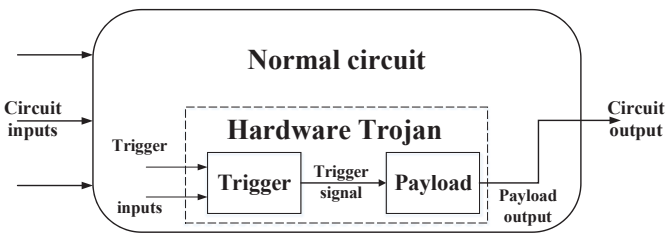

Figure 1. The typical structure of a hardware Trojan in [10].

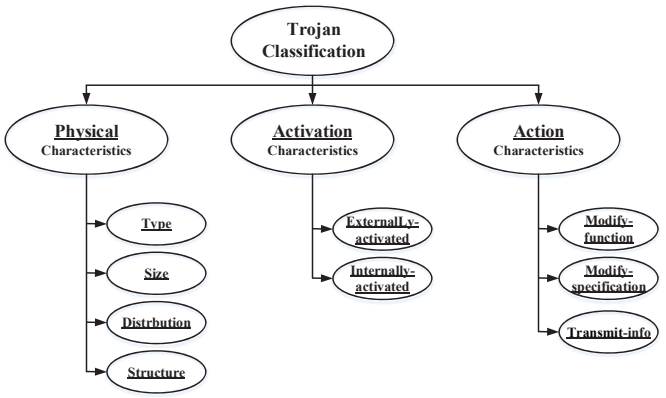

Figure 2. Taxonomy of Trojans in [11].

According to hardware Trojan physical, activation and action characteristics, we can divide hardware Trojans into three main types[11], which is shown in Fig.2.

B. Support Vector Machine (SVM)

The Support Vector Machine (SVM) first proposed by Cortes and Vapnik in 1995, which is an efficient algorithm for both classification and regression problems. For a given sample training set as shown in Fig.3, we try to find a decision boundary from the sample space, separating class 1 (y = +1) and class 2 (y = -1). In sample space, decision boundary can be described by the following equation:

$$\omega^T x + b = 0 \tag{1}$$

where, ω is normal vector, which decide the direction of the decision boundary; x is eigen vector and b is bias, which decide the distance between the decision boundary and the origin.

If the decision boundary is chosen correctly, there is y = +1, $\omega^T x + b > 0$ and y = -1, $\omega^T x + b < 0$. We set:

$$\begin{cases} \omega^T x + b \geq +1, & y = +1; \\ \omega^T x + b \leq -1, & y = -1. \end{cases} \tag{2}$$

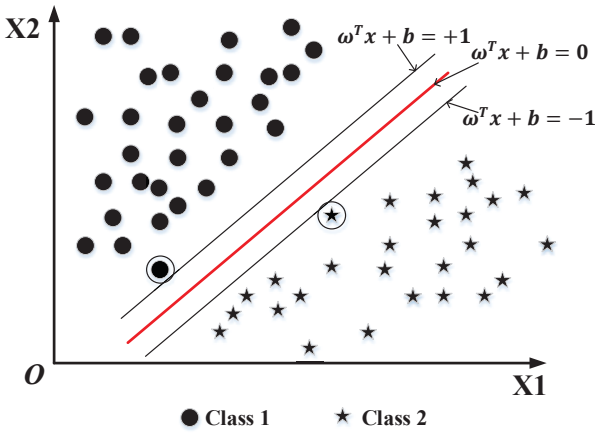

Figure 3. An example of SVM.

In SVM, the closest data points to decision boundary are defined as Support Vectors (SVs). In this example, SVs make equality in (2) they are circled in Fig.3. These SVs are used to predict the class of test record in the "prediction phase"[12].

The corresponding model of the decision boundary in the above example is:

$$f(x) = \omega^T x + b \tag{3}$$

where, ω and b is model parameter.

SVM is to find the parameters ω, b and to get the decision boundary implement classification.

The above classification is linear. In order to develop complex nonlinear classifier, the main solution is to use a kernel function.

The above formula becomes like this:

$$f(x) = \omega^T \phi(x) + b \tag{4}$$

where, $\phi(x)$ is the mapping function of x.

then define the kernel function as:

$$\kappa(x_i, x_j) = < \phi(x_i), \phi(x_j) > = \phi(x_i)^T \phi(x_j) \tag{5}$$

The kernel function can map the sample from the original space to a higher-dimensional feature space, making the sample linearly separable in the high-dimensional space. Common kernel functions are shown in TABLE I.

III. Experiments and Results

A. ICs Used in Our Experiments

We used AES-128(Advanced Encryption Standard) circuit as the benchmark circuit in this paper. The hardware Trojan was proposed by [8], the purpose of the Trojan in the circuit is to leak the secret key of AES algorithm through a stealthy channel. The stealthy channel is based on the theory of Code-Division Multiple Access (CDMA). First, CDMA code sequence is created by a pseudo-random number generator (PRNG). Subsequently, the generated code sequence is changed to secret information bits by XOR modulate. The modulated sequence is forwarded to a leakage circuit (LC) to set up a covert CDMA channel in the power side-channel[8]. Fig.4 shows the structure of this kind of Trojan. This IC is called as AES128-T100. In our experiments, we use SAKURA-G FPGA board with Xilinx

TABLE I. Common Kernel Function

Kernel name	Kernel function
Linear	$\kappa(x_i, x_j) = {x_i}^T x_j$
Polynmoial	$\kappa(x_i, x_j) = ({x_i}^T x_j)^d$
Gaussian	$\kappa(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right)$
Sigmoid	$\kappa(x_i, x_j) = \tanh(\beta {x_i}^T x_j + \theta)$

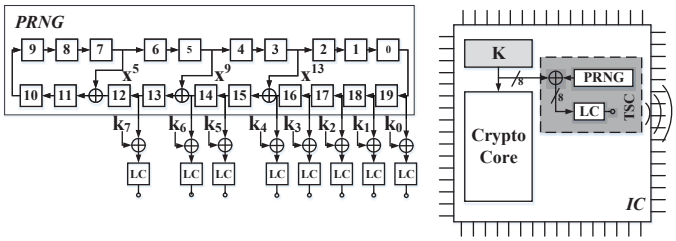

Figure 4. The Trojan in [8].

TABLE II. AREA ANALYSIS IN SAKURA-G FPGA

Circuit name	Number of Slice Registers used	Number of Slice LUTs used	Number of LUT Flip Flop pairs used
AES-128 (Trojan-free)	5698	8773	9618
AES-128 (Trojan-in)	5698	8759	9656

SPARTAN-6 to implement AES-128 (both Trojan-Free circuit and Trojan-in circuit), two SPARTAN-6 FPGAs are integrated on this board and serve as the communication circuit and AES-128 circuits, respectively.

TABLE II shows area analysis of Trojan-free circuit and Trojan-in circuit.

B. Data Acquisition and Preprocessing

The first step is data acquisition, we collected 1000 power waveforms as samples in both situation (Trojan-free and Trojan-in), respectively. As shown in Fig.5.

However, the power waveforms which are collected from the oscilloscope include many high frequency noises. Therefore, data preprocessing is required before Trojan detection. Its steps including: (1) Filter the original power waveform to remove high frequencies; (2) Reduce sampling points of the power waveform by averaging (i.e. reduce the number of feature vectors); (3) Sort power waveforms randomly; (4) Use the processed data to get a M×N matrix, where M is the number of power waveforms, N is the number of sampling points; (5) Divide the M×N matrix into three parts,

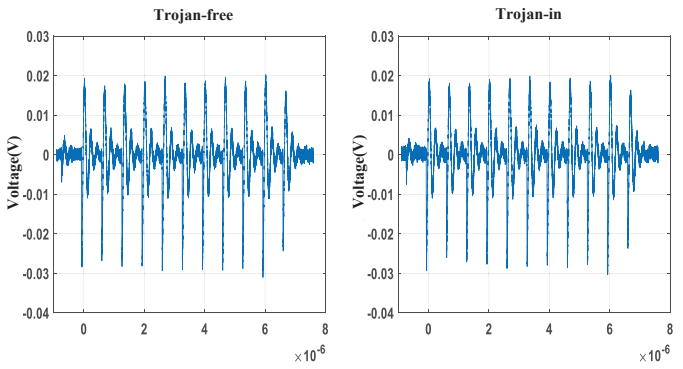

Figure 5. Power waveform before data processing.

corresponding to the training set, cross-validation set and test set, respectively.

After the data preprocessing, we get a 2000×5000 matrix. Then we divide it into three parts, which are shown as follow:

- Training set (60%): a 1200×5000 matrix;
- Cross-validation set (20%): a 400×5000 matrix;
- Test set (20%): a 400×5000 matrix.

The final step is to generate the output matrix of the training set and cross validation set.

There are only "0" and "1" in the matrix, where "0" is refer to Trojan-free and "1" is refer to Trojan-in. The matrix size is as follows:

- Outputs of training set: a 1200×1 matrix;
- Outputs of cross-validation set: a 400×1 matrix.

C. Train and Predict by SVM

In our experiment, we choose Gaussian kernel as our kernel function and use cross-validation set to select the optimal values of the penalty parameter C and kernel function parameter σ. First, we set the range of C and σ very large (e.g. 0.01, 0.05, 1, 5, 10, 50, 100, 500). Then, gradually reduce the range according to the training results. Finally, we can get the result of the parameter selection as shown in Fig.6. The parameters maybe not the same in different tests, the reason is random-ordered power waveforms.

We use the true positive rate (TPR), the true negative rate (TNR) and the accuracy as indices for the classification results, they are calculated from the following four indices:

- True positive (TP): The number of Trojan-in samples predicted to be Trojan-in correctly;
- False positive (FP): The number of Trojan-free samples predicted to be Trojan-in mistakenly;
- False negative (FN): The number of Trojan-in samples predicted to be Trojan-free mistakenly;
- True negative (TN): The number of Trojan-free samples predicted to be Trojan-free correctly.

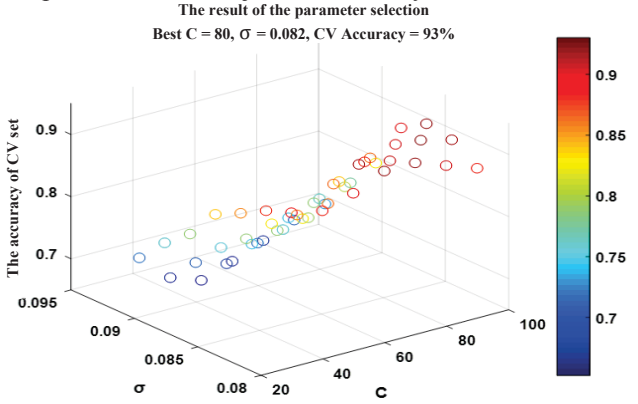

Figure 6. Use cross-validation set to select optimal C and σ.

TABLE III. CLASSIFICATION RESULTS OF AES128-T100

Test number	Parameter		Indices		
	C	*σ*	*TPR*	*TNR*	*Accuracy*
1	70	0.084	93%	90.5%	91.75%
2	80	0.084	95.5%	92%	93.75%
3	70	0.08	93%	92%	92.5%
4	60	0.08	90.5%	91.5%	91%
5	50	0.09	89.5%	91.5%	90.5%
6	60	0.08	90.5%	95%	92.75%
7	80	0.082	92.5%	93%	92.75%
8	90	0.092	96%	83.5%	89.75%

The formula of TPR is TP/(TP+FN), the formula of TNR is TN/(TN+FP) and the formula of accuracy is (TN+TP)/(TP+FP+FN+TN). In this paper, we can call the TPR as the rate of Trojan detection. TABLE III shows the classification results of AES128-T100 and Fig.7 (a) - (h) show the actual results of classification.

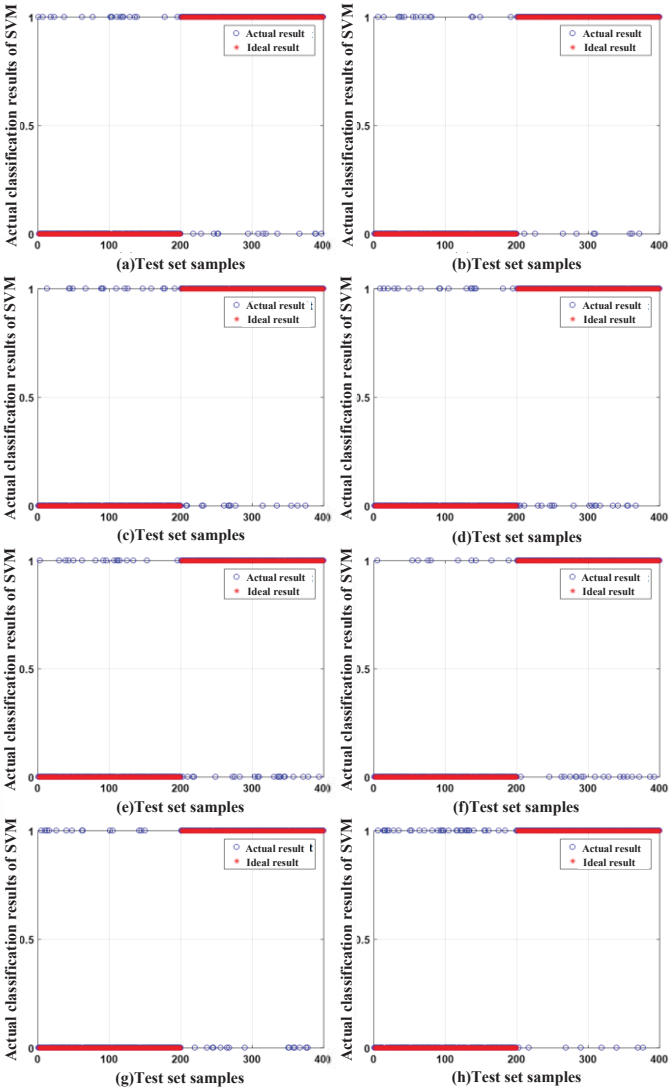

Figure 7. The actual results of classification.

From the TABLE III, we can see a good Trojan detection result: The rate of average accuracy is 91.8475%, the average value of TNR is 91.125% and the average value of TPR is 92.5625%. In other words, the recognition rate of Trojan-in circuit is higher than that of Trojan-free circuit.

As mentioned in the previous section, the parameters C and σ are different in most tests, but all of them within a reasonable range. In the best case, the highest recognition rate can reach 93.75%, the C equals 80 and σ equals 0.084.

In Test 8, the detection accuracy of the Trojan-free circuit is only 83.5%, probably because the parameter selection is inaccurate, but we still believe the effectiveness of our detection method.

IV. FUTURE DIRECTIONS AND CONCLUSIONS

In this paper, we combine machine learning with Trojan detection. By using the SVM to analyze the side-channel power consumption information to distinguish whether there is a Trojan in the circuit. The experimental results demonstrate that the Trojan detection rate is up to 93% and the classification accuracy is up to 91.8475%. Above results show that our method is effective and reliable, it can be used in hardware Trojan detection.

In the future, we will detect more different types of Trojans and apply more machine learning methods to Trojan detection to increase the Trojan detection rate.

ACKNOWLEDGMENT

This work was supported by the National Major Program "Core of Electronic Devices, High-End General-Purpose Chips, and Basic Softwares" of the Ministry of Industry and Information Technology of China (No. 2017ZX01030301).

REFERENCES

[1] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi and L Wang et al. "A survey on chip to system reverse engineering," JETC, 13(1):6:1–6:34, 2016.

[2] M. Fyrbiak, S. Wallat, P. Swierczynski, M. Hoffmann, S. Hoppach and M. Wilhelm et al. "HAL−The Missing Piece of the Puzzle for Hardware Reverse Engineering, Trojan Detection and Insertion," in IEEE Transactions on Dependable and Secure Computing, vol. 16, no. 3, pp. 498-510, 1 May-June 2019.

[3] G. Zarrinchian and M. S. Zamani, "Latch-based structure: a High resolution and self-reference technique for hardware trojan detection," in IEEE Transactions on Computers, vol. 66, no. 1, pp. 100-113, 1 January. 2017.

[4] Y. Yang, L. Wu, X. Zhang and J. He, "A novel hardware trojan detection with chip ID based on relative time delays," 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, pp. 163-167, 2017.

[5] K. Hasegawa, M. Oya, M. Yanagisawa and N. Togawa, "Hardware Trojans classification for gate-level netlists based on machine learning," 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, pp. 203-206, 2016.

[6] T. Inoue, K.Hasegawa, M. Yanagisawa and N. Togawa, "Designing hardware trojans and their detection based on a SVM-based approach," 2017 IEEE 12th International Conference on ASIC (ASICON), Guiyang, pp. 811-814, 2017.

[7] Q. Cui, K. Sun, S. Wang, L. Zhang and D. Li, “Hardware trojan detection based on cluster analysis of mahalanobis distance,” 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, pp. 234-238, 2016.

[8] L. Lin, M. Kasper, T. Güneysu, C. Paar and W. Burleson, “Trojan side‐channels: lightweight hardware trojans through side-channel engineering,” 11th International Cryptographic Hardware and EmbeddedSystems(CHES),pp.382-395, 2009.

[9] S. Bhunia, M. S. Hsiao, M. Banga and S. Narasimhan, “Hardware trojan attacks: threat analysis and countermeasures,” in Proceedings of the IEEE, vol. 102, no. 8, pp. 1229-1247, Augest. 2014.

[10] S. Bhunia and M. M. Tehranipoor, “The hardware trojan war: attacks, myths and defenses,” Springer Publishing Company, Incorporated, 2017.

[11] X. Wang, M. Tehranipoor and J. Plusquellic, “Detecting malicious inclusions in secure hardware: challenges and solutions,” 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, pp. 15-19, 2008.

[12] A. Kulkarni, Y. Pino and T. Mohsenin, “SVM-based real-time hardware Trojan detection for many-core platform,” 2016 17th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, pp. 362-367, 2016.

[13] “Trust-HUB.” http://www.trust-hub.org.