OERCA: A Novel Approach for Hardware Trojan Detection Based on Combinatorial Testing

Yi-lun ZHANG∗, Ying-jian YAN, Xin CHUAN and Jun-wei LI
Zhengzhou Information Science and Technology Institute, Zhengzhou, China
∗Corresponding author

Keywords: Hardware trojan, Logic testing, RCA.

Abstract. With the rapid development of the integrated circuit industry, hardware Trojan detection has become a research hotspot in the field of hardware security. Logic testing is a very effective method among Trojan detection. However, the test vector set generated by this method increases sharply with the increase of circuit size. And it is difficult to rapidly detect hardware Trojans, even affecting the availability of detection technology. Therefore, this paper presents a hardware Trojan detection method that can effectively reduce the test vector set and enhance the pertinence of test vector set. By constructing the rare combination array (RCA) of circuit, the range of rare node combinations is effectively reduced. And the size of the test vector set is shortened by at least once excitation of rare combination array (OERCA). Experiments show that the hardware Trojan detection method proposed in this paper can reduce greatly the length of test vector set compared to other methods based on logic testing.

Introduction

Hardware Trojan [1] was first proposed in 2007, which refers to malicious circuits that are inserted into hardware circuits and can be inserted at various stages of chip design, causing circuit failures, information leakage, and other hazards. Therefore, the hardware Trojan detection technology has become one of the research hotspots in the field of secure circuits.

Common detection methods include reverse engineering, side channel analysis and logic testing. Reverse engineering [2] compares the chip with the original layout of the chip after photographing it layer by layer to determine whether there are hardware Trojans in the chip. However, as the size of the chip increases, this method is very expensive. The detection method based on the side channel analysis [3] mainly determines whether a hardware Trojan is inserted in the circuit by analyzing the side channel signal such as the power consumption in the circuit. There are two main drawbacks of this method. One is that it need a golden circuit that is not infected with a hardware Trojan, and the other is that it is susceptible to environmental noise and process noise [4].

Compared with the hardware Trojan detection method based on side channel analysis, the advantage of hardware Trojan detection method based on logic testing is that it doesn't need to consider the influence of noise, and it doesn't need to have the golden circuit of uninfected hardware Trojan. The main difficulty of this method is to find a test vector that can efficiently activate the hardware Trojan. Therefore, this paper studies how to generate a test vector set that is as small as possible and effectively detect a combinational hardware Trojan.

The remaining structure of this paper is as follows. Section 2 summarizes the combinational hardware Trojan model based on rare nodes and analyzes the current logic testing. Section 3 gives a combinational hardware Trojan detection method for rare nodes is proposed. Section 4 aims at inserting combinational hardware Trojans in the ISCAS circuit, and we provide results with the OERCA detection method; Section 5 concludes the paper.

Hardware Trojan Detection

In order to hide hardware Trojans better for hardware Trojan attacker, hardware Trojan is generally triggered by the node with a lower transition probability inside the circuit. This node with a low
transition probability is called a rare node. The attacker will not use all rare nodes as hardware Trojan triggers by selecting some rare nodes to trigger the hardware Trojan. Selecting some rare nodes to trigger the hardware Trojan greatly reduces the change of side channel information such as power consumption for selecting all the nodes, making the hardware Trojan more threatening, and the combination of rare nodes has a lower transition probability than that of a single rare node. Therefore, this kind of combinational hardware Trojan model based on rare nodes is highly concealed and can cause great harm to the circuit.

As shown in Figure 1, there are the inputs $I_1, I_2, ..., I_n$ of the circuit, and the outputs $O_1, O_2, ..., O_n$. Hardware Trojan designers find some of the rare nodes in the circuit as triggers of the hardware Trojan, e.g., the combination of $N_1$ and $N_4$ triggers $T_1$ value and then passes to the payload $P_1$, which passes to the output finally, making the original correct output value error. A hardware Trojan attacker can activate the hardware Trojan effectively at any time by inputting a special vector $IV$.

![Figure 1. Combinational hardware Trojan model based on rare nodes.](image)

Against this kind of hardware Trojan, Rajat Subhra Chakraborty et al. [5] proposed the MERO method to count rare nodes. At least all rare nodes are activated $N$ times, then all rare node combinations can be activated, thereby triggering the hardware Trojan. Sayandeep Saha et al. [6] combined genetic algorithm (GA) and Boolean satisfiability (SAT) to improve the efficiency of MERO bit-by-bit altering candidate vectors. And through fault simulation, it can test whether the payload can propagate to the output.

Paris Kitsos et al. [7] proposed that the hardware Trojan trigger the Trojan through partial input of the circuit, and then the hardware Trojan can be detected by generating overlays of these possible Trojan trigger signals. Detectors can cover a large number of input combinations by combinatorial testing to generate a smaller test vector set, thus greatly reducing the length of test vector set.

The method proposed by the previous researchers can detect a hardware Trojan based on combinations of rare nodes, but there are still some deficiencies. There are two questions in MERO detection methods. One is that the resulting test vectors may be too large, with more internal nodes in the circuit. The other is that the test time may be too long. Due to the bit-by-bit change for each vector, the algorithm may execute at a slow rate. The GA+SAT method also has the problem of excessively large test vectors for circuits with more rare nodes. In this paper, the detection method of OERCA is proposed to reduce the size of the test vector set effectively based on the idea of MERO and combinatorial testing.

**Combinational Hardware Trojan Detection Method**

For each rare node, it is activated $N$ times independently, which may lead to activation several rare events for some times. At the same time, it may not trigger for some difficult-to-trigger combinations. Therefore, this paper considers reducing the scope of rare node combinations, using the idea of combinatorial testing to generate a rare combination array that can cover all rare node combinations at least once.
The steps of the OERCA detection method proposed in this paper are as follows. Firstly, the rare nodes are searched and filtered, then the rare combination array is constructed. Finally, the rare combination array is activated at least once to quickly find the test vector set.

**Finding Rare Nodes**

In order to find the rare node of circuit, this paper simulates the circuit and analyzes the rare node of the circuit. In the random simulation, the transition probability of a rare node is relatively low compared to other nodes, assuming that a node’s 1,0 probability is $P_1, P_0$ respectively, then the transition probability $P_t$ of the node is as shown in equation (1). The transition probability of the rare nodes generally below 0.2 or 0.1.

$$P_t = P_1 \cdot P_0 \quad (1)$$

When searching for rare nodes, we need to consider whether the random vector can activate all the nodes to count the transition information of the nodes. Therefore, we use the coverage report of model sim to get the toggle coverage of the random vector used in the simulation. Usually meeting the requirement in 99%, this shows that the random vector generated can activate most of the nodes.

In order to reduce the scope of the rare nodes, two factors need to be considered. One is that there may be input and output nodes in the statistical rare nodes. The input node does not belong to the internal rare node, and the output node directly becomes valid after being used as a trigger.

Therefore, we consider removing these nodes. The second is that there may be nodes with correlation in the statistical rare nodes, as shown in Figure 2. If $N_1$ and $N_2$ are related nodes, (a) shows the case where the output of $N_1$ is connected to the input of $N_2$, and (2) shows the case of the input of $N_1$ is connected to the input of $N_2$. When the input or output of the next node is completely related to the previous node, the two nodes is generally related; otherwise, the logic values of the two nodes are not necessarily related. Since there is a certain logical relationship of these related nodes, these related nodes will always remain at a fixed value when selecting the $n$ random vectors that activate a certain node to be simulated. Therefore, this paper generates $n$ random test vectors in the test vector generation algorithm by analyzing the correlation of nodes, which is used to filter rare nodes.

![Figure 2. related nodes with $N_1$ and $N_2$.](image)

**Constructing Rare Combination Array**

This paper constructs a rare combination array for all rare nodes and narrows the scope of possible hardware Trojan triggers. According to the analysis of Section 2, a hardware Trojan attacker selects some rare nodes among all rare nodes as the trigger part of the hardware Trojan, making the hardware Trojan hide better. For example, the hardware Trojan attacker selects the $W_1, W_2$ nodes in 10 nodes, and 0, 1 are respectively as rare values of two rare nodes. There are four cases of hardware Trojan triggers. In order to compress the number of possible Trojan triggers, the paper reduces the range of the combination of rare nodes by combinatorial testing.

Combinatorial testing is firstly used for software testing. The concept of a coverage array (CA) has been introduced in order to represent all situations that are detected. A coverage array $CA(N; t, k, v)$ is an $N \times k$-order matrix defined on a set of $v$-elements, satisfying for each $N \times t$-degree submatrix, at least all $t$-element ordered subsets on $v$ are at least once. In [7], the method of combinatorial testing applies in the input of the circuit. This paper uses the method of combinatorial testing to apply to compress the number of the combinations of rare nodes. A rare combination array (RCA) is generated for all rare nodes in the circuit, that is, it covers the case where $t$ nodes are
selected among all rare nodes. The paper will use the strength of $t=2,3,4$ as the number of rare nodes triggered by the hypothetical hardware Trojan, which is called combination correlation coefficient; $v$ represents the dimension, since each node has a value of 0 or 1, $v$ is a binary set; $k$ stands for the number of all rare nodes in the circuit under test; $N$ represents the total number of all combinations determined by the parameters $t$, $k$, $v$. This paper assumes that the total number of rare nodes of the circuit $k$ and the selected number of rare nodes $t$ are determined.

Table 1 Number of RCA size and covered combinations.

<table>
<thead>
<tr>
<th>$t$</th>
<th>RCA size</th>
<th>Covered combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13</td>
<td>13944</td>
</tr>
<tr>
<td>3</td>
<td>42</td>
<td>762272</td>
</tr>
<tr>
<td>4</td>
<td>117</td>
<td>30872016</td>
</tr>
<tr>
<td>5</td>
<td>317</td>
<td>987904512</td>
</tr>
</tbody>
</table>

When the c880 circuit is selected, Table 1 presents the number of the RCA size and the actual number covered rare node combinations for different combination correlation coefficient. The number of rare nodes in the c880 circuit is 84. Assuming that the hardware Trojan attacker uses two nodes of the 84 rare nodes as the trigger signal to implement the combinational hardware Trojan, there are actually 13944 possible combinations. Using ACTS tool can successfully compress the search space into 13 combinations, reducing it to 0.093%. The rare combination array is RCA (13;2,84,2) in this case. As the combination correlation coefficient $t$ increases, the effect of compression becomes more obvious. Therefore, for detecting all rare node combinations, RCA can greatly reduce the range of detecting rare node combinations.

Test Vector Generation Algorithm

In order to find the test vector set that can activate the rare combinatorial array RCA, the most intuitive method is to search for the test vector of each combination by using random vectors. However, as the number of rare nodes in the circuit increases, the random number set required by this method also increases dramatically, and all combinations cannot be activated as soon as possible. Therefore, this paper considers the following methods to reduce the scope of the search.

The goal of the algorithm is to generate test vectors as soon as possible to activate all combinations at least once in the rare combination array RCA, by looking for the candidate vector set which can activated the largest number of combinations each time until the RCA is completely covered. The following is the pseudo-code for this algorithm.
The following is a description of the parameters of the algorithm. The input $N$ is represented as all rare nodes in the circuit, $m$ is represented as the number of rare nodes $N$ in the circuit, $Z$ represents the selected rare node combination array RCA, and the output $T(t,s)$ is the generated test set, where $s$ is the length of the test vector set. This algorithm is mainly based on the idea of loop iteration. The most important part is how to cover as many combinations of rare arrays as possible in one loop. Each cycle is divided into the following steps according to the structure of the algorithm:

The goal of lines 5-9 of the algorithm is to search for the rare node $N_c$ with the lowest transition rate. Through experiments, it is found that when the transition probability of a rare node is low, the range of vectors that activate it is generally small, so it can be found faster than other rare nodes. If the selected rare node has a large range of vectors, we can proceed directly to the next cycle. The 10-11th line of the algorithm is to generate the test vector $T_c$ which can activate the lowest transition probability $N_c$. The candidate vectors which activate the rare node is generated by the ATPG algorithm and we randomly select $n$ test vectors from it. In order to find $T_c$, the idea of backtracking is used to infer the input from the node to determine certain bits of the test vector that activates the rare node. When the value of $n$ is larger, more rare node combinations may be covered. However, as the value of $n$ increases, the range of candidate vectors generated also increases. The 13th to 17th lines of the algorithm screen out the test vector with a Hamming distance of 0, by which the RCA is successfully activated. The vector $T_j[n]$ that successfully activates a combination is placed in the $T(t,s)$ set. At the end of the loop, the active combination $z$ is removed and the activated node $N_c$ is removed from $N$, and then the next cycle is performed until all the combinations $Z$ in the RCA are activated. When $Z$ is an empty set, the cycle is stopped, and the vector set $T(t,s)$ is the final test vector set at this time. If traversed all the rare value, we can consider the vectors that activate the normal values of the nodes. Compared with the selection of test vector sets from random vectors, this method can greatly compress the candidate test vector set and shorten the time for generating test vectors.

**Experiments**

In this paper, an experimental verification of a combinational hardware Trojan detection method based on rare nodes is performed. In order to compare with the previous experiments, a random
node combination is randomly selected the ISCAS as original circuit, and finally the paper gives experimental results and analysis with the OERCA detection method.

The Design of the Circuit under Test

The hardware Trojan targeted by this paper is triggered by a combination of rare nodes. As shown in Figure 3, an example of a combinational hardware Trojan is given. In Figure 3, $W_1 \sim W_8$ indicate the selected rare nodes, the dashed box indicates the inserted three hardware Trojan triggers, and $S_1$, $S_2$, and $S_3$ indicate the nodes that the hardware Trojan wants to change. The hardware Trojan designed in this paper is mainly divided into two parts: the trigger and the payload. For the trigger, the activation lengths $l$ of the three hardware Trojans are 2, 3, 4. Except that the rare values of $W_1$ and $W_5$ are 0, rare values of other rare nodes are 1, and the hardware Trojan can only be triggered when the input value of 1 is activated simultaneously. For example, the first Trojan trigger is connected to the gate through $W_1$, $W_3$, and $W_4$. The Trojan can only be triggered when $W_3$ and $W_4$ trigger their rare value 1 and $W_1$ triggers its normal value 1. The design of payload needs to consider whether the inserted node can change the circuit function. Generally, the effective data path and control path are set. When the hardware Trojan is triggered effectively, the circuit output can be changed immediately.

In this experiment, the activation length $l$ of each Trojan inserted in the ISCAS combinational circuit is between 2-4. Hardware Trojan designers configure a Trojan trigger by selecting different rare nodes and their combined values. And they select the relevant node that affects the circuit function as the connection point of the payload in the circuit.

![Figure 3. The example of combinational hardware Trojan](image)

Experimental Results

In this paper, two experiments are carried out for the proposed hardware Trojan detection method. The first is to apply the test vector generation algorithm proposed to test vector sets for different combination correlation coefficient $t$ to compare the search efficiency of the proposed method with the random vector method. The second is to detect the circuit of the hardware Trojan whose activation length is 2, 3, 4 and apply the proposed method to generate the required test vector set. The number of test vector set was compared with the results of the literatures [5] and [6].
In this paper, a rare node combination array (RCA) is generated for rare nodes in the c880 circuit. Figure 4 shows that 3000 random vectors and candidate vectors recommended in this paper are used for simulation, and the number of candidate vectors activating successfully the combination of RCA when the combination correlation coefficient $t$ is 2-6. When $t$ is 3, the number of candidate vectors in this paper increases to 3.5 times compared to the number of combinations that the random vector can activate, and when $t$ is 6, the number of combinations of candidate vectors recommended in this paper is 3.2 times more than that of the random vector. This shows that the test vector generation method in this paper can greatly speed up the search time and reduce the search for candidate vectors. And in the case of $t$=2, the candidate vector of this paper can successfully activate two rare combinations, and 3000 random vectors are not activated successfully.

Table 2 shows the number N of test vectors obtained by using the OERCA hardware Trojan detection method under the assumption that the combination correlation coefficient $t$ is 2, 3, and 4. In this paper, the Trojans with different active Trojan activation lengths $l$ are detected by different test vectors generated by the combination correlation coefficient $t$. The results obtained from the test vector set with the combination correlation coefficient $t$ of 2, 3, and 4 in Table 2 are compared with the results obtained using previous methods [5] and [6], respectively. It can be found that in the case of ensuring the activation of a combinational hardware Trojan based on rare nodes, the detection method proposed in this paper greatly improves the pertinence of the test vectors and improve the efficiency of the test. For example, the detection of the c2670 circuit in this paper requires 174 test vectors, while the MERO method requires 9340. Because the method proposed in this paper is based on a combination of hardware nodes based on rare nodes, literature [5], [6] is a general detection method used to detect combinational and sequential Trojans, so the method of this paper has greatly improved the pertinence of test vector.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$t$</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>MERO</td>
</tr>
<tr>
<td>c2670</td>
<td>N</td>
<td>20</td>
<td>57</td>
<td>174</td>
<td>9340</td>
</tr>
<tr>
<td>c3540</td>
<td>N</td>
<td>22</td>
<td>65</td>
<td>206</td>
<td>15900</td>
</tr>
<tr>
<td>c7552</td>
<td>N</td>
<td>24</td>
<td>72</td>
<td>230</td>
<td>16358</td>
</tr>
<tr>
<td>c5315</td>
<td>N</td>
<td>22</td>
<td>70</td>
<td>222</td>
<td>15850</td>
</tr>
<tr>
<td>s35932</td>
<td>N</td>
<td>19</td>
<td>71</td>
<td>225</td>
<td>7343</td>
</tr>
</tbody>
</table>

Table 2 Test vector length compared to other methods.
Conclusion

The paper summarizes the combinational hardware Trojan model based on rare nodes and proposes the OERCA hardware Trojan detection method. Compared with the methods of MERO and GA+SAT, the method proposed in this paper only needs to activate these rare node combinations at least once, which improves the detection efficiency of hardware Trojan. Compared with the original hardware Trojans detection using combinatorial testing, the detection method of this paper expands the scope of the hardware Trojan detection that the applications of combinatorial testing based on the rare nodes inside the circuit. The experimental results show that the test vector set generated by the proposed algorithm can effectively trigger hardware Trojans, and the pertinency of test vector set is greatly improved. In future study, it will involve how to extend to the sequential Trojans detection to increase the scope of this detection method.

References


