IET Computers & Digital Techniques

Special Issue: Hardware-Assisted Techniques for Security and Protection of Consumer Electronics

IET Journals
The Institution of Engineering and Technology

Circuit enclaves susceptible to hardware Trojans insertion at gate-level designs

ISSN 1751-8601
Received on 30th June 2018
Accepted on 30th August 2018
E-First on 1st October 2018
doi: 10.1049/iet-cdt.2018.5108
www.ietdl.org

Seyed Mohammad Sebt[1], Ahmad Patooghy[2], Hakem Beitollahi[1] ✉, Michel Kinsy[2]

[1]*Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*
[2]*Electrical and Computer Engineering Department, Boston University, Boston, MA, USA*
✉ *E-mail: beitollahi@iust.ac.ir*

Abstract: A hardware Trojan (HT) is an extra circuitry inserted into a chip design with the malicious aim of functionality alteration, reliability degradation or secret information leakage. It is normally very hard to find HT activation signals since such signals are intended to activate upon occurring very rare conditions on specific nets of the infected circuit. A security engineer would have to search among thousands of gates and modules to make sure about the non-existence of design-time HTs in the circuit. The authors propose efficient net susceptibility metrics to significantly speedup functional-HT detection in gate-level digital designs. The proposed metrics perform a computationally low overhead analysis on the controllability and observability parameters of each net of the under HT-test circuit. Then, using a proposed net classifier method, a very low percentage of circuit nets is determined as HT trigger suspicious nets. To show practicality and detection accuracy of the proposed metrics, gate-level circuits of Trust-HUB benchmark suite are examined by the proposed metrics. Results confirm a 100% HT trigger detection with a low false positive as compared with previous metrics. More importantly, unlike previously proposed methods, the authors detection accuracy is totally independent of the switching probability of circuit inputs.

1 Introduction

Distributed and multi-stage integrated circuit (IC) manufacturing paradigm gains more attention nowadays with the aim of reducing manufacturing costs as well as time to market [1, 2]. In this paradigm, different stages of IC manufacturing, e.g. design integration, fabrication, testing, and packaging might be done by different companies [1]. Although this manufacturing trend drastically decreases the IC production costs, an adversary with malicious aims would have more chances to impact the chip manufacturing chain [2, 3]. This issue introduces a new class of vulnerabilities in modern chips known as hardware Trojans (HTs) [3]. An HT is any unauthorised modification in the design of a chip to either reach or ease reaching malicious goals [3]. Although all stages of the manufacturing chain are potentially vulnerable to HT insertion, this probability is higher at the design time because attackers mostly have easier access to IC design houses [4]. Third party intellectual properties which are mainly considered as trusted modules may contain unwanted circuitries and may be another source of HT insertion into designs [5].

An HT circuit usually consists of a trigger and a payload part in order to activate and perform intended purposes, respectively [3]. Most of the previously introduced HTs are internally triggered meaning that inputs of the trigger part are connected to some nets of the infected circuit [1]. Based on their design, under very rare conditions when the intended values appear on these nets, the trigger part generates an HT activation signal to enable the payload part [1]. This fires the main functionality of the HT which may be system reliability/stability reduction [6, 7], operational failure [6], secret information leakage [8].

Taking into account that current IC design houses are untrusted, a netlist generated by the physical synthesis of such design houses may not be secure [9–13]. To prevent infected designs from being fabricated or used in critical applications, different HT detection mechanisms with different approaches have been proposed. Some papers measure the physical characteristics of fabricated chips, e.g. derived current, critical path delay, power consumption. Then they perform side channel analysis (SCA) to detect possible HTs inserted into the chip. The main idea behind these methods is that any modification in the design or fabrication would affect the chip's physical characteristics [7, 14, 15]. By their nature, (i) SCA methods cannot be used to find HTs in an unfabricated IC, i.e. design stage, (ii) they have limited accuracy in detecting small HTs due to the process variation noise in nano-scale fabrication technologies [1, 3], (iii) in most cases, there is a need for having access to either a golden (HT-free) IC or several numbers of untrusted ICs.

Logic testing and verification methods could be also applied on the design net-list to find possible inserted HTs even before chip fabrication. Logic test-based HT-detection methods apply specific input patterns to the design in order to activate the HT circuit and observe its impacts on the output nets of the design. These methods try to excite rare switching nets, which are mainly considered as nets driving the HT trigger part [15]. However, (i) finding the smallest set of test vectors, and (ii) finding a test vector to excite multiple-trigger HTs are two challenging issues in the test-based HT detection methods [3]. Methods proposed in [10, 11] try to find unwanted circuitries by doing verification searches on the gate level netlist of the design. Although such methods have high detection accuracy, their time complexity makes them impractical even for moderate circuits [16].

The contributions of this study are the following items:

- We introduce a set of susceptibility assessment metrics for gate level circuits. The proposed metrics analyse circuit nets and obtain a net-list vulnerability map which helps security engineers quickly detect malicious circuitries in the design.
- We propose a classifying method to investigate the obtained vulnerable map and extract the most suspicious nets of the design. The sub-linear time complexity of the method makes the proposed metrics a practical solution, even for commercial designs with a large number of gates in their netlist.

The rest of this paper is structured as follows. Backgrounds and motivational experiments are presented in Section 2. The proposed HT susceptibility metrics and susceptibility assessment method are presented in Sections 4 and 3, respectively. Results of the experiments and comparisons of the proposed metrics are presented in Section 5 and finally, Section 6 concludes the paper.

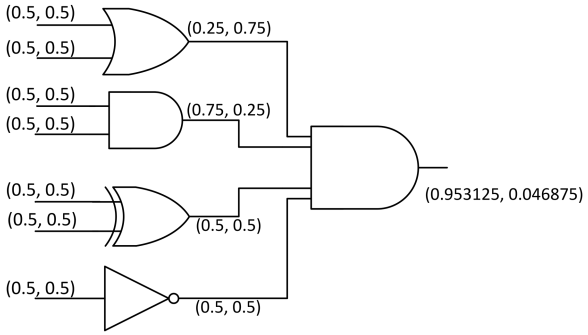

Fig. 1 *Computing SP (P_0, P_1) for nets of a sample circuit*

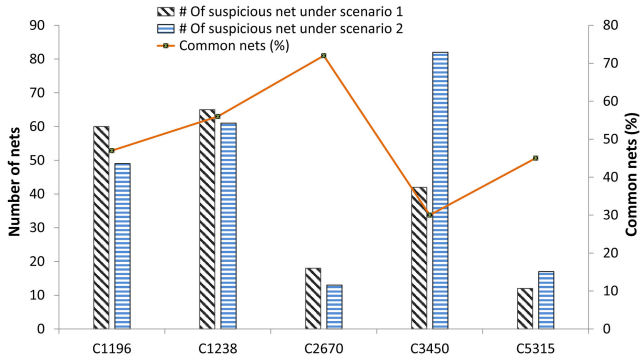

Fig. 2 *Detection variation of the SPA method under two scenarios*

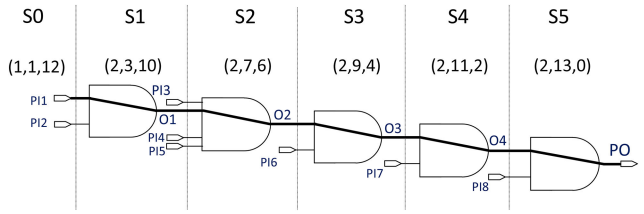

Fig. 3 *Computing CC0, CC1 and CO for a simple circuit nets*

2 Backgrounds and motivations

Since HJ need to be hidden most of the time, low switching nets of a design are strong candidates for HT insertion [1, 3, 9, 17]. Based on this, several papers have applied switching probability analysis (SPA) on nets of a circuit to flag nets with switching probability (SP) values lower than a predefined threshold as suspicious nets [9, 17–20]. The SP value of each circuit net is computed as $P_0 \times P_1$ where P_0 and P_1 are the probabilities of having 0 and 1 in that net, respectively. Fig. 1 shows how the SP value is computed for nets of a sample circuit. In this example, P_0 and P_1 for the output of each gate is computed based on gate type and its input probabilities. In almost all of the mentioned papers, P_0 and P_1 of the circuit's primary inputs are assumed to be equal to 0.5, which means that (i) input vectors are assumed to have a uniform distribution and (ii) primary inputs are assumed uncorrelated [9]. However, these assumptions are not valid in real world applications, due to data correlation. Consequently, HT analysis/detection under these unrealistic assumptions may shift the obtained results and the accuracy of made decisions. To show this, we have done a SPA on five combinational circuits of the ISCAS89 benchmark. For each circuit, a set of nets with bottom 3% of SPs are obtained for the following two scenarios: (i) $P_0 = P_1 = 0.5$ and (ii) P_0 = a random value between 0.22 and 0.45, $P_1 = 1 - P_0$.

According to the chart shown in Fig. 2, for circuit C1196, the set of suspicious nets contains 60 and 49 nets for the first and second scenarios, respectively. The difference between the two sets (39 nets resulting in 53% similarity) is high enough to show a significant variation in extracting suspicious nets under two scenarios. Other benchmark circuits show the same behaviour as can be seen in Fig. 2. As a result, some missed nets by SPA may have very low SPs when real application vectors are applied to the circuit. This can help an intelligent adversary to insert HT triggers in nets which are missed by the SPA method.

To overcome the inaccuracy of SP-based analysis, Waksman *et al*. [10] proposed a tool named FANCI which uses a control value (CV) metric to represent the impact of input nets of a combinational module on its outputs. The CV of an output net is computed by counting the number of output flips in the truth table of the module when only one input is changing and all other inputs are fixed. Using the CV metric, an output net is marked as a suspicious net if the average CV of all its inputs is lower than a predefined threshold. In fact, the FANCI tool checks whether the module output is loosely coupled to the inputs or not. Since the CV computation process is time consuming, computing CV for a subset of the truth table is proposed in the study, which obviously reduces the detection accuracy [10].

Zhang *et al*. [11] use a non-SP method to detect potential HT trigger nets. They have presented a tool named VeriTrust with the main idea of finding redundant inputs in the combinational logic cone of an output net. Functional verification tests are used to detect unactivated inputs in the form of a sum of products (SOP)/ products of sum (POS). The extracted SOP/POS forms are then analysed to find redundant inputs. Due to the limitations of random verification testing, there might be a large number of unactivated inputs in the circuit which increase the computational overhead of the VeriTrust method. Both FANCI and VeriTrust assume that HT trigger signals are generated by the combinational logic within one sequential stage of the circuit. Researchers in [4] have used this limitation to design HTs which cannot be detected by FANCI and VeriTrust.

Test-based HT-detection methods are based on two important parameters, namely controllability and observability [21]. These two parameters indicate how hard it is to set up a given circuit net to a given value (controllability) and subsequently observe its effects on circuit outputs (observability). SCOAP as a famous testability analysis [22] presents three key metrics, 0-combinational controllability (CC0), 1-combinational controllability (CC1) and combinational observability (CO) for each circuit net. To compute CC0/CC1 for circuit nets, CC0 and CC1 of circuit primary inputs are set to 1. For the output of each gate, CC0 and CC1 are computed based on the gate type and CC0/CC1 of the gate inputs. As an example, in order to set signal O (output of an n-input OR gate) to 0, all inputs of the gate should be set to 0 which means that $\mathrm{CC0}(O) = \sum_{i=1}^{n} \mathrm{CC0}(\text{input } i) + 1$. To compute CO of the circuit nets, CC0 and CC1 parameters of all circuit nets should be computed first, then CO of circuit primary outputs are set to 0, as their values could be observed with the lowest effort. CO of a gate input is computed using CO of the gate output and CC0/CC1 of other inputs of the gate. As an example, to observe the effect of changing input j of an n-input AND gate, all other inputs should be set to 1, i.e. $\mathrm{CO}(\text{input } j) = \mathrm{CO}(O) + \sum_{i=1, i\neq j}^{n} \mathrm{CC1}(\text{input } i) + 1$. In Fig. 3, a simple circuit with eight inputs and one output is illustrated. The triplets written below the stage names are (CC0, CC1, CO) of nets PI1, O1, O2, O3, O4 and PO, respectively.

Researchers in [23, 24] have proposed HT detection metrics based on SCOAP parameters. Their main objective is to find very hard to test nets of a circuit by investigating nets with high CC0, CC1 and CO values. Salmani and Tehranipoor [23] have stated that circuit nets with low $(\sqrt{\mathrm{CC0}^2 + \mathrm{CC1}^2} \times \mathrm{CO})^{-1}$ are more suspected to be an HT trigger net, since such nets need to have high CC0, CC1 and CO values. With some modifications and the same reasoning, Salmani [24] presented a metric named $|\langle \mathrm{CC}; \mathrm{CO} \rangle|$ which is defined as $\sqrt{\mathrm{CC0}^2 + \mathrm{CC1}^2 + \mathrm{CO}^2}$. Authors claimed that circuit nets with very high $|\langle \mathrm{CC}; \mathrm{CO} \rangle|$ values are very hard to test and subsequently suspected to be related to an HT trigger circuitry.

In the next, we will show that how metrics of [23, 24] may mislead us to a set of nets which are not actually good HT trigger candidate nets from the attacker point of view. To check this, we have done a set of testability experiments on our previously used benchmark circuits. CC0, CC1 and CO parameters of four ISCAS-89 benchmark circuits are computed to find nets with the lowest/highest metrics of papers [23, 24], respectively. Beside, a set of 100K pseudo-random input vectors are applied to the circuits

and the number of transitions (NoT), from $0 \rightarrow 1$ and $1 \rightarrow 0$ is counted for each net of benchmark circuits. From each circuit, four nets that have the lowest/highest metrics of papers [23, 24] are selected and shown in Table 1. For each circuit, a parameter named AvgNoT is calculated twice (i) average NoT of ten nets with the lowest $(\sqrt{CC0^2 + CC1^2} \times CO)^{-1}$ and (ii) average NoT of ten nets with the highest $|\langle CC; CO \rangle|$. As an example, average NoT of ten nets with a minimum $(\sqrt{CC0^2 + CC1^2} \times CO)^{-1}$ value for circuit C5315 is 32,765.4. For this circuit, average NoT of ten nets with the maximum $|\langle CC; CO \rangle|$ is 20,277.3. It can be clearly seen that the proposed metrics of [23, 24] do not guarantee to find low switching activity nets of the circuit.

3 Proposed HT susceptibility metrics

As discussed in Section 2, raw SCOAP parameters do not give thorough information about switching activity of circuit nets. Instead, they should be processed to lead us in finding HT trigger nets. In this section, we propose two innovative SCOAP-based metrics to efficiently find HT trigger nets in the gate-level net lists.

To find a set of suspicious nets in a gate-level design, we follow the most important feature [9, 14] of HJs, i.e. a very low rate of activity in time. This feature, in turn, needs to have a very low activity HT trigger nets in the design. A low activity net Net_i has the switching rate of $\alpha = (S_i/2^n)$ much lower than other nets of the circuit, where n is the number of circuit inputs, 2^n is the total number of input patterns, and S_i is the NoT of Net_i with respect to all 2^n applied input patterns. We know that a very low activity net tends to always keep the same logic state, i.e. '1' or '0'. This means that such a net would have either low CC0 and high CC1 or high CC0 and low CC1. We conclude this as the main feature of a low switching net is that the difference between its CC0 and CC1 values is relatively high. To show how much this conclusion is valid, we setup an experiment on the same set of benchmark circuits used in the previous experiments. Fig. 4*a* illustrates how the $|CC1 - CC0|$ value is correlated with the switching activity of nets in four benchmark circuits. In this experiment, we applied the same 100K pseudo-random input vectors. The net switching activity is calculated as $\alpha = (\text{NoT}/\text{number of test vectors})$. Then, nets are categorised as (i) high switching nets, with $\alpha > 0.4$, (ii) medium switching nets, with $0.2 < \alpha < 0.3$ and (iii) low switching nets, with $\alpha < 0.05$. The average $|CC1 - CC0|$ value for nets of each category is calculated and shown in Fig. 4*a*. As an example, the average $|CC1 - CC0|$ value for high switching nets of circuit C7552 is approximately eight times the average $|CC1 - CC0|$ value of medium switching nets. The same notable difference can be seen for other examined circuits in Fig. 4*a*. This observation confirms that a low switching HT trigger net could be identified with a high difference between its CC1 and CC0 values. Accordingly, we define our first HT trigger susceptibility parameter (HTS1) for Net_i as:

$$\text{HTS1}(\text{Net}_i) = \frac{|\text{CC1}(\text{Net}_i) - \text{CC0}(\text{Net}_i)|}{\text{Max}[\text{CC1}(\text{Net}_i), \text{CC0}(\text{Net}_i)]}. \quad (1)$$

The HTS1 parameter varies in the interval [0,1). Its minimum value occurs when $\text{CC1}(\text{Net}_i) = \text{CC0}(\text{Net}_i)$ indicating that Net_i is a high activity net. For very low switching nets, $|CC1 - CC0|$ is relatively high and HTS1 approaches 1.

We can still do more pruning in finding HT trigger candidate nets. Suppose a net with a high $|CC1 - CC0|$ value located at either (i) output port of a gate with a high number of inputs, or (ii) output port of a consecutive chain of similar 0/1-contorallable gates. Such a net would have a high $|CC1 - CC0|$ value (as Fig. 3 shows for net O2). Although this net has a low switching activity compared to its neighbouring nets, changing its value is not hard enough for being an HT trigger net. To disregard such nets from our suspicious set of nets, we take into account that an HT trigger net is expected to be very hard to control but not so hard to observe. Its switching activity should be very low but when switched, there should be a way for it through the HT payload to affect circuit outputs. In fact, we believe that having a high HTS1 value is a necessary condition for an HT trigger net but it is not sufficient. To make the condition sufficient, a potential HT trigger net should have a high HTS1 and at the same time relatively low observability. Based on this discussion, we define the observability to controllability ratio (OCR) as

$$\text{OCR}(\text{Net}_i) = \frac{\text{CO}(\text{Net}_i)}{\text{CC1}(\text{Net}_i) + \text{CC0}(\text{Net}_i)}, \quad (2)$$

which gives us an idea about the difficulty of observing a net compared to its controllablity. OCR gives a better intuition for nets with HTS1 close to 1 where OCR could be approximated to CO/CC1 or CO/CC0. Our investigations show a direct relation between OCR and the switching activity of the circuit nets. To see this, a set of simulations is done by applying 100K input vectors to four ISCAS89 benchmark circuits. Nets with $\text{HTS1} > 0.9$ are selected and their average OCR values are calculated. Selected nets are then classified into two groups based on their switching activities. Fig. 4*b* confirms our analysis regarding the relation of the OCR value with the switching activity of nets. A notable difference between average OCR values of the high HTS1 nets with $\alpha < 0.01$ and high HTS1 nets with $\alpha > 0.05$ can be clearly seen. As an example, for circuit C7552, the average OCR value of nets with $\alpha < 0.01$ is approximately six times smaller than the average OCR of nets with $\alpha > 0.05$. This means that the switching activity of nets that have a high HTS1 value and their OCR value is smaller than the average is significantly lower than other nets.

Unlike the HTS1 parameter, the OCR parameter is not bounded and it belongs to $[0,\infty)$. To have a bounded HT trigger metric which also has characteristics of the OCR parameter, our second HT susceptibility parameter, HTS2 is defined as follows:

$$\text{HTS2}(\text{Net}_i) = \frac{1}{1 + \text{OCR}(\text{Net}_i)}. \quad (3)$$

Nets with a very high observability value are usually close to the circuit primary inputs and have a HTS2 value close to 0. We discard them from the set of HT trigger susceptible nets.

Our observations clarify that nets with high HTS1 values are low switching as compared with other nets in their proximity. Among these nets, those with lower OCR, i.e. higher HTS2, are more suspected to be an HT trigger net since their switching activity is lower. The set of HT trigger susceptible nets specified by the mentioned conditions may contain a large number of nets when examining very large or commercial circuits. Accordingly, an appropriate classification method is proposed in the next section to help assess the susceptibility of the circuit nets faster using the HTS metrics.

4 Proposed net classifier method

The proposed HTS parameters determine a set of suspicious nets in the circuit under HT-test. Although the obtained nets are all suspected to be an HT trigger net, they have different susceptibility levels based on their specific properties such as logical depth and switching activity. To distinguish the most susceptible nets in large circuits, a method is presented in this section which is called hardware Trojan susceptibility assessing (HTSA). The proposed HTSA method includes two major phases as illustrated in Fig. 5.

4.1 Phase I: net pruning by HT metrics

This phase takes the gate-level netlist of a circuit under HT-test as input. If the circuit contains sequential elements, e.g. latches or flit flops, it is converted to the full-scan mode first. Then, CC0, CC1, CO, HTS1, and HTS2 parameters are calculated for all nets of the circuit under HT-test. The purpose of the first phase is to analyse all circuit nets in order to extract a set of hard to excite nets which are suspected to be HT trigger nets. To do this, we define the following two pruning filters to limit the number of suspicious nets in the circuit under HT-test.

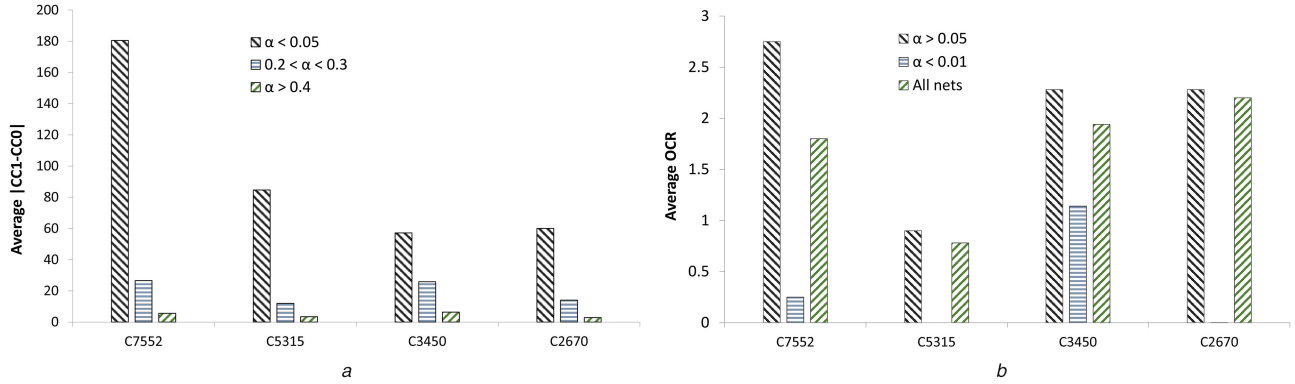

Fig. 4 *How* $|CC1 - CC0|$ *and OCR scatter nets on different* α *values*
(a) $|CC1 - CC0|$ versus α, ***(b)*** OCR versus α for nets with HTS1 > 0.9

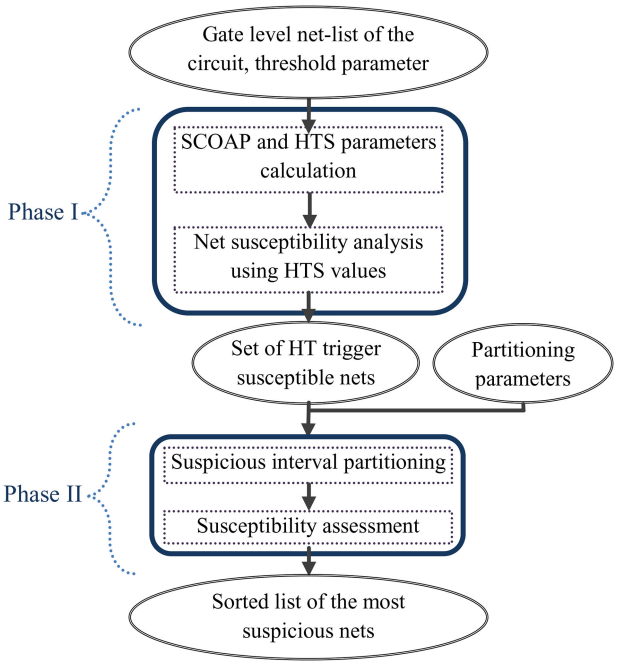

Fig. 5 *Phases of the proposed HTSA method*

Filter I) Nets with HTS1 > $HTS1_{th}$ are extracted from the circuit. These nets have lower switching activity than their neighbouring nets. To discuss the $HTS1_{th}$ parameter, assume $HTS1_{th} \rightarrow 1$, this causes the set of nets to shrink but it also increases the probability of missing inserted HT trigger nets. To find an appropriate value for the $HTS1_{th}$ parameter, we have done an experiment on five circuits of the ISCAS89 and ITC99 benchmarks. Sequential circuits are first converted to the full-scan mode and 100K pseudo-random input vectors are then applied to the circuits. After calculating a number of transitions and the HTS1 value for each net, the average HTS1 value for nets with $\alpha < 0.1$ and $\alpha < 0.01$ are computed for each circuit. As shown in Table 2, the average HTS1 value of nets with $\alpha < 0.01$ in all circuits is close to 0.9. So, we used $HTS1_{th} = 0.9$ in our first defined pruning filter.

Filter II) Among nets that passed Filter I, nets with relatively high HTS2 would pass the second filter. As discussed earlier in Section 4, nets with higher HTS2 have a higher chance to be low-switching in real practice. To consider this, we select nets satisfying $\text{Max}(0.5, HTS2_{avg}) \leq HTS2 < 1$ as nets that pass the second pruning filter. In this inequality, $HTS2_{avg}$ is the average HTS2 value of those nets that passed the first filter. It should be noted that although nets with HTS1 > $HTS1_{th}$ and HTS2 = 1 might be very low switching, such nets are not suspected to be an HT trigger. The reason is that the effect of their excitation could be easily seen on the observable points of the circuit which causes the inserted HT to be detected as soon as it is triggered. A C++ program is developed which takes a gate level net-list of a circuit as an input and calculates SCOAP and HTS parameters for each net of the input

Table 1 Evaluation of the proposed metrics in [23, 24]

Circuit, maximum NoT	Selected suspicious nets based on [23]				Selected suspicious nets based on [24]			
	Net name	Metric value	NoT	Avg. NoT	Net name	Metric value	NoT	Avg. NoT
C7752, 50,400	10,067	2.67×10^{-5}	37,769	82,448.8	9618	506.6	5597	5317.818
	10,239	2.67×10^{-5}	37,467		9820	472.6	50,046	
	9775	2.61×10^{-5}	615		10,577	1027.9	813	
	9408	1.36×10^{-5}	603		887	1040	679	
C5315, 50,431	6805	9.20×10^{-5}	50,074	32,765.4	7631	224.1	12,797	20,277.3
	6930	8.43×10^{-5}	50,082		6131	225.8	21,767	
	7527	5.68×10^{-5}	50,110		1156	226.06	11,807	
	7528	5.64×10^{-5}	49,998		7504	226.4	409	
C3450, 50,545	5348	1.60×10^{-5}	33,495	43,496.6	5117	524.1	49,466	45,330.9
	5349	1.60×10^{-5}	33,495		5139	524.1	49,909	
	5354	1.55×10^{-5}	36,561		5159	531.2	49,466	
	5355	1.55×10^{-5}	36,561		5182	531.2	49,909	
C2670, 50,534	3852	2.37×10^{-5}	49,680	15,414.4	2213	572.1	49,880	41,783.5
	3817	2.31×10^{-5}	20,633		3531	574.1	50,006	
	3838	2.77–05	8897		3599	574.1	50,006	
	3734	2.98×10^{-5}	1185		2376	574.2	49,880	

circuit. The program returns a set of nets that satisfy both the above defined filters.

4.2 Phase II: net classifier method

The second phase of the classifier method tries to partition nets that passed the first and second filter based on their HTS2 similarity. The method divides the input HTS2 interval, i.e. $[\mathrm{Max}(0.5, \mathrm{HTS2}_{\mathrm{avg}}), 1)$ to multiple sub-intervals and discards those sub-intervals containing the highest number of nets. The basic idea behind this method is that nets with unique *HTS*2 values, i.e. there are no other nets with the same HTS2 value, are the most suspicious nets. The reason is that in a large circuit such as a processor, there is a high number of array-style nets which are used to interconnect different structural units of the circuit, i.e. these nets carry controlling/data signals with a width of more than one bit. Many of these nets are originated from and destined to the same units of the circuit which causes their testability parameters to be very close. So, even if a group of array-style nets passes the defined filters, they are not good candidates for HT insertion. In fact, HT trigger circuitries are inserted at more unique nets of the circuit to ensure very low switching activity for HT trigger net. This causes an HT trigger net to have a semi-unique HTS2 value with a large number of other nets of the circuit.

To validate this, we setup an experiment as follows. The first phase of the HTSA method is applied to HT free versions of four large circuits of the Trust-HUB site [25]. Then, for each circuit, the set of nets passing two defined filters are examined to find a number of nets in which their HTS2 values are unique. Results which are presented in Table 3 show the very high similarity between nets of the circuits according to their HTS2 values; only a very low (2, 1, 2, 29%) number of nets have unique HTS2. Only in circuit s38584, 29% of nets have unique HTS2 values, other benchmarks show at most 2% uniqueness.

Given the above analysis, in the second phase of the HTSA, the following steps are performed:

- Passed nets are sorted in ascending order of their HTS2 value to find the minimum HTS2 value which is called Min_HTS2.
- The HTS2 interval [Min_HTS2, 1) is partitioned to K sub-intervals with equal length l as $[\mathrm{Min\_HTS2}, \mathrm{Min\_HTS2} + l], [\mathrm{Min\_HTS2} + l, \mathrm{Min\_HTS2} + 2l], \ldots, [\mathrm{Min\_HTS2} + \mathrm{K}l, 1)$, which would be a total of K sub-intervals.
- The obtained sub-intervals are sorted in the ascending order of their net number.
- Nets which are unique in their sub-intervals (normally such nets are found in single-net intervals) are then extracted and considered as the final set of most suspicious nets.
- The extracted most suspicious nets are sorted in the descending order of their HTS2 values. The index of each net in the sorted list is called susceptibility index (SI), $\mathrm{SI} > 0$, which has an inverse relation with susceptibility of nets. Net with $\mathrm{SI} = 1$ is the most susceptible net of a circuit under HT-test according to the proposed HTSA method.

To maximise the susceptibility assessment accuracy, parameter l should be set to a value which maximises the number of sub-intervals containing only one net. The value of l should be set in a way that maximises the number of sub-intervals containing only one net. It should be noted that when $l \to 0$, the number of sub-intervals containing zero net as well as the computational overhead of the classifier method would grow. To prevent this, after truncating HTS2 values to five floating digits, the minimum distance between HTS2 values is considered as l.

Table 2 Effect of HTS1 value on the α

Circuit-benchmark name	Average HTS1 value of nets with	
	$\alpha < 0.1$	$\alpha < 0.01$
C7552-ISCAS89	0.86	0.89
C5315-ISCAS89	0.87	0.92
C3540-ISCAS89	0.85	0.91
B22s-ITC99	0.90	0.92
B12s-ITC99	0.78	0.87

Table 3 Net similarity in different HT-free Trust-HUB circuits

Circuit name	Total similar nets, %
WB_Conmax	98
VGA_LCD	99
EthernetMAC10GE	98
s38584	71

5 Experimental results

A C++ program along with a visual basic script is used to find the most suspicious nets for selected benchmarks. The program calculates the HTS metrics for gate-level circuits of the Trust-HUB benchmark (a set of widely used [10, 11, 23, 24] HT infected circuits). All calculations of the HTS parameters are done with $\mathrm{HTS1}_{\mathrm{th}} = 0.9$ under six-digit floating point precision. In these

Table 4 Results of applying the HTSA method on gate level circuits of Trust-HUB

Circuit name	N	$\lvert S\rvert$	Calculated l	$\lvert MS\rvert$	HT trigger $\in$ MS? (Y/N)	HTS1, HTS2 Of the HT trigger	SI of the HT trigger
S15850-T100	3306	105	0.00006	47	Y	0.9938, 0.9690	5
S35932-T100	8034	67	0.2	3	Y	0.9090, 0.8780	2
S35932-T200	8025	66	0.02	2	Y	0.9482, 0.8591	1
S38417-T100	9125	289	0.0003	14	Y	0.9970, 0.9964	1
S38417-T200	9128	289	0.0003	14	Y	0.9892, 0.9894	1
S38584-T300	13,537	268	0.0002	30	Y	0.9183, 0.8412	22
S38584-T200	13,375	133	0.0002	74	Y	0.9482, 0.8840	28
RS232-T1000	341	2	0.4	2	Y	0.9782, 0.9947	1
RS232-T1100	339	2	0.4	2	Y	0.9781, 0.9444	1
RS232-T1200	339	2	0.2	2	Y	0.9694, 0.9854	1
RS232-T1300	337	2	0.2	2	Y	0.9760, 0.9846	1
RS232-T1400	340	3	0.005	2	Y	0.9814, 0.9909	1
RS232-T1500	343	2	0.4	2	Y	0.9782, 0.9947	1
RS232-T1600	340	2	0.2	2	Y	0.9652, 0.9834	1
VGA_LCD-T100	149,475	12,294	0.0003	6	Y	0.9821, 0.9715	3
EthernetMAC10GE-T700	194,609	14,516	0.00002	98	Y	0.9534, 0.6818	77
EthernetMAC10GE-T710	194,795	14,516	0.00002	98	Y	0.9743, 0.7920	54
EthernetMAC10GE-T720	194,795	14,515	0.00002	97	Y	0.9811, 0.8372	48
EthernetMAC10GE-T730	194,795	14,515	0.00002	97	Y	0.9333, 0.6037	81

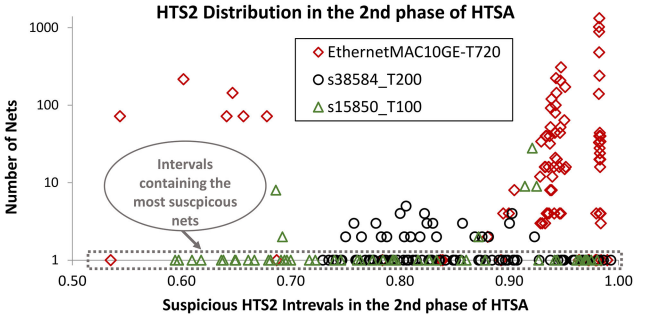

Fig. 6 *Distribution of nets based on defined HTS2 intervals shows two disjoint areas of (i) HTS2 intervals containing only one net, and (ii) HTS2 intervals with multiple nets*

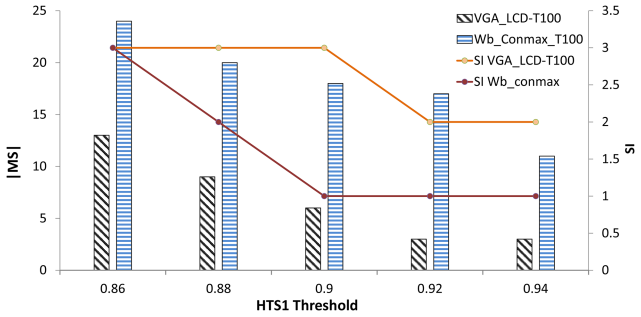

Fig. 7 *Effects of changing $HTS1_{th}$ on |MS| and the SI parameters*

benchmark circuits, we know where HTs are inserted, we will try to find a set of suspicious nets by the HTSA method and check if the real HT trigger net belongs to the set. Results of applying the proposed HTSA method to the gate level Trust-HUB benchmark circuits are presented in Table 4. In this table, N is the total number of circuit nets, S is the set of susceptible nets which are extracted by the first phase of the HTSA method, MS is the set of the most suspicions nets extracted by applying both phases of the HTSA method, and SI is the susceptibility index of the HT trigger in the set of the most suspicious nets, i.e. MS. The HTS1 and HTS2 parameters of the HT trigger net are also presented in the sixth column of the table.

As it can be seen in Table 4, in all circuits, the HT trigger net is a part of MS set with a remarkable susceptibility index. This confirms that the HTSA method can efficiently find real HT trigger nets. Results show that in many cases, the SI of the HT trigger net is one, meaning that the HTSA method finds the HT trigger net as the first-ranked suspicious nets of the circuit. For very large circuits like EthernetMAC10GE, the set of the most suspicious nets identified by the proposed HTSA method has only 97–98 members, meaning that at most 0.05% nets of the circuit should be examined to check if they are really HT triggers. For most of the benchmark circuits, this percentage is <0.03% which means that the search space is reduced to >99.97%. Using our HTSA metric prior to a test-based HT detection method yields great time savings.

To see the effectiveness of the proposed HTSA method, Fig. 6 shows the distribution of nets based on defined HTS2 intervals for three sample benchmarks. As it can be seen in this figure, the most suspicious nets look well distinguishable based on having unique HTS2 values.

To investigate the effects of $HTS1_{th}$ variations on the assessment accuracy of the HTSA method, the value of $HTS1_{th}$ is spanned from 0.86 to 0.94 and number of the most susceptible nets and also the SI of the HT trigger nets are obtained. Results for two large circuits of the Trust-HUB benchmark are shown in Fig. 7.

As it can be seen in this figure, the number of the most susceptible nets, i.e. |MS|, decreases as $HTS1_{th}$ parameter increases. Several low SI nets are removed from the MS as the $HTS1_{th}$ increases. This obviously causes the HT trigger nets to appear with lower SI. Although increasing the $HTS1_{th}$ value in the interval of 0.86–0.94 would drop some lower importance nets from the bottom of our list, $HTS1_{th}$ cannot be set as 1, or even cannot

Table 5 Comparison of the results obtained by the *SPA* and the *HTSA* methods

Circuit name	|MS|	|MTP|	SPA found the HT trigger (Y/N)?
EthernetMAC10GE-T700	98	479	N
EthernetMAC10GE-T710	98	479	N
EthernetMAC10GE-T720	97	479	N
EthernetMAC10GE-T730	97	479	N
VGA_LCD-T100	6	12,078	Y
S38584_T300	30	155	N
S38584_T200	74	69	Y
S35932_T100	3	1100	N
RS232_T1000	2	2	Y
RS232_T1100	2	2	Y
RS232_T1200	2	2	Y
RS232_T1300	2	2	Y
RS232_T1400	2	3	Y
RS232_T1500	2	2	Y
RS232_T1600	2	1	Y
S38417_T100	14	27	Y
S38417_T200	14	23	Y
S15850_T100	2	37	Y

approach 1, because this may remove the HT trigger net from the list as well.

As stated previously, several researchers have used different algorithms to generate test vectors and excite rare switching nets found by the SPA method. The disadvantages of SPA as a method for finding HT trigger nets are described in Section 2. To support our claim, we have done some experiments with the SPA method as well. Table 5 compares the results of the SPA and the proposed HTSA methods when applying to the seven different circuits of the Trust-HUB benchmark. In Table 5, MS is the set of the most suspicious nets extracted by the proposed HTSA method and MTP is the set of nets with the minimum SP. The third column of Table 5 indicates whether the HT trigger net is found among the nets with minimum SP or not. In large circuits, such as EthernetMAC10GE and VGA_LCD, results of two methods show a notable different output. The main reason is that the purpose of the SPA method is not to analyse the susceptibility of the nets but to find a set of rare switching nets which are assumed to be more suspected to be HT infected than other nets.

Due to the high number of nets in commercial circuits, it is important for an HT detection method to have a feasible time requirement. For example, methods such as FANCI and VeriTrust, which are not based on signal probability analysis are impractical due to their very high time complexity. The time complexity of FANCI and VeriTrust methods is greater than $O(2^n)$, where n is the number of circuit inputs [16, 24]. These methods need to analyse truth tables formed by applying random verification vectors to the circuit modules [10, 11]. Our proposed method, however, has a much lower time complexity compared to the mentioned methods. Time complexities of the first and second phases of the HTSA method are $O(N)$ and $O(N)$, respectively, where N is the number of circuit nets. It should be added that the time complexity of the second phase is a fraction of $O(N)$ since we apply the second phase on a subset of circuit nets not all of them. Fig. 8 shows the execution time of the HTSA method on circuits with a different number of gates and compares it with $O(N)$. As it can be seen in Fig. 8, the HTSA method is at least 3.6 times faster than $O(N)$. These results are obtained by running the HTSA method on an Intel Core i5 3.3 GHz CPU.

When comparing with SPA methods, the time complexity of the proposed HTSA method is approximately the same, i.e. $O(N)$ for the SPA and $O(N) + O(N) = O(N)$ for HTSA. So from the time view, the HTSA method could be an appropriate replacement for the SPA method. The proposed HTSA method could be easily combined with any logic testing-based HT detection method to help it find inserted HTs efficiently and more accurately.

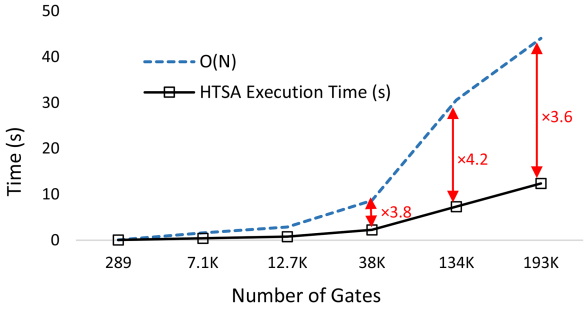

Fig. 8 *Execution time of HTSA method on circuits with different number of gates as compared to the linear time complexity*

6 Conclusions

This study has proposed innovative gate-level metrics which are inspired by testability parameters to aid test-based HT detection methods in finding HT trigger nets. The proposed metrics highlights the nets of a circuit under HT-test which are more likely to act as an HT trigger net. The set of determined suspicious nets are not dependent on the correlation of input vectors applied to the primary inputs of the circuit. A classifier method named HTSA is also presented which utilises an interval-based partitioning idea to further reduce the number of suspicious nets. This makes the proposed metrics a feasible security solution for commercial and large circuits having a huge number of gates. The proposed method is applied to gate-level circuits of Trust-HUB benchmark. The extracted sets of suspicious nets for all circuits are very small (at most 0.05% of nets of the circuit) and HT trigger nets are detected in all experiments.

7 References

[1] Mukhopadhyay, D., Chakraborty, R.S.: '*Hardware security: design, threats, and safeguards*' (Chapman & Hall/CRC, Boca Raton, USA, 2015, 1st edn.)

[2] Tehranipoor, M., Wang, C.: '*Introduction to hardware security and trust*' (Springer New York, New York, NY, 2012)

[3] Bhunia, S., Hsiao, M.S., Banga, M., *et al.*: 'Hardware Trojan attacks: threat analysis and countermeasures', *Proc. IEEE*, 2014, **102**, (8), pp. 1229–1247

[4] Zhang, J., Yuan, F., Xu, Q.: 'DeTrust: defeating hardware trust verification with stealthy implicitly-triggered hardware Trojans'. Proc. 2014 ACM SIGSAC Conf. on Computer and Communications Security, New York, NY, USA, 2014, pp. 153–166

[5] Sengupta, A., Bhadauria, S., Mohanty, S.P.: 'TL-HLS: methodology for low cost hardware Trojan security aware scheduling with optimal loop unrolling factor during high level synthesis', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2017, **36**, (4), pp. 655–668

[6] Bhasin, S., Danger, J.L., Guilley, S., *et al.*: 'Hardware Trojan horses in cryptographic IP cores'. 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), Santa Barbara, USA, 2013, pp. 15–29

[7] Francq, J., Frick, F.: 'Introduction to hardware Trojan detection methods'. Proc. 2015 Design, Automation & Test in Europe Conf. & Exhibition, EDA Consortium, Grenoble, France, 2015, pp. 770–775

[8] Liu, Y., Jin, Y., Makris, Y.: 'Hardware Trojans in wireless cryptographic ICs: silicon demonstration & detection method evaluation'. Proc. Int. Conf. on Computer-Aided Design, Piscataway, NJ, USA, 2013, pp. 399–404

[9] Salmani, H., Tehranipoor, M., Plusquellic, J.: 'A novel technique for improving hardware Trojan detection and reducing Trojan activation time', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2012, **20**, (1), pp. 112–125

[10] Waksman, A., Suozzo, M., Sethumadhavan, S.: 'FANCI: identification of stealthy malicious logic using Boolean functional analysis'. Proc. 2013 ACM SIGSAC Conf. on Computer & Communications Security, New York, NY, USA, 2013, pp. 697–708

[11] Zhang, J., Yuan, F., Wei, L., *et al.*: 'Veritrust: verification for hardware trust', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2015, **34**, (7), pp. 1148–1161

[12] Lesperance, N., Kulkarni, S., Cheng, K.-T.: 'Hardware Trojan detection using exhaustive testing of k-bit subspaces'. The 20th Asia and South Pacific Design Automation Conf., Tokyo, Japan, 2015, pp. 755–760

[13] Banga, M., Hsiao, M.S.: 'Trusted RTL: Trojan detection methodology in pre-silicon designs'. 2010 IEEE Int. Symp. on Hardware-Oriented Security and Trust (HOST), Anaheim, USA, 2010, pp. 56–59

[14] Ghandali, S., Becker, G.T., Holcomb, D., *et al.*: 'A design methodology for stealthy parametric Trojans and its application to bug attacks'. Int. Conf. on Cryptographic Hardware and Embedded Systems (CHES), Santa Barbara, USA, 17 August 2016, pp. 625–647

[15] Bhasin, S., Regazzoni, F.: 'A survey on hardware Trojan detection techniques'. 2015 IEEE Int. Symp. on Circuits and Systems (ISCAS), Lisbon, Portugal, 2015, pp. 2021–2024

[16] Haider, S.K., Jin, C., Ahmad, M., *et al.*: 'HaTCh: formal framework of hardware Trojan design and detection'. Tech. Rep, 943, Univ. Connecticut, Cryptol. ePrint Arch., 2014

[17] Chakraborty, R.S., Wolff, F.G., Paul, S., *et al.*: 'MERO: a statistical approach for hardware Trojan detection'. Cryptographic Hardware and Embedded Systems (CHES 2009), Lausanne, Switzerland, 2009, vol. 5747, pp. 396–410

[18] Bhunia, S., Abramovici, M., Agrawal, D., *et al.*: 'Protection against hardware Trojan attacks: towards a comprehensive solution', *IEEE Des. Test*, 2013, **30**, (3), pp. 6–17

[19] Zhou, B., Zhang, W., Thambipillai, S., *et al.*: 'Cost-efficient acceleration of hardware Trojan detection through fan-out cone analysis and weighted random pattern technique', *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2016, **35**, (5), pp. 792–805

[20] Saha, S., Chakraborty, R.S., Nuthakki, S.S., *et al.*: 'Improved test pattern generation for hardware Trojan detection using genetic algorithm and boolean satisfiability'. Cryptographic Hardware and Embedded Systems (CHES 2015), Saint Malo, France, 2015, vol. 9293, pp. 577–596

[21] Kantipudi, K.R.: 'Controllability and Observability' ELEC7250–001 VLSI Testing (Spring 2005), Instructor: Professor Vishwani D. Agrawal (2010)

[22] Goldstein, L.H., Thigpen, E.L.: 'SCOAP: sandia controllability/observability analysis program'. Proc. 17th Design Automation Conf., New York, NY, USA, 1980, pp. 190–196

[23] Salmani, H., Tehranipoor, M.M.: 'Vulnerability analysis of a circuit layout to hardware Trojan insertion', *IEEE Trans. Inf. Forensics Sec.*, 2016, **11**, (6), pp. 1214–1225

[24] Salmani, H.: 'COTD: reference-free hardware Trojan detection and recovery based on controllability and observability in gate-level netlist', *IEEE Trans. Inf. Forensics Sec.*, 2017, **12**, (2), pp. 338–350

[25] Salmani, H., Tehranipoor, M., Karri, R.: 'On design vulnerability analysis and trust benchmark development'. IEEE Int. Conf. on Computer Design (ICCD), Asheville, USA, 2013