# Hardware Trojan Detection Based on Correlated Path Delays in Defiance of Variations with Spatial Correlations

Fatma Nur Esirci, Alp Arslan Bayrakci
Department of Computer Engineering
Gebze Technical University, Turkey
e-mail: {fesirci, abayrakci}@gtu.edu.tr

***Abstract*—Hardware Trojan (HT) detection methods based on the side channel analysis deeply suffer from the process variations. In order to suppress the effect of the variations, we devise a method that smartly selects two highly correlated paths for each interconnect (edge) that is suspected to have an HT on it. First path is the shortest one passing through the suspected edge and the second one is a path that is highly correlated with the first one. Delay ratio of these paths avails the detection of the HT inserted circuits. Test results reveal that the method enables the detection of even the minimally invasive Trojans in spite of both inter and intra die variations with the spatial correlations.**

***Index Terms*—hardware security, path delay, malicious circuit**

## I. Introduction

Integrated circuit (IC) designers outsource their designs to private manufacturing companies mostly located in foreign countries due to mainly the economical reasons. Military and mission critical systems require a high level of trust and it must be certainly detected whether a malicious circuit called hardware Trojan (HT) is inserted by an adversary into their designs during the untrusted manufacturing stages. There are many different types of hardware Trojans that may change the functionality, leak confidential information like the keys, disable the circuit or decrease its performance.

Recently, there is a lot of research on HT creation, classification and detection methods [1], [2]. Nondestructive methods are cheap and time-efficient in contrast to destructive methods. They are mainly classified as logic testing and side channel analysis (SCA). Unlike the logic testing methods, SCA methods do not require the activation of Trojan circuit because they basically measure side channel signals like supply current or propagation delay in order to detect the effect of HT. However, SCA methods suffer from unavoidable, nonnegligible and increasing effects of process variations, which in fact exhibit spatial correlations. Path delay measurement, dimensionality reduction and delay fingerprinting are respectively used to detect Trojans by [3]. At speed delay calculation offers a method of shadow register insertion into the design in order to easily measure path delays [4]. Utilization of the shorter paths can magnify the delay effect of the Trojan. The shortest path through a possibly Trojan inserted edge (interconnect) is used for detection in [5]. But, it requires all chips coming from a lot to either have an inserted Trojan or be free from Trojans. Both path delay and leakage current are used for the characterization of gates in [6]. In order to get rid of the effect of process variations, technique in [7] uses supply current in conjunction with the maximum frequency. The change in current with respect to the change in maximum frequency results in a better detection performance as both are affected from variations similarly unless there is an inserted HT. SCA based methods need the Trojan circuit to be large enough to have a noticeable effect on the side channel signal. For power based methods, detection rates can be increased by regional power measurements [7], whereas for delay based methods, it can be increased by shorter path utilization [5].

In this paper, we combine the idea of multiple parameters in [7], utilization of the shortest paths in [5] and the stochastic gate delay model of [8] in a novel manner to produce a new HT detection methodology. The main contribution of the paper consists of a new technique to smartly select two highly correlated paths, one of which is the shortest path that passes through an interconnect suspected to have an HT on it. Delay ratio of these two paths reveals the existence of HT by hiding the effects of variations as both paths are affected very similar from the variations. During test stage, it only requires path delay measurements twice the number of suspected HT locations in the circuit. SCA methods are known to have two weak points: small sized Trojans and process variations. Therefore, we verify our method using a minimally invasive small sized HT (Fig. 1) and a realistic variation model considering both inter die variations and intra die variations with spatial correlations while variation amounts are taken from ITRS [9]. Proposed method targets the untrusted manufacturing stage and ignores the adversary involvement due to an untrusted CAD tool or IP usage. Inherently, our method, like any other delay based method, cannot detect any Trojan that does not affect any path delay by any means.

The paper is organized as follows. Section II gives some background, formulates the target problem and demonstrates the difficulty of the HT detection in the existence of the variations. Section III presents the main methodology and the path extraction algorithm devised in this paper to detect any HT insertion. Section IV displays the improvement achieved by the proposed method through the experiments.

978-3-9815370-8-6/17/$31.00 

## II. Background & Motivation

### A. Process Variations

Using an accurate variation model in evaluating HT detection methods based on side channel analysis is crucial as the main challenge is to distinguish the effect of inserted HT from the effect of process variations. A loose variation model may yield good simulation results, yet the actual performance may be much worse.

In modern process technologies, the intra die variation component is as important as the inter die variation and also exhibits spatial correlations, which must be modeled for accurate analysis. As a result of the spatial correlation, gates residing closer to each other have more correlated random parameters, whereas this correlation decreases while distance between the gates increases. In this paper, we adopt four level quad-tree model devised in [10] in order to imitate the spatially correlated random variables.

Random variable vector $X$ contains the values of all random parameters inside the circuit with an associated joint probability density function $f(X)$ that is constructed according to our variation model. Drawing a sample from $f(X)$ provides in a manner random parameter values of a manufactured chip.

### B. Problem Formulation

We convert the circuit netlist into a graph where each cell (gate) in the circuit corresponds to a node and each interconnect between two gates corresponds to an edge of the graph. An edge between nodes $n_p$ and $n_q$ is represented as $< n_p, n_q >$ and a path $P$ of k gates is shown as $< n_{g_1}, n_{g_2}, ...., n_{g_k} >$ where $n_{g_1}$ and $n_{g_k}$ are gates connected to a primary input and a primary output respectively. The delay of such a path is the summation of gate delays inside the path:

$$d(P,X) = \sum_{i=1}^{k} d(n_{g_i}, X) \tag{1}$$

where $d(P,X)$ is path delay and $d(n_{g_i}, X)$ is the delay of node $n_{g_i}$ with respect to sample $X$.

An edge that is suspected to have an HT inserted is called a *suspected edge*. Delay of a path passing through the suspected edge with an inserted HT can be written as:

$$d(\hat{P},X) = \sum_{i=1}^{k} d(n_{g_i}, X) + d(n_{load}, X) \tag{2}$$

where $\hat{P}$ is the path with an inserted HT and $n_{load}$ represents the payload node of HT. The aim of HT detection using delay based SCA is to distinguish the additional summation component in (2) from the noise of variations.

Without loss of generality, we use the smallest combinational type HT shown in Fig. 1. This HT consists of two gates: AND gate is used to trigger only when both of its inputs become 1 and XOR gate is used to invert the signal on circuit path when it is triggered. Payload gate is connected to an actual interconnect of the circuit shown by dashed line in the figure. Inputs of the AND gate must be smartly selected so that the HT can only be triggered by a 'rare event' and therefore, it cannot be detected by conventional logic testing. Of course, this is not a sophisticated HT, but it serves the purpose of augmenting the difficulty of detection. Normally, this type of Trojan, having very small impact on side signals, is the most challenging one for SCA methods. Other more sophisticated combinational or sequential Trojans are easier to detect by delay based SCA, because they have a larger payload than the one in Fig. 1.

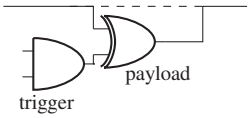


Fig. 1. *HT* circuitry used in the paper.

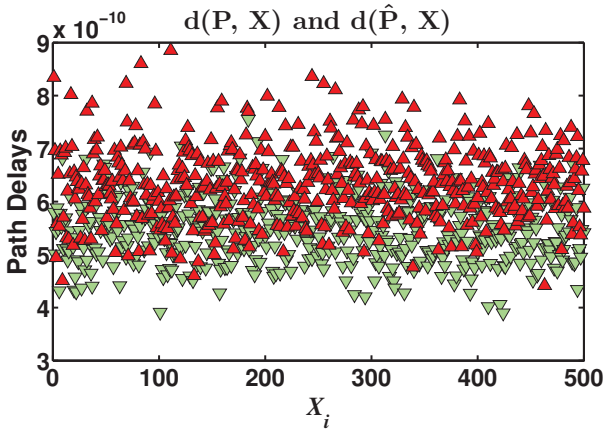


Fig. 2. Interference of variation and HT effects for a sample path from c3540. Red, up-triangles are samples (chips) with HT and green, down-triangles are samples without HT.

### C. Difficulty of the Problem

In order to observe the effect of HT on path delay in the presence of variations, 1000 sample points are drawn from $f(X)$ that is constructed according to our variation model explained in Section II-A. First, we randomly select an edge of c3540 circuit [11] as suspected edge. Then, for 500 of the 1000 samples, HT of Fig. 1 is inserted on the suspected edge in the circuit and for the other half, HT is not inserted. To increase the relative effect of the inserted HT on the path delay, shortest delay path passing through the suspected edge is used. Delay is obtained by transistor level Spice simulation for each of the 1000 samples. Delays of both the path without HT ($d(P,X_i)$ with green, down-triangles) and same path with inserted HT ($d(\hat{P},X_i)$ with red, up-triangles) are marked on the same plot shown in Fig. 2. The figure reveals that, although we utilize the shortest path passing through the suspected edge as devised in [5], the fluctuations of path delay due to process variations definitely hide the effect of HT and cause large detection errors.

## III. HT Detection Methodology

### A. Overview of the Path Extraction Algorithm and the Delay Ratio Computation

Suppressing the process variations is possible by using multiple parameters [7]. But for this method to work, the parameters must be affected from the variations in a similar way, so that the ratio of the parameters can hide the effect of the variations. Therefore, the more correlated the parameters are, the better their ratio hides the variations, which increases the significance of multiple parameter selection.

In this paper, we devise an efficient way to extract multiple "correlated" parameters (path delays) from the circuit, which is summarized by *extractPaths* algorithm. The algorithm repeats the same procedure for each *suspected edge* that is suspected to have an inserted Trojan in the circuit. Suspected edges in the circuit can be determined as it is done for determining sensitive paths in [12]. At worst case, all edges in the circuit can be targeted as suggested in [5], which increases the number of iterations to the number of edges in the circuit[1].

**Algorithm *extractPaths*** Extracts $P_{susp}$ and $P_{corr}$ paths

1: **for** each suspected edge $< n_s, n_t >$ **do**

{**Extraction of the *suspected path***}

2: Collect in $\pi_{fromPI}$ set, the shortest delay path starting from each primary input (PI) to $n_s$

3: Collect in $\pi_{toPO}$ set, the shortest delay path starting from $n_t$ to each primary output (PO)

4: $\pi_{susp} = \pi_{fromPI} \times \pi_{toPO}$ (cartesian product of two sets)

5: (*suspected path*) $P_{susp}$ = shortest path in $\pi_{susp}$

{**Extraction of the *correlated path***}

6: Collect in correlated path candidates set $\pi_{corr}$, the paths "near" $P_{susp}$

7: **for** each path $P_i$ in $\pi_{corr}$ **do**

8: $r_i = Corr(d^{SLE}(P_i, X), d^{SLE}(P_{susp}, X))$

9: **end for**

10: (*correlated path*) $P_{corr}$ = path with highest $r_i$

11: **end for**

*extractPaths* algorithm is composed of two main blocks, whose tasks are summarized below:

1) *Extraction of the Suspected Path (lines 2 to 5)*: There may be more than one path passing through the edge that is suspected to have an inserted HT on it. We collect in suspected path candidates set, the smallest delay path *passing through the suspected edge*, for each primary input-output pair. Among these candidates, we choose the one with the smallest delay so that relative effect of HT on path delay increases, which makes HT more detectable. We call this path *suspected path*, i.e. $P_{susp}$. Details of suspected path extraction is given in Section III-B.
2) *Extraction of the Correlated Path (lines 6 to 10)*: There may be an exponential number of paths in the circuit, some of which have a higher correlation with the suspected path ($P_{susp}$). Therefore, a method is required to rule out paths without inspecting all of them. Because of inherently existing spatial correlations, paths closer to the suspected path are expected to be more correlated with the suspected path. Exploiting this fact, we devise a method to collect paths that are "closer" to the suspected path into the *correlated path candidates set*. After the extraction of the correlated path candidates, the one having the highest correlation is assigned as the *correlated path*, i.e. $P_{corr}$. Details of the correlated path extraction is given in Section III-C.

After the suspected path for a suspected edge and the correlated path are extracted, delays of both paths are measured for each manufactured chip, i.e. for each sample point ($X$) drawn from joint probability function ($f(X)$). The ratio ($R(X)$) of these highly correlated path delays is taken as shown in (3) in order to reduce the variation noise. Our method suggests to use that ratio to classify chips according to the existence of Trojan inserted on the corresponding suspected edge.

$$R(X) = d(P_{susp}, X)/d(P_{corr}, X) \tag{3}$$

The two extracted paths for each suspected edge must be sensitizable in order to measure their delays. For this purpose, any sensitization criteria well known in the literature [13] can be employed. We use a satisfiability based static sensitization method in [14] to detect sensitizable paths. If one of the two paths, i.e. $P_{susp}$ and $P_{corr}$, is not sensitizable, then the next path from the respective set of candidates ($\pi_{susp}$ and $\pi_{corr}$) is picked instead. If the suspected path is changed, then the set of correlated path candidates should be reconstructed.

In order to classify the chips with zero error (no false negatives or positives), the resultant path delay ratio values of the HT inserted chips ($\hat{R}(X)$) must be at "sufficient distance" from the ratio values of the HT-free circuits ($R(X)$). Hence, variation of the ratio values due to the process variations can not hide the effect of HT. Using highly correlated paths decreases the amount of the "sufficient distance" by reducing the variation of the ratio values.

For instance, for exactly the same path, the same suspected edge and the same HT of Fig. 2, after *extractPaths* algorithm is applied and the path delay ratio for each sample (chip) is computed as shown in (3), the resultant path delay ratios for the 500 samples with HT ($\hat{R}(X)$) and 500 samples without HT ($R(X)$) can be seen on Fig. 3. It is obviously seen that $\hat{R}(X)$ values (red up-triangles) and $R(X)$ values (green down-triangles) can be easily distinguished, which means that HT can be detected with zero error without any need for a golden chip. Transformation from Fig. 2 to Fig. 3 summarizes the aim of the proposed method.

[1]Number of edges is $O(N^2)$, where $N$ is the number of nodes in the circuit. But it is practically not very far from $N$. For instance, for ISCAS'85 test circuits in this paper, the number of edges is always less than $2N$.

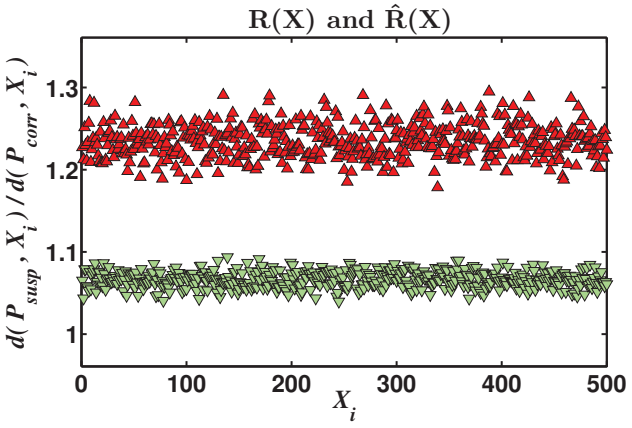


Fig. 3. The ratios of the paths extracted by *extractPaths* algorithm for the sample path from c3540. Ratio of samples with HT shown by red, up-triangles and without HT by green, down-triangles.

All path pairs ($P_{susp}$ and $P_{corr}$) for all suspected edges can be extracted by the algorithm just after the design stage ends without waiting for the manufacturing. During manufacturing test stage, the only procedure that must be performed is delay measurement of the extracted path pairs for all chips. Then, chips are classified according to the resultant path delay ratios.

### B. Extraction of the Suspected Path Candidates

Gate delays are required in extracting the suspected path candidates. Therefore, even before starting *extractPaths* algorithm, the delay of each gate ($n$) in the circuit ($d^{SLE}(n, X_{nom})$) is determined for only the nominal values of random parameters, i.e. $X_{nom}$. For this procedure, we use a fast gate delay model called Stochastic Logical Effort (SLE) [8], which is also able to catch the variation of delay with respect to varying circuit parameters.

Shortest path from each primary input to the first node of the suspected edge ($< n_s, n_t >$) is collected by calling Dijkstra for each primary input and storing into $\pi_{fromPI}$ set, the shortest delay path until $n_s$ (line 2). Second, Dijkstra is called once more for the second node of the suspected edge ($n_t$) in order to collect in $\pi_{toPO}$ set the shortest path from $n_t$ to each primary output (line 3). Last, the cartesian product of $\pi_{fromPI}$ and $\pi_{toPO}$ sets produces all paths from the primary inputs to the primary outputs, each of which passes through the suspected edge and is the shortest delay path for the corresponding $< primary\ input - primary\ output >$ pair. The shortest path in this set is also the shortest path passing through the suspected edge and is assigned as $P_{susp}$. Any path in this set can be used as $P_{susp}$ but using the shortest one increases the probability of detection. If the shortest path is not sensitizable or no corresponding correlated path candidate is found, then next path in the $\pi_{susp}$ set is assigned as $P_{susp}$.

### C. Extraction of the Correlated Path Candidates

Having $P_{susp}$, we need to collect the paths correlated with the suspected path as line 6 of *extractPaths* algorithm suggests.

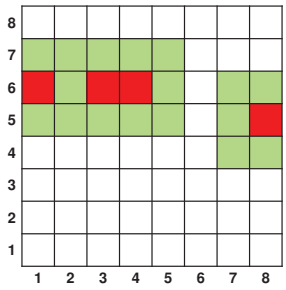


Fig. 4. The division of circuit layout and a sample suspected path, which consists of gates located at red (dark grey) grids. Adjacent grids are shown by green (light grey).

We make use of the fact that random parameters of the circuit devices are more correlated when the devices are located closer to each other as a result of spatial correlations. Consequently, search for a correlated path turns into searching for a path whose gates reside in a close vicinity of $P_{susp}$.

First, we divide final layout of the design into grids as demonstrated in Fig. 4. For instance, assume that the gates of the suspected path ($P_{susp}$), are located at grid locations (1,6), (3,6), (4,6), (8,5) as shown by red. We accept each path that has all its gates reside at a subset of red grids as a *correlated path candidate*. Due to the spatial correlations, gates of such a correlated path candidate must be affected from the variations similar to the gates of $P_{susp}$ residing at the same grids.

The procedure to extract the correlated path candidates is as follows: we first exclude the nodes (gates), which are not located at any one of the grids of $P_{susp}$ (red grids), from the graph. Then for the remaining part of the graph paths are stored in the set of correlated path candidates ($\pi_{corr}$). With this technique, there is a possibility that no correlated path candidate is found, because either the grid size is very small or the circuit has a small number of gates. In such a situation, we relax the constraint of residing at same grids with $P_{susp}$ and allow the adjacent grids. For our sample case, this corresponds to allowing both red and green grids in Fig. 4. Hence, there is a trade-off on the number of grids or divisions. When we decrease the grid size, gates at the same grid have much similar (correlated) random parameter values, which is better. However, the possibility of finding a correlated path candidate decreases as there are now less number of gates residing at the same grids.

After extracting the correlated path candidates, we compute the correlation coefficient of each candidate with the suspected path using *Corr* function. *Corr* function at line 8 computes the correlation coefficient per candidate using path delays computed for all sample points drawn according to the joint probability density function ($f(X)$). For each sample, delays of both paths are computed using fast SLE gate delay model. Resultant correlation coefficients are used to discover the most correlated path and assign it to $P_{corr}$. SLE may not give exact correlation coefficients but it is used to approach the actual ranking of the coefficients.

## IV. Results

Among all process parameters, the most significant ones are transistor channel length ($L_{eff}$) and threshold voltage ($V_t$) [15]. Effective channel length, $L_{eff}$, is assumed to have a $3\sigma/\mu$ ratio of 12% and threshold voltage, $V_t$, a $3\sigma/\mu$ ratio of 20% according to the ITRS report [9]. Half of the total variation is assumed to come from inter die variations and the other half from the intra die component [16]. Widely adopted ISCAS'85 benchmark circuits [11] are used in the experiments. Sensitization of the extracted suspected and correlated paths is tested by using a static sensitization criteria based satisfiability technique in [14]. Circuits are synthesized by NanGate 45nm Open Standard Cell Library [17]. Correlated path candidates are collected using the grid shown in Fig. 4.

In order to simulate 1000 manufactured chips, we draw 1000 samples according to $f(X)$ of our variation model [10] and apply precise transistor level Spice to simulate manufacturing test stage delay measurements. For each of nine ISCAS'85 circuits, we randomly assign three edges of the circuit graph as *suspected edges* (*SE1*, *SE2* and *SE3*), which means 27 different experiments. For each experiment, we inserted the HT in Fig. 1 onto each suspected edge for half of the thousand chips and the other half do not have any HT. For instance, $Exp(c432,SE2)$ represents the experiment on c432 circuit and for the second suspected edge. For each of 27 experiments, we extract both the suspected path ($P_{susp}$) and the correlated path ($P_{corr}$) using *extractPaths* algorithm explained in Section III. Ratio ($R(X)$) of (3) is computed by dividing the delay of $P_{susp}$ to the delay of $P_{corr}$ for each of the 1000 samples.

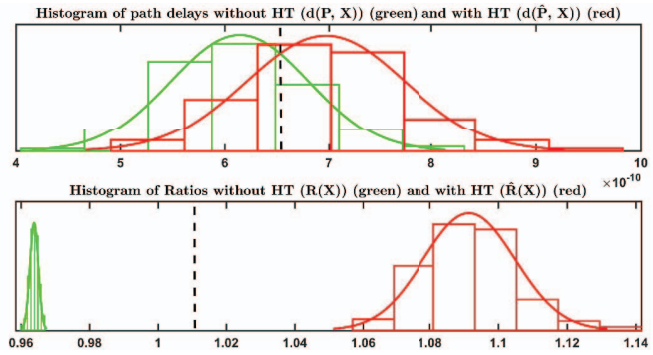


Fig. 5. The path delays histogram and ratios histogram for $Exp(c2670,SE3)$. Green corresponds to 500 HT-free samples and red corresponds to 500 samples that have inserted HT on the path.

Fig. 5, shows two histograms that belong to $Exp(c2670,SE3)$. Upper one shows histogram of the delay of $P_{susp}$ for a thousand samples, half of which have HT. It is the histogram version of Fig. 2. Although $P_{susp}$ is the shortest path passing through the suspected edge, the upper histogram shows how path delays with and without HT are mixed into each other. Using this path delay solely and being conservative in selecting secure chips—in order not to mark any chip with HT as secure—results in almost no secure chips although half of them are secure. Lower plot in Fig. 5

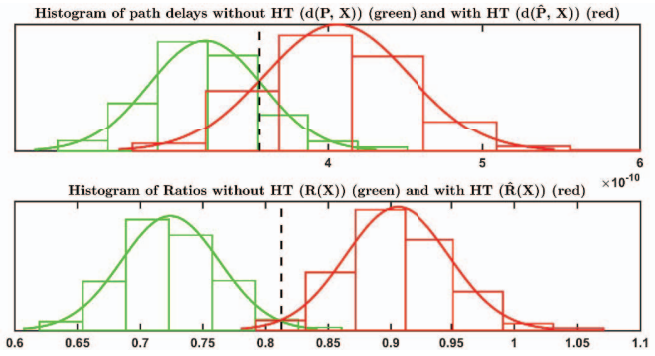


Fig. 6. The path delays histogram and ratios histogram for $Exp(c499,SE1)$. Green corresponds to 500 HT-free samples and red corresponds to 500 samples that have inserted HT on the path.

is the histogram version of Fig. 3. It shows the distribution of the path delay ratio computed by dividing the delays of $P_{susp}$ and $P_{corr}$. This lower histogram reveals that using $R(X)$ ratio definitely separates the HT-free samples from the HT inserted ones.

Vertical dashed line on the histogram plots represents the boundary for least error in the classification of the samples according to the existence of HT. For instance, for $Exp(c2670,SE3)$ in Fig. 5, if we use only the delay of $P_{susp}$ (upper histogram) to classify circuits, then the resultant number of misclassified samples—either false positive or false negative—is 289 out of 1000 samples. Whereas, if the ratio ($R(X)$) of the suspected and correlated path delays (lower histogram) is used for the classification, then all samples are accurately classified resulting in zero error. Very similar histogram plots are obtained in all experiments with the exception of $Exp(c499,SE1)$, whose histograms are shown by Fig. 6. For $Exp(c499,SE1)$, the number of misclassified samples is 17 although the path delay ratio is used for the classification. But even for this experiment, the number of misclassified samples becomes zero when we select another suspected path candidate from the $\pi_{susp}$ set than the shortest path. Correlation of the two extracted paths is estimated a priori by line 8 of the algorithm. Therefore, when all $r_i$ values are found to be small, it is better to change the initial suspected path choice. By this minor modification of the algorithm, $Exp(c499,SE1)$ results in no misclassified samples.

Table I summarizes all results for all 27 experiments. First column of the table is the number of gates in the suspected path. Second column is the number of paths in suspected path candidates set representing the number of suspected path alternatives. Third column is the number of correlated path candidates extracted by searching only the same grids with $P_{susp}$ and fourth column is the number of correlated path candidates extracted by allowing the adjacent grids as explained in Section III-C. Adjacent grid permission is given only when the number of correlated path candidates is small or the resultant correlation coefficients are low where other rows are shown by dash. Fifth column shows the number of the

TABLE I
RESULTS FOR ALL 27 EXPERIMENTS

| | | # of gates in $P_{susp}$ | size of $\pi_{susp}$ | size of $\pi_{corr}$ same grids | size of $\pi_{corr}$ adjacent grids | Miscl. samples wrt $P_{susp}$ (out of 1000 samples) | Miscl. samples wrt $R(X)$ (out of 1000 samples) |
|---|---|---|---|---|---|---|---|
| c432 | *SE1* | 4 | 54 | 2 | 11 | 13 | 0 |
| | *SE2* | 6 | 5 | 1 | 30 | 42 | 0 |
| | *SE3* | 3 | 180 | 1 | 22 | 9 | 0 |
| c499 | *SE1* | 7 | 28 | 3 | 18 | 151 | 17 |
| | *SE2* | 5 | 32 | 2 | 56 | 110 | 0 |
| | *SE3* | 5 | 40 | 1 | 2 | 145 | 0 |
| c880 | *SE1* | 10 | 6 | 4 | 135 | 123 | 0 |
| | *SE2* | 6 | 16 | 5 | 73 | 36 | 0 |
| | *SE3* | 6 | 26 | 2 | 3 | 44 | 0 |
| c1355 | *SE1* | 14 | 192 | 109 | - | 198 | 0 |
| | *SE2* | 4 | 105 | 7 | 25 | 12 | 0 |
| | *SE3* | 3 | 106 | 6 | 7 | 8 | 0 |
| c1908 | *SE1* | 5 | 3 | 3 | 15 | 66 | 0 |
| | *SE2* | 7 | 252 | 576 | - | 170 | 0 |
| | *SE3* | 15 | 25 | 26 | - | 196 | 0 |
| c2670 | *SE1* | 8 | 6 | 26 | - | 85 | 0 |
| | *SE2* | 9 | 16 | 7 | 64 | 119 | 0 |
| | *SE3* | 17 | 66 | 612 | - | 289 | 0 |
| c3540 | *SE1* | 14 | 20 | 34 | - | 244 | 0 |
| | *SE2* | 13 | 245 | 63 | - | 216 | 0 |
| | *SE3* | 8 | 2 | 32 | - | 95 | 0 |
| c5315 | *SE1* | 4 | 2 | 13 | - | 45 | 0 |
| | *SE2* | 4 | 24 | 14 | - | 51 | 0 |
| | *SE3* | 3 | 1 | 51 | - | 29 | 0 |
| c7552 | *SE1* | 9 | 9 | 27 | - | 126 | 0 |
| | *SE2* | 12 | 6 | 179 | - | 195 | 0 |
| | *SE3* | 7 | 4 | 12 | - | 127 | 0 |

misclassified samples out of 1000 samples when the suspected path delays are solely used for classification, bearing in mind that the results would be worse if longer paths would be used. This column basically represents the solution suggested by [5]. Last column is the number of misclassified samples out of 1000 if the path delay ratio of two paths is used for classification as proposed in this paper.

Provided that the design and test stages are trustworthy, the experimental results reveal that the proposed method can suppress the variations and enables the HT detection. Required number of delay measurements at the manufacturing test stage is twice the number of suspected edges in the circuit. Having many alternatives for the suspected and correlated paths, not only reinforces the success of the algorithm but also hinders the adversary's capability to circumvent the method.

## ACKNOWLEDGMENT

This research is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the project number 112E237.

## REFERENCES

[1] M. Tehranipoor and C. Wang. *Introduction to Hardware Security and Trust*. SpringerLink : Bücher. Springer, 2011.

[2] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. *IEEE Design Test of Computers*, 27(1):10–25, Jan 2010.

[3] Y. Jin and Y. Makris. Hardware trojan detection using path delay fingerprint. In *IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 51–57, June 2008.

[4] J. Li and J. Lach. At-speed delay characterization for ic authentication and trojan horse detection. In *IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 8–14, June 2008.

[5] B. Cha and S. K. Gupta. Trojan detection via delay measurements: A new approach to select paths and vectors to maximize effectiveness and minimize cost. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1265–1270, March 2013.

[6] L. Zhang and C. H. Chang. Hardware trojan detection with linear regression based gate-level characterization. In *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pages 256–259, Nov 2014.

[7] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. G. Wolff, C. A. Papachristou, K. Roy, and S. Bhunia. Hardware trojan detection by multiple-parameter side-channel analysis. *IEEE Transactions on Computers*, 62(11):2183–2195, Nov 2013.

[8] A. A. Bayrakci. Stochastic logical effort as a variation aware delay model to estimate timing yield. *Integration, the VLSI Journal*, 48:101–108, 2015.

[9] http://www.itrs.net/. International Technology Roadmap for Semiconductors (ITRS) Report 2011 Edition.

[10] A. Agarwal, D. Blaauw, and V. Zolotov. Statistical timing analysis for intra-die process variations with spatial correlations. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, 2003.

[11] F. Brglez and H. Fujiwara. A neutral netlist of 10 combinational benchmark circuits. In *Proceedings IEEE International Symposium on Circuits and Systems*, page 695Ű698, 1985.

[12] Y. Jin, N. Kupp, and Y. Makris. Dftt: Design for trojan test. In *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*, pages 1168–1171, Dec 2010.

[13] H. C. Chen and D. H.-C. Du. Path sensitization in critical path problem [logic circuit design]. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 12(2):196–207, November 2006.

[14] L. Guerra e Silva, J. Marques-Silva, L. M. Silveira, and K. A. Sakallah. Satisfiability models and algorithms for circuit delay computation. *ACM Trans. Des. Autom. Electron. Syst.*, 7(1):137–158, January 2002.

[15] D. Sylvester, K. Agarwal, and S. Shah. Invited paper: Variability in nanometer cmos: Impact, analysis, and minimization. *Integration, the VLSI Journal*, 41(3):319–339, May 2008.

[16] J. Cong, P. Gupta, and J. Lee. Evaluating statistical power optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11):1750–1762, Nov 2010.

[17] http://www.nangate.com/. NanGate 45nm Open Cell Library.