Enhancing Hardware Trojan Detection Sensitivity Using Partition-Based Shuffling Scheme

Ahmad Shabani and Bijan Alizadeh, *Senior Member, IEEE*

***Abstract*—This paper presents a new test pattern generation scheme alongside a design for trust (DfTr) methodology for detecting hardware Trojan through side channel-based analysis approaches. The proposed scheme takes advantage of the proposed Hamming distance-based reordering and partition-based shuffling methods. The key idea behind the proposed work is that instead of distributing the circuit activity profile among all the nets with low transition probability at once, it can only be restricted to the smaller subsets of rare nets one at a time, while the rest of subsets remain at the lowest activity. As a result, a remarkable reduction in the background circuit activity by more than 41% and favorable enhancement of detection sensitivity about 31% are achieved compared to the state-of-the-art method.**

***Index Terms*— Hardware Trojan, Set Partitioning, Hamming Distance, Trojan Detection, Side Channel Analysis**

I. INTRODUCTION

Hardware Trojans are the malicious circuitry which are intentionally mounted by an untrusted person or entity aiming at altering the functionality, degrading the reliability, and leaking vital information of the chip [1]. The common assumption in hardware trust community is that an intelligent adversary would likely employ a Trojan trigger circuit by supplying it with the low transition probability nets (i.e., rare nets) for preserving its rarity of activation.

One promising way to alleviate the stealthy nature of Trojan is to increase the transition probability of the rare nets by inserting some additional circuitry into the main circuit [2]–[5]. These modifications are referred to as design for trust (DfTr) approaches [6]. Authors of [2], [4] developed two different DfTr methods for increasing the transition probability of the rare nets by inserting the scan flip-flops and a MUX-based structures into the main circuit. Later, authors of [3] proposed another insertion procedure which is based on XOR gates followed by flip-flops. In [5] authors developed a novel DfTr approach called PMTP with the aim of maximizing the transition probability of the rare nets. They introduced the PMTP rules and conflicts to ensure the proper propagation of the maximum transition probability. Then, the PMTP rules are satisfied through MAX-SAT and conflicts are resolved by inserting AND/OR into the main circuit. The existing DfTr methods mainly focus on the fully activation of Trojans through logic testing or side channel-based analysis approaches and they do not provide a unified solution for detecting Trojans through both ways. However, the standalone application of logic testing or side channel-based approaches is not sufficient for reliable detection, and in most of the cases the combination of both approaches is required [7]. The main weakness in the existing methods [2]–[5] is that any effort to increase the Trojan activity for raising the number of full activations of Trojan will increase the background circuit activity as well, and in turn this might adversely affect the side channel detection sensitivity.

In this paper, to provide a unified methodology for detecting hardware Trojans, a new test pattern generation (TPG) scheme is integrated with PMTP approach [5] using Hamming-Distance (HD)-based reordering and partition-based shuffling schemes. The main objective is to enhance the side channel sensitivity by reducing background activity while maintaining the number of Trojan full activations sufficient for logic testing approach. The contributions of the paper are as follows:

- We integrate the proposed TPG scheme with PMTP approach to enhance the detection sensitivity using HD-based reordering and partition-based shuffling methods.
- The partitioning problem is converted to a *Set Partitioning Problem* where some primary inputs are partitioned into multiple subsets, each of which covers different but almost the same number of rare nets.
- The proposed test set is generated by using shuffling method where a random set is reordered according to the maximum and minimum HDs, then the generated sets will be shuffled based on the partitioning results.

The rest of the paper is organized as follows. The problem specification and the proposed methodology is introduced in Sections II and III. Then, the experimental results are extracted in Section IV. Finally, a conclusion will be presented.

II. PROBLEM SPECIFICATION AND THE PROPOSED METHOD

To the best of our knowledge, the PMTP method [5] outperforms the existing DfTr approaches in terms of the number of full activations of Trojan. To maximize the activity of the rare nets, they specified the full transition (*FT*) paths between the rare nets and the primary inputs of the circuit. These paths are specified based on the minimum logical depth rule and are sensitized through some primary inputs fed by random-based test vectors to increase the transition at the rare nets. The rest of the primary inputs are stuck at their controlling values of '0' or '1' specified by the MAX-SAT solver to let the maximum transition be propagated. However, these high activity random-based vectors not only increase the Trojan activity, but also the

A. Shabani (Email: ah.shabani@ut.ac.ir) is with the Design, Verification and Debugging of Embedded Systems Laboratory (DVDES), School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14395-515, Iran.

B. Alizadeh (Corresponding author. Email: b.alizadeh@ut.ac.ir) is with the Design, Verification and Debugging of Embedded Systems Laboratory (DVDES), School of Electrical and Computer Engineering, College of Engineering, University of Tehran, 14395-515, Iran and also with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran 19538-33511, Iran.

background circuit activity, resulting in a low side channel detection sensitivity. Developing a pattern generation strategy to localize switching activity and reduce the background noise is an extremely challenging task since there is correlation between the Trojan activity and the total number of background transitions. Accordingly, we present a pattern generation strategy alongside the PMTP approach to enhance the side channel sensitivity by localizing the circuit activity on the target partitions covering a subset of rare nets, while keeping quiet the rest of the partitions. Fig. 1 shows the proposed integrated DfTr methodology incorporating 1) the PMTP approach [5] and 2) the proposed TPG scheme. The test set generated by the PMTP approach can be divided into two parts of mutable and immutable parts. The immutable part remains constant for all the test vectors where its logic values are specified by MAX-SAT, while the mutable part refers to the part that frequently varies where it is supplied by the random-based test vectors. To localize activity, we develop our TPG scheme on the mutable part since this part is responsible for generating activity in the circuit. This part of test vector injects activity into the circuit through some primary inputs called *Stimulating Input* (*SI*) which affects both Trojan and background activity simultaneously. Thus, instead of random test pattern, we proposed a guided test patterns which can target the subsets of rare nets one at the time while reducing background circuit activity.

In Fig.1, the proposed unified methodology takes the circuit netlist as input and using the PMTP approach generates the modified circuit netlist by employing AND/OR insertion-point structures into the main circuit. The proposed TPG scheme consists of two main phases: 1) Stimulating input partitioning and 2) Partition-based shuffling. In this concept, the stimulating inputs are the primary inputs (i.e., endpoints) of the *FT* paths. In the first phase, the stimulating inputs are extracted and then partitioned into multiple subsets. In the second phase, the random test vectors are reordered based on minimum and maximum HDs. Then, the two reordered sets are shuffled based on the partitioning results to generate the mutable part. Finally, the mutable and immutable parts are combined to generated the complete test set.

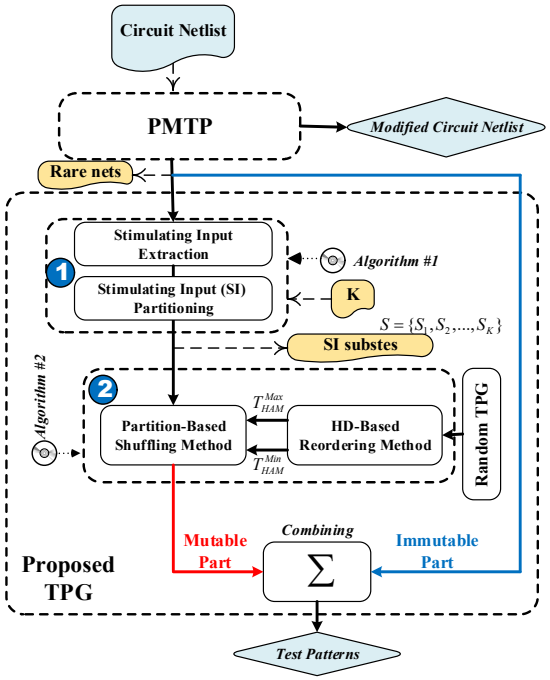

Fig. 1. The proposed integrated DfTr methodology

III. Proposed Test Pattern Generation Scheme

Stimulating Inputs Partitioning Method: The stimulating input partitioning method is described in Algorithm #1. Before partitioning process, we need to extract the stimulating inputs (P=$\{p_1,p,...,p_N\}$), where N is the total number of stimulating inputs, by following each of the rare nets ($n_i \in \mathcal{LT}$) through FT paths ($\mathcal{X}$) towards primary inputs (Stage #1). In this backward process, when the primary inputs is reached, it is added to the P list and assigned as the stimulating input of the rare net n_i ($STM\{n_i\}$). Then, the coverage of each $p_i \in P$ ($CC\{p_i\}$) is calculated by checking how many rare nets are covered by each stimulating input. The number of rare nets which are sensitized by a stimulating input is defined as the Coverage Count (CC) of that input.

Algorithm 1: Stimulating Inputs Partitioning

Input: Circuit netlist, List of FT paths ($\mathcal{X}$), number of partitions (K), List of rare nets ($\mathcal{LT}$);
Output: Generating subsets $S = (S_1, S_2, \dots, S_K)$

STAGE #1: /* Stimulating Inputs Extraction */

For *each rare node* $n_i \in \mathcal{LT}$ ***do***
  $sel = n_i$;
  While $(Fain\_in(sel) \neq Primary\ input)$ ***do***
    $sel = Fain\_in(sel) | Fain\_in(sel) \in \mathcal{X}$
  endwhile
P= Push *sel //P: Set of stimulating inputs*
$STM\{n_i\} = sel$ *// sel is the stimulating input of a rare net* n_i
endfor
$CC\{p_i\}$ = ***Find*** *coverage counts of all* $p_i \in P$

STAGE #2: /* Greedy Partitioning */

P=Sort *P in descending order of coverage count (CC).*
For *each stimulating input* $p_i \in P$ ***do***
 S_i=***Find*** *the set* $(S_1, S_2, \dots, S_K)$ *with smaller sum of coverages.*
 Assign p_i *to* S_i
endfor

Since the test designer is not aware of which rare nets are selected as Trojan's trigger and we assign an identical portion of test vectors to each generated subsets, we need to make sure that each subset covers relatively the same number of rare nets (i.e., the balanced subsets in terms of the coverage count) which is translated to the same amount of activity assigned to each subset. The problem ahead is to partition the set of stimulating inputs (P) into K balanced subsets (S_i) such that the difference between the sum of the coverage counts of each subset is minimized. In fact, this is an optimization version of the *Set Partitioning Problem* [8], which is a task of deciding whether a given multiset P of positive integers can be partitioned into two subsets of s_1 and s_2 such that the sum of the numbers in each subset is equal.

According to Stage #2 of Algorithm #1, one simple heuristics approach to the above problem is the greedy algorithm, which iterates through the set of stimulating inputs sorted in descending order of coverage count, assigning each of them to whichever subset has the smaller sum of the coverages. The

proposed partitioning process is shown in Fig. 2 in which the set of stimulating inputs P={p_1, p_2,..., p_{13}} is partitioned into three subsets S={S_1, S_2, S_3}, $K = 3$ with relatively the same number of total coverage counts. In Fig. 2, the number in the boxes represents the coverage count of each p_i and the P set is initially sorted in descending order of coverage count. We can evaluate the partitioning efficiency as the maximum difference of sum of coverages (*Inaccuracy*) as a percentage of the average sum which is called the *Relative Inaccuracy (RI)*.

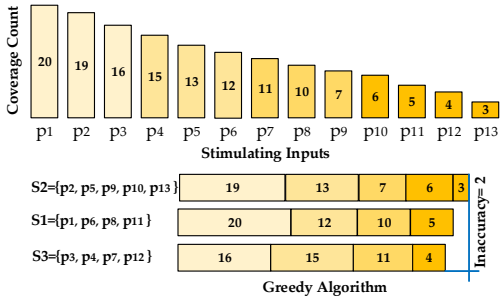

Fig. 2. An example of stimulating inputs partitioning problem

Partition-Based Shuffling Method: The key idea to lower the background circuit activity as well as enhancing the detection sensitivity is to generate the guided test patterns associated with each partition in such a way that once a target partition is under high activity, the rest of the partitions remain idle as far as possible. A simple heuristic to lower the backgr-ound circuit activity is to have similar input vectors, which represents the sorting of test vectors with minimum HDs. Similarly, to maximize the activity of the target partition, the random test vectors are sorted based on maximum HDs. Finally, the sorted sets are shuffled based on the partitioning results (i.e., shuffling step). We called this solution as the partition-based shuffling method.

Algorithm #2 takes the random test set {t_1,t_2,...,t_{Ntest}} and the partitioning results {$S_1, S_2, \dots, S_K$}, where K is the number of partitions and N_{test} is the number of test vectors (Line 1). Each partition S_i includes one or multiple elements of stimulating inputs p_i. This algorithm produces the shuffled test patterns (T^i_{shuf}) individually targeting each of the partitions (Line 2). In the first step, the random set is sorted based on the maximum and minimum HDs (Lines 3-4). For two test vectors of equal length, the HD is the number of bit positions in which the two bits are different. In order to calculate the HD between two test vectors, we perform their XOR operation, and then count the total number of 1s in the resultant output [9]. Thus, two test sets sorted based on maximum HD (T^{Max}_{Ham}) and minimum HD (T^{Min}_{Ham}) are generated. Then, for each partition S_i, an identical number of test vectors (i.e., N_{test}/K) from T^{Min}_{Ham} is selected and then the shuffling process is performed for that partition (Lines 7-8). During the shuffling process, we take the headmost test vector from T^{Min}_{Ham} (i.e., t_{min}) and T^{Max}_{Ham} (i.e., t_{max}) set (Line 9). The Pop function removes and returns the first item in the list. For each t_{min} of the target partition S_i, some bits where their positions are specified by the target partition are replaced with the corresponding bit positions of t_{max}. These bit positions are specified by the index of elements of target partition (*Index*(p_i)$= i$) (Lines 10-12). For example, if partition #1 includes the elements of {p_2, p_4, p_8}, the second, fourth, and eighth bit positions of each t_{min} is replaced with the corresponding bit positions of t_{max}. This process is repeated for the rest of partitions as shown in Fig. 3.

Algorithm 2: Partition-Based Shuffling Method

1. **Input:** Random test pattern T_{orig} = {t_1,t_2,...,t_{Ntest}}, Partitions S = {$S_1, S_2, \dots, S_K$}
2. **Output:** Shuffled test patterns T_{shuf} = {T^1_{shuf}, T^2_{shuf} ,...,T^K_{shuf}}
3. T^{Min}_{Ham} = **Sort** T_{orig} based on minimum $hamming\_dist(t_i, t_j)$
4. T^{Max}_{Ham} = **Sort** T_{orig} based on maximum $hamming\_dist(t_i, t_j)$
5. /* $hamming\_dist(t_i, t_j)$ returns the total number of 1s in $XOR(t_i, t_j)$*/
6. ***For*** *each partition* $S_i \in S$ ***do***
7. ***Initial*** *count=0; i=0; i++;*
8. ***While*** *(count<*(N_{test}/K*))****do***
9. t_{min} =***Pop***(T^{Min}_{Ham}); t_{max} =***Pop***(T^{Max}_{Ham}); *count++; /****Pop*** *function removes and return the first item in the list*/*
10. ***For*** *each partition element* $p_i \in S_i$ ***do***
11. $t_{min}[Index(p_i)] = t_{max}[Index(p_i)]$ // *Index*(p_i)$= i$
12. ***endfor***
13. ***Push*** (T^i_{shuf}, t_{min})
14. ***endwhile***
15. ***Push*** (T_{shuf}, T^i_{shuf}) // {T^1_{shuf}, T^2_{shuf} ,...,T^K_{shuf}}
16. ***Endfor***

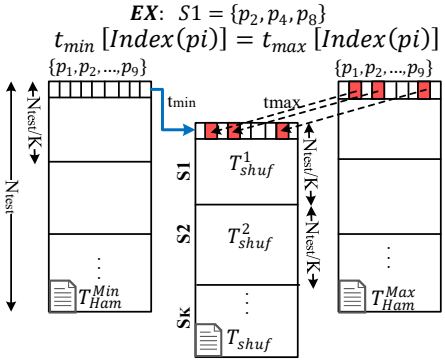

Fig. 3. An example of partition-based shuffling algorithm

IV. Experimental Results

We have implemented the entire proposed methodology using Perl and used a greedy algorithm described in Algorithm #1 with complexity of $O(NLogN)$ to partition a set of N stimulating inputs into K balanced subsets. Fig. 4 shows the relative inaccuracy (RI) of partitioning process for different partition cardinalities K. It is concluded from Fig. 4 that there is not a clear trend between RI and K value since the partitioning inaccuracy mainly depends on the coverage counts which is a circuit-dependent parameter. However, the analysis shows that the average RI for $K = 4$ is lower than the other K values and it generates more balanced subsets. The RI and the rare nets covered by each subset (CC ($subset$ #i)) have been provided in Table I. From Table I, the average RI of partitioning process is 2.58% for $K = 4$ which is satisfying as the generated subsets cover relatively the same number of rare nets.

One metric to assess the side channel sensitivity is the Trojan to circuit activity (TCA) which is defined as the ratio of Trojan activity to the background activity. Also, to measure the

efficiency of the logic testing approach, we analyze the number of full activations of Trojans. To expose the superior efficiency of the proposed methodology in detecting Trojans through both logic testing and side channel-based analysis approaches, we apply the proposed test set to 100 randomly inserted Trojan samples and compute these two metric for each Trojan instance. We would then take the average of these two metrics, which reflects the efficiency of our test set to enable detection of different Trojans. Also, we have randomly inserted different Trojan circuits with different sizes and difficulty of activations into the main circuits. The characteristics of employed Trojan models are described in [4], [5]. The stimulating inputs of each circuit are partitioned into four subsets and 1000 test patterns are generated by the proposed scheme to individually trigger each partition. Table II shows the average number of Trojan full activations over 100 random Trojan samples with different Trojan circuits after applying the proposed partition-based shuffling scheme. The trigger's inputs of Trojans are randomly selected from the rare nets whose transition probabilities are below a specified threshold (P_{th}). From Table II, the proposed methodology can fully activate all the Trojan circuits excluding Trojan #4. None of the existing DfTr approaches [2], [4], [5] can also activate such a large Trojan since it incorporates 12-input trigger and it is hard-to-detect through logic testing in a limited number of test vectors ($N_{test} = 1000$). Luckily, the contribution of this Trojan into circuit activity is high and it can be detected through power-based analysis approaches. Table III shows the average side channel detection sensitivity of different Trojan circuits after applying the proposed partition-based shuffling scheme. As the size of Trojan increases from Trojan #1 to Trojan #4, so does the side channel detection sensitivity, but the number of full activations of Trojan will be lessened through logic testing approach according to Table II.

In Table IV, we consider four different cases of Trojan trigger selection. In each case, trigger's inputs are randomly selected from the rare nets covered by a single or multiple subsets. Tables V and VI report the background activity and detection sensitivity in different cases of trigger selection for Trojan #4 and using $K = 4$. The results show that the proposed scheme can drastically lower the background circuit activity to more than 43%, on average, compared to the PMTP [5]. Moreover, the detection sensitivity is favorably augmented by more than 31%. Fig. 5 shows the distribution of maximum TCA and maximum Trojan activity over 100 random Trojan samples of Trojan #3 and Trojan #4 for s5378 circuit. The box plots indicate the minimum, first quartile, median, third quartile, and maximum value of the distribution during applying 1000 test patterns. From Fig. 5, it can be seen that the distribution of maximum TCA for both Trojans #1 and #2 is enhanced in our approach compared to the other approaches [2], [3], [5], with relatively the same Trojan activity. Also, the background circuit activity shows a considerable reduction around 50% compared to PMTP approach. Fig. 6 shows the transient TCA over 100 random Trojan samples during applying 1000 test patterns. We assigned an identical portion of test patterns for each partition and consider four different cases of trigger selection based on the partitions from which the trigger's inputs are selected. In

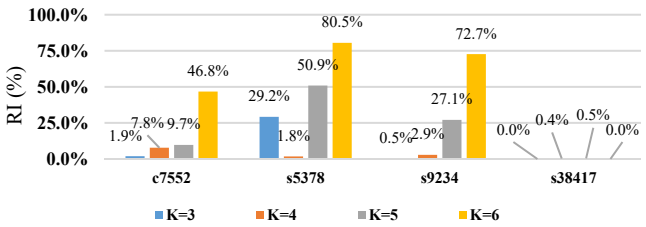

Fig. 4. Relative inaccuracy of partitioning versus different K values

TABLE I: NUMBER OF RARE NETS COVERED BY EACH PARTITION (K=4)

Circuit	c7552	c5315	s5378	s9234	s15850	s38417
CC(subset #1)	38	64	57	138	175	272
CC(subset #2)	40	64	56	141	177	272
CC(subset #3)	39	63	57	137	176	271
CC(subset #4)	37	63	56	137	176	271
CC(Total)	154	254	226	553	932	1086
RI (%)	7.79%	1.57%	1.77%	2.9%	1.1%	0.4%

TABLE II: NUMBER OF FULL ACTIVATIONS OF DIFFERENT TROJANS

Circuit	P_{th}	Trojan #1	Trojan #2	Trojan #3	Trojan #4
c2670	0.10	127	63	3	0
c3540	0.05	114	47	15	0
c5315	0.10	87	32	10	0
c7552	0.05	103	40	25	0
s5378	0.05	170	65	44	0
s9234	0.05	115	52	16	0
s13207	0.05	138	85	5	0
s38417	0.05	105	40	14	0

TABLE III: DETECTION SENSITIVITY (TCA) FOR DIFFERENT TROJANS

Circuit	P_{th}	Trojan #1	Trojan #2	Trojan #3	Trojan #4
c2670	0.10	3.238E-03	1.046E-02	1.282E-02	2.170E-02
c3540	0.05	2.407E-03	8.620E-03	1.261E-02	2.080E-02
c5315	0.10	1.520E-03	5.500E-03	8.890E-03	1.370E-02
c7552	0.05	1.800E-03	7.400E-03	1.140E-02	1.890E-02
s5378	0.05	2.080E-03	8.600E-03	1.450E-02	2.520E-02
s9234	0.05	1.210E-03	4.040E-03	6.315E-03	1.270E-02
s13207	0.05	5.100E-04	1.686E-03	2.261E-03	5.600E-03
s38417	0.05	1.021E-03	3.523E-03	5.207E-03	8.632E-03

TABLE IV: DIFFERENT CASES OF TRIGGER INPUT SELECTION OF TROJAN

Cases	Trigger's inputs selected from rare nets covered by:
CASE #1	Partition #1
CASE #2	Partition #1 & Partition #2
CASE #3	Partition #1 & Partition #2 & Partition #3
CASE #4	Random Selection

CASE #1, all the Trigger's inputs are randomly selected from partition #1. According to the CASE #1, the transient TCA values are mostly restricted to the partition from which the rare nets are selected (i.e., partition #1), while the TCA values of the rest of the partitions remain almost zero for most of the times. Also, the TCA of the proposed scheme shows the higher peak than the other approaches. This distinguished activity profile can help test engineer to detect the presence of Trojan more reliable. This observation can relatively identify the rare nets used as trigger's inputs with higher resolution. Once the trigger selection broadens to multiple partitions (e.g., CASE #2), the sensitivity profile also distributes among those partitions.

V. CONCLUSION

In this paper, a new test pattern generation scheme was developed to enhance the side channel detection sensitivity of Trojan by using the proposed HD-based reordering and

TABLE VI: BACKGROUND CIRCUIT ACTIVITY OF THE PROPOSED AND THE PMTP APPROACHES FOR DIFFERENT TCA SENSORS FOR TRIGGER SELECTION (K=4)

Circuit	CASE #1		CASE #2		CASE #3		CASE #4		Average		DEC (%)
	Proposed	PMTP	Proposed	PMTP	Proposed	PMTP	Proposed	PMTP	Proposed	PMTP	
c7552	1.5E+05	3.1E+05	1.5E+05	3.1E+05	1.5E+05	3.1E+05	1.5E+05	3.1E+05	1.5E+05	3.1E+05	51.61%
s5378	1.2E+05	2.3E+05	1.2E+05	2.3E+05	1.2E+05	2.3E+05	1.2E+05	2.3E+05	1.2E+05	2.3E+05	47.83%
s9234	2.6E+05	4.9E+05	2.6E+05	4.9E+05	2.6E+05	4.9E+05	2.6E+05	4.9E+05	2.6E+05	4.9E+05	46.94%
s15850	5.0E+05	7.7E+05	5.0E+05	7.7E+05	5.0E+05	7.7E+05	5.0E+05	7.7E+05	5.0E+05	7.7E+05	35.06%
s38417	8.5E+05	1.3E+06	8.5E+05	1.3E+06	8.5E+05	1.3E+06	8.5E+05	1.3E+06	8.5E+05	1.3E+06	34.62%
Average decrease in background circuit activity compared to PMTP approach:											43.21%

TABLE VII: SIDE CHANNEL DETECTION SENSITIVITY OF THE PROPOSED AND THE PMTP APPROACHES FOR DIFFERENT TCA SENSORS FOR TRIGGER SELECTION (K=4)

Circuit	CASE #1		CASE #2		CASE #3		CASE #4		Average		INC (%)
	Proposed	PMTP	Proposed	PMTP	Proposed	PMTP	Proposed	PMTP	Proposed	PMTP	
c7552	0.0154	0.0114	0.0132	0.0129	0.0196	0.0142	0.0214	0.0112	0.0174	0.0124	40.32%
s5378	0.0217	0.0178	0.0252	0.0198	0.0251	0.0194	0.0208	0.0143	0.0232	0.0179	29.61%
s9234	0.0116	0.0099	0.0101	0.0083	0.0126	0.0085	0.0100	0.0081	0.0111	0.0087	27.59%
s15850	0.0058	0.0045	0.0062	0.0042	0.0060	0.0052	0.0048	0.0044	0.0057	0.0046	23.91%
s38417	0.0052	0.0036	0.0049	0.0035	0.0050	0.0036	0.0044	0.0032	0.0048	0.0035	37.14%
Average increase in Trojan to circuit activity compared to PMTP approach:											31.71%

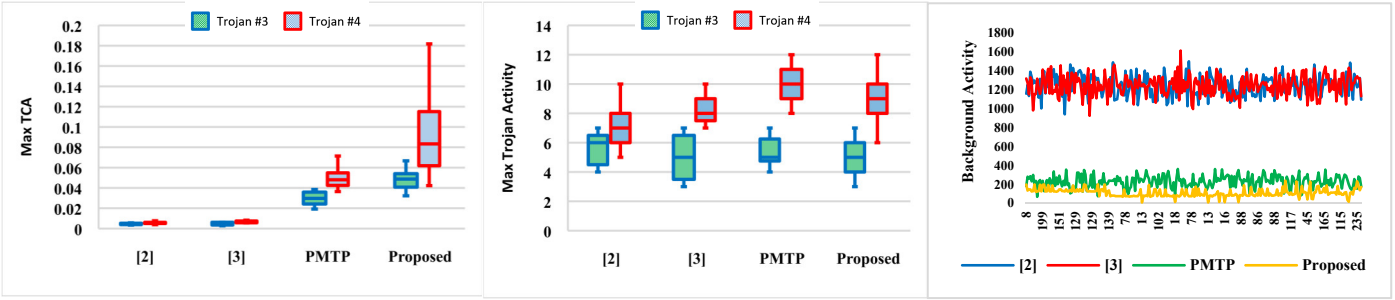

Fig. 5. Distribution of maximum TCA, Trojan activity, and background activity over 100 random Trojan samples of Trojan #3 and #4 for s5378

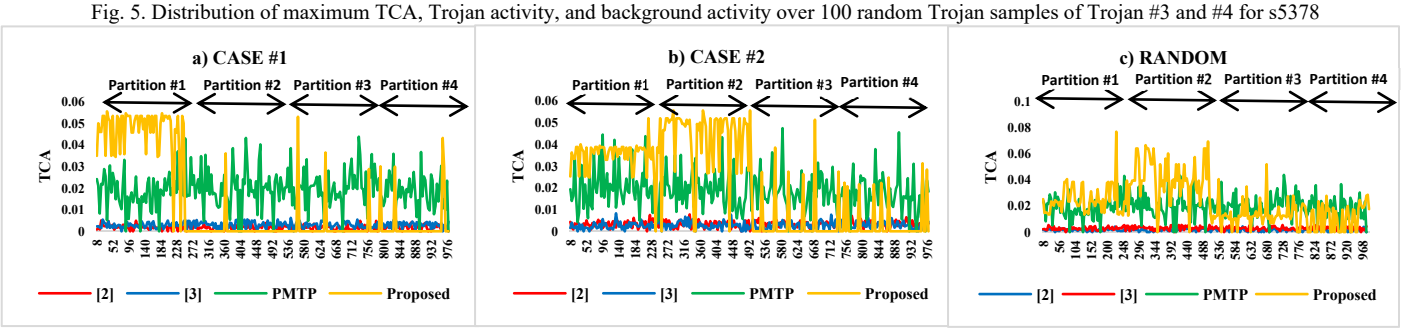

Fig. 6. Comparison of side channel sensitivity over 100 random Trojan samples, a) CASE #1; b) CASE #2; c) Random (s5378, $k = 4$, N_{test}=1000)

partition-based shuffling methods. Unlike the previous works, the detection sensitivity profile of the proposed approach is much higher and almost restricted to the partitions from which the Trojan trigger's inputs are selected. As a result, the Trojan-infected circuits can be distinguished with more reliability and the trigger's inputs are identified with higher resolution.

REFERENCES

[1] S. Bhunia and M. M. Tehranipoor, *The hardware Trojan war: Attacks, myths, and defenses*. 2017.

[2] B. Zhou, W. Zhang, S. Thambipillai, and J. K. J. Teo, "A low cost acceleration method for hardware trojan detection based on fan-out cone analysis," in *2014 International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS 2014*, 2014, p. 28.

[3] A. Mondal, M. H. Mahalat, A. R. Medapati, S. Roy, and B. Sen, "XOR based Methodology to Detect Hardware Trojan utilizing the Transition Probability," in *Proceedings of the 2018 8th International Symposium on Embedded Computing and System Design, ISED 2018*, 2018, pp. 215–219.

[4] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A novel technique for improving hardware trojan detection and reducing trojan activation time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 112–125, 2012.

[5] A. Shabani and B. Alizadeh, "PMTP: A MAX-SAT-Based Approach to Detect Hardware Trojan Using Propagation of Maximum Transition Probability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 1, pp. 25–33, 2020.

[6] K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor, "Hardware trojans: Lessons learned after one decade of research," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 1, pp. 1–23, 2016.

[7] M. Tehranipoor, H. Salmani, and X. Zhang, *Integrated circuit authentication: Hardware trojans and counterfeit detection*, vol. 9783319008. Springer, 2014.

[8] R. E. Korf, "Multi-way number partitioning," in *IJCAI International Joint Conference on Artificial Intelligence*, 2009, pp. 538–543.

[9] Y. Huang, S. Bhunia, and P. Mishra, "MERS: Statistical test generation for side-channel analysis based Trojan detection," *Proceedings of the ACM Conference on Computer and Communications Security*, vol. 24-28-Octo, no. 1, pp. 130–141, 2016.