HTOutlier: Hardware Trojan Detection with Side-Channel Signature Outlier Identification

Jie Zhang[†‡], Haile Yu[†] and Qiang Xu[†‡]
[†]CUhk REliable Computing Laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Email: {jzhang,hlyu,qxu}@cse.cuhk.edu.hk

ABSTRACT

Hardware Trojan (HT) is a growing concern for the semiconductor industry. As a non-invasive and inexpensive approach, side-channel analysis methods based on signatures such as power, current, or circuit delay are widely used for HT detection. However, the effectiveness of these methods is greatly challenged by the ever-increasing process variation (PV) effects with technology scaling. In this work, considering the inherent relationship among side-channel signatures in a chip, we formulate the HT detection problem as a signature outlier identification problem, and solve it by comparing each signature with an estimated value from other signatures. Experimental results on benchmark circuits show that the proposed technique is much more effective than existing solutions.

1. INTRODUCTION

Today's integrated circuit designs involve many third-parties during the design and manufacturing process (e.g., IP core providers and foundries), and hence they are vulnerable to a wide range of malicious alterations, namely hardware Trojans (HTs) [1, 2]. A variety of techniques have been proposed for HT detection, as surveyed in [1, 2, 3]. Since traditional VLSI testing techniques are not suitable for HT detection without knowledge of the HT and trigger conditions [3], the mainstream HT detection method is side-channel analysis. The idea behind is that HTs must affect some side-channel signatures, e.g., path delay [4, 5], supply current [6, 7, 8] or power consumption [9], even when functionally-unactivated.

With the above, one HT detection method is to compare side-channel signature of the under-validated chips against that of HT-free reference chips, referred to as *reference-based method* [4, 7, 9, 10, 11]. These techniques are very sensitive to process variation (PV) and hence their effectiveness is greatly challenged with technology scaling. To tackle this problem, HT detection by gate-level characterization (GLC) was proposed recently [6, 8, 12, 13]. GLC-based method models PV effect on each gate as a scaling factor of the nominal value and treats the side-channel signature as a linear combination of gate characteristics. Then, by minimizing measurement errors (MEs) during optimization, the existence of HT will manifest themselves as abnormal scaling factors. GLC-based method is conceptually interesting and PV-resistant, but its scalability is a serious challenge due to the inherent computational complexity to characterize each and every gate in the circuit.

In this paper, we focus on HTs inserted during the manufacturing process. Different from existing techniques, we formulate the HT detection problem as a *side-channel signature outlier identification* problem. That is, given a set of signatures generated by different input patterns for a chip wherein some can be affected by the HT (if it exists) while others cannot, to detect whether HT exists or not is equivalent to determine whether there exists HT-affected signatures. We define an HT-affected signature as an *outlier* for a given set of signatures. Considering the inherent relationship among side-channel signatures due to the fact that same gates are likely to be used for different signatures and gates are correlated with each other by spatial correlation, outliers can be detected by comparing each measured signature with an estimated value derived from other measured signatures. Our approach, namely *HTOutlier*, has the advantages of being both PV-resistant and scalable, as demonstrated in our experimental results. In addition, *HTOutlier* does not require the existence of a trustworthy golden IC for reference.

The remainder of this paper is organized as follows. Section 2 surveys related work in this domain. In Section 3, we formulate the HT detection problem as a signature outlier identification problem. Next, we describe the signature model and estimation function in Section 4. The HT detection algorithm is discussed in Section 5. Experimental results are then presented in Section 6 to demonstrate the effectiveness of the proposed solution. Finally, Section 7 concludes this paper.

2. RELATED WORK

As discussed earlier, there are mainly two kinds of side-channel analysis methods for HT detection: reference-based method and GLC-based method. We summarize them as follows.

2.1 Reference-Based HT Detection

Reference-based HT detection method is conceptually simple. For a chip under validation, it is claimed to contain HTs if its signature ($\hat{S}$) is outside a user-defined confidence interval pre-computed by signatures from HT-free reference chips. Early reference-based methods adopted direct chip-to-chip comparison for HT detection [9]. To increase the HT contribution ratio, several region-to-region comparison methods were presented, wherein the chip is partitioned into multiple regions and only one region is activated at a time, controlled by either primary inputs [1, 10] or reordered scan chains [14]. In order to mitigate PV effects on HT detection, Du *et al.* proposed a so-called self-referencing comparison method in [7]. In this technique, by exploiting systematic correlation among regions, a combination of multi-region signatures of the chip under validation is compared against that of the reference chip. While intuitively effective, there is no theoretic supports and it is rather difficult to obtain a good confidence interval with this method due to its complex comparison method.

Although simple and scalable, reference-based HT detection methods have several inherent limitations: (i). Reference chips are not easy to acquire; (ii). PV effects result in significant variation of the signatures; (iii). Since it is very difficult to determine useful input patterns.

2.2 GLC-Based HT Detection

Gate-level characterization for HT detection was first presented in [6, 8]. In GLC-based methods, side-channel signatures are modeled as a linear combination of gate characteristics, wherein the PV effect on each gate is modeled as a scaling factor that represents the deviation from the nominal value. Then, linear programming or quadratic programming is used to characterize scaling factors, by minimizing measurement errors. Since the signature given by the mathematic signature model does not consider HT, an inserted HT would impact the

978-1-4673-2340-6/12/$31.00

estimation operation on scaling factors. Then, HT can be detected by observing abnormal scaling factors.

Although GLC-based methods are theoretically interesting and resistant to PV effects, they are not scalable due to the large amount of variables and equations involved and the limited controllability of internal nodes. To mitigate the above problems, [13] partitioned the large circuit into small sub-circuits with primary input control to reduce the problem complexity. However, the HT detection capability would be also reduced since input patterns are constrained within a small range. [12] proposed to adopt thermal control technique to characterize more gates by considering the impact of temperature on PV, but such approach could not solve the scalability problem. Considering the ever-increasing transistor-to-pin ratio with technology scaling, it is rather difficult, if not impossible, for GLC-based methods to be applicable to industrial designs.

The limitations of existing HT detection methods motivate the proposed *HTOutlier* technique, as detailed in the following.

3. PROBLEM FORMULATION

Consider a set of signatures ($\hat{\mathbf{S}} = \{\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_N\}$) for a chip which are measured under different input patterns. We assume one or more signatures are HT-affected while the others are not. Signatures are correlated with each other due to: (i). the same kinds of gates are likely to be activated when generating signatures; (ii). there is inevitable systematic correlation in the chip. With the above, we formulate the HT detection problem as an *outlier identification problem* [15] as follows.

For each signature $\hat{S}_i$, we define its *k neighboring signature set* ($\mathbf{T}_k(\hat{S}_i) = \{\hat{S}_{i1}, \hat{S}_{i2}, \ldots, \hat{S}_{ik}\}$) composed of k signatures that are most correlated with $\hat{S}_i$. The correlations between signatures are calculated according to the signature model presented in Section 4.1. The size of neighboring signature set (k) is user-defined. The *estimation function* $G(\hat{S}_i)$ is defined as the operation that estimates $\hat{S}_i$ based on its neighboring signature set $\mathbf{T}_k(\hat{S}_i)$, detailed in Section 4.2. To detect the outlier in the given signature set, we compare each measured signature with its corresponding estimated value, as discussed in Section 5. A signature is regarded as an outlier if it is outside a user-defined *confidence interval* that is constructed according to its corresponding estimated value and estimation variance. Any found outlier indicates that the chip is HT-inserted.

4. SIGNATURE MODEL AND ESTIMATION FUNCTION

In this section, we first introduce the signature model used in this paper, then present the estimation function in detail, and finally demonstrate the impact of PV model error and measurement errors on the estimation function.

4.1 Signature Model

In this paper, we adopt the transient current in our side-channel analysis. The proposed *HTOutlier* framework, however, can also be used with other kinds of signatures, such as delay and power.

The PV model used refers to [17, 18]. The most dominant PV on the transient current derives from the variation of the threshold voltage (V_{th}). Thus, the transient current of the gate is given as:

$$I_g = k_g(V_{DD} - V_{th_0} - V_c - V_s - V_r)^2,$$

where k_g is a constant for all gates composed of common parameters, V_{DD} is the supply voltage, and V_{th0} is the nominal value of threshold voltage. V_c, V_s and V_r are random variables. Ignoring the second order term of PV, the above equation can be simplified as:

$$I_g \approx k_g[V_{go}^2 - 2V_{go}(V_c + V_s + V_r)], \quad V_{go} = V_{DD} - V_{th_0}.$$

Then, the transient current of the chip under input pattern i can be obtained by summing transient currents of transited gates. Similar to previous work [7, 18], we assume chips are divided into $N \times N$ [1] rectangular regions, and the systematic variation is identical within a region. Therefore, the transient current of the chip under input pattern i can be written as:

$$I_i = k_g n_i V_{go}^2 - 2k_g n_i V_{go} V_c - 2k_g V_{go} \sum_{k=1}^{N^2} n_{ik} V_{sik} - 2k_g V_{go} \sum_{j=1}^{n_i} V_{rij}, \quad (1)$$

where n_{ik} is the number of transitions in region k and V_{sik} is the systematic variation in region k. We name Eq. (1) as the *signature model (S_i)* under input pattern i. For each measured signature $\hat{S}$, we build its signature model S as the above, where the condition of transited gates is obtained through the logic simulation. Since the HT is inserted during the manufacturing process, the designer can have a golden netlist. The signature model is based on the golden netlist, and hence it must not include any HT effect.

4.2 Estimation Function

The effectiveness of *HTOutlier* mainly relies on the estimation function which estimates a signature according to its neighboring signatures. As shown above, any signature can be represented as the sum of the contributions of gates (e.g., Eq. (1)). This implies that signatures for a chip are possibly correlated with each other. There are two reasons. The first is that signatures are likely to contain the contribution from the same gates. The second is that contributions of gates are possibly correlated due to the systematic correlation. This observation invokes us to estimate a signature by others.

To estimate a signature from its neighboring signatures by exploiting their dependencies, we adopt an estimation function based on the *ordinary kriging (OK)* algorithm [16] which is the best linear unbiased estimator. According to OK, the estimation function for a signature ($\hat{S}_i$) is a linear combination of its neighboring signatures ($\mathbf{T}_k(\hat{S}_i)$), given as:

$$G(\hat{S}_i) = \sum_{t=1}^{k} \lambda_t \hat{S}_{it}, \qquad \sum_{t=1}^{k} \lambda_t = 1, \quad (2)$$

where $\Lambda = (\lambda_1, \ldots, \lambda_k)^T$ denotes the weight matrix, and $G(\hat{S}_i)$ denotes the estimated value. The sum of weights is equal to one. This allows us to have an unbiased estimator without prior knowledge of the stationary mean of signatures.

The weights for each neighboring signature are determined according to their signature models. In order to provide the optimal weights, we minimize the estimation variance (σ_k^2) subject to the unbiasedness condition, represented as:

$$\begin{aligned} &\text{Objective: } \min \sigma_k^2(G(S_i)) = \min var(G(S_i) - S_i); \\ &\text{Constraint: } E|G(S_i) - S_i| = 0. \end{aligned} \quad (3)$$

With these weights, estimated value can be calculated by Eq. (2) The above estimator is scalable to the large-scale design, since its complexity depends on the size of the neighboring signature set.

To solve Eq. (3), two requirements should be satisfied. (i). The mean of all signatures can be unknown but should be identical. (ii). The variograms between any two signatures as shown in Eq. (5) should be given. The first requirement can be satisfied by normalizing signatures. The normalization operation for the signature S_i given in Eq. (1) is shown as:

$$S_i = \frac{S_i}{2k_g V_{go} n_i} \approx \frac{1}{2} V_{go} - V_c - \frac{1}{n_i} \sum_{k=1}^{N^2} n_{ik} V_{sik}. \quad (4)$$

In Eq. (4), we remove the random variation, as it is equal to zero and its variance is approximately equal to zero with large value n_i. Thus, S_i follows the normal distribution $S_i \sim N(\frac{1}{2}V_{go} - V_c, \sigma^2(S_i))$ where $\sigma^2(S_i)$ denotes the intra-chip variance of S_i. V_c is unknown, but it has identical effect on the chip. Thus, the mean of all signatures is equal to $\frac{1}{2}V_{go} - V_c$. In the following Section 4 and 5, without the special description, when we mention the signature, we mean the normalized signature. For the second requirement, variograms between any two signatures are calculated with the help of the PV model which can be provided by foundries, given as:

$$\gamma(S_i, S_j) = \frac{1}{2} var(S_i - S_j) = \frac{1}{2} C_i^T \Omega C_i + \frac{1}{2} C_j^T \Omega C_j - C_i^T \Omega C_j, \quad (5)$$

where C_i shows the matrix of the number of transited gates and Ω denotes the covariance matrix among different regions. C_i is obtained by the logic simulation while Ω is calculated by the PV model.

[1] N=100 in the experiment.

With the normalized signatures and their variograms, we can solve Eq. (3). Due to the space limitation, we do not show how to solve it in details. The weight matrix (Λ) to estimate $\hat{S}_i$ are given as:

$$\Lambda_0 = \Gamma_0^{-1}\gamma_0, \quad \Lambda_0 = (\lambda_1, \ldots, \lambda_k, \mu)^T, \tag{6}$$

where Γ_0 and γ_0 can founded in [16]. The estimated weights are the first k elements of Λ_0. After obtaining weights of neighboring signatures, the estimated signature value is then calculated by Eq. (2). The minimized estimation variance which shows how reliable to estimate $\hat{S}_i$ is given as:

$$\sigma_k^2(G(S_i)) = \Lambda_0^T\gamma_0 = C_i^T\Omega C_i - \Lambda c_0 + \mu. \tag{7}$$

To be mentioned, weights and estimation variance are obtained from signature models which are free from HT effects. Given the estimated signature value and estimation variance, the confidence interval can be constructed which is used to determine the HT later.

The estimation function provides the smallest estimation variance for any signature with given neighboring signatures, indicated by objective function. From the estimation variance as shown in Eq. (7), it is obvious that the estimation variance does not include the inter-chip variation (σ_c^2). This is because that the inter-chip variation has identical effect on all gates in the chip and it is thus modeled as the part of unknown constant mean of the signature. The upper bound of the estimation variance is given as follows.

Theorem 1 *The upper bound of estimation variance given by the estimation function is two times of the smallest variogram between the estimated signature and its neighboring signatures.*

PROOF. Suppose the neighboring signature (S_{it}) has the smallest variogram with the estimated signature (S_i). Set weight 1 for S_{it} and weight 0 for remaining neighboring signatures. Under this weight arrangement, the estimation variance is equal to two times of variogram between S_i and S_{it}, $\sigma_k^2(G(S_i)) = var(S_i - S_{it}) = 2\gamma(S_i, S_{it})$. Since the above estimation variance is obtained under special weights, the minimized estimation variance should be less than or equal to two times of the smallest variogram between the estimated signature and its neighboring signatures. ∎

The bigger the correlation between two signatures, the smaller the variogram between them. This implies that the estimation process can greatly mitigate PV effects if neighboring signatures are highly correlated with to-be-estimated one. That is why we select the most correlated signatures with estimated one as neighboring signatures. Moreover, the estimation variance decreases with the increase of the size of neighboring signature set, but large-size set would incur high computational time.

Since weights and estimation variance are obtained based on the signature models which are free from HT effects, the existence of any HT-affect signature would impact the estimation process. We use this fact to detect the outlier.

4.3 The Impact of PV model and ME

Inaccurate PV model and measurement error would affect the performance of the estimation function. To address them, we adopt the conservative estimation, as detailed in the following.

The PV model is used to calculate variograms between signatures, and hence its error would affect the estimation process. The worst-case variogram is obtained by enlarging PV effects. According to [18], we increase the inter-chip variation and systematic variation by 5% and 3% and decrease all correlations between any two signatures by 3%. To consider ME in the estimation process, we model it as an additional noise (e_m) following the normal distribution ($e_m \sim N(0, \sigma_m^2)$) in the signature model. As a result, ME would only enlarge variograms by σ_m^2. As mentioned, enlarging variograms would not change estimated weights but change the estimation variance. With the ME, the estimation variance in Eq. (7) should be modified as: $\sigma_k^2(G(S_i^*)) = \Lambda_0^T\gamma_0^* = C_i^T\Omega C_i - \Lambda c_0 + \mu + (1 + \Lambda^T\Lambda)\sigma_m^2$, where γ_0^* includes ME. Since generally, $\Lambda^T\Lambda \ll 1$ under large number of weights, the estimation variance under ME is approximately equal to the sum of the estimation variance under noiseless condition and the variance of ME. The above equation implies that the estimation variance is dominated by the σ_m^2 when it is large. Since ME affects all HT detection techniques significantly, previous work adopted multiple measurements to mitigate it. By averaging over M measured values, the variance of ME becomes $\frac{1}{M}\sigma_m^2$.

5. HT DETECTION ALGORITHM

Considering the stealthy feature of HT, we assume designers cannot know which signatures can be HT-affected in the HT-inserted chip in the beginning. However, the existence of any HT-affect signature would impact the estimation process. To detect outliers, we compare each signature with corresponding user-defined confidence interval constructed according to estimated value and estimation variance. We consider a signature as a suspicious outlier if it is outside the user-defined confidence interval.

Algorithm 1: Procedure of HT Detection

```
1  //Determine suspicious outlier set Θ:
2  Normalize signatures and calculate their variograms;
3  do
4      foreach Ŝ_i ∈ Ŝ do
5          Calculate G(Ŝ_i) and construct Confidence Interval by Eq. (2);
6          Move Ŝ_i from Ŝ into Θ if Ŝ_i ∉ Interval;
7      end foreach
8  until (No more outliers are found);
9  //Determine signature outlier set Θ:
10 foreach Ŝ_j ∈ Θ do
11     Put Ŝ_j back to Ŝ, and initialize outliernum = 0;
12     foreach Ŝ_i ∈ Ŝ do
13         Calculate G(Ŝ_i) and construct Confidence Interval by Eq. (2);
14         outliernum++, if Ŝ_i ∉ Interval
15     end foreach
16     Ŝ_j is a true outlier if outliernum! = 0; otherwise it is not.
17 end foreach
```

Algorithm 1 presents the procedure of HT detection. In the first step, we calculate variograms between signatures (Line 2). Then, there is a loop that runs at least once. In each run, we validate each signature to determine whether it is a suspicious outlier according to the confidence interval (Line 6-8). However, a true outlier might be masked if its neighboring signature set contains outliers. In order to solve this problem, we verify remaining signature again, and stop the verification until no more suspicious outliers are found. After that, $\hat{\mathbf{S}}$ is guaranteed to contain HT-free signatures only. However, a suspicious outlier can be an HT-free signature, as it is possible that some of its neighboring signatures that contain the HT contribution influence the estimation process. In order to remove these false outliers, we do the one-by-one checking for each suspicious outlier (Line 12-25). We add one suspicious outlier back to $\hat{\mathbf{S}}$ each time. If there are no suspicious outliers detected, it means the suspicious outlier being checked is a false outlier; otherwise, it is a true outlier. By double checking, all outliers can be determined.

6. EXPERIMENTAL RESULTS

In this section, we compare the proposed *HTOutlier* technique against the direct comparison method and the self-referencing method. We do not compare with GLC-based methods because they require quite different signatures[2].

6.1 Experimental Setup

Experiments are conducted on several ISCAS benchmark circuits, whose layouts are generated with commercial physical design tools. HSPICE is employed to obtain the nominal values of each type of gates with the 70nm *Predictive Technology Model* (PTM). Monte Carlo simulation is used to estimate the PV effect on transient current. We simulate circuits and obtain the supply current under different test vectors. Similar to previous work [6, 7], the supply current is averaged among multiple (100) measurement results.

[2]When compared to our *HTOutlier* technique, the main limitation of GLC-based methods is their computational complexity.

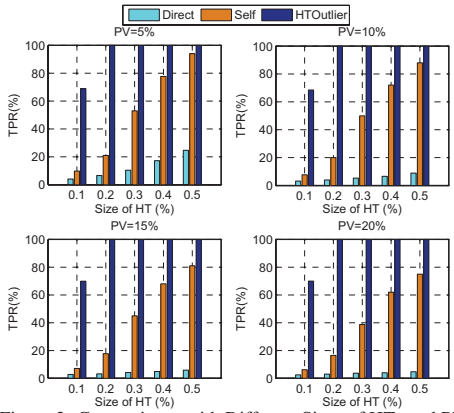

Figure 2: Comparisons with Different Sizes of HTs and PVs.

Similar to the experimental methodology employed in [6, 13, 7], we insert a number of gates into the original circuit as HTs (without concerning their malicious behavior), ranging from 0.1% to 0.5% of the total number of gates in the circuit. We adopt two performance metrics, *True Positive Rate (TPR)* and *True Negative Rate (TNR)* which present the proportion of HT-inserted chips correctly detected among all HT-inserted chips and the proportion of HT-free chips correctly detected among all HT-free chips, respectively. For fair comparison, we compare TPRs of the three methods under the same TNR under the same confidence level for HT-free signatures (95% in the experiment).

6.2 Results and Discussion

Circuits	TPR(%)		
	Direct	Self	*HTOutlier*
s1423	5.4	21.2	98.8
s5378	4.9	20.5	99.7
s9234	5.9	22.4	99.3
s13207	6.3	19.5	99.8
s15850	5.8	21.2	99.5
s35932	4.1	18.9	99.7
s38584	5.2	19.4	99.9

Table 1: Comparison Among Three Methods

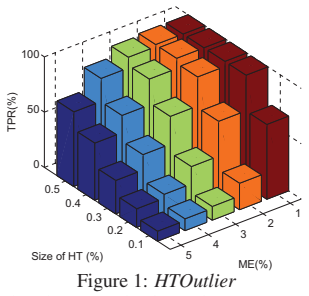

Figure 1: *HTOutlier*

Table 1 presents the TPRs of the three methods under 0.2% HT, 10% PV and 1% ME. As can be observed, *HTOutlier* greatly outperforms the other two methods because the impact of PV effects is alleviated by minimizing estimation variance. As expected, direction comparison results in the lowest HT detection capability without any PV-resistance property. Self-referencing method mitigates PV effects to some extent and hence achieves better resolution.

Next, we evaluate the performance of the three methods on circuit s38584 under different sizes of HT and diverse PVs, given 1% ME. From Fig. 2, we can observe that, with the increase of HT size, the TPRs of all the three methods increase accordingly. At the same time, we can see that, the TPR of *HTOutlier* quickly approaches 100% when the HT size is larger than 0.2%, while the TPR of self-referencing method ranges roughly between 80-90% even when the HT size is 0.5%. Direct comparison results in less than 20% TPR in almost most cases. Moreover, both the direct comparison method and self-referencing method have some performance degradation with the increase of PV while the *HTOutlier* does not, as shown in Fig. 2. This, again, indicates that the *HTOutlier* is highly resistant to PV effects. Since the self-referencing method has adopted an intuitive self comparison which to some extents reduces PV, it has smaller performance degradation than the direct comparison.

At last, we examine the influence of ME on the *HTOutlier*. As shown in Fig. 1, TPRs decrease with the increase of ME. when ME is 1%, *HTOutlier* has pretty good detection performance. Under 2%, 3% and 4% ME, the TPR is still very high when the size of HT is large (0.5%,0.4%), but it is lower than 20% when the size of HT is 0.1%. When the ME continues to increase up to 5%, *HTOutlier* can still achieve around 60% TPR under 0.5% HT. This result is really important to guide designers to increase detection ability of *HTOutlier* by reducing ME through multiple measurements.

7. CONCLUSION

In this paper, we formulate the HT detection problem as a signature outlier identification problem, and propose a novel *HTOutlier* technique to solve it, by exploiting the inherent correlations among signatures from the same chip. Experimental results demonstrate that *HTOutlier* is effective for HT detection in the presence of process variation and measurement error.

8. ACKNOWLEDGEMENTS

This work was supported in part by the General Research Fund CUHK418111 from Hong Kong SAR Research Grants Council (RGC) and in part by RGC Grant Direct Allocation under grant No. 2050488.

9. REFERENCES

[1] M. Tehranipoor and F. Koushanfar, A survey of hardware Trojan taxonomy and detection. In *IEEE Design and Test of Computers*, vol. 27, pp. 10–25, 2010.

[2] R. Karri, et al., Trustworthy hardware: Identifying and classifying hardware Trojans. *Computer*, vol. 43, pp. 39–46, Oct. 2010.

[3] R. Chakraborty, S. Narasimhan, and S. Bhunia, Hardware Trojan: Threats and emerging solutions. In *IEEE International High Level Design Validation and Test Workshop (HLDVT)*, pp. 166–171, Nov. 2009.

[4] Y. Jin and Y. Makris, Hardware Trojan detection using path delay fingerprint. In *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 51–57, 2008.

[5] D. Rai and J. Lach, Performance of delay-based trojan detection techniques under parameter variations. In *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 58–65, 2009.

[6] Y. Alkabani and F. Koushanfar, Consistency-based characterization for IC Trojan detection. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pp. 123–127, 2009.

[7] D. Du, et al., Self-referencing: a scalable side-channel approach for hardware Trojan detection. In *Proc. International Conference on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 173–187, 2010.

[8] M. Potkonjak, et al., Hardware Trojan horse detection using gate-level characterization. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 688–693, 2009.

[9] D. Agrawal, et al., Trojan detection using IC fingerprinting. In *Proc. IEEE Symposium on Security and Privacy*, pp. 296–310, 2007.

[10] M. Banga and M. S. Hsiao, A region based approach for the identification of hardware Trojans. In *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 40–47, 2008.

[11] S. Narasimhan, et al., Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach. In *Proc. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pp. 13–18, 2010.

[12] S. Wei, S. Meguerdichian, and M. Potkonjak, Gate-level characterization: foundations and hardware security applications. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 222–227, 2010.

[13] S. Wei and M. Potkonjak, Scalable segmentation-based malicious circuitry detection and diagnosis In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pp. 483–486, 2010.

[14] H. Salmani, M. Tehranipoor, and J. Plusquellic, A layout-aware approach for improving localized switching to detect hardware Trojans in integrated circuits, In *Proc. IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, 2010.

[15] V. Chandola, A. Banerjee, and V. Kumar, Anomaly detection: A survey. In *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, July 2009.

[16] N. Cressie, Statistics for spatial data. In *Terra Nova*, vol. 4, no. 5, pp. 613–617, 1992.

[17] S. Sarangi, et al., VARIUS: A model of process variation and resulting timing errors for microarchitects. In *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 3–13, 2008.

[18] J. Xiong, V. Zolotov, and L. He, Robust extraction of spatial correlation. In *Proc. International Symposium on Physical design (ISPD)*, pp. 2–9, 2006.