Consistency-based Characterization for IC Trojan detection

Yousra Alkabani
CS Dept., Rice University
6100 Main St., MS-132
Houston, TX 77005
yousra@rice.edu

Farinaz Koushanfar
ECE and CS Dept(s)., Rice University
6100 Main St., MS-380
Houston, TX 77005
farinaz@rice.edu

ABSTRACT

A Trojan attack maliciously modifies, alters, or embeds unplanned components inside the exploited chips. Given the original chip specifications, and process and simulation models, the goal of Trojan detection is to identify the malicious components. This paper introduces a new Trojan detection method based on nonintrusive external IC quiescent current measurements. We define a new metric called *consistency*. Based on the consistency metric and properties of the objective function, we present a robust estimation method that estimates the gate properties while simultaneously detecting the Trojans. Experimental evaluations on standard benchmark designs show the validity of the metric, and demonstrate the effectiveness of the new Trojan detection.

1. INTRODUCTION

Continuous miniaturization of CMOS to new technologies require costly upgrades of manufacturing technologies and processes. Due to this high cost, in today's semiconductor business model IP providers, IC designers, and foundries are separate companies often located in different parts of the World. Some of the potential threats of this business model includes unauthorized use and theft of IPs, piracy of ICs, and addition of Trojans. A Trojan component maybe inserted in the design to enable the attacker to monitor, control, or steal information from the chip, or maybe used to remotely enable or disable all or parts of the chip. Establishing the IC trust in face of offshore and untrusted fabrication is a challenging and important problem especially since the ICs are the kernels of businesses, government, and defense in the modern world.

Detecting Trojans post fabrication encounters a number of challenges. First, the number of gates in a design has been exponentially growing, while the number of external pins has only linearly increased. Thus, there is a limited controllability and observability to the complex internals of ICs. Second, there are many opportunities for Trojan insertion at the various fabrication steps. Since the foundries are equipped with advanced technologies, processes and state-of-the-art facilities, they are able to strategically plan and hide Trojans. Third, the increasing amplitude of process variations introduces uncertainty in the measured characteristics, further complicating the Trojan (change) detection process. Forth, classic testing methods are insufficient for Trojan detection since they assume a Trojan-free netlist.

Employing noninvasive IC measurements has been shown to be effective and promising in detecting some of the possible IC Trojans. The use of transient power side-channel signals for Trojan detection was investigated [1, 4, 10]. A number of statistical methods for change detection by hypothesis testing, principal component extraction, calibration, and empirical sensitivity analysis were proposed for transient power and timing measurements [1, 6, 11]. Much more research and development is still needed for finding scalable and cost-effective Trojan testing and identification methods, establishing detection bounds, and improving statistical detection, calibration, and sensitivity analysis.

This paper presents a novel algorithm for detecting Trojans based on quiescent (static) current measurements. We formally define the problem and establish its complexity. Linear translation of the measurements to gate-level characteristics is demonstrated. We introduce a measure for the integrity of the gate-level estimation called *consistency*. Our contributions are as follows. (i) We present a formal formulation for detecting Trojans by quiescent current measurements and analysis. The problem is shown to be NP-complete. (ii) We define the consistency metric and propose an efficient iterative algorithm that exploits the consistency measure for finding the anomalous gates. (iii) We study the performance of the algorithm for different measurement errors and the Trojan sizes. (iv) We show how the signature of an added Trojan can be used for classifying the newly tested chips that are similarly exploited. (v) We introduce a method for calibration of non-random process variations. The method exploits the properties of the Trojan gradient and its differences with the spatial frequency of correlated variations. (vi) Experimental evaluations on benchmark circuits with different Trojan sizes show that the method is efficient for Trojan detection and robust against measurement noise and process variations.

2. PRELIMINARIES

Testing for Trojan. We rely on unobtrusive leakage current (a.k.a. I_{DDQ} or quiescent power supply current) tests for Trojan identification. The I_{DDQ} testing has been shown to be an effective method for detecting the faults in CMOS

ICCAD'09, November 2–5, 2009, San Jose, California, USA.

circuits [12]. While the objective of Trojan detection and the Trojan models are different than the stuck-at, open, and bridge fault models that are classically addressed by I_{DDQ} tests, the ATPG input vectors for the tests are applicable. The measurements are assumed to be stationary and the measurement noise is assumed to be independently and identically distributed.

Adversarial model. In our model, Trojan IC has the same set of input/output pins as the original design and the same form factor. The assumption is that all the ICs have already passed the standard parametric and functional tests and do not have known defects or faults, including stuck-at faults, open faults, or bridging faults [5]. In this paper, we use the terms anomalous, Trojan, and *abnormal* interchangeably.

3. RELATED WORK

Wang et al. [14] discussed a number of key methods for IC modifications and provide a general framework for the hardware Trojan attack classification. The first comprehensive Trojan detection study was done by Agrawal et al. who used the transient power signal analysis (dynamic current or I_{DDT}) to detect the Trojan alteration [1]. The technique first destructively tested a control group of (assumed to be) unaltered ICs to extract the fingerprint for the normal chip behavior. The rest of the chips were tested against the extracted fingerprints by using standard hypothesis testing against the principal components (a.k.a Karhunen-Love transform). The drawbacks of the approach was the overhead of testing destructively both in terms of time and expense, and noise/process variation (PV) sensitivity. A region-based testing approach was proposed by Banga et al. [4]; they use the dynamic power signatures to identify the regions with abnormal behavior and then perform closer tests in those regions. The behavior abnormality is studied by comparing to the simulated intended waveform. Power supply transient analysis for Trojan detection was also investigated by Rad et al. [10, 11]. In their method, the supply current is measured from multiple ports to overcome the small Trojan ratio to the background current and is then calibrated for PV. Calibration is done by transforming and comparing the test IC currents to those produced by Trojan-free simulation models. Gate-level estimation as a method for post-silicon IC characterization has been introduced earlier [2, 3, 13, 9]. Potkonjak et al. combined the gate-level estimation with constraint manipulation to detect Trojans using delay and leakage side channels [9]. Our consistency-based detection provides a sound formal basis for robust estimation of the gate-level characteristics and Trojan detection.

The use of path-delay fingerprints was proposed by Jin and Markis [6] who employed principal component analysis to characterize the original circuit and used the distance of the tested paths to the principal components to find the Trojan. This approach may require a large number of path measurements. Wolff et al [15] suggest an approach that first identifies a set of target "hard-to-observe" sites for a Trojan and then uses ATPG to generate patterns to activate the Trojan. This approach could be efficient for Trojans that have a few inputs. The analysis complexity and test set size would render it impractical for larger Trojans. While the existing approaches have shown promise in their ability to detect deviations from the original design plan, the focus have been on detection heuristics. This paper provides a formal treatment of detection formulation by analyzing the complexity of the problem, and providing new consistency metrics that are efficient in addressing the problem.

4. TROJAN DETECTION

Our Trojan detection method is built upon the gate leakage estimation. The first step is to generate the input vectors that enable leakage current (I_{DDQ}) measurements. Next, we apply the measurement vectors and map the measured values to gate leakages (Section 4.1). The anomalies are detected by comparing to the Trojan-free nominal simulation values while considering PV. Our consistency-based Trojan detection algorithm use the properties of the optimization objective and employ the gate leakage estimations, sensitivity analysis, and calibration for PV.

4.1 Gate leakage estimation

Problem. GATE LEAKAGE ESTIMATION.

Given: A combinational IC with fully available netlist and the CMOS technology process design kit and simulation models. The original design plan has K gates $G_1, \ldots, G_K$, N_{in} primary inputs, and N_{out} primary outputs. The gates are single-output and each gate may implement an arbitrary logic function. The nominal leakage values of each gate for each input is available from the simulation models. Such models are standard and are given to the designers to ensure the competency of the design with the process.

On each IC, we test a set of J input vectors $\nu_1, \ldots, \nu_J$ where each input vector is a tuple with cardinality N_{in} with elements in {0,1}. An input vector changes the internal state of the gates whose input is affected by the input vector transition. For each input, the leakage current at the external power supply pin is measured and recorded.

Objective: For each tested chip, estimate the gate leakage values for the incident input vectors.

To address the problem, the overall measured current at the output pin is written as the sum of the gate leakages (the impact of interconnect leakages could also be included but we omit it because of its low amplitude compared to gate leakages). The gate leakages are inherently unique at each IC post-fabrication due to PV. Thus, for the input ν_j, the total measured leakage ($I^{meas}_{leak}(\nu_j)$) can be written as:

$$\forall \nu_j \qquad I^{meas}_{leak}(\nu_j) + \epsilon(\nu_j) = \sum_{k=1}^{K} I^{k}_{leak}(q_{jk}) \qquad (1)$$

where $\epsilon(\nu_j)$ is the measurement error for the input ν_j, q_{jk} is the input at the gate G_k after applying ν_j to the primary inputs, and $I^k_{leak}(q_{jk})$ is the leakage of gate G_k with q_{jk} as its inputs. By taking multiple measurements for each input one would be able to reduce the random measurement error, but would increase the test time.

Assuming the gate's nominal leakage value for each input combination is known from the simulations, the objective can be transformed to finding the deviations in gate leakages from their nominal values. If the gate G_k has Q inputs, it would have 2^Q input combinations and a leakage value corresponding to each input combination. Let the nominal value for $I^k_{leak}(q_{jk})$ from the simulation models is $I^{k(nom)}_{leak}(q_{jk})$. On each IC, for the gate G_k we define a *scaling factor* denoted by ϕ_k quantifying G_k's leakage deviation from the nominal

values as follows:

$$\forall q_{jk} \qquad I^{k}_{leak}(q_{jk}) = \phi_k \;\; I^{k(nom)}_{leak}(q_{jk}). \tag{2}$$

The assumption is that the leakage of one gate is scaled the same way over all input combinations, when compared to the nominal values. Equation 2 becomes:

$$\forall \nu_j \qquad I^{meas}_{leak}(\nu_j) + \epsilon(\nu_j) = \sum_{k=1}^{K} \phi_k \;\; I^{k(nom)}_{leak}(q_{jk}). \tag{3}$$

Assume for now that the given input vector set of cardinality J has a full I_{DDQ} test coverage. One can write a convex quadratic program with linear constraints (C's) dictated by the leakage summations and the objective function (OF) of minimizing the square root of measurement noise (MSE(E)), i.e.,

$$OF: \qquad \min \sqrt{\textstyle\sum_{j=1}^{J} \epsilon(\nu_j)^2} = \min \;\; MSE(E) \tag{4}$$

$$C's: \;\; \forall j \;\; I^{meas}_{leak}(\nu_j) + \epsilon(\nu_j) = \textstyle\sum_{k=1}^{K} \phi_k \;\; I^{k(nom)}_{leak}(q_{jk}),$$

where E is a vector with cardinality J with elements $\{\epsilon(\nu_j)\}_{1 \le j \le J}$. Assuming that the above system is full-rank, one can always solve the equations where the unknowns are the scaling factors of the gates and the measurement errors.

The input vectors should be such that each gate at least once changes its state (i.e., is controllable), independent of any other gates in the circuit. If this condition is not satisfied and some gates are collinear (i.e., the input to those gates are not independently changeable), we lump the collinear gates together in our equation and form a smaller subspace of the coefficient matrix which is full-rank. In this case after the gate leakages are found, the impact of the lumped gates can be quantified together. We emphasize that the quadratic convex optimization defined by Equation 4, works optimally for a full-ranked matrix, but the solvers can find numerical approximations for many cases of the matrices that are not full rank. To maximize controllability, we employ the known high coverage input generation methods for I_{DDQ} testing [5]. We note that one of the major advantages of I_{DDQ} testing compared with the timing tests is its high coverage.

Since the measured leakage is an additive function of the circuit components, and the Trojan is not known to the simulation models, the impact of an inserted Trojan would change the estimates of the overall gate leakages. We use this fact to detect the Trojan.

Problem. TROJAN DETECTION BY GATE LEAKAGE ESTIMATION.

Given: The same inputs as Section 4.1.

Objective: Estimate the leakage value for each gate on each IC and identify the gates with anomalous (abnormal) leakage values.

An abnormal gate is the one which has a large deviation in its estimated leakage characteristics compared with its nominal value from the simulations. To quantify this deviation, we use the Euclidean distance between the estimated gate leakage and its nominal value. Identification of the abnormal gates is an integer problem combined with the quadratic problem of gate leakage estimation. Thus, we require a mixed integer optimization problem. In fact, this optimization is in the form of a robust estimation and distance-based outlier detection problem where the MSE distance between the outlier and the estimated values from non-outliers are used for distinguishing and removing the outliers (anomalies) [8]. The outlier distance to the estimation is removed by reweighing to nominal values. Such problems contain an uncertainty about the values of the variables and the interval of the benign variables (because of the measurement error and PV) and have been shown to be NP-complete [7].

4.2 The detection algorithm

In this section, we develop efficient heuristics to address the above complex problem. For the MSE estimation, an *influence function* (IF) is defined which measures the impact of a single anomaly on the estimator standardized by the proportion of contamination. A bounded IF is a desirable attribute since unbounded IF allows the impact of anomalous observations to grow. Unfortunately, the IF for an MSE estimator is well known to be proportional to the size of the anomaly; meaning that a highly discrepant measurement can completely destroy the MSE estimation. To alleviate this problem, one option is to use as an error measure a metric which is not sensitive to outliers e.g. median errors. However, the median is known to generate much higher inference error. Furthermore, under the assumption of Gaussian errors and no outliers the MSE estimator is the best known for minimizing the estimation error [8]. One solution for decreasing the influence of outliers and to still retain the desirable estimation properties of MSE is to use iterative MSE estimation where in each iteration the scaling factors are reweighed and adjusted. We opt to use Gaussian kernel function for iterative reweighing. The kernel function is 1 minus Gaussian distribution with a mean of 1. The variance of the Gaussian is chosen such that the weights given to the scaling factors within the expected range of process variation are close to zero, and weights given to points that are further from the expected range are close to 1 (nominal scaling factor).

The details of the detection algorithm are shown in pseudocode 1. The inputs to the problem are the combinational circuit, non-invasive leakage measurement for J input vectors, the nominal gate leakage values, and the minimum required improvement in consistency Δ^2_{th}. The outputs of the algorithm are the scaling factors of the gates and the set of anomalous gates. In Step 1, we use the optimization formulation in Equation 4 to estimate the gate leakages in form of scaling factors. Step 2 calibrates the scaling factors for inter-chip and intra-chip correlations. The *consistency* of estimation is initialized to 0 and the iteration counter is set to 1 in Step 3. Consistency and calibration will be described in details later in this section. In Steps 4-9, there is a loop that runs at least once. In each run, the gate scaling factors are reweighed using the Gaussian kernel and the measurement relationships are adjusted for reweighing. In Step 7, we do re-estimation of the scaling factors. Then, we update the iteration counter in Step 8. The loop terminates at Step 9 if the improvement in the consistency (Δ^2(consistency)) is less than Δ^2_{th}. Finally, anomalous gates are identified as the gates with highest change in scaling factors ($\phi^0 - \phi^i$).

The overall complexity of the algorithm is dominated by the time needed for solving a quadratic convex program. It is possible to further reduce this complexity but we did not include the results in this paper because of space constraints. For example, using a linear objective function would reduce the complexity of the scaling factors estimation to linear time.

Pseudocode 1 - Trojan detection

```
use equation 4 to estimate the scaling factors (φ^0);
calibrate the scaling factors;
consistency=0; i=1;
do
        use the kernel to reweigh the scaling factors;
        adjust the measurements accordingly;
        re-estimate the scaling factors (φ^i);
        i++;
while ( Δ^2(consistency) ≥ Δ^2_th);
identify the anomalous gates by comparing φ^i, φ^0;
```

• **The consistency metric.** Our assumption is that measurement errors are i.i.d Gaussian random variables, furthermore we assume that the variations in gate scaling factors are also i.i.d distributed according in $\mathcal{N}(0, \sigma^2)$. The consistency metric measures the distance between the real scaling factors and the estimated ones. The assumption of Normal distribution is based on the distance from real values of scaling factors. We apply the same distribution to approximate the distribution of benign scaling factors. The consistency metric is the square distance between the initial estimate and the i^{th} iteraton value, i.e., $\Delta^2(consistency) = \sum_{k=1}^{K}(\phi_k^0 - \phi_k^i)^2$. Assuming that the deviation in the gates scaling factors from their nominal values is only due to random process variations (systematic variations are calibrated) and the anomalous gates are reweighed, we derive the following lemma for calculating the distribution of Δ^2(consistency).

Lemma 1. The probability distribution of Δ^2(consistency) has a Chi-square (χ^2_K) distribution with K degrees of freedom (DF).

Proof. Follows immediately from the fact that the measurement errors are distributed i.i.d. Gaussian. □

For a large K (K $\geq$ 30) this distribution can be safely approximated by a Gaussian. The Δ^2_{th} is set to be twice the variance of the estimator's distribution.

• **Calibration.** A key step in anomaly detection algorithm is adjusting the scaling factors such that the estimated values are not impacted by the PV. The PV components: random, inter-chip correlations, and intra-chip correlations. Since the average scaling factor is 1, the inter-chip correlations can be calibrated by shifting the scaling factor values of all gates on one IC to have a mean value of 1. The intra-chip correlations are spatial. Based on our studies, the rate of spatial change in leakage characteristics because of even a small Trojan addition (i.e., Trojan gradiant) across the chip layout is much sharper than the rate of spatial change in leakage characteristics due to the intra-chip correlations. Note that the most challenging types of Trojans are the small ones: if a larger Trojan trying to emulate the process variations is inserted in the design, it would be easy to detect. The difference between the added Trojans gradient and the intra-chip correlations suggests employing a filter over the discrete 2D space of the IC's layout that removes the soft edges caused by the intra-chip correlation. Thus, the filter should be a 2D highpass filter over the chip layout space. Note that the random variations would have an extremely high frequency and would not be adjusted using this method. After shifting the mean of the scaling factors and applying the high-pass filtering method on the spatial variation of the scaling factor values, the impact of the systematic inter-chip and intra-chip correlations would be removed. The only PV component that would remain is the random variations.

5. EXPERIMENTAL EVALUATIONS

In this section, we evaluate the detection methods on standard MCNC'91 benchmarks. The circuits are synthesized and mapped using ABC synthesis tool. The library used for mapping includes 2-input, 3-input, and 4-input NAND and NOR gates, in addition to inverters. The nominal values of the leakage current of the different gates are estimated using HSPICE. Placement is done by the Dragon tool. TetraMAX ATPG is used for IDDQ test generation. Matlab is used to perform the simulations and solve the quadratic programs (QPs). The process variation is applied as follows: 12% random variations, 60% intra-die variation with 60% correlation of the total variation, 20% of the total variation is uncorrelated intra-chip variation and the remaining variation is inter-chip variation.

Using the QP formulation described in section 4.1, we evaluate the scaling factors of the different gates in a circuit while optimizing MSE error. The average percentage MSE error in estimation is evaluated in Table 1, for different circuits. The first column represents the name of the circuit denoted by *Ct*. The second column *size* shows the number of gates in the circuit, while the next two columns *#inputs* and *#outputs* are the numbers of inputs and outputs. The last three columns show the percentage error in characterization while imposing 3%, 5%, and 10% measurement noise. The number of measurements is double the size of each benchmark. On the average, the error is 4%, 5.8%, and 10.3% in the case of 3%, 5%, and 10% measurement noise, respectively. In general, increasing the number of measurements improves the characterization error. For instance, employing half of the measurements used in Table 1 on average increases the characterization error by 1%.

Table 1: Gate characterization error vs. measurement noise.

Ct	size	inputs	outputs	3%	5%	10%
c8	165	28	18	5.6	7	11.6
C432	206	36	7	1.7	3.5	7.2
C880	353	60	26	2.9	5.1	11
C1355	512	41	32	8.5	10	12.1
C499	532	41	32	2.9	4.5	9
C1908	615	33	25	3.2	4.9	10.5
C3450	1131	50	22	4	5.9	9.8
C5315	1796	178	123	3.1	5.5	10.8

The QP formulation is used as a part of the detection algorithm described in section 4.2. The scaling factors from the QP are filtered using the existing Matlab 4^{th} order high-pass filter to remove the effect of systematic variations.

The iteration number is shown on the x-axis and the value of the consistency metric is on the y-axis. Three cases are shown: the case with no Trojan (denoted by *no Trojan*), the case with one extra gate (denoted by *1 gate*), and when adding three extra gates (denoted by *3 gates*). The results are smoothed over 100 runs of different PV simulations for the same benchmark and the same Trojan. It can be seen on the figure that the consistency function is monotonic and non-decreasing. The slope of the consistency curve increases as the number of Trojan gates increase.

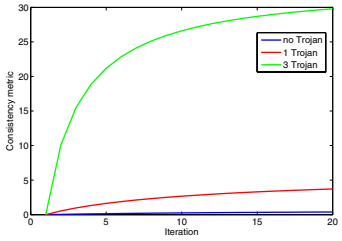

Figure 1: Δ^2(consistency) versus iteration.

Our clustering results show close vicinity of the anomalous gates to the inserted Trojan with a median cluster size of 1 or 2 on the benchmark circuits, for one added Trojan gate and 100 random PV instances on each benchmark. We also studied the Trojan signature similarities across the identically attacked chips. Trojan detection was run on 50 chips with no Trojans and 50 chips with 1 Trojan inserted in the same place in circuit C432. Each IC was an instance of the PV model. Figure 2 shows the boxplots of the scaling factors for G_1 and G_2, the two gates labeled the most as anomalous. The scaling factor of G_1 and G_2 in Trojan-free case are shown on the first and third box, and the Trojan cases are shown on the second and forth box respectively. The median of scaling factor (shown in the middle of the box) is close to 1 for no Trojan and around 1.5 with the Trojan. Using the differences, one could classify new chips based on the signatures from the scaling factors of only a few gates. For example, for 100 new ICs with 1 Trojan, we were able to classify 99% of the ICs correctly by comparing the scaling factors of the two gates. The classification decision is made by maximum likelihood; the scaling factors probabilities for Trojan or no Trojan cases are extracted from the boxplots in Figure 2.

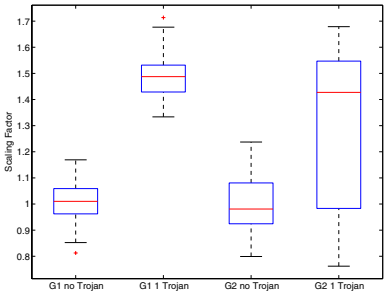

Figure 2: Scaling factors for two detected anomalous gates for 1 gate Trojan in C432.

6. CONCLUSION

We have introduced a formal Trojan detection method that uses noninvasive measurements of static current for performing gate leakage estimation. We formulated the Trojan detection as an optimization problem for minimizing the estimation error. We defined a consistency metric based on the expected error distribution. We showed that the problem is NP-complete and developed an efficient algorithm that iteratively improves the consistency in the estimation error. Calibration for inter-chip and intra-chip correlations was done by shifting the mean values and filtering the low frequency correlations. Experimental evaluations on benchmarks confirmed that the method is efficient for Trojan detection and robust to measurement noise and process variation.

Acknowledgment

This work is partly supported by the NSF Career Award number 0644289, and by ONR YIP grant N000140910831.

7. REFERENCES

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using ic fingerprinting. In *IEEE S& P*, pages 296–310, 2007.

[2] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak. Trusted integrated circuits: A nondestructive hidden characteristics extraction approach. In *Information Hiding*, pages 102–117, 2008.

[3] Y. Alkabani, F. Koushanfar, and M. Potkonjak. Input vector control for postsilicon leakage current minimization in the presence of manufacturing variability. In *DAC*, 2008.

[4] M. Banga and M. Hsiao. A region based approach for the identification of hardware trojans. In *HOST*, pages 43–50, 2008.

[5] N. Jha and S. Gupta. *Testing of Digital Systems*. Cambridge University Press, 2003.

[6] Y. Jin and Y. Makris. Hardware trojan detection using path delay fingerprint. In *HOST*, pages 51–57, 2008.

[7] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer Academic Publishers, Dordrecht, 1997.

[8] A. M. L. P. J. Rousseeuw, editor. *Peter J. Rousseeuw, Annick M. Leroy*. Wiley, 2003.

[9] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware trojan horse detection using gate-level characterization. In *DAC*, 2009.

[10] R. Rad, J. Plusquellic, and M. Tehranipoor. Sensitivity analysis to hardware trojans using power supply transient signals. In *HOST*, pages 3–7, 2008.

[11] R. Rad, X. Wang, J. Plusquellic, and M. Tehranipoor. Power supply signal calibration techniques for improving detection resolution to hardware trojans. In *ICCAD*, pages 632–639, 2008.

[12] S. Sabade and D. Walker. IDDX-based test methods: A survey. *ACM Trans. Design Automation of Electronic Systems*, 9(2):159–198, 2004.

[13] D. Shamsi, P. Boufounos, and F. Koushanfar. Noninvasive leakage power tomography of integrated circuits by compressive sensing. In *ISLPED*, pages 341–346, 2008.

[14] X. Wang, J. Plusquellic, and M. Tehranipoor. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *HOST*, pages 15–22, 2008.

[15] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty. Towards trojan-free trusted ICs: problem analysis and detection scheme. In *DATE*, pages 1362–1365, 2008.