Hardware Trojan Horse Benchmark via Optimal Creation and Placement of Malicious Circuitry

Sheng Wei† Kai Li‡ Farinaz Koushanfar‡ Miodrag Potkonjak†

†Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095
{shengwei, miodrag}@cs.ucla.edu

‡Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005
{kai.li, farinaz}@rice.edu

ABSTRACT

This paper proposes Hardware Trojan (HT) placement techniques that yield challenging HT detection benchmarks. We develop three types of one-gate HT benchmarks based on switching power, leakage power, and delay measurements that are commonly used in HT detection. In particular, we employ an iterative searching algorithm to find rarely switching locations, an aging-based approach to create ultra-low power HT, and a backtracking-based reconvergence identification method to determine the non-observable delay paths. The simulation results indicate that our HT attack benchmarks provide the most challenging representative test cases for the evaluation of side-channel based HT detection techniques.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Physical Security*

General Terms

Security

Keywords

Hardware Trojan, benchmark, process variation, gate-level characterization

1. INTRODUCTION

1.1 Motivation

Hardware Trojans (HTs) are malicious components within integrated circuits (ICs) embedded by an untrusted foundry during the manufacturing process. HT detection has recently drawn a great deal of attention because IC outsourcing has become a trend in IC industry; the security and integrity of the manufactured ICs are of great concerns [16].

DAC 2012, June 3-7, 2012, San Francisco, California, USA.

Since DARPA issued its first call for the study of hardware systems security in 2005 in general and hardware Trojans in particular, over a hundred HT detection techniques have been proposed [15]. Among them, side-channel based HT detection based on power and delay monitoring has become the most focused area [2, 3, 10, 11, 12, 21]. However, until now, there are no standard and publicly accepted test benchmarks for evaluating the various side-channel-based HT detection solutions, making it difficult for IC design companies or researchers to select the most effective HT detection solutions or to compare them.

This paper develops the first systematic way of placing HTs in a target design, in such a way that the most challenging HT test benchmarks can be created. Furthermore, we develop a complete set of HT metrics that can be used to evaluate the difficulty of detecting an arbitrary HT placement. Our idea for HT benchmark creation is to hide the malicious circuitry inside an IC, where the activities of the HTs are either low or unobservable due to the limitations of the side-channel measurements. Our observation is that there are three main conventional test modes that are used for IC testing and HT detection: switching power, leakage power, and delay. Therefore, we create HTs that either do not have an impact or have an exponentially low probability of impacting any of the three test modes.

The impact of HTs can be hidden within the fluctuations due to process variation (PV); its impact can be so small that it is below the sensitivity of modern instruments. There are three key observations behind this claim. The first is that there are many common design structures that have reconvergent paths with high delay discrepancies. Hence, the attacker can easily add malicious circuitry in such a way that the external delay measurements are not impacted. A small example in Figure 1 illustrates this situation. Since path B through gates 1-3-4-5-6-7 is much longer than path A (1-2-H-7), the addition of gate H cannot be detected using delay measurements from inputs $I1$ and $I2$ to output O.

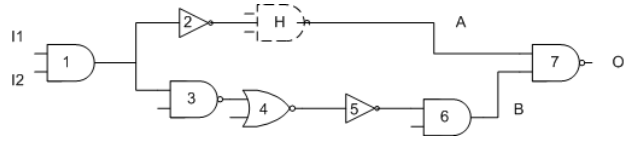

Figure 1: Example of a HT attack with no delay impact.

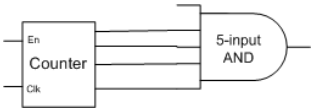

Figure 2: Example of a HT attack with no switching power impact.

Similarly, as shown in Figure 2, it is easy to embed a HT that is activated only on a very rare event (combination of its inputs). The rare event can be triggered only after the IC is active for much longer than any standard testing time. This example shows an AND gate that has many inputs from the most significant bits of a large modulo counter.

Until now, HT detection using leakage current was considered to be the most reliable technique. This is because any gate is subject to several types of leakage energies regardless of its location, inputs, and activation pattern. For example, the subthreshold leakage energy of a gate is given by the following formula [13]:

$$P_{leakage} = 2 \cdot n \cdot \mu \cdot C_{ox} \cdot \frac{W}{L} \cdot (\frac{kT}{q})^2 \cdot D \cdot V_{dd} \cdot e^{\frac{\sigma \cdot V_{dd} - V_{th}}{n \cdot (kT/q)}} \quad (1)$$

where W is gate width, L is gate length, V_{th} is threshold voltage, V_{dd} is supply voltage, n is subthreshold slope, μ is mobility, C_{ox} is oxide capacitance, D is clock period, ϕ_t is thermal voltage $\phi_t = kT/q$, and σ is drain induced barrier lowering (DIBL) factor.

Analysis of the formula indicates that leakage energy depends exponentially on the difference between the supply and threshold voltages. Therefore, an attacker can make the leakage of the HT gates negligibly small, by either resizing W and L or by creating gates that use only high V_{th} transistors. Hence, those gates will have several orders of magnitude lower leakage than regular gates and would be difficult to detect via leakage power measurements.

Our techniques create new types of exceptionally powerful and difficult to detect HTs. The key idea is to use power (or clock) gating in such a way that, in the default mode, the HT is powered off. The HT is activated by a single AND gate whose output is difficult to set to the value '1' (activation condition) by anybody except the attacker. The gate is intentionally aged or implemented to have a very high threshold voltage that corresponds to ultra (exponentially) low leakage energy, which cannot be detected even by state-of-the-art instruments. Therefore, it cannot be detected by any technique that measures switching and/or power. Finally, the HT is placed in such a way that it does not have an impact on delay between any pair of flip-flops. The HT is activated by an input vector sequence that is known by the attacker but otherwise has an exponentially low probability of occurrence.

2. RELATED WORK

2.1 Process Variation

Process variation is the deviation of IC parameter values from nominal specifications after the manufacturing process [4]. There is a wide consensus that PV is the dominant source of challenging problems related to HT detection. For example, it automatically invalidates all approaches based on the existence of a golden model that aim to compare an IC under analysis to its HT-free version. It also easily explains the discrepancy of any global measurement as a consequence of the embedded HT. Recently, a significant progress has been made in the field of HT detection using gate-level characterization techniques where a large set of global measurements is used to calculate variation-dependent characteristics of each gate post-silicon [14, 17, 18, 19, 20, 22]. Therefore, the variability aspects of PV are now addressed well.

However, PV has one more important consequence. It can produce HTs that have such a small impact on the measurements that they are below the sensitivity of modern instruments. Hence, the attacker can easily add malicious circuitry in such a way that external variation caused by the HTs is hidden in the presence of PV.

2.2 Hardware Trojan Detection

Tehranipoor et al. [15] provided a comprehensive survey of HT detection. There are three HT detection techniques that are most relevant to our HT benchmark creation. Jin and Makris proposed using the statistical delay path analysis in wireless cryptographic circuits [8]. However, as our earlier example showed, it would be easy to place HTs that do not impact any delay paths. Also, the attacker can easily resize gates in such a way that HTs are completely hidden. Kim et al. [9] advocate run time detection of HTs after their activation by observing the system bus behavior. Unfortunately, one can easily create HTs that do not alter the system bus characteristics. Agrawal et al. [1] construct IC fingerprints using side-channels (e.g., power and temperature) for a given design and authenticate the IC instances by comparing the fingerprints. However, while their technique does not take PV into consideration, we fully investigate and integrate the PV impact in our benchmark creation process.

3. PRELIMINARIES

There are typically two possible sources of power dissipation on an IC. One is from gate switching (also termed switching power or dynamic power), where the ICs dissipate power by charging the load capacitances of wires and gates. The other source is static power (also termed leakage), where the gates dissipate power due to the leakage current even if they do not switch. The gate-level leakage model is presented in Equation (1). The gate-level switching power model [13] is described by equation (2), where the switching power is dependent on switching probability α, gate width W, gate length L, and supply voltage V_{dd}:

$$P_{switching} = \alpha \cdot C_{ox} \cdot W \cdot L \cdot V_{dd}^2, \quad (2)$$

We use the delay model in [13] that relates the gate delay to its sizing and operating voltages:

$$Delay = \frac{k_{tp} \cdot k_{fit} \cdot L^2}{2 \cdot n \cdot \mu \cdot \phi_t^2} \cdot \frac{V_{dd}}{(ln(e^{\frac{(1+\sigma)V_{dd} - V_{th}}{2 \cdot n \cdot \phi_t}} + 1))^2} \cdot \frac{\gamma_i \cdot W_i + W_{i+1}}{W_i}, \quad (3)$$

where subscripts i and $i+1$ represent the the driver and load gates, respectively; γ is the ratio of gate parasitic to input capacitance; and k_{tp} and k_{fit} are fitting parameters.

Furthermore, we employ the aging model proposed in [5] for our IC aging-based HT creation. The time dependence of V_{th} shifts due to negative bias temperature instability

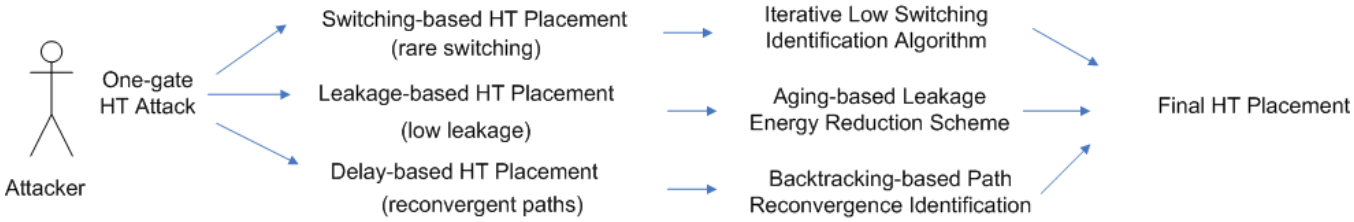

Figure 3: Overall flow for HT creation and placement.

(NBTI) follows fractional power law of the stress time, as shown in the following equation:

$$\Delta V_{th} = A \cdot exp(\beta V_G) \cdot exp(-E_\alpha / kT) \cdot t^{0.25}, \qquad (4)$$

where V_G is the applied gate voltage; A and β are constants; E_α is the measured activation energy of the NBTI process; T is the temperature; and t is the stress time.

4. ONE-GATE HARDWARE TROJAN BENCHMARK CREATION

4.1 Overall HT Creation Flow

Our idea in creating challenging HT test benchmarks is to embed HTs that induce minimum observable variations into the target design. In order to achieve this goal, we employ a one-gate HT trigger that switches the malicious circuitry on and off during the IC operation. In order to increase the difficulty level for detection, the one-gate HT trigger powers on the malicious circuitry only when a rare event occurs, which is defined and activated by the attacker. In this way, the only observable variation before the activation of malicious circuitry is the single HT gate embedded in the circuit. Therefore, to further complicate the detection attempts, the attacker would hide the single HT gate in the circuit and make it difficult to be detected by the commonly used detection methods.

We consider three possible HT creation models that an attacker may consider to minimize the possibility of detection. The proposed HT models correspond to the three most commonly used side-channels for HT detection, namely leakage power, switching power, and delay. Figure 3 shows the overall flow of creating the three types of one-gate HTs: (1) For the switching-based HT placement, we design an iterative low switching identification algorithm that searches for the most rarely switching locations in the target design; (2) In the leakage-based HT model, we develop an aging-based leakage power reduction scheme to minimize the observable variations in leakage power; and (3) For the consideration of timing-based HT, we employ a backtracking-based algorithm to identify the reconvergent paths in the circuit, where delay variations caused by the HT is not observable. Then, we combine the three sets of locations and find a number of specific locations that minimize the observability of all three properties. Finally, we create challenging HT benchmarks by embedding the one-gate HT trigger at one of these locations.

4.2 Low Leakage HT Benchmark

The leakage power-based HT model corresponds to the HT detection techniques that leverage whole circuit or gate-level leakage power tracing. In this case, the idea for hiding the one-gate HT is to minimize its leakage power consumption. Therefore, the embedded HT gate would cause a limited variation in leakage power and has a high probability of hiding under the measurement errors in the existing leakage power-based detection approaches.

Our implementation of such a low leakage HT gate is based on the observation that the gate-level leakage power decreases exponentially with the increase in the threshold voltage (following Equation (1)), and that the threshold voltage can be increased by IC aging process (following Equation (4)). Therefore, our idea is to intentionally age the embedded HT gate in the post-silicon stage to reduce its leakage power to the greatest extent.

We develop a satisfiability (SAT)-based approach to determine the input vectors that can stress the transistors and age the HT gate. Since the output signal of each gate can be expressed as a boolean expression of the input vectors, SAT can determine the input vectors that generate a specific signal pattern. SAT is one of the first known NP-complete problems. Several very high quality SAT solvers are readily available for delivering fast and accurate SAT solutions [7]. We leverage the SAT solutions to find the aging input vectors that stress the HT gate at the expectant location in the design.

One of the consequences of the aging-based low power HT creation is that it may cause a delay degradation, due to the aging of the one-gate HT as well as a set of other gates in the circuit by applying the selected input vectors. The increased delay may be observable by a timing-based HT detection approach. To address this issue, we compensate for the delay degradation due to aging by employing adaptive body bias (ABB). ABB has been proposed as an effective approach to compensate for the PV impact on performance and power consumption. It provides the ability to manipulate transistor threshold voltage through the body effect and thus enables either a forward or a reverse body effect to change threshold voltage [6]. Here we use ABB to manipulate the threshold voltage of critical gates (e.g., gates that are on the critical path), so that the variation in the circuit delay can be compensated for.

4.3 Rare Switching HT Benchmark

In the switching power-based HT model, the goal is to insert the HT in such a way that it can be switched only by a rare set of input vectors. Consequently, there is a limited probability for the one-gate HT to exhibit any switching activity during the normal IC operation; on the other hand, the attacker can apply the rare input vectors to activate the malicious circuitry at any time. Figure 4 shows our simulation results regarding the switching activities of all gates on ISCAS benchmark C499. Our observation in this

example is that all gates can be switched by a certain set of input vectors. Also, there exist gates that switch very often (e.g., more than 50% of the time) and, similarly, there are a small set of gates that have relatively low (but non-zero) switching activities (e.g., less than 5% of the time).

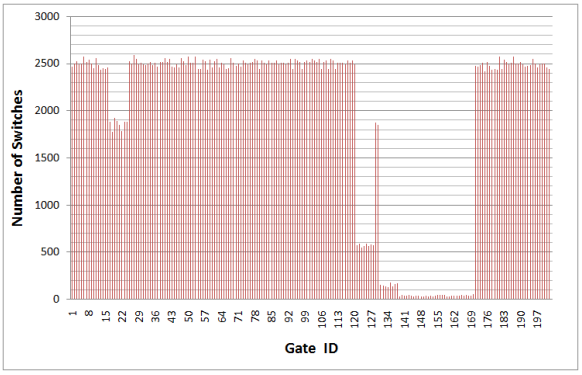

Figure 4: Switching activities of all gates on ISCAS benchmark C499, under the application of 5000 pairs of input vectors.

We develop a low switching identification algorithm that iteratively searches the locations on the original design and finds out those locations that lead to the most rare switching activities. Then, we connect the obtained input signals to a single HT gate that is expected to have low switching activity. Pseudocode 1 describes the detailed algorithm for finding such a gate set on the target circuit and placing the HT gate.[1] We start with the random simulation results, as shown in Figure 4, and find the most rarely switching gate in the design. Next, we iteratively add one more gate from the design to the candidate group. This gate is the most correlated with the existing gates in the group and has the least switching activity. The algorithm terminates after K iterations and provides us with K locations in the circuit that can drive a rarely switching HT trigger.

4.4 Timing-based HT Benchmark

The delay-based HT model utilizes the limitation in delay measurements that only the delay of one single path is measurable from a specific input to a specific output. Furthermore, in the cases where there are multiple parallel paths between an input/output pair, it is difficult to map the delay measurement to one of the paths that are in parallel. Therefore, the HT gate can be well hidden within one of the parallel paths without being discovered by the existing delay-based characterizations.

Based on this thought, we develop a backtracking-based search algorithm to find out all the possible parallel paths in the target circuit for HT insertion. In particular, we analyze the structure of the netlist and identify the reconvergence points between each pair of input and output. Here we define reconvergence points as the node in the netlist that is the end point of more than one paths. In the case of reconvergence, none of the paths are measurable in terms of delay, because it is not clear which path is being measured even though one can measure the end to end delay from a specific input to the reconvergence point. As long as a path is not measurable, it can serve as a difficult case for delay-based HT detection method. The reconvergence identification algorithm converts the netlist of the design to a direct graph. Then, the problem of reconvergence identification converts to a graph theory problem that searches for all the nodes that have an in-degree of at least 2.

[1] $switching(\cdot)$ is the function to find out the switching activity of gate g_i via simulation of random input vectors; and $SAT(\cdot)$ is the procedure to determine whether the specific gates are switchable via SAT problem solving. The details of the SAT approach is introduced in Section 4.5.

Pseudocode 1 Iterative searching algorithm for placing rare switching HT.

Input: Netlist Net;
Onput: A set of locations L that result in rare switching one-gate HT;
1: Find the most rarely switching gate g_0 via simulation of random input vectors;
2: Insert g_0 into L;
3: **for** $i \leftarrow 1; i < K; i++$ **do**
4: **for all** Gates t that are controlled by the transitive fan-in of L **do**
5: **if** $switching(g_i) < switching(L)$ && $SAT(L + g_i)$ is solvable **then**
6: $g_i \leftarrow t$;
7: **end if**
8: **end for**
9: Insert g_i into L;
10: **end for**

4.5 Summary of HT Benchmarks

The three HT models provide us with a systematic way of evaluating an arbitrary HT placement strategy in terms of the difficulty levels for detection. For example, if a single HT gate is embedded at one of the reconvergent paths, where the leakage power consumption is lower than the measurement resolution and the switching probability is small, it would create an ultra challenging case for the HT detection techniques.

Following this idea, we define the first systematic benchmarking strategy for creating and quantifying the HT attacks with various difficulty levels. The difficulty level of a HT attack model can be evaluated using a triplet $< d, l, s >$, where d is a boolean variable indicating whether the HT gate is observable via delay measurement (i.e., whether it is on one of the reconvergent paths); l is a boolean variable representing whether the leakage power of the HT gate is below the resolution of the leakage power characterization, and s is the switching probability of the inserted HT gate at the specific location. We can test and evaluate a HT detection approach using the proposed benchmark, by observing the most difficult level of HT that it can successfully detect.

5. EXPERIMENTAL RESULTS

We evaluate our HT benchmark creation method on a set of ISCAS'85, ISCAS'89, and ITC'99 benchmarks. For each benchmark circuit, we first embed a single HT at the location determined by our approach. Then, we evaluate the leakage power, switching power of the HT gate, as well as its observability under delay measurements. The combination of the three metrics quantifies the difficulty level of detecting such a HT attack.

5.1 Low Leakage-based HT

Table 1 and Table 2 shows the trend of total leakage energy reductions by varying the V_{th} increase during the aging process from 10% to 100%. We observe that the leakage energy can be reduced by up to 28X, which enables the placement of the ultra-low leakage HTs on all circuit locations. Furthermore, we observe that after the delay compensation of the non-HT gates is done using adaptive body biasing, the leakage energy reduction can still be up to 18X. The results indicate that we are able to place the low leakage HT gate without impacting the delay characteristics of the design, which makes the HT difficult to detect using both delay and leakage power-based characterizations. Furthermore, for the larger designs such as C7552 (shown in Table 2), we obtain a larger rate of leakage energy reduction.

Table 1: Leakage energy reduction via aging for HT benchmark creation (Benchmark C6288).

V_{th} Increase	Without Delay Compensation	With Delay Compensation
10%	2.0	1.9
20%	3.7	3.4
30%	6.3	5.6
40%	9.7	8.2
50%	13.4	10.9
60%	17.0	13.2
70%	20.2	15.0
80%	23.1	16.4
90%	25.8	17.5
100%	28.3	18.5

Table 2: Leakage energy reduction via aging for HT benchmark creation (Benchmark C7552).

V_{th} Increase	Without Delay Compensation	With Delay Compensation
10%	2.2	2.0
20%	4.3	4.0
30%	8.7	7.7
40%	16.9	14.4
50%	31.3	25.9
60%	55.2	44.0
70%	91.0	69.9
80%	138.9	102.6
90%	195.9	138.7
100%	256.8	173.7

5.2 Rare Switching-based HT

Figure 5 shows our simulation results for rare switching-based HT creation. The boxplots show the statistical distributions of the switching activities for all gates in the ISCAS'85 benchmark circuits, obtained from the simulation of 10,000 randomly generated input vectors for each design. For each box in the plot, the lower and upper edges correspond to the 25th and 75th percentiles of the distribution. The line in the middle of each box indicates the median of the distribution. The smallest and largest points are also shown if they happen outside a range from the box. In most cases the switching probability ranges from 20% to 50%, and it is very rare to have gates that can never be switched.

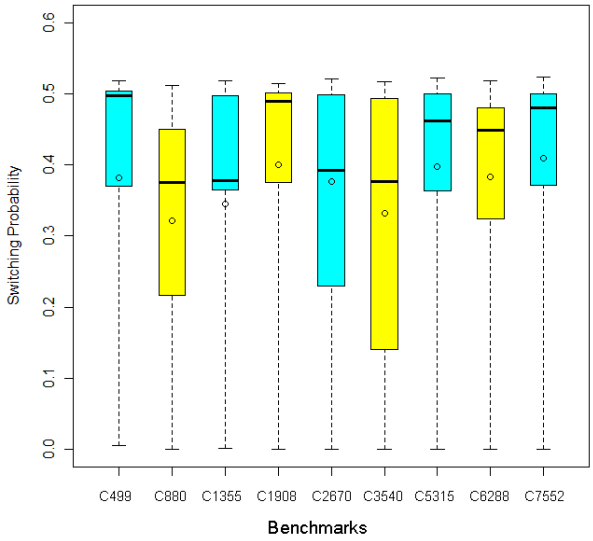

Figure 5: Simulation results of switching activities of all gates on ISCAS'85 benchmarks; 10,000 random input vectors were applied to each design.

However, after applying our iterative low-switching identification algorithm and feeding the obtained input pins to a single AND gate, we obtain a maximum of 0.78% switching probability while simulating 10,000 input vectors. Also, we have used SAT to show that for each AND gate, there is at least one input vector that could activate the malicious circuitry. Therefore, our results indicate that the attacker can use the rare activation condition to trigger the malicious circuitry during the system operation, while the single HT gate with low switching probability is difficult to detect when the malicious circuitry is dormant.

5.3 Timing-based HT

Table 3 summarizes our simulation results regarding delay characterizable gates on a set of ISCAS'85, ISCAS'89, and ITC'99 benchmarks. As discussed in Section 4.4, we cannot characterize the delay of a path if there exist parallel reconvergent path from the input to the output. From the simulation results, we observe that there is no full coverage of all gates in any of the evaluated benchmarks in terms of delay characterization. The highest achieved rate of coverage on the benchmark set is 60%, which still leaves a large portion of the circuit susceptible to HT placement without the risk of being detected.

6. CONCLUSION

We have developed three hardware Trojan attack models for creating HT detection benchmarks. The attack models are based on the consideration of hiding the one-gate HT trigger at low leakage, rare switching, and reconvergent locations on the circuit to bypass the security check by the commonly used side-channel based approaches. We showed that the proposed one-gate HT models can successfully compromise the detection attempts and provide a quantitative metric for evaluating HT detection mechanisms. Simulation

Table 3: Simulation results regarding uncharacterizable gates due to reconvergences. The high percentage of uncharacterizable gates in each design indicates that there is a large number of candidate locations for embedding the non-detectable one-gate HT trigger.

Benchmark	Gates	# Inputs	# Outputs	# Gates Subject to Reconvergence	% Gates Subject to Reconvergence
C499	202	41	32	80	39.6%
C880	383	60	26	208	53.5%
C1355	546	41	32	546	100%
C1908	880	33	25	739	84.0%
C2670	1193	233	140	931	78.0%
C3540	1669	50	22	1542	92.4%
C5315	2307	178	123	1924	83.4%
C7552	3512	207	108	2552	72.7%
S38584	19253	5	304	237	84.3%
B17	32192	37	97	19283	59.9%

results on a set of ISCAS'85, ISCAS'89, and ITC'99 benchmarks verified the effectiveness of the HT benchmarks. The resulting HT benchmark circuits and tools can be found at http://www.cs.ucla.edu/∼shengwei/htbench.html.

7. ACKNOWLEDGEMENTS

This work was supported in part by the NSF under Award CNS-0958369, Award CNS-1059435, and Award CCF-0926127 and in part by the DARPA/MTO Grant N66001-11-1-4103.

8. REFERENCES

[1] D. Agrawal et al. Trojan detection using IC fingerprinting. In *IEEE Symposium on Security and Privacy (SP)*, pages 296–310, 2007.

[2] Y. Alkabani and F. Koushanfar. Consistency-based characterization for IC trojan detection. In *ICCAD*, pages 123–127, 2009.

[3] M. Banga and M.S. Hsiao. VITAMIN: Voltage inversion technique to ascertain malicious insertions in ICs. In *HOST*, pages 104–107, 2009.

[4] S. Borkar et al. Parameter variations and impact on circuits and microarchitecture. In *Design Automation Conference (DAC)*, pages 338–342, 2003.

[5] S. Chakravarthi et al. A comprehensive framework for predictive modeling of negative bias temperature instability. In *International Reliability Physics Symposium (IRPS)*, pages 273–282, 2004.

[6] T. Chen and S. Naffziger. Comparison of adaptive body bias (ABB) and adaptive supply voltage (ASV) for improving delay and leakage under the presence of process variation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,*, 11(5):888–899, 2003.

[7] N. Een and N. Sorensson. An extensible SAT-solver. In *International Conferences on Theory and Applications of Satisfiability Testing (SAT)*, pages 333–336, 2004.

[8] Y. Jin and Y. Makris. Hardware Trojans in wireless cryptographic ICs. *IEEE Design Test of Computers*, 27(1):26–35, 2010.

[9] L. Kim, J.D. Villasenor, and C.K. Koc. A Trojan-resistant system-on-chip bus architecture. In *IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2009.

[10] F. Koushanfar and A. Mirhoseini. A unified framework for multimodal submodular integrated circuits trojan detection. *IEEE Transactions on Information Forensics and Security*, 6(1):162–174, 2011.

[11] F. Koushanfar, A. Mirhoseini, and Y. Alkabani. A unified submodular framework for multimodal IC trojan detection. In *Information Hiding*, pages 17–32, 2010.

[12] F. Koushanfar and M. Potkonjak. CAD-based security, cryptography, and digital rights management. In *Design Automation Conference (DAC)*, pages 268–269, 2007.

[13] D. Markovic et al. Ultralow-power design in near-threshold region. *Proceedings of the IEEE*, 98(2):237–252, 2010.

[14] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware trojan horse detection using gate-level characterization. In *Design Automation Conference (DAC)*, pages 688–693, 2009.

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

[16] M. Tehranipoor et al. Trustworthy hardware: Trojan detection and design-for-trust challenges. *IEEE Computer Magazine*, 44(7):66–74, 2011.

[17] S. Wei, S. Meguerdichian, and M. Potkonjak. Gate-level characterization: Foundations and hardware security applications. In *Design Automation Conference (DAC)*, pages 222–227, 2010.

[18] S. Wei, S. Meguerdichian, and M. Potkonjak. Malicious circuitry detection using thermal conditioning. *IEEE Transactions on Information Forensics and Security*, 6(3):1136–1145, 2011.

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

[20] S. Wei and M. Potkonjak. Integrated circuit security techniques using variable supply voltage. In *Design Automation Conference (DAC)*, pages 248–253, 2011.

[21] S. Wei and M. Potkonjak. Scalable consistency-based hardware Trojan detection and diagnosis. In *International Conference on Network and System Security (NSS)*, pages 176–183, 2011.

[22] S. Wei and M. Potkonjak. Scalable hardware Trojan diagnosis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2011.