Hardware Trojan Horse Detection Using Gate-Level Characterization

Miodrag Potkonjak Ani Nahapetian Michael Nelson Tammara Massey
Computer Science Department
University of California, Los Angeles (UCLA)
Los Angeles, CA 90095
{miodrag, ani, tmassey}@cs.ucla.edu

ABSTRACT

Hardware Trojan horses (HTHs) are the malicious altering of hardware specification or implementation in such a way that its functionality is altered under a set of conditions defined by the attacker. There are numerous HTHs sources including untrusted foundries, synthesis tools and libraries, testing and verification tools, and configuration scripts. HTH attacks can greatly comprise security and privacy of hardware users either directly or through interaction with pertinent systems and application software or with data. However, while there has been a huge research and development effort for detecting software Trojan horses, surprisingly, HTHs are rarely addressed. HTH detection is a particularly difficult task in modern and pending deep submicron technologies due to intrinsic manufacturing variability.

Our goal is to provide an impetus for HTH research by creating a generic and easily applicable set of techniques and tools for HTH detection. We start by introducing a technique for recovery of characteristics of gates in terms of leakage current, switching power, and delay, which utilizes linear programming to solve a system of equations created using non-destructive measurements of power or delays. This technique is combined with constraint manipulation techniques to detect embedded HTHs. The effectiveness of the approach is demonstrated on a number of standard benchmarks.

Categories and Subject Descriptors
B.7.m [**Hardware**]: Integrated Circuits – *Miscellaneous.*

General Terms
Experimentation, Security

Keywords
Hardware Trojan horses, gate-level characterization, linear programming, manufacturing variability.

1. INTRODUCTION

Since semiconductor manufacturing demands a large capital investment, the role of contract foundries has dramatically grown, increasing exposure to theft of masks, attacks by insertion of malicious circuitry, and unauthorized excess fabrication [1]. The development of hardware security techniques is exceptionally difficult due to reasons that include limited controllability and observability (50,000+ gates for each I/O pin in modern designs) [7], large size and complexity (the newest Intel processor has 2.06B transistors), variety of components (e.g., clock, clock distribution interconnect, and finite state machine), unavoidable design bugs, possibility of attacks by non-physically connected circuitry (e.g., using crosstalk and substrate noise), many potential attack sources (e.g. hardware IP providers, CAD tools, and foundries), potentially sophisticated and well-funded attackers (foundries and foreign governments), and manufacturing variability that makes each IC coming from the same design unique [5][11].

There are several broad types of malicious hardware attacks that we consider. The first is gate resizing, where the attacker intentionally changes the sizing factors of one or more gates in such a way that the circuit passes all standard timing test, but its timing for a certain inputs is incorrect or its switching or leakage power are globally or locally increased drastically. Note that many other gate sizing attacks can be envisioned, including one where the sizes of the gates are altered in such a way that the calculation of internal signals is facilitated through altered timing or switching power. In the second type of attack, the adversary adds one or more gates so that the functionality of the design is altered. It is important to observe that the gates can be added so that no timing path between primary inputs and flip-flops (FFs) and primary outputs and FFs is altered. However, leakage power is always altered because even if the attacker gates the added circuitry, the gating requires an additional gate. Our HTH detection approach is generic in a sense that it can easily be retargeted to other circuit components, such as interconnect by considering more comprehensive timing and/or power models.

The main technical obstacle to HTH detection is manufacturing variability, which may have a very significant impact on timing and power characteristics. Its difficulty is often compounded by low controllability and observability. Our approach is non-destructive and aims to minimize measurement test time. Currently, we target an off-line scenario, but the approach can be applied in the case of timing on-line. Even power-based techniques can be applied on-line by measuring the change of timing characteristics due to rise in temperature. However, for the sake of focus, we restrict our attention to the off-line case.

The basis for our approach is gate-level characterization using a set of timing and/or power measurements. The measurements are treated as a set of linear equations with imposed measurements errors. They are processed using linear programming (LP). When we address detection of additional ghost circuitry using LP, we impose additional constraints on our LP formulation in such a way that the results indicate what circuitry is added and where. Essentially, we use LP to ask if the characterization of gates is significantly more consistent under the assumption of added circuitry.

Finally, it is important to observe that gate-level timing characterization has other numerous applications beyond security. For example, it addresses in an elegant way static timing analysis and greatly facilitates dynamic timing analysis. It can be also used for tailoring post-silicon optimization. For example leakage power vectors can be calculated much more precisely when the gate sizing information is available. Also, it can be used for inexpensive silicon process characterization.

2. RELATED WORK

Most available hardware security, as well as hardware-based secure system methods, are based upon implementations of digital cryptography protocols [11]. In traditional cryptographic protocols, security is provided by trapdoor mathematical functions and digital keys, which make the protocols resilient to algorithmic attacks [17]. However, digital hardware security keys can be attacked in a number of ways including side-channel, electromigration, imaging and fault injection [5][6].

To address the above shortcomings and vulnerabilities, a new generation of security techniques based on manufacturing variability (MV) has been developed [14] [15][21]. With scaling of feature sizes, the physical limits of the devices are reached and uncertainty in the device size increases [7]. Variations in transistor feature sizes and thus, in gate characteristics are inevitable (e.g., delay or power). In present and pending technologies, the variation is large compared to the device dimensions. As a result, VLSI circuits exhibit a high variability in both delay and power consumption.

In contrast, the proposed new scheme does not require storing of secret information, does not require any exchange of secret information, and is much faster and less power expensive. Most available hardware security, as well as hardware-based secure system methods are based upon implementations of digital cryptography protocols [6]. In traditional cryptographic protocols, security is provided by trapdoor mathematical functions and digital keys, which make the protocols resilient to algorithmic attacks [10]. However, digital hardware security keys can be attacked in a number of ways including side-channel, electromigration, imaging and fault injection [3][4].

To address the above shortcomings and vulnerabilities, a new generation of security techniques based on manufacturing variability (MV) has been developed [8][9][12][15][16]. Note that the manufacturing variations are because of the intense industrial CMOS feature scaling. With scaling of feature sizes, the physical limits of the devices are reached and uncertainty in the device size increases [5]. Variations in transistor feature sizes and thus, in gate characteristics are inevitable (e.g., delay or power). In present and pending technologies, the variation is large compared to the device dimensions. As a result, VLSI circuits exhibit a high variability in both delay and power consumption.

3. TROJAN HORSE (HTH) ATTACK

3.1 Motivation

A Hardware Trojan Horse (HTH) is an intentional hardware alteration of the design specification or of the corresponding implementation. These alterations only affect the circuit's functionality in a few specific circumstances and are hidden otherwise. HTHs are more difficult to detect, diagnose, and mask than design bugs or manufacturing faults since they are intentionally implanted to be unperceivable by the current debugging and testing methodologies and tools. The vast number of possibilities for implementing HTHs further complicates their detection.

Diagnosis of HTHs can be especially intricate since a large number of Trojans may be present simultaneously in an IC. In addition, HTHs are not necessarily present in all ICs coming from a design. Since HTHs are embedded within the circuit and are active only under certain, very rare conditions, detection methods must be complex. In addition, the introduction of new technologies, ultra large scale integration, and intrinsic MV of deep submicron technologies also increase the difficulty of HTH uncovering. Implementation of HTHs by the attackers and its detection/prevention by the designers is a battle that is only bounded by the creativity, knowledge, and skills of each side. Although the IC layout data is shared with the foundry, the designers and the tool developers have the advantage of making the first and last move. Thus, they can construct their design to be the least suitable to HTH placement, and they can also choose the type of post-silicon HTH detection procedure.

3.2 Example of HTH Attacks

Here we present specific HTH, in an attempt to describe the nature of HTH attacks in general. A simple, yet powerful HTH attack is presented in Figure 1, which shows how ghost circuitry can be activated in a cell phone when specific inputs or data are detected at specific memory locations. The unshaded portion of the circuit represents the HTH circuitry when it is activated by a HTH caller ID number. Upon activation, the attacker bitstream (ABS) is activated and the initial cell phone design is corrupted. In this example, HTHs will either cause the cell phones to malfunction or cause confidential information to be leaked. Important information can be disclosed after activation of the HTH. The exploited phone can automatically dial a hidden spy third party when certain numbers are dialed. Ghost circuitry (HTH) may be difficult to identify by traditional timing/power analysis techniques. To avoid timing analysis-based detection, an attacker only needs to ensure that no path delays between the inputs to flip-flops (FFs) or between the outputs to FFs are increased. Also, the switching power can remain stable until the trigger of the attacker's caller ID activates the HTH.

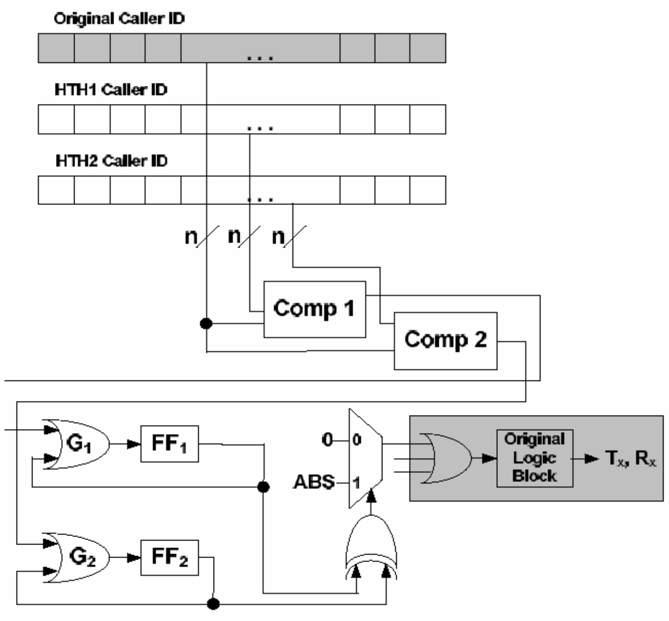

Figure 1. Example of a cell phone HTH

The crosstalk caused by HTH in Figure 2 is particularly difficult to detect since no physical connections between the ghost and the affected circuitry is necessary. Our SPICE simulations of wire crosstalk demonstrate that for two wires closer than 0.1 micrometers, the affected wire's delay can increase by more than four times, as shown in Table 1. A prevention method, during the place and route phase, is to carefully place the nonfunctional interconnects that fill the empty spaces to facilitate crosstalk.

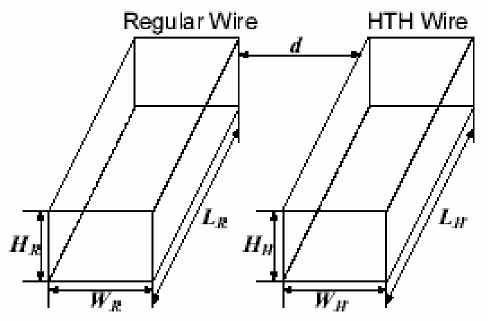

Figure 2. Example of a wire crosstalk HTH

Table 1. Interconnect capacitance vs. distance of the HTH wire

Distance	0.1	0.2	0.5	1.0
Increase (%)	420	105	34	3.2

Finally, consider an example from photonic crystals-based on-chip optical interconnects, one the future, potentially pervasive IC technologies. Due to its bandwidth, low power consumption, high speed, feature size compatibility, and easy transduction to and from electric signals, photonic crystals-based on-chip optical interconnects are one of the leading candidates for implementation of global on-chip interconnects [13]. However, if an attacker places a single transducer on the optical fiber, data being communicated on the high bandwidth channel can be interfered/obtained with no timing or electromagnetic impact.

3.3 HTH Classification

In the modern design flows of ASICs, ASIPs, and microprocessors, most of the phases may be potentially untrusted [1]. CAD tools, hardware intellectual properties (IPs), design libraries, testing and verification tools, untrusted foundries, untrusted testing facilities, and even untrusted designers in large design teams are all possible malicious threats. With field programmable gate arrays (FPGAs), the situation is even worse because the software and configuration files can also be attacked.

Even though the number and types of potential hardware attacks are essentially unlimited, we currently classify HTH attacks into the seven following categories: (i) damage objectives; (ii) components and mechanisms of the attack; (iii) components of the IC under attack; (iv) duration and initiation mechanisms; (v) design phase implantation and usage phase; (vi) optimization level; and (vii) customization level.

The HTH attackers desiderata may include alteration of the computed results, slowing the IC, increasing the power consumption, releasing confidential data, and facilitating side-channel attacks by making the gates' power consumptions observable at output. Attackers can employ various techniques including excessive switching, interconnect resizing, and substrate noise addition. Components, such as gates, clocks, and memory are vulnerable to physical attacks. Attacks may be randomly or actively initiated, e.g., event triggered. The impact may be temporary/permanent or recoverable/unrecoverable. The effectiveness and the detection difficulty of attacks can be quantitatively optimized. The attack may be on a specific weak IC component or on a predetermined set of components. For example, incorrectly sized gates are very slow on the design's critical path. Sizing variations caused by MV in ICs may be exploited for this attack. Unfortunately, the broad spectrum of possible attacks prevents the development of universally effective defense and protection measures. To mitigate this problem, our research work develops generic hardware security techniques that can easily be retargeted to new tasks.

We focus on detecting the differences between the taped-out chip's characteristics and post-silicon tests. We also address the more difficult HTH discovery problems where the malware is implanted within the hardware intellectual property (IP), cell and module libraries, CAD tools, and FPGA configuration software.

3.4 Global Flow

To carry out HTH detection, we first carry out gate-level characterization, by using non-destructive leakage power and timing delay measurements to create linear programs. Once, the scaling factor of all or most of the gates in the IC have been approximated, then we are able to carry out a suite of statistical techniques for determining the presence of HTHs.

4. GATE-LEVEL CHARACTERIZATION

Gate-level characterization (GLC) aims to recover the post-silicon and unique properties of each IC in the presence of manufacturing variability. GLC calculates the relevant characteristics of each gate of the design using a limited number of nondestructive measurements. We currently target delay, switching power, and leakage measurements and use GLC on combinatorial gates. However, other options such temperature and response to radiation may be also used for this approach. Even though we currently consider only combinatorial gates, one can generalize our techniques and easily apply our approach to sequential elements and interconnects. Due to space limitations and for the sake of clarity, we discuss the simple structure of GLC applied to leakage power, in this section. The key observations are that each gate consumes an energy that is proportional to its manufacturing variability scaling factor, *si*. Each gate *gi* depends on its input. Table 2 demonstrates this for a two input NAND gate for 90 nanometer technologies.

Table 2. Leakage current for different inputs

Input Vector	00	01	10	11
NAND-2 Leakage	0.776	10.39	4.137	15.15

GLC leakage characterization can be formulated as a set of linear equations. We assume that the design structure is known. Given an IC with *n* gates and *m* input pins, *K* different input vectors can be applied ($K << 2m$). The total leakage current is measured for each input vector. The goal is to find the scaling (sizing) factor for each gate's leakage. The stated problem is an over-constrained system of linear equations. Each equation has the form:

$$\sum_{j=1}^{n} s_j I_{k,j}^{nom} = \hat{I}_k^{total}$$

where $I_{k,j}^{nom}$ is the nominal leakage power for gate *j* for the input vector *k*, $\hat{I}_k^{total}$ is the total measured leakage current for the *k*-th input that is the sum of the correct value and a measurement error.

Depending on the choice of optimization objective function, the above problem can be stated in a linear, convex, or nonlinear program format. First, each equation is transformed into a constraint where the difference of the left and the right side of equation is less than $|\varepsilon_k|$ for the *k*-th equation. The objective is to minimize an appropriate error norm defined over the *K* error terms, ε_k's, e.g., the l_1 error norm is $\sum_{k=1}^{k} |\varepsilon_k|$, that can be linearly stated by a set of auxiliary variables ρ_k as

$$\sum_{k=1}^{k} \partial_k \text{ , s.t., } \partial_k \geq \varepsilon_k \text{ and } \partial_k \geq -\varepsilon_k \text{ .}$$

The stated system of equations can be optimally solved using linear programming. The relative advantages of linear, convex, and nonlinear programs are that they provide a fine trade-off between accuracy, run time, and modeling flexibility. While linear programming (LP) is fast, it assumes piece-wise linear error models. On the other hand, while non-linear programming is slow and has potential convergence problems, it can fit nonlinear error norms and, more importantly, handle non-linear scaling and leakage power models. There are two major potential difficulties with solving the system of equations with linear, convex, or non-linear programs. The first is that two or more scaling variables can be correlated in all equations and thus one cannot find their actual values. The second is that the large circuits cannot be solved in a reasonable amount of time.

To resolve the above concerns and also to address several others, we created a more elaborate GLC flow. Our current GLC flow has five phases that are embedded within a loop that terminates once the user defined accuracy criteria is satisfied or the runtime limit is reached. The five phases are:
(i) measurement organization;
(ii) equation analysis and selection;
(iii) solving system of equations;
(iv) results post-processing;
and (v) results validation.

First, we select the input vectors. In the case of delay measurements, we also decide on relative input arrival times. Second, we analyze the equations (constraints) that maximize the corresponding matrix rank to solve for the maximal number of scaling variables in a numerically stable manner. Third, we derived several heuristics for the task, which we use. Fourth, after the equations are solved for different input vectors, we combine their solutions to minimize the probability of large errors. Finally, learn-and-test and resubstitution statistical validation techniques are used to estimate bounds for the calculated scaling factors.

4.1 Gate-Level Characterization using Linear Programming (LP)

In evaluating the LP formulation for gate-level characterization, we studied three properties of input variables: (i) distribution type, (ii) the mean of the absolute measurement errors, and (iii) the number of constraints.

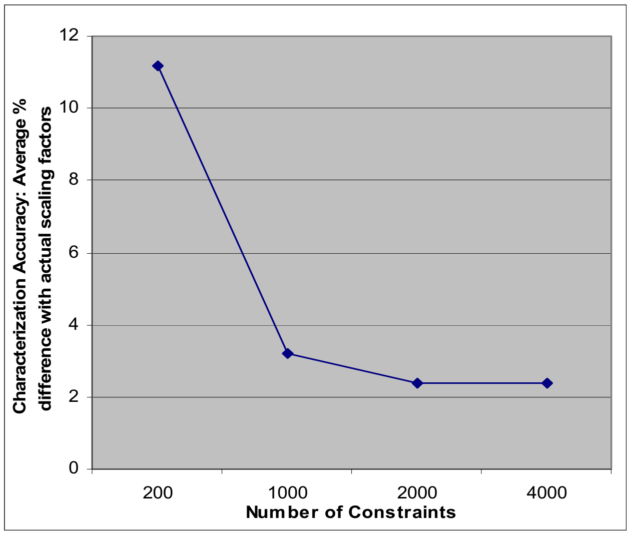

Figure 3. Accuracy of the gate-level characterization vs. the relative measurement error (%) for different number of constraints (equations) on c432 benchmarks

In Figure 3, we evaluate the accuracy of the gate characterization procedure versus the uniform measurement error and the number of constraints (NC) the c432 benchmark. The number of constraints refers to the number of equations that we used in the

LP, corresponding to the number of different inputs used to measure the leakage. We see that even though increasing the NC on average improves the accuracy, there exists some cases where with NC=2000 there is more accuracy than with NC=4000, because of the random variations in the subset of used equations.

Table 3 presents the results for LP-based GLC using leakage power, delay, and both leakage power and delay for various ISCAS85 benchmarks. The average error in GLC is presented for various percentages of average measurement errors. Overall in this paper, our results demonstrate that we are able to characterize gates with an error smaller than the measurements error.

Table 3. LP-based GLC for leakage power, delay, or both, for five ISCAS85 different benchmarks

Accuracy of Solvable Scaling Factor Values				
Measure Err	**1%**	**2%**	**5%**	**10%**
c432 Power	0.48	0.92	2.02	4.57
c432 Delay	0.33	0.71	1.59	4.04
c432 Both	0.14	0.71	0.98	2.63
c499 Power	0.72	1.57	3.93	7.82
c499 Delay	0.32	0.77	1.59	1.90
c499 Both	0.06	0.24	1.00	1.22
c880 Power	0.92	1.80	3.98	8.02
c880 Delay	0.30	0.80	1.09	1.23
c880 Both	0.08	0.69	0.92	1.20
c1355 Power	0.88	1.71	3.47	6.62
c1355 Delay	0.12	0.71	0.80	1.09
c1355 Both	0.14	0.66	0.78	0.92
c1908 Power	0.94	1.74	4.02	8.83
c1908 Delay	0.19	0.32	0.79	0.85
c1908 Both	0.19	0.22	0.52	0.58

5. HTH DETECTION

The goal of HTH detection is to identify the presence of HTHs and to estimate the probability of added ghost circuitry. HTH detection problem can be stated in many different frameworks that result in sharply different difficulties and false positive/false negative likelihoods. A key observation is that certain HTHs are impossible to detect without test points or analysis methods beyond GLC. For example, if an added gate has the same inputs as a gate of the same type in the original design, solely GLC techniques will not be able to identify it. Another important observation is that the difficulty of detection depends on several factors, such as the number and characteristics of the added gates, the interconnects, the measurement errors, the original design size, the design structure, and the number of conducted measurements.

We conduct HTH detection using three novel techniques: (i) statistical analysis; (ii) constraint manipulations; and (iii) comparison with technological and physical laws. In the first technique, we analyze the variable residuals and errors in individual equations. For example, a systematic positive measurement error is a strong indicator that a ghost circuitry is added. In the second technique, we manipulate constraints and the objective function in a nonlinear program. The solver in the nonlinear program indicates if a ghost circuitry is present. For instance, we may add the same extra variable to the right side of each constraint. Now, if the gates can be characterized in a more accurate and consistent manner, it is an indicator that a malware is present. This example assumes that only one gate is added. Finally, we also compare the GLC results to the relative characteristics of the gates with respect to the well-established physical design and technological laws. For example, if two gates are next to each other in the original layout, but the GLC determines that the gates have highly different scaling factors, we can deduce that they are placed further than the layout specification and it is likely that a ghost circuitry is placed in between them.

Once an HTH is detected, the natural next phase is to identify its location. We conduct HTH diagnosis using a two phase process. In the first phase, we identify the type of gates added and their position. After we determine which interconnects and gates receive signals and where these gates are physically located, our integer linear and nonlinear programming techniques can make the equations maximally consistent. Consistency will be achieved by adding gates that are not in the initial specification and by allowing the program to assign inputs to these gates. In the second phase, we correlate the assigned inputs with the existing input and intermediate signal locations to identify the most likely location of the added gate's signal. The high local correlation of sizing factors identifies the most likely HTH location using the maximum likelihood principle. We conducted a study of our HTH diagnosis technique by adding a single inverter to various ISCAS85 benchmarks. The inverter receives its input from a randomly selected gate in the benchmark and has no output to any of the original gates. Even if we set the measurement error to 5%, we were able to correctly detect/ diagnose the presence/ location of the inverter with higher than 98% accuracy.

Table 4. Detection accuracy using a single extra variable

Bench-marks	**% of Measurement Error**			
	1	**2**	**5**	**10**
c17	100	100	100	100
c432	100	100	100	98
c499	100	100	100	96
c880	100	100	100	95
c1355	100	100	98	95
c1908	100	100	98	94

Table 4 presents the results where an extra variable is added to the LP constraints per measurement.

Another approach we use is to examine the correlations between an input value(s) and the overall switching or leakage energy consumption and/or delay. This technique aims to leverage the observation that if a ghost gate is set to almost always use the same inputs, that input will have a significantly higher consumption than other sets of inputs. For all types of gates, this difference is a least two times higher. By fixing one or more inputs to a specific value and searching all combinations, it is very likely that there will be a subset of the inputs that impact the HTH with a higher probability.

The main technical difficulty is to account for the unknown scaling factors of the gates. We address this in the following two ways. First, we calculate the scaling factors of the gates using out linear programming maximum likelihood procedure for gate characterization. Then, we can compare the expected energy for various input vectors and find that there is a consistent bias due to the HTH. The second technique iterates the first technique by perturbing the scaling factors in order to address uncertainty about the actual gate scaling factors.

6. CONCLUSION

We have developed a system of techniques for two major classes of hardware Trojan horse detection and diagnosis: gate resizing and embedding of ghost circuitry. The techniques are non-destructive and apply a system of delay or power measurements followed by singular value decomposition and linear programming (LP) processing. The final step is either statistical data analysis or the more effective addition of LP constraints. The effectiveness of the approaches is demonstrated on a number of benchmarks.

7. REFERENCES

[1] Defense Science Board (DSB) study on high performance microchip supply, http://www.acq.osd.mil/dsb/reports/2005-02-hpms report final.pdf, 2006.

[2] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan detection using ic fingerprinting. In *IEEE Symposium on Security and Privacy (SP)*, pages 296–310, 2007.

[3] R. Anderson, M. Bond, J. Clulow, and S. Skorobogato. Cryptographic processors-a survey. *Proceedingsof the IEEE*, 94(2):357–369, 2006.

[4] R.J. Anderson. *Security Engineering: A guide to building dependable distributed systems*. John Wiley and Sons, 2001.

[5] K. Bernstein, D.J. Frank, A.E. Gattiker, W. Haensch, B.L. Ji, S.R. Nassif, E.J. Nowak, D.J. Pearson, and N.J. Rohrer. High-performance CMOS variability in the 65-nm regime and beyond. *IBM Journal of Research and Development*, 50(4/5):433–450, 2006.

[6] D. Hwang, P. Schaumont, K. Tiri, and I. Verbauwhede. Securing embedded systems. *IEEE Security & Privacy*, 4(2):40–49, 2006.

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

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

[9] K. Lofstrom, W.R. Daasch, and D. Taylor. IC identification circuits using device mismatch. In *International Solid State Circuits Conference (ISSCC)*, pages 372–373, 2000.

[10] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook Of Applied Cryptography*. CRC Press, 1997.

[11] A. Srivastava, D. Sylvester, and D. Blaauw. *Statistical Analysis and Optimization for* VLSI*: Timing and Power*. Series on Integrated Circuits and Systems. Springer, 2005.

[12] Y. Su, J. Holleman, and B. Otis. A 1.6J/bit stable chip ID generating circuit using process variations. In *International Solid State Circuits Conference (ISSCC)*, page to appear, 2007.

[13] G. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *Design Automation Conference (DAC)*, pages 9–14, 2007.

[14] E. Yablonovitch. Can nano-photonic silicon circuits become an intra-chip interconnect technology? In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 309–309, 2007.

[15] M. Nelson, A. Nahapetian, F. Koushanfar, M. Potkonjak. SVD-based Ghost Circuitry Detection. In *Information Hiding (IH),* 2009.

[16] R. Rad, X. Wang, J. Plusquellic, and M. Tehranipoor. Taxonomy of Trojans and Methods of Detection for IC Trust. In *International Conference on Computer-Aided Design (ICCAD)*, 2008.