Statistical Screening for IC Trojan Detection

Youngjune Gwon, H. T. Kung, Dario Vlah
Harvard University

Keng-Yen Huang, Yi-Min Tsai
National Taiwan University

***Abstract*—We present statistical screening of test vectors for detecting a *Trojan*, malicious circuitry hidden inside an integrated circuit (IC). When applied a test vector, a Trojan-embedded chip draws extra leakage current that is unfortunately too small for the detector in most cases and concealed by process variation related to chip fabrication. To remedy the problem, we formulate a statistical approach that can screen and select test vectors in detecting Trojans. We validate our approach analytically and with gate-level simulations and show that our screening method leads to a substantial reduction in false positives and false negatives when detecting IC Trojans of various sizes.**

I. Introduction

IC Trojan detection is a challenging task. A natural strategy for detecting extra circuitry is to examine the measured, static power consumption under the circuit's input combinations called test vectors. If a sufficient number of test vectors reveal the power measurement above what is expected from a Trojan-*free* circuit, then the circuit is declared Trojan-*embedded*. A class of techniques using such detection strategy is known as *side-channel analysis* [1].

While it may appear conceptually simple, the implementation of side-channel power analysis is challenging for several reasons. First of all, IC Trojans are presumably small circuits. Thus their power consumption is also small and can be completely hidden in normal variance. Secondly, process variations are introduced at chip fabrication time. Leakage current—the term we will use equivalently with (leakage) power, or power consumption, throughout the paper—for the same gate drawn under the same test vector varies depending on where the gate is on the chip, the chip in a die, the die on a wafer, and the wafer in a lot, as well as environmental factors such as variations in temperature and power sources. Lastly, test vectors drive a different set of gates in a chip to higher power consuming states. This means power consumption of the chip will be test vector dependent, and so will the expected power consumption of a Trojan-free chip and the threshold power consumption for the detection be.

For these reasons, IC Trojan detection is considered a hard problem that needs to cope with various statistical uncertainties. A Trojan-embedded circuit, for example, could sometimes consume less power—not more—than the original circuit without the Trojan under certain test vectors and chips. It is therefore necessary to address the process and test vector-induced variations, which call for statistical robustness in detection methods.

In this paper we focus on statistical screening of test vectors that reduces false positives and false negatives in an IC Trojan detection scheme based on *leakage current* side-channel analysis. We describe our detection methodology and explain how false positive and negative rates are calculated in Section II. Section III presents our method for screening test vectors and explains the underlying idea. In Section IV, we empirically validate our approach and discuss the key results. We describe related work in Section V, and Section VI concludes the paper.

II. Multi-chip Trojan Detection Procedure

Suppose that we are given a number of *test chips* embodying the same circuit-under-test (CUT). Our Trojan detection procedure consists of three tasks: (1) conduct single-chip test for all test chips and declare findings for each chip; (2) analyze the probabilities of false positive and false negative declarations in the preceding task; and (3) draw a statistical conclusion suggesting whether the CUT is Trojan-free or Trojan-embedded.

Task 1. Conduct single-chip test on multiple chips. For each test chip, we apply all test vectors[1] to the CUT on the chip as input vectors of the circuit. For each test vector, we measure empirical power consumption (pw) and record its ideal (or gold) power consumption (g) defined as the sum of mean power consumptions of all gates at their respective states driven by the test vector.

We make the following declarations about the Trojan presence on the chip:

- Positive if $pw > g + u$ for at least $L\%$ of test vectors
- Negative if $pw < g + v$ for at least $L\%$ of test vectors
- Inconclusive otherwise

where u and v are certain thresholds with $0 < v \leq u$ and $L > 50$. We choose u and v around the expected power consumption of Trojan.

Task 2. Analyze probabilities of false positive and false negative declarations. A positive declaration resulted from $pw > g+u$ in Task 1 for required $L\%$ of test vectors is false if the CUT turns out to be Trojan-free. This is known as a *false positive* declaration. Note that under the same test vector, the leakage power consumed by a Trojan-free CUT may vary from chip to chip due to process variation. Fig. 1(a) illustrates the probability density function (PDF) of this power consumption. Fig. 1(b) depicts the corresponding PDF of a Trojan-embedded CUT under the same test vector. The second PDF is essentially the first PDF shifted to right by the expected leakage power drawn by the Trojan under the test vector. We note the use of Gaussian-like PDFs for convenience. They actually provide a reasonably accurate approximation in real-world scenarios when the number of gates turned on by the test vector is

[1] Applying all possible test vectors, however, is not feasible for some circuits whose input space is too large, *e.g.*, CPUs. For such cases, we assume that a reasonably sized (and properly selected) subset of test vectors is available.

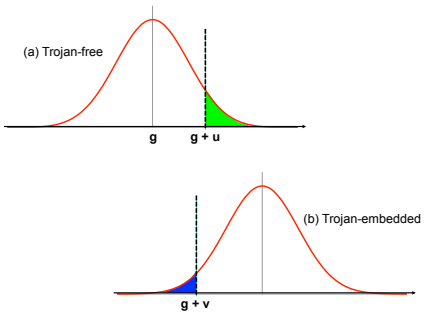

Fig. 1. Probability density function for (a) Trojan-free and (b) Trojan-embedded CUT under a test vector

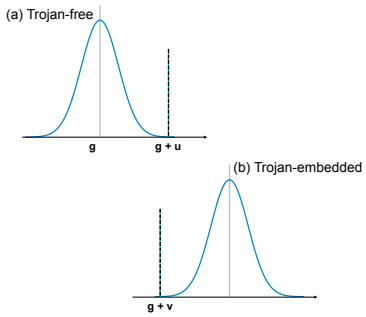

Fig. 2. Probability density function for (a) Trojan-free and (b) Trojan-embedded CUTs under a screened test vector

large. The probability of false positive declaration is the area highlighted in green along the right tail of the PDF in Fig. 1(a). Similarly, the probability of false negative declaration is the area colored in blue from Fig. 1(b).

Task 3. Infer statistical conclusion on whether or not Trojan is present based on test results from multiple chips. After completing all single-chip tests on Q test chips, we can determine the likelihood of Trojan presence based on the outcome. Following a classical hypothesis testing framework, we formulate our null hypothesis H_0 as "the CUT is Trojan-free." Suppose ρ out of Q chips have been declared positive. Then assuming H_0 is true, we can compute the probability of this outcome as follows:

$$\begin{aligned} &\textit{Probability of } \rho \textit{ chips declared positive} \\ &\quad = Pr\{\text{Exactly } \rho \text{ chips with } pw > g + u\} \\ &\quad = \binom{Q}{\rho} p^{\rho}(1-p)^{Q-\rho} \end{aligned}$$

where p is the probability of a false positive declaration defined in Task 2. (p can be empirically determined from statistics on the findings of Task 1.) For example, consider $Q = 10$ and $p = 0.1$ with observed $\rho = 5$. The above expression yields a probability of 0.0015, which is assumed to be smaller than the threshold we set for hypothesis testing. Thus we reject H_0 and conclude that the CUT is more likely Trojan-embedded (with a *false positive rate* of 0.0015). We can similarly derive the false negative rate based on the number of chips that receive a negative declaration.

III. Statistical Screening of Test Vectors to Reduce False Detection

To lower process variation induced false positive and negative rates, we use test vectors screening to select those which will lead to reduced tail size of the PDFs in Fig. 1. Intuitively speaking, a decrease of the tail size on the right side of the PDF will reduce false positive declarations. That is, the PDF under each test vector survived from screening will exhibit a smaller variance. On the other hand, a decrease of the tail size on the left side will reduce false negative declarations. A successful screening will transform the PDFs of Fig. 1 to look like the PDFs in Fig. 2. Observing these principles, we specify our method for statistical screening of test vectors for a given test chip:

(a) Apply all test vectors and measure their leakage currents;
(b) Organize test vectors into W groups of N randomly shuffled test vectors;
(c) For each group, sort test vectors into b bins according to the measured leakage current;
(d) For each bin, fit the leakage current distribution of test vectors into a Gaussian PDF, obtain parameters μ_{fitted} and σ_{fitted}, and filter out those test vectors whose measured leakage current is less than $\mu_{fitted} - \alpha \cdot \sigma_{fitted}$ and greater than $\mu_{fitted} + \beta \cdot \sigma_{fitted}$.

The bins in Step (c) are centered around b equally spaced leakage current values—note this will in general result in different number of test vectors per each bin. Step (c) also assures that test vectors in the same bin will have similar g, the gold leakage current response. We will carry out Step (d) only for those bins that have sufficiently many test vectors (e.g., > 100) to allow meaningful statistical screening. After the screening of Step (d), for remaining test vectors the power consumption will likely exhibit smaller gaps between the idealistic g and the empirical leakage current (a pw equivalent). For example, we have noticed from our test circuits of Section IV.B that the screening procedure reduces the maximum gap across all test vectors up to 65 nA in leakage current. Subsequently, power consumption of the CUT under these test vectors will have smaller variances, thus more effective to bring out the subtle contribution from small Trojan circuitry. Furthermore, by filtering out the test vectors with relatively low power consumption in Step (d), we assure that the remainders will turn on sufficient gates to allow more stable statistics.

It is important to note that the screening of Step (d) follows *asymmetric* filtering—that is, we cut the left tail using smaller filtering parameter α than β for the right tail (*e.g.*, $\alpha = 1$ and $\beta = 2$). We make the usual assumption that the leakage current (or power) distribution of logic gates is best modeled using the log-normal distribution [2]. Thus, the leakage current distribution of the test chip should follow the sum of log-normal random variables. The left side of a log-normal PDF has a steep, short tail while its right side has a heavy tail, which explains our asymmetric filtering.

TABLE I
LOG-NORMAL PARAMETERS FOR A 2-INPUT NAND GATE

Input (state)	μ (nA)	σ (nA)
00	.223	.082
01 or 10	4.578	3.026
11	13.109	16.785

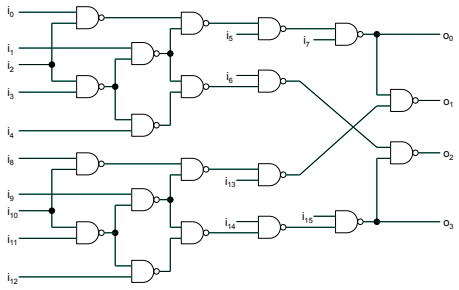

Fig. 3. `double-c17` combines two ISCAS-85 `c17`s.

IV. PERFORMANCE EVALUATION

Our empirical validation features a benchmark circuit of 100 NAND gates with which we simulated the chip leakage current under all possible combinations of test vector inputs. We wrote custom C code that propagates precise, internal logic states of all gates in the benchmark circuit and followed a log-normal leakage current model for each gate. This section explains our experimental methodology and discusses empirical results.

A. Simulating Log-normal Leakage Current Distribution

The physical variations in a circuit, which are normally distributed, have an exponential effect on current. This is why the log-normal distribution serves a good approximation for leakage current. To measure the total power consumption of a circuit, we must first know an input combination to each gate (*i.e.*, gate state), which is driven by a test vector applied at the circuit input. Our simulator propagates logic states of all gates and determines the total leakage current by summing individual gate's leakage contribution. Table I [3] summarizes the log-normal parameters to estimate leakage current for a 2-input NAND gate fabricated under an arbitrary process.

B. Benchmark Circuit

We adopted circuit `c17` from the ISCAS-85 benchmark suite [4] as a building block. The original `c17` consists of 6 NAND gates. We combined two `c17` blocks to create `double-c17`, which contains 20 NAND gates as depicted in Fig. 3. Lastly, we use five `double-c17` blocks to produce our benchmark circuit, namely `double-c17x5` as shown in Fig. 4. The `double-c17x5` benchmark has 16 input pins, which yields a test vector space of size 2^{16}.

C. Generating Clean and Trojan-embedded Test Chips

We generated 10 unmodified `double-c17x5` chips representing Trojan-free CUTs. They are generated independently and randomly using the log-normal parameters of Table I such that the same gates in the CUT (locality) among different chips induce different leakage currents.

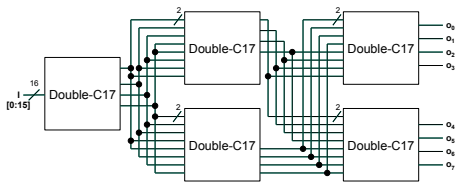

Fig. 4. 100-NAND gate `double-c17x5` benchmark circuit used for evaluation.

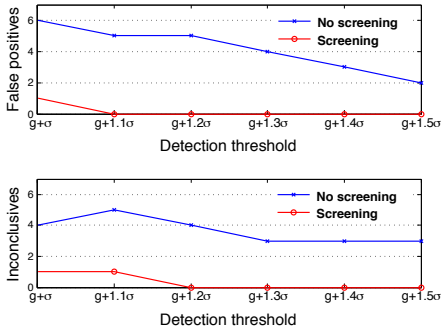

Fig. 5. Effects of test vectors screening in reducing the number of false positive or inconclusive declarations among 10 chips

We generated *five* different IC Trojans of sizes 1 to 5 NAND gates, namely `trojan-1` to `trojan-5` (*i.e.*, our smallest Trojan is a single NAND gate). Next, we inserted each Trojan (`trojan-`*i*) to a clean `double-c17x5` at its upper middle `double-c17` block. We instantiated 10 chips for each Trojan (totaling 50 altogether). Finally, we ran logic simulations for the clean and Trojan circuits to acquire the leakage current measurements.

D. Evaluation Scenarios

We have the following evaluation scenarios.

1) Scenario 1 – false positives: we run the Trojan detection procedure on the 10 Trojan-free chips to determine the number of chips that are declared positive (*i.e.*, these are false positive declarations), using unscreened and screened test vectors. We expect the number of false positives to be smaller with the screened test vectors.

2) Scenario 2 – false negatives: we run the Trojan detection procedure on the 50 Trojan-embedded chips and determine the number of chips that are declared negative (*i.e.*, these are false negative declarations). Note that `trojan-1` would be more difficult to detect. Again, we expect the number of false negative declarations to be smaller with our screening method.

E. Results and Discussion

Fig. 5 presents the number of false positive and inconclusive declarations. We applied the detection threshold $g + u$ with variable $u \in \{\sigma, 1.1\sigma, 1.2\sigma, 1.3\sigma, 1.4\sigma, 1.5\sigma\}$[2]. From the threshold set at $g + 1.2\sigma$, our screening method could achieve both zero false positive and zero inconclusive. In contrast,

[2]We determine σ using Fernandes and Vemuri's method [2] that estimates the sum of log-normals with the parameters from Table I.

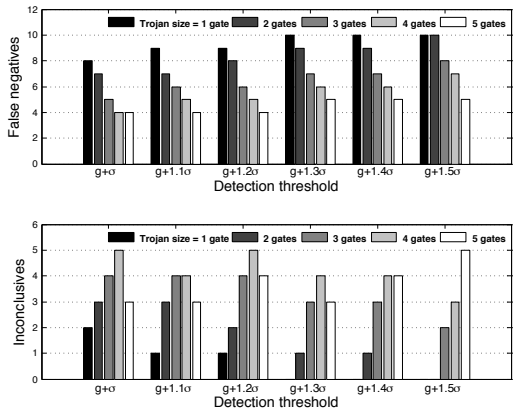

Fig. 6. False negative declarations among ten chips for unscreened test vectors

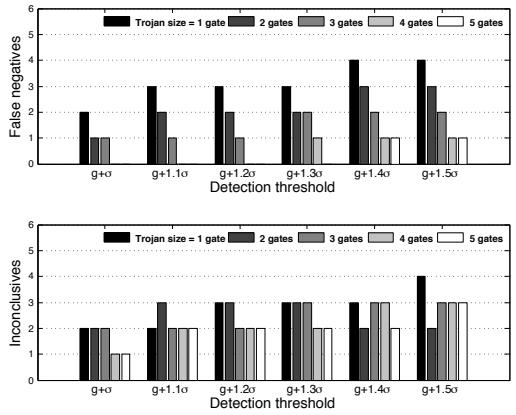

Fig. 7. Improved false negative declarations among ten chips with statistical screening

for the unscreened test vectors none of our test setting could achieve zero false positive declaration.

Figs. 6 and 7 depict false negative declarations for unscreened and screened test vectors. As with the scenario for false positive declarations, we used the same values for v ranging from σ to 1.5σ to set detection thresholds. While false negatives for smaller Trojan sizes of 1, 2, or 3 NAND gates (*i.e.*, only 1–3% the size of the benchmark circuit) could not be eliminated completely, our test vector screening significantly outperformed unscreened test vectors, resulting a 60–85% reduction in false negative detection.

An increase in detection threshold causes the number of false positive declarations to drop. This holds true for both unscreened and screened test vectors as the threshold moves along the right side of the Trojan-free PDF eventually reducing the area underneath the curve to zero thereby incurring no false positives. As illustrated in Fig. 2, screening removes false positives more effectively by reducing the tail size. A similar explanation can be made for the case of false negative along the left side of the Trojan-embedded PDF curve. We point out that the superior performance of the screened method is largely due to the use of asymmetric screening with $\alpha = 1$ and $\beta = 2$ as discussed in Section III.

The last step for Trojan detection is to set up the hypothesis testing described in Section II. From Scenario 1 with the single-chip threshold $g+1.2\sigma$, we found that $p < 0.01$ with our screening method and $p = 0.44$ for unscreened test vectors. Using a set of $Q = 10$ chips, the observed $\rho = 0$ for screened test vectors yielded 0.951, and $\rho = 6$ for unscreened test vectors yielded 0.26 false positive rates. Despite the large ρ value of 6, unscreened test vectors resulted an indecisive, mediocre probability 0.26, which is not high enough to make H_0 stand true with confidence. On the other hand, if we decide to reject H_0, then we would have an incorrect conclusion of the CUT being Trojan-embedded with a high 0.26 false positive rate. Either conclusion is unsatisfactory. In contrast, tests of Task 1 under our screening method would result $\rho = 0$ with high probability of 0.951. This makes H_0 stand true with high confidence and conclude the CUT Trojan-free, which is indeed correct for this set of chips.

V. Related Work

Side-channel analysis for IC Trojan detection has been discussed popularly at major security and circuit conferences. We stress that our work here is not about inventing a new side-channel analysis scheme, but about improving the performance of side-channel techniques by reducing statistical uncertainty surrounded test vector measurements. Thus, existing techniques such as *IC Fingerprinting* [1], *GLC* [5], and *DISTROY* [3] can embrace our method and benefit from it.

VI. Conclusions

Trojan detection is fundamentally a statistical procedure to remove uncertainties of process variations that make a hard problem for side-channel analysis. In response to this problem, we have introduced a novel statistical screening method for selecting the test vectors that help substantially reduce false positives and false negatives in detecting IC Trojans. In addition, we have proposed a multi-chip detection procedure that provides a general framework for statistical detection methods. With gate-level simulations, we have shown that the proposed method should work for practical circuits. Test vector screening can potentially benefit a variety of IC Trojan detection approaches such as compressive sensing based detection [3]. In future work, we plan to explore links to those approaches.

Acknowledgment

This material is based on research sponsored by Air Force Research Lab under agreement number FA8750-10-2-0115 and FA8750-10-2-0180. U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Lab or U.S. Government.

References

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection Using IC Fingerprinting," in *IEEE S&P*, 2007.

[2] R. Fernandes and R. Vemuri, "Accurate Estimation of Vector Dependent Leakage Power in the Rresence of Process Variations," in *ICCD*, 2009.

[3] Y. Gwon, H. T. Kung, and D. Vlah, "DISTROY: Detecting IC Trojans with Compressive Measurements," in *USENIX HotSec*, Aug. 2011.

[4] M. C. Hansen, H. Yalcin, and J. P. Hayes, "Unveiling the ISCAS-85 Benchmarks," *IEEE Design & Test*, vol. 16, pp. 72–80, July 1999.

[5] M. Nelson, A. Nahapetian, F. Koushanfar, and M. Potkonjak, *SVD-Based Ghost Circuitry Detection*. Springer-Verlag, 2009, pp. 221–234.