Statistical Screening for IC Trojan Detection

Youngjune Gwon, H. T. Kung, Dario Vlah, Harvard University
Keng-Yen Huang, Yi-Min Tsai, National Taiwan University

Presented at the 2012 IEEE International Symposium on
Circuits and Systems (ISCAS 2012) in Seoul, Korea
May 21, 2012
Introduction: IC Trojan

- Malicious circuitry inserted into chip
  - Not a bug, doesn’t happen accidentally!
  - Injected *on purpose* by an adversary

- Detect before activated, or face consequences
  - Malfunction
    - Perform incorrect operations and fail regular tasks
  - Breach of security
    - Leaks critical privacy information that needs be protected

- Detection methods
  - IC deprocessing
  - Functional testing
  - *Side-channel analysis* $\Rightarrow$ our baseline approach
Trojan Leakage Side-channel Analysis

- Key idea: measure and examine leakage current under sufficient number of test vectors before conclusion

Test vectors
(Circuit’s input combination bit patterns)

Circuit Under Test (CUT)

Measured leakage current values

Analysis

Conclusion

Reference (ideal) values

v_N \ldots v_2 v_1 

x_N \ldots x_2 x_1 

\mathbf{g}_1 \mathbf{g}_2 \ldots \mathbf{g}_N 

Done offline

Process database: contains fab process-specific gate leakage models

EDA/Tool

DB
Challenges

- Side-channel analysis is difficult in reality

- Trojans are extremely small circuits
  - Trojan leakage current could be indistinguishable even with idealistic assumption of under normal variance in leakage current

- Process variations
  - Same gates under same test vector can draw different leakage current
    - Due to difference in fab technologies, locality within chip/die/wafer/lot

- Test vectors drive circuit’s power consumption state
  - But largely unknown and unpredictable
  - Fortunately nominal reference values \( g \) available to us
Multi-chip Testing for Trojan Detection

- Intuition: instead of testing only one chip out of batch, test multiple chips $\Rightarrow$ improves confidence of statistical conclusion

Test each sample chip independently & draw combined statistical conclusion
Multi-chip Trojan Detection Procedure (1/3)

- Given: $Q$ test chips drawn from a batch representing circuit-under-test (CUT) fabricated under known process

Step 1. Perform single-chip test on each of $Q$ chips

- Apply all* test vectors and measure leakage currents ($i_{\text{Leak}}$) drawn
- Look up ideal leakage current value ($g$) for each test vector
  - Available from EDA simulation tool based on circuit design and process-specific modeling or from measuring reference/gold circuit
- Declare Trojan presence on tested chip:
  - Positive if $i_{\text{Leak}} > g + u$ for at least $L\%$ of test vectors
  - Negative if $i_{\text{Leak}} < g + v$ for at least $L\%$ of test vectors
  - Inconclusive otherwise

Note: $0 < v \leq u$, and for instance, $L > 50$

*: if possible; otherwise use some reasonable subset of test vectors
Multi-chip Trojan Detection Procedure (2/3)

Step 2. Analyze probabilities of false positive and false negative declarations

Trojan-free distribution under a test vector

Trojan-embedded PDF

\[ p = \Pr\{\text{false positive declarations}\} \]

\[ q = \Pr\{\text{false negative declarations}\} \]

Th (decision threshold)
Step 3. Infer statistical conclusion about Trojan presence with multi-chip test results

- Compute $\eta$, probability of $\rho$ out of $Q$ chips declared Trojan-embedded
  
  $$\eta = \binom{Q}{\rho} p^\rho (1 - p)^{Q-\rho}$$

  - Note: $p$ = probability of false positive declarations determined in previous steps

- Set up hypothesis testing
  - Null hypothesis $H_0$: CUT is Trojan-free if $\eta > Th$, otherwise reject $H_0$
    - Probability $\eta$ should be convincingly large to make $H_0$ stand

- Example: $Q = 10$, $\rho = 5$ declared positive with $p = 0.1$
  - Resulting probability $\eta = 0.0015$ is too small $\Rightarrow$ reject $H_0$ and take alternate hypothesis instead: CUT is more likely Trojan-embedded
Statistical Screening of Test Vectors

- Process variations induce false positives and negatives

- Key intuition: select test vectors that can reduce tail size of leakage probability distributions
  - Such screening will reduce both false positive and false negative rates

Can this be done?
Screening Method Proposed

1. Apply all test vectors $\mathbf{v} = [v_1 \ldots v_N]^T$ and examine measured leakage currents $\mathbf{i} = [i_{\text{Leak} 1} \ldots i_{\text{Leak} N}]^T$ with their reference values

2. Organize test vectors into $W$ random groups
   – Each group contains $M$ randomly shuffled test vectors

3. For each group, sort test vectors into $b$ bins by their measured leakage currents
   – Bins are centered around $b$ equally spaced leakage current values

4. For each bin, fit test vectors into Gaussian PDF by their measured leakage current values

5. Filter out test vectors with leakage current less than $\mu_{\text{fitted}} - \alpha \cdot \sigma_{\text{fitted}}$ and greater than $\mu_{\text{fitted}} + \beta \cdot \sigma_{\text{fitted}}$
Screening Method Illustrated

- All test vectors \( \mathbf{v} = \{v_i\}_{i=1}^{N} \)
- \( W \) random groups
- Each group contains \( M \) random test vectors

For each group, do bin packing by sorting test vectors with measured leakage currents.

- \( b \) equally spaced bins according to leakage current values

For each bin, fit leakage current values of test vectors into a Gaussian PDF \( \Rightarrow \) obtain \( \mu_{\text{fitted}} \) and \( \sigma_{\text{fitted}} \)

Filter out test vectors with leakage current less than \( \mu_{\text{fitted}} - \alpha \cdot \sigma_{\text{fitted}} \) and greater than \( \mu_{\text{fitted}} + \beta \cdot \sigma_{\text{fitted}} \)
Screening Method Explained

- **Why Gaussian fitting?**
  - Leakage current distribution best modeled as sum of *log-normals*
    - Each gate = log-normal random variable
    - See, *e.g.*, Rao *et al.* [*IEEE Trans. on VLSI Systems*, Feb. 2004]
  - Central Limit Theorem (CLT)
    - Sum of *any* (independent) random variables approaches to a Gaussian

- **Asymmetrical filtering**
  - $\alpha \neq \beta$
  - Typically $\alpha < \beta \implies$ cut more test vectors on left tail than right
    - We used $\alpha = 1$ and $\beta = 2$
  - Why?
    - Log-normal PDF has heavier tail on right than its short, steeper left tail
Performance Evaluation of Proposed Statistical Screening Procedure

- Benchmark circuit has 100 NAND gates
  - Built using ISCAS-85 c17 blocks
  - 16 input bits $\Rightarrow$ test vector space $\in 2^{16}$
- Logic simulator written in C
  - Pre-ran all possible test vectors and cached results
- Trojan circuits: 5 different configurations
  - Placed 1 to 5 NAND gates at one of c17 blocks to obtain trojan–1/2/3/4/5 (with increasing leakage current)
    - trojan–1 yields smallest leakage, thus most difficult to detect
  - Q = 10 for each Trojan configuration
    - Total 50 Trojan-embedded chips generated

- Experiments
  - Scenario 1 – false positives: using Q = 10 Trojan-free chips
  - Scenario 2 – false negatives: using Q = 10 Trojan-embedded chips for each of 5 different Trojan configurations
Scenario 1 Results – Reducing False Positives

Q=10 chips \implies \text{tested chip is declared Trojan-embedded if measured leakage current} > Th \text{ for } L = 70\% \text{ of test vectors}

Test vector screening results significant reduction in false positives!

We tried different \textit{Th} (threshold) values
Scenario 2 Results – Reducing False Negatives

Q=10 chips are embedded with Trojan circuit = trojan-1  
⇒ same detection rule ($L = 70\%$, tried different $Th$ values)

Test vector screening results significant reduction in false negatives too!

Using all test vectors  Using screened test vectors

<table>
<thead>
<tr>
<th>$Th$</th>
<th># of false negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g+\sigma$</td>
<td>8</td>
</tr>
<tr>
<td>$g+1.1\sigma$</td>
<td>10</td>
</tr>
<tr>
<td>$g+1.2\sigma$</td>
<td>9</td>
</tr>
<tr>
<td>$g+1.3\sigma$</td>
<td>8</td>
</tr>
<tr>
<td>$g+1.4\sigma$</td>
<td>2</td>
</tr>
</tbody>
</table>
Hypothesis Testing with Q=10 Trojan-free Chips

- Consider \( Th = g + 1.2\sigma \) and \( L = 70\% \) set to declare positive
  - Our screening method yielded \( p = \Pr\{\text{false positive declaration}\} < 0.01 \)
  - Without our screening method: \( p = 0.43 \)

- Results from multi-chip testing
  - Our screening method concluded \( \rho = 0 \) chips Trojan-embedded \( \implies \) corresponds to probability \( \eta = 0.951 \)
  - Without our screening method, \( \rho = 6 \implies \eta = 0.26 \)

- Our method resulted high probability value of 0.951 to make \( H_0 \) (CUT has no Trojan) stand with strong confidence
  - This conclusion is correct and good

- Without our screening method, low probability value of 0.26 would probably reject \( H_0 \) and take alternative hypothesis \( H_1 \) (CUT is Trojan-embedded)
  - This conclusion is incorrect
Summary

- Trojan side-channel analysis is hard in reality
  - Fab process variations worsen already difficult problem
  - Must incorporate statistical techniques to strengthen confidence of detection

- Our solutions
  - Multi-chip side-channel analysis testing
  - Statistical screening of test vectors
    - Aims to remove test vectors that could result leakage current outliers
    - Screened test vectors incur smaller gap from ideal leakage current value \(\Rightarrow\) coherent Trojan side-channel analysis is possible

- We substantially reduced false positives, false negatives, and inconclusives

- Additionally we quantify how confident our conclusion is
  - Hypothesis testing
Thank You!

- Any questions?

- You may direct additional questions to: gyj@eecs.harvard.edu
Supporting Slides
Why Leakage Current Side-channel?

- Total power of circuit, \( P_{\text{Total}} = P_{\text{Dynamic}} + P_{\text{Leakage}} \)

- Circuit’s dynamic power is a function of two parameters
  - Clock frequency \( f \) and switching activity \( N \)
  - Hold circuit’s input value constant to decharacterize \( P_{\text{Dynamic}} \)

- \( P_{\text{Leakage}} = I_{\text{Leakage}} \cdot V_{\text{DD}} \implies \text{depends only on leakage current} \)
  - Leakage current decided by gate characteristics: how many gates, fab process technology, locality within wafer/die/chip/lot

- Thus, leakage current is ideal candidate to fingerprint circuits
  - Every gate contributes
  - Each gate has unique \( I_{\text{Leakage}} \) property