Integrated Circuit (IC) Decamouflaging: Reverse Engineering Camouflaged ICs within Minutes

Mohamed El Massad†

with Siddharth Garg† and Mahesh Tripunitara‡

†New York University, ‡University of Waterloo
“Extracted an IC with embedded encryption hardware and 12K gates of digital logic....Now we understood the encryption, had the keys and full chip simulations running” — [Torrance+, CHES’09]

Boffins deduce chip's crypto just by looking at it

Smartcard hacking enters script-kiddie phase

4 Aug 2011 at 04:35, Dan Goodin

Black Hat Hackers have released tools that unlock the software stored on heavily fortified chips so
Proposed Solution: IC Camouflaging

- Use of dummy contacts to camouflage a gate. [US6791191]

- Identity of camouflaged gate cannot be determined by attacker. [R+, CCS’13]
  - Ex: \{XOR, NAND, NOR\} look identical to attacker

Defender vs. Attacker

Defender selects gates to camouflage

Camouflaged Circuit

Original Circuit

Black-Box Circuit

Applies inputs

Observes outputs
Camouflagging has a per-gate cost (area/delay/power).

Claim \([R+, CCS'13]\): if a small number of judiciously selected gates (> 140) are camouflaged \(\implies\) attacker would need “1000’s of years” to decamouflage.

\([R+, CCS'13]\) seemingly resolves cost vs. security trade-off.
Polynomial-time attack strategy if gates can be simultaneously justified and sensitized.
Claim \([R^+, \text{CCS'13}]\): If gates \emph{cannot} be simultaneously justified and sensitized, attacker must resort to \textbf{brute-force attack} \(\rightarrow\) exponential complexity in number of camouflaged gates.

Procedure to camouflage gates such that this property is satisfied.
Each input eliminates a subset of solutions (aka *completions*).

A set of inputs *sufficient* to eliminate all but the right completion $\rightarrow$ discriminating set.
$C$ is the camouflaged circuit.
$X$ is a completion, i.e., assignment to camouflaged gates.
$C_X$ is the camouflaged circuit with completion $X$.
$C$ is the blackbox circuit.

**Definition**

$I$, a set of input patterns, is **discriminating** if:

for every incorrect completion $X$, $\exists i \in I$ s.t. $C_X(i) \neq C(i)$
This Paper: In practice, both query cost and computational cost of attack are low → IC decamouflaging in minutes.

Credit: liv9.ca
Devising the Two Procedures

**DISC-SET-DEC**, Inputs: $C, I, C(I)$. Is $I$ NOT a discriminating set?

**Certificate for **DISC-SET-DEC**: Distinct completions $X_1$ and $X_2$ that agree on all inputs in $I$ but not on new input $i \notin I$.  
$\implies \in \text{NP}$

**COMPLETION-DEC**, Inputs: $C, I, C(I)$.  
$\exists$ a completion $X$ such that $C_X(I) = C(I)$?

**Certificate for **COMPLETION-DEC**: A valid completion $X$.  
$\implies \in \text{NP}$
Oracle for DISC-SET-Dec

\[\langle X_1, X_2, i \rangle\]

\[\langle C, I, \mathcal{C}(I) \rangle\]

\[\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}\]

Initial = \emptyset

Oracle for COMPLETION-Dec

\[\langle C, I, \mathcal{C}(I) \rangle\]

\[\langle X, \rangle\]
Building the Oracles

\[ \text{Disc-Set-Dec/Completion-Dec} \leq \text{(reduce)} \leq \text{CNF-SAT} + \text{SAT Solver (e.g., MiniSat)} \]

Credit: bigcommerce.com, aclib.net
### Benchmarks

<table>
<thead>
<tr>
<th>B'mark</th>
<th>Inputs</th>
<th>Outputs</th>
<th>Gates</th>
<th>Camouflaged</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>36</td>
<td>7</td>
<td>160</td>
<td>10</td>
</tr>
<tr>
<td>s298</td>
<td>3</td>
<td>6</td>
<td>133</td>
<td>6</td>
</tr>
<tr>
<td>s400</td>
<td>3</td>
<td>6</td>
<td>164</td>
<td>7</td>
</tr>
<tr>
<td>s444</td>
<td>3</td>
<td>6</td>
<td>181</td>
<td>7</td>
</tr>
<tr>
<td>s713</td>
<td>35</td>
<td>23</td>
<td>393</td>
<td>9</td>
</tr>
<tr>
<td>c5315</td>
<td>178</td>
<td>123</td>
<td>2406</td>
<td>63</td>
</tr>
<tr>
<td>c7552</td>
<td>207</td>
<td>108</td>
<td>3512</td>
<td>65</td>
</tr>
<tr>
<td>s5378</td>
<td>35</td>
<td>49</td>
<td>2779</td>
<td>56</td>
</tr>
<tr>
<td>s9234</td>
<td>19</td>
<td>22</td>
<td>5597</td>
<td>79</td>
</tr>
<tr>
<td>s38584</td>
<td>38</td>
<td>304</td>
<td>19234</td>
<td>128</td>
</tr>
</tbody>
</table>

Same number of gates camouflaged as in \([R+, CCS'13]\).
Time to Decamouflage

Brute-force $\rightarrow$ Our Attack: $10^{13}$ Years $\rightarrow$ 50 Minutes.
Why Does the Attack Work?

Discriminating sets (i.e., query costs) are small, in practice.

Camouflaging insecure even with \( > 5 \times \) increase in cost.
Can IC Camouflaging Work?

- Increase attacker’s query-complexity.
- Increase # possible gate-types.
Strong caution for IC designers.

Appealing claims on secure IC camouflaging with low cost need to be vetted carefully.

Mindset rooted in foundations is helpful.
Related Work

**Chipworks.**
Inside the Apple Lightning Cable.

**Degate.**
Reverse engineering integrated circuits with degate.
http://www.degate.org/documentation/

L.-W. Chow, J. P. Baukus, and W. M. Clark, Jr.
Integrated circuits protected against reverse engineering and method for fabricating the same using vias without metal terminations.

J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri.
Security Analysis of Integrated Circuit Camouflaging.

**SypherMedia.**
Syphermedia library circuit camouflage technology.
http://www.smi.tv/solutions.htm

A. Baumgarten, A. Tyagi and J. Zambreno.
Preventing IC piracy using reconfigurable logic barriers.
Why Does the Attack Work?
Increased # Camouflaged Gates (c5315)

Mohamed El Massad†

IC Decamouflaging