Using Logic-Based Reduction for Adversarial Component Recovery*

J. Todd McDonald, Eric D. Trias, Yong C. Kim, and Michael R. Grimaila

Center for Cyberspace Research
Air Force Institute of Technology
WPAFB, OH

*The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government
Outline

• Protection Context
• Polymorphic Variation as Protection
• Hiding Properties of Interest
• Framework and Experimental Results
Protection Context

• Embedded Systems / “Hardware”
  • Increasingly represented as reprogrammable logic (i.e., software!)
  • We used to like hardware because it offered “hard” solutions for protection (physical anti-tamper, etc.)

• Our beginning point: what happens if hardware-based protections fail?
  • Hardware protection: I try to keep you from physically getting the netlist/machine code
  • Software protection: I give you a netlist/machine code listing and ask you questions pertaining to some protection property of interest

• Protection/exploitation both exist in the eye of the beholder
Protection Context

Development America's Airmen Today ... for Tomorrow

- Critical military / commercial systems vulnerable to malicious reverse engineering attacks
  - Financial loss
  - National security risk
- Reverse Engineering and Digital Circuit Abstractions
  - Architectural (Behavioral)
  - Register Transfer Language (RTL)
  - Gate Level
  - Transistor Level
  - Layout

\[\text{INCREASING DETAIL}\]
Polymorphic Variation as Protection

- **Experimental Approach:**
  - Consider practical / real-world / theoretic circuit properties related to security
  - Use a variation process to create polymorphic circuit versions
    - *Polymorphic* = many forms of circuits with semantically equivalent or semantically recoverable functionality
  - Characterize algorithmic effects:
    - Empirically demonstrate properties
    - Prove as intractable
    - Prove as undecidable
Two Roads Met in the Woods…
and I Went Down Both…

Semantic Changing
Black-Box Refinement
Semantic Transformation
Polymorphic Generation

Program Encryption
Random Program Model

Semantic Preserving
Polymorphic Generation

Obfuscation

What can I measure?
What can I characterize?
What are the limits if I am only allowed to retain functionality?

What can I prove / not prove under RPM?
Defining Obfuscation

• Since we can’t hide all information leakage….

• Can we protect intent?
  • Tampering with code in order to get specific results
  • Manipulating input in order to get specific results
  • Correlating input/output with environmental context

• Can we impede identical exploits on functionally equivalent versions?

• Can we define and measure any useful definition of hiding short of absolute proof and not based solely on variant size?
Hierarchy of Obfuscating Transforms

Develop America's Airmen Today ... for Tomorrow

Logical View

Functional Hiding
Control Hiding
Component Hiding
Signal Hiding
Topology Hiding (Gate Replacement)

Physical Manifestation

Side Channel Properties

Air University: The Intellectual and Leadership Center of the Air Force
Integrity - Service - Excellence
Polymorphic Variation as Protection

Algorithm and Variant Characterization:

Selection:
1) Random
2) Deterministic
3) Mixture

Replacement
1) Random
2) Deterministic
3) Mixture
Framework and Experimental Results

- When does (random/deterministic) iterative selection and replacement:
  1) Manifest hiding properties of interest?
  2) Cause an adversarial reverse engineering task to become intractable or undecidable?

- What role does logic reduction and adversarial reversal play in the outcome (ongoing)

- Are there circuits which will fail despite the best variation we can produce? (yes)
Components

- Components are building block for virtually all real-world circuits
- Given:
  - circuit $C$
  - gate set $G$
  - input set $I$
  - integer $k > 1$, where $k$ is the number of components
- Set $M$ of components \{\(c_1, \ldots, c_k\)} partitions $G$ and $I$ into $k$ disjoint sets of inputs and/or gates.
- Four base cases
  - Based on input/output boundary of component and the parent circuit
Component Recovery

Develop America's Airmen Today ... for Tomorrow

Air University: The Intellectual and Leadership Center of the Air Force
Integrity - Service - Excellence
Independent Components and Induced Redundancy

ORIGINIAL

WHITE-BOX VARIANTS

REDUCED VARIANTS

Develop America's Airmen Today ... for Tomorrow

Air University: The Intellectual and Leadership Center of the Air Force

Integrity - Service - Excellence
Observing Independent Component Hiding

Develop America's Airmen Today ... for Tomorrow

Selection:
Random Algorithm
3 gates / 4 gates

Replacement:
Redundant / Standard

Replacement Gate Types:
20 / 50 / 100 / 500 / 1000

Iteration #:
C17-C17-C17
RandomAlgorithm
4 Gates Standard
1000 Iterations

“Variants”

Reduction Experiment

“Reduction Round”
- Algorithm 5: Reduce Constant 0/1 with Inverter Inputs
- Algorithm 7: Reduce Two Gates to Buffer/NOT/Constant 0/1
- Algorithm 4: Reduce Constant 0/1
- Algorithm 10: Reduce AND/OR/NAND/NOR with Inverter Inputs
- Algorithm 3: Reduce Inverter with Successor XOR/XNOR
- Algorithm 2: Reduce Inverter
- Algorithm 12: Reduce Diamond pattern
- Algorithm 1: Reduce Buffer
- Algorithm 11: Reduce V pattern
- Algorithm 6: Reduce Two Gates to AND/NAND/OR/NOR
- Algorithm 9: Reduce Gate with Opposite Inputs
- Algorithm 8: Reduce XOR/XNOR Gates to Buffer/NOT

“Reduced Variant”

Air University: The Intellectual and Leadership Center of the Air Force
Integrity - Service - Excellence
<table>
<thead>
<tr>
<th>Iterations</th>
<th>Standard(3)</th>
<th>Redundant(3)</th>
<th>Standard(4)*</th>
<th>Redundant(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>44.74 %</td>
<td>42.50</td>
<td>25.00</td>
<td>55.93</td>
</tr>
<tr>
<td>50</td>
<td>58.90</td>
<td>68.49</td>
<td>21.14</td>
<td>55.74</td>
</tr>
<tr>
<td>100</td>
<td>60.80</td>
<td>78.74</td>
<td>25.33</td>
<td>62.88</td>
</tr>
<tr>
<td>500</td>
<td>75.80</td>
<td>80.18</td>
<td>26.35</td>
<td>75.62</td>
</tr>
<tr>
<td>1000</td>
<td>77.41</td>
<td>84.22</td>
<td>31.56</td>
<td>73.46</td>
</tr>
</tbody>
</table>

Note: *Best

**Graph:**
- **Y-axis:** Avg. % of Reduction
- **X-axis:** 20 iteration, 50 iteration, 100 iteration, 500 iteration, 1000 iteration

- **Lines:**
  - Blue: Standard(3)
  - Red: Redundant(3)
  - Green: Standard(4)
  - Purple: Redundant(4)
Develop America's Airmen Today ... for Tomorrow

<table>
<thead>
<tr>
<th>Variant (Obfuscated)</th>
<th>Reduced (Avg)</th>
<th>Reduced (Best)</th>
<th>Reduced (Worst)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gates</td>
<td>1096</td>
<td>173 (84.22%)</td>
<td>158 (85.58%)</td>
</tr>
<tr>
<td>Levels</td>
<td>265</td>
<td>40 (84.91%)</td>
<td>35 (86.79%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obfuscated</th>
<th>Reduced (Avg)</th>
<th>Reduced (Best)</th>
<th>Reduced (Worst)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gates</td>
<td>2133</td>
<td>1483 (30.47%)</td>
<td>1474 (30.90%)</td>
</tr>
<tr>
<td>Levels</td>
<td>614</td>
<td>426 (30.62%)</td>
<td>425 (30.78%)</td>
</tr>
</tbody>
</table>
## Case Study

**Develop America's Airmen Today ... for Tomorrow**

<table>
<thead>
<tr>
<th>Variant Algorithm</th>
<th>c432-c499</th>
<th>c432-c880</th>
<th>ISCAS Merge</th>
<th>Buffer-100</th>
<th>Buffer-500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
<td>S</td>
<td>C</td>
<td>O</td>
<td>S</td>
</tr>
<tr>
<td>Pattern Based Reduction</td>
<td>-</td>
<td>85%</td>
<td>21-29%</td>
<td>-</td>
<td>63%</td>
</tr>
<tr>
<td>Size/Levels</td>
<td>-</td>
<td>89%</td>
<td>24-36%</td>
<td>-</td>
<td>72%</td>
</tr>
<tr>
<td>Independent Components (pattern-based reduction)</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Logic Cells (Quartus II)</td>
<td>133</td>
<td>155</td>
<td>165</td>
<td>173</td>
<td>184</td>
</tr>
<tr>
<td>Independent Components (as realized by Quartus II)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

- O - original circuit
- S - Simple
- C - Complex
- nn - not tested
- xx - too big based on I/O

Air University: The Intellectual and Leadership Center of the Air Force

Integrity - Service - Excellence
Conclusions

Develop America's Airmen Today ... for Tomorrow
Hiding Properties of Interest

General Intuition and Hardness of Obfuscation

The ONLY true “Virtual Black Box”

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>4</th>
<th>5</th>
<th>Y6</th>
<th>Y7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>AND(3,2)</td>
<td>OR(4,1)</td>
<td>XOR(4,3)</td>
<td>NAND(5,6)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

“The How”

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>Y6</th>
<th>Y7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>XOR(4,3)</td>
<td>NAND(5,6)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Semantic Behavior
• Is perfect or near topology recovery useful (therefore, is topology *hiding* useful)?
  • In some cases, yes
  • Foundation for other properties (signal / component hiding)
  • For certain attacks, it is all that is required

• Accomplishing topology hiding
  • Change basis type (normalizing distributions, removing all original)
  • Guarantee every gate is replaced at least once
  • Multiple / overlapping replacement = diffusion

**Topology:**
Gate fan-in
Gate fan-out
Gate type
### Experiment 1: Measuring “Replacement” Basis Change

**Waveform for Experiment 1:**

<table>
<thead>
<tr>
<th>Waveform</th>
<th>Number of Gates</th>
</tr>
</thead>
<tbody>
<tr>
<td>c432</td>
<td>120 (4 ANDs + 79 NANDs + 19 NORs + 18 XORs + 40 inverters)</td>
</tr>
<tr>
<td>Decomposed</td>
<td>230 (60 ANDs + 151 NANDs + 19 NORs + 40 inverters)</td>
</tr>
<tr>
<td>Decomposed NOR</td>
<td>843 (843 NORs)</td>
</tr>
</tbody>
</table>
Experiment 1a: Measuring “Replacement” Basis Change

$\Omega = \{\text{NOR}\} \rightarrow \Omega = \{\text{AND, NAND, OR, XOR, NXOR}\}$
Experiment 1b: Measuring “Replacement” Basis Change

\[ \Omega = \{\text{NAND}\} \rightarrow \Omega = \{\text{AND, NOR, OR, XOR, NXOR}\} \]
Experiment 2: Measuring “Replacement” Uniform Basis Distribution

ISCAS-85  c1355

Iterative Random Selection Algorithm:

Selection Strategy:
- 5% 1) Single Gate: Random
- 75% 2) Two Gate: Random
- 5% 3) Two Gate: Largest Level
- 5% 4) Two Gate: Output Level
- 5% 5) Two Gate: Random Level
- 5% 6) Two Gate: Fixed Level

Replacement Strategy:
- Random 6-GATE Basis

<table>
<thead>
<tr>
<th></th>
<th>C1355</th>
<th>Decomposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>506 gates ( 56 ANDs + 416 NANDs + 2 ORs + 32 buffers + 40 inverters )</td>
<td>550 gates ( 96 ANDs + 416 NANDs + 6 ORs + 32 buffers + 40 inverters )</td>
</tr>
<tr>
<td>Decomposed NAND</td>
<td>730 gates ( 730 NANDs )</td>
<td></td>
</tr>
</tbody>
</table>
Experiment 2: Measuring “Replacement” Uniform Basis Distribution

\[ \Omega = \{\text{NAND}\} \rightarrow \Omega = \{\text{AND, NAND, OR, NOR, XOR, NXOR}\} \]

“Single 4000 Iteration Experiment”
Experiment 2: Measuring “Replacement” Uniform Basis Distribution

\[ \Omega = \{\text{NAND}\} \rightarrow \Omega = \{\text{AND, NAND, OR, NOR, XOR, NXOR}\} \]

“Multiple 4000 Iteration Experiments”
Experiment 2: Measuring “Replacement” Uniform Basis Distribution

Develop America's Airmen Today ... for Tomorrow

$\Omega = \{\text{NAND}\} \rightarrow \Omega = \{\text{AND, NAND, OR, NOR, XOR, NXOR}\}$

“Multiple 4000 Iteration Experiments”
Experiment 3: Measuring “Replacement”
Smart Random Selection

ISCAS-85  c432

Iterative Smart Random 2-Gate Selection Algorithm:

Selection Strategy: **Smart** Two Gate Random
Replacement Strategy: Random Equivalent
Experiment 3: Measuring “Replacement” Smart Random Selection

\[ \Omega = \{ \text{NOR} \} \rightarrow \Omega = \{ \text{AND, NAND, OR, XOR, NXOR} \} \]
Things We’ve Learned Along the Way

Develop America's Airmen Today ... for Tomorrow

- What algorithmic factors influence hiding properties the most?
  - Iteration number
  - Selection size
  - Replacement circuit generation (redundant vs. non-redundant)

- Ongoing work in:
  - Increasing selection size
  - Determinist generation
  - Integrated logic reduction
  - Formal models: term rewriting systems, abstract interpretation, graph partitioning
Obfuscation Comparison Models

Develop America's Airmen Today ... for Tomorrow

VBB
\[ P_1 \rightarrow O \rightarrow O(P_1) \rightarrow ??? \rightarrow TT_{P_1} \]

RPM
\[ P_1 \rightarrow O \rightarrow O(P_1) \rightarrow ??? \rightarrow P_R \]

IND
\[ P_1 \rightarrow O \rightarrow O(P_1) \rightarrow P_2 \rightarrow O \rightarrow O(P_2) \]
\[ P_1 \rightarrow ??? \rightarrow O(P_1) \]
\[ P_2 \rightarrow ??? \rightarrow O(P_2) \]
\[ P_1, P_2 \in \delta_f \]

BP
\[ P_1 \rightarrow O \rightarrow O(P_1) \rightarrow O(P_1) \rightarrow O \rightarrow O(O(P_1)) \]
\[ P_1 \rightarrow ??? \rightarrow O(P_1) \rightarrow ??? \rightarrow O(O(P_1)) \]
Experiment 1a: Measuring % of ORIGINAL GATES

- 600
- 675
- 600
- 600
Experiment 1a: Measuring “Replacement”

\[ \Omega = \{\text{NOR}\} \rightarrow \Omega = \{\text{AND, NAND, OR, XOR, NXOR}\} \]

 ISCAS-85 c1355

# of NORs

~7500 # of Iterations

Integrity - Service - Excellence
Experiment 2: Measuring “Replacement”

\[ \Omega = \{ \text{NAND} \} \rightarrow \Omega = \{ \text{AND, NAND, OR, NOR, XOR, NXOR} \} \]
Experiment 2: Measuring “Replacement”

\[ \Omega = \{\text{NAND}\} \rightarrow \Omega = \{\text{AND, NAND, OR, NOR, XOR, NXOR}\} \]

“Multiple 4000 Iteration Experiments”