DETERMINISTIC VARIATION FOR ANTI-TAMPER APPLICATIONS

J. Todd McDonald, Yong C. Kim, Daniel Koranek

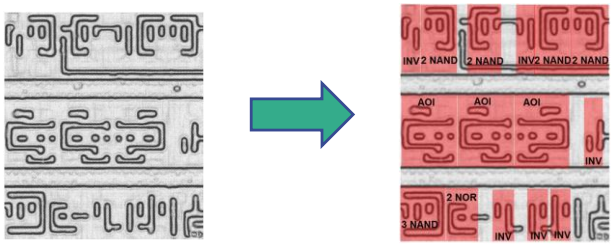
Dr. Jeffrey “Todd” McDonald, Ph.D.
Center for Forensics, Information Technology, and Security
School of Computer and Information Sciences
University of South Alabama

Motivation

- Computing technology in national infrastructure is a strategic resource
  - Malicious reverse engineering shortens technological advantage
  - Adversaries understanding our technology can manipulate, clone, subvert
- Protection Tools
  - Physical access
  - Encryption
  - Tamper-proofing
  - Watermarking / fingerprinting
  - Obfuscation
- We consider limits of obfuscation of *combinational* circuit logic

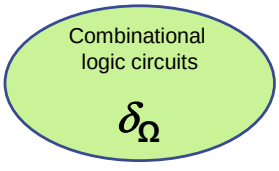
Example

- Reverse engineering of Mifare Classic RFID tag
  - Dutch government previously invested over $2 billion in new transit ticketing system
  - Nohl *et al.*[1] exposed transistors to identify gate level structures
  - From gate level structures components are identifiable
  - Revealed cryptographic keys enabling free access to Dutch transit system



Obfuscation

- O: $\delta_\Omega \rightarrow \delta_\Omega$
  - Efficient (running time/size): $O(C) = C'$
  - Semantic equivalence: $\forall x: C(x) = C'(x)$
  - Security property

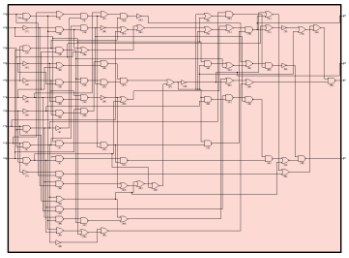


- Theoretically, ideal obfuscation not possible
  - No efficient algorithm exists to create a virtual black box
  - There are circuits which no algorithm can obfuscate
- Theoretically, ideal virus detection not possible
  - No efficient algorithm exists that can detect all future viruses
  - There are viruses that no algorithm can detect
- For security, we prefer something over nothing…
  - We still use AV products, despite their lack
  - We still investigate obfuscation to know what is possible practically

Protection Goals

- Reverse engineering [2,3] = design recovery at higher abstraction level, understanding abstract relationships

Given unstructured combinational logic $C \in \delta_{\Omega}$



??

Key Abstractions:

Topology
Signals
Components
Control functions

Discovery of known abstractions allows ID of other unidentified, unstructured patterns [4] :

Library modules / Repeated modules
Expected global structures
Computed functions
Control functions
Groupings of module outputs (bus structure)

Characterizing Security Properties

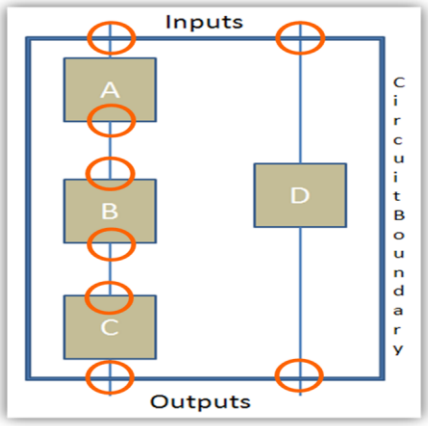
- Practical definition of security → reducing or eliminating amount of abstract information present
  - Circuits built from predefined components
  - Primary adversarial reverse engineering goal
- Security Property = Component Hiding:
  - Given original component configuration, remove or reduce information about component relationships to prevent recovery of original abstractions
- Issues
  - Measuring the abstract information present
  - Worst-case scenarios
  - Measurement only focuses on one attack vector

- Components are building block for virtually all real-world circuits

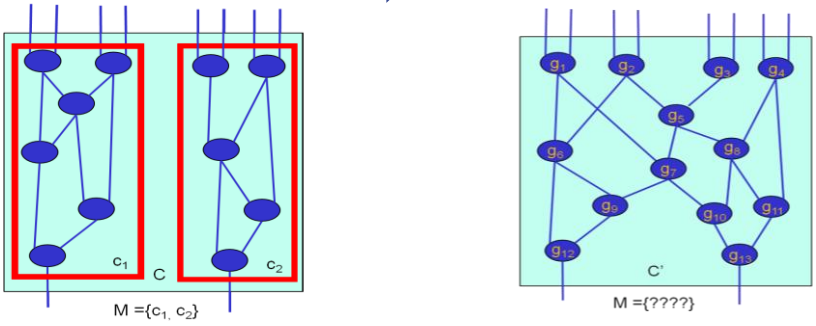
- Given:
  - circuit *C*
  - gate set *G*
  - input set *I*
  - integer $k > 1$, where k is the number of components

- 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



$C \in \delta_\Omega$ $\Rightarrow$ $C' = O(C),\ C' \in \delta_\Omega$



Two step process:

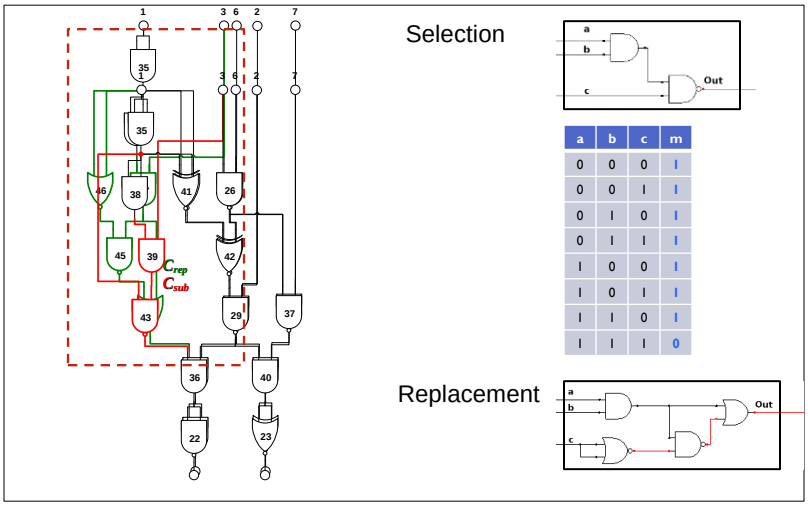
1) Enumerating all candidate subcircuits ($O(n!)$, n = # of gates)
2) Identifying known (library) components from candidates

We implement a version of the White algorithm[5] ($O(n^3)$) to perform component identification

- Two qualities of the obfuscator in view:
  - Given a publicly known algorithm (Kerckhoff's principle), what effect does knowledge of the algorithm have on adversarial analysis?
  - Given the distribution of circuits produced by the algorithm, do variants have measurable component hiding?

- Maximizing Randomness
  - Adversary does not benefit appreciably by knowledge of the obfuscating algorithm
  - Variants may or may not actually demonstrate component hiding

- Maximizing Determinism
  - Adversary can use knowledge of the technique as input to the deobfuscating algorithm
  - Determinism can target the actual security property, i.e., component hiding



Component hiding manifests as an *artifact* of small, iterative selection/replacements in some experimental configurations

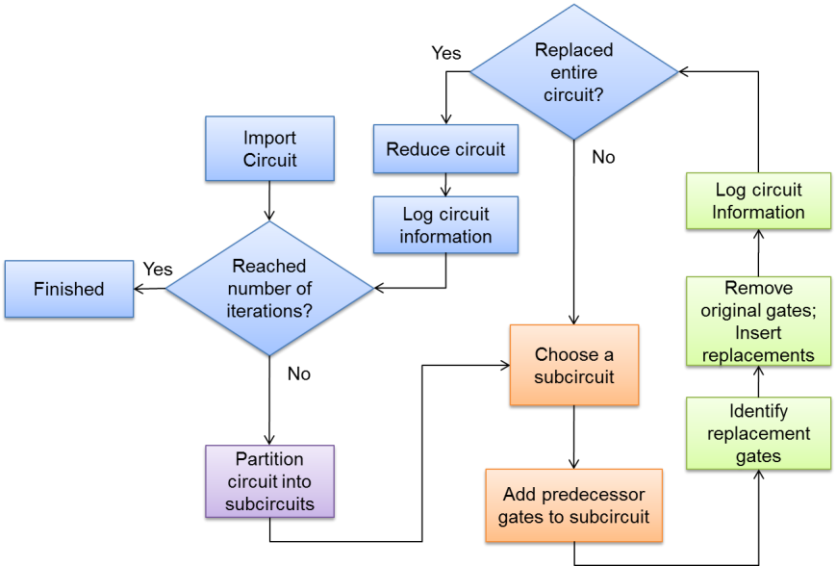
Observations from Empirical Study

- Selection size and replacement size influence manifestation of hiding properties
- Goal for replacement:
  - Uniform, random selection possibility from ALL possible circuits
  - Replacement libraries are static, generated out of band
- Limitation: generating FULL circuit libraries for 4-5 gate circuits is the practical/workable limit
  - Disk storage/indexing/query time/generation time become issues
  - # of circuits related to integer series A005439, A00366 [the number of Boolean functions of *n* variables whose ROBDD contains at least *n* branch nodes]

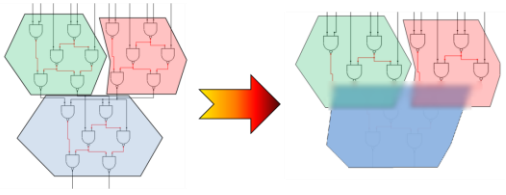
# of GATES

g	1	2	3	4	5	6	7
A000366	1	2	7	38	295	3098	42271
A005439	2	8	56	608	9440	198272	5410688
A000366 * 6^{n-1}	6	72	1,512	49,248	2,293,920	144,540,288	11,833,174,656

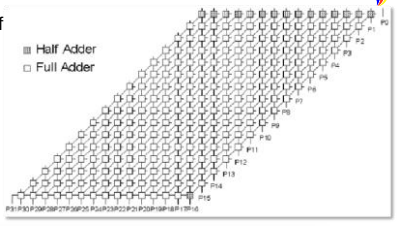
Component Fusion: A Deterministic Approach

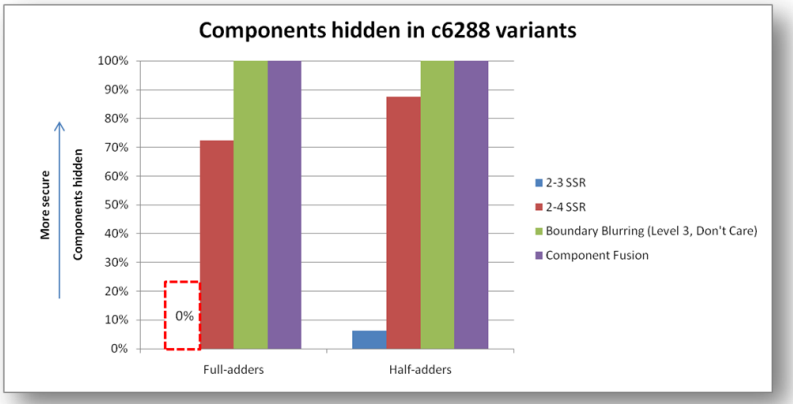


- Deterministic selection
  - Ensures replacement of entire circuit every experiment
    - Partitions the circuit into subcircuits
  - Hides known existing information
    - Uses component definitions to partition subcircuits
  - Ensures selection/replacement operations will overlap
    - Adds predecessor gates to each subcircuit
- Deterministic replacement
  - Uses a randomized circuit synthesizer
    - Increases the speed of finding replacements
    - Implements subcircuit connections as a *virtual black box*



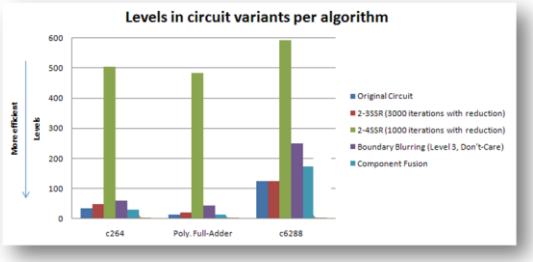
- c6288 ISCAS-85 Benchmark 16-bit multiplier
  - Composed of 224 full adder components and 16 half adder components (hard test case)
  - With no protection, all components identified with a single pass in 1.15 minutes of ID algorithm
  - With component fusion, same ID algorithm does not identify any adder/half-adder components
  - 50 experiments using random (SSR), boundary blur (another deterministic method [6]), and component fusion

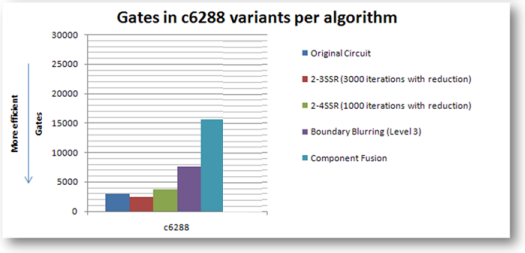


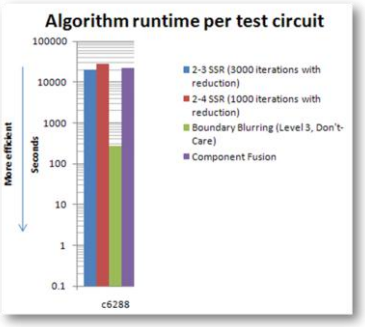


Component Fusion: Empirical Results

- Average efficiency of obfuscation algorithm and variants







~2 hours per variant

Tradeoffs: speed/delay (levels) vs. size/power (gates)

Conclusions

- Component fusion improves component recovery results 37% over the best random selection/replacement technique
- Gate size in variants was on average 350% larger than the original circuit; levelization ~75% increase
- Future work
  - Reduce variant size further using integrated logic reduction techniques
  - Richer set of circuits…
  - Integrate random method with component fusion and other deterministic techniques
  - Integrate other analysis methods for component ID (machine learning, formal approaches like abstract interpretation)
  - Measure other attack vectors/analysis methods for signals, topology, control recovery

This research was supported in part by

Air Force Office of Scientific Research
AFOSR

- [1] Nohl, Karsten, David Evans, Starbug Starbug, and Henryk PlÄotz. Reverse-engineering a cryptographic RFID tag". *SS'08: Proceedings of the 17th conference on Security symposium, 185{193. USENIX Association, Berkeley, CA, USA, 2008*
- [2] Hansen, M., Yalcin, H., and Hayes, J. Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering. *IEEE Design & Test of Computers*, 16, 3 (1999), 72–80.
- [3] Kim, Yong C. and J. Todd McDonald. "Considering Software Protection for Embedded Systems". *Crosstalk The Journal of Defense Software Engineering,* 22(6):4-8, 2009.
- [4] Chikofsky, E. and Cross, James H., Reverse engineering and design recovery: a taxonomy. *IEEE Software,* 7(1):13-17, 1990.
- [5] White, J. L., Wojcik, A. S., Chung, M., and Doom, T. E. 2000. Candidate subcircuits for functional module identification in logic circuits. In *Proceedings of the 10th Great Lakes Symposium on VLSI* (Chicago, Illinois, United States, March 02 - 04, 2000). GLSVLSI '00. ACM, New York, NY, 34-38.