Developing a Deterministic Polymorphic Circuit Generator Using Boolean Logic Representation

Can we create a deterministic circuit generation algorithm that approaches a uniform random selection from the set of all circuits that implement a specific function?

Trinity Stroud and J. Todd McDonald
University of South Alabama, School of Computing

tls1627@jagmail.southalabama.edu, jtmcdonald@southalabama.edu

As the presence of software and hardware becomes steadily more pronounced in the many and varied applicable areas of society today, the theft of intellectual property (IP) embedded in such technology has increasingly become a problem. Obfuscation is the act of transforming a piece of software or hardware such that recovering information related to the original function or structure is made more difficult. In the context of circuits and software protection, this process of obfuscation involves the selection of subcircuits within a circuit and the replacement of said subcircuits with functionally equivalent variants. In this research, we design and implement an obfuscation algorithm that operates by iteratively performing random Boolean logic expansions (RBLE) on a given circuit's Boolean expression in order to preserve and conceal its function. This research compares the effectiveness of this RBLE algorithm with two random selection/replacement obfuscation algorithms involving static circuit libraries and circuit generation.

Introduction

Intellectual property (IP) is currently embedded in both software and hardware that are used in almost every area of society today. As companies can typically have billions of dollars invested in such IP, the theft of IP has become a major concern for tech companies and countries around the world.

This research project concerns polymorphism and circuit protection for the purpose of approaching an efficient obfuscation algorithm, which could serve as a defense against adversarial reverse engineering and, ultimately, IP theft.

Selection/Replacement Algorithms

A selection/replacement obfuscation algorithm [1] involves iteratively replacing parts of a circuit with functionally equivalent variants with some different structure until a desired level of security is reached and the program is deemed to be obfuscated.

Our RBLE algorithm performs selection on a circuit's Boolean expression, randomly determines from a list of Boolean logic laws one to apply to the subexpression that would alter the expression's form while preserving its function, and performs replacements with the resulting subexpression.

The following steps detail the expansion of a Boolean expression belonging to the circuit family C2-1-1 to an expression representing the C2-1-6 circuit of the same function, where CX-Y-Z is the circuit family for those circuits of X inputs, Y outputs, and Z gates:

```
o1 = (i0 * i1)                                   size(C)=1
1: ((i0+i0)* i1)                          rule 8, size(C')=2
2: (((i0+i0)+0)*i1)                       rule 11, size(C')=6
3: (((i0 + i0)+(i1*0))*i1)                 rule 1, size(C')=7
4: (((i0 + i0)+(i1*(i0^i0)))*i1)           rule 3, size(C')=5
5: (((i0+i0)+(i1*(i0^i0)))*(i1*i1))        rule 7, size(C')=6
o1 = (((i0+i0)+(i1*(i0^i0)))*(i1*i1))            size(C')=6
```

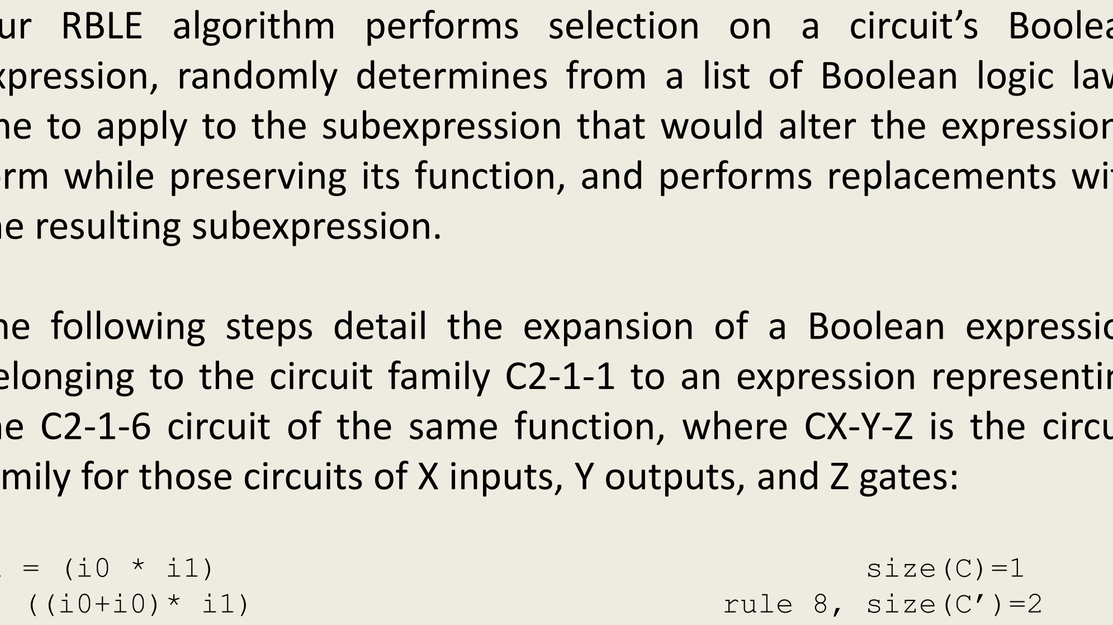


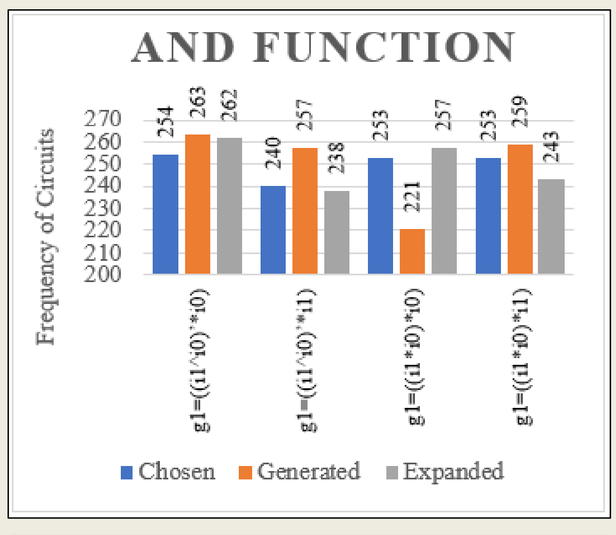
Figure 1: C2-1-1 Partitioned by Function

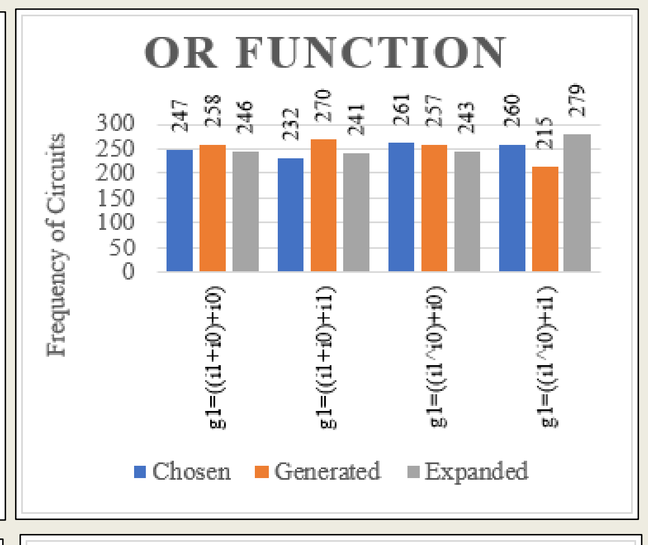
Methodology & Results

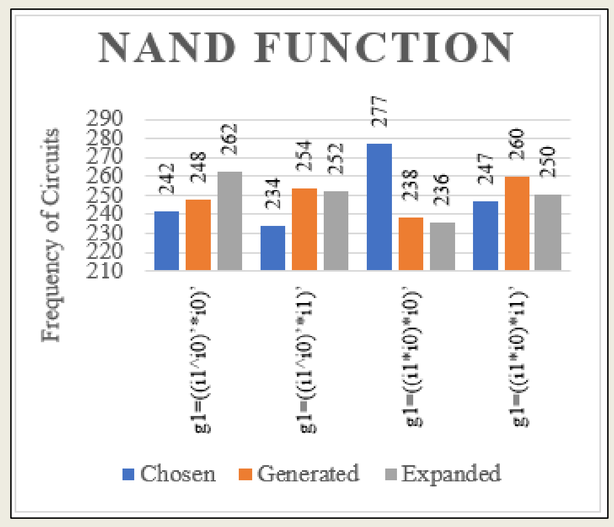
In order to gauge the effectiveness of this RBLE algorithm in relation to a circuit generator, which randomly generates a circuit of a given size, and a circuit chooser, which randomly picks a circuit from a previously enumerated static library, we performed experiments to track which circuits were selected as replacements by these algorithms for each of the 4 functions of C2-1-1 that can be expanded to their respective functions within C2-1-2.

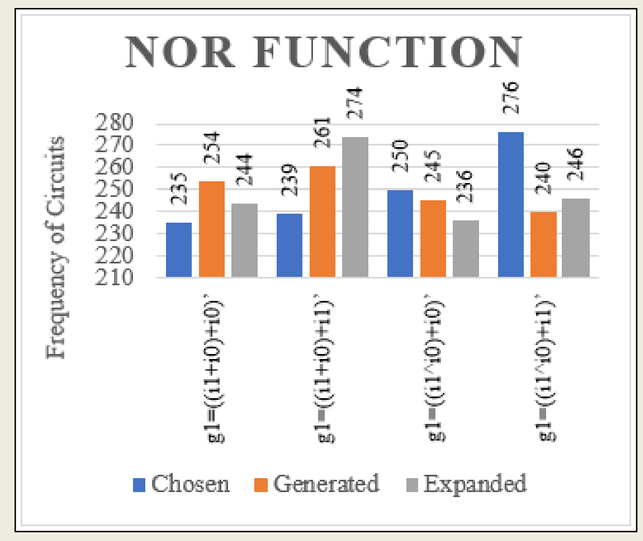
As this was done for each of the 3 obfuscation algorithms, this resulted in 4 sets of 3 circuit distributions each being produced. We then compared the distributions of circuits selected in order to determine how closely the RBLE algorithm approaches a uniform random selection.

The following tables detail the frequencies with which of each of the circuits in each of the 4 functions examined were used as replacements for their functionally equivalent C2-1-1 variants by the 3 different obfuscation algorithms in 1000 trials:









The distribution of circuits generated or chosen by each the obfuscation algorithms are shown to be relatively equal no matter the function, circuit, or obfuscation approach used, with no circuit for any function being heavily favored by any obfuscation approach.

References

[1] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," IEEE Transactions on Computers, vol. C-35, no. 8, pp. 677-691, Aug. 1986, doi: 10.1109/TC.1986.1676819.