Evolving Cryptographically Sound Boolean Functions

Stjepan Picek
Faculty of Electrical
Engineering and Computing
Unska 3
Zagreb, Croatia
stjepan@computer.org

Domagoj Jakobovic
Faculty of Electrical
Engineering and Computing
Unska 3
Zagreb, Croatia
domagoj.jakobovic@fer.hr

Marin Golub
Faculty of Electrical
Engineering and Computing
Unska 3
Zagreb, Croatia
marin.golub@fer.hr

ABSTRACT

This paper explores the evolution of Boolean functions for a cryptographic usage, with genetic algorithms and genetic programming. We also experiment with a new mutation operator and a new kind of initialization process. Results obtained show that those modifications can help in obtaining better solutions. The results indicate that it is possible to obtain high quality Boolean functions with algorithms that are not tailor-made for this purpose. Additionally, among the algorithms tested, the best performance was obtained with variations of genetic programming.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*heuristic methods*

General Terms

Algorithms, Experimentation

Keywords

Heuristic Methods, Genetic Algorithms, Genetic Programming, Boolean Functions, Cryptography, Experimental Results

1. INTRODUCTION

One of interesting real-world applications of evolutionary computation (EC) methods is the creation of Boolean functions with the properties relevant for the area of cryptography. Boolean functions for the use in cryptography can be designed by random search, algebraic construction or by heuristic methods [5]. Boolean functions and S-boxes (S-box can be regarded as a vectorial Boolean function) are used to introduce nonlinearity into otherwise linear systems. For a detailed description of block and stream ciphers refer to [6].

The goal of our paper is to explore the possibilities for the development of good Boolean functions via EC methods. That is done using genetic algorithm (GA) and genetic programming (GP).

2. BASIC PROPERTIES OF BOOLEAN FUNCTIONS

Boolean function is a mapping from $\{0,1\}^n$ to $\{0,1\}$. Vectorial Boolean function or $n \times m$ S-box is a mapping from $\{0,1\}^n$ to $\{0,1\}^m$. An $n \times m$ S-box can be constructed by combining m Boolean functions.

Basic and unique representation of a Boolean function is a truth table (TT). When the total order is assigned, a Boolean function f with n inputs has a truth table with 2^n elements, where each element $\in \{0,1\}$. Boolean functions can be also uniquely represented with algebraic normal form (ANF) and Walsh spectrum (W_s).

In this paper we considered balancedness (BAL) [2], nonlinearity (NL) [1], algebraic degree (deg) [8], correlation immunity (CI) [7], algebraic immunity (AI) [7], propagation criterion (PC) [4], absolute indicator (AC) [9] and sum-of-square indicator (SSI) [9] properties of Boolean functions. For a Boolean function to have good cryptographic properties we want it to be balanced, with high nonlinearity, algebraic degree, algebraic immunity, correlation immunity, low absolute indicator and sum-of-square indicator.

It is not possible to create a Boolean function with all the optimal properties. Connections between some properties are known, while relations between some others are still not clear. For a more detailed discussion about trade-off between properties and their attainable values refer to [1] [5].

3. ENVIRONMENTAL SETTINGS AND RESULTS

Genetic algorithm and genetic programming are used in their standard form. In all the experiments, maximizing the value of the fitness function is the objective. Common parameters are the following: the size of Boolean function is 8 (the size of the truth table is 256), number of independent runs for each experiment is 30 and the population size is 500. We experimented with two encodings: the first is a bit-string encoded truth table (abbreviated GA), and the second is a genetic programming tree with Boolean primitives (GP). Two selection schemes are also used: a steady-state tournament selection (SST) and a generational roulette-wheel (RW).

The abbreviation of an algorithm in Table 1 can contain 'BAL' which means that the *balanced mutation* is used (mutation that preserves the balancedness of a truth table), '+VAR' represents the adaptive mutation rate (where the probability is increased as the evolution starts to stagnate), and 'ORT' means the initial population is created with an orthogonal array. The algorithms are applied to 2 fitness functions: the first one considers only balancedness

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

Table 1: Results of the Experiments
Best Solutions

Algorithm	$(NL, DEG, AI, CI, AC, SSI, PC, W_s)$
Fitness_1 = BAL + NL	
GA_SST+VAR_ORT	(114, 7, 4, 0, 56, 126976, 0, 145)
GP_SST+VAR	(116, 6, 3, 0, 32, 87040, 0, 90)
Fitness_2 = Fitness_1 + AI + CI + DEG	
GA_SST_BAL	(114, 7, 4, 0, 48, 128128, 0, 143)
GP_SST	(116, 7, 3, 0, 40, 95104, 0, 101)

Best Algorithm Statistics

Algorithm	Min	Max	Mean	Stdev
GA	114.5	120.3	115.1	1.1
GP	112.3	119.4	116.6	1.8
GA_BAL	114.8	120.9	116.1	1.9

and non-linearity, while the second one adds three other criteria. Table 1 shows the used criteria, as well as all the properties of the best solutions (in brackets). Details about the experiments and results are available online at http://gp.zemris.fer.hr/boolean/.

3.1 Results

With the objective to find the best possible individuals, stopping criterion is 300 generations without improvement for GA and 30 generations for GP. To try to find the best algorithm, we conduct a statistical analysis on 3 algorithms that produced the best solutions. As a fitness function we use Fitness_2 and steady state tournament selection for all the algorithms since the results obtained for that set of parameters are the best. The results are displayed in the lower part of Table 1.

4. CONCLUSIONS

Finding Boolean functions with good cryptographic properties is a difficult problem. As expected, every modification of basic GA or GP algorithms displayed differences in the performance of the algorithm. The first fitness function displayed very good results and some solutions even outperform functions from existing research. Simple fitness has the advantage that there are no conflicts between variables, and some high quality properties inherently mean that other properties will also be good. In Fitness_2 function we tried to control more properties of a Boolean function. That approach showed the best results in average. Here, it could be prudent to use a weighted fitness function approach. Modifications in the mutation operator and initialization process displayed good performance and should be considered in future research of this kind.

The results of statistical analysis show that there are significant statistical differences in the performance between the algorithms. Research on the evolution of cryptographically good Boolean function was mostly reserved for a cryptographic perspective: to find the best possible solution. In that kind of research, evolutionary computation is just a tool so there is little data on the performance of different algorithms. Here, we tried to find cryptographically good Boolean function but with the emphasis on the evolutionary computation side.

5. REFERENCES

[1] A. Braeken. *Cryptographic Properties of Boolean Functions and S-Boxes.* PhD thesis, Katholieke Universiteit Leuven, 2006.

[2] L. Burnett. *Heuristic Optimization of Boolean Functions and Substitution Boxes for Cryptography.* PhD thesis, Faculty of Information Technology, Queensland University of Technology, 2005.

[3] L. Burnett, W. Millan, E. Dawson, and A. Clark. Simpler methods for generating better boolean functions with good cryptographic properties. *Australasian Journal of Combinatorics*, 29:231–247, 2004.

[4] T. W. Cusick and P. Stanica. *Cryptographic Boolean Functions and Applications.* Elsevier Inc., San Diego, USA, 2009.

[5] K. Goossens. Automated creation and selection of cryptographic primitives. Master's thesis, Katholieke Universiteit Leuven, 2005.

[6] J. Katz and Y. Lindell. *Introduction to Modern Cryptography.* Chapman and Hall/CRC, Boca Raton, 2008.

[7] F. Laffite. *The boolfun Package: Cryptographic Properties of Boolean Functions.*

[8] M. Read. Explicable boolean functions. Master's thesis, Department of Computer Science, The University of York, 2007.

[9] X. Zhang and Y. Zheng. Gac-the criterion of global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.