Brill Academic Publishers
P.O. Box 9000, 2300 PA Leiden,
The Netherlands

Lecture Series on Computer and Computational Sciences
Volume 1, 2006, pp. 1-4

Designing S-boxes through Evolutionary Computation

E.C. Laskari[†,♭], **G.C. Meletiou**[‡,♭], **M.N. Vrahatis**[†,♭,1]

[†]Computational Intelligence Laboratory, Department of Mathematics,
University of Patras, GR-26110 Patras, Greece

[‡]A.T.E.I. of Epirus, P.O. Box 110, GR–47100 Arta, Greece

[♭]University of Patras Artificial Intelligence Research Center (UPAIRC),
University of Patras, GR-26110 Patras, Greece

Received 1 July, 2006; accepted in revised form 31 July, 2006

Abstract: Substitution boxes (S-boxes) are of major importance in cryptography as they are used to provide the property of confusion to the corresponding cryptosystem. Thus, a great amount of research is devoted to their study. In this contribution, a new methodology for designing strong S-boxes is studied and two Evolutionary Computation methods, the Particle Swarm Optimization and the Differential Evolution algorithm are employed to tackle the problem at hand.

Keywords: Evolutionary Computation, S-boxes, Boolean functions, Nonlinearity, Autocorrelation, Particle Swarm Optimization, Differential Evolution

Mathematics Subject Classification: 90C15, 90C56, 90C90

1 Introduction

Substitution boxes (S-boxes) are basic components of many contemporary cryptosystems. They are nonlinear mappings in the sense of Boolean structure that take as input a number of bits and transform them into some number of output bits. S-boxes are of major importance in cryptography as they are used to provide the property of confusion to the corresponding cryptosystems and, in some cases, they comprise their only nonlinear part. As the security of the cryptosystems utilizing S-boxes mainly depends on their choice, a great amount of research is devoted into the design of good S-boxes. The effectiveness of evolutionary heuristics, such as hill climbing [7], Genetic Algorithms [8] and Simulated Annealing [1], on the design of strong regular S-boxes was recently studied with promising results.

Evolutionary Computation (EC) methods are inspired from evolutionary mechanisms such as natural selection and social and adaptive behavior. Genetic Algorithms, Genetic Programming, Evolution Strategies, Differential Evolution, Particle Swarm Optimization and Ant Colony Optimization are the most commonly used paradigms of EC. An advantage of all EC methods is that they do not require objective functions with good mathematical properties, such as continuity or differentiability. Therefore, they are applicable to hard real-world optimization problems that involve discontinuous objective functions and/or disjoint search spaces [2, 3, 12]. Furthermore, EC methods have tackled effectively and efficiently a number of hard and complex problems in

[1]Corresponding author: e-mail: vrahatis@math.upatras.gr, Phone: +30 2610 997374, Fax: +30 2610 992965

numerous scientific fields [4, 6, 9, 10]. Thus, in this contribution, a new methodology for the design of strong regular S-boxes is presented, and two Evolutionary Computation methods, namely the Particle Swarm Optimization method (PSO) and the Differential Evolution method (DE), are employed to address the problem at hand.

The first results using the traditional quality measures for S-boxes indicate that the proposed methodology is effective. Moreover, we will extend the study of the proposed methodology to spectrum based cost functions.

2 Theoretical Background and Problem Formulation

Before presenting the problem formulation, let us provide the necessary theoretical notions of S-boxes. Let $f : \mathbb{B}^n \mapsto \mathbb{B}^m$ denote an S-box mapping n Boolean input values to m Boolean output values. In the case where $m = 1$, f is a single output Boolean function, and if the number of inputs mapping to 0 is equal to the number of inputs mapping to 1, the Boolean function is called *balanced.* Balance is an important property for Boolean functions used in cryptographic applications as it ensures that the function cannot be approximated by any constant function. Generalizing the notion of balance for the multiple output functions, if each possible output value of m binary components appears equally as output of the function, i.e., 2^{n-m} times, then the function is called *regular.* In case of S-boxes with $n = m$, the S-boxes are called *bijective* and all possible outputs appear exactly one time each. For simplicity reasons, in the rest of this contribution we will refer to single output Boolean functions just as Boolean functions and to the multiple output case as S-boxes.

For the design of cryptographically strong Boolean functions and S-boxes two traditional quality measures exist, the *nonlinearity* and the *autocorrelation.* In order to define these two measures, the definitions of *linear* and *affine* Boolean functions, the *Walsh Hadamard Transform*, the *polarity truth table* and the *Parseval's theorem* are needed. A *linear Boolean function* selected by $\omega \in \mathbb{B}^n$ is denoted by $L_\omega(x) = \omega_1 x_1 \oplus \omega_2 x_2 \oplus \cdots \oplus w_n x_n$, where $w_i x_i$, for $i = 1, \ldots, n$, denotes bitwise AND of the ith bits of ω and x and $\oplus$ denotes bitwise XOR. The set of *affine Boolean functions* consists of the set of linear Boolean functions and their complements, i.e., all functions of the form $A_{\omega,c} = L_\omega(x) \oplus c$, where $c \in \mathbb{B}$.

For a Boolean function f a useful representation is the *polarity truth table* which is given by $\hat{f}(x) = (-1)^{f(x)}$. The *Walsh Hadamard Transform* (WHT) of a Boolean function f is defined as $\hat{F}_f(\omega) = \sum_{x \in \mathbb{B}^n} \hat{f}(x) \hat{L}_\omega(x)$. The WHT is a measure for the correlation among the Boolean function f and the set of linear Boolean functions. In general two Boolean functions f, h are considered to be *uncorrelated* when $\sum_x \hat{f}(x)\hat{h}(x) = 0$. The maximum absolute value taken by the WHT is denoted as $WH_{\max}(f) = \max_{\omega \in \mathbb{B}^n} \left|\hat{F}_f(\omega)\right|$ and is related to the nonlinearity of f. Specifically, the *nonlinearity* N_f of a Boolean function f is defined as $N_f = \frac{1}{2}(2^n - WH_{\max}(f))$. Regarding the WHT, as proved by the *Parseval's theorem* it holds that $\sum_{\omega \in \mathbb{B}^n} (\hat{F}_f(\omega))^2 = 2^{2^n}$, which results in $WH_{\max}(f) \geqslant 2^{n/2}$. The set of functions with nonlinearity equal to this lower bound are called *bent* functions but they are never balanced. The set of balanced functions with maximum nonlinearity and the determination of bounds for balanced functions are important open problems [7].

The autocorrelation of a Boolean function is a measure of its self-similarity and is defined by $\hat{r}_f(s) = \sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{f}(x \oplus s)$, where $s \in \mathbb{B}^n$. The maximum absolute value taken by the autocorrelation is denoted as $AC_{\max}(f) = \max_{s \in \mathbb{B}^n \setminus \{0^n\}} \left| \sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{f}(x \oplus s)\right|$.

The nonlinearity and autocorrelation measures for Boolean functions can be extended to S-boxes by defining a set of functions f_β, that are linear combinations of the outputs of the corresponding S-box. Specifically, for an S-box $f : \mathbb{B}^n \mapsto \mathbb{B}^m$, a function $f_\beta(x)$, for each $\beta \in \mathbb{B}^m$, that is linear combination of the m outputs of f, is defined, as $f_\beta(x) = \beta_1 f_1(x) \oplus \cdots \oplus \beta_m f_m(x)$, where $f_j(x)$, for $j = 1, \ldots, m$ denotes the jth bit of the S-box's output. There are $2^m - 1$ non trivial functions

f_β, and for each such function the WHT value, denoted as $\hat{F}_\beta(\omega)$, and the autocorrelation value, denoted as $\hat{r}_\beta(s)$ are obtained directly by the former definitions. Thus, the nonlinearity of an S-box is the lowest nonlinearity over all $2^m - 1$ corresponding $f_\beta(x)$ functions, and the autocorrelation of the S-box is the highest autocorrelation over the same $f_\beta(x)$ functions.

Research on the design of Boolean functions and S-boxes addressing the problem as an optimization task, aims at obtaining functions with high nonlinearity or/and low autocorrelation. Regarding the nonlinearity of an S-box, the corresponding optimization problem is minimizing subject to f the function

$$\max_{\beta \in \mathbb{B}^m, \omega \in \mathbb{B}^n} \left| \hat{F}_\beta(\omega) \right|, \tag{1}$$

and, for the autocorrelation the problem is formulated as minimizing subject to f the following function

$$\max_{\beta \in \mathbb{B}^m \setminus \{0^m\}, s \in \mathbb{B}^n \setminus \{0^n\}} \left| \hat{r}_\beta(s) \right|. \tag{2}$$

3 The Proposed Methodology and Discussion

To address the problem of designing regular S-boxes as an optimization task, we employ two Evolutionary Computation methods, the Particle Swarm Optimization method (PSO) and the Differential Evolution (DE) method. As all Evolutionary Computation optimization methods, both PSO and DE methods utilize a population of possible solutions (of the problem at hand), which is randomly initialized within the search space. Sequentially, an objective function is used to evaluate each population member, and evolutionary operators are employed to evolve the population members in order to produce offsprings with better values of the objective function. This procedure is repeated until the population converges into one solution or until a predefined value of the objective function is obtained.

For the present case problem, a population of regular S-boxes is randomly initialized by randomly swapping the components of an array containing 2^{n-m} copies of each of the 2^m possible outputs [8]. For the representation of each possible solution various techniques can be used. One of them is the usage of the truth table of the corresponding S-box output in decimal form. In this way the optimization problem is transformed into a discrete optimization task. Both, the Particle Swarm Optimization method and the Differential Evolution method, have proved to be efficient in handling discrete optimization tasks [4, 5, 6] through the technique of rounding off the real values of the solution to the nearest integer [11].

For the evaluation of the proposed solutions the traditional measures of nonlinearity and autocorrelation, given by Eqs. (1) and (2), are initially used. Furthermore, the effectiveness of the presented methodology using the new spectrum based cost functions proposed in [1] is studied.

An important point in the proposed methodology is the evolving of different S-boxes to produce offsprings. These offsprings can either be regular S-boxes or not. In previously published research [1, 7, 8], in order to obtain regular S-boxes as solutions, an initialization with random regular candidate solutions takes place and then the optimization method utilized, is responsible for maintaining the regularity of the produced offsprings. In this contribution, a new technique for the construction of regular S-boxes is proposed. Specifically, we allow the exploration by the employed method of all the search space, i.e., search among feasible (regular) and unfeasible (non regular) solutions, but for the evaluation of a possible solution, the proposed candidate is transformed to the closest (by means of Hamming distance) regular one, for every wrong component. Thus, the method is allowed to perform better exploration of the search space and moreover its dynamic is retained.

The first results of the proposed approach, using the traditional measures of nonlinearity and autocorrelation for regular bijective S-boxes, are comparable to the corresponding of other more complex heuristic methods using the same objective functions. Furthermore, the new methodology required less computational cost to obtain the same results in almost all cases. Thus, the proposed methodology can be considered effective in tackling the problem of designing strong S-boxes. Of course, more experiments utilizing also the new spectrum based cost functions are required to conclude on the efficiency of the new approach and will be presented.

References

[1] J.A. Clark, J.L. Jacob, and S. Stepney. Searching for cost functions. In *CEC 2004: International Conference on Evolutionary Computation, Portland OR, USA, June 2004*, pages 1517–1524. IEEE, 2004.

[2] D.B. Fogel. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence.* IEEE Press, Piscataway, NJ, 1995.

[3] J. Kennedy and R.C. Eberhart. *Swarm Intelligence.* Morgan Kaufmann Publishers, 2001.

[4] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, and M. N. Vrahatis. Evolutionary computation based cryptanalysis: A first study. *Nonlinear Analysis: Theory, Methods and Applications*, **63** e823–e830(2005).

[5] E. C. Laskari, G. C. Meletiou, Y. C. Stamatiou, and M. N. Vrahatis. Applying evolutionary computation methods for the cryptanalysis of feistel ciphers. *Applied Mathematics and Computation*, to appear, (2006).

[6] E.C. Laskari, K.E. Parsopoulos, and M.N. Vrahatis. Particle swarm optimization for integer programming. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation*, pages 1576–1581, Hawaii, HI, IEEE Press, 2002.

[7] W. Millan. How to improve the nonlinearity of bijective S-boxes. *LNCS*, **1438** 181–192 (1998).

[8] W. Millan, L. Burnett, G. Carter, A. Clark, and E. Dawson. Evolutionary heuristics for finding cryptographically strong S-boxes. *LNCS*, **1726** 263–274(1999).

[9] K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, **1**(2–3) 235–306(2002).

[10] V.P. Plagianakos and M.N. Vrahatis. Parallel evolutionary training algorithms for "hardware–friendly" neural networks. *Natural Computing*, **1**(2–3) 307–322(2002).

[11] S.S. Rao. *Engineering Optimization–Theory and Practice.* Wiley Eastern, New Delhi, 1996.

[12] H.-P. Schwefel. *Evolution and Optimum Seeking.* Wiley, New York, 1995.