This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier

Human behaviour based optimization supported with self-organizing maps for solving the S-box design Problem

RICARDO SOTO[1], BRODERICK CRAWFORD[1], FRANCISCO GONZÁLEZ[1], RODRIGO OLIVARES[2],

[1]Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2241, Valparaíso, Chile. +56 32 2273636 (e-mail: ricardo.soto, broderick.crawford, francisco.gonzalez@pucv.cl)

[2]Universidad de Valparaíso, Blanco 951, Valparaíso, Chile. +56 32 2507000 (e-mail: rodrigo.olivares@uv.cl)

Corresponding author: First Francisco González (e-mail: francisco.gonzalez@pucv.cl).

ABSTRACT The cryptanalytic resistance of modern block and stream encryption systems mainly depends on the substitution box (S-box). In this context, the problem is thus to create an S-box with higher value of nonlinearity because this property can provide some degree of protection against linear and differential cryptanalysis attacks. In this paper, we design a scheme built on a human behavior-based optimization algorithm, supported with Self-Organizing Maps to prevent premature convergence and improve the nonlinearity property in order to obtain strong 8×8 substitution boxes. The experiments are compared with S-boxes obtained using other metaheuristic algorithms such as Ant Colony Optimization, Genetic Algorithm and an approach based on chaotic functions and show that the obtained S-boxes have good cryptographic properties. The obtained S-box is investigated against standard tests such as bijectivity, nonlinearity, strict avalanche criterion, bit independence criterion, linear probability and differential probability, proving that the proposed scheme is proficient to discover a strong nonlinear component of encryption systems.

INDEX TERMS Cryptography, Substitution box, Self-Organizing Maps, Metaheuristics.

I. INTRODUCTION

One of the primary techniques that allow us to ensure security over digital information is Cryptography. We can generally characterize it into a symmetric cipher and asymmetric cipher. An asymmetric cipher utilizes a different key for the processes of encryption and decryption. The public key encrypts the plain text into cipher text and private key reverts this process. Symmetric block cipher uses the same key for encryption and decryption. The substitution box (S-box) is one of the most important components in symmetric cryptographic systems such as Data Encryption Standard (DES), and Advanced Encryption Standard (AES). Claude Shannon proves that a strong cryptographic system must integrate two fundamental properties: confusion and diffusion [1]. A S-Box provides the first property, *hiding the relationship between secret key and cipher text*. This is the only nonlinear component that determines the strength of the entire block encryption algorithm. Therefore, the main concern in the construction of strong cryptosystem is the design of a secure S-box. The strength of an S-box can be measured using several criteria, for example nonlinearity, balance, strict avalanche criterion, XOR profile, bit independent criterion, transparency order [2]. These criteria allow us to assert the resistance of a substitution box to cryptanalysis [3] [4]. A large number of works focuses on using the nonlinearity as a fitness value as so do we. General methods are used for the design of substitution box e.g. random methods [5], algebraic constructions, chaotic maps and heuristic methods. As examples of the last method, we can mention works carried out using Ant Colony Optimization [6], Genetic Algorithm [7], [8], Simulated Annealing [9], [10], Leaders and Followers [11] and Particle Swarm Optimization [12]. In this paper, we use a novel optimization algorithm based on the human behavior [13] integrated it with Self-Organizing Maps in order to avoid the problem of premature convergence [14]. This issue occurs when the population of the optimization algorithm reaches a suboptimal state where the operators can no longer produce new solutions which outperform the best found solution so far. In that case, the search process will likely be trapped in a region containing a non-global

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

optimum. A simple explanation for the occurrence of this phenomenon is the loss of diversity. To overcome this, we train the Self-Organizing Map in order to allow us to assess the intensity of the exploration process on the regions through which the optimization algorithm has passed and take an appropriate decision on the search process. The experiments generated by the hybrid proposal show that the generated S-box has good cryptographic properties in contrast with S-boxes obtained using other algorithms such as Ant Colony Optimization, Genetic Algorithm, and constructions based on chaotic functions. The resulting S-box is evaluated against cryptographic criteria such as bijectivity [15], [16], nonlinearity, strict avalanche criteria, differential uniformity, proving to be a competitive scheme for the problem at hand.

This work is organized as follows: The related work is introduced in the next section. Section III describes the substitution box problem. Section IV explains optimization algorithm based on human behavior. Section V deals with the Self-Organizing Maps. Section VI details the integration of the two techniques. Section VII shows the experimental results. Finally, conclusions and future works are exposed in Section VIII.

II. RELATED WORK

During the last decades, the design of substitution boxes, have gained relevance in two categories, the design through optimization algorithms and the approaches based in chaotic systems. Optimization algorithms aim to build substitution boxes through an iterative process, focusing to improve a S-box in relation to one or more properties. The chaotic systems approach has proven to be a tool valid for the construction of substitution boxes since the high degree of sensibility of its initial parameters allows a high divergence in the results and this means that it is practically impossible to predict their behavior. Regarding works with methods for the generation of substitution boxes, [10] proposed a Simulated Annealing with Hill-Climbing approach and use cost functions based on the whole Walsh Hadamard spectrum and autocorrelation spectrum. They evaluate the results according to nonlinearity, autocorrelation and algebraic degree. [17] proposed an adaptive alternative to static strategies of the hill climbing technique, combining the weak and strong hill climbing techniques. The obtained algorithm, Dynamic Hill Climbing, has the quality of being adaptive, because it can decide when to choose and strong or weak hill climbing. [18] tested that the choice of cost function can dramatically impact on the efficiency of the optimization methods for generating Boolean functions. In the specific case of Simulated Annealing behaves when it is used with a cost function inspired by Parseval's Theorem. [19] presents a method that joins special genetic algorithm with total tree searching, using a special cost function to build S-boxes with dimensions of 5×5, 6×6, 7×7, 8×8. [20] proposed an evolutionary algorithm based on a theorem for the construction of a permutation, that finds S-boxes with good cryptographic criterion such as high nonlinearity, low autocorrelation, low difference, and high algebraic immunity. [21] improved a gradient descent method to decrease the nonlinearity of given vectorial Boolean functions with minimum δ-uniformity, finding substitutions that can be used in modern symmetric algorithms. [5] proposed a random method that choose some fixed set of starting S-boxes. The output S-boxes are obtained by making various compositions of the starting S-boxes. They generate S-boxes of $n \times n$, with $n = 8, 10, 12$, and compare themselves with several random methods. The results show that the rates of good S-boxes among those generated by various methods do not depend substantially on the method of generation. [22] presented a methodology based on the classical Fisher-Yates shuffle technique combined with chaotic map to serve as random number generator for the capable execution of the shuffle technique. The performance of the method is evaluated against bijective property, nonlinearity, strict avalanche criteria and equiprobable I/O XOR distribution. [23] proposed an algorithm to build nonlinear substitution components using a chaotic Boolean fit function, that are employed to image encryption. [24] shows a generation of an S-box with fractional chaotic Rössler system and use the resulting S-box to present a new watermarking technique. The quality of the S-box is evaluated against standard criteria such as bit independence criterion, nonlinearity analysis, strict avalanche criterion, linear approximation probability and differential approximation probability.

[25] used a two-dimensional Gingerbreadman chaotic map and S_8 symmetry group to construct an S-box applied to image encryption applications. [26] detailed the clonal selection algorithm, that is a form of artificial immune algorithm, mixed with a modified hill climbing method. This approach can elaborate large quantity of highly nonlinear bijective S-boxes, with low differential uniformity and low autocorrelation. [27] exposed a chaotic S-box based on the interlace of logistic map and bacterial foraging optimization algorithm. It uses the logistic map to build a set of S-boxes and then executes the optimization algorithm with the cost function combining the nonlinearity and the differential uniformity. [28] utilized a method with the conjunction of Cuckoo Search Algorithm and chaotic maps to enhance the nonlinearity of a S-box. This approach produces an S-box with nonlinearity 109.25, which is a good indicator of strength of the cipher against linear attacks. [12] described a particle swarm optimization algorithm with the generation of initial population using chaotic Renyi map. Their experiments were conducted across different scenarios, with variable population size, number of iterations, and linear increase in inertial weight. The results show that this method complies with standard criterions and propose an image encryption application. [29] presented a scheme with a new discrete compound chaotic system, Logistic-Sine system, to generate S-Box with satisfactory cryptographic performance. [30] propounded a method with Cuckoo Search algorithm that uses a discrete-space chaotic map for the initial population generation, and thus improve the searching performance and convergence speed of Cuckoo Search. The performance

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

of the algorithm was evaluated with bijectivity, nonlinearity, strict avalanche criteria, bit independence criteria, differential uniformity, and linear probability. [31] presented a scheme that initially generates a S-box based on Lorenz map and that is improved through the scan path of Hilbert curve. [32] proposed a new modular approach consists of three operations such as new transformation, modular inverses, and permutation. The obtained S-Box is evaluated against standard criterions such as nonlinearity, fixed points, SAC and BIC properties, differential uniformity and linear approximation probability. [11] shows that the concurrence between exploration and exploitation can affect the quality of the heuristic search and propose a design based on the Leaders and Followers metaheuristic [98], that deals with the problem stated before. They optimize the nonlinearity and transparency order of S-boxes. Using machine learning techniques, they improved the point on which the change between exploration and exploitation must occur.

Our main contribution is to address the substitution box design problem and the premature convergence problem. The resolution of the first problem is made through an optimization algorithm based on human behavior. The second problem will be solved through the application of Self-Organizing Maps. In addition, we incorporate the concept of novelty that allows us to reward or punish solutions, this is done based on the accumulation of information provided by the optimization algorithm, which is processed by the Self-Organizing Maps and allows us to modify the behavior of the optimization algorithm online. The combination of both techniques proves to be competitive in terms of nonlinearity results of the substitution boxes found and also provides a visualization mechanism of the behavior of the optimization algorithm when the latter performs the travel by the search space.

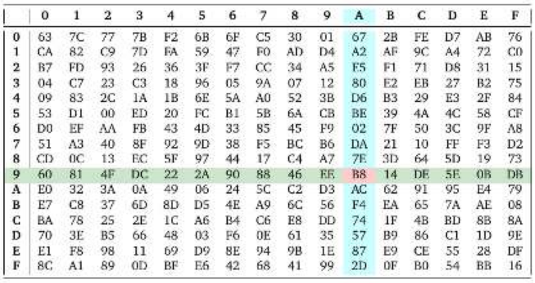
III. SUBSTITUTION BOX

Substitution boxes are a fundamental component in cryptography, mainly, in block ciphers, in order to exhibit a high quality of nonlinearity property [33]. In the case that a substitution box or boxes do not comply with this property, it can be established that the encryption algorithm does not guarantee an adequate level of security [34].

Formally, a substitution box S is a function or correspondence of n input bits to m output bits, $S : Z_2^n \Rightarrow Z_2^m$, that is, an S-box can be viewed as a Vector Boolean function of n input bits and m output bits. When $n = m$ the function is reversible and therefore bijective. However, on many occasions, the substitution box of the block ciphers is not bijective. For example, the Data Encryption Standard (DES) employs S-boxes in which the number of input bits (six) is greater than the number of output bits (four). The number of Boolean functions eligible to design a S-box with n input bits and m output bits is given by $2^{m^{2^n}}$ therefore, for small values of n and m, the search space is quite extensive.

Figure 1 shows the Advanced Encryption Standard(AES) SBox, this SBox presents a non-linearity of 112. The operation of the substitution box is with 8 input bits and 8 output bits, the procedure is as follows:

- Input value: 9A(Hex) ⇒ 10011010(Bin)
- Taking the first 4 bits determine the row 1001 ⇒ row 9, the last 4 bits represent the column 1010 ⇒ column A
- Output value: B8(Hex) ⇒ 10111000



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

FIGURE 1: AES SBox

A. PRELIMINARIES

A Boolean function is a discrete function $f : F_2^n \rightarrow F_2^m$, that maps n input bits, to m output bits. It can be represented with a truth table, in its algebraic form or its decimal form, among others.

An example of a truth table representation with functions of two input variables

x_1	x_2	$\overbrace{x_1x_2}^{f_0}$	$\overbrace{(x_1x_2) \oplus x_1}^{f_1}$	$\overbrace{x_1 \oplus x_2}^{f_2}$
1	1	1	0	0
1	0	0	1	1
0	1	0	0	1
0	0	0	0	0

Algebraic Normal Form (ANF) describes a Boolean function in terms of sum (XOR) and product (AND) of the input variables. The function f is formally defined as follows:

$$\begin{aligned} f(x_1, ..., x_n) = a_0 \oplus a_1x_1 \oplus ... \oplus a_nx_n \oplus a_{12}x_1x_2 \oplus ... \\ ... \oplus a_{n-1n}x_{n-1}x_n \oplus a_{12...n}x_1x_2...x_n \end{aligned} \tag{1}$$

For Example $f(x_1, x_2, x_3) = 1 + x_3 + x_2x_3 + x_1x_2$, with one ouput bit.

x_1	x_2	x_3	$f_{(x_1,x_2,x_3)}$
0	0	1	1
0	0	0	0
0	1	1	1
0	1	0	1
1	0	1	1
1	0	0	0
1	1	1	0
1	1	0	0

The truth table of a Boolean function can be represented by means of a vector with elements in decimal format. This

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

vector will have a length of 2^n, where n is the number of input bits. For example, if the truth table of a function of three input variables and three output variables:

$$\text{Truth Table} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

The corresponding decimal representation is as follows:

$$\text{Truth Table} = \begin{bmatrix} 1 & 4 & 5 & 3 & 6 & 2 & 7 & 0 \end{bmatrix}$$

The Hamming weight of a Boolean function is defined as the amount of 1 present in their truth table representation. In the previous example the Hamming weight is 12.

A Boolean function is balanced if its Hamming weight is exactly 2^{n-1}. From another standing point, if all the elements in their decimal representation, they are unique.

The Hamming distance is the difference between two Boolean functions of the same dimension. It is the number of bits that differs between functions.

Algebraic degree, considering the Algebraic Normal Form in (1), the algebraic degree of an $n - variable$ Boolean function $f(x)$, denoted by $deg(f)$, is the number of variables of the largest product term of the function's ANF having a non-zero coefficient. For example:

$$\begin{aligned} f_1 &= 1 + x_1 + x_1x_2 & deg(f) &= 2 \\ f_2 &= x_2 + x_3 + x_1x_2x_3 & deg(f) &= 3 \end{aligned}$$

A Boolean function with algebraic degree at most one is called an affine Boolean function. The general form of an affine Boolean function of n variables is:

$$\begin{aligned} f_{afin(x_1,x_2,\ldots,x_n)} = a_nx_n + a_{n-1}x_{n-1} + \ldots \\ \ldots + a_2x_2 + a_1x_1 + a_0 \end{aligned} \tag{2}$$

with $a_i \in \{0, 1\}$. If the term constant a_0 is zero, the function is called linear boolean function. For a Boolean function of n variables, there are 2^{n+1} affine boolean functions. For example for $n = 2$, The affine functions are:

$$\begin{aligned} f_1 &= a_0 & f_5 &= a_1x_1 + a_2x_2 \\ f_2 &= a_0 + a_1x_1 & f_6 &= a_1x_1 \\ f_3 &= a_0 + a_2x_2 & f_7 &= a_2x_2 \\ f_4 &= a_0 + a_1x_1 + a_2x_2 & f_8 &= 0 \end{aligned}$$

The Walsh-Hadamard Transformation (WHT), of a n variables Boolean function f, represented by its polarity form $\hat{f}$, is denoted by $\hat{F}_f(w)$ and defined as:

$$\begin{aligned} \hat{F}_f(w) &= \sum_{x \in \mathbb{B}^n} \hat{f}(x)(-1)^{<w,x>} \\ &= \sum_{x \in \mathbb{B}^n} (-1)^{f(x) \oplus <w,x>} \\ &= \sum_{x \in \mathbb{B}^n} \hat{f}(x)\hat{l}_w(x) \end{aligned} \tag{3}$$

where $\hat{l}_w(x)$ is the signed function of the linear function $l_w(x) =< w, x >$.

$\hat{F}_f(w) \in [-2^n, 2^n], \forall w \in \mathbb{B}^n$ and $\hat{F}_f(w)$ is known as a *spectral walsh coefficient*, while the real-value vector of all 2^n spectral coefficients is referred to as the WHT Spectrum. The maximum absolute value, taken by $\hat{F}_f$, is given by: $WHT_{max}(f) = max_{(w \in \mathbb{B}^n)}|\hat{F}_f(w)|$.

Hadamard matrix, is a binary matrix of dimensions $2^n \times 2^n$, whose element of the i-th row and j-th column, is $W(i, j)$.

$$H_{2^n} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix} \tag{4}$$

For example:

$$W_{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$W_{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The nonlinearity of Boolean functions is one of the fundamental cryptographic elements. It measures the distance from a Boolean function to the closest affine Boolean function. Low nonlinearity Boolean functions are susceptible to linear and differential attacks.

Let $f(x)$ a Boolean function. For any affine function $a(x) = a_0 + l(x)$, the distance between $f(x)$ and $a(x)$ is:

$$\begin{aligned} d(f(x), a(x)) &= wt(f(x) \oplus a(x)) \\ &= \sum_{x=0}^{2^n-1} (f(x) \oplus a(x)) \\ &= \frac{1}{2} \sum_{x=0}^{2^n-1} (1 - (-1)^{f(x)+a(x)}) \\ &= 2^{n-1} - \frac{1}{2}(-1)^{a_0} \sum_{x=0}^{2^n-1} (-1)^{f(x)+l(x)} \end{aligned} \tag{5}$$

Usign $l(x) = \langle w, x \rangle = w_1x_1 \oplus w_2x_2 \oplus \ldots \oplus w_nx_n$, where w is the coefficient vector; the above can be written:

$$\begin{aligned} d(f(x), a(x)) &= 2^{n-1} - \frac{1}{2}(-1)^{a_0} \sum_{x=0}^{2^n-1} (-1)^{f(x)+\langle w,x \rangle} \\ &= 2^{n-1} - \frac{1}{2}(-1)^{a_0} S_{(f)}(w) \end{aligned} \tag{6}$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

where $S_{(f)}(w)$ is the Walsh-Hadamard transform of $f(x)$ on w. Notice that $a_0 \in \{0,1\}$, and $\langle w, x\rangle$ represents all affine boolean functions. Then the nonlinearity of $f(x)$:

$$nl(f) = 2^{n-1} - \frac{1}{2} max|S_{(f)}(w)| \tag{7}$$

In this work, the nonlinearity is used as the fitness function to optimize.

IV. HUMAN BEHAVIOR BASED OPTIMIZATION

Currently, humans try to move towards a personal goal in order to achieve a better existence, however, this does not apply to all. We can establish that a good individual is the one who fulfils his purpose, even though the goal itself can change along the time. The purpose of individuals is highly variable, so one individual can achieve his purpose in the study of jazz while another can does it in playing soccer. It is also evident that there are categories of quality among individuals who are dedicated to a particular objective or purpose. In addition, everybody meets a variety of individuals throughout time and uses their concepts and suggestions to higher their lives, in order that any contact is viewed as a gathering with a consultant that may be effective or ineffective. In some things, owing to that consultation, that person may change his skilled field and pursue a far better role in another field to boost himself [13]. The human behavior optimization algorithm takes its inspiration on the ideas mentioned before. Firstly, an initial random population is generated. Then, the individuals or solutions are spread among several fields or groups of purpose. In each field, individuals will try to improve themselves by means of the process described in section 4.2. After, each individual will find a random advisor from the whole population and start to consult with him. Lastly, as we mentioned before, individuals may change the purpose that they are pursuing. This is achieved by the process delineated in section 4.4. To conclude the optimization algorithm the stopping criteria will be checked, and if one of them reaches, the algorithm stops. The Human Behavior Based Optimization (HBBO) consists of the five steps as follows:

- Step 1: Initialization
- Step 2: Education
- Step 3: Consultation
- Step 4: Field changing probability
- Step 5: Finalization

The individuals or solutions of this work represent boolean functions.

A. INITIALIZATION

The initial population is randomly generated and spread among the fields evenly. In an optimization problem with N_{var} variables, an individual is defined as follows:

$$Individual = [x_1, x_2, ..., x_{Nvar}] \tag{8}$$

HBBO generates N_{pop} of individuals, which forms the society, and randomly spreads them among N_{field} of initial fields. The fields are composed by a number of individuals according to the following equation:

$$N_{ind} = round\left\{N_{pop}/N_{field}\right\} \tag{9}$$

where N_{ind} is the number of initial individuals in i-th field.

B. EDUCATION

In the education process, every individual tries to learn and improve itself by moving around the best individual of its field, which is named expert individual and is the one who has the best function value in each field.

Minimum similarity is calculated by the following equation:

$$sMin = k_1 s \tag{10}$$

where k_1 corresponds to a parameter of HBBO, which gives the weight factor and s is the distance between the best individual of the field and the individual evaluated belonging to that field. Maximum similarity is calculated by the following equation:

$$sMax = k_2 s \tag{11}$$

where k_2 correspond to a parameter of HBBO, which gives the weight factor. We calculate a random similarity between the two previous values $sMax$ and $sMin$:

$$sRandom = \alpha(sMax - sMin) + sMin \tag{12}$$

where α is a random number using uniform distribution between 0 and 1. Every component of the individual in the education process, will be replaced by the expert individual's respective component until the similarity between the individuals is greater than the similarity measured in the previous step or no components are to be changed. In case of obtaining a better fitness, the changes will be conserved, otherwise the individual is preserve in the original state.

C. CONSULTATION

All the individuals present in the population, except the best individual of it, will perform the consultation process. This process consists on finding and adviser and start consult with him. The advisor will modify a number of variables of the consulting individual. The number of variables to be changed is subject to the following equation:

$$N_c = round\left\{\sigma \times N_{var}\right\} \tag{13}$$

where σ is the consultation factor, which determines the number of random variables N_c that may be changed during the consultation process. We apply the strategy of conserving the changed variables if the consulting process results in an improvement of the value of the fitness function.

D. FIELD CHANGING PROBABILITY

In each iteration, it is determined if any individual will be transferred from one field to another. The changing probability to each field is calculated employing a rank probability

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

method, where every field is sorted according to their expert individual function value, as follows:

$$Sort\,fields = [field_1, field_2, ..., field_n] \tag{14}$$

where the expert individual of $field_1$ and $field_n$ has the worst and the best function values, respectively. After that, the changing probability for each field can be calculated as follows:

$$P_i = \frac{O_i}{N_{field} + 1} \tag{15}$$

where P_i and O_i are the field changing probability and the order for the i-th field to be sorted, respectively. With this method, the field in which the expert individual of the whole society belongs, has lower probability of being selected. On the other hand, the field that has a worse function value of its expert individual, will be more likely to be selected for the process of moving an individual of this field to another one. Then we generate a random number between 0 and 1, and it is used in the following expression:

$$if\,rand \leq P_i \rightarrow field\,changing\,occurs \tag{16}$$

If the expression is satisfied, an individual from this field will be transferred to a different one. Once we have a field to operate with, a selection probability for each individual will be defined as follows:

$$P.S_j = \left| \frac{f(Individual_j)}{\sum_{k=1}^{N_{ind}} f(Individual_k)} \right| \tag{17}$$

where $P.S_j$ is the selection probability for the j-th individual and N_{ind} is the number of individuals in the selected field. After that, by using the roulette wheel selection method [35], an individual will be selected and will change his field to another random different field. According to the use of this algorithm in [36], we modify this operator to prevent the fields from being unbalanced. Every certain number of iterations, the solutions are redistributed equally among the number of fields.

E. FINALIZATION

The stopping criteria selected in this work is the number of iterations reaches to maximum iterations.

V. SELF-ORGANIZING MAPS

Self-organizing map (SOM) was introduced by Teuvo Kohonen in the year 1980 [37]. It is an unsupervised neural network architecture for data analysis which generates a nonlinear mapping of data to lower dimensions. Because of the nature of the unsupervised network, it is a appropriate method for clustering problems and data exploration among others. In these networks, neurons acquire a knowledge in an unsupervised way in view of the fact that is not an objective output that the network has to come up with. Therefore, a pattern to let the network know does not exist. This last fact forces the network to come upon the frequent patterns in the number of inputs. The self-organizing map it is similar to the Vector Quantization, in which the space of input data (on vector form), also known as feature vectors, is divide into a finite number of contiguous regions, respectively each region is described optimally by a single model vector. In addition, in the self-organizing map, the models are spatially, globally ordered.

The SOM models are corresponding with the nodes of a regular, typically two-dimensional grid, Figure 2.

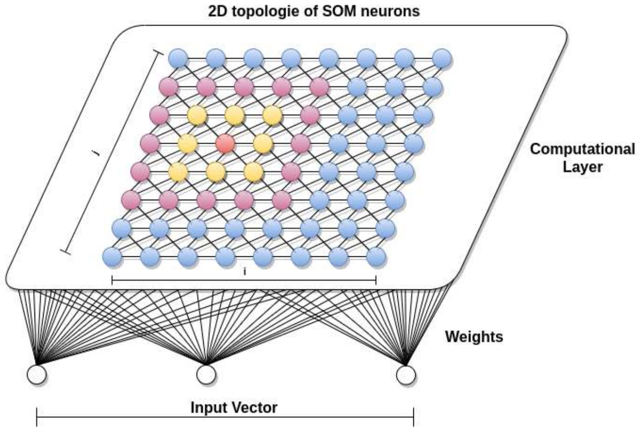


FIGURE 2: Two dimensional SOM

The SOM algorithm builds the models in such a way that the most similar ones will be aligned with nodes that are closer, together on the grid. While models that are are less similar will be positioned further apart in the grid.

To apprehend this idea, we can formulate in the following way: Every input data item must choose the model that best fits the input item, and this model, along with a subset of its spatial neighbors in the grid, must be changed to improve matching.

The modification is focus on a particular node that hold the winner model. As effect of this, the spatial neighborhood in the network around the winner is also modified at a time. The local ordering of the models in this neighborhood will raise. The subsequent modifications will change different models, leading to a modification on the entire network.

Figure 2 shows a two-dimensional topology of a self-organizing map, where the input layer is connected with all the neurons of the computational layer, using weights. The SOM Network's aim is to locate the neuron of the computational layer with the weights that are most similar to the values of the input layer. In order to do this, each neuron calculates the discrepancy between the pattern of the input and the set weights of each neuron of the computational layer. The winning neuron, also known as Best Matching Unit (BMU), is the one with the minimal distance between its weights and the set of inputs. To measure the distance be-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

tween the inputs and the SOM neurons, we can use Euclidean distance, correlation, direction cosine or block distance.

The Euclidean distance between the neurons of the computational layer and the input vector is calculated according to this equation:

$$d_{i,j,(t)} = \sqrt{\sum_{h=1}^{k}(W_{i,j,h} - X_k)^2} \tag{18}$$

where X_k is the input of the k-input neuron, $W_{i,j,h}$ is the weight for each neuron h and $d_{i,j,(t)}$ is the Euclidean distance of the (i, j) neuron in t relative to the input for a network with $i \times j$ neurons in the computational layer and k elements, in the input layer. The best matching unit is the one with the shortest Euclidean distance computed as:

$$g_{i,j} = Min(\forall d_{i,j}) \tag{19}$$

The best matching unit obtains an output equal to one, all other neurons receives an output of zero and then the weights of the winning neuron are calibrated with a learning function to become closer to the input vector. With this operation, the winning neuron has a higher probability to be matched when a new input vector similar to previous enter the network. On the other hand, if the new input vector is different, the chances of the winning neuron to be selected are reduced.

The equation that modifies the weights of the winning neuron and of the neighborhood function neurons is the following:

$$W_{jik}(t+1) = W_{jik}(t) + \alpha \cdot \lfloor X_k(t) - W_{jik}(t) \rfloor \tag{20}$$

where $X_k(t)$ is the input vector in t, $W_{jik}(t)$ is the weight that connects the k input with the (j, i) neuron in t and α is the learning rate

The neighborhood function makes it possible to change the weights of the best matching unit and of the nearest neurons to identify similar inputs. The radio of the proximate neurons decreases with the number of iterations of the model to reach a higher and better differentiation from every single neuron.

VI. INTEGRATION BETWEEN SOM AND HBBO

Optimization algorithms produce a large amount of information regarding to the characteristics of the solutions, as they advance in their iterations. This information, as common case, is not historically stored. If we can build a history of the optimization process; this can give us relevant knowledge about how the optimization algorithm has been progressing and we could also make decisions that modify its behavior during the run. For example increasing its intensity level either in an exploration or exploitation phase. One key aspect of this behavior controlling is deal with the problem of premature convergence, according to [14]. Premature convergence occurs when the population of an algorithm reaches a suboptimal state where the algorithm operators can no longer produce new solutions better than existing ones. In this case, we can say that the algorithm is trapped in a region that contains non-global optimum. An explanation for this phenomenon is the loss of diversity.

The integration between self-organizing maps and the optimization algorithm is based on the work presented in [38]. The main aim is that the history of the search process will be stored for allowing us to generate decisions that affect the behavior of the optimization algorithm. To achieve this purpose, we will use two tables that are generated from of the classification provided by the self-organizing map in regards of the generated solutions, storing the number of times that the best matching unit was activated by a particular solution. The first table will be called population distribution table. It stores the frequency of activation with respect to the solutions that are present in a given moment in the population. We will refer to this table by means of $E_p(i)$. The second table, search history table, will save the activation frequency with respect to all the solutions generated during the iterations of the optimization algorithm. We will refer to this table through $E_h(i)$. By analyzing the first table, we can account for the diversity of the population in a particular iteration. The higher the number of activated neurons, the greater it will be the diversity of the population. On the other hand in the search history table we will observe the behavior of the optimization algorithm in general. For example, if a neuron has been barely activated, we can affirm that the optimization algorithm has not sufficiently covered that area.

With the information contained in the two tables mentioned previously, we will proceed to incorporate three modifications on the optimization algorithm:

- The concept of novelty: Novelty refers to the number of similar solutions, according to the classification of the SOM, that were found during the running of HBBO. In other words, novelty is the quality of a solution, in terms of being compared with the search history table. In the sense that if we have found many solutions similar to the one we are evaluating, their degree of novelty will be low. On the contrary, if the search history table contains a low number of encounters of solutions similar to the one evaluated, the value of the novelty will be higher. We will use the following function to calculate the novelty metric.

$$novelty(f) = \frac{1}{E_h(b_s)} \tag{21}$$

where s represents a solution and b_s is the BMU on the trained Self-Organizing Map. In other words, novelty is the inverse of the activation frequency in the search history table. This generates a method by which the solutions can remain in the population, which depends on the contribution in the exploration of new search subspaces. At the first iterations of the optimization algorithm, the activation frequencies of highly exploited regions will be greater, and thus the novelty factor of the corresponding solutions will be low. And so, a

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

percentage of these solutions will disappear giving place to solutions with higher novelty.

- An operator for creating new solutions: We will introduce new solutions with a high novelty factor into the population. Such solutions correspond to the neurons with low activation count in the search history table. This incorporation is done by creating solutions that belongs to BMU with the lowest activation frequency in the search history table. To keep the population size constant, the solutions with the lowest novelty will be removed from the population. This operator replaces solutions of overcrowded areas to sparsely populated regions.
- Balance between exploration and exploitation: The level of exploration in the current state of the HBBO is defined by counting the number of different activated neurons in the population distribution table. A balance function is used to determine the number of neurons which ideally should be activated in the current iteration of the search process. This optimal number of activated units r is calculated by:

$$r = 1 - \frac{i}{i_{max}} \cdot \mid\mid U \mid\mid$$

where i is the number of iterations already performed, i_{max} is the maximum number of iterations and $\mid\mid U \mid\mid$ is the total number of neurons of the SOM. The reseed fraction equals the difference between the ideal and the measured neuron count.

A. PSEUDOCODE FOR HBBO-SOM

Algorithm 1 depicts the proposed procedure for the implementation of HBBO. Lines 1, 2 and 3, we set the initial parameters for the optimization algorithm; initial population are generated and spread the solutions among the fields. Regarding the loop between Lines 5 and 9, every solution except the first individual of each field, execute the education process. For loop between Lines 10 – 12, we apply the consultation procedure for each solution, except the best individual of all the population. Finally, from Line 13 to 19, we proceed to perform the procedure to move random individuals to another random field. Algorithm 2 shows the proposed implementation of SOM. The Line 1 execute the function that saves the activation frequencies of the population distribution table, and the search history table. The Line 2 executes the procedure that is responsible for checking if there is the necessity of incorporating new solutions and determining the number of these. The loop on Lines 3 to 5 replace a number of solutions defined in the step before, and the function on Line 4 will defines from which best matching unit the solutions are randomly created (from the neurons with the lowest count in the search history table) and which solutions with the lowest fitness functions will be deleted.

Algorithm 1 *HBBO*

```
1:  Set initial parameters
2:  Generate initial population
3:  Spread population among fields
4:  while ($i \leq MaximumIteration$) do
5:      for $i = 1 : n$ (n number of fields) do
6:          for $j = 2 : m$ (m number of individuals in $field_i$) do
7:              Education($individual_{ij}$)
8:          end for
9:      end for
10:     for $i = 1 : n$ (n number of individuals) do
11:         Consultation($individual_i$)
12:     end for
13:     for $i = 1 : n$ (n number of fields) do
14:         for $j = 1 : m$ (m number of individuals in $field_i$) do
15:             if check $individual_j$ of $field_i$ = True then
16:                 Move $individual_j$ to random field
17:             end if
18:         end for
19:     end for
20:     Call to Algorithm 2: Self Organizing Maps
21: end while
```

Algorithm 2 *SOM*

```
1: Save Activation Frequency Tables()
2: Check Reseeding()
3: for $i = 1 : n$ (n number of new solutions) do
4:     ReplaceSolutions()
5: end for
6: Reset Population Distribution Table()
```

VII. EXPERIMENTAL RESULTS

This work finds bijective S-Box with 8 input and output bits. The implementation of both algorithms has been done in C++. For the construction and analysis of the proposed substitution box we have used the library presented by [39], which is a collection of C++ classes designed for analyzing vector boolean functions. The construction and training of the self-organizing map were carried out using the library [40]. The experiments were launched on a dual Intel Xeon E5-2690 with 32GB RAM running in Debian 10. The values of the parameters of the optimization algorithm were the following: $k1 = 0.8$, $k2 = 1.2$, $cf = 0.2$, $individuals = 150$, $iterations = 5000$. The parameters of the SOM were the following: $Rows = Columns = 10$, $epochs = 2000$, $samples = 10000$. Source code can be found on [99]

In Figure 3, we can see the number of the new solutions incorporated through and experiments with 1000 iterations and 90 solutions as an initial population quantity.

In Figures 4, 5 and 6, the color intensity refers to the number of times that a particular neuron has been activated (this information is store in the search history table). The lighter the green color, the smaller the number of solutions that have been associated with a particular neuron, allowing us to see in a certain way which regions have been explored to a greater or lesser extent. As we advance in the iterations of the optimization algorithm, the self-organizing map helps

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

Method	Min NL	Max NL	ACNV
[41] [42]	84	106	100.0
[43]	98	108	102.3
[44]	96	106	102.5
[45]	100	106	103.0
[46]	96	106	103.0
[47]	98	108	103.0
[48]	98	108	103.2
[49]	100	106	103.2
[50]	99	106	103.3
[51]	96	108	103.5
[52]	101	108	103.8
[53]	101	106	103.8
[54]	102	106	104.0
[55]	98	108	104.0
[56]	100	106	104.0
[57]	102	106	104.0
[58]	98	108	104.0
[59]	102	108	104.5
[60] [61]	100	108	104.7
[62] [63]	102	108	104.7
[64]	100	108	104.75
[65]	100	107	104.8
[66]	104	106	105.0
[67]	102	108	105.2
[68]	102	108	105.3
[69]	100	110	105.5
[70]	98	110	105.5
[71]	102	110	105.5
[72]	104	108	105.7
[73]	102	108	106.0
[74] [75] [76]a	104	110	106.0
[76]c	106	108	106.0
[77]	104	110	106.2
[78]	104	110	106.5
[79]	106	108	106.5
This Work	**102**	**110**	**106.5**
[80] [81]	106	108	106.7
[82]	104	108	106.7
[6]	106	110	107.0
[76]b	104	108	107.0
[83]	106	108	107.5
[84]	106	110	107.75
[8]	108	108	108.0
[85]	104	110	108.0
[86]	106	110	108.5
[87]	108	112	109.0
[88] [89] [90]	112	112	112.0
[91]b	112	112	112.0
[91]a	114	114	114.0
[91]c	114	116	114.5

TABLE 1: Experimental Results

us to incorporate solutions belonging to regions of the search space little explored, this can be observed taking into account the homogenization of the color in the figures as we move forward in iterations.

As a counterexample, Figure 6 shows the situation where we do not incorporate the solution replacement operator. In this image, we can observe a certain imbalance between regions. For example, the region in the row 7 column 8 has a value of the activation frequency of 2049, in contrast to the regions in row 5, columns 2 and 3, with an activation frequency value of 71 and 152 respectively.

A. NONLINEARITY

The results shown in Table 2 consider the average nonlinearity of the S-box coordinates only [91]. The works are based on optimization algorithms and chaotic approaches.

B. BIJECTIVITY

For an $n \times n$ S-box, the bijectivity is confirmed if it satisfies the equation:

$$wt\left(\sum_{i=1}^{n} a_i f_i\right) = 2^{n-1} \tag{22}$$

where $a_i \in \{0,1\}, (a_1, a_2, ..., a_n) \neq (0, 0, ..., 0)$ and wt is the Hamming weight. All S-boxes found satisfies the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

164	177	91	2	126	0	63	158	66	54	146	150	44	51	157	7
245	236	212	210	15	254	207	4	110	246	186	183	127	68	93	125
111	208	45	33	225	65	106	82	34	214	182	86	153	56	116	129
219	199	151	248	39	78	209	109	175	62	185	57	249	228	196	163
40	155	250	59	55	61	17	96	223	252	31	140	201	11	103	218
23	131	174	35	198	159	60	202	149	187	255	122	16	80	20	72
234	213	70	76	12	152	137	195	160	100	168	192	115	220	8	191
27	148	138	88	244	243	58	14	77	197	216	30	64	117	97	123
222	231	83	25	10	230	130	184	29	107	161	22	200	49	145	189
5	206	73	92	147	101	221	171	28	124	113	48	118	239	128	233
38	238	227	3	170	135	67	71	203	114	190	32	9	224	21	193
141	217	105	53	226	253	134	229	194	50	99	1	102	36	204	179
144	173	172	47	181	169	142	42	242	89	19	75	85	26	133	95
215	108	24	104	143	121	237	18	178	79	136	205	235	69	132	176
211	251	167	98	94	43	241	120	13	6	247	84	156	112	46	87
90	37	41	154	162	81	232	240	188	74	165	139	52	180	166	119

TABLE 2: Proposed SOM-HBBO based S-Box

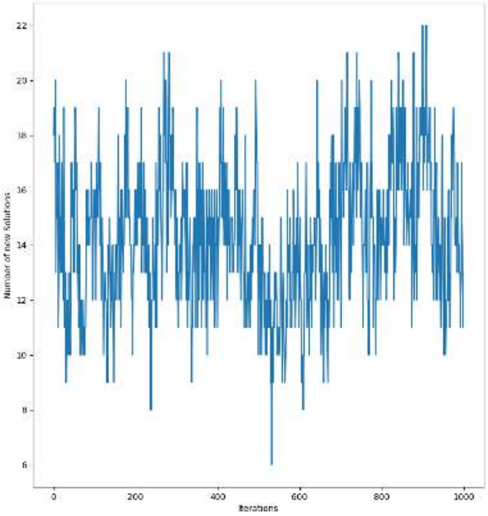


FIGURE 3: Incorporation of new Solutions

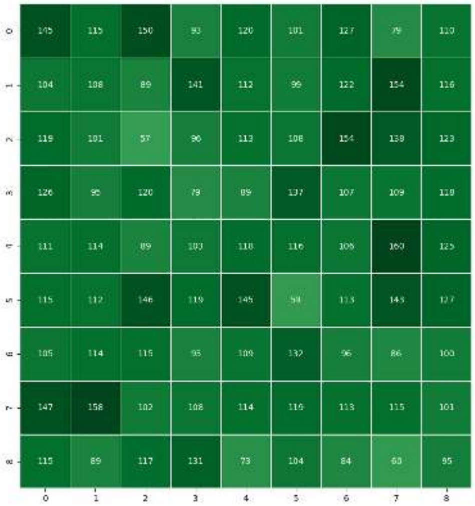


FIGURE 4: HeatMap of diversity 100 Iterations

equation stated before.

C. ALGEBRAIC DEGREE

The proposed S-box has a algebraic degree of $deg(f) = 6$, which is a good indicator of resistance to higher-order differential attacks, algebraic attacks or cube attack [92] and [93].

Algebraic attacks, have been introduced [94] and [95]. They recover the secret key, or at least the initialization of the system, by solving a system of multivariate algebraic equations. In order to identify a cryptographic algorithm's immunity to this kind of attacks, the criterion of algebraic immunity was established. The proposed S-box, has value of algebraic immunity equals to 4 which is the maximum value for a substitution box with these dimensions.

D. STRICT AVALANCHE CRITERIA

Strict avalanche criterion was proposed by Webster and Tavares [96]. If a boolean function satisfy Strict Avalanche Criteria, it means the output bit would change with a probability of half whenever a single input bit is changed. This can be screened by determining dependency matrix of S-box under examination. The dependency matrix for one of the substitution boxes founded is provided in Table 3. The SAC of the proposed S-box is 0.4943 which is having a negligible deviation of 0.0056 from the ideal value 0.5.

0.5000	0.4219	0.5000	0.5156	0.5312	0.5312	0.4688	0.4844
0.4844	0.4531	0.4531	0.4531	0.4844	0.4844	0.5000	0.5312
0.5000	0.5312	0.4844	0.4375	0.5469	0.4531	0.5156	0.5781
0.4688	0.5781	0.5469	0.4375	0.5156	0.4844	0.5625	0.4844
0.4844	0.4688	0.4062	0.5156	0.5312	0.4531	0.5312	0.5000
0.5000	0.5000	0.4688	0.5000	0.4219	0.4688	0.5156	0.4062
0.5000	0.4531	0.5156	0.5312	0.5312	0.5469	0.5312	0.5312
0.4375	0.5000	0.5156	0.4844	0.5000	0.4531	0.4688	0.5469

TABLE 3: Dependency matrix for avalanche effect

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

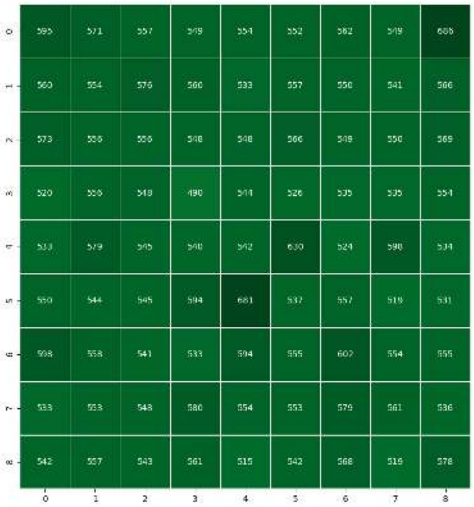


FIGURE 5: HeatMap of diversity 500 Iterations

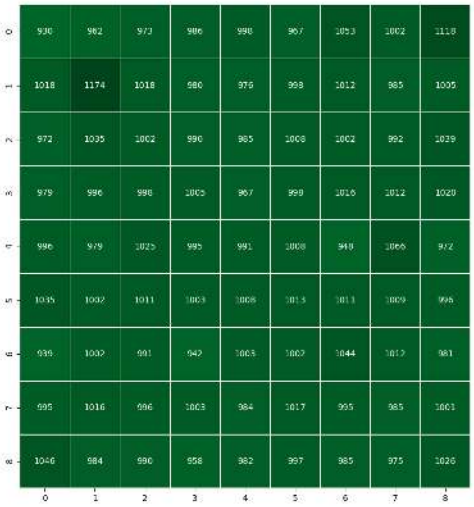


FIGURE 6: HeatMap of diversity 900 Iterations

E. BIT INDEPENDENT CRITERION(BIC)

BIC property can be explained as all the avalanche variables should be pair-wise independent for a given set of avalanche vectors, generated by the complementing of a single plaintext bit. Results are shown in table 4.

F. DIFERENTIAL UNIFORMITY

An S-box having differential uniformity uniquely maps an input differential Δx to an output differential Δy. An S-box is said to be immune to differential cryptanalysis, if it has as low value of maximum differential approximation probability as possible. The differential approximation probability (DAP) measures differential uniformity and is mathematically defined as:

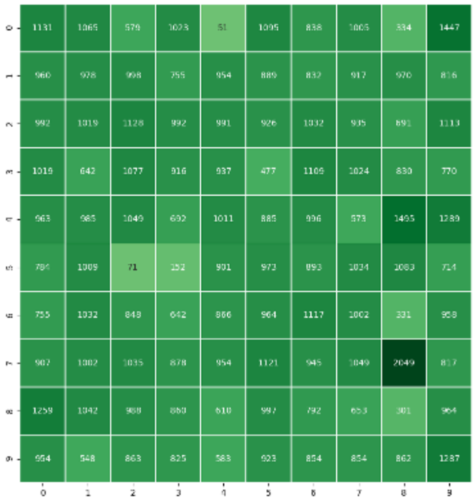


FIGURE 7: HeatMap with no replacement operator 900 Iterations

—	106	100	106	102	104	102	106
106	—	100	106	106	102	104	104
100	100	—	106	104	102	96	106
106	106	106	—	106	104	100	106
102	106	104	106	—	104	106	90
104	102	102	104	104	—	102	108
102	104	96	100	106	102	—	106
106	104	106	106	90	108	106	—

TABLE 4: Bit independent criterion

$$DAP(\Delta x \to \Delta y) = \frac{\#\{x \in X | S(x) \oplus S(x \oplus \Delta x) = y\}}{2^m}$$

where X is the set of input values and 2^m is the total number of elements. The maximum DAP for proposed S-box is 0.0468. Table 5 shows a comparative results with another works.

G. LINEAR APPROXIMATION PROBABILITY

The maximum value of the imbalance of an event is called Linear approximation probability LP. The parity of the input bits selected by the mask Γx is equal to the parity of the output bits selected by the mask Γy. The definition of LP according to [4] of a given S-box is as follows:

$$LP = \max_{\Gamma x, \Gamma y \neq 0} \left| \frac{\#\{x \in X \mid x \cdot \Gamma x = S(x) \cdot \Gamma y\}}{2^n} - \frac{1}{2} \right| \quad (23)$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

S-Box	DAP
This work	0.0468
[6]	0.0391
[48]	0.0468
[45]	0.0468
[51]	0.0391
[52]	0.0546
[49]	0.0391
[64]	0.0468
[22]	0.0468
[97]	0.0468

TABLE 5: Comparison of max differential probability of some S-boxes

where Γx and Γy are the input and output value respectively, where X is the set of all posible inputs; and 2^n is the number of elements. The maximum value of *LP* for our proposed S-box is 0.1484375.

VIII. CONCLUSION AND FUTURE WORK

The substitution box is one of the most important components in symmetric cryptographic systems such as Data Encryption Standard, and Advanced Encryption Standard. An S-Box provides the characteristic of hiding the relationship between secret key and cipher text. This is the only nonlinear component that determines the strength of entire block encryption algorithm. Therefore, the main problem in the design of strong cryptosystem is the construction of a secure S-box.

In this work we have successfully combined a metaheuristic based on human behavior with Self-Organizing Maps, that avoid the problem of premature convergence. This algorithm was used to maximize the property of nonlinearity of the S-box in combination with the concept of novelty of the solutions. Novelty refers to the number of times that the Self-Organizing Map has found similar solutions during the execution of the optimization algorithm.

The S-boxes founded in this work are competitive with the state of the art results presented in several publications. It is worth mentioning that not only the quality but also the quantity of good S-boxes is relevant, as those found with this scheme.

The results could be further improved incorporating an initial generation of solutions that inherently possess a higher degree of diversity in relation to the hamming distance, for example, using techniques such as clustering. Another technique that could be used to increase the efficiency of the algorithm is to use a machine learning method such as SVM to obtain a model that can be used as a substitute for the objective function and allows us to calculate the nonlinearity of an S-box in less time than the calculation used in the original version. Another approach that we could take is the incorporation of an algorithm that allows us to find the optimal values of the parameters of the algorithms used, either the optimization algorithm as from the Self-Organizing Map. This could be done for example with Autonomous Search.

ACKNOWLEDGMENT

I want to thank professors Ricardo Soto, Broderick Crawford and Rodrigo Olivares, for their support and guidance in this work. Francisco González was supported by Postgraduate Grant Pontificia Universidad Católica de Valparaso, Chile, 2021. Broderick Crawford is supported by Grant CONICYT/FONDECYT/REGULAR/1210810. Ricardo Soto is supported by Grant CONICYT/FONDECYT/REGULAR/1190129.

REFERENCES

[1] C. E. . Communication theory of secrecy systems. The Bell System Technical Journal, 28(4):656–715, 1949.
[2] S. Picek, M. Cupic, and L. Rotim. A new cost function for evolution of s-boxes. Evolutionary Computation, 24(4):695–718, 2016.
[3] Biham E., Shamir A., and Cryptol J. Differential cryptanalysis of des. like cryptosystems, 4(1):3, 1991.
[4] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In Advances in Cryptology — EUROCRYPT '93, pages 386–397, 1994.
[5] Lambić Dragan and Živković Miodrag. Comparison of random s-box generation methods. Publications de l'Institut Mathematique, 93:109 – 115, 2013.
[6] M. Ahmad, D. Bhatia, and Y. Hassan. A novel ant colony optimization based scheme for substitution box design. Procedia Comput. Sci., 57:572–580, 2015.
[7] William Millan, Andrew Clark, and Ed Dawson. An effective genetic algorithm for finding highly nonlinear boolean functions. In Information and Communications Security, pages 149–158. Springer Berlin Heidelberg, 1997.
[8] K.-W. Wang, Y.and Wong, C. Li, and Y. Li. A novel method to design sbox based on chaotic map and genetic algorithm. Phys. Lett. A, 376:827–833, Jan 2012.
[9] John Clark, Jeremy Jacob, and Susan Stepney. The design of s-boxes by simulated annealing. New Generation Computing, 23:219–231, 07 2004.
[10] John A. Clark, Jeremy L. Jacob, and Susan Stepney. The design of s-boxes by simulated annealing. New Generation Computing, 23(3):219–231, Sep 2005.
[11] Antonio Bolufé-Röhler and Dania Tamayo-Vera. Machine learning based metaheuristic hybrids for s-box optimization. Journal of Ambient Intelligence and Humanized Computing, 11(11):5139–5152, Nov 2020.
[12] M. Ahmad, I. A. Khaja, A. Baz, H. Alhakami, and W. Alhakami. Particle swarm optimization based highly nonlinear substitution-boxes generation for security applications. IEEE Access, 8:116132–116147, 2020.
[13] Seyed-Alireza Ahmadi. Human behavior-based optimization: a novel metaheuristic approach to solve complex optimization problems. Neural Computing and Applications, 28(1):233–244, Dec 2017.
[14] D. B. Fogel. An introduction to simulated evolutionary optimization. IEEE Transactions on Neural Networks, 5(1):3–14, 1994.
[15] Amjad Hussain Zahid and Muhammad Junaid Arshad. An innovative design of substitution-boxes using cubic polynomial mapping. Symmetry, 11(3), 2019.
[16] Amjad Hussain Zahid, Muhammad Junaid Arshad, and Musheer Ahmad. A novel construction of efficient substitution-boxes using cubic fractional transformation. Entropy, 21(3), 2019.
[17] William Millan, Joanne Fuller, and Ed Dawson. New concepts in evolutionary search for boolean functions in cryptology. Computational Intelligence, 20(3):463–474, 2004.
[18] John A. Clark and Jeremy L. Jacob. Two-stage optimisation in the design of boolean functions. In Information Security and Privacy, pages 242–254, 2000.
[19] Petr Tesa. A new method for generating high non-linearity s-boxes. 2010.
[20] M. Yang, Z. Wang, Q. Meng, and L. Han. Evolutionary design of s-box with cryptographic properties. In 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops, pages 12–15, 2011.
[21] O. Kazymyrov, V. Kazymyrova, and R. Oliynykov. A method for generation of high-nonlinear s-boxes based on gradient descent. IACR Cryptol. ePrint Arch., 2013:578, 2013.
[22] Musheer Ahmad, Parvez Mahmood Khan, and Mohd Zeeshan Ansari. A simple and efficient key-dependent s-box design using fisher-yates shuffle

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

technique. In Gregorio Martínez Pérez, Sabu M. Thampi, Ryan Ko, and Lei Shu, editors, Recent Trends in Computer Networks and Distributed Systems Security, pages 540–550, 2014.

[23] Majid Khan, Tariq Shah, and Syeda Iram Batool. Construction of s-box based on chaotic boolean functions and its application in image encryption. Neural Computing and Applications, 27(3):677–685, Apr 2016.

[24] Sajjad Shaukat Jamal, Muhammad Usman Khan, and Tariq Shah. A watermarking technique with chaotic fractional s-box transformation. Wireless Personal Communications, 90(4):2033–2049, Oct 2016.

[25] Majid Khan and Zeeshan Asghar. A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and s8 permutation. Neural Computing and Applications, 29(4):993–999, Feb 2018.

[26] Georgi Ivanov, Nikolay Nikolov, and Svetla Nikova. Cryptographically strong s-boxes generated by modified immune algorithm. volume 9540, pages 31–42, 01 2016.

[27] Ye Tian and Zhimao Lu. Chaotic s-box: Intertwining logistic map and bacterial foraging optimization. Mathematical Problems in Engineering, 2017:6969312, Nov 2017.

[28] T. Akhtar, N. Din, and J. Uddin. Substitution box design based on chaotic maps and cuckoo search algorithm. In 2019 International Conference on Advanced Communication Technologies and Networking (CommNet), pages 1–7, 2019.

[29] Qing Lu, Congxu Zhu, and Xiaoheng Deng. An efficient image encryption scheme based on the lss chaotic map and single s-box. IEEE Access, 8:25664–25678, 2020.

[30] Hussam S. Alhadawi, Mazlina Abdul Majid, Dragan Lambić, and Musheer Ahmad. A novel method of s-box design based on discrete chaotic maps and cuckoo search algorithm. Multimedia Tools and Applications, 80(5):7333–7350, Feb 2021.

[31] Badr M. Alshammari, Ramzi Guesmi, Tawfik Guesmi, Haitham Alsaif, and Ahmed Alzamil. Implementing a symmetric lightweight cryptosystem in highly constrained iot devices by using a chaotic s-box. Symmetry, 13(1), 2021.

[32] Amjad Hussain Zahid, Eesa Al-Solami, and Musheer Ahmad. A novel modular approach based substitution-box design for image encryption. IEEE Access, 8:150326–150340, 2020.

[33] C. Carlet, D. K. Dalai, K. C. Gupta, and S. Maitra. Algebraic immunity for cryptographically significant boolean functions: analysis and construction. IEEE Transactions on Information Theory, 52(7):3105–3121, 2006.

[34] Francisco Rodríguez-Henríquez, Nazar Abbas Saqib, Arturo Díaz-Pérez, and Cetin Koc. Cryptographic Algorithms on Reconfigurable Hardware. 01 2007.

[35] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. 1989.

[36] R. Soto, B. Crawford, F. González, E. Vega, C. Castro, and F. Paredes. Solving the manufacturing cell design problem using human behavior-based algorithm supported by autonomous search. IEEE Access, 7:132228–132239, 2019.

[37] T. Kohonen. The self-organizing map. Proceedings of the IEEE, 78(9):1464–1480, 1990.

[38] Heni Ben Amor and Achim Rettinger. Intelligent exploration for genetic algorithms: Using self-organizing maps in evolutionary computation. GECCO 2005 - Genetic and Evolutionary Computation Conference, pages 1531–1538, December 2005.

[39] José Antonio Álvarez Cubero and Pedro J. Zufiria. Algorithm 959: Vbf: A library of c++ classes for vector boolean functions in cryptography. ACM Trans. Math. Softw., 42(2), May 2016.

[40] S. Habdank-Wojewódzki and R. Rybarski. The kohonen neural network library. Overload, 74, Aug 2006.

[41] M. Khan. A novel image encryption scheme based on multiple chaotic s-boxes. Nonlinear Dynamics, 82(1):527–533, Oct 2015.

[42] M. Khan, T. Shah, and S.I. Batool. Construction of s-box based on chaotic boolean functions and its application in image encryption. Neural Comput. Appl., 27(3):677–685, Apr 2016.

[43] S. S. Jamal, M. U. Khan, and T. Shah. A watermarking technique with chaotic fractional s-box transformation. Wireless Pers. Commun., 90(4):2033–2049, Oct 2016.

[44] M. Khan and Z. Asghar. A novel construction of substitution box for image encryption applications with gingerbreadman chaotic map and s8 permutation. Neural Comput. Appl., 29(4):993–999, 2018.

[45] G. Chen, Y. Chen, and X. Liao. An extended method for obtaining sboxes based on three-dimensional chaotic baker maps. Chaos, Solitons, Fractals, 31(3):571–579, Feb 2007.

[46] M. Khan, T. Shah, H. Mahmood, M. A. Gondal, and I. Hussain. A novel technique for the construction of strong s-boxes based on chaotic lorenz systems. Nonlinear Dyn, 70(3):2303–2311, Nov 2012.

[47] M. Khan, T. Shah, H. Mahmood, and M. A. Gondal. An efficient method for the construction of block cipher with multi-chaotic systems. Nonlinear Dyn, 71(3):489–492, Feb 2013.

[48] G. Jakimoski and L. Kocarev. Chaos and cryptography: Block encryption ciphers based on chaotic maps. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., 48(2):163–169, 2001.

[49] F. Özkaynak and A. B. Özer. A method for designing strong sboxes based on chaotic lorenz system. Phys. Lett. A, 374(36):3733–3738, Aug 2010.

[50] G. Tang, X. Liao, and Y. Chen. A novel method for designing sboxes based on chaotic maps. Chaos, Solitons, Fractals, 23(2):413–419, Jan 2005.

[51] M. Asim and V. Jeoti. Efficient and simple method for designing chaotic s-boxes. ETRI J., 30(1):170–172, Feb 2008.

[52] G. Tang and X. Liao. A method for designing dynamical s-boxes based on discretized chaotic map. Chaos, Solitons, Fractals, 23(5):1901–1909, Mar 2005.

[53] F. Özkaynak and S. Yavuz. Designing chaotic s-boxes based on timedelay chaotic system. Nonlinear Dyn, 74(3):551–557, Nov 2013.

[54] G. Chen. A novel heuristic method for obtaining s-boxes. Chaos, Solitons, Fractals, 36(4):1028–1036, May 2008.

[55] M. Khan, T. Shah, and M. A. Gondal. An efficient technique for the construction of substitution box with chaotic partial differential equation. Nonlinear Dyn, 73(3):1795–1801, Aug 2013.

[56] M. Khan and T. Shah. A construction of novel chaos base nonlinear component of block cipher. Nonlinear Dyn, 76(1):377–382, Apr 2014.

[57] H. Liu, A. Kadir, and Y. Niu. Chaos-based color image block encryption scheme using s-box. AEU-Int. J. Electron. Commun., 68(7):676–686, Jul 2014.

[58] M. Khan, T. Shah, and Batool S.I. A new implementation of chaotic s-boxes in captcha. Signal, Image Video Process, 10(2):293–300, Feb 2016.

[59] L. Liu, Y. Zhang, and X. Wang. A novel method for constructing the sbox based on spatiotemporal chaotic dynamics. Appl. Sci., 8(12):2650, Dec 2018.

[60] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, and H. Mahmood. A group theoretic approach to construct cryptographically strong substitution boxes. Neural Comput. Appl., 23(1):97–103, Jul 2013.

[61] F. Özkaynak, V. Çelik, and A. B. Özer. A new s-box construction method based on the fractional-order chaotic chen system. Signal, Image Video Process., 11(4):659–664, May 2017.

[62] I. Hussain, T. Shah, and M. A. Gondal. A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm. Nonlinear Dyn, 70(3):1791–1794, Nov 2012.

[63] F. Özkaynak. An analysis and generation toolbox for chaotic substitution boxes: A case study based on chaotic labyrinth rene thomas system. Iranian J. Sci. Technol., Trans. Elect. Eng., 44:89–98, Jun 2019.

[64] M. Khan and T. Shah. An efficient construction of substitution box with fractional chaotic system. Signal, Image Video Process, 9(6):1335–1338, Sep 2015.

[65] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal. Construction of s8 liu j s-boxes and their applications. Comput. Math. Appl., 64(8):2450–2458, Oct 2012.

[66] Özkaynak, F. From biometric data to cryptographic primitives: A new method for generation of substitution boxes. Proc. Int. Conf. Biomed. Eng. Bioinf. 27–33,volume 2017.

[67] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood. A novel method for designing nonlinear component for block cipher based on td-ercs chaotic sequence. Nonlinear Dyn, 73(1-2):633–637, Jul 2013.

[68] M. Belazi, A.and Khan and S. El-Latif A. A. A., Belghith. Efficient cryptosystem approaches: S-boxes and permutation substitution-based encryption. Nonlinear Dyn, 87(1):337–361, 2017.

[69] I. Hussain, T. Shah, H. Mahmood, and M. A. Gondal. A projective general linear group based algorithm for the construction of substitution box for block ciphers. Neural Comput. Appl., 22(6):1085–1093, May 2013.

[70] M. Khan and T. Shah. A novel image encryption technique based on hénon chaotic map and s8 symmetric group. Neural Comput. Appl., 25(7-8):1717–1722, 2014.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

[71] A. Belazi and El-Latif A.A.A. A simple yet efficient s-box method based on chaotic sine map. Optik, 130:1438–1444, Feb 2017.

[72] G. Liu, W. Yang, W. Liu, and Y. Dai. Designing s-boxes based on 3-d four-wing autonomous chaotic system. Nonlinear Dyn, 82(4):1867–1877, Dec 2015.

[73] F. U. Islam and G. Liu. Designing sbox based on 4d-4wing hyperchaotic system. 3D Res., 8(1), Mar 2017.

[74] Ünal Çavuşoğlu, Sezgin Kaçar, Ahmet Zengin, and Ihsan Pehlivan. A novel hybrid encryption algorithm based on chaos and s-aes algorithm. Nonlinear Dynamics, 92(4):1745–1759, Jun 2018.

[75] X. Wang, A. Akgul, V.-T. Pham, D. Hoang, and X. Nguyen. A chaotic system with in nite equilibria and its s-box constructing application. Appl. Sci., 8(11):2132, Nov 2018.

[76] Xiong Wang, Ünal Çavuşoğlu, Sezgin Kacar, Akif Akgul, Viet-Thanh Pham, Sajad Jafari, Fawaz Alsaadi, and Xuan Nguyen. S-box based image encryption application using a chaotic system without equilibrium. Applied Sciences, 9(4):781, Feb 2019.

[77] A. Zengin, I. Pehlivan, and S. Kaçar. A novel approach for strong s-box generation algorithm design based on chaotic scaled zhongtang system. Nonlinear Dyn, 87(2):1081–1094, Jan 2017.

[78] T. Farah, R. Rhouma, and S. Belghith. A novel method for designing s-box based on chaotic map and teaching learning-based optimization. Nonlinear Dyn, 88(2):1059–1074, 2017.

[79] D. Lambi. S-box design method based on improved one-dimensional discrete chaotic map. J. Inf. Telecommun., 2(2):181–191, Apr 2018.

[80] F. Özkaynak. Construction of robust substitution boxes based on chaotic systems. Neural Comput. Appl., 31(8):3317–3326, Aug 2019.

[81] D. Lambi. A novel method of s-box design based on discrete chaotic map. Nonlinear Dyn, 87(4):2407–2413, Mar 2017.

[82] T. Ye and L. Zhimao. Chaotic s-box: Six-dimensional fractional lorenz duffing chaotic system and o-shaped path scrambling. Nonlinear Dyn, 94(3):2115–2126, 2018.

[83] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad. A novel efficient substitution-box design based on fire fly algorithm and discrete chaotic map. Neural Comput. Appl., 31(11):7201–7210, Nov 2019.

[84] L. Yi, X. Tong, Z. Wang, M. Zhang, H. Zhu, and J. Liu. A novel block encryption algorithm based on chaotic s-box for wireless sensor network. IEEE Access, 7:53079–53090, 2019.

[85] X. Zhang, Z. Zhao, and J. Wang. Chaotic image encryption based on circular substitution box and key stream buffer. Signal Process., Image Commun., 29(8):902–913, Sep 2014.

[86] E. Al Solami, M. Ahmad, C. Volos, M. Doja, and M. Beg. A new hyperchaotic system-based design for efficient bijective substitution-boxes. Entropy, 20(7):525, Jul 2018.

[87] D. Lambi. A novel method of s-box design based on chaotic map and composition method. Chaos, Solitons, Fractals, 58:16–21, Jan 2014.

[88] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood. An efficient approach for the construction of lft s-boxes using chaotic logistic map. Nonlinear Dyn, 71(1-2):133–140, Jan 2013.

[89] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood. Eficient method for designing chaotic s-boxes based on generalized baker's map and tderc chaotic sequence. Nonlinear Dyn, 74(1-2):271–275, 2013.

[90] A. Belazi, A. A. Abd El-Latif, A.-V. Diaconu, R. Rhouma, and S. Belghith. Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. Opt. Lasers Eng., 88:37–50, Jan 2017.

[91] M. M. Dimitrov. On the design of chaos-based s-boxes. IEEE Access, 8:117173–117181, 2020.

[92] C. Boura and A. Canteaut. On the influence of the algebraic degree of f^{-1} on the algebraic degree of $g \circ f$. IEEE Transactions on Information Theory, 59(1):691–702, 2013.

[93] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of keccak and luffa. In Fast Software Encryption, pages 252–269, 2011.

[94] Nicolas T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, Advances in Cryptology - CRYPTO 2003, pages 176–194, 2003.

[95] Nicolas T. Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings, volume 2656 of Lecture Notes in Computer Science, pages 345–359, 2003.

[96] A. F. Webster and S. E. Tavares. On the design of s-boxes. In Advances in Cryptology — CRYPTO '85 Proceedings, pages 523–534, 1986.

[97] Muhammad Asif Gondal, Abdul Raheem, and Iqtadar Hussain. A scheme for obtaining secure s-boxes based on chaotic baker's map. 3D Research, 5(3):17, Aug 2014.

[98] Yasser Gonzalez-Fernandez and Stephen Chen. Leaders and followers – a new metaheuristic to avoid the bias of accumulated information. 05 2015.

[99] Francisco Gonzalez Molina. Hbbo som sbox source code, 2021. 10.6084/M9.FIGSHARE.14600232

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087139, IEEE Access

FRANCISCO GONZÁLEZ MOLINA is a Software Developer at Pareti S.A, Valparaíso, Chile. He received his Master degree in Informatics Engineering from the Pontificia Universidad Católica of Valparaíso, Chile, in 2019. Currently, he is a doctoral student in Computer Science Engineering at Pontificia Universidad Católica of Valparaíso, Chile.He has been working in Software Develoment for more than ten years and including other aspects as Server Managment and Pentesting tasks. Mainly researching in Metaheuristics.

RICARDO SOTO is a Full Professor and Head of the Computer Science Department at the Pontifical Catholic University of Valparaíso, Chile. He received his PhD degree in Computer Science from the University of Nantes, France, in 2009. His areas of research interest include mainly Metaheuristics, Global Optimization, and Autonomous Search. In this context, He has published about 180 scientific papers in different international conferences and journals, some of them top ranked in Computer Science, Operational Research, and Artificial Intelligence. Most of these papers are based on the resolution of real-world and academic optimization problems related to industry, manufacturing, rostering and seaports.

BRODERICK CRAWFORD is a Full Professor of the Computer Science Department at the Pontifical Catholic University of Valparaíso, Chile. He received his PhD degree in Computer Science from the Universidad Técnica Federico Santa María of Valparaíso, Chile, in 2011. His areas of research interest include mainly Combinatorial Optimization, Metaheuristics, Global Optimization, and Autonomous Search. In this context, He has published about +300 scientific papers in different international conferences and journals, some of them top ranked in Computer Science, Operational Research, and Artificial Intelligence. Most of these papers are based on the resolution of benchmark and real-world optimization problems.

RODRIGO OLIVARES received the Ph.D. degree in informatics engineering from Pontificia Universidad Católica de Valparaíso in 2019. He received the M.Sc. degree at Pontificia Universidad Católica de Valparaíso in 2015. He is currently Assistant Professor with the School of Informatics Engineering at Universidad de Valparaíso. His areas of research interest include mainly Reactive and Self-Adaptive Metaheuristics, Global Optimization, and Machine Learning. Professor Olivares has been author of several contributions in relevant scientific journals and prestigious conferences about optimization, artificial intelligence, and swarm intelligence algorithms.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/