GENERATION OF CRYPTOGRAPHICALLY STRONG KEY-DEPENDENT 8-BIT S-BOXES

Alejandro Freyre, Adrián Alfonso, Reynier A. de la Cruz and Pablo Freyre*
Institute of Cryptography, University of Havana, Cuba.

ABSTRACT
A widely used strategy in the design of block ciphers is the generation of key-dependent substitution boxes (S-Boxes), *i.e.*, secret S-Boxes that depend randomly on the cipher key, guaranteeing security due to the uncertainty provided by the randomness of the S-Box during the encryption process and not due to its cryptographic properties. On the other hand, in the specialized literature there are several methods to construct cryptographically strong S-Boxes to be used in block ciphers, however, the S-Boxes built by such methods are fixed and need to be generated beforehand. Inspired by finding key-dependent S-Boxes with good cryptographic properties, we present in this paper a new strategy to generate key-dependent and cryptographically strong 8-bit S-Boxes from 4-bit permutations and heuristic search.

KEYWORDS: 8-bit S-Boxes, 4-bit permutations, key-dependency, heuristic search.
MSC: 11T71, 90C27, 90C59, 94A60.

RESUMEN
Una estrategia ampliamente utilizada en el diseño de cifradores de bloque es la generación de cajas de sustitución (S-cajas) dependientes de la llave, *i.e.*, S-cajas secretas que se generan aleatoriamente dependiendo de la llave de cifrado, garantizando seguridad debido a la incertidumbre que provee la aleatoriedad de la S-caja durante el proceso de cifrado y no por sus propiedades criptográficas. Por otra parte en la literatura especializada existen varios métodos para construir S-cajas fuertes criptográficamente, sin embargo, las S-cajas construidas por estos métodos son fijas y necesitan ser generadas a priori. Inspirados por encontrar S-cajas dependientes de la llave con buenas propiedades criptográficas presentamos en este trabajo una nueva estrategia para generar S-cajas de 8 bits que dependen de la llave y son criptográficamente fuertes a partir de permutaciones de 4 bits y búsqueda heurística.

PALABRAS CLAVE: S-cajas de 8 bits, permutaciones de 4 bits, dependencia de la llave, búsqueda heurística.

1. INTRODUCTION

Key-dependent S-Boxes have been used in the design of block ciphers since the very beginnings of modern cryptography. In the early 70s, the block cipher Lucifer [42] used key-controlled S-Boxes to

[*] pfreyre@matcom.uh.cu

provide 16 bits of extra security. Later, other block ciphers like Khufu [40] in 1990 and Twofish [6] in 1998 were presented by its designers with key-dependent S-Boxes. Besides, dynamic modifications of known block ciphers with key-dependent S-Boxes are the variant of the soviet standard Magma presented in [45], the variants of Serpent [3] presented in [24, 33] and the variants of the advanced encryption standard (AES) [14] presented in [7, 34, 43].

Of course, the previous are not the only examples that can be cited, in [8, 30] the authors present a survey of dynamic block ciphers with key-dependent S-Boxes. Moreover, in the course of 2021 several new methods to generate key-dependent S-Boxes have been proposed in the specialized literature [17, 25, 26]. Unfortunately, as the security of this approach lies in the uncertainty provided by the randomness of the S-Box, not always key-dependent S-Boxes are cryptographically good S-Boxes, reason why many researchers prefer to use fixed (not secret) S-Boxes having good cryptographic parameters rather than key-dependent ones.

With regards to the generation of fixed S-boxes with good cryptographic parameters, the literature survey contains four general approaches to build these S-Boxes: algebraic constructions, pseudo-random generation, heuristic search and constructions from small S-Boxes. In the first case S-Boxes are built by means of well known algebraic rules, like the S-Box of the standard AES [14]. In the other hand, pseudo-random S-Boxes are generated following a series of independent experiments, like the S-Box of the standard of the Russian Federation Kuznyechik. In addition, heuristic techniques work on a set of S-Boxes searching for the improvement of one or more of their cryptographic properties [22, 27]. Finally, the construction of S-Boxes from small ones and finite field multiplication allow the generation of high nonlinear substitutions with optimal algebraic characteristics [15, 18]. All these methods have their own advantages and disadvantages. Thus, every year the search for new and cryptographically better S-Boxes is a widely researched field. In this fashion, this paper introduces a new strategy to generate robust key-dependent 8-bit S-Boxes mixing some approaches from the existing literature.

2. PRELIMINARIES

Let $\mathbb{F}_2^n$ the n-dimensional vector space over the finite field $\mathbb{F}_2$ with two elements, 0 and 1, let $\mathbf{0} = (0, 0, \dots, 0)$ be the null vector of $\mathbb{F}_2^n$ and furthermore, we denote $\mathbb{F}_2^{n*} = \mathbb{F}_2^n \setminus \{\mathbf{0}\}$. For $n \in \mathbb{Z}_+$[1] let $\mathbb{F}_{2^n}$ denote the field with 2^n elements. Such field is defined as the set of polynomials with coefficients from $\mathbb{F}_2$ and degree at most $n-1$. The field addition is the usual addition of polynomials, and the field multiplication is the multiplication of polynomials modulo a fixed irreducible polynomial of degree n. It can be summarized by the isomorphism

$$\mathbb{F}_{2^n} \cong \mathbb{F}_2[X]/P(x),$$

where $P(x)$ is an irreducible polynomial, i.e. $P(x)$ cannot be factored into polynomials of strictly lower degree.

For any $n \in \mathbb{Z}_+$, the vectors from $\mathbb{F}_2^n$ can be interpreted as integers, such that the leftmost bits

[1] The notation $\mathbb{Z}_+$ denotes the set of positive integers. For $n \in \mathbb{Z}_+$, $\mathbb{Z}_n$ denotes the set $\{0, 1, 2, \dots, n-1\}$.

correspond to the most significant bits. Let $u \in \mathbb{F}_2^n$, then, the same vector can be written as follows:

$$u = (u_0, \dots, u_{n-1}) \in \mathbb{F}_2^n, \quad \Leftrightarrow \quad \sum_{i=0}^{n-1} u_i X^{n-i} \in \mathbb{F}_2[X], \quad \Leftrightarrow \quad u = \sum_{i=0}^{n-1} u_i 2^{n-1-i} \in \mathbb{Z}_n.$$

2.1. S-Boxes and their representations

An *(n,m)-function* F is a mapping from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$, i.e., $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. When $m = 1$, F=f is called a Boolean function and if $m > 1$ the function F is known as vectorial Boolean function or substitution box (S-Box), which very often is denoted by $\mathcal{S}$. Any substitution box $\mathcal{S}$, can be defined as the vector $\mathcal{S} = (f_1, f_2, ..., f_m)$ where the Boolean functions $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$ are called the coordinate functions of $\mathcal{S}$. The set of all linear combinations of the coordinate functions is called the component functions of $\mathcal{S}$, and they are involved in the determination of most of the cryptographic properties of the substitution [10].

Any substitution $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ can be represented as a list of values (lookup table) with each output value ranging from 0 to $2^m - 1$. Altogether, the S-Box can be represented as the binary matrix of 2^n rows and m columns, where each column represent one of the coordinate functions of the substitution, which is known as truth table. In addition, the *n-bit* S-Box $\mathcal{S}$ can be represented as a univariate polynomial in $\mathbb{F}_{2^n}[X]$ as follows [10]:

$$\mathcal{S}(X) = \sum_{i=0}^{2^n-1} A_i X^i, A_i \in \mathbb{F}_{2^n}$$

This polynomial representation is unique, since if not, there would exist two distinct polynomials of degree less than or equal to $(2^n - 1)$ taking the same value at 2^n distinct points, which is impossible. One should note that the univariate representation is dependent on the basis chosen for identifying $\mathbb{F}_2^n$ and $\mathbb{F}_{2^n}$ [10]. Finally, we refer to the *algebraic normal form (ANF)* representation of each component function of one substitution S, given that the algebraic degree and the algebraic immunity of both, Boolean functions and S-Boxes, is related to this representation. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be an arbitrary n-variable Boolean function. For some fixed $i = 0, 1, ..., n-1$, f can be written as a sum over $\mathbb{F}_2$ of distinct t-order products of its arguments, $0 \leq t \leq n-1$. This is called the *algebraic normal form* of f [15].

2.2. Relevant properties of S-Boxes

Let $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a substitution box, S is said to be balanced if each value $x \in \mathbb{F}_2^m$ appears the same number of 2^{n-m} times. When $n = m$, it is usual that S is a bijective mapping from $\mathbb{F}_2^n$ to itself, $i.e$, that each output appears exactly once. These S-Boxes are called permutations on $\mathbb{F}_{2^n}$ [11] and in what follows we restrict ourselves to the study of such S-Boxes.

For any $u, v \in \mathbb{F}_2^n$ the *Walsh–Hadamard transform* $\mathcal{W}_{\mathcal{S}}(u, v)$ of an n-bit S-Box $\mathcal{S}$ is defined as [11]:

$$\mathcal{W}_{\mathcal{S}}(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle v, \mathcal{S}(x) \rangle \oplus \langle u, x \rangle} \tag{2.1}$$

where $\langle x, y\rangle = \bigoplus_{i=0}^{n-1} x_i y_i$ is the *scalar product* of the vectors $x, y \in \mathbb{F}_2^n$. Here, $\oplus$ represents the addition modulo two or bitwise e**X**clusive **OR** (XOR). The property of nonlinearity is directly related to the maximum absolute value of the Walsh-Hadamard transform of S and it can be expressed as:

$$\mathcal{N}_{\mathcal{S}} = 2^{n-1} - \frac{1}{2} \max_{\mathbf{0} \neq v, u \in \mathbb{F}_2^n} |\mathcal{W}_{\mathcal{S}}(u, v)| \tag{2.2}$$

According to Nyberg [36], for any function $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and any $u, v \in \mathbb{F}_2^n$ one can define

$$\Delta_{\mathcal{S}}(u, v) = \#\{x \in \mathbb{F}_2^n : \mathcal{S}(x \oplus u) \oplus \mathcal{S}(x) = v\}$$

and the *differential uniformity* (also called *δ-uniformity*), denoted by $\delta_{\mathcal{S}}$, is defined as:

$$\delta_{\mathcal{S}} = \max_{\mathbf{0} \neq u, v \in \mathbb{F}_2^n} \Delta_{\mathcal{S}}(u, v). \tag{2.3}$$

For any n-bit permutation $\mathcal{S}$, the *δ-uniformity* of $\mathcal{S}$ satisfies the following inequality $\delta_{\mathcal{S}} \geq 2$, as explained in [10, 11, 36].

Recall the representation of Boolean functions and S-Boxes in their algebraic normal form. The *algebraic degree* of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is the maximum order of the terms appearing in its algebraic normal form. Hence, the algebraic degree of a substitution box $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is the maximum algebraic degree of its component functions [11]. Moreover, one should note that the minimum degree of S, *i.e.* the smallest degree of the component Boolean functions of S, must be as high as possible [15]. In this paper we denote such degree as $deg(\mathcal{S})$. For any n-bit S-Box $\mathcal{S}$, the following inequality holds [15]:

$$1 \leq deg(\mathcal{S}) \leq n - 1 \tag{2.4}$$

The annihilator of a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a Boolean function $g : \mathbb{F}_2^n \to \mathbb{F}_2$ such that $f \cdot g = 0$ [15]. For any Boolean function *f*, the algebraic immunity of f is the minimum value d such that f or $f \oplus 1$ has nonzero annihilator of degree d. There are different definitions of the algebraic immunity of S-Boxes [11]. Particularly, we focus on the concept of graph algebraic immunity. Let $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an arbitrary permutation. The *graph algebraic immunity* of $\mathcal{S}$ is defined as [15]:

$$AI_{gr}(\mathcal{S}) = \min\{deg\ \ p | 0 \neq p \in \mathbb{F}_2[z_1, ..., z_{2n}], p(gr(S)) = 0\} \tag{2.5}$$

where $gr(S) = \{(x, \mathcal{S}(x)) | x \in \mathbb{F}_2^n\} \subseteq \mathbb{F}_2^{2n}$. In [5], some bound on the values of the algebraic immunity is given such that the graph algebraic immunity of $\mathcal{S}$ is upper bounded by the value *d*, which is the minimum positive integer that satisfies:

$$\sum_{i=0}^{d} \binom{2n}{i} > 2^n \tag{2.6}$$

There exists certain methods of analysis of block ciphers exploiting the existence of polynomial relations involving the input x of the S-Box $\mathcal{S}$ and its output $\mathcal{S}(x)$ (see [11])and, in order to increase the strength of a block cipher against these methods, we need to maximize the graph algebraic immunity parameter.

Two n-bit S-Boxes $\mathcal{S}_1$ and $\mathcal{S}_2$ are *linear* (resp. *affine*) *equivalent* if there exist linear (resp. affine) mappings A_1, A_2, such that $\mathcal{S}_2 = A_2 \circ \mathcal{S}_1 \circ A_1$. It is well-known that δ-uniformity, nonlinearity and (minimum) algebraic degree remains invariant under linear (resp. affine) equivalence [11].

Finally, an element $x \in \mathbb{F}_2^n$ is called a *fixed point* of an n-bit S-Box $\mathcal{S}$, if $\mathcal{S}(x) = x$.

3. STATE-OF-THE-ART ON S-BOXES

The most popular way to construct cryptographically good S-Boxes is the generation based on the inversion in the finite field $\mathbb{F}_{2^n}$ as remarked in [27]. For instance, the S-Box of AES was built using this kind of construction, which achieves the best known values for algebraic degree, nonlinearity and differential uniformity. Although, the finite field inversion has a flaw with respect to the existence of a system of polynomial equations with low degree which leads a weakness towards algebraic attacks [13]. Such vulnerability is measured through the property of graph algebraic immunity which is taken instead of the notion of the component algebraic immunity of S-Boxes [39].

In literature related to S-Boxes one can find a variety of methods which solve the algebraic vulnerabilities of the finite field inversion, exhibiting optimal values of algebraic degree and graph algebraic immunity while values of nonlinearity and differential uniformity of the resulting S-Boxes are not optimal but are still good [15, 18, 27, 31, 35] (See Table 1). According to authors of [15, 18], among other researchers, a cryptographically strong 8-bit S-Box $\mathcal{S}$ is an 8-bit permutation with absence of fixed points that satisfies the following cryptographic criteria:

1. Low value of differential uniformity, *i.e.*, $\delta_{\mathcal{S}} \leq 8$;
2. High value of nonlinearity, *i.e.*, $\mathcal{N}_{\mathcal{S}} \geq 100$;
3. Maximum value of minimum algebraic degree, *i.e.*, $deg(\mathcal{S}) = 7$;
4. Maximum value of graph algebraic immunity, *i.e.*, $AI_{gr}(\mathcal{S}) = 3$.

Although the nonlinearity values of 8-bit bijective S-Boxes ranges up to a value of 112 and the best known value of differential uniformity for these permutations is 4 [11], there are no references of such a substitution which have optimal algebraic characteristics and nonlinearity above 108 and/or differential uniformity below 6, which today constitutes an open problem for cryptography researchers around the world [23]. It worth to remark that the goal of the present paper is not to solve the aforementioned problem, instead, we use the proposal from [15, 18], with the best results reported in public literature, to generate key-dependent 8-bit S-Boxes having good cryptographic properties.

S-Box $\mathcal{S}$	of AES	from [15]	from [18]	from [27]	from [31]	from [35]
$\delta_{\mathcal{S}}$	4	6	6	6	8	6
$\mathcal{N}_{\mathcal{S}}$	112	108	108	104	104	104
$deg(\mathcal{S})$	7	7	7	7	7	7
$AI_{gr}(\mathcal{S})$	2	3	3	3	3	3

Table 1: Some of the best results reported in the public literature w.r.t. the cryptographic properties of 8-bit S-Boxes generated by different methods and comparison with the S-Box of AES.

3.1. Generation of key-dependent 8-bit S-Boxes

Most of proposals referring to key-dependent S-Box generation does not take into account the algebraic characteristics of the results. Furthermore, the values of nonlinearity and differential uniformity of the resulting substitutions are not good enough to be considered strong S-Boxes. Instead, the security provided by key-dependent S-Boxes rest in the uncertainty they grant into the encryption process, e.g. a pseudo-random 8-bit permutation provides approximately 1684 bits of uncertainty but it can have undesirable cryptographic properties. Such key-dependent S-Boxes are proposed in [1, 19, 32, 41]. Perhaps the best results in this regard belong to key-dependent S-Boxes based on affine transformations to the original AES S-Box [4, 7, 19, 29, 34] ensuring high nonlinearity, low differential uniformity and optimal algebraic degree, although they still do not solve the graph algebraic immunity issue. To this day, how to generate cryptographically strong key-dependent S-Boxes is an unsolved problem in the specialized literature. In this fashion, recent works as [2, 17, 25, 26] attempt to find a set of good key-dependent 8-bit permutations, however, a deeper analysis on these papers shows that the results obtained are still far from the desired ones. Table 2 present the set of properties analyzed in this paper for various key-dependent S-Boxes of new generation.

S-Box $\mathcal{S}$	from[2]	from [17]	from [25]	from [26]
$\delta_{\mathcal{S}}$	4	10	10	10
$\mathcal{N}_{\mathcal{S}}$	112	94	92	96
$deg(\mathcal{S})$	7	7	6	6
$AI_{gr}(\mathcal{S})$	2	3	3	3

Table 2: Comparison between the best results w.r.t. the cryptographic properties of key-dependent 8-bit S-Boxes generated by some of the recent methods presented in the literature.

4. THE CORE OF AN OPTIMIZATION ALGORITHM FOR GENERATING ROBUST KEY-DEPENDENT S-BOXES

In this section we introduce the main components of the algorithm we present in Section 5 of this paper for the generation of robust key-dependent S-Boxes. Firstly, we briefly resume the construction proposed by de la Cruz in [15] and later we describe the algorithm designed by Freyre in [20] to exploit the advantages of de la Cruz like constructions to produce high quality 8-bit S-Boxes.

4.1. Generation of 8-bit S-Boxes from smaller 4-bit S-Boxes

De la Cruz introduced in [15] a new method for constructing $2k$-bit bijective S-Boxes using k-bit permutations and finite field multiplications in $\mathbb{F}_{2^k}$. More exactly, this construction uses the permutation polynomial $\mathcal{P}_d(x) = x^{2^k-2}$ over $\mathbb{F}_{2^k}$ and two k-bit permutations h_1 and h_2 in such a way that permutations $\mathcal{S} : \mathbb{F}_{2^{2k}} \longrightarrow \mathbb{F}_{2^{2k}}$ are defined for all $l, r \in \mathbb{F}_{2^k}$ as $\mathcal{S}(l \parallel r) = (l_1 \parallel r_1)$. The output value $(l_1 \parallel r_1)$ of the aforementioned construction is computed by the rules:

- $l1 = \begin{cases} h_1(l), & \text{if} \quad r = 0; \\ \mathcal{P}_d(l \otimes r), & \text{if} \quad r \neq 0. \end{cases}$
- $r1 = \begin{cases} h_2(r), & \text{if} \quad l_1 = 0; \\ l_1 \otimes \mathcal{P}_d(r), & \text{if} \quad l_1 \neq 0. \end{cases}$

where the operator ($\bigotimes$) denotes the finite field multiplication of two elements $a, b \in \mathbb{F}_{2^k}$. The concrete 8-bit S-Boxes compiled in Table 1 shows the better results obtained in [15] using the permutations $h_1 = h_2 = \{\mathsf{0}, \mathsf{1}, \mathsf{e}, \mathsf{9}, \mathsf{f}, \mathsf{5}, \mathsf{c}, \mathsf{2}, \mathsf{b}, \mathsf{a}, \mathsf{4}, \mathsf{8}, \mathsf{d}, \mathsf{6}, \mathsf{3}, \mathsf{7}\}$. From now on we will refer to this construction as $\mathbf{\Pi}$ and its high level structure is shown in Figure 1.

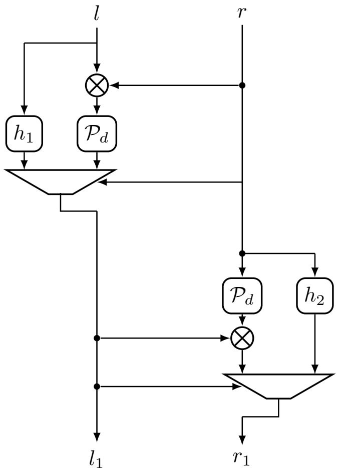

Figure 1: High level structure of construction Π.

To test the quality of the 8-bit S-Boxes generated by $\mathbf{\Pi}$ the author of [15] conduct an exhaustive search over all affine equivalence classes for 4-bit permutations checking that the following properties holds:

$$98 \leq \mathcal{N}_{\mathbf{\Pi}} \leq 108; \quad 6 \leq \delta_{\mathbf{\Pi}} \leq 18$$

The largest experiment carried out involves a maximum of 2^{20} 4-bit permutations with the particular characteristic that $h_1 = h_2 = h$. Nonetheless, the author shows that it is not necessary that $h_1 = h_2$ to obtain 8-bit S-Boxes which satisfy the aforementioned criteria on their properties. In this experiment the search is aborted if construction $\mathbf{\Pi}$ produces an 8-bit S-Box with nonlinearity value of 108 and satisfying the remaining cryptographic criteria mentioned in Section 3; however, the statistical analysis conducted in Section 3 of [20] for the construction $\mathbf{\Pi}$ reveals that the algebraic degree of some S-Boxes generated by such construction is often equal to 6. Hence, we have decided to check when an S-Box produced through construction $\mathbf{\Pi}$ entirely satisfies the cryptographic criteria established in Section 3 of this paper, *i.e.*,

$$\mathcal{N}_{\Pi} \geq 100; \quad \delta_{\Pi} \leq 8; \quad deg(\mathbf{\Pi}) = 7; \quad AI_{gr}(\Pi) = 3$$

To do so, we select at random 2^{20} different pairs of 4-bit permutations h_1, h_2, then we apply $\mathbf{\Pi}$ using h_1, h_2 as underlying components and for each of the generated 8-bit S-Boxes we check their values of nonlinearity, differential uniformity, algebraic degree and graph algebraic immunity. In Table 3 we present the number of S-Boxes with a fixed value of nonlinearity that passed the remaining criteria.

Nonlinearity	100	102	104	106	108
Number of S-Boxes	1063	26484	52771	590	0

Table 3: S-Boxes passing all cryptographic criteria.

As we can see, only the **7.7%** of the S-Boxes tested in our experiment satisfy the criteria exhibited in Section 3. Also, this experiment shows that S-Boxes having nonlinearity 102 and 104 are the common outputs of construction $\mathbf{\Pi}$ while, in the other hand, we do not find any permutation such that their nonlinearity value reach 108.

4.2. A simple local search method with good results

Heuristic techniques have been successfully applied to the generation of cryptographically strong S-Boxes; and particularly, for 8-bit S-Boxes. Most of the research papers on optimization which achieve the best values of nonlinearity use the help of some functions over the linear spectrum of the target S-Box to improve the quality of the final results [12, 22, 27, 28, 38, 44]. For instance, the author of [20] obtains a set of S-Boxes with the best cryptographic parameters reported in [15, 18] by means of the combination of a novel cost function introduced by Freyre *et al.* [21] and a simple heuristic method. The algorithmic proposal of [20] exploits several $\mathbf{\Pi} - \mathit{like}$ constructions of S-Boxes reducing the traditional search space for 8-bit S-Boxes (with 256! elements) to the search space for the requested 4-bit permutations (with 16! elements) with the assistance of two mutation operators: *swap* and *insert* [16] whose procedure are presented below.
Let $p = (p_0, p_1, \cdots, p_{k-1})$ be a permutation of k elements and $0 \leq x \neq y \leq k-1$ where $k \geq 2$, then

- $p_1 = \mathsf{SwapMutation}(p, x, y) \Longleftrightarrow p_1[i] = \begin{cases} p[i] & \text{if} \quad 0 \leq i < k, \quad i \neq x, y \\ p[y] & \text{if} \quad i = x \\ p[x] & \text{if} \quad i = y \end{cases}$

- $p_2 = \mathsf{InsertMutation}(p, x, y) \Longleftrightarrow p_2[i] = \begin{cases} p[i] & \text{if} \quad 0 \leq i < x \quad \text{or} \quad y < i < k \\ p[y] & \text{if} \quad i = x \\ p[i-1] & \text{if} \quad x < i \leq y \end{cases}$

The first of these operators (*swap*) is used to construct the neighborhood of a given 4-bit permutation h, while the second one (*insert*) is used to prevent the algorithm from falling into regions with local optimum values and continue the search for an S-Box with desired cryptographic characteristics. Furthermore, after the *insert mutation* the best S-Box found by the algorithm is set to be the output of the selected construction, e.g. $\mathbf{\Pi}$, using the 4-bit component result of the mutation instead of using the local optimum S-Box reached and therefore avoiding biased comparisons in the next iterations of the method.

5. PUTTING ALL TOGETHER: ROBUST KEY-DEPENDENT 8-BIT S-BOXES

Recall from the statistical analysis conducted in Section 4.1 whose results are shown in Table 3 that construction $\mathbf{\Pi}$ produce a good solution in approximately 1 of every 14 resulting permutations and it does not frequently produce S-Boxes with the best cryptographic parameters reported for any 8-bit permutation whose algebraic characteristics are optimal, *i.e.*, nonlinearity and differential uniformity equals to 108 and 6 respectively [15, 18, 23]. However, the method introduced in [20] explore the search space of 4-bit components requested by $\mathbf{\Pi}$ such that the aforementioned 8-bit S-Boxes are repeatedly produced with an average of approximately $\mathbf{2^{12}}$ solution evaluations. Moreover, this method ensures that all of the generated 8-bit S-Boxes satisfies the security criteria established in Section 3. Hence, we made some modifications to this proposal in order to obtain a set of cryptographically strong key-dependent 8-bit S-Boxes.

The heuristic method we propose receive as input two pseudo-random 4-bit permutations h_1, h_2, two pseudo-random 8×8 invertible binary matrices $\mathcal{A}, \mathcal{B}$, two binary vectors $\alpha, \beta \in \mathbb{F}_2^8$ and the desirable minimum value of nonlinearity of the 8-bit S-Box that will be produced by the method, $\mathcal{N}_G$. It worth to remark that all the input parameters of the method are dynamically generated from the underlying block cipher's key. Firstly, in the initialization stage we set up all the relative to the optimization process carried later, e.g, set up the best solution found by the algorithm as the result of applying construction $\mathbf{\Pi}$ using the 4-bit permutations h_1, h_2. Then, the optimization phase begins with a search for any combination of 4-bit permutations h'_1 and h'_2 generated, for given input values $i, j \in \mathbb{F}_2^4$, as follow

- $S_1 = \mathbf{\Pi}(h_1,\ \mathsf{SwapMutation}(h_2, i, j))$
- $S_2 = \mathbf{\Pi}(\mathsf{SwapMutation}(h_1, i, j),\ h_2)$
- $S_3 = \mathbf{\Pi}(\mathsf{SwapMutation}(h_1, i, j),\ \mathsf{SwapMutation}(h_2, i, j)))$

Then we select the best of the $\mathcal{S}_i$ according to the fitness conditions of our problem. The condition of improvement of $\mathcal{S}'$ over $\mathcal{S}$ is given by the relation:

$$\mathcal{N}_{\mathcal{S}} < \mathcal{N}_{\mathcal{S}'} \text{ or } (\mathcal{N}_{\mathcal{S}'} = \mathcal{N}_{\mathcal{S}} \text{ and } \mathsf{Cost}(\mathcal{S}') < \mathsf{Cost}(\mathcal{S}))$$

where the function $\mathsf{Cost}(\mathcal{S})$ is taken from [21].

Again, we compare the best of the $\mathcal{S}_i$ with the best solution of the algorithm, and if better, the 4-bit permutations who generate $\mathcal{S}_i$ substitute h_1 and h_2 for the following rounds of the procedure. If the search do not find any combination of 4-bit permutations h'_1 and h'_2 so that $\mathbf{\Pi}(h'_1, h'_2)$ satisfies the above condition w.r.t $\mathbf{\Pi}(h_1, h_2)$ then we reset the search process following the procedure of [20] through the insert mutation whose purpose is discussed in the preceding section. Finally, if the optimization method reach a pair of 4-bit permutations which guarantees that $\mathbf{\Pi}(h_1, h_2)$ satisfies the differential uniformity and algebraic restrictions and also its nonlinearity is at least $\mathcal{N}_G$, then the algorithm returns an affine equivalent transformation of $\mathcal{S} = \mathbf{\Pi}(h_1, h_2)$ which is intended to eliminate a high number of fixed points in the resulting S-Box. The affine equivalent transformation ensures that the properties of the S-Box holds and solves the problem of minimizing the number of fixed points. The proposed affine transformation proceed as follows:

$$\mathcal{S}'(x) = \mathsf{Ext}_{\mathsf{Affine}}[\mathcal{S}, \mathcal{A}, \mathcal{B}, \alpha, \beta](x) = \mathcal{A}(\mathcal{S}(\mathcal{B}(x) \oplus \alpha)) \oplus \beta, \qquad \forall x \in \mathbb{F}_2^8$$

Value of $\mathcal{N}_G$	Average solution evaluations	Average time (Seconds)
100	334	4.273
102	346	4.123
104	345	4.058
106	455	5.068
108	73004	1358.253

Table 4: Average number of solution evaluations and execution time for each value of $\mathcal{N}_G$ before K-DOA stops.

where α, β are selected a priori from $\mathbb{F}_2^8$. Moreover, the affine transformation is not restricted to the elimination of the fixed points of the S-Box resulting from the optimization process, it has an important role towards the security of the proposed S-Box generation scheme and it is discussed later in this section. The pseudo-code of the Key-Dependent Optimization Algorithm (K-DOA) is presented in Algorithm 1.

5.1. Advantage and performance of K-DOA

If one carefully analyze the results presented in Table 3, it may be understandable that there is a high probability of obtaining an S-Box having desired cryptographic parameters by means of K-DOA. Moreover, it is very likely that for input permutations h_1, h_2, K-DOA finds in a single iteration a pair of pemutations h'_1, h'_2 such, once applied construction $\mathbf{\Pi}$, the resulting 8-bit S-Box have nonlinearity equals to 102 or 104. Then, one can assume that for any input $(h_1, h_2, \mathcal{A}, \mathcal{B}, \alpha, \beta, \mathcal{N}_G)$ of K-DOA the output will be an 8-bit S-Box $\mathcal{S}$ which satisfies the criteria introduced in Section 3. Hence, we measure the performance of K-DOA by running 100 independent executions of the algorithm for all possible values of parameter $\mathcal{N}_G$. The goal of these experiments is to average the number of solution evaluations that K-DOA runs before it reaches its stop condition. In Table 4 we present the results of these experiments, as well as the average execution time recorded for each value of $\mathcal{N}_G$.

As we already analyzed, the construction $\mathbf{\Pi}$ only produce good results in approximately 1 out of 13 cases and therefore a direct generation of an S-Box using $\mathbf{\Pi}$ may result in some flaws w.r.t the cryptographic properties in the scope of this paper. Conversely, K-DOA ensures that the nonlinearity value of the resulting S-box is at least $\mathcal{N}_G$, which does not mean that this value cannot be higher than the selected parameter. For instance, for $\mathcal{N}_G = 104$ we do produce S-Boxes having nonlinearity 106. Such results are possible since the remaining criteria assumed as algorithmic constraints prevent that any S-Box is returned unless the aforementioned restrictions are satisfied too, even when the S-box satisfies the nonlinearity criteria, *i.e.*, $\mathcal{N}_S \geq \mathcal{N}_G$. In the particular case of $\mathcal{N}_G = 108$, Table 4 shows that these S-boxes are produced in approximately $1.11 \cdot 2^{16}$ solution evaluations . If compared with the results in Table 3, where none of the 8-bit S-Boxes generated achieve this value of nonlinearity, one may easily realize the advantages of K-DOA with respect to only use the construction $\mathbf{\Pi}$.

```
input  : Two pseudo-random 4-bit permutations $h_1, h_2$.
input  : Two pseudo-random invertible $8 \times 8$ binary matrices $\mathcal{A}$ and $\mathcal{B}$.
input  : Two random vectors $\alpha, \beta \in \mathbb{F}_2^8$
input  : A desired minimum value of nonlinearity $100 \leq \mathcal{N}_G \leq 108$.
output: An 8-bit S-Box having good cryptographic parameters.
// Initialization
$\mathcal{S} \leftarrow \Pi(h_1, h_2)$ // Apply construction $\Pi$ using $h_1, h_2$ as 4-bit components.

// Optimization
while True do
    upgrade = NULL
    for $0 \leq i < 15$ do
        for $j = i + 1,\ j < 16$ do
            // Generate the offspring of solutions from $h_1, h_2$ as stated in Section 5
            and set $\mathcal{S}'$ to be the best of the offpring
            $\mathcal{S}' \leftarrow \mathsf{BEST}(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$
            if $\mathcal{N}_{\mathcal{S}} < \mathcal{N}_{\mathcal{S}'}$ or ($\mathcal{N}_{\mathcal{S}} = \mathcal{N}_{\mathcal{S}'}$ and $\mathsf{Cost}(\mathcal{S}') < \mathsf{Cost}(\mathcal{S})$) then
                $\mathcal{S} \leftarrow \mathcal{S}'$
                if $deg(\mathcal{S}) = 7$ and $AI_{gr}(\mathcal{S}) = 3$ and $\delta_{\mathcal{S}} \leq 8$ and $\mathcal{N}_{\mathcal{S}} \geq \mathcal{N}_G$ then
                    return $\mathsf{Ext}_{\mathsf{Affine}}(\mathcal{S}, \mathcal{A}, \mathcal{B}, \alpha, \beta)$
                else
                    // Record the permutations $h_1', h_2'$ which generate $\mathcal{S}'$
                    upgrade = $(\mathsf{h}_1', \mathsf{h}_2')$
                end
            end
        end
    end
    if upgrade = NULL then
        $h_1 \leftarrow \mathsf{InsertMutation}(h_1, 0, 15)$
        $h_2 \leftarrow \mathsf{InsertMutation}(h_2, 0, 15)$
        $\mathcal{S} \leftarrow \Pi(h_1, h_2)$
    else
        $h_1 \leftarrow$ upgrade.$\mathsf{h}_1'$
        $h_2 \leftarrow$ upgrade.$\mathsf{h}_2'$
        $\mathcal{S} \leftarrow \Pi(h_1, h_2)$
    end
end
```

Algorithm 1: Pseudo-code of K-DOA.

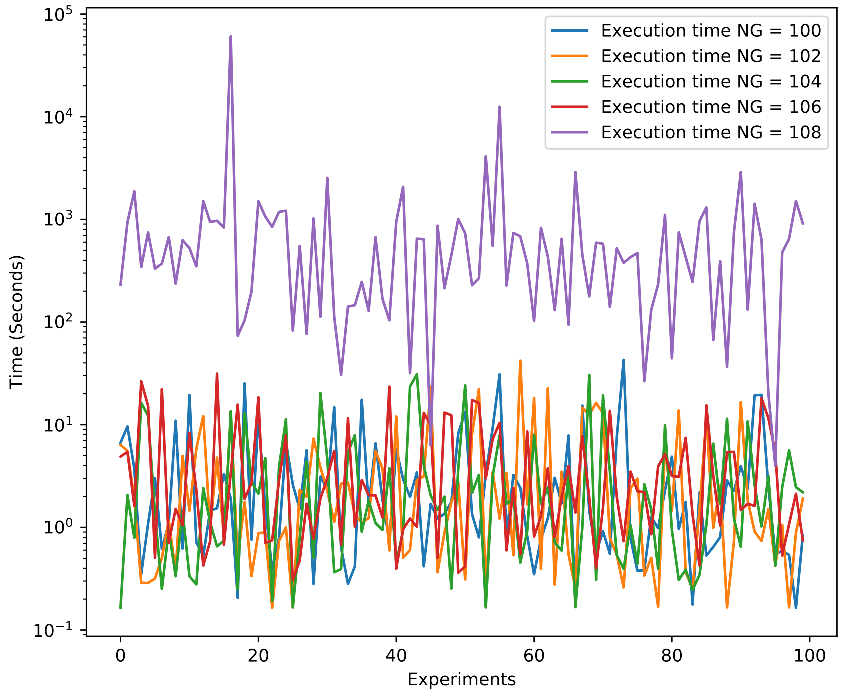

Figure 2: Behavior of the execution time for $100 \leq \mathcal{N}_G \leq 108$.

In terms of performance, one can notice from the results presented in Table 4 that obtain a higher nonlinearity value came with a higher execution time of the algorithm, which is intuitively expected since K-DOA produce, for a selected parameter $\mathcal{N}_G$, S-Boxes with nonlinearity equal or greater than this value and they must be generated as quick as possible. Thus, the relationship between the desired nonlinearity value and the execution time must be considered before any practical implementation of K-DOA. Figure 2 shows the behavior of the execution time of the experiments conducted in this section for $100 \leq \mathcal{N}_G \leq 108$.

Obviously, the superiority in terms of time consumption of K-DOA to produce S-Boxes with nonlinearity at least 108 (purple plot) with respect to the other nonlinearity values implies that, although from the theoretical point of view they constitute an interesting result, special conditions are needed for its practical application. Nevertheless, the results in Table 4 w.r.t the average execution time of our method for $100 \leq \mathcal{N}_G \leq 106$, one may see that they are feasible for real-life application in symmetric encryption schemes. In addition, Figure 3 shows the augmented plot of Figure 2 w.r.t the behavior of the execution time of the experiments conducted in this section for these nonlinearity values which is corresponding to data presented in Table 4, where the worst execution time of K-DOA does not exceed the 50 seconds.

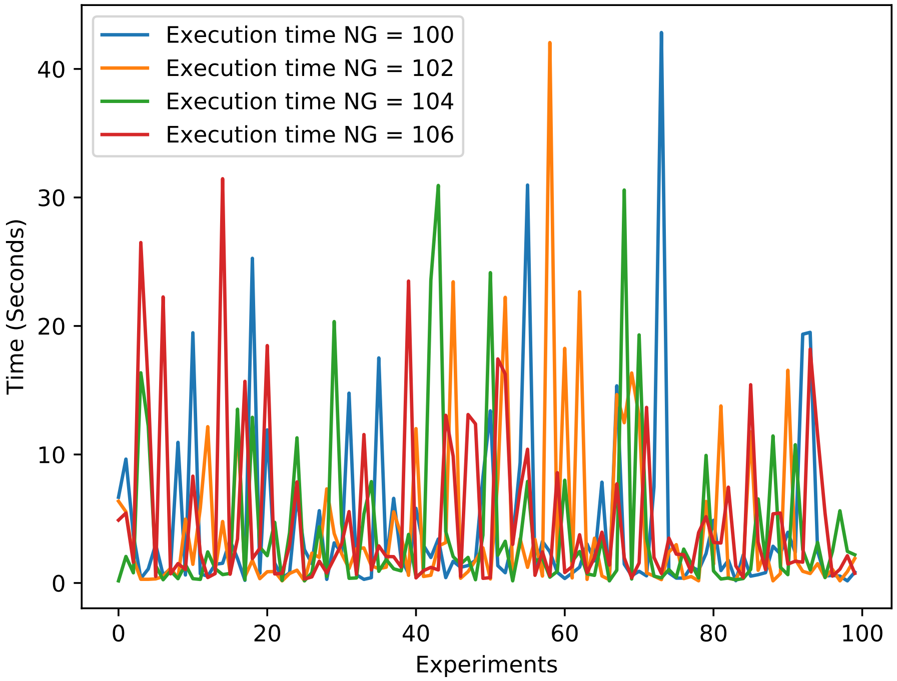

Figure 3: Behavior of the execution time for $100 \leq \mathcal{N}_G \leq 106$.

A lighter version of K-DOA can be applied to reduce the execution time of the algorithm by supplying a single 4-bit permutation as input instead of two, similar to the presented in [20]. Although it substantially reduce the execution time of the original method, it also reduce the search space of the input permutations from $(16!)^2$ to 16!, compromising the security of output S-Box. In the next section we board the security analysis of the output S-Boxes of our proposal.

5.2. Security analysis of the S-Boxes produced by K-DOA

In the security branch, due to the hardness of exploring the whole space of 4-bit permutations to find 8-bit S-Boxes having the aforementioned properties in Section 2, one can roughly estimate the number of S-Boxes resulting of K-DOA as

$$\#\mathcal{S} = \frac{(16!)^2}{13} \cdot \#\mathsf{Ext}_{\mathsf{Affine}}[\mathcal{S}, \mathcal{A}, \mathcal{B}, \alpha, \beta] \approx 2^{84} \cdot 2^{140} = 2^{224}$$

which provides sufficient uncertainty to avoid a brute force approach for guessing a secret 8-bit S-Box. To continue the security analysis, let's now suppose that an attacker is able to successfully predict an S-Box $\mathbf{S}$ resulting from K-DOA, then to predict the key bits used for constructing permutations h_1 and h_2 as well as boolean matrices $\mathcal{A}$, $\mathcal{B}$ using the knowledge of $\mathbf{S}$ is a difficult task. It is well-known that the construction of Π has the so-called TU-decomposition [9, 37]. So, an adversary can use the results given in [37] for obtaining this decomposition. However, it will be hard in first place, to obtain exactly those transformations used by the K-DOA when constructing an S-Box with the desired properties. Recall, that the construction of Π uses the permutation polynomial $\mathcal{P}_d(x) = x^{2^k-2}$ over $\mathbb{F}_{2^k}$ and two k-bit permutations h_1 and h_2 in such a way that the nonlinear bijective transformation $\Pi : \mathbb{F}_{2^{2k}} \longrightarrow \mathbb{F}_{2^{2k}}$ is defined as $\Pi(l \parallel r) = (l_1 \parallel r_1)$, for all $l, r \in \mathbb{F}_{2^{2k}}$. The output value $(l_1 \parallel r_1)$ of the aforementioned construction is computed by using the next relations:

- $l_1 = h_1(l)$ if $r = 0$, else $l_1 = \mathcal{P}_d(l \otimes r)$
- $r_1 = h_2(l)$ if $l_1 = 0$, else $r_1 = l_1 \otimes \mathcal{P}_d(r)$

As we can see from the output values of Π, there are $(2^k-1)^2$ values which are unchanged despite the selection of small subcomponents h_1 and h_2, respectively. In this way, when swapping a pair of values in h_1 and h_2 we can expect that at most $2^{2k}-(2^k-1)^2 = 2^{k+1}-1$ values of the resulting permutation Π could be changed. In a such scenario, no matter how the key-dependent subcomponents h_1 and h_2 are generated, an adversary will always predict the aforementioned $(2^k-1)^2$ unchanged values. For this reason, we utilize the extended affine transformation, which ensure that changing even one bit of the key used to construct the S-Box Π, should always lead to a different resulting permutation. Moreover, by using the extended affine transformation we guarantee that any pairs of keys, lead to extremely different S-Boxes.

6. CONCLUSIONS

In this paper we have introduced an algorithm to produce cryptographically strong key-dependent 8-bit S-Boxes which combines two research areas on S-Box generation: constructions from small S-boxes and optimization methods. Our proposal Key-Dependent Optimization Algorithm (K-DOA) allows to generate a large set of key-dependent permutations with maximum value of minimum algebraic degree $deg(\mathcal{S}) = 7$, maximum value of graph algebraic immunity $AI_{gr}(\mathcal{S}) = 3$ and values of differential uniformity $\delta_{\mathcal{S}} \leq 8$ and nonlinearity $\mathcal{N}_{\mathcal{S}} \geq 100$ comparable with those reported in the specialized literature for static 8-bit S-Boxes having good cryptographic properties.

RECEIVED: AUGUST, 2022.
REVISED: FEBRUARY, 2023.

REFERENCES

[1] AGARWAL, P., SINGH, A., AND KILICMAN, A. (2018): Development of key-dependent dynamic s-boxes with dynamic irreducible polynomial and affine constant **Advances in Mechanical Engineering**, 10(7):1687814018781638.

[2] AL-DWEIK, A. Y., HUSSAIN, I., SALEH, M. S., AND MUSTAFA, M. T. (2021): A novel method to generate key-dependent s-boxes with identical algebraic properties **arXiv preprint arXiv:1908.09168v2**.

[3] ANDERSON, R., BIHAM, E., AND KNUDSEN, L. (1998): Serpent: A proposal for the advanced encryption standard **NIST AES Proposal**, 174:1–23.

[4] AO, T., RAO, J., DAI, K., AND ZOU, X. (2017): Construction of high quality key-dependent s-boxes **Nonlinearity**, 13(14):15.

[5] ARMKNECHT, F. AND KRAUSE, M. (2006): Constructing single-and multi-output boolean functions with maximal algebraic immunity In **International Colloquium on Automata, Languages, and Programming**, pages 180–191. Springer.

[6] B., S. AND OTHERS. (1998): Twofish: A 128-bit block cipher. In **First AES Candidate Conference**. National Institute of Standards and Technology.

[7] BAI, K. AND WU, C. (2016): An aes-like cipher and its white-box implementation **The Computer Journal**, 59(7):1054–1065.

[8] BINTI MOHAMED, K., ALI, F. H. H. M., ARIFFIN, S., AND PAUZI, M. N. M. (2018): A review of cryptography based on key dependent s-box in block cipher **Selangor Science & Technology Review (SeSTeR)**, 2(2):1–8.

[9] BIRYUKOV, A., PERRIN, L., AND UDOVENKO, A. (2016): Reverse-engineering the s-box of streebog, kuznyechik and stribobr1 In **Annual International Conference on the theory and applications of cryptographic techniques**, pages 372–402. Springer.

[10] CANTEAUT, A. (2016): **Lecture notes on cryptographic Boolean functions** Inria, Paris, France.

[11] CARLET, C. (2021): **Boolean Functions for Cryptography and Coding Theory** Cambridge University Press.

[12] CLARK, J. A., JACOB, J. L., AND STEPNEY, S. (2005): The design of s-boxes by simulated annealing **New Generation Computing**, 23(3):219–231.

[13] COURTOIS, N. T. AND PIEPRZYK, J. (2002): Cryptanalysis of block ciphers with overdefined systems of equations In **International conference on the theory and application of cryptology and information security**, pages 267–287. Springer.

[14] DAEMEN, J. AND RIJMEN, V. (2020): **The Design of Rijndael: The Advanced Encryption Standard (AES). Second Edition.** Springer.

[15] DE LA CRUZ JIMÉNEZ, R. A. (2017): Generation of 8-bit s-boxes having almost optimal cryptographic properties using smaller 4-bit s-boxes and finite field multiplication In **International Conference on Cryptology and Information Security in Latin America**, pages 191–206. Springer.

[16] EIBEN, A. E., SMITH, J. E., ET AL. (2003): **Introduction to evolutionary computing**, volume 53 Springer.

[17] EJAZ, A., SHOUKAT, I. A., IQBAL, U., RAUF, A., AND KANWAL, A. (2021): A secure key dependent dynamic substitution method for symmetric cryptosystems **Peer J Computer Science**, Vol. 7:pp. 587.

[18] FOMIN, D. B. (2019): New classes of 8-bit permutations based on a butterfly structure **Mat. Vopr. Kriptogr.**, 10(2):169–180.

[19] FREYRE, P., CUELLAR, O., DÍAZ, N., AND ALFONSO, A. (2020): From aes to dynamic aes **Journal of Science and Technology on Information security**, 1(11):11–22.

[20] FREYRE-ECHEVARRÍA, A. (2021): On the generation of cryptographically strong substitution boxes from small ones and heuristic search In **10 th Workshop on Current Trends in Cryptology (CTCrypt 2021)**, page 112.

[21] FREYRE-ECHEVARRÍA, A., ALANEZI, A., MARTÍNEZ-DÍAZ, I., AHMAD, M., EL-LATIF, A., AHMED, A., KOLIVAND, H., AND RAZAQ, A. (2020a): An external parameter independent novel cost function for evolving bijective substitution-boxes **Symmetry**, 12(11):1896.

[22] FREYRE-ECHEVARRÍA, A., MARTÍNEZ-DÍAZ, I., PÉREZ, C. M. L., SOSA-GÓMEZ, G., AND ROJAS, O. (2020b): Evolving nonlinear s-boxes with improved theoretical resilience to power attacks **IEEE Access**, 8:202728–202737.

[23] GORODILOVA, A. A., TOKAREVA, N. N., AGIEVICH, S. V., CARLET, C., GORKUNOV, E. V., IDRISOVA, V. A., KOLOMEEC, N., KUTSENKO, A. V., LEBEDEV, R. K., NIKOVA, S., ET AL. (2020): On the sixth international olympiad in cryptography nsucrypto **Journal of Applied and Industrial Mathematics**, 14(4):623–647.

[24] I., Y. (2019): Proposed a permutation and substitution methods of serpent block cipher. **Ibn AL-Haitham Journal For Pure and Applied Science**, 32(2):131–144.

[25] IBRAHIM, S. AND ABBAS, A. M. (2020): A novel optimization method for constructing cryptographically strong dynamic s-boxes **Ieee Access**, 8:225004–225017.

[26] IBRAHIM, S. AND ALHARBI, A. (2020): Efficient image encryption scheme using henon map, dynamic s-boxes and elliptic curve cryptography **IEEE Access**, 8:194289–194302.

[27] IVANOV, G., NIKOLOV, N., AND NIKOVA, S. (2015): Cryptographically strong s-boxes generated by modified immune algorithm In **International Conference on Cryptography and Information Security in the Balkans**, pages 31–42. Springer.

[28] IVANOV, G., NIKOLOV, N., AND NIKOVA, S. (2016): Reversed genetic algorithms for generation of bijective s-boxes with good cryptographic properties **Cryptography and Communications**, 8(2):247–276.

[29] JING, M.-H., CHEN, Z.-H., CHEN, J.-H., AND CHEN, Y.-H. (2007): Reconfigurable system for high-speed and diversified aes using fpga **Microprocessors and Microsystems**, 31(2):94–102.

[30] JUREMI, J., SULAIMAN, S., SAAD, N. H. M., AND RAMLI, J. (2019): A survey on various dynamic s-box implementation in block cipher encryption algorithm **Journal of Applied Technology and Innovation**, 3(1).

[31] KAZYMYROV, O., KAZYMYROVA, V., AND OLIYNYKOV, R. (2014): A method for generation of high-nonlinear s-boxes based on gradient descent. **Mat. Vopr. Kriptogr.**, 5(2):71–78.

[32] KNUDSEN, L. R. (2014): Dynamic encryption **Journal of Cyber Security and Mobility**, Vol. 3(4):357–370.

[33] M., T., G., D., AND H., S. (2018): Block cipher s-box modification based on fisher-yates shuffle and ikeda map. In **2018 IEEE 18th International Conference on Communication Technology (ICCT)**, pages 59–64. IEEE.

[34] MALIK, M. S. M., ALI, M. A., KHAN, M. A., EHATISHAM-UL-HAQ, M., SHAH, S. N. M., REHMAN, M., AND AHMAD, W. (2020): Generation of highly nonlinear and dynamic aes substitution-boxes (s-boxes) using chaos-based rotational matrices **IEEE Access**, 8:35682–35695.

[35] MENYACHIKHIN, A. (2016): Spectral-linear and spectral-difference methods for generating cryptographically strong s-boxes **CTCrypt Preproc. Yaroslavl**, pages 232–252.

[36] NYBERG, K. (1991): Perfect nonlinear s-boxes In **Workshop on the Theory and Application of of Cryptographic Techniques**, pages 378–386. Springer.

[37] PERRIN, L. (2019): Partitions in the s-box of streebog and kuznyechik **IACR transactions on symmetric cryptology**, pages 302–329.

[38] PICEK, S., CUPIC, M., AND ROTIM, L. (2016): A new cost function for evolution of s-boxes **Evolutionary computation**, 24(4):695–718.

[39] PRENEEL, B. ET AL. (2009): On the algebraic immunities and higher order nonlinearities of vectorial boolean functions **Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes**, 23:104.

[40] R., M. (1990): Fast software encryption functions. In **Conference on the Theory and Application of Cryptography**, pages 477–501. Springer.

[41] ROHIEM, A., DIAA, A., MOHAMMED, F., ET AL. (2009): Generation of aes key dependent s-boxes using rc4 algorithm **Aerospace Sciences & Aviation Technology, ASAT**, 13.

[42] SMITH, J. (1971): The design of lucifer, a cryptographic device for data communication. **IBM Research, White Plains, RC 3326**.

[43] SRISAKTHI, S. AND SHANTHI, A. (2020): Towards the design of a stronger aes: Aes with key dependent shift rows (kdsr) **Wireless Personal Communications**, 114:3003–3015.

[44] TESAŘ, P. ET AL. (2010): A new method for generating high non-linearity s-boxes **Radioengineering**, 19(1):23–26.

[45] YEH, Y., LIN, C., AND WANG, C. (2000): Dynamic gost. **Journal of Information Science and Engineering**, 16(6):857–861.