

## CA-based S-boxes for Secure Ciphers

Mirosław Szaban<sup>1</sup> and Franciszek Seredyński<sup>2,3</sup>

<sup>1</sup> Institute of Computer Science, University of Podlasie, Sienkiewicza 51, 08-110  
Siedlce, Poland

<sup>2</sup> Institute of Computer Science, Polish Academy of Sciences, Ordona 21, 01-237  
Warsaw, Poland

<sup>3</sup> Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008  
Warsaw, Poland

### Abstract

Block ciphers are widely used in modern cryptography. Substitution boxes (S-boxes) are main elements of these types of ciphers. In this paper we propose a new method to create S-boxes, which is based on application of Cellular Automata (CA). We present results of testing CA-based S-boxes. These results confirm that CA are able to realize efficiently a Boolean functions corresponding to classical S-boxes. Proposed CA-based S-boxes offer cryptographic properties comparable or better than classical S-box tables.

**Keywords:** Cellular Automata, S-Box, Block Cipher, Cryptography, Boolean Functions

### 1 Introduction

Cryptography plays an important role in security of data in the modern world. Two main cryptography systems are used today to provide a secure communication: secret and public-key systems. An extensive overview of currently known or emerging cryptography techniques used in both types of systems can be found in Schneier (1996). The main concern of this paper are cryptosystems with a secret key. The main interest of this work are Boolean functions used in S-boxes and applied in efficient algorithms in secret key systems. Many known secure standards of symmetric key cryptography use efficient and secure algorithms working on the base of S-boxes, such as e.g. Fips Pub 46-3 (1999), FIPS PUBS 197 (2001). S-boxes are ones of the most important components of block ciphers.

In the next section the concept of S-box and its most known applications are presented. Section 3 describes a few cryptographic criteria to examine Boolean functions realized with S-boxes. Section 4 presents the concept of CA. In section 5 the idea of substitution of S-boxes by CA is proposed. Section 6 presents results of examination of CA-based S-boxes, their quality measured by efficient criteria and comparison with earlier proposals. The last section concludes the paper.

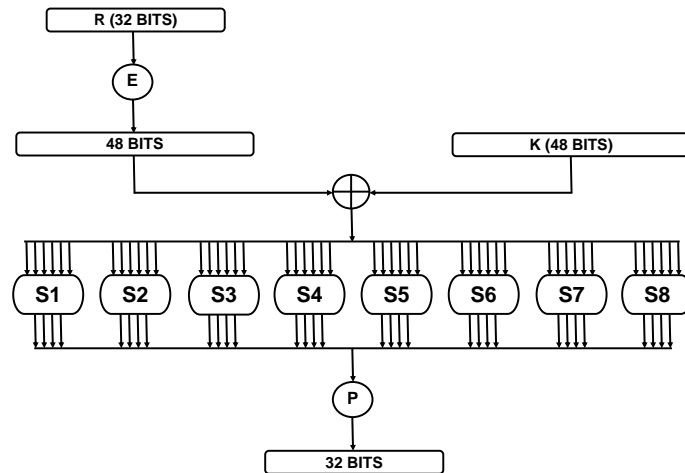


FIGURE 1: Schema presenting an application of S-boxes in DES algorithm (on the base (Fips Pub 46-3, 1999)).

## 2 S-boxes in Cryptography

S-box is a function  $f$ , which from each of  $n$  Boolean input values of  $B^n$  block consisting of  $n$  bits  $b_i$  ( $i \leq n$ ) generates some  $k$  Boolean output values called  $B^k$  block consisting of  $k$  bits  $b_j$  ( $j \leq k$  and  $k \leq n$ ):  $f : B^n \rightarrow B^k$ , what corresponds to the mapping  $(b_0, b_1, \dots, b_n) \rightarrow (b_0, b_1, \dots, b_k)$ . When  $n$  is equal to  $k$ , the function  $f$ , from  $n$  different input values maps  $n$  different outputs values, and such a S-box is called bijective (Fuller et al., 2004).

One of well known application of S-boxes is applying them in Data Encryption Standard (DES) as the "heart" of this algorithm (Fips Pub 46-3, 1999). In DES algorithm 64 input bits are changed by Initial Permutation. After that 64-bit block are transformed into two blocks of bits composed of 32 bits. One of this two blocks is block  $R$  (see, Figure 1). The next operation in the algorithm is operation named  $E$  and denoted as function which takes a block of 32 bits as input and yields a block of 48 bits as output. Function  $E$  takes 32 bits of block  $R$  and gives a new block composed of 48-bits. Operation  $\oplus$  denotes bit-by-bit addition modulo 2 and creates from block  $E(R)$  and 48-bits block of key  $K$  a new block of bits (see, Figure 1). In the next step, 48 bits  $(E(R) \oplus K)$  are cut into to eight blocks composed of 6 bits each which are sent to eight S-boxes. Reassuming, these eight wide known functions  $S1, \dots, S8$  collectively transform the 48 bit input block into 32 bit output block (see, Figure 1).

Each of the unique selection functions  $S1, \dots, S8$  are tables composed of 16-columns and 4-rows. Each function takes a 6-bit block as input and yields a 4-bit block as output.

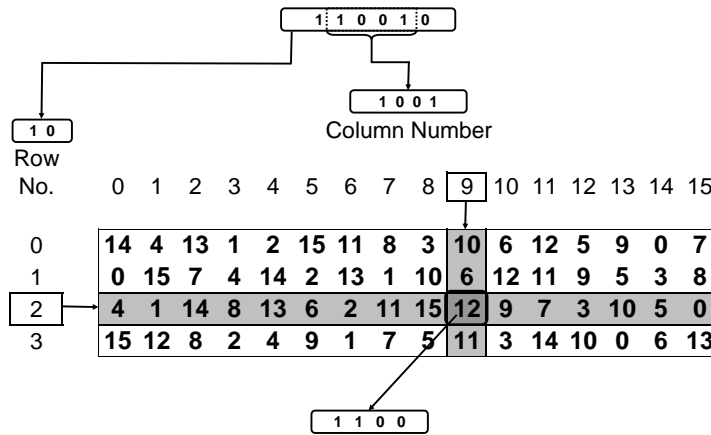


FIGURE 2: Function S-box S1 (in DES algorithm) represented as table and its work (on the base (Fips Pub 46-3, 1999)).

Let us consider the function  $S1$  represented in Figure 2 as table. Suppose that the input block of this function is the block  $B^6$ , e.g.  $110010$ . Two bits from  $B^6$ , the first and last one (e.g.  $10$ ) define row  $2$  of the  $S1$  block. Four middle bits  $1001$  define the column  $9$  of the  $S1$  block. Intersection of the column  $9$  and row  $2$  points the number  $12$ , e.g.  $1100$ , and these bits are considered as the  $B^4$  output block.

S-boxes are also used in modern symmetric key cryptography systems, e.g. in the new standard Advanced Encryption Standard (AES) (see, FIPS PUBS 197 (2001)). AES is successor of DES, and provides much better cryptographic quality than DES.

### 3 The Most Important Cryptographic Criteria for Boolean Functions

In our study we propose to use CA as a function which can be characterized by the same properties and realize the same functions as wide known S-boxes. A motivation for applying CA to realize S-boxes steams from potentially very interesting features of CA. On one side CA has a computational possibility equivalent to Universal Turing Machine (Wolfram, 2002), what means that such functions can be realized. What more, CA of a given size and governing rules (see, Section 4) can potentially represent not one but a number of S-box functions, what can simplify designing cryptography systems. The important issue is also efficiency of running cryptography systems. CA is a highly parallel system, easy in hardware implementation, what results in high efficiency of CA-based systems.

Quality of S-boxes designing with use of CA must be verified by required properties of S-boxes. Most important theorems for this purpose are recalled from cryptographic literature (Millan, 2005), (Clark et al., 2005), (Fuller et al., 2004), (Dawson et al., 2000).

A Boolean Function  $f : Z_2^n \rightarrow Z_2$  maps  $n$  binary inputs to a single binary output. Number of possible outputs is  $2^n$ . List of all possible outputs is the *truth table*. *Polarity* form of *truth table* is denoted by  $\hat{f}(x)$  and defined by:

$$\hat{f}(x) = (-1)^{f(x)}. \quad (1)$$

Boolean function is named linear function, when it can be expressed as an *XOR* of input variables. Let  $x = (x_1, x_2, \dots, x_n)$  be an input variables then linear function defined by  $\omega \in Z_2^n$  is expressed by equation:

$$L_\omega(x) = \omega_1 x_1 \otimes \omega_2 x_2 \otimes \dots \otimes \omega_n x_n, \quad (2)$$

where  $\omega_i x_i$  denotes *AND* operation on  $i$ -th bits of  $\omega$  and  $x$ , operation  $\otimes$  denotes *XOR* (exclusive *OR*) on bits. Set of *affine functions* is the set composed of linear functions and its complements.

Walsh Hadamard Transform  $\hat{F}_f(\omega)$  defines correlation between a function  $f$  and relevant linear function  $L_\omega(x)$ . That says how well the linear function approximates function  $f$ . Walsh Hadamard Transform is a product of polar forms  $f$  and  $L_\omega$  expressed by:

$$\hat{F}_f(\omega) = \sum_{x \in B^n} \hat{f}(x) \hat{L}_\omega(x). \quad (3)$$

Absolute maximum value in space of transforms is defined by:

$$WH_{max}(f) = \max_{\omega \in B^n} |\hat{F}_f(\omega)|. \quad (4)$$

The non-linearity  $N_f$  of Boolean Functions  $f$  is the minimum distance to the set of affine functions and is calculated as:

$$N_f = \frac{1}{2}(2^n - WH_{max}(f)). \quad (5)$$

The more higher is the non-linearity of observed ciphers ( $WH_{max}$  is low) the cipher is more difficult to cryptanalysis.

Another important property of stream ciphers is autocorrelation  $AC_f$ . Autocorrelation is similar to correlation, but polar form  $f(x)$  correlates with polar form  $f(x \otimes s)$  its *shifted version*. The Autocorrelation Transform of a Boolean function  $f$  is given by equation:

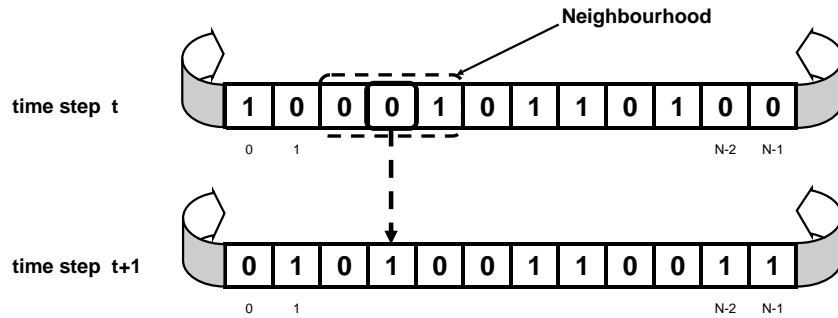
$$\hat{r}_f(s) = \sum_x \hat{f}(x) \hat{f}(x \otimes s), \quad (6)$$

where  $s \in Z_2^n - \{0\}$ . Absolute maximum value of any autocorrelations is denoted by equation:

$$AC_f = \max_{s \neq 0} \left| \sum_x \hat{f}(x) \hat{f}(x \otimes s) \right|. \quad (7)$$

The lowest is the autocorrelation of observed ciphers the cipher is more difficult to attacks.

### 1D Cellular Automata



### Rule of CA

Neighbourhood radius  $r=1$ , rule  $01011010_2 = 90_{10}$

Neighbourhood state	111	110	101	100	011	010	001	000
Rule	0	1	0	1	1	0	1	0

FIGURE 3: 1D Cellular automata with neighbourhood equal to 1.

## 4 The Concept of Cellular Automata

One dimensional (1D) CA is in the simplest case a collection of two-state elementary cells arranged in a lattice of the length  $N$ , and locally interacting in a discrete time  $t$ . For each cell  $i$  called a central cell, a neighbourhood of a radius  $r$  is defined, consisting of  $n_i = 2r + 1$  cells, including the cell  $i$ . When considering a finite size of CA, and a cyclic boundary condition is applied, it results in a circle grid (see, Figure 3).

It is assumed that a state  $q_i^{t+1}$  of a cell  $i$  at the time  $t+1$  depends only on states of its neighbourhood at the time  $t$ , i.e.  $q_i^{t+1} = f(q_i^t, q_{i1}^t, q_{i2}^t, \dots, q_{in}^t)$ , and a transition function  $f$ , called a rule, which defines a rule of updating a cell  $i$  (Figure 3). A length  $L$  of a rule and a number of neighbourhood states for a binary uniform CA is  $L = 2^n$ , where  $n = n_i$  is a number of cells of a given neighbourhood, and a number of such rules can be expressed as  $2^L$ . Figure 3 presents an example of the rule 01011010 (called also rule 90) for  $r = 1$ . The length  $L$  of the rule consists of 8 bits and is called a short rule. For CA with e.g.  $r = 2$  the length of a rule is equal to  $L = 32$ , and a number of such rules is  $2^{32}$  and grows very fast with  $L$ . CA for systems with a secret key were first studied by Wolfram (Wolfram, 1986). The author applied in his research one dimensional uniform CA. One dimensional uniform CA use only one rule as transition function, in opposite to one dimensional nonuniform CA, which use more than one rule to update cells of CA.

## 5 Cellular Automata and Constructing S-Boxes

A classic S-box is a function expressed as a table (composed of natural numbers). Cryptographic literature shows us many examples and methods of searching of S-box tables. Qualities of S-box are measured with use of different functions which examine its different properties. Some the most important testing functions are presented in section 3, also in (Millan, 1998), (Millan et al., 1999), (Clark et al., 2005). In these works, authors study the subject and use many different methods of searching through this huge space of S-box's tables. Recently the heuristic methods (see, (Millan et al., 1999), (Clark et al., 2005)) are widely used in discovering new tables. These methods give good results very fast. Each of these methods in the consequence still searches for the combinations of a numbers in table. We would like to propose another method without using a table as the base of S-box. In our approach CA is expected to perform the same tasks as S-box.

Major principle of S-box work is as follows. S-box from each of  $n$  input binary values generate some  $k$  output binary values. This condition will be satisfied by proposed CA, which from each of  $n$  input binary values generates some  $n$  output binary values. In this proposition creation of specific table is unnecessary, because the CA as a tool equivalent to Turing's Machine (see, (Wolfram, 2002)) can realize any function, in particular functions related to S-box.

We propose CA performing the role of S-box and implementing as a vector composed of:

- initial state of CA
- rule/rules applied to CA
- number of CA time steps
- input/output bits of CA-based S-box.

Selected cells of CA (in its initial state) serve as input bits of S-box, and the same cells, after declared time steps, are considered as the output of the S-box. To construct CA performing S-box function it is necessary to find appropriate CA rules and verify produced results according to S-box functions criteria.

## 6 Designing CA-based S-boxes and Their Analysis

### 6.1 Searching for CA Rules and Construction CA-based S-box

We start from examining uniform CA of the size 8 cells, with use of all 256 short rules ( $r = 1$ ). As a bijective S-box, CA with a size equal to 8 cells (an initial CA state corresponds to S-box input) is examined in time steps  $t$  (a CA state at this moment corresponds to S-box output) equal to 5, 6, 7, 8, 30, 50, 100 (see, Table 1). Non-linearity and autocorrelation values of all examined 256 CA rules were calculated to select the best CA rules. The best selected at this stage of experiments CA rules are as follows: 30, 57, 86, 99, 135, 149. These rules provide non-linearity and autocorrelation values higher then the other examined rules, and there scores are presented in Table 1.

Generally, higher value of  $N_f$  means that S-box provides higher quality related to non-linearity criterion, in opposite to autocorrelation of S-box in which higher quality of S-box corresponds to lower value of  $AC_f$ .

TABLE 1: CA rules, which with use of CA provide the best non-linearity  $N_f$  and autocorrelation  $AC_f$  in different time steps. CA size is equal to 8 cells, number of CA runs is equal to 100 for time steps from the set  $\{5, 6, 7, 8, 10, 30, 50, 100\}$ . Rules are selected from set of 256 rules with neighborhood radius  $r = 1$ .

Time steps	Best rules	$(N_f, AC_f)$
5	57, 99	(106, 64)
6	57, 99	(106, 56)
7	57, 99	(102, 64)
8	57, 99 30, 86, 135, 149	(104, 56) (101, 60)
10	57, 99 30, 86, 135, 149	(108, 64) (101, 68)
30	57, 99 30, 86, 135, 149	(108, 64) (106, 64)
50	57, 99 30, 86, 135, 149	(108, 64) (102, 72)
100	57, 99 30, 86, 135, 149	(104, 56) (104, 80)

The quality of results presented in Table 1 in terms of values of  $(N_f, AC_f)$  appears to be comparable with the quality of classic S-boxes presented in (Millan, 1998), (Millan et al., 1999), (Clark et al., 2005). The best (worst) theoretical values of non-linearity and autocorrelation corresponding to this 8 bit CA-based S-Box are equal to 128 (0) for non-linearity and 0 (256) for autocorrelation, respectively. Scores of the best CA rules presented in Table 1 for non-linearity are changing in the range [101, 108] and for autocorrelation in the range [56, 80]. Values of non-linearity and autocorrelation obtained in experiments can be successfully compared with results (lower or similar) presented by Millan (1998), Millan et al. (1999) and Clark et al. (2005). It can be concluded that behavior of CA which implements S-box provides a good quality.

During all conducted experiments presented in this paper we assume that we have to do with CA-based S-box with 8 inputs and 8 outputs, despite the fact that a size of CA will be larger than 8. The next step of the research was a verification of CA rules quality from the point of view of bijectivity. When the CA size is equal to 8 input/output cells then diversity of outputs obtained from each of possible inputs is lower than 20%. Therefore, CA-based S-box was examined with a number of cells ranging from 8 to 500, because larger CA size provide higher diversity. During experiments the problem of allocation of 8 examined input/output bits in larger CA arises. An initial state of large CA was randomly selected, but input/output bits of CA-based S-box need to be determined. These 8 bits (main bits) are always located in one block (first bits in CA – for simplicity) as a part of CA's state. Other bits of large CA forms background, environment for evolution of block of main bits.

Another idea on how to arrange bits in the background is to locate bits separately in CA cells. In the series of tests main bits were located in CA cells in the distance 1, 2, 5 and 10 cells from each other. The remaining CA cells were

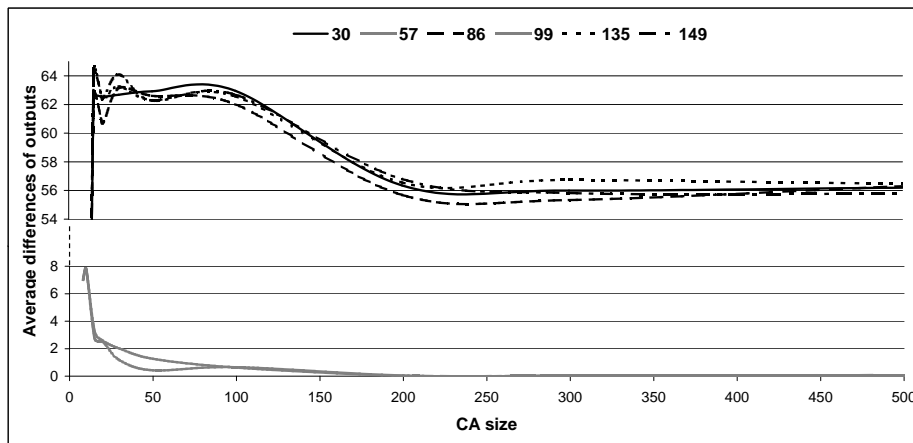


FIGURE 4: Diversification of outputs (in %) given by CA (after 100 time steps) with rules: 30, 57, 86, 99, 135, 149 for CA's sizes in the range [8, 500] averaged over 1000 CA runs.

set randomly. Results of experiments (not presented in this paper) show that in fact the way of arranging main bits in the background has small influence on diversification of outputs – the difference is of the range of 2%.

Figure 4 present results of the verification in bijectivity condition. One can see that for rules 57 and 99 (see, Figure 4) the maximal number of different outputs of CA-based S-box is equal to only 8% values out of 256 values. It means that these rules are not suitable for the purpose of S-box. Therefore, in our next experiments these rules can not be taken into account. Figure 4 shows results for rules 30, 86, 135 and 149. One can see that results for these rules are much better than for the previous ones: the maximal number of different outputs is equal to about 63% and this feature concerns a relatively wide size of CA up to 100 cells. These results seem to be promising and therefore for the next study we focused on CA with 100 cells. Reassuring, in all our next experiments CA size will be equal to 100, but number of main bits (input/output of CA-based S-box) will be equal to 8. Large CA (100 cells) will be evaluate in discreet time steps (100 time steps), but we will be observe and examine only main bits (8 input/output bits) in large CA.

The last conducted experiments concerned examination of CA rules which passed both verification procedures. During two previous experiments rules 30, 86, 135 and 149 was selected as the best rules from the set of CA rules with neighborhood radius equal to 1. For these CA rules the first experiment was conducted with new CA size, i.e. values of non-linearity and autocorrelation were calculated.

## 6.2 Comparison of CA-based S-box with Classical S-boxes

In the 1000 CA runs for random background of bits for CA size equal to 100 cells, main bits were arranged in one block. In each runs CA was running 100 time steps. The results of experiment are presented in Table 2.

TABLE 2: The best non-linearity  $N_f$  and autocorrelation  $AC_f$  obtained with use of CA size equal to 100 cells, in 1000 runs for 100 time steps. Rules {30, 86, 135, 149} are selected from set of 256 rules with neighborhood radius  $r = 1$ .

Rule	$(N_f, AC_f)$	The highest quality $(N_f, AC_f)$	The lowest quality $(N_f, AC_f)$
30	(90, 112)	(105, 44), <b>(110, 48)</b> , (108, 48)	(98, 112), (106, 104), (90, 88)
86	(91, 96)	<b>(108, 48)</b> , (106, 48), (109, 52)	(104, 96), (93, 92), (91, 76)
135	(91, 104)	<b>(108, 48)</b> , (106, 48), (109, 52)	(96, 104), (103, 100), (91, 84)
149	(91, 100)	<b>(109, 44)</b> , (108, 48), (106, 48)	(99, 100), (107, 100), (91, 76)

The table presents for each rule the following results: minimal guaranteed values of non-linearity ( $N_f$ ) and autocorrelation ( $AC_f$ ), three selected the highest quality values of  $(N_f, AC_f)$  observed in each set of CA runs, and three selected the lowest quality  $(N_f, AC_f)$  observed in each set of CA runs. Guaranteed values presented in Table 2 mean that in all CA runs we could not find the lower value for  $N_f$  is not less than 90, and the higher value for  $AC_f$  is not greater than 112. The highest and lowest quality presented in 3-rd and 4-h column of Table 2 correspond to single runs, selected from all set of 1000 CA runs. One can see that the highest quality of non-linearity is equal to 110 and corresponds to the rule 30. Low level of autocorrelation for this rule is also provided. On the other hand the rule 149 is characterized by the best value of autocorrelation equal to 44, with high level of non-linearity.

In (Millan, 1998) and (Clark et al., 2005) for the same range of S-box, S-boxes were found with values of  $N_f$  ranging from [80, 100] and [90, 100], respectively. The best autocorrelation values  $AC_f$  presented in (Millan, 1998) and (Clark et al., 2005) are equal {98, 100} and {80, 102}, respectively. If we compare these results with our results we can conclude that (a) even our guaranteed values of  $(N_f, AC_f)$  are comparable with their results, and (b) our best results are better than pointed in their study.

Our guaranteed values of  $N_f$  equal to {90, 91} for appropriate CA rules includes in ranges of non-linearity values presented in (Millan, 1998) and (Clark et al., 2005). Similarly our guaranteed values of  $AC_f$  equal to {96, 100, 104, 112} for appropriate CA rules are comparable with their best results.

On the other hand our the best results presented in 3-rd column of Table 2 characterizing by much higher values of non-linearity  $N_f$  (110, 108, 108 and 109 for appropriate CA rules) are better than the value equal to 100 found in (Millan, 1998) and (Clark et al., 2005). Also, the best results of autocorrelation  $AC_f$  presented in 3-rd column of Table 2 characterizing by much lower values (48, 48, 48 and 44 for appropriate CA rules) are better than values equal to 98 and 80, presented in (Millan, 1998) and (Clark et al., 2005), respectively.

In our all of experiments CA-based S-box gives different results in single runs (better or worst). Despite these results, important property is frequency of distribution in calculated results. Figure 5 and Figure 6 show frequency (in %) of distribution for obtained non-linearities and autocorrelations in 1000 CA runs, respectively. One can see that most of obtained results for  $N_f$  ranging from [98, 108] and from [52, 84] for  $AC_f$ . These ranges of  $N_f$  values keep better quality

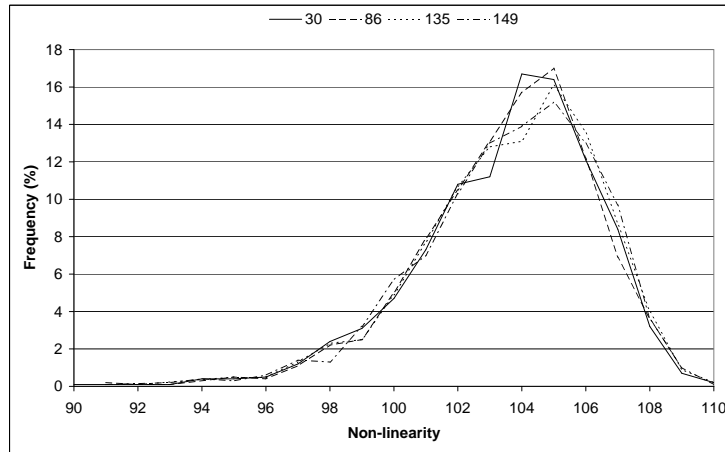


FIGURE 5: Frequency (in %) of distribution of non-linearity for tested CA with rules: 30, 86, 135, and 149. Number of CA runs is equal to 1000, CA size and CA time steps are equal to 100.

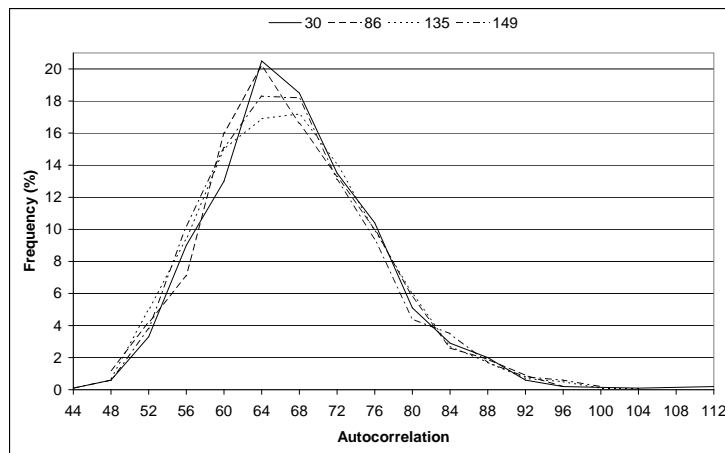


FIGURE 6: Frequency (in %) of distribution of autocorrelation for tested CA with rules: 30, 86, 135, and 149. Number of CA runs is equal to 1000, CA size and CA time steps are equal to 100.

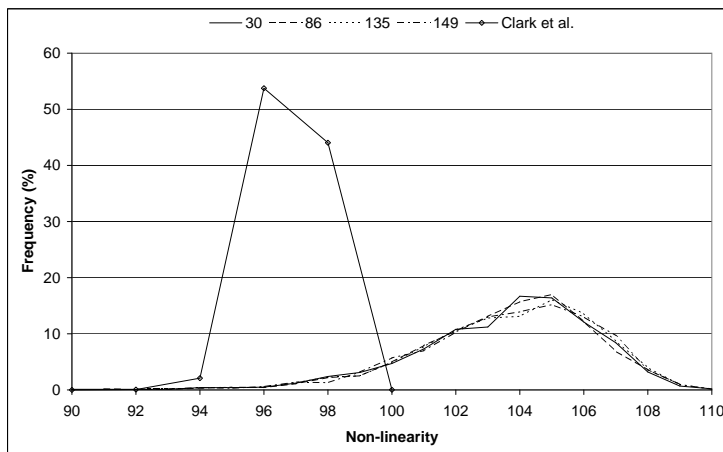


FIGURE 7: Frequency (in %) of distribution for non-linearity in our CA-based S-boxes with rules: 30, 86, 135, 149 and Clark et al. results (Clark et al., 2005).

than ranges  $[80, 100]$  and  $[90, 100]$  for results of non-linearity presented in (Millan, 1998) and (Clark et al., 2005). Also, the most of our results concerning autocorrelation  $AC_f$  (ranging from  $[44, 112]$ ) are better than results 98 and 80 presented in (Millan, 1998) and (Clark et al., 2005), respectively.

Frequency of distribution in our results and results given by Clark et al. (2005) are presented in Figure 7. One can see that in Clark results most S-boxes give their values of non-linearity from the range  $[96, 98]$  and the best S-boxes give non-linearity equal to 100. Our results provide most of CA-based S-boxes with non-linearity from the range  $[98, 108]$  and the best CA-based S-boxes gives values even equal to 110.

Lets observe non bijective S-boxes. These  $8 \times m$  S-boxes, from 8 inputs provide  $m$  outputs ( $m = 2, 3, 4, 5, 6, 7$ ). When we observe how quality changes for not bijective S-boxes, proposed in literature (Millan, 1998), (Millan et al., 1999) and (Clark et al., 2005), we can conclude that exists a relationship between values of non-linearity and autocorrelation of the best S-boxes and a number of output bits. If a number of output bits grows then quality of S-boxes goes down (i.e., value of non-linearity goes down and value of autocorrelation grows). For these observations we can conclude that non-linearity and autocorrelation of our proposed CA, which realize S-box functions provide higher values than some values obtained in (Millan, 1998), (Millan et al., 1999), (Clark et al., 2005), also for  $8 \times m$  S-boxes. Our best CAs (see, Table 2) keeps higher quality (higher non-linearity, lower autocorrelation) than result (108, 56) presented in (Clark et al., 2005) for  $8 \times 5$  S-boxes.

## 7 Conclusions and Future Work

The paper presents a new idea of creating S-boxes using CA approach. Applying CA to create S-boxes eliminates inefficient tables which are used in classical approach. CA from input block of bits generates output block of bits and this output is evaluated by the same examine criteria like the traditional S-box. Obtained preliminary results are very promising. Conducted experiments have shown that CA-based S-box is characterized by a high non-linearity and low autocorrelation. These values correspond to values related to classical S-boxes or outperform them. The open issue is the questions of: (1) enlarging the maximal value of the number of possible output values of CA-based S-box, (2) using reversible CA for creating S-boxes and (3) designing the dynamical S-boxes. This issue is the subject of a current research.

## References

- J. A. CLARK, J. L. JACOB, S. STEPNEY (2005), The Design of S-Boxes by Simulated Annealing, *New Generation Computing*, Vol. 23, No. 3, Ohmsha and Springer, 219 - 231
- E. DOWSON, W. MILLAN, L. SIMPSON (2000), Designing Boolean Functions for Cryptographic Applications, *Contributions to General Algebra 12*, Verlag Johannes Heyn, Klagenfurt, 1-22
- FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION (1999), Fips Pub 46-3, Reaffirmed October 25, <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATIONS (FIPS PUBS) 197 (2001), AES, November 26, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- J. FULLER, W. MILLAN, E. DAWSON (2004), Multi-objective Optimisation of Bijective S-boxes, *Proceedings of CEC'04*, Portland, OR.
- W. MILLAN (1998), How to Improve the Non-linearity of Bijective S-boxes, *Lecture Notes in Computer Science*, Volume 143, Springer-Verlag 181-192
- W. MILLAN, L. BURNETT, G. CARTER, A. CLARK, E. DAWSON (1999), Evolutionary Heuristics for Finding Cryptographically Strong S-Boxes, ICICS'99
- W. MILLAN (2005), New Cryptographic Applications of Boolean Function Equivalence Classes, *Lecture Notes in Computer Science*, Vol. 3574. Springer-Verlag, 572-583
- B. SCHNEIER (1996), *Applied Cryptography*, Wiley, New York
- S. WOLFRAM (1986), Cryptography with Cellular Automata, *Crypto '85 Proceedings, Lecture Notes in Computer Science*, Volume 218, Springer, 429-432
- S. WOLFRAM (2002), *A New Kind of Science*, Wolfram Media Inc., Champaign, Illinois