Annales UMCS Informatica AI VIII, 2 (2008) 27–41
DOI: 10.2478/v10065-008-0022-4

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Designing cryptographically strong S–boxes with the use of cellular automata

Mirosław Szaban[1]*, Franciszek Seredynski[2,3]

[1] *The University of Podlasie, Computer Science Department, Sienkiewicza 51, 08-110 Siedlce, Poland*
[2] *Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland*
[3] *Institute of Computer Science, Polish Academy of Sciences, Ordona 21, 01-237 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 the results of testing CA-based S–boxes. These results confirm that CA are able to realize efficiently the Boolean function corresponding to classical S–boxes the proposed CA-based S–boxes offer cryptographic properties comparable or better than classical S–box tables.

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 [**1**]. The main concern of this paper are cryptosystems with

*Corresponding author: *e-mail address:* mszaban@ap.siedlce.pl a secret key. The main interest of this work are the 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 basis of S–boxes, such as e.g. FIPS PUB [**2**], FIPS PUBS [**3**]. S-boxes are 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 the 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 the results of examination of CA-based S–boxes, their quality measured by efficient criteria and comparison with the earlier proposals. The last section concludes the paper.

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 $k \leq n)$:

$$f : B^n \to B^k,$$

which corresponds to the mapping $(b_0, b_1, \ldots, b_n) \to (b_0, b_1, \ldots, b_k)$. When n is equal to k, the function f, from n different input values maps n different outputs values, and S-box is called bijective, Fuller et al. [**4**].

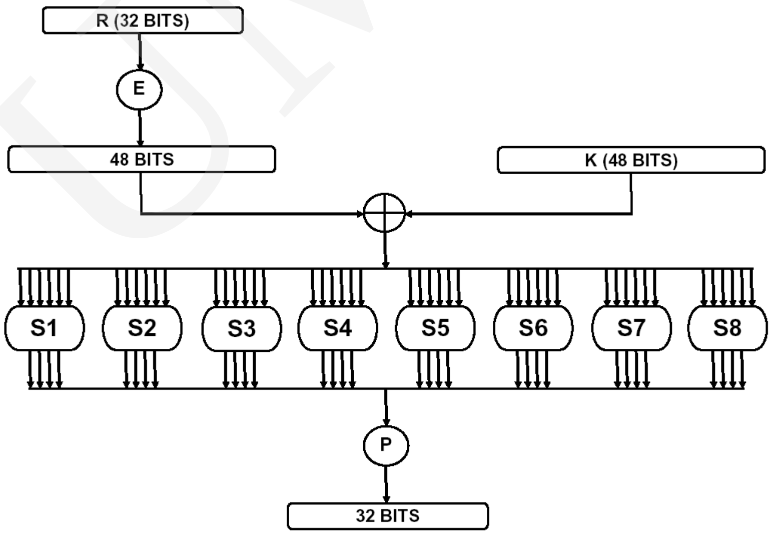
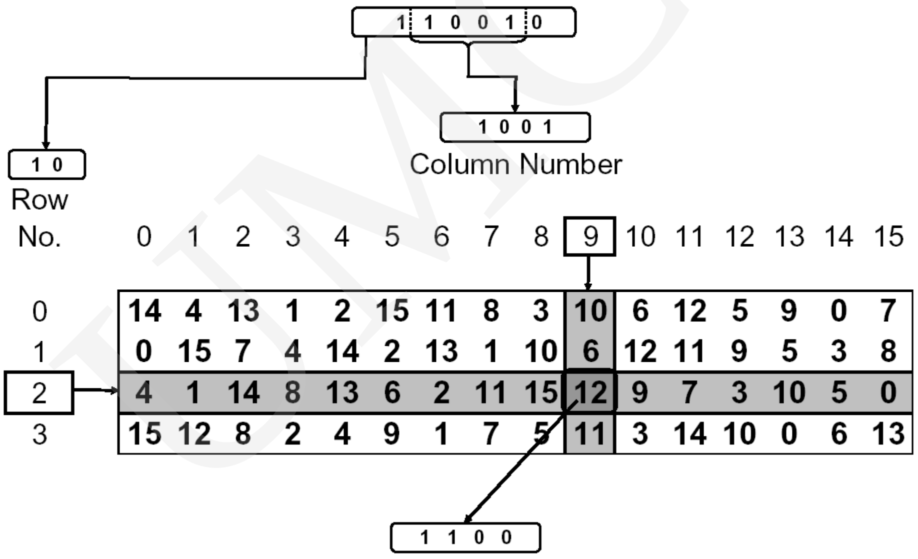


Fig. 1. Scheme presenting an application of S–boxes in the DES algorithm (on the basis of FIPS PUB [**2**])

One of the well known applications of S–boxes is their use in Data Encryption Standard (DES) as the "heart" of this algorithm [**2**]. In the DES algorithm 64 input bits are changed by Initial Permutation. After that 64–bit blocks are transformed into two blocks of bits composed of 32 bits. One of these two blocks is block R (see Fig. 1). The next operation in the algorithm is named E and denoted as a 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 the addition of modulo 2 and creates from block $E(R)$ and 48–bit block of key K a new block of bits (see Fig. 1). In the next step, 48 bits $(E(R) \oplus K)$ are cut into eight blocks composed of 6 bits each which are sent to eight S–boxes. Reassuming, these eight widely known functions $S1, \ldots, S8$ collectively transform the 48–bit input block into 32–bit output block (see Fig. 1).

Each of the unique selection functions $S1, \ldots, S8$ are the 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.



Row No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Fig. 2. Function S–box $S1$ (in the DES algorithm) represented as a table and its work (on the basis of FIPS PUB [**2**])

Let us consider the function $S1$ represented in Figure 2 as a 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 column 9 of the $S1$ block. The intersection of 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 [**3**]). AES is a successor of DES, and provides much better cryptographic quality than DES.

3. Most Important Cryptographic Criteria for the 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 widely known S–boxes. A motivation for applying CA to realize S–boxes stems from potentially very interesting features of CA. On one hand, CA has a computational possibility equivalent to Universal Turing Machine (see Wolfram [**5**]), which means that such functions can be realized. What is more, CA of a given size and governing rules (see Section 4) can potentially represent not one but a number of S–box functions, which 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, which results in high efficiency of CA–based systems.

Quality of S–boxes designed with the use of CA must be verified by required properties of S–boxes. The most important theorems for this purpose are recalled from cryptographic literature (see Millan [**6**], Clark et al. [**7**], Fuller and Millan [**4**], Dawson and Millan [**8**]).

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

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

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

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

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

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

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

The absolute maximum value in space of transforms is defined by:

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

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

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

The higher the non–linearity of the observed ciphers is (WH_max is low), the more difficult the cipher is for cryptanalysis.

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

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

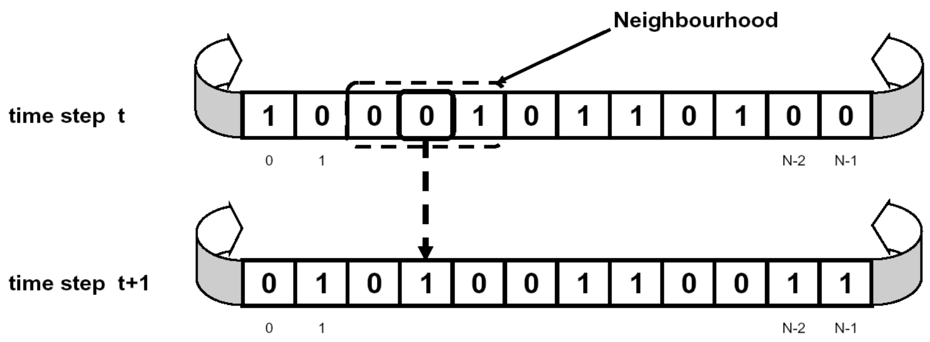
where $s \in Z_2^n \backslash \{0\}$. The absolute maximum value of any autocorrelation is denoted by the equation:

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

The lower the autocorrelation of the observed ciphers is, the more difficult for attacks the cipher is.

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 Fig. 3).



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

Fig. 3. 1D Cellular automata with the neighbourhood equal to 1

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. $q_i^{t+1} = f(q_i^t, q_{i1}^t, q_{i2}^t, \ldots, q_{in}^t)$, and a transition function f, called a rule, which defines a rule of updating a cell i (see Fig. 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. Fig. 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 the systems with a secrete key were first studied by Wolfram [**9**]. The author applied in his research one dimensional uniform CA. One dimensional uniform CA use only one rule as a transition function, contrary to one–dimensional nonuniform CA, which use more than one rule to update the 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 for S–box tables. Qualities of S–box are measured with the use of different functions which examine its different properties (see Millan [**10**], Millan et al. [**11**], Clark et al. [**7**], Nedjah and de Macendo Mourelle [**12**]). Some of the most important testing functions are presented in section 3. In [**10**], [**11**], [**7**] and [**12**] the 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. [**11**], Clark et al. [**7**]) 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 numbers in the 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.

The major principle of S–box work is as follows. S–box from each of n input binary values generates some k output binary values. This condition will be satisfied by the 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 [**5**]) can realize any function, in particular functions related to S–box.

We propose CA performing the role of S–box as a vector composed of:
– initial state of CA
– rule/rules applied to CA
– number of CA time steps
– selected cells of CA in which input bits of S-box (in the initial state of CA) are placed, and the same cells are considered as the output of the S–box (after the declared time steps).

To construct CA performing S–box function, it is necessary to find appropriate CA rules and verify produced results according to the S–box functions criteria.

6. Designing CA-based S–boxes and Their Analysis

6.1. Searching for CA Rules and Construction of CA–based S–box.

We start with examining uniform CA of the size 8 cells, with the 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 (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 CA rules selected at this stage of experiments are as follows: 30, 57, 86, 99, 135 and 149. These rules provide non-linearity and autocorrelation values higher than the other examined rules, and their scores are presented in Table 1.

Generally, a higher value of N_f means that S–box provides higher quality related to the non-linearity criterion, contrary 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 the 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 experiments is equal to 100 for each time step. Rules are selected from the set of rules with the neighborhood radius $r = 1$

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

The quality of the results presented in Table 1 in terms of the values of (N_f, AC_f) appears to be comparable with the quality of classic S–boxes presented in [**10**], [**11**], [**7**]. 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. The 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]. The values of non–linearity and autocorrelation obtained in experiments can be successfully compared with the results (lower or similar) presented by Millan [**10**], Millan et al. [**11**] and Clark [**7**]. It can be concluded that behaviour 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 sizes 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 form background, the environment for evolution of a block of main bits.

Another idea 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 set randomly. The results of the 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%.

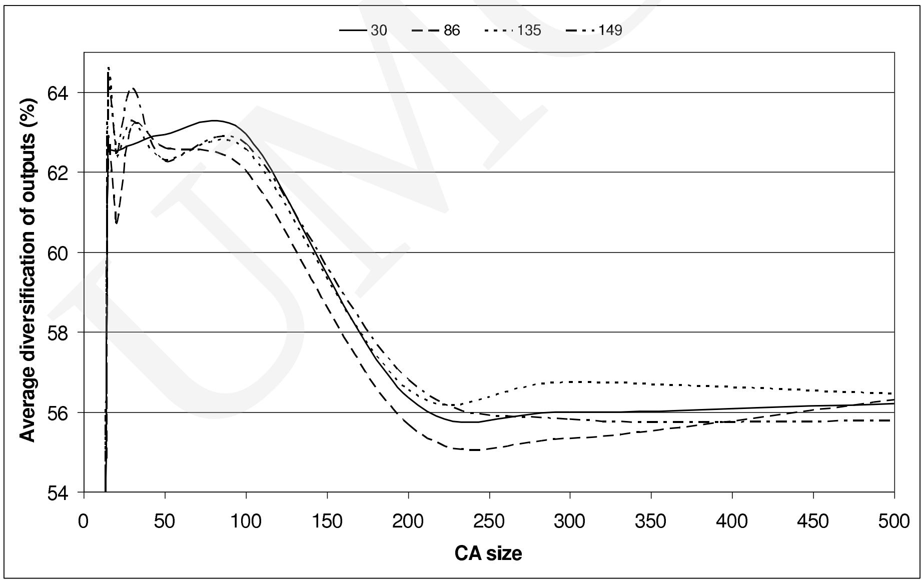


Fig. 4. 1D average diversification of outputs (in %) given by CA (after 100 time steps) with rules: 57, 99 for CA's sizes in the range [8, 500] of 1000 experiments

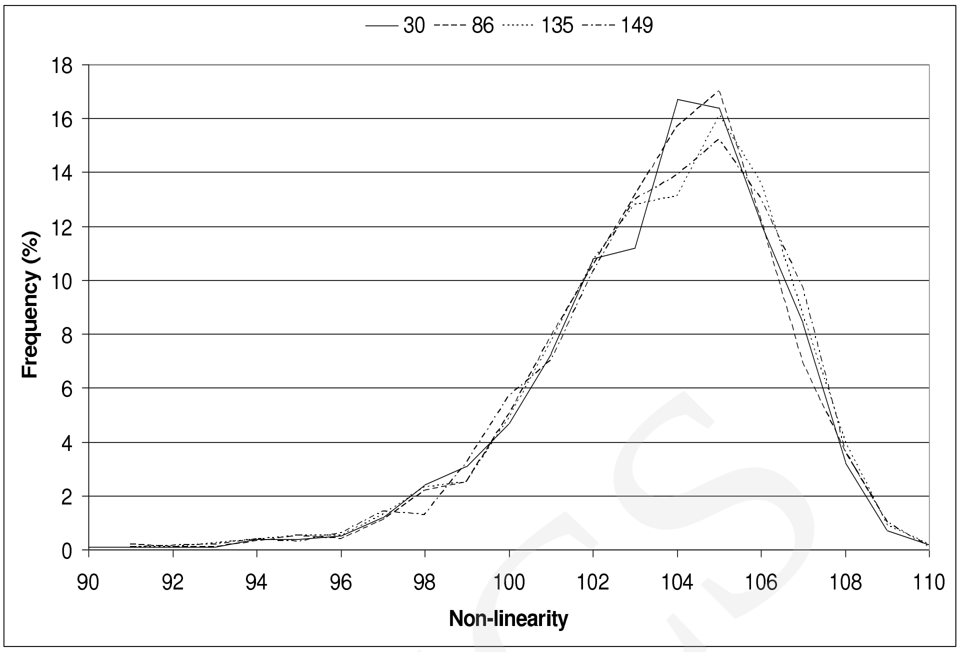


Fig. 5. 1D average diversification of outputs (in %) given by CA (after 100 time steps) with rules: 30, 86, 135, 149 for CA's sizes in the range [8, 500] of 1000 experiments

Figures 4 and 5 present the results of verification in bijectivity condition. One can see that for rules 57 and 99 (see Fig. 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. Fig. 5 shows the results for rules 30, 86, 135 and 149. One can see that the 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. Reassuming, in all our next experiments CA size will be equal to 100, but the number of main bits (input/output of CA–based S–box) will be equal to 8. Large CA (100 cells) will be evaluated in discreet time steps (100 time steps), but only main bits (8 input/output bits) in large CA will be observed and examined.

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

6.2. Comparison of CA–based S–box with Classical S–boxes.

In 1000 experiments for random background of bits in CA with 100 cells, main bits were arranged in one block. In each experiment CA was run 100 time steps. The results of the experiments are presented in Table 2.

Table 2. The best non–linearity N_f and autocorrelation AC_f obtained with the use of CA with the CA size equal to 8 cells, in 1000 experiments for each time step. The rules are selected from the set of rules with the 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), (110, 48), (108, 48)	(98, 112), (106, 104), (90, 88)
86	(91, 96)	(108, 48), (106, 48), (109, 52)	(104, 96), (93, 92), (91, 76)
135	(91, 104)	(108, 48), (106, 48), (109, 52)	(96, 104), (103, 100), (91, 84)
149	(91, 100)	(109, 44), (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 highest quality values of (N_f, AC_f) observed in each set of experiments, and three lowest quality (N_f, AC_f) observed in each set of experiments. The guaranteed values presented in Table 2 mean that in all experiments we could not find the value for N_f less than 90, and the value for AC_f not greater than 112. The highest and lowest qualities presented in the 3-rd and 4-h columns of Table 2 correspond to single experiments, selected from all sets of 1000 experiments. One can see that the highest quality of non-linearity is equal to 110 and corresponds to rule 30. Low level of autocorrelation for this rule is also provided. On the other hand, rule 149 is characterized by the best value of autocorrelation equal to 44, with high level of non–linearity.

In [**10**], [**7**] for the same range of S–box, S–boxes were found with the values of N_f ranging from [80, 100] and [90, 100], respectively. The best autocorrelation values AC_f presented in [**10**], [**7**] 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 those pointed in their study.

Our guaranteed values of N_f equal to $\{90, 91\}$ for the appropriate CA rules are included in the ranges of non–linearity values presented in [**10**], [**7**]. Similarly, our guaranteed values of AC_f equal to $\{96, 100, 104, 112\}$ for the appropriate CA rules are comparable with their best results.

On the other hand, our best results presented in the 3-rd column of Table 2 characterized by much higher values of non–linearity N_f (110, 108, 108 and 109 for the appropriate CA rules) are better than the value equal to 100 found in [**10**], [**7**]. Also, the best results of autocorrelation AC_f presented in the 3-rd column of Table 2 characterized by much lower values (48, 48, 48 and 44 for the appropriate CA rules) are better than those equal to 98 and 80, presented in [**10**], [**7**], respectively.

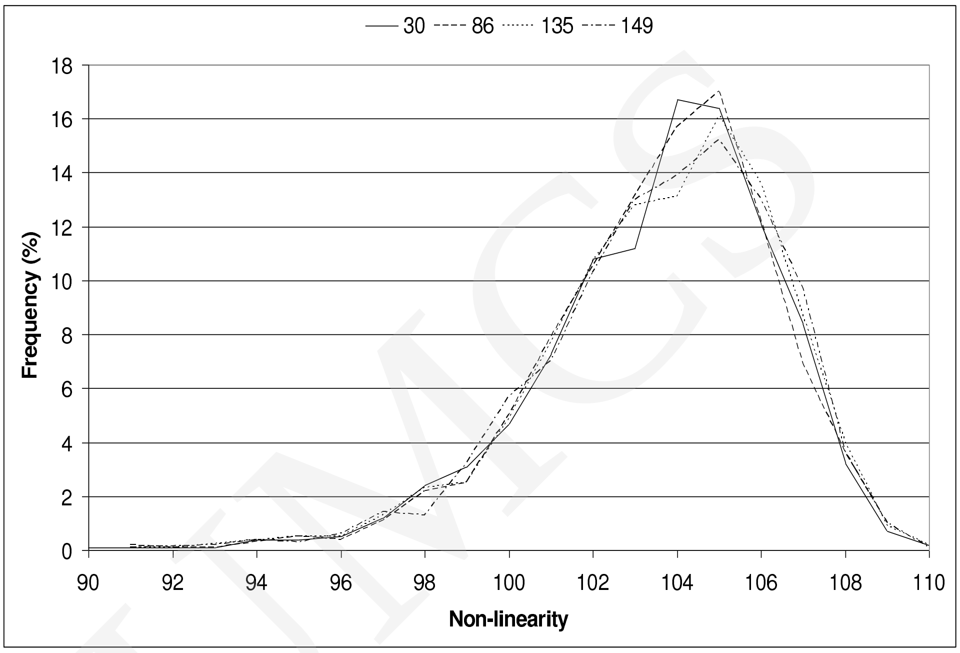


Fig. 6. Frequency (in %) of distribution of non–linearity for the tested CA with rules: 30, 86, 135, and 149. The number of experiments is equal to 1000, CA size and CA time steps are equal to 100

In our all experiments CA–based S–box gives different results in single experiments (better or worse). Despite these results, important property is frequency of distribution in the calculated results. Figs 6 and 7 show frequency (in %) of distribution for the obtained non–linearity and autocorrelations in 1000 experiments, respectively. One can see that most of the obtained results for N_f range in [98, 108] and [52, 84] for AC_f. These ranges of N_f values keep better quality than ranges [80, 100] and [90, 100] for the results of non–linearity presented in [**10**], [**7**]. Also, most of our results concerning autocorrelation AC_f (ranging in [44, 112]) are better than results 98 and 80 presented in [**10**], [**7**], respectively.

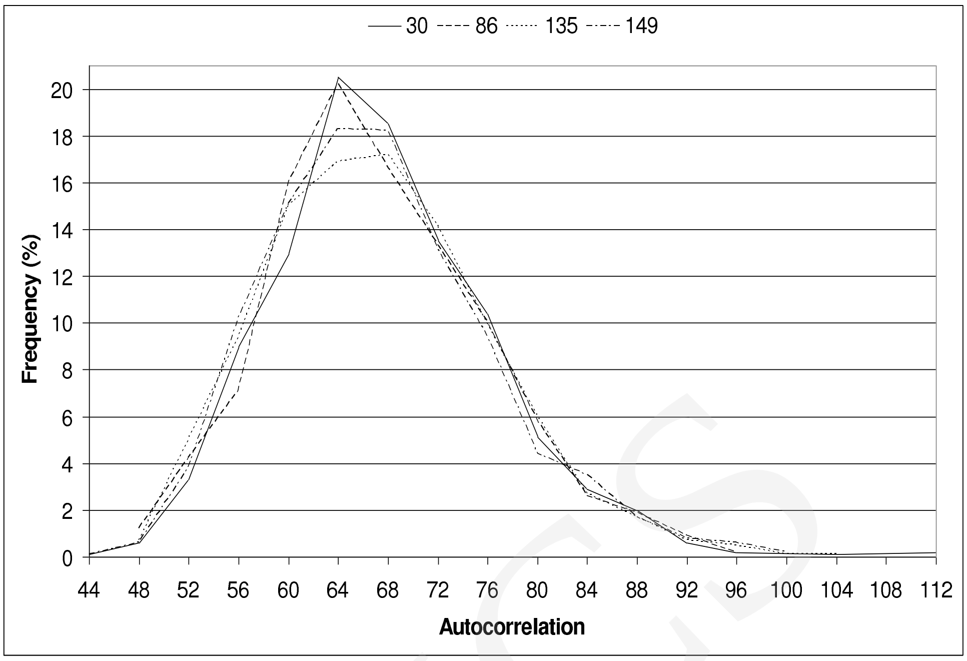


Fig. 7. Frequency (in %) of distribution of autocorrelation for the tested CA with rules: 30, 86, 135, and 149. The number of experiments is equal to 1000, CA size and CA time steps are equal to 100

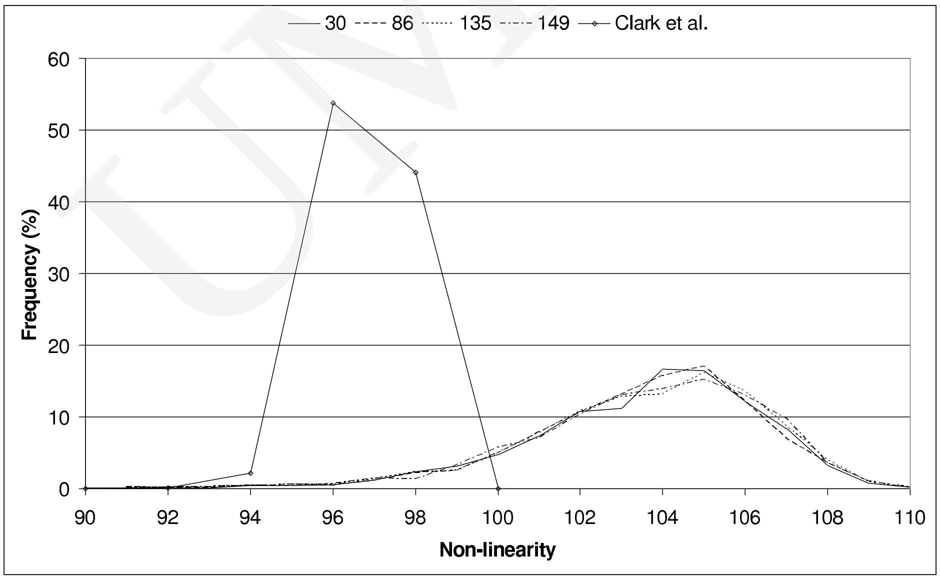


Fig. 8. Frequency (in %) of distribution for non-linearity in our CA-based S-boxes with rules: 30, 86, 135, 149 and Clark et al. results [**7**]

Frequency of distribution in our results and those given by Clark et al. [**7**] are presented in Fig. 8. One can see that in the Clark's results most S–boxes give their values of non–linearity in 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 in the range [98, 108] and the best CA–based S–boxes give values even equal to 110.

Let us 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 ([**10**], [**11**], [**7**] and [**12**]), we can conclude that there exists a relationship between the values of non–linearity and the 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., the value of non–linearity goes down and the value of autocorrelation grows). For these observations we can conclude that non–linearity and autocorrelation of our proposed CA, which realize the S–box functions provide higher values than some values obtained in [**10**], [**11**], [**7**] and [**12**], also for $8 \times m$ S–boxes. Our best CA_s (see Table 2) maintains higher quality (higher non–linearity, lower autocorrelation) than the result (108, 56) presented in [**7**] and the result (110,56) presented in [**12**] for 8×5 S–boxes.

7. Conclusions and Future Work

The paper presents a new idea of creating S–boxes using the CA approach. Applying CA to create S–boxes eliminates inefficient tables which are used in the classical approach. CA from the input block of bits generates the output block of bits and this output is evaluated by the same examination criteria as the traditional S–box. The obtained preliminary results are very promising. The experiments have shown that the CA–based S–box is characterized by a high non–linearity and low autocorrelation. These values correspond to those related to the classical S–boxes or outperform them. The open issue is the question of enlarging the maximal value of the number of possible output values of the CA–based S–box. This issue is the subject of current research.

References

[1] Schneier B., *Applied Cryptography*, Wiley, New York (1996).
[2] Federal Information Processing Standards Publication, Fips Pub 46-3, Reaffirmed (1999) October 25, http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
[3] Federal Information Processing Standards Publications (FIPS PUBS) 197, AES, (2001) November 26, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[4] Fuller J., Millan W., Dawson E., *Multi-objective Optimisation of Bijective S–boxes*, Proceedings of CEC'04. Portland, OR. (2004).

[5] Wolfram S., *A New Kind of Science*, Wolfram Media Inc., Champaign, Illinois, (2002).

[6] Millan W., *New Cryptographic Applications of Boolean Function Equivalence Classes*, Lecture Notes in Computer Science 3574 (2005) 572.

[7] Clark J. A., Jacob J. L., Stepney S., *The Design of S-Boxes by Simulated Annealing*, New Generation Computing 23(3) (2005) 219.

[8] Dowson E., Millan W., Simpson L., *Designing Boolean Functions for Cryptographic Applications*, Contributions to General Algebra 12 (2000) 1.

[9] Wolfram S., *Cryptography with Cellular Automata*, Crypto '85 Proceedings, Lecture Notes in Computer Science 218 (1986) 429.

[10] Millan W., *How to Improve the Non-linearity of Bijective S-boxes*, Lecture Notes in Computer Science 143 (1998) 181.

[11] Millan W., Burnett L., Carter G., Clark A., Dawson E., *Evolutionary Heuristics for Finding Cryptographically Strong S–Boxes*, ICICS'99 (1999).

[12] Nedjah N., de Macendo Mourelle L., *Designing Substitution Boxes for Secure Ciphers*, International Journal Innovative Computing and Application 1(1) (2007) 86.