

# HOW TO IMPROVE THE SECURITY OF THE CRYPTOSYSTEM BASED ON A BLOCK CIPHER?

Piotr Mroczkowski, Przemysław Rodwald  
Military Communication Institute  
Cryptology Department  
05-130 Żegrze, Poland  
{p.mroczkowski, p.rodwald}@wil.waw.pl

## ABSTRACT

*The article proposes a general framework for improving the security of the cryptosystem based on the symmetric block cipher. The main idea is based on possibility of changing S-boxes in encryption/decryption algorithm. In order to make it possible, it is necessary to generate identical substitution boxes by an encryption and decryption party. This is the main reason, why a deterministic methods of generating S-boxes based on the pseudorandom sequences is proposed. In the article the possibilities of generating S-boxes using pseudorandom sequences will be presented.*

## 1. INTRODUCTION

The confidentiality of the information, transmitted via contemporary telecommunications systems, is ensured by implementation of a cryptosystem in cryptographic devices. A cryptosystem is defined as a quintuple  $(PT, CT, K, E_k, D_k)$ , where:  $PT$  – plaintext,  $CT$  – ciphertext,  $K$  – key space,  $E_k$  – encryption algorithm,  $D_k$  – decryption algorithm. As an encryption/decryption algorithm the block cipher can be used. A lot of modern block ciphers are built using Shannon's concept which uses two basic transformations: confusion and diffusion. Such product cipher use S-boxes (substitution boxes) that provide confusions and P-boxes (permutation boxes) that provide diffusions and spread out the output bits to different S-boxes of the next round. Simple substitution and transposition ciphers individually do not provide a very

high level of security. However, by combining these transformations it is possible to obtain strong product cipher which guarantees resistance against linear and differential cryptanalysis. The strength of such cipher mainly comes from the property of S-boxes. It is well known that weaknesses of substitution boxes may be compensated for by the increased number of rounds. Another way to improve security of such cryptosystem is possibility of changing S-boxes in the product cipher.

The S-box  $S: Z_2^n \rightarrow Z_2^m$  is a nonlinear transformation, which transforms the  $n$ -binary sequence to the  $m$ -binary sequence. They can be constant (generated at the stage of specification and used during the lifetime of algorithm, e.g. DES, AES) or variable (generated before a session on the basis of data, which are available to the encryption/decryption party and used during the session). The idea of improving the security of cryptosystem built of the block cipher, based on the replacement S-boxes by new generated S-boxes. This causes, that for the same plaintext and main key, we obtain different ciphertext. We can say, that we have another block cipher. The changing of S-boxes prevents from receiving enough information to execute the effective cryptanalytical attack.

So the main question is how to change S-boxes without delivering them to the cryptographic devices. The answer is really simple. The encryption/decryption party has to generate identical substitution boxes, which will be used in the encryption/decryption process. Therefore deterministic methods of generating S-boxes using sequences generated

by pseudorandom generator are proposed. The encryption and decryption party has to have the same parameters of the generation process, for example the seed and the algebraic module, so that it can generate identical sequences. Using these sequences and presented in chapter 2 methods, it is possible to generate identical S-boxes, which can be change in the block cipher.

S-boxes, which can be applied in the block cipher, should fulfill the cryptographical criterion like balancedness, nonlinearity, strict avalanche criterion. The article proposes two algorithms of generation S-boxes using pseudorandom sequences and algorithm of generation S-boxes with guaranteed parameters like nonlinearity and distance to SAC. Therefore, generated S-boxes will be checked towards their nonlinearity and distance to SAC.

## 2. THE S-BOX GENERATION METHOD USING BINARY SEQUENCE

The analysis of opportunities of replacement substitution boxes in block ciphers was preceded by analysis of generation them using binary sequence. With this view two constructions are proposed which allow to generate truth table of the S-box  $S: Z_2^n \rightarrow Z_2^m$  from binary sequence [5].

### Construction 1

Let  $Z=(z_0, z_1, z_2, \dots, z_{r-1})$ ,  $z_i \in Z_2$ , be a binary sequence. Using this sequence we construct  $m$ -binary vectors:  $\underline{z}_0=(z_0, \dots, z_{m-1})$ ,  $\underline{z}_1=(z_m, \dots, z_{2m-1})$ , ...,  $\underline{z}_k=(z_{km}, \dots, z_{(k+1)m-1})$ ,  $z_i \in Z_2^m$ . The first nonzero vector  $\underline{z}_l=(z_{lm}, \dots, z_{(l+1)m-1})$  is the first row in the generated truth table. The next vectors  $\underline{z}_{l+1}$ ,  $\underline{z}_{l+2}$ , ... are checked if they are different from the vectors in the truth table. In the case of the positive verification such a vector is the next row in truth table, otherwise the vector is omitted and the next vector is checked. The process is continued until the truth table is full.

**Algorithm 2.1** Algorithm of S-box  $S: Z_2^n \rightarrow Z_2^m$  generation using construction 1

Input:  $Z=(z_0, z_1, z_2, \dots, z_{r-1})$ ,  $z_i \in Z_2$  – the binary sequence,

Output:  $Sb=[s_0, s_1, \dots, s_{2^n-1}]$  – the S-box table.

```

l:=0;
repeat
  Sb[0]:=0;
  for j:=0 to m-1 do:
    Sb[0]:= Sb[0] + z1 · 2j;
    l:=l+1;
until Sb[0]≠0;
for i=1 to 2n-1 do:
  repeat:
    Sb[i]:=0;
    for j:=0 to m-1 do:
      Sb[i]:= Sb[i] + z1 · 2j;
      l:=l+1;
      if (Sb[i] ∈ {s0, ..., si-1}) then spr:=1,
      otherwise spr:=0;
    until spr=0;

```

### Construction 2

Let:  $Z^0=(z_0^0, z_1^0, \dots, z_{r-1}^0)$ ,  $z_i^0 \in Z_2$ ,  
 $Z^1=(z_0^1, z_1^1, \dots, z_{r-1}^1)$ ,  $z_i^1 \in Z_2, \dots$ ,  
 $Z^{m-1}=(z_0^{m-1}, z_1^{m-1}, \dots, z_{r-1}^{m-1})$ ,  $z_i^{m-1} \in Z_2$ .

be  $m-1$  random sequences.

Using them we construct  $m$ -binary vectors in the following way:  $\underline{z}_0=(z_0^0, z_1^0, \dots, z_{m-1}^0)$ ,  $\underline{z}_1=(z_0^1, z_1^1, \dots, z_{m-1}^1), \dots$ ,  $\underline{z}_k=(z_k^0, z_k^1, \dots, z_k^{m-1})$ ,  $\underline{z}_i \in Z_2^m$ . The first nonzero vector  $\underline{z}_l=(z_l^0, z_l^1, \dots, z_l^{m-1})$  is the first row in the generated truth table. The next vectors  $\underline{z}_{l+1}$ ,  $\underline{z}_{l+2}$ , ... are checked if they are different from the vectors in the truth table. In the case of the positive verification such a vector is the next row in truth table, otherwise the vector is omitted and the next vector is checked. The process is continued until the truth table is full.

**Algorithm 2.2** Algorithm of S-box  $S: Z_2^n \rightarrow Z_2^m$  generation using construction 2

Input:  $Z^0=(z_0^0, z_1^0, \dots, z_{r-1}^0)$ ,  
 $Z^1=(z_0^1, z_1^1, \dots, z_{r-1}^1), \dots$ ,  
 $Z^{m-1}=(z_0^{m-1}, z_1^{m-1}, \dots, z_{r-1}^{m-1})$  – binary sequences,  $z_i^j \in Z_2$ ;

Output:  $Sb=[s_0, s_1, \dots, s_{2^n-1}]$  – the S-box table.

```

l:=0;
repeat
  Sb[0]:=0;
  for j:=0 to m-1 do:
    Sb[0]:=Sb[0]+Z1j·2j;
    l:=l+1;
until Sb[0]≠0;
for i=1 to 2n-1 do:
  repeat:
    Sb[i]:=0;
    for j:=0 to m-1 do: Sb[i]:=Sb[i] + Z1j·2j;
    l:=l+1;
    if (Sb[i]∈{s0...si-1}) then spr:=1,
    otherwise spr:=0;
  until spr=0;

```

The presented above methods were implemented. Generated S-boxes were verified towards their nonlinearity and strict avalanche criterion.

### 3. GENERATION S-BOXES USING RANDOM BINARY SEQUENCE

The random numbers play a crucial part in cryptography. They are used in many cryptographical applications and devices and may be generated exclusively by hardware binary sequence generator, for example SGCL-1 [3], which was designed in Military Communication Institute. Using this generator and proposed method 1000000 S-boxes  $S: Z_2^8 \rightarrow Z_2^8$  were generated. They were tested towards their nonlinearity and strict avalanche criterion. The number of S-boxes depending on nonlinearity and distance to SAC was shown in the table 3-1, 3-2 and illustrated on the schemes 3.1÷3.4.

S-box	nl(S)	construction 1	construction 2
$S: Z_2^8 \rightarrow Z_2^8$	70	0	1
	72	1	0
	74	1	0
	76	4	3
	78	25	25
	80	153	175
	82	699	666
	84	2849	2874
	86	11420	11380
	88	41955	41527
	90	132211	131664
	92	313810	314461
	94	385570	386342
	96	109767	109350
	98	1535	1532

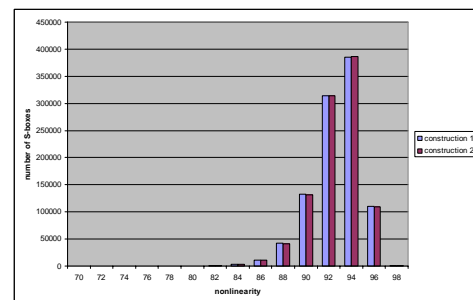
Table 3-1 The results of tests of S-boxes from the point of view of nonlinearity

The maximum value of the nonlinearity is 98 and it is achieved in 0,1535% (construction 1) and 0,1532% (construction 2) generated S-boxes. The nonlinearity, which accomplishes the maximum number of S-boxes is equal 94 and it is achieved in 38,5570% (construction 1) and 38,6342 (construction 2) S-boxes.

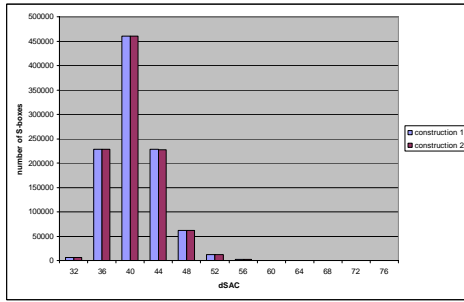
S-box	dSAC(S)	construction 1	construction 2
$S: Z_2^8 \rightarrow Z_2^8$	32	6359	6303
	36	228686	228574
	40	459965	460667
	44	227993	227715
	48	61804	61624
	52	12644	12608
	56	2190	2144
	60	307	318
	64	44	41
	68	6	5
	72	1	1
	76	1	0

Table 3-2 The results of tests of S-boxes from the point of view of distance to SAC

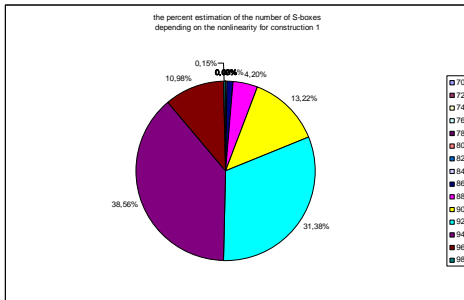
The minimum value of the dSAC is 32 and it is achieved in 0,6359% (construction 1) and 0,6303% (construction 2) generated S-boxes. The dSAC, which accomplishes the maximum number of S-boxes is equal 40 and it is achieved in 45,9965% (construction 1) and 46,0667 (construction 2) S-boxes.



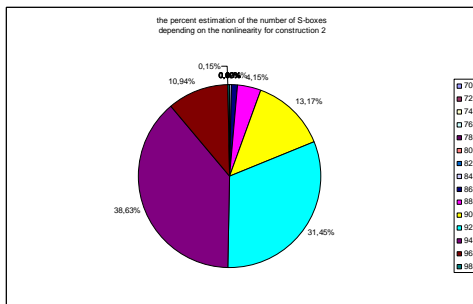
Scheme 3.1 The number of S-boxes depending on nonlinearity



Scheme 3.2 The number of S-boxes depending on dSAC



Scheme 3.3 The percent estimation of the number of S-boxes depending on the nonlinearity (construction 1)



Scheme 3.4 The percent estimation of the number of S-boxes depending on the nonlinearity (construction 2)

If we assume that ‘good’ S-box should have nonlinearity at the level of 90 and dSAC at more than 48 then the probability, that generated S-box will satisfy above-mentioned assumptions, is very high and is equal 0,9286 for both constructions.

#### 4. GENERATION S-BOXES USING PSEUDORANDOM BIT SEQUENCE

The pseudorandom numbers play a important part in cryptography, too. They are used in many cryptographical applications and devices and may be generated by

pseudorandom generators, for example Legendre’s generator [2], BBS generator [1], inverse generator [1] and so on. The pseudorandom sequence generators are composed as linear or nonlinear. In cryptography nonlinear pseudorandom generators are used because of characteristics like:

- deterministic method of generating sequences – this feature allows to generate the same sequence by independent computation party;
- improvement of quality generated sequences;
- nonlinearity of the generated sequence – very important feature, as S-boxes should have high nonlinearity, so they should be generated using nonlinear sequences.

The Legendre’s generator is defined in the following way. Let  $p$  be an odd prime number then  $X_n = 1$  if  $n$  jest quadratic residue modulo  $p$  and  $X_n = 0$  otherwise, where  $n$  takes following natural value. The Legendre’s sequences satisfy Golomb’s postulates [2].

Using Legendre’s generator and proposed constructions 1000000 S-boxes  $S: Z_2^8 \rightarrow Z_2^8$  were generated. They were tested towards their nonlinearity and strict avalanche criterion. The number of S-boxes depending on nonlinearity and distance to SAC was showed in the table 4-1, 4-2 and illustrated on the schemes 4.1÷4.4.

S-box	nl(S)	construction 1	construction 2
$S: Z_2^8 \rightarrow Z_2^8$	74	2	0
	76	6	3
	78	28	39
	80	156	143
	82	666	643
	84	2816	2942
	86	11295	11430
	88	41727	41761
	90	131692	132081
	92	314333	314254
	94	385979	384790
	96	109769	110420
	98	1531	1494

Table 4-1 The results of tests of S-boxes from the point of view of nonlinearity

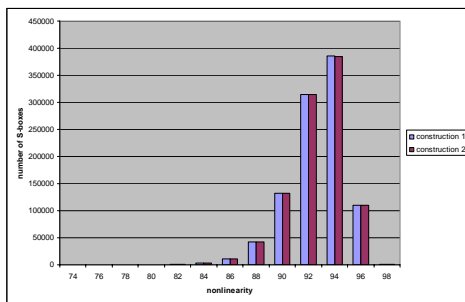
The maximum value of the nonlinearity is 98 and it is achieved in 0,1531%

(construction 1) and 0,1494% (construction 2) generated S-boxes. The nonlinearity, which accomplishes the maximum number of S-boxes is equal 94 and achieves it 38,5979% (construction 1) and 38,4790 (construction 2) S-boxes.

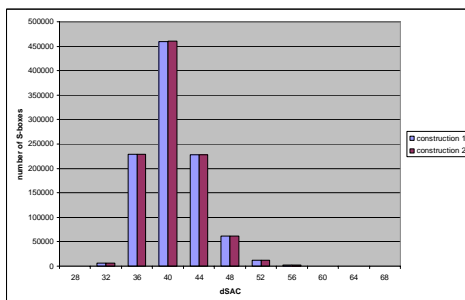
S-box	dSAC(S)	construction 1	construction 2
$S: Z_2^8 \rightarrow Z_2^8$	28	1	1
	32	6254	6169
	36	229105	228828
	40	459813	460664
	44	228003	228002
	48	61710	61344
	52	12628	12547
	56	2095	2079
	60	341	315
	64	42	48
68	8	3	

Table 4-2 The results of tests of S-boxes from the point of view of the distance to SAC

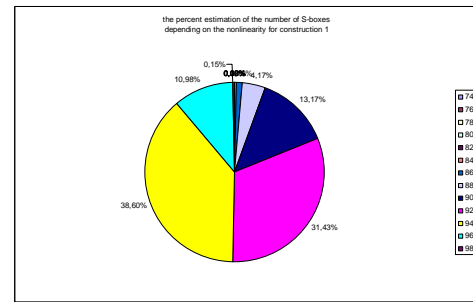
The minimum value of the distance to the strict avalanche criterion is 28 and it is achieved in 0,0001% (both constructions) generated S-boxes. The dSAC, which accomplishes the maximum number of S-boxes is equal 40 and it is achieved in 45,9813% (construction 1) and 46,0664 (construction 2) S-boxes.



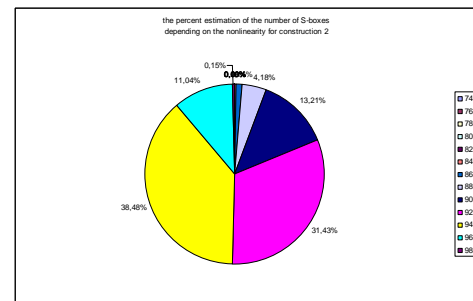
Scheme 4.1 The number of S-boxes depending on nonlinearity



Scheme 4.2 The number of S-boxes depending on dSAC



Scheme 4.3 The percent estimation of the number of S-boxes depending on the nonlinearity (construction 1)



Scheme 4.4 The percent estimation of the number of S-boxes depending on the nonlinearity (construction 2)

If we assume that ‘good’ S-box should have nonlinearity at least the level of 90 and dSAC at more than 48 then the probability, that generated S-box will satisfy above-mentioned assumptions, is very high and is equal 0,9290 (construction 1) and 0,9289 (construction 2).

## 5. THE METHOD OF IMPROVING THE SECURITY OF THE CRYPTOSYSTEM BASED ON THE BLOCK CIPHER

The idea of improving the security of cryptosystem based on block cipher depends on replacement of substitution boxes in the encryption/decryption algorithm. So that, it would be possible, the encryption and decryption party have to generate identical S-boxes for the cryptographic session. This

possibilities are given by the method of generating S-boxes from pseudorandom sequences, because the encryption/decryption party can generate identical bits streams and as a results thanks to proposed method – identical S-boxes. With this view we can apply the following method based on the DH algorithm [3] to arrange the seed:

1. the encryption/decryption party has to have identical numbers  $g, m$ ;
2. the encryption party chooses a number:  $1 < x < m$  and calculates:  $X = g^x \text{ mod } m$ ;
3. the decryption party chooses a number:  $1 < y < m$  and calculates:  $Y = g^y \text{ mod } m$ ;
4. the encryption/decryption party exchanges numbers  $X$  and  $Y$ ;
5. the encryption/decryption party calculates the seed:  $s = g^{xy} \text{ mod } m$ .

The encryption/decryption party using the seed  $s$  and module  $m$  generates Legendre's a sequence and then substitution boxes. The generated S-boxes should characterize high nonlinearity (at least  $nl_{min}$ ) and low dSAC (at most  $dSAC_{max}$ ), so the algorithm of generation of such S-boxes is as follow:

**Algorithm 5.1** Algorithm of the S-boxes table generation

Input:  $nl_{min}$  – the minimal value of the nonlinearity of the generated S-boxes;  
 $dSAC_{max}$  – the maximal value of the dSAC of the generated S-boxes;

Output:  $TSB = [SB_0, SB_1, \dots, SB_{r-1}]$  – the table of S-boxes.

```

for i:=0 to r-1 do:
  repeat:
     $SB_i := SbGen$ ;
    // SbGen - a S-box generation
    algorithm using constr. 1
    or constr. 2;
     $nl := Licz\_Nl(SB_i)$ ;
    // Licz_Nl - a nonlinearity of
    the S-box calculation algorithm;
     $dsac := Licz\_dSAC(SB_i)$ ;
    // Licz_dSAC - an dSAC of the S-box
    calculation algorithm;
  until ( $nl \geq nl_{min}$  and  $dsac \leq dSAC_{max}$ )

```

The exchanging of the S-boxes in block ciphers causes that a different block cipher is applied in the cryptosystem. This makes

impossible to collect sufficient amount of information to carry back the cryptanalytical attack and in consequence rises the security of cryptosystem.

## 6. CONCLUSIONS

The sequences generated by the Legendre's a generator satisfy statistical tests (FIPS 140-1), have statistical characteristic like random sequences and are strongly nonlinear. For this reason they carry out experiments of making substitution boxes using the proposed method and Legendre's a sequences. The researches of nonlinearity and dSAC of generated S-boxes show that, giving the maximum nonlinearity and minimum dSAC up, it is possible to generate 'good' S-boxes. It gives the possibility of making S-boxes used in block ciphers and make them replaceable. This treatment secures cryptosystems against many cryptographical attacks, especially differential and algebraic cryptanalysis.

## BIBLIOGRAPHY

- [1] T.W. Cusik, C. Ding, A. Renevall, "Stream ciphers and number theory", North Holland Mathematical Library, 1998.
- [2] I. D. Damagard, "On the randomness of Legendre and Jacobi sequences", Eurocrypt 1988, Springer-Verlag, 1988.
- [3] M. Leśniewicz, „Sprzętowy generator ciągów losowych do zastosowań kryptograficznych”, XVII KST Bydgoszcz 2001.
- [4] A. Menezes, P. van Oorschot, S. Vanstone "Handbook of applied cryptography", CRC Press, 1997.
- [5] P. Mroczkowski, A. Paszkiewicz, „About the designing method of strong cryptographically boolean functions”, ENIGMA 2007.

## BIOGRAPHY

Piotr Mroczkowski was born in Ilża, Poland, in 1975. He was graduated from the Cybernetics Faculty of the Military University of Technology in Warsaw in 2000. Since 2000 he has been working at the Military

Communication Institute. He is interested in cryptology and computer science.

Przemysław Rodwald was born in Lębork, Poland, in 1977. He was graduated from the Cybernetics Faculty of the Military University of Technology in Warsaw in 2001. Since 2002 he has been working at the Military Communication Institute. He is currently working in the NATO Internet Protocol Security Task Force group. His main objects of interest: hash functions and cryptanalysis.