A New Method for Generating High Non-linearity S-Boxes

Petr TESAŘ

CEZ, a.s., Duhova 2, 140 53 Prague 4, Czech Republic

petr.tesar@cez.cz

Abstract. *Substitution boxes are important parts in many block and stream ciphers. The emergence of a range of crypto-attacks has led to the development of criteria for repelling such attacks. The non-linearity criterion provides some protection against well-known attacks, such as linear cryptanalysis and differential cryptanalysis. The open problem is constructed by generating methods which will be rapid and will generate S-boxes with the highest possible non-linearity. This paper deals with a new rapid method for generating regular 8x8 S-boxes with non-linearity up to a value of 104. The new method combines the special genetic algorithm with total tree searching.*

Keywords

Cryptography, regular 8x8 S-box, non-linearity criterion, generation method.

1. Introduction

All modern block and stream ciphers have one or more non-linear elements. S-box is one of the most used non-linear cornerstones of modern ciphers. We denote the substitution table of an *N*-input *K*-output Boolean function by f: $B^N \rightarrow B^K$, mapping each combination of *N* Boolean input values to some combination of *K* Boolean output values. *B* is a 1-dimensional Boolean space {0,1}. For single-output functions if the number of combinations mapping to 0 is the same as the number mapping to 1, then the function is said to be *balanced*. For the multiple-output case if each *K*-bit output value appears the same number of times, we say that the function is *regular*.

To be cryptographically secure, the S-box must be regular in order to prevent the system from releasing statistical information on the plaintext when the ciphertext is known. Cryptographically good S-boxes must also fulfill other criteria. One of the most significant criteria is so-called non-linearity. The goal of this paper is to construct a new rapid method for generating special S-boxes with the desired value of non-linearity.

2. Definitions

This section provides some definitions of relevance to Boolean functions with cryptographic applications.

2.1 Single-output Case

The definitions in this section are taken from [2].

Definition 1 – Linear Boolean Function

A linear Boolean function selected by $\boldsymbol{\alpha} \in B^N$, is denoted by

$$L_\alpha(x) = \alpha_1 x_1 \oplus \alpha_2 x_2 \oplus \ldots \oplus \alpha_N x_N \tag{1}$$

where $\alpha_i x_i$ denotes the bitwise AND of the *i*-th bits of $\boldsymbol{\alpha}$ and $\boldsymbol{x}$, and $\oplus$ denotes bitwise XOR.

Definition 2 – Affine Boolean Function

The set of affine Boolean functions is the set of linear Boolean functions and their complements

$$A_{\alpha,c}(x) = L_\alpha(x) \oplus c \tag{2}$$

where $c \in B$.

Definition 3 – Walsh Hadamard Transform

For a Boolean function *f*, the Walsh Hadamard Transform $\hat{F}_f$ is defined by

$$\hat{F}_f(\alpha) = \sum_{x \in B^N} \hat{f}(x) \hat{L}_\alpha(x) \tag{3}$$

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

We denote the maximum absolute value taken by the Walsh Hadamard Transform by

$$WHT_{\max}(f) = \max_{\alpha \in B^N} \left| \hat{F}_f(\alpha) \right|. \tag{4}$$

Definition 4 – Non-linearity

The nonlinearity N_f of a Boolean function *f* is its minimum distance to any affine function. It is given by

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

It is known that for N even, the maximum non-linearity attainable is [8]

$$N_{\max}(N) = 2^{N-1} - 2^{\frac{N}{2}-1} \quad (6)$$

but such functions (bent functions) are not balanced.

2.2 Multi-output Case

The NxK S-box is a Boolean function with N-bit input and K-bit output Boolean variables $\boldsymbol{x} = (x_1, x_2, ...x_N)$ and $\boldsymbol{y} = (y_1, y_2, ..., y_K)$. The NxK S-box is a set of K single-output Boolean functions $(f_1, f_2, ..., f_K)$ where $f_i(x) = y_i$ for $i = 1, ..., K$.

Each $\beta \in B^K$ defines a function that is a linear combination $f_\beta(x)$ of the K outputs of the S-box.

$$f_\beta(x) = \beta_1 f_1(x) \oplus .... \oplus \beta_K f_K(x). \quad (7)$$

For each such function f_β the non-linearity is defined in the usual way.

There are 2^K-1 non-trivial functions obtainable in this way. The notion of non-linearity is readily extended to the NxK S-box. Non-linearity is the worst (lowest) non-linearity of all 2^K-1 non-trivial single output functions obtained as indicated above.

Only regular S-boxes with $N = K = 8$ will be examined in this paper. The main reason is that a byte (= 8 bits) is a basic computer memory element, and there are many byte-oriented ciphers.

3. Generating Methods - Overview

Three main classes of methods for generating S-boxes and Boolean functions with the desired non-linearity are given in the publicly-available literature:

1. Random searching
2. Construction methods
3. Evolutionary (or genetic) searching

The simplest method is random searching. Non-linearity 98 is the highest value for this method in public papers ([1] and [3]) for 8x8 S-boxes. Four 8x8 S-boxes with non-linearity 100 were found in an experiment by the author with 50 million random 8x8 S-boxes. Non-linearity 112 is the highest value for balanced Boolean functions [7].

Hill climbing, as a representative of construction methods, is taken from papers [1] and [4]. The available limit of non-linearity is 100 for 8x8 S-boxes [1] and 114 for balanced Boolean functions [7].

Paper [6] provides the hill climbing method to produce 9x32 S-boxes with good cryptographic properties.

In paper [7] the modified hill climbing method is considered. The method enables the application of hill climbing techniques to modify the bent functions used to design balanced highly non-linear Boolean functions. The main idea of the method is to invert the hill climbing algorithm. A bent function is used as input data instead of a randomly generated Boolean function, and the non-linearity of the bent function is decreased to a required value, instead of increasing the non-linearity of a randomly generated Boolean function. In the 8-input case, balanced Boolean functions are constructed with non-linearity up to 116.

Evolutionary searching methods have many variants. A common schema is:

a) We have an initial set of objects (S-boxes or Boolean functions), known as a farm. Let M be the size of the farm. The initial farm is mostly produced by a random search.

b) We derive successors from the objects of the farm. We count the cost function for all successors. We order them by this cost function. The best M of them become the new farm.

c) We repeat step b) until the object with the desired feature is found or the STOP criterion is fulfilled.

There are many parameters – the number M, the genetic method for creating successors, the number of successors, the type of cost function, and the STOP criteria. The crucial parameter is the type of cost function. Older versions used the non-linearity value as the cost function. In paper [3], the authors applied genetic algorithms followed by hill climbing to evolve 8xK regular S-boxes for $K = 1,\ldots,8$. The available limit of non-linearity was 100 in these cases for 8x8 S-boxes. A much smarter cost function was published in [2]. This cost function *WHS* (see Definition 5) has two variables and thus defines the broad set of cost functions. A new search technique inspired by the cooling processes of molten metals – *annealing* – together with the cost function *WHS* is proposed in [2]. For 8x8 S-boxes this method repeatedly generates 8x8 S-boxes with non-linearity 102.

In paper [5] Millan et al. applied a genetic algorithm to construct cryptographically strong balanced Boolean functions with non-linearity up to 116, in the 8-input case.

4. New Method – GaT

The new method presented in this paper combines the special genetic algorithm with total tree searching. The name of the method is GaT ("Genetic and Tree"). The best version of the GaT algorithm uses the special cost function *WHS* from [2].

Definition 5 – Cost function *WHS*

For an *N*x*K* S-box *P* the cost function $WHS_{R,X}(P)$ is defined by

$$WHS_{R,X}(P) = \sum_{\beta \in B^K} \sum_{w \in B^N} \left| \left| \hat{F}_\beta(w) \right| - X \right|^R \qquad (8)$$

where *X* and *R* are real-valued parameters and $\hat{F}_\beta(w)$ is the Walsh Hadamard Transform by Definition 3 equation (3). Low *WHS* values are better.

In [2], *R*=3.0 and the *X* integer value from the interval <-4, +4> were used with a special search algorithm - annealing.

For an algorithm that generates good 8x8 S-boxes, it will be shown that a much better (*R, X*) parameter pair exists. In a huge experiment with the *R* integer value from the interval <3, 25> and the *X* integer value from the interval <-25, 25>, all 1173 pairs (*R, X*) were rated by some heuristic measure by the author, and the best pair was *R*=7 and *X*=21. The best pair of the subset from paper [2] was *R*=3 and *X*=4 (832. position).

The general schema of the GaT method is taken from the author.

STEP 0: We define the parameters of the method, especially the integer *N* - dimension of the *NxN* S-box, where *N*=8, integer *M* - number of S-boxes in the farm, integer *C* - number of successors for each farm's S-box, integer *I* - maximum number of iterations in the genetic part of the algorithm, even integer *NT* - non-linearity value to go to the tree part of the algorithm, integer *Z* - maximum number of overall searched S-boxes and even integer *NEL* - desired non-linearity value. We define the cost function *CF*. If *NEL* is less or equal to *NT*, then the tree part of the algorithm will not be used.

STEP 1: Set zero to number of iterations. Set *M*x*C* to number of overall searched S-boxes. Randomly generate *M*x*C* S-boxes. This set is called the population. For all S-boxes from the population, count the cost function *CF* and non-linearity. Set as current the best non-linearity from the population to the integer *CN*. If any S-box from the population has non-linearity greater or equal to *NEL*, then GoTo STEP 9 or if any S-box from the population has non-linearity greater or equal to *NT*, then GoTo STEP 4. Otherwise order the S-boxes by the cost function *CF* and the best *M* of them become a farm. Then GoTo STEP 2.

STEP 2: For each S-box *P* from the farm, randomly select *C* successors from the neighborhood *N*(*P*). The neighborhood *N*(*P*) is a set of all *N*x(*N*-1) S-boxes by simply swapping the output values associated with two input values of *P*. The original *P* can also be selected, and repetitions of successors are allowed. Now we have the new population. Then, for all S-boxes from the new population, count the cost function *CF* and the non-linearity. Update *CN*. Increase the number of iterations by one. Increase the number of overall searched S-boxes by *M*x*C*. If the number of iterations is greater than *I*, or if the number of overall searched S-boxes is greater than *Z*, then GoTo STEP 10, otherwise GoTo STEP 3.

STEP 3: If any S-box from the population has non-linearity equal to *NEL*, then GoTo STEP 9, or if any S-box from the population has non-linearity equal to *NT*, then GoTo STEP 4. Otherwise order the S-boxes by the cost function *CF* and the best *M* of them become the new farm. Then GoTo STEP 2.

STEP 4: Let P_1 be the S-box with non-linearity greater or equal to *NT* and with cost function $CF(P_1)$. Set *CN:=NT*. Search step by step overall $N(P_1)$, until

a) find S-box with non-linearity equal to *NEL*, then GoTo STEP 9, or

b) find S-box with non-linearity greater than *CN*, then GoTo STEP 5, or

c) find S-box with non-linearity equal to *CN* and a better cost function than $CF(P_1)$, then GoTo STEP 5, or

d) no S-box from $N(P_1)$ has values to go to option a), b) or c), or the number of overall searched S-boxes is greater than *Z*, then GoTo STEP 10.

STEP 5: Let P_2 be an S-box from option STEP 4 b) or c). Set *CN*:=non-linearity of P_2.

Save P_1 with ordinal number of P_2 over $N(P_1)$ into a LIFO (=Last In First Out) stack and set *j*=2 (ordinal number of the diagnostic S-box). Then GoTo STEP 6.

STEP 6: Search step by step overall $N(P_j)$ until:

a) find S-box with non-linearity equal to *NEL*, then GoTo STEP 9, or

b) find S-box with non-linearity greater than *CN*, then GoTo STEP 7, or

c) find S-box with non-linearity equal to *CN* and a better cost function than $CF(P_j)$, then GoTo STEP 7, or

d) the number of overall searched S-boxes is greater than *Z*, then GoTo STEP 10, or

e) no S-box from $N(P_j)$ has values to go to option a), b) or c), then GoTo STEP 8.

STEP 7: Set *j:=j*+1. Let P_j be the S-box from the option STEP 6 b) or c). Set *CN*:=non-linearity of P_j. Save P_{j-1} with ordinal number of P_j over $N(P_{j-1})$ into the LIFO stack. Then GoTo STEP 6.

STEP 8: Set *j:=j*-1. If *j*=0, then GoTo STEP 10, else resume P_j from LIFO stack and GoTo STEP 6.

STEP 9: STOP algorithm. S-box with non-linearity equal or greater than *NEL* has been found.

STEP 10: STOP algorithm. S-box with non-linearity equal or greater than *NEL* was not found.

5. Experimental Results

We can compare the main types of algorithms for generating 8x8 S-boxes with the desired non-linearity value. The criterion is the number of overall searched S-boxes to discover the S-box with the desired non-linearity value. Tab. 1 includes the mean values of the overall searched 8x8 S-boxes for the five methods under investigation. The results were obtained by the author. The result of the Hill Climbing method is taken from paper [1].

Method \ Non-linearity	98	100	102	104
Random searching	695	12.5 M	NRP	NRP
Hill Climbing	NA	2500	NRP	NRP
GaT with CF=Non-linearity	171	29648	NRP	NRP
GaT with CF=$WHS_{3,4}$	217	2714	79385	NRP
GaT with CF=$WHS_{7,21}$	64	522	9859	3.239 M

Tab. 1. Mean values of criterion (M is million, NA means not available, and NRP means the method is not able to produce repeatedly.).

The values of Tab. 1 are obtained on the basis of many huge experiments. For example, 70 S-boxes with non-linearity 104 were generated.

The optimal parameters of the GaT method for generating 8x8 S-boxes with *NEL*=104 are

$M=C=10$,
$I=200$,
$NT=102$,
$Z=9000000$,
$CF=WHS_{7,21}$.

6. Conclusion

The new GaT method with cost function $WHS_{7,21}$ is statistically faster (significance level 1%) than the openly public methods known to the author for generating 8x8 S-boxes with non-linearity 98 – 102, and it is capable of repeatedly producing 8x8 S-boxes with non-linearity 104.

The method is not capable of producing 8x8 S-boxes with non-linearity higher than 104.

The 5x5, 6x6 and 7x7 S-boxes were also generated by GaT, for guidance. The results are shown in Tab. 2.

S-box	Non-linearity
5x5	10
6x6	22
7x7	48

Tab. 2. Maximum obtained non-linearity.

The GaT parameters for generating small S-boxes are not optimal. Better parameters for these small S-boxes will be sought in future, and the results will be compared with the standard methods.

A future plan is to examine GaT with other criteria, for example autocorrelation.

References

[1] MILLAN, W. How to improve the nonlinearity of bijective S-Boxes. In *Australian Conference on Information Security and Privacy,* 1998, p. 181 - 192.

[2] CLARK, J. A., JACOB, J. L., STEPNEY, S. The design of S-Boxes by simulated annealing. *New Generation Computing Archive*, September 2005, vol. 23, issue 3.

[3] MILLAN, W., BURNETT, L., CARTER, G., CLARK, A., DAWSON, E. Evolutionary heuristics for finding cryptographically strong S-Boxes. Springer-Verlag*, Lecture Notes in Computer Science,* 1999, vol. 1726*,* p. 263 – 274.

[4] MILLAN, W., CLARK, A., DAWSON, E. Boolean function design using hill climbing methods. In *4th Australian Conference on Information Security and Privacy.* Ed. Bruce Schneier, Springer-Verlag, Lecture Notes in Computer Science, 1999, vol. 1587, p. 1 – 11.

[5] MILLAN, W., CLARK, A., DAWSON, E. Heuristic design of cryptographically strong balanced Boolean functions. In *Advances in Cryptology EUROCRYPT'98,* Springer-Verlag, Lecture Notes in Computer Science, 1998, vol. 1403, p. 489 – 499.

[6] BURNETT, L., CARTER, G., DAWSON, E., MILLAN, W. Efficient methods for generating MARS-Like S-Boxes. In *7th International Workshop on Fast Software Encryption 2000,* Bruce Schneier, ed. Springer-Verlag, Lecture Notes in Computer Science, 2001, vol. 1978, p. 300 – 314.

[7] IZBENKO, Y., KOVTUN, V., KUZNETSOV, A. *The Design of Boolean Functions by Modified Hill Climbing Method (datasheet)*. 17 pages. [Online] Cited 2008. Available at: http://eprint.iacr.org/2008/111.pdf

[8] Mac WILLIAMS, F.J., SLOANE, N.J.A. *The Theory of Error Correcting Codes*. Amsterdam: North-Holland Publishing Company, 1978.

About the Author

Petr TESAŘ was born in Prague. He received his M.Sc. in mathematics from the Math. Phys. Faculty of Charles University Prague in 1974. His research interests include cryptography algorithms and e-signatures.