A Method For Generation Of High-Nonlinear S-Boxes Based On Gradient Descent

Oleksandr Kazymyrov, Valentyna Kazymyrova, Roman Oliynykov

Abstract

Criteria based on the analysis of the properties of vectorial Boolean functions for selection of substitutions (S-boxes) for symmetric cryptographic primitives are given. We propose an improved gradient descent method for increasing performance of nonlinear vectorial Boolean functions generation with optimal cryptographic properties. Substitutions are generated by proposed method for the most common 8-bits input and output messages have nonlinearity 104, 8-uniformity and algebraic immunity 3.

Keywords: substitution, nonlinearity, symmetric ciphers, vectorial Boolean function

1 Introduction

S-boxes are one of the main components that determine the robustness of modern symmetric cryptographic primitives. They typically perform the mapping of n-bit input block to the output of m-bits length. Representation of S-boxes varies depending on kind of problem they are used for. In stream ciphers substitutions are usually presented in the form of vectorial Boolean functions [1]. Permutations are a subclass of substitutions and are widely used in block ciphers in a table form. It is quite easy to transform a substitution from one form to another.

To protect cryptographic primitives against various types of attacks a substitution must satisfy a number of criteria [2, 3]. Taking into account the large number of existing characteristics, their controversy and partial interdependence, it is likely impossible to generate a substitution that satisfies all known requirements. This became a reason to use a substitution satisfying only mandatory criteria essential for a particular symmetric algorithm. Such substitutions are called optimal [4]. Optimality criteria may vary depending on which cipher is considered. Generating of permutations with optimal criteria is a time- and resource-consuming task, especially for large n and m.

This problem is partially solved by involving the classes of vectorial Boolean functions, extended affine (EA) and Carlet-Charpin-Zinoviev (CCZ) equivalencies [1, 5]. However, the majority of existent functions have extreme characteristics of δ-uniformity and nonlinearity, but at the same time do not possess other properties (i.e., high algebraic immunity) necessary for symmetric cryptographic primitives. It was shown in 2010 that such substitutions exist [6]. Tesar proposed an algorithm based on combination of genetic algorithm and total tree searching. In this paper we give more simple and efficient way of generation substitutions.

In [7] the authors have enhanced the method for generating secure Boolean functions based on gradient ascension (Hill Climbing) method [8]. In this paper we propose a modified version of the gradient descent method for vectorial case, i.e. for functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$.

2 Preliminaries

Arbitrary substitution can be represented in at least three different forms: algebraic normal form (ANF), over the field $\mathbb{F}_{2^n}$ and a lookup table. The majority of block ciphers S-boxes have a lookup table form because of their simple description and understanding. At the same time, an arbitrary permutation can always be associated with a vectorial Boolean function F in $\mathbb{F}_{2^n}[x]$. If the substitution is a permutation, then the function F is unique.

A natural way of representing $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ is algebraic normal form

$$\sum_{I \subseteq \{1,\dots,n\}} a_I \left(\prod_{i \in I} x_i \right), \qquad a_I \in \mathbb{F}_2^m,$$

sum is calculated in $\mathbb{F}_2^m$ [1]. Algebraic degree of F is the degree of its ANF.

F is called affine if it has the algebraic degree at most 1. When $F(0) = 0$ affine vectorial Boolean function is linear.

Two functions $F, G : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ are called EA-equivalent if there are such affine permutation functions $A_1(x) = L_1(x) + c_1$, $A_2(x) = L_2(x) + c_2$ and arbitrary linear function $L_3(x)$ that [1, 5]

$$F(x) = A_1 \circ G \circ A_2(x) + L_3(x).$$

If $L_3(x)$ is a constant from the vector space F_2^m, then the functions F and G are called affine-equivalent, and if $L_3(x) = 0, c_1 = 0, c_2 = 0$ they are linear equivalent. Affine equivalence was used to prevent the appearance of fixed points during generation of substitutions for cipher Rijndael [9].

Arbitrary vectorial Boolean function F is δ-uniform if for any $a \in \mathbb{F}_2^n \setminus \{0\}$ and $b \in \mathbb{F}_2^m$ the equation $F(x) + F(x + a) = b$ has no more than δ solutions [1]. Vectorial Boolean functions that are used as substitutions in block ciphers must have a small value of δ-uniformity for a sufficient level of protection against differential attacks [1, 3].

Nonlinearity criterion is closely related to the Walsh transformation, which can be described by the function

$$\lambda(u, v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{v \cdot F(x) + u \cdot x},$$

where the symbol "$\cdot$" denotes the scalar product in vector spaces $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$. Substitutions with small values of Walsh coefficients are optimally protected against linear cryptanalysis [1, 3]. S-boxes with minimal values of $\lambda(u, v)$ exist only for odd n.

These two criteria are most significant when selecting substitutions for new ciphers. However, there are many other criteria such as: propagation criterion, maximum value of autocorrelation spectrum, correlation immunity, algebraic immunity, strict avalanche criterion, etc. [1, 2, 10]. Necessity for most of these criteria has not yet been proven. For example, the substitution used in AES does not satisfy most of them [2].

In this paper the optimal substitution refers to a permutation with

- maximum algebraic degree;

- maximum algebraic immunity with the minimum number of equations;
- maximum values of δ-uniformity and nonlinearity limited by parameters listed above;
- absence of fixed points (cycles of length 1).

For example, for $n = 8$ an optimal permutation has algebraic degree 7, algebraic immunity 3 and 441 equations, δ-uniformity under 8, nonlinearity over 100 and without fixed points.

3 Generation of Substitutions With Chosen Parameters

In [7] the main idea is to decrease the nonlinearity of given Boolean bent sequences. In other words, in given bent sequence (truth table) some bits are changed so that the new sequence is balanced and the nonlinearity is close to nonlinearity of bent function. In this paper is proposed to use the same approach, but with two significant differences

- use vectorial Boolean functions instead of Boolean functions;
- use vectorial Boolean functions (substitutions) with minimum δ-uniformity instead of bent-functions (sequences).

Additionally,[7] was shown that even a small change of fixed number of bits in the bent-sequence does not guarantee the achievement of nonlinearity closed to the maximum.

However, for the vectorial case the following proposition was proven [11].

Proposition 1 *Let $F : \mathbb{F}_{2^n} \mapsto \mathbb{F}_{2^n}$. Function G is determined so that*

$$\begin{cases} G(p_1) &= F(p_2) \qquad p_1 \neq p_2; \\ G(p_2) &= F(p_1); \\ G(x) &= F(x) \quad x \notin \{p_1, p_2\}. \end{cases}$$

Then

$$\delta(F) - 4 \leq \delta(G) \leq \delta(F) + 4,$$
$$Nl(F) - 2 \leq Nl(G) \leq Nl(F) + 2.$$

The nonlinearity function (Nl) of arbitrary vectorial function F is calculated as follows

$$Nl(F) = 2^{n-1} - \frac{1}{2} \cdot max_{u \neq 0, v \in \mathbb{F}_{2^n}} \left| \lambda(u, v) \right|.$$

We propose a new method to generate substitutions based on Proposition 1. The algorithm takes as input a bijective vectorial Boolean function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ with a minimum value of δ-uniformity, and number of values (NP) in function, which have to be changed during the optimization of cryptographic parameters.

The main steps of the algorithm are presented bellow.

1. Generate a substitution S based on F.
2. Swap NP values of S randomly and generate substitution S_t.
3. Test the S-box S_t for all criteria depending on their computational complexity. If S_t satisfies all of them except the cyclic properties, then go to step 3. Otherwise repeat step 2.
4. Apply equivalence (e.g. affine) to S_t in order to achieve the required properties of cycle structure.
5. Output of the algorithm. Required substitution will be stored in S_t.

Theoretical obtaining of swap iterations' number for arbitrary n-bit vectorial function F becomes an additional topic for the research.

4 Practical Results

Before the algorithm was designed practical opportunity of finding optimal substitutions for $n = 8$ had been tested. The challenge was to find several

CCZ-nonequivalent substitutions with nonlinearity equal or greater than 100. For practical realization a cluster with 4096 processors was used [12].

The program generated a random permutation and checked it for optimality. After 12 hours of cluster operation it was found 27 optimal permutations, four of which were CCZ-nonequivalent. An example of the permutation in hexadecimal notation is given in Table 1.

Table 1: An Example of Substitution with Nonlinearity 100

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	14	9D	B9	E7	67	4C	50	82	CA	E5	1D	31	0A	C6	B2	51
1	A2	D8	54	90	D0	CE	2D	7D	C7	7E	D7	94	DF	83	8E	6C
2	66	D2	6F	16	1E	76	FE	CC	AA	5A	8F	17	BD	2C	AC	EA
3	7B	65	A9	10	C0	92	EE	BE	6A	6E	48	96	95	E9	32	BC
4	A1	42	D5	A7	81	B4	5F	E6	C2	5D	AD	3A	B7	0C	8D	01
5	98	FD	12	02	75	13	0F	6B	22	E2	AB	F7	7F	BA	97	D1
6	64	D9	C4	59	AF	23	33	37	DE	AE	60	05	63	A8	52	A5
7	4E	E0	DD	71	F2	24	34	57	47	A4	B3	9E	2F	C1	B8	CB
8	2B	D4	0D	36	91	8B	9C	26	25	61	A3	D6	EB	35	53	F4
9	2E	88	80	E4	30	DB	FC	0E	77	8C	93	A6	78	06	E1	EC
A	F9	03	A0	27	DA	EF	5C	00	7A	45	E8	40	1A	4B	5E	73
B	C3	FF	F5	F3	B0	C5	49	21	FA	11	39	84	43	38	85	07
C	F0	79	46	F8	E3	1F	09	B6	CD	55	1C	1B	FB	7C	ED	6D
D	15	56	86	20	68	4A	41	4F	D3	99	08	F6	3F	89	62	04
E	CF	C8	69	9F	19	5B	44	9B	87	B1	3D	BB	DC	2A	BF	58
F	3C	8A	18	3E	72	0B	28	4D	B5	9A	C9	74	29	F1	3B	70

This substitution has the following characteristics

- nonlinearity 100;
- absolute value of the autocorrelation 96;
- minimum algebraic degree 7;
- 8-uniform;
- algebraic immunity: a system of 441 equations of the 3rd degree.

Furthermore, search for substitution with nonlinearity 102 was conducted. However, after 48 hours of cluster operation, which is approximately equal to 22 years of a single-processor computer operation, no substitutions were found. Thus, we can conclude that from a practical point of view, the generation of such permutations is computationally extremely difficult.

However, the algorithm described above allows to find such substitutions. For example, consider the function $F(x) = x^{-1}$. The value of NP equals 26 was experimentally found for $n = 8$ with providing the necessary properties of the substitution. An example of such permutation is presented in Table 2.

Table 2: An Example of Substitution with Nonlinearity 104

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	68	8D	CA	4D	73	4B	4E	2A	D4	52	26	B3	54	1E	19	1F
1	22	03	46	3D	2D	4A	53	83	13	8A	B7	D5	25	79	F5	BD
2	58	2F	0D	02	ED	51	9E	11	F2	3E	55	5E	D1	16	3C	66
3	70	5D	F3	45	40	CC	E8	94	56	08	CE	1A	3A	D2	E1	DF
4	B5	38	6E	0E	E5	F4	F9	86	E9	4F	D6	85	23	CF	32	99
5	31	14	AE	EE	C8	48	D3	30	A1	92	41	B1	18	C4	2C	71
6	72	44	15	FD	37	BE	5F	AA	9B	88	D8	AB	89	9C	FA	60
7	EA	BC	62	0C	24	A6	A8	EC	67	20	DB	7C	28	DD	AC	5B
8	34	7E	10	F1	7B	8F	63	A0	05	9A	43	77	21	BF	27	09
9	C3	9F	B6	D7	29	C2	EB	C0	A4	8B	8C	1D	FB	FF	C1	B2
A	97	2E	F8	65	F6	75	07	04	49	33	E4	D9	B9	D0	42	C7
B	6C	90	00	8E	6F	50	01	C5	DA	47	3F	CD	69	A2	E2	7A
C	A7	C6	93	0F	0A	06	E6	2B	96	A3	1C	AF	6A	12	84	39
D	E7	B0	82	F7	FE	9D	87	5C	81	35	DE	B4	A5	FC	80	EF
E	CB	BB	6B	76	BA	5A	7D	78	0B	95	E3	AD	74	98	3B	36
F	64	6D	DC	F0	59	A9	4C	17	7F	91	B8	C9	57	1B	E0	61

It has the following properties

- nonlinearity 104;
- absolute value of the autocorrelation 80;
- minimum algebraic degree 7;
- 8-uniform;
- algebraic immunity: a system of 441 equations of the 3rd degree.

During 1 hour of cluster operations 1152 permutations with nonlinearity 104 were generated, that shows the effectiveness of the proposed method.

Additional tests have shown that for the nonlinearity greater than 104, the substitutions are not optimal in terms of algebraic immunity. However, there are permutations with nonlinearity 106 and algebraic immunity 2, in which the number of equations is small (e.g. 2). Hereby, the question about existence of substitutions with algebraic immunity 3 and nonlinearity more than 104 remains open.

5 Conclusions

The proposed method is based on the already known method of gradient descent, but was adopted for vectorial case. It allows to find substitutions with desired properties, in contrast to the previous one, which only could find separate Boolean functions. Such substitutions can be used in modern symmetric algorithms that demand high level of robustness against various types of attacks.

References

[1] Crama Y., Hammer P.L. Boolean Models and Methods in Mathematics Computer Science, and Engineering. Encyclopedia of Mathematics and its Applications // Cambridge University Press. — 2010.

[2] Oliynykov R. Kazymyrov O. An Impact Of S-Box Boolean Function Properties To Strength Of Modern Symmetric Block Ciphers // Radio Engineering. — Kharkiv, 2011. — V. 166. — P. 11-17.

[3] Rijmen V. Cryptanalysis and design of iterated block ciphers, PhD Thesis. // University of Leuven: 1997.

[4] Kazymyrov O., Oliynykov R. Vectorial Boolean Functions Application for Substitutions Generation for Symmetric Cryptographic Transformation // Applied Radio Electronics. — Kharkiv, 2012. (In Russian)

[5] Budaghyan L., Kazymyrov O. Verification of Restricted EA-Equivalence for Vectorial Boolean Functions // Lecture Notes in Computer Science. — 2012. — V. 7369. — P. 108-118.

[6] TesaŘ P. A New Method for Generating High Non-linearity S-Boxes // Radioengineering. — 2010. V. 19, NO. 1. — P. 23-26.

[7] Izbenko Y., Kovtun V., Kuznetsov A. The Design of Boolean Functions by Modified Hill Climbing Method. — http://eprint.iacr.org/2008/111.pdf, 01.09.2013.

[8] Millan W., Clark A., Dawson E. Boolean Function Design Using Hill Climbing Methods // Lecture Notes in Computer Science Volume. — 1999. — V. 1587. P. 1-11.

[9] Daemen J., Rijmen V. AES Proposal: Rijndael. — http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf, 10.03.2013.

[10] Logachev O.A., Salnikov A.A., Yaschenko V.V. Boolean functions in coding theory and cryptology // MCCME, Moscow. — 2004. (In Russian)

[11] Yu Y., Wang M., Li Y. Constructing differential 4-uniform permutations from know ones. — http://eprint.iacr.org/2011/047.pdf, 10.03.2013.

[12] Technical details of Hexagon. — https://www.notur.no/hardware/hexagon, 01.09.2013.