Combinatorial method for Boolean SAC functions designing

Dr. N.G. BARDIS[1,2]
[1]Adjunct Assistant Professor
Department of Automation
Technological Education Institute of Halkis
34400 Psahna, Halkis, Evia, Greece

[2]Research Associate
Hellenic Army Academy
Vari, 16673 Attiki, Greece

Abstract: - In this paper a new approach for designing Boolean functions that satisfy the Strict Avalanche Criterion (SAC) is presented. The advantage of the suggested approach is the simplicity of its realization and the significant greater number of the generated functions compared to the known methods. The formalized procedure for construction of nonbalanced and balanced SAC-functions is described in detail; and examples of function design are given.

Key-Words: - Boolean functions, balance Boolean functions, SAC Boolean functions.

1 Introduction

Recent advancements in information security techniques makes the application and further development of Boolean functions theory extremely important and necessary since it covers a significant part of contemporary cryptographic algorithms. Development of new cryptographic arrangements for information security in computer networks and mobile communication systems requires solving for some theoretical and engineering problems. These problems are associated with the design and application of Boolean functions that correspond to the total and conditioned entropy maximum criteria or, in other words, the balanced functions that satisfy the Strict Avalanche Criterion (SAC).

The class of defined Boolean functions is the basis for obtaining so-called one-way functional transforms, i.e. those whose inverse transform cannot be performed and which are widely adopted by modern information technologies. The problem of finding the roots of the nonlinear Boolean equation system is part of a category of mathematical problems that cannot be solved analytically. In fact, the only way to find the inverse transform is by searching. In the case that these equations consist of functions that satisfy the total and conditioned entropy criteria, complete searching should be performed to solve the inverse transform problem. This is an impossible effort in practice due to the rather large number of variables.

In practice, in order to use balanced Boolean functions that correspond to the Strict Avalanche criterion, some formalized methods of their design should be worked out.

2 Principal definitions and properties of SAC-functions

A Boolean function $f(x_1,\ldots, x_n)$ of n variables is defined on 2^n possible tuples of their values that compose a set Z, which assumes values from the set {0,1}. The function satisfies the maximal total entropy criterion, i.e. is balanced, if there is equal probability that it will take the values of zero or one:

$$\sum_{x_1,\ldots,x_n \in Z} f(x_1,\ldots,x_n) = 2^{n-1} \quad (1)$$

A Boolean function $f(x_1,\ldots, x_n)$ satisfies the maximum conditioned entropy criterion, i.e. the Strict Avalanche Criterion (SAC), if there is a 50% probability that complementing a single input bit it results in changing the output bit.

$$\forall x_j, j = 1,\ldots,n: \quad \sum_{x_1,\ldots,x_n \in Z} f(x_1,\ldots,x_j,\ldots,x_n) \oplus f(x_1,\ldots,\bar{x}_j,\ldots,x_n) = 2^{n-1} \quad (2)$$

Webster and Tavares introduced the notion of the Strict Avalanche Criterion for Boolean functions for the first time in connection with the study of design of S-boxes of DES construction. Since then the SAC-function design problem has been vividly discussed in relevant literature. For practical use, the functions are required to satisfy total and conditioned entropy and have a high degree of

nonlinearity.

In this case nonlinearity – $N(f(x_1,\ldots,x_n))$ of the Boolean function $f(x_1,\ldots,x_n)$ is determined as the minimal Hamming's distance to the linear functions:

$$N(f(x_1,...,x_n)) = \min_{a_k \in \{0,1\}, k=0,...,n} \sum_{x_1 \ldots x_n \in Z} (f(x_1,...,x_n) \oplus (a_0 \oplus \bigoplus_{j=1,...,n} a_j \cdot x_j)) \quad (3)$$

The characteristics of Boolean function's autocorrelation are closely connected with the SAC effect. The Boolean function's correspondence to the SAC, is that its autocorrelation coefficient is equal to zero upon the change of a variable. This property is of great significance for pseudo-random sequence generators. The concept of the SAC criterion may be extended in principle over a greater number of variables – functions that have a 50% chance that their value will change upon inversion of k input variables are called PC(k)-functions [2]. Correspondingly, such a function has a zero correlation coefficient upon inversion of k variables. If k = n, a function PC(n) that has a 50% chance it will change its output upon the inversion of all the n variables is called a bent-function. It was proved [6], that the bent–function's nonlinearity is the maximum possible, but in principle it cannot be a balanced one. The notion of a bent-function may be only defined for even values of n and its nonlinearity is equal to:

$$N(f_b(x_1,...,x_n)) = 2^{n-1} - 2^{n/2-1} \quad (4)$$

For balanced Boolean functions $f(x_1,\ldots,x_n)$ of n variables the value of nonlinearity N(f) with the constraint n>3 has limit superior [6]:

$$N(f) \le 2^{n-1} - 2^{n/2-1} - 2 \text{ for even n} \quad (5)$$

$$N(f) \le \lfloor 2^{n-1} - 2^{n/2-1} \rfloor \quad \text{for odd n}$$

where $\lfloor x \rfloor$ is the maximal integer which is less than or equal to x.

For practical purposes, the SAC-functions that may be considered as PC(1)-functions have the widest application. Particularities of balanced Boolean SAC-functions manifest themselves in specific properties of their spectrum [3]. To obtain the spectrum $F(w_1,\ldots,w_n)$ of the Boolean function $f(x_1,\ldots,x_n)$, the direct Walsh transform should be performed according to the following formula:

$$F(\overline{W}) = \sum_{\overline{X} \in Z} f(\overline{X}) \cdot (-1)^{\overline{X} \cdot \overline{W}} \quad (6)$$

The inverse Walsh transform, that is obtaining the Boolean function $f(x_1,\ldots,x_n)$ by its spectrum $F(w_1,\ldots,w_n)$, is achieved through to the formula:

$$f(\overline{X}) = 2^{-n} \cdot \sum_{\overline{W} \in Z} F(\overline{W}) \cdot (-1)^{\overline{X} \cdot \overline{W}} \quad (7)$$

The Boolean function $f(x_1,\ldots,x_n)$ correspondence to the SAC-criterion may be determined by its spectrum properties $F(w_1,\ldots,w_n)$: a function $f(x_1,\ldots,x_n)$ is a SAC if and only if its spectrum $F(w_1,\ldots,w_n)$ satisfies the condition:

$$\sum_{\overline{W} \in Z} (-1)^{\overline{W}} \cdot F^2(\overline{W}) = 2^n \cdot F(0,...,0) - 2^{2 \cdot n-2} \quad (8)$$

Note that the meaning of the spectrum $F(0,\ldots,0)$ equals the number of ones in the truth table of the function $f(x_1,\ldots,x_n)$, i.e. the function $f(x_1,\ldots,x_n)$ may be called balanced if its spectrum on the zero tuple $F(0,\ldots,0)$ is equal to 2^{n-1} . Taking this into account, a Boolean function $f(x_1,\ldots,x_n)$ is balanced and corresponds to SAC if its spectrum $F(w_1,\ldots,w_n)$ holds the condition:

$$\sum_{\overline{W} \in Z} (-1)^{\overline{W}} \cdot F^2(\overline{W}) = 2^{2n-1} \quad (9)$$

Thus, if a Boolean function $f(x_1,\ldots,x_n)$ corresponds to the total and conditioned entropy maximum criteria, i.e. it is a balanced SAC-function, then the sum of its spectrum $F(w_1,\ldots,w_n)$ squares, which may be considered as analogous to the energy spectrum, has the maximal value [3].

For practical applications of balanced SAC-functions and, in particular, for pseudorandom function generator design [1,4,5], the task of k-th order balanced functions synthesis arises.

A Boolean function $f(x_1,\ldots,x_n)$ corresponds to the Strict Avalanche Criterion of the k-th order (SAC of the k-th order) if a Boolean function $h(x_1,\ldots,x_{n-k})$ into which $f(x_1,\ldots,x_n)$ is transformed at fixed values (zero or one) of its any k variables also corresponds to the Strict Avalanche Criterion.

3 Contemporary State of Balanced functions Effective Design

In practice, the most pressing problem is designing a method to automate the synthesis of Boolean balanced SAC-functions. Most applications require obtaining orthogonal systems of such functions with high nonlinearity and from a large number of variables (in the order of hundreds). Due to the large number of variables, the design methods for balanced Boolean functions suitable for practical application have to operate with the Algebraic Normal Form (ANF) of the functions rather than with their truth tables. The latter requires memory capacity that is beyond contemporary engineering resources available today.

In practice, the most important quality criteria in the design procedures of balanced Boolean SAC-functions are the following:

- The amount of allocated computational and memory resources in the design process;
- The number of n variable functions that can be designed ('till now the problem of the total number of balanced SAC-functions determination for n variables remains open [4]);

Due to the importance placed on the automated design of balanced SAC-functions for modern information processing, a number of approaches have been suggested during the last decade.

Historically, the first methods for obtaining balanced Boolean functions were those suggested by Forre R. [3], on the basis of which are laid the properties stated above for this class of functions. Analysis of formula (9) reveals, that it is possible, in principle, to construct all the spectra $F(w_1,\ldots,w_n)$ for which condition (9) is held. All the balanced SAC-functions may be obtained through the inverse Walsh transform of each constructed spectrum $F(w_1,\ldots,w_n)$ using (7). However, it should be stated [3] that a real Boolean function $f(x_1,\ldots,x_2)$ does not correspond to each spectrum $F(w_1,\ldots,w_n)$ that satisfies condition (9). In order to decrease the number of non-productive inverse transforms (7) and to assure high degrees of nonlinearity, it was suggested in [3] to somehow find a balanced SAC-function $f(x_1,\ldots,x_n)$ and to obtain its spectrum $F(w_1,\ldots,w_n)$ by Walsh transform. It was further suggested that the family of the spectra $F_1(w),\ldots,F_h(w)$, $h<2^n$, for which (9) is held and for which the real balanced SAC-functions corresponds, should be obtained through alteration of the signs of the components $F(w_1,\ldots,w_n)$ in an arbitrary way on all the 2^n tuples $w_1,\ldots,w_n$. The real balanced SAC-functions may be obtained by inverse Walsh transform.

From a processing aspect, the Forre method does not correspond to the requirements imposed above for the design of balanced SAC-functions. This, because it operates with a function's truth tables and the spectra's value tables whose capacity is proportional to 2^n. The inversion of the Walsh transform to expression (7) demands intensive computer time that is also proportional to 2^n.

Balanced SAC-functions of high nonlinearity may be obtained by de-concatenation of a bent-function [6], however obtaining the bent-functions themselves from a large number of variables is a rather difficult problem whose solution requires substantial computational and memory resources.

The recursive process of obtaining balanced Boolean SAC-functions of n variables using four non-balanced SAC-functions of n-1 variables demands much less resources [1]. The greatest disadvantage of this method is the fact that it only makes it possible to obtain a rather small amount of the total balanced SAC-functions.

An analytical design method of balanced SAC functions was suggested by Kurosawa K. and Satoh T. [4]. In essence, the method's idea consists of dividing n variables into two non-overlapping sets with s and t variables (n=s+t). Further on, a linear function $g(x_1,\ldots,x_s)$ of s variables and a binary matrix Q with dimensionality equal to s × t are formed. In so doing, the number of one-components of the product $Q\cdot\gamma_1$ of matrix Q by any s-component vector γ_1 with one non-zero component and the product $\gamma_2\cdot Q$ of any t-component vector γ_2 with one non-zero component is more than or equals one. The vector formed by the coefficients of the function $g(x_1,\ldots,x_s)$ is to be linear-independent of the vectors formed by the columns of matrix Q. A balanced SAC-function is formed according to the formula:

$$f(x_1,\ldots,x_n)=[x_1,\ldots,x_s]\cdot Q\cdot \cdot[x_{s+1},\ldots,x_n]^T\oplus g(x_1,\ldots,x_s) \quad (10)$$

The drawback of this method lies in the difficulty of finding a matrix Q at rather high values of n. Great resources are required to find the vector system that is not linearly dependent on the coefficient vector of the linear function Q. A better method to obtain balanced SAC-functions was proposed in [5]. The principal weakness of this method – as the method, which proposed in [4]- is that it allows only a small number of balanced SAC-functions to be obtained from the whole amount of variables.

The short review of the existing approaches to the design problem of the balanced SAC-functions shows that the published methods do not fully correspond to the above stated criteria and that is why development of more effective formal SAC-function design procedures is actually and practically important.

4 Combinatorial method for SAC-functions designing

Figures and Tables should be numbered as follows: Fig.1, Fig.2, … etc and Table 1, Table 2, ….etc.

For designing functions which correspond to the SAC it is proposed to use a system $\Omega=\{\varphi_1,\varphi_2,\ldots,\varphi_q\}$ of basic functions with t variable, that correspond to the specified criterion. The functions $\varphi(x_1,x_2,\ldots,x_t)$, that are included in the system must satisfy two requirements:

First, correspond to the Strict Avalanche Criterion with respect to t variables:

$$\forall j \in \{1,...,t\}:$$
$$H(\varphi_i(x_1,...,x_j,...,x_t) \oplus \varphi_i(x_1,...,x_j \oplus 1,...,x_t)) = 2^{t-1} \quad (11)$$

Second, the function, formed as XOR by any pair of functions belonging to the system Ω must be balanced:

$$\forall \varphi_i, \varphi_j \in \Omega, \varphi_i \neq \varphi_j : H(\varphi_i \oplus \varphi_j) = 2^{t-1} \quad (12)$$

In the simplest case the system Ω, can be selected SAC - functions of t variables with the minimum Hamming weight (minimum number of ones in the truth table), which take the value "one" on different sets.

Since the minimum Hamming weight for SAC-functions of t variables is equal to 2^{t-2}, the function formed as XOR by any pair of such functions will have in the truth table 2^{t-1} ones, i.e., it will be balanced.

It is also obvious, that the number of functions belonging to the system Ω is always equal to 4.

In the Table 1 are shown the amount of SAC-functions with minimum Hamming weight when $t \in [2..4]$.

For example for t=2 there exist only 4 Boolean SAC- functions, which satisfy the given conditions. Their representation is shown below in the form of values sequence on the sets 00, 01, 10, 11:

φ_1=0001, φ_2=0010, φ_3=0100, φ_4=1000

Accordingly, there is only one version of selecting the system Ω of basic functions.

For t=3 there are already 16 SAC- functions with minimum Hamming weight 2^{t-2}=2, and their systems, which satisfy the above given requirements - 250. As an example one of these systems is given below for t=3:

φ_1=0 1 1 0 0 0 0 0,
φ_2=1 0 0 1 0 0 0 0
φ_3=0 0 0 0 0 1 1 0,
φ_4=0 0 0 0 1 0 0 1

The system below is an example of system Ω of basic functions for t=4 :

φ_1=1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0
φ_2=0 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0
φ_3=0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0
φ_4=0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 1

Table 1

Number of t variables	Total number of SAC functions with minimum Hamming weight	Number of possible systems Ω
2	4	1
3	16	250
4	148	23280
5	22632	$\approx 10^7$

The essence of the proposed method lies on the fact that the truth table of the Boolean function $f(x_1,x_2,\ldots,x_n)$ from n variables, that correspond to the SAC, is constructed from 2^{n-t} fragments, with every h^{th}, for which, $h \in \{0,\ldots,2^{n-t}\}$ - $\phi_h(x_1,\ldots,x_t)$ is one of the basic functions φ_1, φ_2, φ_3, φ_4 or their inversions:

$\phi_h \in \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_1 \oplus 1, \varphi_2 \oplus 1, \varphi_3 \oplus 1, \varphi_4 \oplus 1\}$.
$\varphi_1, \varphi_2, \varphi_3, \varphi_4$

In other words, the function $f(x_1,\ldots,x_n)$ is constructed in the form of the superposition of the functions, which belong to the extended basic system on all possible sets of variables $x_{t+1},\ldots,x_n$.

Lets denote as ψ_h the function, which has value of one on the h^{th} set X_h of variables $x_{t+1},\ldots x_n$, i.e.:

$$\psi_h = \prod_{j=t+1}^{n} (x_j \oplus a_j^h) : \sum_{j=t+1}^{n} (1 - a_j^h) \cdot 2^{j-t-1} = h \quad (13)$$

the Boolean SAC- function $f(x_1,x_2,\ldots,x_n)$ then is formed as:

$$f(x_1,...,x_n) = \bigoplus_{h=0}^{2^{n-t}} \phi_h(x_1,...,x_t) \cdot \psi_h(x_{t+1},...,x_n) \quad (14)$$

Function (14) satisfies SAC with respect to the variable x_i, $i \in \{1,\ldots,t\}$ because of the fact that this property is held by the functions ϕ of the extended basis Θ which are contained in (14).

$$H(f(x_1,...x_i...x_n) \oplus f(x_1...x_j \oplus 1...x_n) =$$
$$= \sum_{h=0}^{2^{n-t}-1} \psi_h(x_{t+1}...x_n) \cdot H(\phi_h(x_1...x_i...x_t) \oplus \phi_h(x_1...x_i \oplus 1...x_t)) =$$
$$= \sum_{h=0}^{2^{n-t}-1} \psi_h(x_{t+1}...x_n) \cdot 2^{t-1} = 2^{n-t} \cdot 2^{t-1} = 2^{n-1} \quad (15)$$

The Boolean function $f(x_1,\ldots,x_n)$ constructed with the above method satisfies the SAC with respect to variable x_j, $j \in \{t+1,\ldots,n\}$, only when the following conditions are fulfilled.

$$H(f(x_1...x_j...x_n) \oplus f(x_1...x_j \oplus 1...x_n) =$$
$$= \sum_{h=0}^{2^{n-t}-1} \sum_{x_1 \cdots x_t \in Q} (\psi_h(x_{t+1}...x_j...x_n) \cdot \phi_h(x_1...x_t) \oplus$$
$$\oplus \psi_h(x_{t+1}...x_j \oplus 1...x_n) \cdot \phi_h(x_1...x_t)) =$$
$$= \sum_{h=0}^{2^{n-t}-1} \psi_h(x_t...x_j...x_n) \cdot H(\phi_h(x_1...x_t) \oplus \phi_{v(h,j)}(x_1...x_t)) = 2^{n-1} \quad (16)$$

where, $v(h,j) = h + (1 - 2 \cdot a_j^h) \cdot 2^j$.

Thus, in order for the Boolean function $f(x_1,\ldots,x_n)$ constructed in accordance with (14) to satisfy the

strict avalanche criterion it is necessary that the functions ϕ_k and ϕ_l selected as the fragments of the constructed function on k^{th} and l^{th} sets of the values of variables $x_{t+1},\ldots,x_n$ ($k,l\in\{0,\ldots,2^{n-t}-1$) would be different, if k and l differ in the value $|k-l|=2^g$ $g\in\{0,1,\ldots,n-t-1\}$. This means, for k^{th} and l^{th} sets of variable $x_{t+1},\ldots,x_n$ must be fulfilled : $\phi_k \neq \phi_l$, $\phi_k \neq \phi_l \oplus 1$.

Based on the theoretical statements the following method of obtaining Boolean functions that satisfy the SAC is proposed.

The number t of the variables for the basic system Ω is selected, then arbitrarily one of the possible basic systems $\Omega=\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ is selected followed by the construction of its extension

$\Theta=\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_1 \oplus 1, \varphi_2 \oplus 1, \varphi_3 \oplus 1, \varphi_4 \oplus 1\}$.

The basic system $\Omega=\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ of functions is arbitrarily divided into two non-empty and not intersecting subsets $\Omega_1\subset\Omega$ and $\Omega_2\subset\Omega$: $\Omega\neq\varnothing$, $\Omega_1\neq\varnothing$, $\Omega_1\cap\Omega_2=\varnothing$, $\Omega_1\cup\Omega_2=\Omega$.

It is obvious that there are 14 ways for the division.

The truth table of the generated function $f(x_1,x_2,\ldots,x_n)$ is built in the form of 2^{n-t} fragments with 2^t bits, which are the functions that belong to basic system Ω.

In so doing the h^{th} fragment $h\in\{0,\ldots,2^{n-t}-1\}$ corresponds to the h^{th} set X_h of the values of variables $x_{t+1},\ldots,x_n$.

If the XOR of the values of variables $x_{t+1},\ldots,x_n$ on the h^{th} set equals to zero, then as the h^{th} fragment ϕ_h any function belonging to Ω_1 is selected, otherwise is selected a function belonging to Ω_2

In other words:

if $\bigoplus_{j=t+1}^{n}(x_j \oplus a_j^h)=0 : \phi_h \in \Omega_1$, and

if $\bigoplus_{j=t+1}^{n}(x_j \oplus a_j^h)=1 : \phi_h \in \Omega_2$.

The obtained in the described way functions ϕ_h of permutation with any h^{th} fragment can be replaced with their inversion $\phi_h \oplus 1$.

We will show, that the proposed method ensures the fulfilment of conditions: $\phi_k \neq \phi_l$ for $|k-l|=2^g$, $k,l\in\{0,\ldots,2^{n-t}-1\}$, $g\in\{0,1,\ldots,n-t-1\}$.

From the fact that the binary representations of k and l are different in one bit only follows that $\bigoplus_{j=t+1}^{n}(x_j \oplus a_j^k) \neq \bigoplus_{j=t+1}^{n}(x_j \oplus a_j^l)$.

In accordance with the proposed method this indicates that the functions ϕ_k and ϕ_l of filling the k^{th} and l^{th} fragments belong to different sets Ω_1 and Ω_2, and consequently: $\phi_k \neq \phi_l$.

Thus, the proposed method of filling the fragments ensures that for any pair k and l which are different in one bit the condition $\phi_k \neq \phi_l$ is fulfilled - necessary so that the function $f(x_1,\ldots,x_n)$ would correspond to SAC.

The proposed method ensures the fact that the formulated above condition for the correspondence to SAC of the Boolean functions designed according to (14) is held. Consequently, the Boolean function $f(x_1,\ldots,x_n)$ designed by the presented method corresponds to SAC.

The proposed method is illustrated by the following example of synthesising a function, which satisfies the SAC with 7 variables (n=7). Let the number t of variables for the basic system Ω to be 3, (t=3), and system Ω itself consists of SAC- functions

φ_1=0 1 1 0 0 0 0 0,

φ_2=1 0 0 1 0 0 0 0

φ_3=0 0 0 0 0 1 1 0,

φ_4=0 0 0 0 1 0 0 1

The resulting function $f(x_1,\ldots,x_7)$ is constructed in the form of the superposition of $2^{7-3}=16$ of the functions $\phi_0,\phi_1,\ldots,\phi_{15}\in\Omega$ with x_1,x_2,x_3 variables on all possible sets of values of variables $x_4,\ldots x_7$. In accordance with the above system Ω is divided into two nonintersecting subsets: $\Omega_1=\{\varphi_2\}$ and $\Omega_2=\{\varphi_1, \varphi_3, \varphi_4\}$.

The condition $\bigoplus_{j=t+1}^{n}(x_j \oplus a_j^h)=0$ is satisfied for $h\in\{0,3,5,6,9,10,12,15\}$ respectively, these fragments are filled with the function $\varphi_2\in\Omega_1$, and the rest with any of the functions $\varphi_1, \varphi_3, \varphi_4\in\Omega_2$. Let on the arbitrarily selected set {5,7,12,13,14,15} the filling functions are replaced with their inversions.

The resulting function $f(x_1,\ldots,x_7)$ is represented in the form of the superposition of fragments, presented in Table 2.

Table 2

h	ϕ_h	h	ϕ_h	h	ϕ_h	h	ϕ_h
0	φ_2	4	φ_1	8	φ_3	12	$\varphi_2\oplus1$
1	φ_1	5	$\varphi_2\oplus1$	9	φ_2	13	$\varphi_3\oplus1$
2	φ_4	6	φ_2	10	φ_2	14	$\varphi_4\oplus1$
3	φ_2	7	$\varphi_1\oplus1$	11	φ_1	15	$\varphi_2\oplus1$

The Algebraic Normal Form (ANF) of the synthesized function is presented:

$f(x_1,\ldots,x_7)=1\oplus x_1\oplus x_2\oplus x_3\oplus x_4\oplus x_5\oplus x_6\oplus x_7\oplus x_1\cdot x_2\oplus x_1\cdot x_5\oplus x_1\cdot x_6\oplus x_1\cdot x_7\oplus x_2\cdot x_4$

$\oplus x_2\cdot x_5\oplus x_3\cdot x_6\oplus x_3\cdot x_7\oplus x_4\cdot x_5\oplus x_5\cdot x_6\oplus x_5\cdot x_7\oplus x_3\cdot x_4\cdot x_5\oplus x_3$

$\cdot x_4 \cdot x_6 \oplus x_3 \cdot x_4 \cdot x_7 \oplus x_2 \cdot x_3 \cdot x_5 \oplus$
$x_2 \cdot x_3 \cdot x_6 \oplus x_2 \cdot x_3 \cdot x_7 \oplus x_1 \cdot x_4 \cdot x_6 \oplus x_1 \cdot x_4 \cdot x_7 \oplus x_1 \cdot x_3 \cdot x_5 \oplus x_1 \cdot x_2 \cdot x_7 \oplus x_1 \cdot x_2 \cdot x_6 + \oplus x_1 \cdot x_2 \cdot x_4 \oplus x_2 \cdot x_3 \cdot x_4 \cdot x_6 \oplus x_2 \cdot x_3 \cdot x_4 \cdot x_7 \oplus x_2 \cdot x_3 \cdot x_4 \cdot x_5 \oplus x_1 \cdot x_3 \cdot x_4 \cdot x_5 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_7 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_6$.

The synthesized function satisfies the SAC, is not balanced and has nonlinearity, equal to 40.

In order for the synthesized by the presented above method Boolean function $f(x_1,\ldots,x_n)$ to be additionally balanced, it is necessary that the following condition is satisfied:

$$\sum_{h=0}^{2^{n-t}-1} \psi_h(x_{t+1}\ldots x_n) \cdot \sum_{x_1\ldots x_t \in U} \phi_h(x_1\ldots x_t) = \quad (17)$$
$$= \sum_{h=0}^{2^{n-t}-1} \sum_{x_1\ldots x_t \in U} \phi_h(x_1\ldots x_t) = 2^{n-1}$$

This condition can be satisfied, if among 2^{n-t} functions $\phi_0, \phi_1, \ldots, \phi_{2^{n-t}-1}$ are exactly 2^{n-t-1} basic functions, which belong to Ω and exactly 2^{n-t-1} function- inversions. A version of fragments selection, which ensures obtaining a balanced function from 7variables, that satisfies the SAC with $\Omega_1 = \{\varphi_2\}$ и $\Omega_2 = \{\varphi_1, \varphi_3, \varphi_4\}$ is given in Table 3.

Table 3.

h	ϕ_h	h	ϕ_h	h	ϕ_h	h	ϕ_h
0	φ_2	4	φ_1	8	φ_3	12	$\varphi_2 \oplus 1$
1	$\varphi_1 \oplus 1$	5	$\varphi_2 \oplus 1$	9	φ_2	13	$\varphi_3 \oplus 1$
2	$\varphi_4 \oplus 1$	6	$\varphi_2 \oplus 1$	10	φ_2	14	$\varphi_4 \oplus 1$
3	φ_2	7	$\varphi_1 \oplus 1$	11	φ_1	15	φ_2

The Algebraic Normal Form (ANF) of the synthesized function is:

$f(x_1,\ldots,x_7) = 1 \oplus x_1 \oplus x_2 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_1 \cdot x_2 \oplus x_1 \cdot x_3 \oplus x_1 \cdot x_4 \oplus x_1 \cdot x_5 \oplus x_1 \cdot x_6 \oplus x_1 \cdot x_7 \oplus x_2 \cdot x_5 \oplus x_3 \cdot x_6 \oplus x_3 \cdot x_7 \oplus x_4 \cdot x_5 \oplus x_5 \cdot x_6 \oplus x_5 \cdot x_7 \oplus x_1 \cdot x_2 \cdot x_6 \oplus x_1 \cdot x_2 \cdot x_7 \oplus x_1 \cdot x_3 \cdot x_5 \oplus x_1 \cdot x_4 \cdot x_6 \oplus x_1 \cdot x_4 \cdot x_7 \oplus x_2 \cdot x_3 \cdot x_4 \oplus x_2 \cdot x_3 \cdot x_5 \oplus x_2 \cdot x_3 \cdot x_6 \oplus x_2 \cdot x_3 \cdot x_7 \oplus x_3 \cdot x_4 \cdot x_5 \oplus x_3 \cdot x_4 \cdot x_6 \oplus x_3 \cdot x_4 \cdot x_7 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_6 \oplus x_1 \cdot x_2 \cdot x_3 \cdot x_7 \oplus x_1 \cdot x_3 \cdot x_4 \cdot x_5 \oplus x_2 \cdot x_3 \cdot x_4 \cdot x_5 \oplus x_2 \cdot x_3 \cdot x_4 \cdot x_6 \oplus x_2 \cdot x_3 \cdot x_4 \cdot x_7$.

The synthesized function satisfies the SAC, is balanced and has nonlinearity, equal to 48.

5. Estimation of the synthesized functions amount

The estimation of the amount of SAC- functions, which can be synthesized by the proposed method, plays an important role. The method assumes the selection of set Ω with 4 base functions. The functions, which compose the set Ω, are divided into two non-empty subsets Ω_1 and Ω_2.

The filling of the fixed $q=2^{n-t-1}$ fragments of the truth table is done with the functions of set Ω_1 and the other 2^{n-t-1} fragments with the functions of set Ω_2.

There are 4 versions of the selection of set Ω_1 when it contains one function, 6 versions of selecting Ω_1 when it contains two functions and 4 versions - when set Ω_1 contains three basic functions.

If set Ω_1 contains one function, then exactly one version of filling the q fragments of the synthesized function's truth table exists.

The number of methods of filling the other q fragments of truth table with the functions of set Ω_2 is 3^q, since any of the $q=2^{n-t-1}$ fragments can be filled with any of the functions of set Ω_2.

If the corresponding q fragments of the truth table are filled with two functions of set Ω_1 (which is possible under the condition $q=2^{n-t-1} \geq 2$) the calculation of the possible versions amount is done as follows.

Two of the q fragments must in any case be filled with two functions from Ω_1. The selection of the fragments pair from q can be executed with $\binom{q}{2} = \frac{(2^{n-t-1})!}{2 \cdot (2^{n-t-1}-2)!}$ methods, and the filling of the selected pair - by two methods.

The rest q -2 of fragment, are filled up with any functions of set Ω_1, which can be executed 2q-2 by versions. The number of variants (methods) of filling other q of the fragments of truth table by the functions of set Ω_1 is composes 2q.

In case when the corresponding q fragments of the truth table are filled with three functions of set Ω_1 (which is possible only under condition $q=2^{n-t-1} \geq 3$) the calculation of the possible versions quantity is done analogously.

Three of the q fragments must in any case be filled by two functions from Ω_1. The selection of three fragments from q can be executed with $\binom{q}{3} = \frac{(2^{n-t-1})!}{6 \cdot (2^{n-t-1}-3)!}$ methods, and the filling of the selected three fragments with three functions- by six methods.

The rest of the q -3 fragments, are filled with any functions of set Ω_1, which can be executed by $\mathbf{3^{q-3}}$ versions. The other q fragments of the truth table can be filled uniquely, since the set Ω_2 in this case contains only one function.

Thus, the amount of versions W of filling the 2^{n-t} fragments of the truth table with 4 fixed basic functions from t variables is determined depending on value $q=2^{n-t-1}$ in the following way:

$$W = 4 \cdot 3^q = 12 : q = 1, (n - t = 1)$$
$$W = 4 \cdot 3^2 + 6 \cdot 2 \cdot 2^2 = 84 : q = 2, (n - t = 2)$$
$$W = 4 \cdot 3^q + 6 \cdot \frac{q!}{(q-2)!} \cdot 2^{2 \cdot q - 2} + 4 \cdot \frac{q!}{(q-3)!} \cdot 3^{q-3} : q \geq 3 (n - t > 3) \quad (18)$$

When synthesising by the proposed method SAC-functions (balanced and non-balanced) from n variables, each of the the synthesized function's $2 \cdot q = 2^{n-t}$ fragments can be filled with basic function or its inversion, i.e., the total amount K_{SAC} of the SAC-functions, that can be synthesized by the proposed method with fixed selection of the basic set is determined by the following formula:

$$K_{SAC} = W \cdot 2^{2 \cdot q} = W \cdot 2^{2^{n-t}} \quad (19)$$

In the case of synthesis of balanced SAC- functions from n variables, exactly half of the $2 \cdot q = 2^{n-t}$ fragments of the truth table must be filled with the basic set functions, and the other half – by the inversions of the basic functions.

So, the total number of versions of the inverted and non-inverted fillings of the $2 \cdot q$ fragments is $\left(\frac{2 \cdot q}{q} \right) = \frac{(2 \cdot q)!}{(q!)^2}$.

Accordingly, the total number K_{BSAC} of balanced SAC- functions, which can be synthesized by the proposed method with fixed set Ω of basic functions is determined by the formula:

$$K_{BSAC} = W \cdot \left(\frac{2 \cdot q}{q} \right) = W \cdot \frac{2^{n-t}!}{(2^{n-t-1}!)^2} \quad (20)$$

For example, with n=4 and t=2 the proposed method makes it possible to synthesize 1344 SAC- functions from the total number 4120 of the existing functions, that satisfy the SAC and 504 of 1367 existing balanced SAC- functions.

It is completely obvious from the given formulas (19,20) that the maximum amount of SAC-functions can be obtained at the smallest value of t=2.

6 Conclusion

The proposed method of designing Boolean functions of special classes is based on the principles of combinatory transpositions and makes it possible to obtain both balanced and non-balanced functions that satisfy the Strict Avalanche Criterion - SAC. The majority of proposed methods currently provide for obtaining only balanced functions of this class.

In comparison with the known methods, the proposed approach is technologically simpler and its realization requires substantially smaller computational recourses, since it does not use the complex operations of the spectral Walsh transforms, search for the linearly independent vectors or bent - functions. According to the conducted experimental investigations, the productivity of the software realization of the developed approach compared to the Kurosawa K. and Satoh T method [4] is higher by approximately 3 orders.

The basic advantage of the proposed combinatorial approach compared to the known ones is the substantially greater number of functions, which can be generated. In particular, for 4 variables the known methods can generate 72 (5%) [4] or 96(7%) [1] of balanced functions from the possible 1368. The proposed method allows to make and obtain 504 functions, or 37%.

References:

[1] Bardis E.G., Bardis N.G., Markovskyy A.P., Spyropoulos A.K. Design of Boolean Function from a Great Number of Variables Satisfying Strict Avalanche Criterion.// *Recent Advances in Signal Processing and Communication.WSES*,1999. pp.107-112.

[2] Cusic T.W. On construction of balanced correlation immune function, in sequences and their application. // *Proceeding of SETA'98-Springer Discrete Mathematics and Theoretical Computer Sciences*,-1999-pp.184-190.

[3] Forre R. The strict avalanche criterion: spectral properties of Boolean functions and extend definition // *Advances in Cryptology – Crypto'88 Proceeding, Lecture Notes in Computer Sciences,403* – 1990 pp.450-468.

[4] Kurosawa K., Satoh T. Design of SAC/PC(l) of Order k Boolean Functions and Three Other Cryptographic Criteria.// *Advances in Cryptology –Eurocrypto'97 Proceeding, Lecture Notes in Computer Science* 1233-1997-pp.433-449.

[5] Polymenopoulos A., Bardis E.G., Bardis N.G., Markovskaja N.A., "Design and Implementation of Boolean Balanced Functions Satisfying Strict Avalanche Criterion (SAC)", *WSES - International Conference on Problems in Applied Mathematics and Computational Intelligence*, ISBN: 960-8052-30-0, 2001, pp. 12-16.

[6] Webster A.F., Tavares S.E. On the design of S-boxed. // *Advances in Cryptology – Crypto'85, Proceeding, Notes in Computer Science,* 332 – 1986 pp.523-535.