International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184, Volume 8, Number 3 (June 2019), pp.40-46

http://www.meacse.org/ijcar

Method of Synthesis for Controlled Generators Balanced SAC-Function

Ali T. AL-Khwaldeh
College of Engineering, Jordan Philadelphia University

Abstract

This paper deals with a controlled Balanced Boolean Function as a construction method to satisfy Strictly Avalanche Criteria (SAC) conditions and effects. Three orthogonal nonlinear components are introduced in this paper, where they are all different than the high-order SAC function, as a result; the proposed generator synthesized algorithm has separate control sets and separate data inputs. In addition, the results show that the proposed technique is regarded as simple and easy to implement, which is justified by applying a fixed control and fixed data inputs over the handled synthesized SAC function generator. Moreover, the built-in oscillator efficiency becomes greater than the high-order SAC functions when it's used as a generator. As a result, we can claim that the proposed method is formalized and implemented to be available and ready used software commercial item.

Keywords: Built-in Oscillator, Controlled Balanced Boolean Function, and Orthogonal Nonlinear.

Introduction

The rapid development of computing systems and information resources integration in the modern computer networks illustrated the need for stimulating further development of the Boolean functions theory as proposed on the early 80s in [1] and the references therein, which underlie many important areas of modern information technologies. Moreover, Boolean functions facehigh pace development because of the software methods for implementing associative access (hash addressing) in databases and linguistic processes [2], methods of encoding and compressing information, and the means of protecting it in computer networks requires the solution of theoretical and technological problems of obtaining and using Boolean functions of special classes. The greatest practical values for listed applications are Boolean functions that meet the criteria for the maximum total and differential entropy, or in the other words, balance functions that satisfy the criterion of Strict Avalanche Criterion (SAC), or abbreviated effect [3].

This class of Boolean functions is in fact the basis for obtaining the so-called one-way functional transformations, that is, transformations for which the inverse transformation can't be realized, and which are widely used in modern information technologies. Indeed, the definition of nonlinear Boolean equations system roots refers to a class of mathematical problems that cannot be solved analytically [4]. Moreover, practically the only method of reverse transformation is busting, which makes it impossible to implement the inverse transformation in practice with many variables. This determines the wide use of SAC-functions in the information systems of algorithmic protection. In addition to information security systems, Boolean functions of this class can be effectively used for hashing, since it is the orthogonal systems of such functions that provide the minimum probability of collisions [2].

For the practical use of Boolean balanced functions that satisfy the criterion of a strict avalanche effect, the task is to develop formalized methods for their synthesis. In practical terms, the problem of synthesizing Boolean functions of these classes must be considered somewhat more widely; moreover, it is necessary to build code-controlled generators of such Boolean functions.

This will significantly increase the efficiency of using special class Boolean functions for many applications, which practically reduce the negative impact of grouping records in the hash memory by using a single reconfigurable mechanism of primary and secondary hash addressing [4] which shows a computational aspect of the quantum hashing technique. In [5] and the references there in, the evolutionary design of hash function pairs is proposed for network filters which require high speed of data processing needed for the filter applications.

The rapid progress of integrated technology allows creating effective hardware implementations of such reconfigurable functions using matrices of programmable elements.

Recently, many new researches have been introduced to apply the Boolean functions in such varying applications such as [6-9].

The rest of the paper is arranged as follows: section II introduces the basic definitions of the Boolean function, section III illustrates analysis of the current state of the problem**,** where the method for synthesizing the generator of balanced Boolean SAC functions is shown. In section IV, a synthesis example is explained, and finally section V concludes the paper.

Basic Definitions

A Boolean function $f(x_1, x_2, ..., x_n)$ is defined as a set of *n* variables consisting of two possible digits X_i, is said to be balanced if the number of ones are equal to the number of zeros, i. e, equal probability for the ones and zeros as shown in (1):

$$\sum_{X \in Z} f(X) = 2^{n-1} \qquad (1)$$

The Boolean function $f(x_1, x_2, ..., x_n)$ satisfies the criterion for a strict avalanche effect if, with a change in the value of any of the n variables, the value of the function changes with a probability of 50% as shown in (2):

$$\sum_{X \in Z} f(X) \oplus f(X \oplus \Delta_j) = 2^{n-1}, \forall j \in \{1,\ldots,n\}, \qquad (2)$$
$$\Delta_j = (d_1,\ldots,d_j,\ldots,d_n), d_j = 1, d_i = 0, \forall i \in \{1,\ldots,n\}, i \neq j,$$

Where Δ_j is the component dyadic vector whose j^{th}component is equal to one, and the remaining components are set to be zeroes. Any Boolean function $f(x_1, x_2, ..., x^n)$ can be represented as a canonical Shannon expansion with respect to the j^{th} variable x_j, accordingly, $f(x_1, x_2, ..., x_n)$ becomes as in (3)

$$f(x_1, x_2, ..., x_n) = x_j \cdot \varphi_j(x_1, ..., x_{j-1}, x_{j+1}, ..., x_n) \oplus \psi_j(x_1, ..., x_{j-1}, x_{j+1}, ..., x_n) \qquad (3)$$

Where φ_j andψ_j are the Boolean functions that do not depend on x_j which means that a function of n-1 variables for which there are 2n-1 possible sets of Xj variables x1, ..., xj-1, xj + 1, ..., xn forming the set Zj. Then, the condition for the correspondence of the function $f(x_1, x_2, ..., x_n)$ to the criterion of a strict avalanche effect can be reduced to the form shown in (4):

$$\sum_{X_j \in Z_j} \varphi_j(X_j) = 2^{n-2}, \forall j \in \{1,\ldots,n\} \qquad (4)$$

The function φ_j (x_1, ..., x_{j-1}, x_{j+1}, ..., x_n) is identified by a number of investigators [10] with the differential of the Boolean function, f (x_1, x_2, ..., x_n) as shown in (5):

$$\frac{\partial f(x_1,\ldots,x_j,\ldots,x_n)}{\partial x_j} = \varphi_j(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n) \quad (5)$$

Accordingly, the Boolean function f (x_1, x_2, ..., x_n) satisfies SAC if its differentials in all variables are balanced.A Boolean function g(x_1, x_2, ..., x_h, k_1, k_2, ... k_h), which is a generator of Boolean balanced SAC functions in n variables, can be reconstructed from the h-bit code K = {k_1, k_2, ..., k_h}, that for any value of the tuning code K, it is transformed into a balanced SAC function from n variables as shown in (6):

$$\forall k_i \in \{0,1\}, i = 1,..., h, \forall j \in \{1,..., n\} : \sum_{X \in Z} (g(X, k_1,..., k_h) \oplus g(X \oplus \Delta_j, k_1,..., k_h)) = 2^{n-1} \quad (6)$$

A special case of a tunable Boolean generator SAC function is the balanced SAC functions of the h^{th}order.The Boolean function, f (x_1, x_2, ..., x_{n+h}) of n+h variables is called a balanced SACfunction of the h^{th}order if for any values of an arbitrary subset h of variables from x_1, x_2, ..., x_{n+h}, it is transformed into balance SAC function from the remaining n variables.Let (θ) be a linear function representing the sum modulo 2 of all variables belonging to the set θ is given in (7):

$$\lambda(\theta) = \bigoplus_{x_k \in \theta} x_k \quad (7)$$

We denote by η(θ) the number of variables that make up the set θ.

Analysis of the current state of the problem

Since the beginning of the 90s,many works have been published on the synthesis of Boolean functions of special classes oriented for use in encryption and information security systems [5-7]. Methods to the synthesis of SAC functions can be divided into three groups:

1. Genetic;
2. Orthogonal basis;
3. Combinatorial.

Genetic methods for the synthesis of Boolean functions of special classes are since some functions of a class somehow turn out, and then a lot of functions having the same properties are formed from it by special mathematical transformations. For practical implementation of this approach, spectral transformations are most often used [3],[4]. A common drawback of genetic methods for synthesizing Boolean SAC functions c is that the problem of obtaining the primary Boolean function of this class remains open.

Another practical approach to generate the SAC function can be obtained by using the properties of the Boolean function orthogonal systems. Within the framework of this approach, most of the existing methods for synthesizing balanced SAC-functions have been developed [10] and [11]. The most known method of obtaining balance SAC-functions on the basis of orthogonal transformations is proposed in [12]. The essence of the method proposed by them is that the variables are divided into two disjoint sets, c and variables, then a linear function of the variables and a binary matrix are formed, the dimension, and the number of unit components of the product of the matrix on any A vector with one non-zero component and the product of any -component vector with one non-zero component is greater than or equal to one. The vector formed by the coefficients of the

function can be linearly independent of the vectors formed by the columns of the matrix. The balance SAC function is formed according to the formula shown in (8):

$$f(x_1,..., x_n) = [x_1,..., x_s] \cdot Q \cdot [x_{s+1},..., x_n]^T \oplus g(x_1,..., x_s) \quad (8)$$

The disadvantages of this method are the difficulty of obtaining a matrix for sufficiently large values. A common drawback of methods for synthesizing Boolean functions based on orthogonal transformations is the essentially small number of SAC-functions that can be obtained within the framework of this approach.Combinatorial methods for obtaining Boolean functions are based on the separation of variables into sets over which special mathematical operations are performed [2].

Most of the research is devoted to the synthesis of individual SAC-functions and does not concern the problem of constructing user-controlled generators of such functions. In several papers [2] and[11], methods for the synthesis of SAC-functions of order h were proposed. These functions can be used as over-generators of SAC-functions, which are re-built by the h-bit code. As controllers, any h of n+h input variables can be used. The disadvantage of this approach is functional redundancy, because in practice the information and control inputs of the function generator are fixed in the processes shown below:
The procedure for synthesizing SAC-functions of order h has an excessively high computational complexity;
S-functions of the h^{th} order have redundant complexity from the point of view of providing properties of functions generated with their help; The number of synthesized SAC-functions of order h is much less than the number of generators with fixed control inputs. Therefore, the approach to constructing a generator that considers the initial separation of control and information variables is more efficient.

Method for synthesizing the generator of balanced Boolean SAC-functions

The proposed method for synthesizing controlled generators of balanced Boolean SAC-functions belongs to the second group of methods in the above classification. It is based on transformations that perform over three nonlinear orthogonal Boolean functions that are synthesized according to certain rules. The method is the development of methods for synthesizing SAC functions, as described in [2] and [12].The essence of the method for synthesizing the function generator g (x_1, x_2, ..., x_{n+h}) of balanced Boolean SAC functions from n variables that is adjustable in h variables is to perform the following sequence of actions:

The set g = {x_1, x_2, ..., x_{n+h}} n+h variables on which the synthesized function g (x_1, x_2, ..., x_{n+h}) is divided into five disjoint subsets which are:

ϑ_1, Q_1, Δ, ϑ_2, Q_2: $\vartheta_1 \cup Q_1 \cup \Delta \cup \vartheta_2 \cup Q_2 = \Omega$, причем $\vartheta_1 \neq \varnothing$, $\Delta \neq \varnothing$, $\vartheta_2 \neq \varnothing$ и $\eta(Q_1 \cup Q_2) \geq h$.Three Boolean functions B_0, B_1 and B_2 are formed as follows in (9):

$$\begin{aligned} B_0 &= \bigoplus_{x_j \in \Delta} x_j \oplus U(Q_1 \cup Q_2) = \lambda(\Delta) \oplus U(Q_1 \cup Q_2) \\ B_1 &= \bigoplus_{x_k \in \vartheta_1 \cup Q_1 \cup \Delta} x_k \oplus S(Q_2) = \lambda(\vartheta_1) \oplus \lambda(Q_1) \oplus \lambda(\Delta) \oplus S(Q_2) \\ B_2 &= \bigoplus_{x_i \in \vartheta_2 \cup Q_2} x_i \oplus R(Q_1) = \lambda(\vartheta_2) \oplus \lambda(Q_2) \oplus R(Q_1) \end{aligned} \quad (9)$$

Where $U(Q_1 \cup Q_2)$- an arbitrary Boolean function defined on the variables of the sets Q1 and Q2, S (Q2) is an arbitrary Boolean function defined on variables belonging to the set Q2, R (Q1) is an arbitrary Boolean function defined on the variables of the set Q1.The resulting balanced SAC function g (x_1, ..., x_{n+h}) is formed as follows in (10):

$$g(x_1,..., x_{n+h}) = B_0 \oplus B_1 \cdot B_2$$
$$B_{V0} = \bigoplus_{x_j \in \Delta} x_j \oplus U_V(D_1 \cup D_2) = \lambda(\Delta) \oplus U_V(D_1 \cup D_2)$$
$$B_{V1} = \bigoplus_{x_k \in \vartheta_1 \cup D_1 \cup \Delta} x_k \oplus S_V(D_2) = \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \lambda(D_1) \oplus S_V(D_2) \quad (10)$$
$$B_{V2} = \bigoplus_{x_i \in \vartheta_2 \cup D_2} x_i \oplus R_V(D_1) = \lambda(\vartheta_2) \oplus \lambda(D_2) \oplus R_V(D_1)$$

Let us prove that the function f_V (Ω-V) = $BV_0 \oplus B_{V1} \cdot B_{V2}$ is a balanced SAC function. To prove the balance of the function f_V (Ω-V), it must be shown that the functions BV_0, BV_1 and BV_2 are balanced and mutually orthogonal, that is, because any linear combination is a balanced function. The function BV_0 is balanced because it is the sum modulo 2 of the linear function λ(Δ) of the variables of the non-empty set Δ и and the function $U_V(D_1 \cup D_2)$, that does not depend on the variables of this set Δ. The function BV_1 is the sum modulo 2 of the linear function $\lambda(\vartheta_1) \oplus \lambda(D_1) \oplus \lambda(\Delta)$ of the variables ϑ_1, D_1, Δ и and the function $S_V(D_2)$ independent of the variables of these sets; hence, function BV_1 is balanced. Similarly, the function BV_2 is the sum modulo 2 of the linear function of the variables ϑ_2, D_2 и and the function RV (D1) defined on the variables of the set D1, since $\vartheta_2 \cap D_1 = \varnothing$ и $D_2 \cap D_1 = \varnothing$, then the function BV_2 is also a balanced function, taking into consideration that the function $B_{V1} \oplus B_{V2}$ is represent table as the sum of the linear components. Let us prove that the function f_V(Ω-V) = $B_{V0} \oplus B_{V1} \cdot B_{V2}$is a balanced SAC function. To prove the balance of the function f_V(Ω-V) it must be shown that the functions BV_0, BV_1 and BV_2 are balanced and mutually orthogonal, that is, that any linear combination is a balanced function. The function BV0 is balanced because it is the sum modulo 2 of the linear function Δλ(Δ) of the variables of the non-empty set Δ and the function UV (D_1andD_2) that does not depend on the variables of this set Δ. The function BV_1 is the sum modulo 2 of the linear function ϑ_1, D_1, Δ (D_1) of the variables $S_V(D_2)$, and the function SV (D_2) independent of the variables of these sets; hence, function BV1 is balanced. Similarly, the function BV_2 is the sum modulo 2 of the linear function $\vartheta_2 \cap D_1 = \varnothing$ и $D_2 \cap D_1 = \varnothing$, of the variables R_2, D_2 and the function RV (D_1) defined on the variables of the set D1, since $\vartheta_2 \cap D_1 = \varnothing$ и $D_2 \cap D_1 = \varnothing$, then the function BV_2 is also a balanced function.

The function $B_{V1} \oplus B_{V2}$ is representable as the sum of the linear component$\lambda(\vartheta_1) \oplus \lambda(\vartheta_2) \oplus \lambda(\Delta)$ и $\phi_1 = \lambda(D_1) \oplus S_V(D_2) \oplus \lambda(D_2) \oplus R_V(D_1)$ function that does not depend on variables belonging to setsϑ_1, Δ и ϑ_2.Therefore, $BV_1$and BV_2 is a balanced function.The function BV_0 and BV_1 can be represented as the sum of the linear component

$$\varphi_j = \frac{\partial U_V(D_1 \cup D_2)}{\partial x_j} \oplus \lambda(\vartheta_2) \oplus \lambda(D_2) \oplus \frac{\partial (R(D_1) \cdot \lambda(D_1))}{\partial x_j} \oplus$$
$$\oplus \frac{\partial R(D_1)}{\partial x_j} \cdot (\lambda(\vartheta_1) \oplus S(D_2) \oplus \lambda(\Delta)) = \lambda(\vartheta_2) \oplus \xi$$

Where ξ is a function that does not depend on the variables of the set X_2. Thus, if x_j D_1, the function j can be represented as a linear combination of the variables of the set X_2 and the function that does not depend on the variables of the set D_2. Hence, the function $D_j \varphi_j = 1 \oplus \lambda(\vartheta_2) \oplus \lambda(D_2) \oplus R_V(D_1) = 1 \oplus B_{V2}$, If x_j, then, that is, in this case the function coincides with the inversion of the balance function BV_2 and, therefore, is a balancedthere is a function φ_j coinciding with the balance function BV_1 and, accordingly, is also balanced.

If $x_j \varphi_j D_2$ then given in (11):

$$\varphi_j = \frac{\partial U_V(D_1 \cup D_2)}{\partial x_j} \oplus \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \lambda(D_1) \oplus \frac{\partial (S_V(D_2) \cdot \lambda(D_2))}{\partial x_j} \oplus$$
$$\oplus \frac{\partial S_V(D_2)}{\partial x_j} \cdot (\lambda(\vartheta_2) \oplus R_V(D_1)) = \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \delta \qquad (11)$$

Where δ- is a function that does not depend on the variables of the sets D1 and D. Thus, if $x_j x_j \in D_2$, the function can be represented as a linear combination of the variable sets $x_j \in D_2$ and the function that does not depend on these variables. Therefore, the function $x_j \in D_2$ for x_j D_2 is also balanced.

Synthesis example

The proposed method for synthesizing tunable code generators of Boolean balanced SAC functions can be illustrated by the following example. Let it be necessary to construct a function g (x_1, ..., x_6, k_1, k_2) transforming for different values of the two-digit code k_1, k_2 (h = 2) into different balanced SAC functions from six variables (n = 6).

$$B_0 = \lambda(\Delta) \oplus U = x_2 \oplus x_4 \cdot x_6 \cdot k_1 \oplus x_5 \cdot x_6 \cdot k_2 \oplus x_4 \cdot x_5 \cdot x_6$$
$$B_1 = \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \lambda(Q_1) \oplus S(Q_2) = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus k_1 \oplus x_6 \cdot k_2$$
$$B_2 = \lambda(\vartheta_2) \oplus \lambda(Q_2) \oplus R(Q_1) = x_3 \oplus x_6 \oplus k_2 \oplus x_4 \cdot x_5 \cdot k_1$$

In accordance with the above method, the function g (x_1, ..., x_6, k_1, k_2) generating for each value of codes k1, k2 the balanced SAC-functions is formed in the following form:

$$g = B_0 \oplus B_1 \cdot B_2 = x_2 \oplus x_4 \cdot x_6 \cdot k_1 \oplus x_5 \cdot x_6 \cdot k_2 \oplus x_4 \cdot x_5 \cdot x_6 \oplus x_1 \cdot x_3 \oplus$$
$$\oplus x_1 \cdot x_6 \oplus x_1 \cdot k_2 \oplus x_1 \cdot x_4 \cdot x_5 \cdot k_1 \oplus x_2 \cdot x_3 \oplus x_2 \cdot x_6 \oplus x_2 \cdot k_2 \oplus x_4 \cdot x_3 \oplus$$
$$\oplus x_2 \cdot x_4 \cdot x_5 \cdot k_1 \oplus x_4 \cdot x_6 \oplus x_4 \cdot k_2 \oplus x_5 \cdot x_3 \oplus x_5 \cdot x_6 \oplus x_4 \cdot x_5 \cdot k_1 \oplus$$
$$\oplus x_5 \cdot k_2 \oplus x_3 \cdot x_6 \cdot k_2 \oplus x_4 \cdot x_5 \cdot x_6 \cdot k_1 \cdot k_2 \oplus x_3 \cdot k_1 \oplus x_6 \cdot k_1 \oplus k_1 \cdot k_2$$

The generated function g (x_1, ..., x_6, k_1, k_2) is a balanced SAC function of eight variables.For each value of the control bits k_1 and k_2 constituting the set V = {k_1, k_2}, the synthesized function g (x_1, ..., x_6, k_1, k_2) is transformed into a balanced SAC function f_V from six variables. For example, for k_1 = 0, k_2 = 0, the synthesized function takes the following form:

$$g_V(k_1 = 0, k_2 = 0) = x_2 \oplus x_4 \cdot x_5 \cdot x_6 \oplus x_1 \cdot x_3 \oplus x_1 \cdot x_6 \oplus x_2 \cdot x_3 \oplus x_2 \cdot x_6 \oplus x_3 \cdot x_4 \oplus$$
$$\oplus x_4 \cdot x_6 \oplus x_5 \cdot x_3 \oplus x_5 \cdot x_6$$

This function is balanced and corresponds to the criterion of a strict avalanche effect. For k_1 = 1, k_2 = 0, the synthesized function g is transformed into another balanced SAC function from six variables:

$$g_V(k_1 = 1, k_2 = 0) = 1 \oplus x_2 \oplus x_3 \oplus x_6 \oplus x_1 \cdot x_3 \oplus x_1 \cdot x_6 \oplus x_1 \cdot x_4 \cdot x_5 \oplus$$
$$\oplus x_2 \cdot x_3 \oplus x_2 \cdot x_6 \oplus x_2 \cdot x_4 \cdot x_5 \oplus x_4 \cdot x_3 \oplus x_3 \cdot x_5 \oplus x_4 \cdot x_5$$

Conclusions

In this paper, a method to develop an approach to the synthesis of balanced Boolean functions is proposed that satisfiesa criterion of a strictly avalanche effect based on transformations of orthogonal basis functions. The main feature of the developed method is that it can be used both for synthesizing fixed balanced SAC functions and for constructing such functions controlled by code generators. When synthesizing fixed SAC functions, the proposed method differs from those known for simplicity and high manufacturability, since it does not use representations of synthesized functions in the form of a truth table.

The developed method, unlike the known method [12], does not require an initial obtaining of an orthogonal system of n functions, but uses only three basic orthogonal functions, and their choice is completely formalized. Since the proposed method assumes an arbitrary choice of three sub-functions from many variables, it is possible to build a larger number of SAC functions in comparison with known methods. These advantages are also preserved when constructing code-driven SAC function generators.

References:

[1] Centre National de la RechercheScientifique (LA 7), Laboratoire IMAG, BP No68,38402 Saint Martin and H&es CEDEX, France 'Computation of Boolean Functions on Networks of Binary Automata", Journal of computer and system sciences, 26, 269 – 277, 1983.

[2] Markovskiy OP, El-Hami I., Ryabukha L.R. The method of maintaining boolean balance functions, which satisfies the criterion of the avalanche efektu // Naukivivisti NTUU "KPI", 2, -K., "EKMO". - P.31-40.

[3] Hamid Ghourchian, Amin Gohari and ArashAmini, "Existence and Continuity of Differential Entropy for a Class of Distributions', IEEE Communication letters 21(7), 1469-1472, 2017. DOI: 10.1109/LCOMM.2017.2689770

[4] Ablayev F., Vasiliev A. (2014) Computing Boolean Functions via Quantum Hashing. In: Calude C., Freivalds R., Kazuo I. (eds) Computing with New Resources. Lecture Notes in Computer Science, vol 8808. Springer, Cham.

[5] Roland DobaiJan Korenek and Lukas Sekanina "Evolutionary design of hash function pairs for network filters" Elsevier Applied Soft Commuting journal, Volume 56, PP: 173-181, 2017.

[6] David Joyner (2013) Review of Cryptographic Boolean Functions and Applications by Thomas Cusick and Pantelimon Stănică, Cryptologia, 37:2, 189-192, DOI: 10.1080/01611194.2013.767683

[7] Justin Gilmer , Michael Saks , Srikanth Srinivasan, Composition limits and separating examples for some boolean function complexity measures, Combinatorica, v.36 n.3, p.265-311, June 2016

[8] AndrisAmbainis , Martins Kokainis , Robin Kothari, Nearly optimal separations between communication (or query) complexity and partitions, Proceedings of the 31st Conference on Computational Complexity, p.1-14, May 29-June 01, 2016, Tokyo, Japan

[9] Bernd Steinbach, Christian Posthoff. (2013) Boolean Differential Equations. Synthesis Lectures on Digital Circuitsand Systems 8:3, 1-158.
Online publication date: 16-Jul-2013.

[10] Seberry J., Zhang X., Zheng Y. Nonlinearity and propagation characteristics of balanced Boolean functions.//Information and Computation Academic Press. 1995.-Vol. 119, No. 1-P.1-13

[11] Saleh Ibrahim Al-Omar. Using generators of Boolean functions to improve the efficiency of the hash memory. // ВісникНаціональноготехнічномуУніверситетуУкраїни "KPI". Інформатика, управліннятаобчислювальнатехніка.- 2003.- № 40.-К., "ВЕК ++", - С.131-140.

[12] Kurosawa K., Satoh T. Design of SAC / PC (l) of Order k Boolean Functions and Three Other Cryptographic Criteria. // Proc. International Conf. Advanced in Cryptology - Eurocrypto'97, LNCS 1233 - 1997-P.433-449.