Method of designing a code controlled generator of Balanced Boolean functions which satisfy the Strict Avalanche Criterion (SAC)

Dr. N.G. BARDIS [1,2]
[1]Adjunct Assistant Professor
Department of Automation
Technological Education Institutes of Halkis
34400 Psahna, Halkis, Evia, Greece
[2]Research Associate
Hellenic Army Academy
Vari, 16673 Attiki, Greece

Abstract: - In this paper a method of designing a code controlled generator of Balanced Boolean functions which satisfy the Strict Avalanche Criterion (SAC) is presented. It is based on using orthogonal nonlinear components. In contrast to other high order SAC functions, this generator designed by the proposed method, has seperate sets of control and information inputs. The advantage of the method is its simplicity and its technological realization.

Key-Words: - Boolean Functions, Balance Functions, SAC Functions

1 Introduction

The dynamical development of the computing systems and the integration of information resources based on computer networks stimulates the theory and applications of Boolean functions scientific researches. This class of functions is widely used in many important directions of modern information technology.

Recent advancement 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. The development of new cryptographic arrangements for information and data security in computer networks and mobile communication systems requires solving some theoretical and engineering problems. These problems are associated with the design and application of Boolean functions that correspond to the total and conditioned criteria of maximum entropy or, in other words, the balanced functions that satisfy the Strict Avalanche Criterion (SAC).

An important class of Boolean functions that is widely used in cryptography is the nonlinear functions which satisfy the Propagation Criterion and, in particular, Strict Avalanche Criterion (SAC). This class of Boolean functions is the base of one-way transformations that are used in block ciphers, hash-algorithms and in generators of pseudorandom binary sequences. The important direction for increasing crypto-resistance of the cryptographic algorithms that are based on Boolean transformations is the use of reconstructed by secret key Boolean functions which satisfy the cryptographical criteria and, in particular, Strict Avalanche Criterion.

For the practical implementation of the way mentioned above for increasing the effectiveness of the cryptographic algorithms it is necessary to develop high technological methods for designing a key-code controlled generator of high nonlinear Balanced Boolean functions which satisfy the Strict Avalanche Criterion (SAC).

2 Main definitions

The Boolean function $f(x_1,x_2,\ldots,x_n)$, which is determined on a set Z of 2^n possible variables n-tuples is balanced, if with the same probabilities the function values equal to zero and one :

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

The Boolean function $f(x_1,x_2,\ldots,x_n)$ satisfies the Strict Avalanche Criterion or is a SAC-function, if by changing one of the input variables, the function value changes with probability 0.5:

$$\sum_{X \in Z} f(X) \oplus f(X \oplus \Delta_j) = 2^{n-1}, \forall j \in \{1,..., n\},$$

$$\Delta_j = (d_1,..., d_j,..., d_n),$$

$$d_j = 1, d_i = 0, \forall i \in \{1,..., n\}, i \neq j,$$

where, Δ_j- n-tuples binary vector, the j-th component

of which equals to one and all others – zero.

Any Boolean function $f(x_1,x_2,\ldots,x_n)$ can be represented in the form of Shannon decomposition with respect to a variable $x_j \in \{ x_1,x_2,\ldots,x_n \}$:

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

where φ_j and ψ_j are Boolean functions which do not depend on the variable x_j.

If $f(X)\oplus f(X\oplus\Delta_j)=\varphi_j(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n)$, then the Boolean function $f(x_1,x_2,\ldots,x_n)$ satisfies the Strict Avalanche Criterion if for any $j\in\{1,\ldots,n\}$ the functions $\varphi_j(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n)$ are balanced, i.e. $\sum_{X\in Z}\varphi_j(X) = 2^{n-1}, \forall j \in \{1,...,n\}$.

Since the functions $\varphi_j(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n)$ do not depend on the variable x_j, they can be considered as a function on n-1 variables. Since all 2^{n-1} possible (n-1)-tuples of the variables $x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n$, compose the set Z_j, then it can be said that the Boolean function $f(x_1,x_2,\ldots,x_n)$ satisfies the Strict Avalanche Criterion if the following condition is satisfied:

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

Some researchers [3] consider the function $\varphi_j(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n)$ as the differential of the Boolean function $f(x_1,x_2,\ldots,x_n)$ with respect to the variable x_j:

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

Therefore, it can be said that the Boolean function $f(x_1,x_2,\ldots,x_n)$ satisfies the Strict Avalanche Criterion, if its differential with respect to all variables are balanced.

Such a Boolean function $g(x_1,x_2,\ldots,x_h,k_1,k_2,\ldots k_h)$ that is transformed to a SAC-function of n variables under any code K controlled by the h-bits code generator $K=\{k_1,k_2,\ldots,k_h\}$ is.

$$\forall k_i \in \{0,1\}, i = 1,...,h, \forall j \in \{1,...,n\}:$$

$$\sum_{X\in Z}(g(X,k_1,...,k_h) \oplus g(X\oplus\Delta_j,k_1,...,k_h)) = 2^{n-1}$$

The Boolean SAC-functions of order h is a special case of the generator of SAC functions controlled by a code [1].

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

Let's denote with $\lambda(\theta)$ the linear Boolean function which is formed by XORing all of its variables of the set $\theta\subseteq Z$:

$$\lambda(\theta) = \underset{x_k\in\theta}{\oplus} x_k .$$

Also let's denote with $\eta(\theta)$ the number of variables which compose the set θ.

3 State of the art of effective design problem of SAC- functions

Due to the important role of Boolean nonlinear SAC-functions for cryptography, a number of approaches to design such functions have been suggested during the last 15 years [1-3]. The known design methods of balanced SAC-functions can be divided in to three groups.

The methods of the first group are based in designing new balanced SAC-functions from functions that are somehow previously obtained. More specifically, the Forre method [2], which is using the spectrum transformations, allows obtaining from one balanced SAC-function a certain set of such functions with the same number of variables. There are also methods that allow obtaining from one SAC-function, other SAC-functions with a higher number of variables. Balanced SAC-functions of high nonlinearity may be obtained by de-concatenation of a bent-function [4], 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.

In the second group are the design methods of SAC-functions that are based on different transformations with orthogonal systems of Boolean functions. A number of interesting approaches have been proposed. The design method of balanced SAC functions that has been suggested by Kurosawa K. and Satoh T.[3], is one of the well known. The disadvantage of this method is the technological complexity, which is related to the use of matrices of high dimensions. Below a number of approaches will be presented, regarding the design of balanced SAC-functions based on orthogonal systems that have a more simple technological realization.

The advantage of the method designing systems of high crypto-resistant using Boolean functions belongs to a special class of orthogonal systems. This method allows the transition in a more simple way than the design method of a single function.

The greatest disadvantage of the genetic methods and the methods based on orthogonal systems is the fact that they allow to obtain only a rather small

amount of the total balanced SAC-functions.

In the third group are the combinational methods of designing balanced SAC-functions. They are based on dividing the set of variables into subsets and forming intermediate functions defined on the variables of these subsets and pairs of subsets [1].

Most of the researches are dedicated to desiging SAC-functions that do not relate to generators of synthesing such functions controlled by a code. In some works [1,3] methods of desiging SAC-functions of high order have been proposed. These functions can be used in generators of SAC-functions controlled by a code. In the case of utilizing such SAC-function, $f(x_1,x_2,\ldots,x_n)$, of order h, as control inputs, can use any h from n variables.

The main disadvantages of utilizing generators of high order SAC-functions controlled by a code in comparison to generator of SAC-functions controlled with fixed control inputs are the following:

- SAC-functions of order h have redundant high complexity
- Number of the existent SAC-functions of order h is significant less.

Therefore the approach of desiging generator SAC-functions using code control inputs, seems to be the more effective.

4 Method for designing Boolean balanced SAC- functions genarator controlled by code

The proposed method for designing of balanced Boolean SAC-functins generators controlled by a code relates to the second group of methods according to the above mentioned classification scheme. It is based on the transformation of three nonlinear orhgogonal Boolean functions. These functions have been already desinged by a special way. The proposed method is the further development of the methods [1,2].

The idea of the proposed method for obtaining the ANF of function $g(x_1,x_2,...,x_{n+h})$, which is controlled by h-bit code generator of balanced SAC-functions on n variables, consists of performing the following sequential actions:

1. The set $\Omega=\{x_1,x_2,\ldots,x_{n+h}\}$ of n+h variables is divided into five subsets that do not intersect ϑ_1, Q_1, Δ, ϑ_2, Q_2: $\vartheta_1 \cup Q_1 \cup \Delta \cup \vartheta_2 \cup Q_2 = \Omega$, such that $\vartheta_1 \neq \varnothing$, $\Delta \neq \varnothing$, $\vartheta_2 \neq \varnothing$ and the size of the set $\eta(Q_1 \cup Q_2) \geq h$.
2. The ANF of the three Boolean functions B_0, B_1 and B_2 are formed in the following way:

$$B_0 = \bigoplus_{x_j \in \Delta} x_j \oplus U(Q_1 \cup Q_2) = \lambda(\Delta) \oplus U(Q_1 \cup Q_2)$$

$$B_1 = \bigoplus_{x_k \in \vartheta_1 \cup Q_1 \cup \Delta} x_k \oplus S(Q_2) = \lambda(\vartheta_1) \oplus \lambda(Q_1) \oplus \lambda(\Delta) \oplus S(Q_2)$$

$$B_2 = \bigoplus_{x_i \in \vartheta_2 \cup Q_2} x_i \oplus R(Q_1) = \lambda(\vartheta_2) \oplus \lambda(Q_2) \oplus R(Q_1)$$

where the set $U(Q_1 \cup Q_2)$ is arbitrary of the Boolean functions determined by the variables of the sets Q_1 and Q_2, $S(Q_2)$ is arbitrary of the Boolean functions determined by the variables of the set Q_2 and $R(Q_1)$ is arbitrary of the Boolean functions determined by the variables of the set Q_1.

3. The resulting function $g(x_1,\ldots,x_{n+h})$ is formed as follows:

$$g(x_1,...,x_{n+h}) = B_0 \oplus B_1 \cdot B_2$$

It can be proved that for any fixed values of the controlling variables of the set $V \subseteq Q_1 \cup Q_2$, the function $g(x_1,x_2,\ldots,x_{n+h})$ is obtained by the transform of the balanced SAC-function $f_V(\Omega\text{-}V)$ determined on the variables of the sets ϑ_1, Δ, ϑ_2 . It must be noted that the number of the controlling variables can vary from 0 ($V=\varnothing$) to $\eta(Q_1 \cup Q_2)$, in the last case $V= Q_1 \cup Q_2$.

Let's denote with D_1 the set of variables that belong to the set Q_1, but do not belong to the set V: $x_j \in D_1$, $j \in \{1,\ldots,n+h\}$, $x_j \in Q_1$, $x_j \notin V$. By analogy, let's denote with D_2 the set of variables that belong to the set Q_2, but do not belong to the set V: $x_j \in D_2$, $j \in \{1,\ldots,n+h\}$, $x_j \in Q_2$, $x_j \notin V$. Sets D_1 and D_2 can be empty.

When some variables of the set V are fixed, the function $U(Q_1 \cup Q_2)$ is transformed to the function $U_V(D_1 \cup D_2)$, and the functions $S_V(D_2)$ and $R(Q_1)$ are transformed to the functions $S_V(D_2)$ and $R_V(D_1)$, correspondently. Therefore, the functions B_0, B_1 and B_2 are transformed to the functions B_{V0}, B_{V1} and B_{V2}, which can be represented as follows:

$$B_{V0} = \bigoplus_{x_j \in \Delta} x_j \oplus U_V(D_1 \cup D_2) = \lambda(\Delta) \oplus U_V(D_1 \cup D_2)$$

$$B_{V1} = \bigoplus_{x_k \in \vartheta_1 \cup D_1 \cup \Delta} x_k \oplus S_V(D_2) = \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \lambda(D_1) \oplus S_V(D_2)$$

$$B_{V2} = \bigoplus_{x_i \in \vartheta_2 \cup D_2} x_i \oplus R_V(D_1) = \lambda(\vartheta_2) \oplus \lambda(D_2) \oplus R_V(D_1)$$

Now it will be demonstrated that the function $f_V(\Omega\text{-}V) = B_{V0} \oplus B_{V1} \cdot B_{V2}$ is a balanced SAC-function. To prove that $f_V(\Omega\text{-}V)$ is balanced it will be necessary to demonstrate that the functions B_{V0}, B_{V1} and B_{V2} are balanced and mutually orthogonal, i.e. any linear combination of the functions B_{V0}, B_{V1} and B_{V2} is a balanced function.

The function B_{V0} is balanced since it is the XOR of a linear function $\lambda(\Delta)$ defined on the variables of the nonempty set Δ with the function $U_V(D_1 \cup D_2)$, which does not depend on the variables of this set Δ.

The function B_{V1} is balanced since it is the XOR of a linear function $\lambda(\vartheta_1)\oplus\lambda(D_1)\oplus\lambda(\Delta)$ defined on the variables of the sets ϑ_1, D_1, Δ with the function $S_V(D_2)$, which does not depend on the variables of these sets ϑ_1, D_1, Δ. Reasoning by analogy, the function B_{V2} is formed as the XOR of a linear function $\lambda(\vartheta_2)\oplus\lambda(D_2)$ defined on the variables of the sets ϑ_2, D_2 with the function $R_V(D_1)$ defined on the variables of the set D_1. Since $\vartheta_2\cap D_1=\varnothing$ and $D_2\cap D_1=\varnothing$ then the function B_{V2} is balanced too. The function $B_{V1}\oplus B_{V2}$ can be represented as the XOR of a linear function $\lambda(\vartheta_1)\oplus\lambda(\vartheta_2)\oplus\lambda(\Delta)$ with the function

$$\phi_1 = \lambda(D_1) \oplus S_V(D_2) \oplus \lambda(D_2) \oplus R_V(D_1)$$

which does not depend on the variables of sets ϑ_1, Δ and ϑ_2. Consequently, the linear combination $B_{V1}\oplus B_{V2}$ is a balanced function.

The function $B_{V0}\oplus B_{V1}$ can be represented as the XOR of a linear function $\lambda(\vartheta_1)$ with the function $\phi_2 = \lambda(D_1) \oplus U_V(D_1 \cup D_2) \oplus S_V(Q_2)$ which does not depend on the variables of the set ϑ_1. Thus, the linear combination $B_{V0}\oplus B_{V1}$ is a balanced function. By analogy, the linear combination of the functions $B_{V0}\oplus B_{V2}$ can be represented as the XOR of the linear component $\lambda(\vartheta_2)\oplus\lambda(\Delta)$ with the function $\phi_3 = \lambda(D_2) \oplus U_V(D_1 \cup D_2) \oplus R_V(Q_1)$ which does not depend on the variables of the sets ϑ_2, Δ. So the linear combination of the functions $B_{V0}\oplus B_{V2}$ is a balanced function. The linear combination of all three functions $B_{V0}\oplus B_{V1}\oplus B_{V2}$ can be represented as the XOR of a linear function $\lambda(\vartheta_1)\oplus\lambda(\vartheta_2)$ defined on the variables of the sets ϑ_1, ϑ_2 with the function

$$\phi_4 = \lambda(D_1) \oplus \lambda(D_2) \oplus U_V(D_1 \cup D_2) \oplus S_V(D_2) \oplus \oplus R_V(D_1)$$

which does not depend on the variables of the sets ϑ_1 and ϑ_2. So the linear combination of all functions $B_{V0}\oplus B_{V1}\oplus B_{V2}$ is a balanced function. Thus, it has been demonstrated that the functions B_{V0}, B_{V1} and B_{V2} are balanced and orthogonal.

Now it is necessary to demonstrate that the function $f_V = B_{V0} \oplus B_{V1}\cdot B_{V2}$ is a SAC-function. For this it will be necessary to demonstrate that any functions $\varphi_j(\Omega\text{-}V\text{-}x_j)$ in the Shannon decomposition form $f_V = x_j \cdot \varphi_j \oplus \psi_j$ with respect to the variable x_j, for $j\in\{1,\ldots,n\}$:$x_j\in\Omega$-V is balanced. It is expedient to analyze the functions $\varphi_j(\Omega\text{-}V\text{-}x_j)$ separately for five cases: $x_j\in\vartheta_1$, $x_j\in D_1$, $x_j\in\Delta$, $x_j\in\vartheta_2$ and $x_j\in D_2$.

If $x_j\in\vartheta_1$ then

$\varphi_j = \lambda(\vartheta_2) \oplus \lambda(Q_2) \oplus R_V(D_2) = B_{V2}$. In this case the function $\varphi_j(\Omega\text{-}V\text{-}x_j)$ is equal to the balanced function B_{V2} and therefore, is also balanced.

If $x_j\in D_1$ then

$$\varphi_j = \frac{\partial U_V(D_1 \cup D_2)}{\partial x_j} \oplus \lambda(\vartheta_2) \oplus \lambda(D_2) \oplus$$

$$\oplus \frac{\partial (R(D_1)\cdot\lambda(D_1))}{\partial x_j} \oplus$$

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

where the ξ- function, which does not depend on the variables of the set ϑ_2. Consequently, in this case the function φ_j can be represented as the XOR of a linear function $\lambda(\vartheta_2)$ with the function ξ that does not depend on the variables of the set ϑ_2. Therefore, if $x_j\in D_1$ then the function φ_j is balanced.

If $x_j\in\Delta$ then

$\varphi_j = 1 \oplus \lambda(\vartheta_2) \oplus \lambda(D_2) \oplus R_V(D_1) = 1 \oplus B_{V2}$

In this case the function φ_j is the inverse of the balanced function B_{V2} and therefore it is balanced too.

In the case when $x_j\in\vartheta_2$, then

$\varphi_j = \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \lambda(D_1) \oplus S_V(D_2) = B_{V1}$

The function φ_j is equal to the balanced function B_{V1} and correspondingly, it is also balanced.

In the case when $x_j\in D_2$ then the function φ_j can be represented as:

$$\varphi_j = \frac{\partial U_V(D_1 \cup D_2)}{\partial x_j} \oplus \lambda(\vartheta_1) \oplus \lambda(\Delta) \oplus \lambda(D_1) \oplus$$

$$\oplus \frac{\partial (S_V(D_2)\cdot\lambda(D_2))}{\partial x_j} \oplus$$

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

where the δ- function, which does not depend on the variables of the sets ϑ_1 and Δ. Consequently, in this case the function φ_j can be represented as the XOR of a linear function $\lambda(Q_1)\oplus\lambda(\Delta)$ defined on variables of sets Q_1, Δ with the function δ than does not depend on the variables of these sets Q_1 and Δ. Thus, in the case when $x_j\in D_2$, the function φ_j is balanced too.

Therefore it has been demonstrated that for all possible variants of the location x_j of the variables set ($x_j\in\vartheta_1$, $x_j\in D_1$, $x_j\in\Delta$, $x_j\in\vartheta_2$ and $x_j\in D_2$) the function $\varphi_j(\Omega\text{-}V\text{-}x_j)$ is a Shannon decomposition form of the function $f_V = x_j \cdot \varphi_j \oplus \psi_j$ with respect to the variable x_j and is balanced. Consequently the function f_v satisfies the Strict Avalanche Criterion (SAC).

5 Designing Example

The suggested method is illustrated by the

following example designing a controlled 2-bits length code (h=2) generator of balanced SAC-function from 6 variables (n=8). Thus, it is necessary to design the function $g(x_1,\ldots,x_6,k_1,k_2)$, whish is transformed into different balanced SAC-functions $f_V(x_1,x_2,\ldots,x_6)$ in cases of all 4 possible values of the 2-bits length control code k_1,k_2.

The set $\Omega=\{x_1,x_2,\ldots,x_6,k_1,k_2\}$ of all variables is divided into five subsets that do not overlap, as follows: $\vartheta_1=\{x_1\}$, $Q_1=\{x_4,x_5,k_1\}$, $\Delta=\{x_2\}$, $\vartheta_2=\{x_3\}$, $Q_2=\{x_6,k_2\}$. The functions $U(Q_1\cup Q_2)$, $S(Q_1)$ and $S(Q_2)$ are arbitrarily selected:

$U(x_4,x_5,x_6,k_1,k_2)=x_4\cdot x_6\cdot k_1\oplus x_5\cdot x_6\cdot k_2\oplus x_4\cdot x_5\cdot x_6$;

$S(x_6,k_2)=x_6\cdot k_2$;

$R(x_4,x_5,k_1) = x_4\cdot x_5\cdot k_1$.

Correspondently, the three functions B_0, B_1 and B_2 are formed as follows:

$$B_0 = \lambda(\Delta)\oplus U = \\ = x_2\oplus x_4\cdot x_6\cdot k_1\oplus x_5\cdot x_6\cdot k_2\oplus x_4\cdot x_5\cdot x_6$$

$$B_1 = \lambda(\vartheta_1)\oplus\lambda(\Delta)\oplus\lambda(Q_1)\oplus S(Q_2) = \\ = x_1\oplus x_2\oplus x_4\oplus x_5\oplus k_1\oplus x_6\cdot k_2$$

$$B_2 = \lambda(\vartheta_2)\oplus\lambda(Q_2)\oplus R(Q_1) = \\ = x_3\oplus x_6\oplus k_2\oplus x_4\cdot x_5\cdot k_1$$

According to step 3 of the method presented above, the ANF of the balanced SAC-function generator $g(x_1,\ldots,x_6,k_1,k_2)$ controlled by the code k_1,k_2 is formed as follows:

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

The generated function $g(x_1,\ldots,x_6,k_1,k_2)$ defined on 8 variables is a balanced SAC-function.

For any possible combinations of values of the controlling variables k_1 and k_2 of the set $V=\{k_1,k_2\}$, the synthesized function $g(x_1,\ldots,x_6,k_1,k_2)$ is transformed into a balanced SAC-function defined on the 6 variables $x_1,x_2,\ldots,x_6$. For example, if $k_1=0$ and $k_2=0$ the generated function $g(x_1,\ldots,x_6,k_1,k_2)$ is transformed into the function:

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

This function is balanced and corresponds to the Strict Avalanche Criterion (SAC).

In case of $k_1=1$, $k_2=0$ the generated function $g(x_1,\ldots,x_6,k_1,k_2)$ is transformed into another balanced SAC-function:

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

6 Conclusion

The proposed method of designing a generator of Balanced Boolean SAC-functions controlled by a code is the development of the approach that is the basis of the special transformations of orthogonal basic functions. The suggested method for the design of a generator of balanced SAC-functions controlled by a code operates only with ANF and removes the processing limitation for obtaining functions from a large number of variables. This allows the design of SAC-functions generators which are controlled by a large length of key codes - hundreds of bits. The developed method in contrast to the known [3] does not require obtaining a preliminary full orthogonal system of nonlinear functions. It is using only three basic orthogonal nonlinear functions with a very simple and completely formalized procedure. The suggested method is completely formalized and has been implemented in the form of a program written in C++.

References:

[1] Bardis E.G., Bardis N.G., Markovskyy A.P., Spyropoulos A.P., "Design of Boolean Function from a Great Number of Variables Satisfying Strict Avalanche Criterion". *Recent Advances in Signal Processing and Communications - IMACS/IEEE,*, ISBN: 960-8052-03-3, pp. 107-112.

[2] 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 P.450-468.

[3] 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.

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