Research on the Balanced Boolean Functions Satisfying Strict Avalanche Criterion

Zhongwei Cao, Fangyue Chen, Bo Chen, and Xia Zhang
Department of Mathematics, School of Science
Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, P. R. China
Email: caoguocun@sina.com / fychen@hdu.edu.cn

***Abstract*—The balanced Boolean functions satisfying the strict avalanche criterion (SAC) are widely used in stream cipher and block cipher. By introducing the representation matrix of the transition function of Boolean function, we provide a special method for seeking the Boolean functions satisfying the SAC. Moreover, the strong strict avalanche criterion (SSAC) of balanced Boolean function is defined and the general formula of the number of the Boolean functions satisfying SSAC is obtained.**

***Keywords*-Boolean function; balancedness; strict avalanche criterion; representation matrices; strong strict avalanche criterion;**

I. Introduction

With the development of computer technique and internet, the block cipher is becoming one of the main methods to guarantee the information security. As one of the cores of cryptography, S-box is an important technique for block cipher and has been widely used in some famous block cryptographic systems, such as data encryption standard (DES) and advanced encryption standard (AES). Basically, an S-box is a collection of Boolean functions with some criterions of cryptographic significance, which are balancedness, strict avalanche criterion (SAC), and other properties.

The SAC is an important property of the Boolean function in the cryptographic systems. Firstly, Webster and Tavares introduced the SAC in the design criteria for cryptographic functions [1]. Then, Feistel, Kam and Davida combined two earlier criteria for cryptographic applications [2], [3]. Forre extended the SAC by defining higher-order SAC [4]. Lloyed showed the characterizing and counting functions satisfying a higher order SAC [5], [6], [7]. O'Connor gave an upper bound for the number of functions satisfying the SAC [8], [9]. Thereafter, Cusick gave a lower bound for the number of functions satisfying the SAC and provided an improvement of lower bound [10]. Furthermore, Youssef and Tavares presented a detailed proof for Cusick's conjecture and modified the lower bound [11], [12], [13], [14]. Gupta and Sarkar considered the problem of constructing perfect nonlinear multiple-output Boolean functions satisfying higher order strict avalanche criteria(SAC)[15]. Tang, Zhang and other people gave a method to construct balanced Boolean functions with high nonlinearity and good autocorrelation properties[16]. Zhang, Jiang and Tang proposed a method to construct the highly nonlinear resilient Boolean functions on n variables(n is even) satisfying strict avalanche criterion [17]. Indeed, it is meaningful to count the balanced Boolean functions satisfying the SAC when the number of the input variables increases.

In this paper, we proposed some representation matrices of the transition function of Boolean function, which are used to get the Boolean functions satisfying the SAC. Moreover, the strong strict avalanche criterion (SSAC) of balanced Boolean function is defined and the general formula of the number of Boolean functions satisfying SSAC is also obtained.

The rest of the paper is organized as follows. Section II introduces the representation matrices for the bit transformation of Boolean function. Section III gives some conditions which offer a standard for us to judge if a balanced Boolean function satisfies the SAC, and discusses the expansion of this kind of functions. In section IV, the number of balanced Boolean functions satisfying SSAC is studied. Finally, Section V gives the conclusion.

II. Representation of Transition Function

An n-bit (or n-dimension) Boolean function is a map $f : \{0,1\}^n \to \{0,1\}, y = f(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in Z_2^n = \{0,1\}^n$ and $y \in Z_2 = \{0,1\}$. Let $k = \sum_{i=1}^{n} x_i \cdot 2^{n-i}$, then k is the decimal code of $\mathbf{x}$, $\mathbf{x}$ is named the input window of the function. There are 2^n different input windows, denoted by $\mathbf{x}^{(k)}$ $(k = 0, 1, \cdots, 2^n - 1)$. Thus, the map f can be rewritten as $f(\mathbf{x}^{(k)}) = y_{k+1}$ $(k = 0, 1, \cdots, 2^n - 1)$. Obviously, such a map can generate an output symbol tape $(y_1, y_2, \cdots, y_{2^n})$ consisting of symbols "0" and "1". Conversely, a symbol tape $(y_1, y_2, \cdots, y_{2^n})$ completely determines a Boolean function. Hence, there exist a total of 2^{2^n} n-bit Boolean functions. Let $\mathbf{y} = (y_1, y_2, \cdots, y_{2^n})^T$, named the output vector of $y = f(\mathbf{x})$, the decimal code of $\mathbf{y}$ is defined as $N = \sum_{i=1}^{2^n} y_i \cdot 2^{i-1}$.

In the following, the symbol tape $\mathbf{y}$ will be considered to be equivalent to the Boolean function $y = f(\mathbf{x})$, $W(\mathbf{x})$ and $W(f(\mathbf{x}))$ (or $W(\mathbf{y})$) denote the Hamming weight of $\mathbf{x}$ and $\mathbf{y}$ respectively. Several commonly used definitions are:

***Definition** 1:* An n-bit Boolean function $y = f(\mathbf{x})$ is said to satisfy balancedness if

$$W(f(\mathbf{x})) = \sum_{k=0}^{2^n-1} f(\mathbf{x}^{(k)}) = \sum_{k=1}^{2^n} y_k = 2^{n-1}. \qquad (1)$$

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.14

***Definition** 2:* An n-bit Boolean function $y = f(\mathbf{x})$ ($n \geq 3$) is said to satisfy the strict avalanche criterion (SAC) if complementing a single input bit results in changing the output bit with probability exactly one half, i.e.,

$$\begin{aligned} W(f(\mathbf{x} \oplus e) \oplus f(\mathbf{x})) &= \sum_{k=0}^{2^n-1} [f(\mathbf{x}^{(k)} \oplus e) \oplus f(\mathbf{x}^{(k)})] \\ &= 2^{n-1} \end{aligned} \tag{2}$$

where $e \in Z_2^n$ with $W(e) = 1$ and " $\oplus$ " denotes the XOR operation.

For a given Boolean function $y = f(\mathbf{x})$, complementing a single input bit means the input $\mathbf{x}$ is changed to $\mathbf{x} \oplus e$ ($W(e) = 1$), at the same time, $y = f(\mathbf{x})$ is converted to $y = f(\mathbf{x} \oplus e)$. Thus, a new definition can be got:

***Definition** 3:* Let $e_i = (0, \cdots, 0, 1, \overbrace{0, \cdots, 0}^{i-1})$ ($i = 1, 2, \cdots, n$), then $y = f(\mathbf{x} \oplus e_i)$ is called the i-th bit transition function of $y = f(\mathbf{x})$.

Obviously, if $y = f(\mathbf{x})$ is balanced, so is $y = f(\mathbf{x} \oplus e_i)$, and if $y = f(\mathbf{x})$ satisfies the SAC, so does $y = f(\mathbf{x} \oplus e_i)$ $i = (1, 2, \cdots, n)$.

***Theorem** 1:* For Boolean function $y = f(\mathbf{x})$ with output vector $\mathbf{y}$, the relationship between $\mathbf{y}$ and $\tilde{\mathbf{y}}_i$, the output vectors of the i-th bit transition function $y = f(\mathbf{x} \oplus e_i)$ ($i = 1, 2, \cdots, n$), is

$$\tilde{\mathbf{y}}_i = A_i \mathbf{y}, \ i = 1, 2, \cdots, n, \tag{3}$$

where

$$A_i = \begin{pmatrix} B_i & 0 & \cdots & 0 & 0 \\ 0 & B_i & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & B_i & 0 \\ 0 & 0 & \cdots & 0 & B_i \end{pmatrix}_{2^n \times 2^n} \tag{4}$$

and

$$B_i = \begin{pmatrix} 0 & C_i \\ C_i & 0 \end{pmatrix}_{2^i \times 2^i} \tag{5}$$

as well as

$$C_i = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}_{2^{i-1} \times 2^{i-1}} \tag{6}$$

The proof of this theorem can be directly obtained from the expressions of $y = f(\mathbf{x})$ and $y = f(\mathbf{x} \oplus e_i)$, ($i = 1, 2, \cdots, n$), the details are omitted here. Clearly, every matrix A_i is symmetric.

Thus, for a given Boolean function $y = f(\mathbf{x})$ with output $\mathbf{y}$, There must be another n Boolean functions with outputs $\tilde{\mathbf{y}}_i$, $i = (1, 2, \cdots, n)$, which will be generated by the theorem. Furthermore, the condition in (2) is transformed into

$$W(\mathbf{y} \oplus A_i \mathbf{y}) = 2^{n-1}, i = (1, 2, \cdots, n). \tag{7}$$

In addition, 2^n Boolean functions will be obtained via these A_i($i = 1, 2, \cdots, n$). In fact, for a given output vector $\mathbf{y}$ of a Boolean function (or directly called $\mathbf{y}$ as a Boolean function), we have $A_1\mathbf{y}$ by A_1, thus ($\mathbf{y}$, $A_1\mathbf{y}$) are two Boolean functions. Then based on these two Boolean functions, we have ($A_2\mathbf{y}$, $A_2A_1\mathbf{y}$) by A_2, thus ($\mathbf{y}$, $A_1\mathbf{y}$, $A_2\mathbf{y}$, $A_2A_1\mathbf{y}$) are four Boolean functions. By A_3, gets eight Boolean functions ($\mathbf{y}$, $A_1\mathbf{y}$, $A_2\mathbf{y}$, $A_2A_1\mathbf{y}$, $A_3\mathbf{y}$, $A_3A_1\mathbf{y}$, $A_3A_2\mathbf{y}$, $A_3A_2A_1\mathbf{y}$). In a similar way, we finally get 2^n Boolean functions ($\mathbf{y}$, $A_1\mathbf{y}$, $A_2\mathbf{y}$, $A_2A_1\mathbf{y}, \cdots, A_n\mathbf{y}, A_nA_1\mathbf{y}, \cdots, A_nA_{n-1}\cdots A_2A_1\mathbf{y}$). Enumerating them

$$\begin{aligned} (\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \cdots, \mathbf{y}^{(2^n)}) = (\mathbf{y}, \ A_1\mathbf{y}, \ A_2\mathbf{y}, \ A_2A_1\mathbf{y}, \cdots, \\ \cdots, A_n\mathbf{y}, \ A_nA_1\mathbf{y}, \cdots, A_nA_{n-1}\cdots A_2A_1\mathbf{y}). \end{aligned} \tag{8}$$

Obviously, for $\mathbf{y}$ and $\tilde{\mathbf{y}}_i$ in Theorem 1, it's easy to got $\mathbf{y} = \mathbf{y}^{(1)}$ and $\tilde{\mathbf{y}}_i = \mathbf{y}^{(2^{i-1}+1)}$ ($i = 1, 2, \cdots, n$).

An interesting fact is that the $2^n \times 2^n$ order matrix consisted of $\mathbf{y}^{(i)}$ ($i = 1, 2, \cdots, 2^n$) is a symmetric matrix.

***Corollary** 1:* If a Boolean function is balanced, there would exist 2^n ones which are also balanced, and if a Boolean function satisfies the SAC, then there exist 2^n ones which also satisfy the SAC.

For example, when $n = 3$, then A_i, $i = 1, 2, 3$, are respectively:

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{9}$$

$$A_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{10}$$

$$A_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{11}$$

If $\mathbf{y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)^T$ is a balanced Boolean function satisfying the SAC, then

$$(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \cdots, \mathbf{y}^{(8)}) = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ y_2 & y_1 & y_4 & y_3 & y_6 & y_5 & y_8 & y_7 \\ y_3 & y_4 & y_1 & y_2 & y_7 & y_8 & y_5 & y_6 \\ y_4 & y_3 & y_2 & y_1 & y_8 & y_7 & y_6 & y_5 \\ y_5 & y_6 & y_7 & y_8 & y_1 & y_2 & y_3 & y_4 \\ y_6 & y_5 & y_8 & y_7 & y_2 & y_1 & y_4 & y_3 \\ y_7 & y_8 & y_5 & y_6 & y_3 & y_4 & y_1 & y_2 \\ y_8 & y_7 & y_6 & y_5 & y_4 & y_3 & y_2 & y_1 \end{pmatrix}. \quad (12)$$

At the same time,

$$\begin{array}{l} \mathbf{y}^{(1)} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)^T = \mathbf{y}, \\ \mathbf{y}^{(2)} = (y_2, y_1, y_4, y_3, y_6, y_5, y_8, y_7)^T = A_1\mathbf{y}, \\ \mathbf{y}^{(3)} = (y_3, y_4, y_1, y_2, y_7, y_8, y_5, y_6)^T = A_2\mathbf{y}, \\ \mathbf{y}^{(5)} = (y_5, y_6, y_7, y_8, y_1, y_2, y_3, y_4)^T = A_3\mathbf{y}. \end{array} \quad (13)$$

Remark 1: There would not exist any Boolean function satisfying the SAC when $n = 2$. In fact, if $\mathbf{y} = (y_1, y_2, y_3, y_4)$ is the output of a 2-bit Boolean function $y = f(\mathbf{x})$, there are

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

and

$$(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \mathbf{y}^{(3)}, \mathbf{y}^{(4)}) = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 \\ y_2 & y_1 & y_4 & y_3 \\ y_3 & y_4 & y_1 & y_2 \\ y_4 & y_3 & y_2 & y_1 \end{pmatrix},$$

then the function satisfies the SAC if and only if $W(y^{(1)} \oplus y^{(2)}) = 2[(y_1 \oplus y_2) + (y_3 \oplus y_4)] = 2$ and $W(y^{(1)} \oplus y^{(3)}) = 2[(y_1 \oplus y_3) + (y_2 \oplus y_4)] = 2$, but this is a contradiction.

III. Conditions Satisfying Balancedness and the SAC

In this section, we give some conditions which guarantee the Boolean function satisfies the SAC, and discuss the number of the Boolean functions satisfying the SAC.

A well known result is:

Lemma 1: If $\mathbf{y}$ is a balanced Boolean function satisfying the SAC, then $\bar{\mathbf{y}} = \mathbf{1} - \mathbf{y} = (\bar{y}_1, \bar{y}_2, \cdots, \bar{y}_{2^n})^T$ is also balanced and satisfies the SAC, where $\bar{y}_i = 1 - y_i,\ i = (1, 2, \cdots, 2^n)$.

Theorem 2: For a 3-bit Boolean function $\mathbf{y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)^T$,

(a) If (y_1, y_2, y_3, y_4) is balanced, i.e., $W(y_1, y_2, y_3, y_4) = 2$, then $\mathbf{y}$ is a balanced Boolean function satisfying the SAC if it satisfies

$$\begin{cases} (y_1, y_2, y_3, y_4) = (\bar{y}_4, \bar{y}_3, \bar{y}_2, \bar{y}_1) \\ (y_5, y_6, y_7, y_8) = (\bar{y}_8, \bar{y}_7, \bar{y}_6, \bar{y}_5). \end{cases} \quad (14)$$

(b) If (y_1, y_2, y_3, y_4) is not balanced, then $\mathbf{y}$ is a balanced Boolean function satisfying the SAC if it satisfies one of the following conditions:

$$(1) \quad (y_1, y_2, y_3, y_4) = (\bar{y}_6, \bar{y}_5, \bar{y}_8, \bar{y}_7). \quad (15)$$

$$(2) \quad (y_1, y_2, y_3, y_4) = (\bar{y}_7, \bar{y}_8, \bar{y}_5, \bar{y}_6). \quad (16)$$

$$(3) \quad (y_1, y_2, y_3, y_4) = (\bar{y}_8, \bar{y}_7, \bar{y}_6, \bar{y}_5). \quad (17)$$

The proof of this theorem is easy, due to space limitations, the details are omitted here.

According to the conditions in Theorem 2, first, four balanced Boolean functions satisfying the SAC can be found, which are respectively:

$$\begin{array}{l} (0,0,1,1,0,1,0,1), (0,0,0,1,1,1,0,1), \\ (0,0,0,1,1,0,1,1), (0,0,0,1,0,1,1,1). \end{array} \quad (18)$$

Through Theorem 1 and its corollary, totally 32 Boolean functions which are balanced and satisfy the SAC are obtained. They are divided into 4 groups, each group of which contains 8 Boolean functions. They are respectively:

$$\begin{array}{l} (0,0,1,1,0,1,0,1), (0,0,1,1,1,0,1,0), \\ (1,1,0,0,0,1,0,1), (1,1,0,0,1,0,1,0), \\ (0,1,0,1,0,0,1,1), (1,0,1,0,0,0,1,1), \\ (0,1,0,1,1,1,0,0), (1,0,1,0,1,1,0,0). \end{array} \quad (19)$$

$$\begin{array}{l} (0,0,0,1,1,1,0,1), (0,0,1,0,1,1,1,0), \\ (0,1,0,0,0,1,1,1), (1,0,0,0,1,0,1,0), \\ (1,1,0,1,0,0,0,1), (1,1,1,0,0,0,1,0), \\ (0,1,1,1,0,1,0,0), (1,0,1,1,1,0,0,0). \end{array} \quad (20)$$

$$\begin{array}{l} (0,0,0,1,1,0,1,1), (0,0,1,0,0,1,1,1), \\ (0,1,0,0,1,1,1,0), (1,0,0,0,1,1,0,1), \\ (1,0,1,1,0,0,0,1), (0,1,1,1,0,0,1,0), \\ (1,1,1,0,0,1,0,0), (1,1,0,1,1,0,0,0). \end{array} \quad (21)$$

$$\begin{array}{l} (0,0,0,1,0,1,1,1), (0,0,1,0,1,0,1,1), \\ (0,1,0,0,1,1,0,1), (1,0,0,0,1,1,1,0), \\ (0,1,1,1,0,0,0,1), (1,0,1,1,0,0,1,0), \\ (1,1,0,1,0,1,0,0), (1,1,1,0,1,0,0,0). \end{array} \quad (22)$$

In a similar way, some conditions of the Boolean function satisfying the SAC can be given when $n = 4$. However, the form would be more complex, and the details are omitted here. It can be known that the number of balanced 4-bit Boolean functions satisfying the SAC is 1368. All these Boolean functions can be divided into 105 groups, which are shown in decimal form in TABLE I.

Table I
A PARTIAL LIST OF THE DECIMAL CODE OF 4-BIT BALANCED BOOLEAN FUNCTIONS SATISFYING THE SAC

No	Decimal Code
1	$\{59232, 59142, 56208, 56073, 48528, 48393, 37083, 37053, 32352, 32262, 24807, 24702, 2523, 2493, 1767, 1662\}$
2	$\{60768, 60681, 56976, 56838, 46992, 46854, 37086, 37047, 31584, 31497, 24813, 24699, 2541, 2427, 1758, 1719\}$
3	$\{60256, 60169, 55184, 55046, 48784, 48646, 37079, 37054, 32096, 32009, 24811, 24701, 2539, 2429, 1751, 1726\}$
4	$\{58992, 57462, 55728, 53433, 47568, 45273, 40203, 39693, 30432, 28902, 28167, 26382, 3687, 3483, 2973, 1902\}$
$\vdots$	$\vdots$
63	$\{58894, 57454, 55565, 53405, 47371, 45211, 40400, 39856, 30215, 28775, 28384, 26480, 3814, 3545, 3001, 1910\}$
64	$\{59088, 57561, 55776, 53478, 47472, 45174, 40206, 39687, 30384, 28857, 28173, 26379, 3741, 3438, 2919, 1947\}$
65	$\{59056, 57529, 55664, 53366, 47584, 45286, 40199, 39694, 30416, 28889, 28171, 26381, 3739, 3431, 2926, 1949\}$
66	$\{58887, 57447, 55563, 53403, 47373, 45213, 40368, 39888, 30222, 28782, 28272, 26592, 3702, 3513, 3033, 2022\}$
67	$\{63888, 63072, 40731, 37113, 28422, 24822, 2463, 1647\}$
68	$\{63840, 63120, 40710, 37110, 28425, 24825, 2415, 1695\}$
69	$\{63753, 62982, 40848, 37023, 28512, 24687, 2553, 1782\}$
70	$\{61593, 61542, 39408, 39183, 26352, 26127, 3993, 3942\}$
$\vdots$	$\vdots$
103	$\{51795, 50595, 44085, 41925, 23610, 21450, 14940, 13740\}$
104	$\{50090, 50005, 43715, 43580, 21955, 21820, 15530, 15445\}$
105	$\{51770, 50485, 44195, 41900, 23635, 21340, 15050, 13765\}$

IV. STRONG STRICT AVALANCHE CRITERION

So far, the number of balanced n-bit Boolean functions satisfying the SAC has not been known. The existing works just make estimations on the upper and the lower bound of its number. In the following, we will define the SSAC, and obtain the general formula of the number of n-bit Boolean functions satisfying SSAC. This kind of functions are very important in the cryptography.

***Lemma** 2:* For two n-bit balanced Boolean functions, $\mathbf{y}^{'} = (y_1^{'}, y_2^{'}, \cdots, y_n^{'})^T$ and $\mathbf{y}^{''} = (y_1^{''}, y_2^{''}, \cdots, y_n^{''})^T$, satisfying the SAC, then $\mathbf{y} = (\mathbf{y}^{'}, \mathbf{y}^{''}) = (y_1^{'}, y_2^{'}, \cdots, y_n^{'}, y_1^{''}, y_2^{''}, \cdots, y_n^{''})^T$ is a $(n+1)$-bit balanced Boolean function satisfying the SAC.

This conclusion can be directly got by the definitions of balancedness and the SAC.

***Definition** 4:* A n-bit Boolean function with output $(y_1, y_2, \cdots, y_{2^n})$ is called balanced Boolean function satisfying the SSAC, if all these m-bit Boolean functions with output $(y_1, y_2, \cdots, y_{2^m})$, $(y_{2^m+1}, y_{2^m+2}, \cdots, y_{2^{m+1}})$, $\cdots$, $(y_{2^n-2^m+1}, y_{2^n-2^m+2}, \cdots, y_{2^n})$, $(m = 3, 4, \cdots, n-1)$, are balanced and satisfy the SAC.

Compared with the Boolean functions satisfying the SAC, the ones satisfying SSAC have higher anti-attacking in S-box design in cryptography system. This kind of Boolean functions have many advantages in the design and analysis of block cipher. For example, the complexity of the Boolean functions has something to do with every input, and since all the sub-sequences from the Boolean function satisfy the SAC, it can resist the differential attacking from the outputs.

Based on the number of 3-bit balanced Boolean function satisfying the SAC and Lemma 2, the number of n-bit Boolean functions satisfying SSAC is easy to be obtained.

***Theorem** 3:* The number of n-bit Boolean functions satisfying SSAC is $2^{5\times 2^{n-3}}$.

V. CONCLUSION

In this paper, we discuss the balanced Boolean functions satisfying the SAC, propose the representation matrix of the transition function of Boolean function, and find some conditions which offer a standard for us to judge if a balanced Boolean function satisfies the SAC. Further, the SSAC of Boolean function is defined, and the recursion formula of the number of this kind of functions is obtained. It can be predicted that these basic research works in this paper will have a positive effect on the cryptography and information science.

ACKNOWLEDGMENT

This research was supported by the NSFC (Grants No. 11171084 and No. 60872093).

REFERENCES

[1] A. F. Webster and S. E. Tavares, *On the design of S-boxes*, in H. C. Williams, ed. Advances in Cryptology-Crypto'85, Lecture Notes in Computer Science, 218, pp. 523-534, 1986.

[2] H. Feistel, *Cryptography and computer privacy*, Scientific American, 228, pp. 15-23, 1973.

[3] J. B. Kam and G. I. Davida, *A structured design of substitution-permutation encryption networks*, IEEE Trans. Comput, 28 pp. 747-753, 1979.

[4] R. Forré *For the strict avalanche criterion: spectral properties of Boolean functions and an extended definition*, in S. Goldwasser, ed., Advances in Cryptology-Crypro'88, Lecture Notes in Computer Science. 403, pp. 450-468, 1990.

[5] S. Lloyd, *Counting functions satisfying a higher order strict avalanche criterion*, in: J.-J. Quisquater and J. Vandewalle, eds., Advances in Ctyptology-Eurocrypt'89, Lecture Notes in Computer Science, 434, pp. 63-74, 1990.

[6] S. Lloyd, *Characterising and counting functions satisfying the strict avalanche criterion of order (n-3)*, in: C. Mitchell, ed. Cryptography and Coding II, pp. 165-172, 1992.

[7] S. Lloyd, *Counting binary functions with certain cryptographic properties*, J. Cryptology, vol, 5, pp. 107-131, 1992.

[8] L. O. Connor, *An upper bound on the number of functions satisfying the Strict Avalanche Criterion*, Inform. Process. Lett, vol, 52 pp, 325-327, 1992.

[9] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts and J. Vandewalle, *Propagation characteristics of Boolean functions*, in LB. Damgard, ed., Advances in Cryptorogy-Eurocrypt'90, Lecture Notes in Computer Science, 473, pp. 161-173, 1991.

[10] T. W. Cusick, *Boolean functions satisfying a higher order strict avalanche criterion*, in: T. Helleseth, ed, Advances in Cryptology-Eurocrypt'93, Lecture Notes in Computer Science, 765, 102-117, 1994.

[11] T. W. Cusick, *Bounds on the number of functions satisfying the Strict Avalanche Criterion*, Inform. Process. Lett, 57, pp, 261-263, 1996.

[12] T. W. Cusick, P. Stanica, *Bounds on the number of functions satisfying the Strict Avalanche Criterion*, Inform. Process. Lett, 60, pp, 215-219, 1996.

[13] A. M. Youssef, S. E. Tavares, *Comment on "Bounds on the number of functions satisfying the Strict Avalanche Criterion"*, Inform. Process. Lett, 60, pp, 271-275, 1996.

[14] J. C. H. Castro, J. M. Sierra, A. Seznec, *The strict avalanche criterion randomness test*, Inform. Process. Lett, 68, pp, 1-7, 2005.

[15] C. K. Gupta, P. Sarkar, *Construction of Perfect Nonlinear and Maximally Nonlinear Multiple-Output Boolean Functions Satisfying Higher Order Strict Avalanche Criteria*, IEEE transactions on information theory, vol. 50, no. 11, 2004.

[16] D. Tang, W. G. Zhang, X. H. Tang, *Construction of balanced Boolean functions with high nonlinearity and good autocorrelation properties*, Designs codes and cryptography, 67, pp. 71-91, 2013.

[17] W. G. Zhang, F. Q. Jiang, D. Tang, *Construction of highly nonlinear resilient Boolean functions satisfying strict avalanche criterion*, Science China-Information Science, 57, no. 4, 2014.