Research Article

Prasanna R. Mishra and Shashi Kant Pandey*

On the algebraic immunity of multiplexer Boolean functions

https://doi.org/10.1515/jmc-2021-0027
received July 27, 2021; accepted June 22, 2022

Abstract: A multiplexer generator is a device that accepts two or more inputs and based on some logic sends one of them as output. In a special case when inputs to a multiplexer generator are 2^k bits and one of them is selected according to the value of a k-bit number, a multiplexer generator can be regarded as a Boolean function in $2^k + k$ variables. We call this generator a multiplexer Boolean function. Boolean functions serve as combiners and filters in cryptographic designs. The study of their cryptographic strength attracts the cryptographer because of the extremely simple and cost effective of their design. The study of algebraic attacks on multiplexer generators is another major concern to judging the suitability for its use in cryptographic designs. In this article, we calculate the algebraic immunity of the multiplexer Boolean function, which is not an obvious task in the case of a Boolean function like a multiplexer generator.

Keywords: Boolean function, multiplexer, algebraic immunity

MSC 2020: 06E30, 94C10, 94Dxx

1 Introduction

In telecommunications and computer networks, a multiplexer generator (or simply a mux) is a device [1] that selects one of several inputs and forwards the selected input into a single line. The main purpose of employing a multiplexer is to share an expensive resource. In digital circuit design, multiplexers are used to implement a Boolean logic. Here the inputs to a multiplexer are binary values (0 or 1). Most commonly, such a multiplexer selects one out of 2^k, $k \in \mathbb{N}$, input lines with the help of k select lines having binary values. A multiplexer of this kind is called a 2^k:1 multiplexer. The working of a 2^k:1 multiplexer is described as follows.

There are 2^k possible values that may be used to label the 2^k input lines. At a particular instant, the input whose label matches with the value at select lines is selected to be sent as output. For example, a 8:1 multiplexer is shown in Figure 1. There are eight input lines and three select lines. The eight input lines are labeled as 000, 001, 010, 011, 100, 101, 110, and 111. In this figure, the select line has values 110 and then the input at label 110 is sent as output which is 0. A 2^k:1 multiplexer takes $2^k + k$ binary values and gives the output as one binary value. Therefore, a 2^k:1 multiplexer can be regarded as a Boolean function in $2^k + k$ variables. We call this Boolean function a multiplexer Boolean function. Boolean functions are used as combiners and filters in cryptographic designs, especially in stream ciphers. A cryptographic Boolean function should be easy to implement, less resource consuming, and should be cryptographically robust [2,3]. In the next section, some essential preliminary definitions for the cryptographic analysis of a Boolean function are presented.

* **Corresponding author: Shashi Kant Pandey,** Department of Mathematics, MSI, GGSIP University, Delhi 110058, India, e-mail: shashikantshvet@gmail.com
Prasanna R. Mishra: SAG, DRDO, Metcalfe House, Delhi 110054, India, e-mail: prasanna.r.mishra@gmail.com

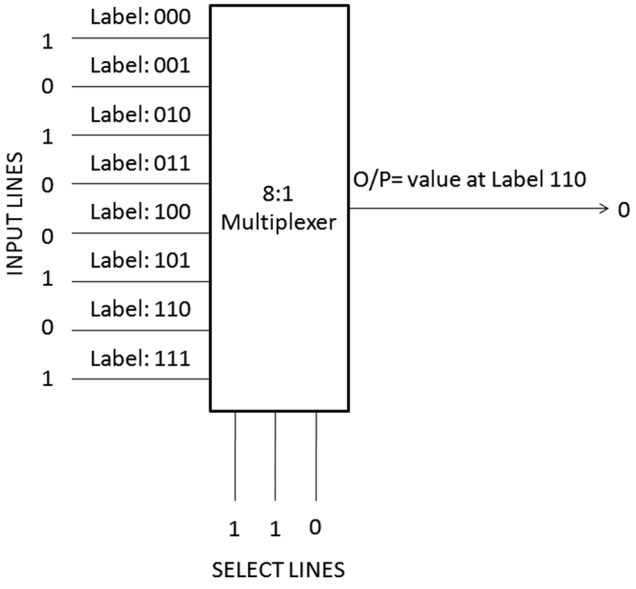

Figure 1: An 8:1 multiplexer.

2 Preliminaries

A Boolean function is a function from the n dimensional vector space $\mathbb{F}_2^n$ over $\mathbb{F}_2$ into $\mathbb{F}_2$. Here the set of all Boolean functions defined on $\mathbb{F}_2^n$ is denoted as $\mathcal{B}_n$. A Boolean function $f \in \mathcal{B}_n$ can be uniquely written as a multivariate polynomial in $\mathbb{F}_2[x_1, x_2, \ldots, x_n]$ as,

$$f(x_1, x_2, \ldots, x_n) = \bigoplus_{P \subseteq \{1,2,3,\ldots,n\}} a_P \prod_{p \in P} x_p. \tag{1}$$

The above representation of any Boolean function is called its *algebraic normal form* of f. The *algebraic degree* of f, denoted as $\deg(f)$, is defined as,

$$\deg(f) = \max\{|P| : P \subseteq \{1, 2, 3, \ldots, n\}, a_P \neq 0\},$$

where $|P|$ denotes cardinality of the set P. A Boolean function of algebraic degree one or less than one is called an *Affine Boolean function*. The set of all Affine Boolean functions in $\mathcal{B}_n$ is denoted as $\mathcal{A}_n$.

Let $f, g \in \mathcal{B}_n$. The Hamming distance between f and g is denoted as $d_H(f, g)$, and it is the number of points at which values of f and g differ. Mathematically,

$$d_H(f, g) = |\{\mathbf{x} \in \mathbb{F}_2^n : f(\mathbf{x}) \neq g(\mathbf{x})\}|.$$

The nonlinearity of a Boolean function is defined as,

$$nl(f) = \min_{g \in \mathcal{A}_n} d_H(f, g), \tag{2}$$

where $d_H(f, g)$ is the Hamming distance between f and g, which is equal to the Hamming weight of $f + g$. The Walsh transformation of a Boolean function is defined as a real valued function W_f on $\mathbb{F}_2^n$, and

$$W_f(w) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+w \cdot x}, \tag{3}$$

where $w \cdot x$ is an inner product of w and $x \in \mathbb{F}_2^n$, which is defined as $w_1x_1 + w_2x_2 + \ldots + w_nx_n$. Walsh transformation and nonlinearity of a Boolean function from $\mathcal{B}_n$ are connected with the following relation:

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{w \in \mathbb{F}_2^n} |W_f(w)|. \tag{4}$$

A Boolean function with flat Walsh spectrum value $2^{n/2}$ is called the bent Boolean function [4–6]. Obviously, it exists only for the even values of n. From this we can obtain the upper bound for the nonlinearity of a bent function which is $2^{n-1} - 2^{n/2-1}$.

For any Boolean function $f \in \mathcal{B}_n$, a nonzero function $g \in \mathcal{B}_n$ is called an annihilator of f if $fg = 0$, and the algebraic immunity of f, denoted by $AI(f)$, is the minimum value of degree (d) such that f or $f + 1$ admits an annihilator of degree d [7]. It is known that the algebraic immunity of an n-variable Boolean function is bounded above by $\lceil n/2 \rceil$ [8]. To resist algebraic attacks, a Boolean function should have a high algebraic immunity.

3 Multiplexer Boolean function

In this section, we define a multiplexer Boolean function in rigorous mathematical terms. As discussed earlier, a 2^k:1, $(k \geq 1)$ multiplexer has $k + 2^k$ inputs and one output. It may be regarded as a Boolean function in $\mathcal{B}_{k+2^k}$. It is known that $\mathbb{F}_2^{k+2^k} \cong \mathbb{F}_2^k \times \mathbb{F}_2^{2^k}$. Therefore, $X \in \mathbb{F}_2^{k+2^k}$ can be written as $X = (Y, Z)$, where $Y \in \mathbb{F}_2^k$ and $Z \in \mathbb{F}_2^{2^k}$. Let f be a 2^k:1, $k \geq 1$, multiplexer Boolean function. Let $Y = (y_0, y_1, \ldots, y_{k-1}) \in \mathbb{F}_2^k$ and $Z = (z_0, z_1, \ldots, z_{2^k-1}) \in \mathbb{F}_2^{2^k}$. Then output of the multiplexer Boolean function is z_t, where t is the decimal number whose binary digits are $y_0, y_1, \ldots, y_{k-1}$ in the order of their increasing significance. We identify t with Y. With this identification, the output of the multiplexer can be written as:

$$f(X) = f((Y, Z)) = z_Y.$$

Another presentation of multiplexer Boolean function, $f(X) = f((Y, Z))$ may be written as:

$$f(X) = \bigoplus_{Y' \in \mathbb{F}_2^k} \delta(Y \oplus Y') z_{Y'}, \tag{5}$$

where δ is the Kronecker delta function. It takes value 1 when all inputs are zero and zero otherwise. δ may be defined as:

$$\delta(Y) = (y_0 \oplus 1)(y_1 \oplus 1)\ldots(y_{k-1} \oplus 1).$$

Clearly, the degree of δ is k and hence from (5), algebraic degree of a multiplexer Boolean function f is $k + 1$. Trade off between algebraic immunity and degree of a Boolean function is one of the methods to fix its cryptographic suitability. Here we found that the degree of multiplexer Boolean function is quite low, which is not a common suitable choice for a filter or combiner functions. Further we study the algebraic immunity of multiplexer Boolean function and present the exact enumeration of algebraic immunity in the next section, which is quite practical measure in the direct use of multiplexer Boolean function as a filter function.

3.1 Algebraic immunity of multiplexer Boolean function

The statistical independence between input and output of a filter function is one of the preferred strength for its cryptographic use. Golić and Morgari have proposed the correlation attack on the multiplexer generator in ref. [9]. The future work on the design of immune multiplexer for enhanced keystream in stream cipher is emphasized in ref. [10].

The next theorem is an alternative definition of multiplexer Boolean function and with perpetuation of the notation in ref. [11], here we adopt the same notations.

Theorem 1. *A Boolean function*

$$f_k(y_0, y_1, \ldots, y_{k-1}, z_0, z_1, \ldots, z_{2^k-1}) \in \mathcal{B}_{k+2^k},$$

where $k \geq 1$ represents a $2^k : 1$ multiplexer Boolean function if and only if it satisfies the recurrence relation:

$$f_{k+1}(Y, Z) = (1 + y_k)f_k(Y, Z^1) + y_k f_k(Y, Z^2), \tag{6}$$

where

$$Y = (y_0, y_1, \ldots y_{k-1}, y_k),$$
$$Z^1 = \left(z_0, z_1, \ldots, z_{2^k-1}\right),$$
$$Z^2 = \left(z_{2^k}, z_{2^k+1}, \ldots, z_{2^{k+1}-1}\right),$$

and

$$Z = (Z^1, Z^2).$$

Proof. We apply induction on k to show that for $k \geq 1$, $f_k(y_0, y_1, \ldots, y_{k-1}, Z^1)$ represents a $2^k : 1$ multiplexer Boolean function. For $k = 1$ we have $f_1(y_0, z_0, z_1) = y_0z_0 + y_0z_1 + z_0$, which is clearly a $2 : 1$ multiplexer Boolean function. Now we assume that the theorem is true for $k \in \mathbb{N}$ and let $t = 2^k y_k + 2^{k-1}y_{k-1} + \ldots + y_0$. Now from the definition of multiplexer Boolean function and the value of t,

$$f_k(y_0, y_1, \ldots, y_{k-1}, Z^1) = z_{t-2^k y_k} \tag{7}$$

and

$$f_k(y_0, y_1, \ldots, y_{k-1}, Z^2) = z_{2^k+t-2^k y_k}. \tag{8}$$

Using (7) and (8) and from induction hypothesis we have,

$$f_{k+1}(Y, Z) = f_{k+1}(y_0, y_1, \ldots, y_{k-1}, y_k, z_0, z_1, \ldots, z_{2^{k+1}-1}) = z_{t-2^k y_k}(1 + y_k) + z_{2^k+t-2^k y_k}y_k. \tag{9}$$

There are two choices of $y_k \in \mathbb{F}_2$ and using them separately on (9), we obtain two following cases:
Case 1: $(y_k = 0)$

$$f_{k+1}(Y, Z) = f_{k+1}(y_0, y_1, \ldots, y_{k-1}, y_k, z_0, z_1, \ldots, z_{2^{k+1}-1}) = z_t \oplus 0 = z_t.$$

Case 2: $(y_k = 1)$

$$f_{k+1}(Y, Z) = f_{k+1}(y_0, y_1, \ldots, y_{k-1}, y_k, z_0, z_1, \ldots, z_{2^{k+1}-1}) = z_t.\ 0 \oplus z_t = z_t.$$

Finally from both of the above cases, the multiplexer Boolean function is

$$f_{k+1}(Y, Z) = f_{k+1}(y_0, y_1, \ldots, y_{k-1}, y_k, z_0, z_1, \ldots, z_{2^{k+1}-1}) = z_t.$$

Thus, induction completes the proof. □

In the next theorem, we show the correspondence between the annihilator of multiplexer Boolean function and its complement from the new representation of this function as discussed in the preceding theorem.

Theorem 2. *Let $X = (Y, Z)$, where $Y \in \mathbb{F}_2^k$, $Z \in \mathbb{F}_2^{2^k}$, and $f_k(X) = f_k((Y, Z))$ is a multiplexer Boolean function. Then there exists a one-to-one correspondence between annihilators of f_k and $f_k + 1$, such that each annihilator of f_k is mapped onto some annihilator of $f_k + 1$ of the same degree.*

Proof. Let $\mathbf{1} \in \mathbb{F}_2^{2^k}$ denote the vector $(1, 1, \ldots, 1)$. Now from the definition of multiplexer,

$$f_k(Y, Z) = z_Y \tag{10}$$

and

$$f_k(Y, Z + \mathbf{1}) = (z + 1)_Y = z_Y + 1. \tag{11}$$

Now (10) and (11) imply that

$$f_k(Y, Z + \mathbf{1}) = 1 + f_k(Y, Z). \tag{12}$$

Now from (12), it is clear that if $g_k(Y, Z)$ is any annihilator of $f_k(Y, Z)$ of degree d, then $g_k(Y, Z + \mathbf{1})$ is an annihilator of degree d of $1 + f_k(Y, Z)$. This ensures one-to-one correspondence of $f_k(Y, Z)$ and $1 + f_k(Y, Z)$. □

The next theorem presents an important information about the algebraic immunity of a multiplexer Boolean function. It refers to the idea that how the algebraic immunity increases with the increment of the number of variables in the recurrence relation of a multiplexer Boolean function presented in Theorem 1. To find the algebraic immunity of f_k, we have to find least degree annihilator of f_k only. We proceed with the following lemma which will be used to establish the aforementioned assertions about algebraic immunity of multiplexer Boolean function. Recall that $\mathcal{AI}(f)$ denotes the algebraic immunity of a multiplexer Boolean function.

Lemma 1. *For $k \geq 1$, let f_k be a multiplexer Boolean function. Then*

$$\mathcal{AI}(f_k) \leq k + 1.$$

Proof. We apply induction on k. For $k = 1$ we have $f_1(y_0, z_0, z_1) = y_0z_0 + y_0z_1 + z_0$. To find its annihilator, we see that

$$(y_0z_0 + y_0)\cdot f_1(y_0, z_0, z_1) = (y_0z_0 + y_0)\cdot(y_0z_0 + y_0z_1 + z_0) = 0,$$

which implies that $y_0z_0 + y_0$ annihilates f_1. Therefore, $\mathcal{AI}(f_1) \leq 2$. Now we assume the theorem to be true for some $k = m$, $m \in \mathbb{N}$. Then from the hypothesis,

$$\mathcal{AI}(f_m) \leq m + 1.$$

Let $g_m(Y, Z) = g_m(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^m-1})$ be an annihilator of the least degree, where $Y = (y_0, y_1, \ldots, y_{m-1})$ and $Z = (z_0, z_1, \ldots, z_{2^m-1})$, then $\deg g_m \leq m + 1$. We define function g_{m+1} as

$$g_{m+1}(Y, y_m, Z) = (1 + y_m)g_m(Y, Z) + y_m g_m\left(Y, z_{2^m}, z_{2^m+1}, \ldots, z_{2^{m+1}-1}\right).$$

It is easy to verify that g_{m+1} annihilates f_{m+1}. As

$$\deg g_m \leq \deg g_{m+1} \leq m + 2,$$

we conclude that

$$\mathcal{AI}(f_k) \leq k + 1.$$

This completes the proof of induction as well as that of lemma. □

Now in the next theorem, we prove the statement made before the preceding lemma.

Theorem 3. *For $k \geq 1$, let f_k be a multiplexer Boolean function. Then,*

$$\mathcal{AI}(f_k) = k + 1.$$

Proof. We apply induction on k. For $k = 1$ we have $f_1(y_0, z_0, z_1) = y_0z_0 + y_0z_1 + z_0$. Let L be the set of all linear and affine functions in variables y_0, z_0, and z_1, therefore, $L = \{y_0 + z_0 + z_1, y_0 + z_0 + z_1 + 1, y_0, z_0, z_1, y_0 + 1, z_0 + 1, z_1 + 1, y_0 + z_0, y_0 + z_0 + 1, z_0 + z_1, z_0 + z_1 + 1, y_0 + z_1, y_0 + z_1 + 1\}$. It is verified exhaustively that no one from L annihilates f_1. In other words, there is no one degree annihilator of f_1, while $y_0z_0 + y_0$ annihilates f_1. Therefore,

$$\mathcal{AI}(f_1) = 2.$$

Now we assume the theorem to be true for some $k = m$, $m \in \mathbb{N}$. Then from the induction hypothesis

$$\mathcal{AI}(f_m) = m + 1.$$

Let $g_{m+1}(y_0, y_1, \ldots, y_m, z_0, z_1, \ldots, z_{2^{m+1}-1})$ be a least positive degree annihilator of f_{m+1}. Without loss of generality, we may assume that

$$g_{m+1}(y_0, y_1, \ldots, y_m, z_0, z_1 \ldots, z_{2^{m+1}-1}) = (1 + y_m)g_m^1 + y_m g_m^2, \tag{13}$$

where g_m^1 and g_m^2 are two polynomials in $y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^{m+1}-1}$. We have

$$f_{m+1}g_{m+1} = 0. \tag{14}$$

Now (6) and (14) imply that

$$g_m^1 \times f_m(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^m-1}) = 0 \tag{15}$$

and

$$g_m^2 \times f_m(y_0, y_1, \ldots, y_{m-1}, z_{2^m}, z_{2^m+1}, \ldots, z_{2^{m+1}-1}) = 0. \tag{16}$$

From (15) and the induction hypothesis, we have either $g_m^1 = 0$ or $\deg g_m^1 \geq m + 1$. Similarly from (16) and induction hypothesis, either $g_m^2 = 0$ or $\deg g_m^2 \geq m + 1$. Observe that both g_m^1 and g_m^2 cannot be simultaneously zero as g_{m+1} is of positive degree. In view of Lemma 1,

$$m + 1 \leq \deg g_m^1 \leq m + 2 \text{ or } m + 1 \leq \deg g_m^2 \leq m + 2.$$

Further if any of the g_m^1 or g_m^2 is zero, other must have degree $m + 1$. In this case, we have $\mathcal{AI}(f_{m+1}) = m + 2$. Now we have two cases:

Case 1: $\{\deg g_m^1 = m + 2\}$

Either this is an impossible case (when $\deg(g_m^1 + g_m^2) \geq m + 2$) or $\mathcal{AI}(f_{m+1}) = m + 2$.

Case 2: $\{\deg g_m^1 = m + 1\}$

In this case, $\deg g_m^2$ must be equal to $m + 1$. For $\deg g_m^2 = m + 2$, $\deg g_{m+1} = m + 3$. Here we observe two things. First, g_m^1 must contain a term involving any of the variables $y_0, y_1, \ldots, y_{k-1}, z_0, z_1, \ldots, z_{2^m-1}$ as a function of $z_{2^m}, z_{2^m+1}, \ldots, z_{2^{m+1}-1}$ and it cannot annihilate

$$f_m(y_0, y_1, \ldots, y_{k-1}, y_k, z_0, z_1, \ldots, z_{2^{m+1}-1}).$$

The second thing we observe is that $g_m^1(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^{m+1}-1})$ is a family of annihilators of $f_m(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^m-1})$ given by different values of variables $z_{2^m}, z_{2^m+1}, \ldots, z_{2^{m+1}-1}$. We claim that in the ANF of g_m^1, there exists a term of degree $m + 1$ which does not involve any of the variables out of $z_{2^m}, z_{2^m+1}, \ldots, z_{2^{m+1}-1}$. This holds true because in such case putting $z_{2^m} = z_{2^m+1} = \ldots = z_{2^{m+1}-1} = 1$ will give an annihilator of $f_m(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^m-1})$ of a positive degree less than $m + 1$. Similarly, we can prove that in the ANF of g_m^2, there exists a term of degree $m + 1$, which does not involve any of the variables out of $z_0, z_1, \ldots, z_{2^m-1}$. Because of this fact, the degree of $g_m^1(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^{m+1}-1}) + g_m^2(y_0, y_1, \ldots, y_{m-1}, z_0, z_1, \ldots, z_{2^{m+1}-1})$ will not be less than $m + 1$. Consequently, $\mathcal{AI}(f_{m+1}) = m + 2$. Thus, from induction the theorem is proved. □

It is essential for cryptographic Boolean functions that they should have high nonlinearity and immunity from all algebraic attacks. In this article, we demonstrated the exact calculation of algebraic immunity of multiplexer Boolean function, which is still quite challenging in case of other well-known key stream generator such as product generator, Geffe generator, stop and go generator, alternating step generator, A5/1 generator, shrinking generator, and Knapsack generator. We found that multiplexer Boolean function or multiplexer generator has a significant algebraic immunity. An interesting finding in this work is the equality of the algebraic immunity and degree of a 2^k:1 multiplexer Boolean function. Therefore, the methods [12–17] based on the concatenation of two or more than two Boolean functions, chosen in some specific manner to

construct a highly nonlinear Boolean function is recommended in case of direct use of multiplexer Boolean function [18,19].

Acknowledgment: We thank all the anonymous reviewers for their valuable comments and suggestions.

Conflict of interest: Authors state no conflict of interest.

References

[1] Horowitz P, Hill W. The art of electronics. 2nd edition. Cambridge University Press; 1980, (1989).
[2] Cusick TW, Stanica P. Cryptographic Boolean functions and applications. Amsterdam: Elsevier/Academic Press; 2009.
[3] Carlet C. Vectorial Boolean functions for cryptography. http://www.math.univ-paris13.fr/ carlet/chap-vectorial-fcts-corr.pdf.
[4] Rothaus OS. On "bent" functions. J Combinatorial Theory Ser A. 1976;20(3):300–5.
[5] Carlet C, Dobbertin H, Leander G. Normal extensions of bent functions. IEEE Trans Inform Theory. 2004;50(11):2880–5.
[6] Mihaljević M, Gangopadhyay S, Paul G, Imai H. Generic cryptographic weakness of k-normal Boolean functions in certain stream ciphers and cryptanalysis of Grain-128. Period Math Hungar. 2012;65(2):205–27.
[7] Meier W, Pasalic E, Carlet C, Algebraic attacks and decomposition of Boolean functions. In: Advances in cryptology–EUROCRYPT 2004, Lecture Notes in Computer Science. Vol. 3027. Berlin: Springer; 2004. p. 474–91.
[8] Courtois NT, Meier W. Algebraic attacks on stream ciphers with linear feedback. In: Advances in cryptology–EUROCRYPT 2003. Lecture Notes in Computer Science. Vol. 2656. Berlin: Springer; 2003. p. 345–59.
[9] Golić JDj, Morgari G. Optimal correlation attack on the multiplexer generator. Inform Process Lett. 2009;109(15):838–41.
[10] Sulaiman HA, Othman MA, Othman MF, AbdRahim Y, Pee NC. Advanced Computer and Communication Engineering Technology: Proceedings of ICOCOE 2015, eBook ISBN 978-3-319-24584-3, Lecture Notes in Electrical Engineering. Vol. 362; 2016.
[11] Carlet C, Dalai DK, Gupta KC, Maitra S. Algebraic immunity for cryptographically significant Boolean functions: analysis and construction. IEEE Trans Inform Theory. 2006;52(7):3105–21.
[12] Nyberg K. Perfect nonlinear S-boxes. In: Advances in cryptology–EUROCRYPT '91 (Brighton, 1991), Lecture Notes in Computer Science. Vol. 547. Berlin: Springer: 1991. p. 378–86.
[13] Nyberg K. On the construction of highly nonlinear permutations. In: Advances in cryptology-EUROCRYPT '92 (Balatonfüred, 1992), Lecture Notes in Computer Science. Vol. 658. Berlin: Springer; 1992. p. 92–8.
[14] Beelen P, Leander G. A new construction of highly nonlinear S-boxes. Cryptogr Commun. 2012;4(1):65–77.
[15] Kavut S, Yücel MD. Generalized rotation symmetric and dihedral symmetric Boolean functions-9 variable Boolean functions with nonlinearity 242. In: Applied algebra, algebraic algorithms and error-correcting codes, Lecture Notes in Computer Science. Vol. 4851. Berlin: Springer; 2007. p. 321–9.
[16] Pasalic E. A design of Boolean functions resistant to (fast) algebraic cryptanalysis with efficient implementation. Cryptogr Commun. 2012;4(1):25–45.
[17] Zhang X-M, Zheng Y. Cryptographically resilient functions. IEEE Trans Inform Theory. 1997;43(5):1740–7.
[18] Wang Q, Carlet C, Stanica P, Tan CH. Cryptographic properties of the hidden weighted bit function. Discrete Appl Math. 2014;174:1–10.
[19] Carlet C, Feng K. An infinite class of balanced functions with optimal algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity. In: Advances in cryptology-ASIACRYPT 2008, Lecture Notes in Computer Science. Vol. 5350. Berlin: Springer; 2008. p. 425–40.