A First Order Recursive Construction of Boolean Function with Optimum Algebraic Immunity

Yindong Chen and Peizhong Lu

Fudan University, Shanghai 200433, China
{chenyd, pzlu}@fudan.edu.cn

Abstract. This paper proposed a first order recursive construction of Boolean function with optimum algebraic immunity. We also show that the Boolean functions are balanced and have good algebraic degrees.

Keywords: stream cipher, algebraic attacks, Boolean function, algebraic immunity

1 Introduction

Recently, algebraic attack has gained a lot of attention in cryptanalysis [1–8]. The main idea of algebraic attack is to deduce the security of a stream cipher to solve an over-defined system of multivariate nonlinear equations. To implement algebraic attack, attackters firstly construct equation system between the input bits (the secret key bits) and the output bits, then recover the input bits by solving the equation system with efficient methods such as Linearization, Relinearization, XL, Grönber bases, etc. [9–11].

Algebraic attack was firstly applied to LFSR (Linear Feedback Shift Register)-based stream cipher by Courtois and Meier [1] in 2003. By searching low degree annihilator, some LFSR-based stream ciphers such as Toyocrypt, LILI-128 [1], SFINKS [5], etc. were successfully attacked. The efficiency of algebraic attack is guaranteed by the existing of low degree multiple for any Boolean function [2]. That is, for any n-variable Boolean function, there exists multiple function with degree no more than $\lceil \frac{n}{2} \rceil$. The core of algebraic attack is to find out minimum degree nonzero annihilators of f or of $f+1$. This minimum degree is related to the complexity of algebraic attacks [2].

To resist algebraic attack, a new cryptographic property of Boolean functions which is known as *algebraic immunity* (AI) has been proposed by Meier *et al.* [2]. The AI of a Boolean function expresses its ability to resist standard algebraic attack. Thus the AI of Boolean function used in cryptosystem should be sufficiently high. Courtois and Meier [1, 2] showed that, for any n-variable Boolean function, its AI is bounded by $\lceil \frac{n}{2} \rceil$. If the bound is achieved, we say the Boolean function have optimum AI. Obviously, a Boolean function with optimum AI has strongest ability to resist standard algebraic attack. Therefore, the construction of Boolean functions with optimum AI is of great importance.

Dalai *et al.* [14, 15] presented Boolean functions with optimum AI in even variables by an recursive construction. It's a second order recursive construction. Further study [14] showed that the functions are not balanced (although it is possible to build balanced ones from them, but there would result in extra computation). Another class of constructions [16–18] contains symmetric functions. Being symmetric, they present a risk if attacks using this peculiarity can be found in the future. Moreover, they do not have high nonlinearities either [19]. Li [21–23] proposed a method to construct all $(2k+1)$-variable Boolean functions with optimum AI from one such given function. The construction has theoretical sense. But the computational complexity of the construction do not have been well studied. Carlet and Feng [24] proposed a well construction based on the Boolean functions' trace representation, recently. Their Boolean functions have not only optimum AI but also high nonlinearity. Furthermore, they also have a good behavior against fast algebraic attacks, at least for small values of the number of variables. The drawback of the construction is the high complexity of the computation for the value of $f(x)$.

Many researches show that, Boolean functions in odd variables have different properties from those in even variables, especially for ones with optimum AI. For example, odd variables Boolean functions with optimum AI must be balanced [25], the majority function is the only symmetric function depending on an odd number of variables which has maximum AI [26], etc.. Dalai's construction [14, 15] is also a case of that. Hence, people sometimes divide Boolean functions into two categories (odd variables Boolean functions and even variables Boolean functions) , and specify their research in one of them [21, 22, 25–29]. In this way, properties of Boolean functions in specific type are found. But the relation between Boolean functions in different party number of variables is omitted.

We propose a first order recursive construction of Boolean function with optimum AI. In the construction, we obtain even variable Boolean function from odd ones, and odd ones from even ones, too. Hence the construction has sense to study the relation between Boolean functions in different party number of variables.

The organization of the paper is as follows. In the following section we give some preliminaries about Boolean functions. In Section III, we present the construction of Boolean functions with optimum AI. Their cryptographic properties are studied in Section IV. Section V concludes the paper.

2 Preliminaries

Let $\mathbb{F}_2 = \{0, 1\}$, be the finite field with two elements. Then a *Boolean function* in n variables is defined as mapping from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. We denote by B_n the set of all n-variable Boolean functions. A basic representation of a Boolean function $f(x_1, \cdots, x_n)$ is by the output column of its *truth table*, i.e., a binary string of length 2^n,

$$f = [f(0, 0, \cdots, 0), f(1, 0, \cdots, 0), \cdots, f(1, 1, \cdots, 1)].$$

Sometimes, we may use a binary string of length 2^n to represent a n-variable Boolean function.

For an n-variables Boolean function f, we define its *support* and *offset* as

$$\begin{aligned} \mathrm{supp}(f) &= \{x \in \mathbb{F}_2^n | f(x) = 1\}, \\ \mathrm{offset}(f) &= \{x \in \mathbb{F}_2^n | f(x) = 0\}. \end{aligned}$$

and denote them by 1_f and 0_f respectively. The *Hamming weight* $\mathrm{wt}(f)$ of f is the size of $\mathrm{supp}(f)$, i.e., $\mathrm{wt}(f) = |\,\mathrm{supp}(f)|$. It counts the number of 1's in the truth table of f. We say f is *balanced*, if the truth table contains an equal number of 1's and 0's, i.e., $\mathrm{supp}(f) = \mathrm{offset}(f)$, implying $\mathrm{wt}(f) = 2^{n-1}$. The Hamming distance between two Boolean functions, f and g, is denoted by $\mathrm{d}(f, g)$ and is the number of places where their truth tables differ. Note that $\mathrm{d}(f, g) = \mathrm{wt}(f + g)$ (by abuse of notation, we also use $+$ to denote the addition in $\mathbb{F}_2$, i.e., the XOR);

Any Boolean function has a unique representation as a multivariate polynomial over $\mathbb{F}_2$, called the *algebraic normal form* (ANF):

$$\begin{aligned} f(x_1, \cdots, x_n) =& a_0 + \sum_{1\le i\le n} a_i x_i + \\ & \sum_{1\le i\le j\le n} a_{ij} x_i x_j + \cdots + a_{12\cdots n} x_1 x_2 \cdots x_n, \end{aligned}$$

where the coefficients $a_0, a_i, a_{ij}, \cdots, a_{12\cdots n} \in \mathbb{F}_2$. The *algebraic degree* $\deg(f)$ of f is the number of variables in the highest order term with nonzero coefficient. A Boolean function is *affine* if it has algebraic degree at most 1 and we denote by A_n the set of all affine functions in n variables.

The *nonlinearity* of an n-variable function f is its distance from the set of all n-variable affine functions, i.e.,

$$\mathrm{nl}(f) = \min_{g\in A_n} (\mathrm{d}(f, g)).$$

To be cryptographically secure [30, 31], Boolean functions used in cryptographic systems must be balanced to prevent the system from leaking statistical information on the plaintext when the ciphertext is known, have high algebraic degree to counter linear synthesis by Berlekamp-Massey algorithm, have high order of correlation immunity to counter correlation attacks, and have high nonlinearity to withstand linear attacks and correlation attacks.

Recently, it has been identified that any combining or filtering should not have a low-degree-multiple. More precisely, it is shown in [1] that, given any n-variable Boolean function f, it is always possible to get a Boolean function g with degree at most $\lceil \frac{n}{2} \rceil$ such that $f \cdot g$ has degree at most $\lceil \frac{n}{2} \rceil$. Therefore, while choosing a Boolean function f, the cryptosystem designer should avoid that the degree of $f \cdot g$ falls much below $\lceil \frac{n}{2} \rceil$ with a nonzero Boolean function g whose degree is also much below $\lceil \frac{n}{2} \rceil$. Otherwise, resulting low degree multivariate relations between key bits and output bits of Boolean function f will allow a very efficient attack. As observed in [1, 2], it is necessary to check that f and $f + 1$ do not admit nonzero annihilators of low degrees.

Definition 1. *Given $f \in B_n$, we define*

$$\mathrm{Ann}(f) = \{g \in B_n | f \cdot g = 0\}.$$

Any function $g \in \mathrm{Ann}(f)$ is called an annihilator of f.

It's explicit that a function g is an annihilator of f if and only if g takes value 0 on $\mathrm{supp}(f)$, i.e.,

$$g \in \mathrm{Ann}(f) \Leftrightarrow 1_f \subseteq 0_g.$$

Definition 2. *Given $f \in B_n$, we define its algebraic immunity, denote by $\mathrm{AI}_n(f)$, as the minimum degree of all nonzero annihilators of f or $f+1$, i.e.,*

$$\mathrm{AI}_n(f) = \min\{\deg(g) | 0 \neq g \in \mathrm{Ann}(f) \cup \mathrm{Ann}(f+1)\}.$$

We usually denote $\mathrm{AI}_n(f)$ by $\mathrm{AI}(f)$ for short, when there is no confusion about the number of variables.

Note that $\mathrm{AI}(f) \leq \deg(f)$, since $f \cdot (f+1) = 0$. As f or $f+1$ must have an annihilator at an algebraic degree $\leq \lceil \frac{n}{2} \rceil$ [1], we have $\mathrm{AI}(f) \leq \lceil \frac{n}{2} \rceil$. If an n-variable Boolean function f satisfies that $\deg(f) = \lceil \frac{n}{2} \rceil$, we say it has optimum AI. The AI of a Boolean function expresses its ability to resist standard algebraic attack. So, Boolean functions with higher AI (even optimum AI) is preferred in cryptosystem. Note that although AI is not a property that can resist all kinds of algebraic attacks, but clearly still a necessary one.

3 A First Order Recursive Construction of Boolean Function with Optimum Algebraic Immunity

From now on, we use a binary string of length 2^n to express an n-variable Boolean function, and denote by "$\|$" the concatenation of binary strings.

For example, let $s, t \in B_2$, and $s = x_1x_2 + x_2 + 1, t = x_1x_2 + x_2$. In the truth table representation, they are $s = 1101, t = 0010$. Let $u = s\|t = 11010010$, then $u \in B_3$, and $u = x_1x_2 + x_2 + x_3 + 1$.

For the denotation "$\|$", the following proposition holds.

Proposition 1. *Given $f_1, f_2 \in B_n$, let $f = f_1\|f_2$, then*

i) $f \in B_{n+1}$*, and* $f = f_1 + x_{n+1}(f_1 + f_2)$*;*
ii) $\deg(f_1), \deg(f_2) \leq \deg(f)$*;*
iii) *for any* $g \in \mathrm{Ann}(f)$*, decompose it as* $g = g_1\|g_2$ *where* $g_1, g_2 \in B_n$*, then* $g_1 \in \mathrm{Ann}(f_1)$ *and* $g_2 \in \mathrm{Ann}(f_2)$*.*

Now, we're proposing a first order recursive construction of Boolean function, and then proving that they have optimum AI.

Construction 1.

$$\begin{cases} \phi_{n+1} = \phi_n \| \phi_n^1, \\ \phi_{n+1}^i = \phi_n^{i-1} \| \phi_n^{i+1}, \end{cases} \tag{1}$$

with base step $\phi_n^0 = \phi_n, \phi_1^j = x_1 + (j \mod 2)$, $i, n \geq 1, j > 0$.

By Proposition 1, (1) can be transformed into the algebraic form as following:

$$\begin{cases} \phi_{n+1} = \phi_n + x_{n+1}(\phi_n + \phi_n^1), \\ \phi_{n+1}^i = \phi_n^{i-1} + x_{n+1}(\phi_n^{i-1} + \phi_n^{i+1}). \end{cases} \tag{2}$$

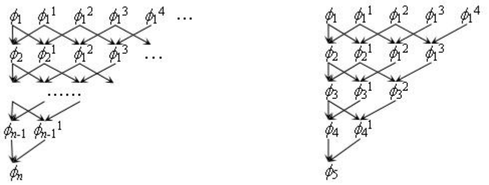
We list part of the Boolean functions in Table 1 and Table 2. To understand the recursion more precisely, see Fig. 1.

Table 1: Boolean functions in Construction 1 (truth table)

$\phi_1 = 10$	$\phi_1^1 = 01$	$\phi_1^2 = 10$	$\phi_1^3 = 01 \cdots$
$\phi_2 = 1001$	$\phi_2^1 = 1010$	$\phi_2^2 = 0101 \cdots$	
$\phi_3 = 10011010$	$\phi_3^1 = 10010101 \cdots$		
$\phi_4 = 1001101010010101 \quad \cdots$			

Table 2: Boolean functions in Construction 1 (ANF)

$\phi_1 = x_1 + 1$	$\phi_1^1 = x_1$	$\phi_1^2 = x_1 + 1$	$\phi_1^3 = x_1 \cdots$
$\phi_2 = x_2 + x_1 + 1$	$\phi_2^1 = x_1 + 1$	$\phi_2^2 = x_1$	$\cdots$
$\phi_3 = x_2x_3 + x_2 + x_1 + 1$	$\phi_3^1 = x_2x_3 + x_3 + x_2 + x_1 + 1$		$\cdots$
$\phi_4 = x_3x_4 + x_2x_3 + x_2 + x_1 + 1 \quad \cdots$			


ϕ_1 ϕ_1^1 ϕ_1^2 ϕ_1^3 ϕ_1^4 ···
ϕ_2 ϕ_2^1 ϕ_1^2 ϕ_1^3 ···
......
ϕ_{n-1} ϕ_{n-1}^1
ϕ_n
ϕ_1 ϕ_1^1 ϕ_1^2 ϕ_1^3 ϕ_1^4
ϕ_2 ϕ_2^1 ϕ_1^2 ϕ_1^3
ϕ_3 ϕ_3^1 ϕ_3^2
ϕ_4 ϕ_4^1
ϕ_5

(a) The recursion of ϕ_n (b) The recursion of ϕ_5

Fig. 1: The recursion of Boolean functions in Construction 1

To prove that ϕ_n has optimum AI, we need intermediate results. For technical reasons, during our proofs, we will encounter certain situations when the degree of a function is negative. As such functions do not exist, we will replace them by function 0.

Lemma 1. *Given $n \geq 1$, assume that the function $\phi_n \in B_n$ has been generated by Construction 1 and $\mathrm{AI}(\phi_t) = \lceil \frac{n}{2} \rceil$ for $1 \leq t \leq n$. If, for some $0 \leq i \leq n-2$, there exists $g \in \mathrm{Ann}(\phi_n^i)$ and $h \in \mathrm{Ann}(\phi_n^{i+1})$ such that $\deg(g+h) \leq \lfloor \frac{n-i}{2} \rfloor - 1$, then $g = h$.*

Proof. We prove it by induction on n.

For the base step $n = 1$, $0 \leq i \leq -1$ implies that functions in the assumption cannot exist, i.e., $g = h = 0$.

Now we prove the inductive step. Assume that, for $n < k$, the induction assumption holds (for every $0 \leq i \leq n-2$). We show it for $n = k$ and for every $0 \leq i \leq n-2$.

Suppose that there exists $g \in \mathrm{Ann}(\phi_k^i)$ and $h \in \mathrm{Ann}(\phi_k^{i+1})$ such that $\deg(g+h) \leq \lfloor \frac{k-i}{2} \rfloor - 1$. We decompose g and h as $g = g_1 \| g_2, h = h_1 \| h_2$ where $g_1, g_2, h_1, h_2 \in B_{k-1}$. By Proposition 1, we have:

$$\begin{aligned} & g + h = (g_1 + h_1) + x_k(g_1 + h_1 + g_2 + h_2), \qquad (3) \\ & g_1 \in \mathrm{Ann}(\phi_{k-1}^{i-1}), g_2 \in \mathrm{Ann}(\phi_{k-1}^{i+1}), \\ & h_1 \in \mathrm{Ann}(\phi_{k-1}^{i}), h_2 \in \mathrm{Ann}(\phi_{k-1}^{i+2}). \end{aligned}$$

And $\deg(g_1 + h_1) \leq \deg(g+h) \leq \lfloor \frac{k-i}{2} \rfloor - 1$.

1) To prove $g_1 = h_1$

a) If $i = 0$, then $g_1 + h_1 \in \mathrm{Ann}(\phi_{k-1})$ since $g_1, h_1 \in \mathrm{Ann}(\phi_{k-1})$. By hypothesis, $\mathrm{AI}(\phi_{k-1}) = \lceil \frac{k-1}{2} \rceil$. Since $\deg(g_1 + h_1) \leq \deg(g+h) \leq \lfloor \frac{k}{2} \rfloor - 1 \leq \mathrm{AI}(\phi_{k-1})$, we have $g_1 + h_1 = 0$, according to induction assumption. That is $g_1 = h_1$.

b) If $i > 0$, then $\deg(g_1 + h_1) \leq \lfloor \frac{k-i}{2} \rfloor - 1 = \left\lfloor \frac{(k-1)-(i-1)}{2} \right\rfloor - 1$, thus $g_1 = h_1$, according to induction assumption.

2) To prove $g_2 = h_2$

Equation (3) changes into $g + h = x_k(g_2 + h_2)$, since $g_1 + h_1 = 0$. Thus $\deg(g_2 + h_2) = \deg(g+h) - 1 \leq \lfloor \frac{k-i}{2} \rfloor - 1 - 1 = \left\lfloor \frac{(k-1)-(i+2)}{2} \right\rfloor - 1$, then $g_2 = h_2$, according to induction assumption.

Hence we get $g + h = 0$, i.e., $g = h$ which finishes the proof. □

Lemma 2. *Given $n \geq 1$, assume that the function $\phi_n \in B_n$ has been generated by Construction 1 and $\mathrm{AI}(\phi_t) = \lceil \frac{t}{2} \rceil$ for $1 \leq t \leq n$. If, for some $0 \leq i \leq n-2$, there exists $g \in \mathrm{Ann}(\phi_n^i) \cap \mathrm{Ann}(\phi_n^{i+1})$ such that $\deg(g) \leq \lfloor \frac{n+i}{2} \rfloor$, then $g = 0$.*

Proof. We prove Lemma 2 by induction on n.

For the base step $n = 1$, it can easily be checked.

Now we prove the inductive step. Assume that, for $n < k$, the induction assumption holds (for every $0 \leq i \leq n-2$). We show it for $n = k$ and for every $0 \leq i \leq n-2$.

Suppose that there exists $g \in \mathrm{Ann}(\phi_k^i) \cap \mathrm{Ann}(\phi_k^{i+1})$ such that $\deg(g) \leq \left\lfloor \frac{k+i}{2} \right\rfloor$. We decompose g as $g = g_1 \| g_2$ where $g_1, g_2 \in B_{k-1}$, then

$$g = g_1 + x_k(g_1 + g_2), \tag{4}$$

according to Proposition 1.

1) If $i = 0$, then

$$\begin{cases} \phi_k = \phi_{k-1} \| \phi_{k-1}^1 \\ \phi_k^1 = \phi_{k-1} \| \phi_{k-1}^2 \end{cases}.$$

By Proposition 1, we have

$$g_1 \in \mathrm{Ann}(\phi_{k-1}), \text{ and } g_2 \in \mathrm{Ann}(\phi_{k-1}^1) \cap \mathrm{Ann}(\phi_{k-1}^2).$$

Since $\deg(g) \leq \left\lfloor \frac{k}{2} \right\rfloor$, we get $\deg(g_2) \leq \left\lfloor \frac{k}{2} \right\rfloor = \left\lfloor \frac{(k-1)+1}{2} \right\rfloor$.
Thus $g_2 = 0$, according to induction assumption.

Then Equation (4) changes into $g = (1 + x_k)g_1$, which implies $\deg(g_1) = \deg(g) - 1 \leq \left\lfloor \frac{k}{2} \right\rfloor - 1 < \left\lfloor \frac{k-1}{2} \right\rfloor$. Since $g_1 \in \mathrm{Ann}(\phi_{k-1})$ and, by hypothesis, $\mathrm{AI}(\phi_{k-1}) = \left\lfloor \frac{k-1}{2} \right\rfloor$, there would be $g_1 = 0$. And then $g = 0$.

2) If $i > 0$, then

$$\begin{cases} \phi_k^i = \phi_{k-1}^{i-1} \| \phi_{k-1}^{i+1} \\ \phi_k^{i+1} = \phi_{k-1}^{i} \| \phi_{k-1}^{i+2} \end{cases}.$$

By Proposition 1, we have

$$g_1 \in \mathrm{Ann}(\phi_{k-1}^{i-1}) \cap \mathrm{Ann}(\phi_{k-1}^{i}),$$
$$g_2 \in \mathrm{Ann}(\phi_{k-1}^{i+1}) \cap \mathrm{Ann}(\phi_{k-1}^{i+2}).$$

Since $\deg(g_2) \leq \deg(g) \leq \left\lfloor \frac{k+i}{2} \right\rfloor = \left\lfloor \frac{(k-1)+(i+1)}{2} \right\rfloor$, we have $g_2 = 0$, according to induction assumption.

Then Equation (4) changes into $g = (1 + x_k)g_1$, which implies $\deg(g_1) = \deg(g) - 1 \leq \left\lfloor \frac{k+i}{2} \right\rfloor - 1 < \left\lfloor \frac{(k-1)+(i-1)}{2} \right\rfloor$. Thus $g_1 = 0$, according to induction assumption. And then $g = 0$.

Hence we get $g = 0$, by 1) and 2). This completes the proof. □

Theorem 1. *The function ϕ_n obtained in Construction 1 has optimum algebraic immunity, for every $n \geq 1$, i.e.,*

$$\mathrm{AI}(\phi_n) = \left\lceil \frac{n}{2} \right\rceil.$$

Proof. We prove Theorem 1 by induction on n.

For the base step $n = 1$, it can easily be checked.

Now we prove the inductive step. Assume that, for $n < k$, the induction assumption holds. We show it for $n = k$.

We have to prove that any nonzero function g such that $g{\cdot}\phi_k = 0$ has degree at least $\lceil \frac{k}{2} \rceil$ (proving that any nonzero function g such that $g{\cdot}(\phi_k{+}1) = 0$ has degree at least $\lceil \frac{k}{2} \rceil$ is similar). Suppose that such a function g with $\deg(g) < \lceil \frac{k}{2} \rceil$ exists. Then, g can be decomposed as $g = g_1 \| g_2$ where $g_1, g_2 \in B_{k-1}$. By Proposition 1, we have

$$\begin{aligned} &g = g_1 + x_k(g_1 + g_2), \\ &g_1 \in \mathrm{Ann}(\phi_{k-1}), g_2 \in \mathrm{Ann}(\phi_{k-1}^1). \end{aligned} \tag{5}$$

By Equation (5), we can see

$$\deg(g_1 + g_2) \le \deg(g) - 1 < \left\lceil \frac{k}{2} \right\rceil - 1 = \left\lfloor \frac{k-1}{2} \right\rfloor,$$

i.e., $\deg(g_1 + g_2) \le \lfloor \frac{k-1}{2} \rfloor - 1$. Thus we get $g_1 = g_2$ by Lemma 1. Then

$$\begin{aligned} &g = (1 + x_k)g_1, \\ &g_1 \in \mathrm{Ann}(\phi_{k-1}) \cap \mathrm{Ann}(\phi_{k-1}^1). \end{aligned} \tag{6}$$

By Equation (6), we can see

$$\deg(g_1) = \deg(g) - 1 < \left\lceil \frac{k}{2} \right\rceil - 1 = \left\lfloor \frac{k-1}{2} \right\rfloor,$$

i.e., $\deg(g_1) \le \lfloor \frac{k-1}{2} \rfloor - 1$. Thus we get $g_1 = 0$ by Lemma 2.

Hence $g = 0$, which completes the proof. □

4 The analysis of other cryptographic properties

In this section, we will analyze other cryptographic properties of the constructed Boolean functions. In the analysis, we lay emphasis on their balance, algebraic degree and nonlinearity.

4.1 Balance

From the recursive definition, we can see that ϕ_n^i's $(n > 1, i \ge 0)$ truth table is concatenated by ϕ_{n-1}^{i-1}'s and ϕ_{n-1}^{i+1}'s. If ϕ_{n-1}^{i-1} and ϕ_{n-1}^{i+1} are both balanced, then ϕ_n^i is of course balanced, too. Since the base functions $\phi_1^j (j \ge 0)$ are balanced (it can be easily checked), we can easily infer the following property by induction on n.

Property 1. The Boolean function $\phi_n^i(n > 1, i \ge 0)$ obtained in Construction 1 is balanced. Specially, ϕ_n is balanced.

4.2 Algebraic Degree

Observing the Boolean functions' ANFs in Table 2, we can find that every function has and only has one term containing x_1, and the unique term is x_1. Actually, we have the following property.

Property 2. Let $\varphi_n^i = x_1 + \phi_n^i (n > 1, i \geq 0)$, then φ_n^i does not have any term containing variable x_1.

Proof. We prove it by induction on n.

For the base step $n = 1$, $\varphi_i^i = 1 + (i \mod 2)$, the assertion obviously holds.

Assume that the induction assumption holds until $n < k$, then we show it for $n = k$.

1) If $i = 0$, then

$$\begin{aligned}\varphi_k^0 &= x_1 + \phi_k^0 = x_1 + x_k(\phi_{k-1}^0 + \phi_{k-1}^1) \\ &= x_1 + (x_1 + \varphi_{k-1}^0) + x_k(x_1 + \varphi_{k-1}^0 + x_1 + \varphi_{k-1}^1) \\ &= \varphi_{k-1}^0 + x_k(\varphi_{k-1}^0 + \varphi_{k-1}^1).\end{aligned}$$

2) If $i > 0$, then

$$\begin{aligned}\varphi_k^i &= x_1 + \phi_k^i = x_1 + x_k(\phi_{k-1}^{i-1} + \phi_{k-1}^{i+1}) \\ &= x_1 + (x_1 + \varphi_{k-1}^{i-1}) + x_k(x_1 + \varphi_{k-1}^{i-1} + x_1 + \varphi_{k-1}^{i+1}) \\ &= \varphi_{k-1}^{i-1} + x_k(\varphi_{k-1}^{i-1} + \varphi_{k-1}^{i+1}).\end{aligned}$$

According to induction assumption, none of φ_{k-1}^0, φ_{k-1}^1, φ_{k-1}^{i-1} and φ_{k-1}^{i+1} has any term containing x_1. Thus, φ_k^i does not have any term containing x_1, either. This completes the proof. □

Property 2 shows that, for any $n > 1, i \geq 0$, ϕ_n^i has and only has one term containing variable x_1, and furthermore, the unique term is x_1. That means we just need to consider the terms excluding variable x_1, when analyze ϕ_n's $(n > 1)$ algebraic degree. We denote by c_n and c_n^i the 2-variable (x_2 and x_3) functions equal to the factors of $x_3 x_4 \cdots x_n$ in the ANFs of ϕ_n and ϕ_n^i, for $n > 3$. From ϕ_n^i's recursion, we can easily infer that $c_{n+1} = c_n + c_n^1, c_{n+1}^i = c_{n+1}^{i-1} + c_{n+1}^{i+1}$. It's difficult to compute c_n and c_n^i from this recursion directly. With the method used in proving Proposition 5 of [14], we get the following property.

Property 3. Let $c_n (n > 3)$ be the 2-variable (x_2 and x_3) function equal to the factor of $x_3 x_4 \cdots x_n$ in the ANF of ϕ_n. And define that: $c_0 = \text{cst}, c_1 = \text{cst}, c_2 = x_2 x_3 + x_2 + x_3 + \text{cst}$. Then we have

$$c_n = \sum_{t=0}^{\lfloor \log_2 n \rfloor} c_{n-2^t} + \text{cst}, \tag{7}$$

where cst is some bit depending on n.

Table 3: Values of c_n for $n \leq 21$

n	c_n	n	c_n
0	cst	11	cst
1	cst	12	cst
2	$x_2x_3 + x_2 + x_3 + \text{cst}$	13	cst
3	$x_2x_3 + x_2 + \text{cst}$	14	cst
4	$x_3 + \text{cst}$	15	cst
5	$x_2x_3 + x_2 + x_3 + \text{cst}$	16	cst
6	$x_3 + \text{cst}$	17	$x_2x_3 + x_2 + x_3 + \text{cst}$
7	cst	18	$x_3 + \text{cst}$
8	cst	19	cst
9	$x_2x_3 + x_2 + x_3 + \text{cst}$	20	cst
10	$x_3 + \text{cst}$	21	cst

We compute c_n for $n \leq 21$, and list them in Table 3.

According to Property 3 and Table 3, it can be easily inferred by induction that

$$c_n(n \geq 4) = \begin{cases} x_2x_3 + x_2 + x_3 + \text{cst} & n = 2^k + 1, \\ x_3 + \text{cst} & n = 2^k + 2, \\ \text{cst} & \text{others.} \end{cases} \tag{8}$$

Since $\phi_n(n \geq 4)$ has no high order terms that containing x_1 (by Property 3), $\deg(\phi_n) \leq n - 1$. According the meaning of c_n, if $\deg(c_n) > 0$, then $\deg(\phi_n) = \deg(c_n) + (n - 3)$. Thus, formula (8) can be deduced into

$$\deg(\phi_n)(n \geq 4) = \begin{cases} n - 1 & n = 2^k + 1, \\ n - 2 & n = 2^k + 2. \end{cases} \tag{9}$$

From formula (2), it's clearly that $\deg(\phi_{k+1}) \geq \deg(\phi_k)$, i.e., $\deg(\phi_n)$ is an increasing function of n. To sum up, we get the following property:

Property 4. Given $n \geq 4$, let $k = \lfloor \log_2 n \rfloor$, then

$$\deg(\phi_n) \begin{cases} = n - 1 & n = 2^k + 1 \text{ or } n = 2^k + 2, \\ \geq n - 2 & \text{others.} \end{cases}$$

We compute $\deg(\phi_n)$ for $n \leq 14$, and list them in Table 4.

Table 4: Values of $\deg(\phi_n)$ for $n \leq 14$

n	3	4	5	6	7	8	9	10	11	12	13	14
$\deg(\phi_n)$	2	2	4	4	5	5	8	8	9	9	11	11

4.3 Nonlinearity

Let $\varphi_n = \varphi_n^0$. Property 2 says that, φ_n^i has no terms containing variable x_1. Thus φ_n^i can be viewed as an $(n-1)$-variable Boolean function in variables $x_2, x_3, \cdots, x_n$. We shall prove that, as an $(n-1)$-variable Boolean function, φ_n has optimum algebraic immunity.

Proposition 2. *Let* $\varphi_n^i = x_1 + \phi_n^i, \varphi_n = \varphi_n^0 (n > 1, i \geq 0)$*, then* φ_n *has optimum algebraic immunity, i.e.,*

$$\mathrm{AI}_{n-1}(\varphi_n) = \left\lceil \frac{n-1}{2} \right\rceil.$$

Proof. In the proof of Property 2, φ_n^i had been showed to have the same recursion as ϕ_n^i. Hence, φ_n could be proved to have optimum algebraic immunity similar to ϕ_n. □

It's easy to check that, for any n-variable Boolean function $f \in B_n$, there is $\mathrm{nl}(x_{n+1} + f) = 2 \cdot \mathrm{nl}(f)$. Since $\phi_n = x_1 + \varphi_n$ and φ_n has no terms containing x_1, we get $\mathrm{nl}(\phi_n) = 2 \cdot \mathrm{nl}(\varphi_n)$.

Loabnov [32] found a relation between Boolean function's algebraic immunity and nonlinearity, i.e.,

Proposition 3. *For any* n*-variable Boolean function* $f \in B_n$*, let* $k = \mathrm{AI}_n(f)$*, then*

$$\mathrm{nl}(f) \geq 2^{n-1} - \sum_{i=0}^{n-k} \binom{n-1}{i} = 2 \sum_{i=0}^{k-2} \binom{n-1}{i}.$$

As to φ_n, Proposition 2 shows that $k = \mathrm{AI}_{n-1}(\varphi_n) = \lceil \frac{n-1}{2} \rceil$, thus

$$\mathrm{nl}(\phi_n) = 2 \cdot \mathrm{nl}(\varphi_n) \geq 4 \sum_{i=0}^{k-2} \binom{n-2}{i}.$$

If $n = 2k$, then

$$\begin{aligned} 4\sum_{i=0}^{k-2} \binom{n-2}{i} &= 2\left(\sum_{i=0}^{k-2} \binom{n-2}{i} + \sum_{i=0}^{k-3} \binom{n-2}{i} + \binom{n-2}{k-2} \right) \\ &> 2\left(\sum_{i=0}^{k-2} \binom{n-2}{i} + \sum_{i=0}^{k-3} \binom{n-2}{i} + 1 \right) \\ &= 2\sum_{i=0}^{k-2} \binom{n-1}{i}. \end{aligned}$$

If $n = 2k+1$, then

$$\begin{aligned}4\sum_{i=0}^{k-2}\binom{n-2}{i} &= 2\left(\sum_{i=0}^{k-2}\binom{n-2}{i} + \sum_{i=0}^{k-2}\binom{n-2}{i}\right)\\ &< 2\left(\sum_{i=0}^{k-1}\binom{n-2}{i} + \sum_{i=0}^{k-2}\binom{n-2}{i} + 1\right)\\ &= 2\sum_{i=0}^{k-1}\binom{n-1}{i}.\end{aligned}$$

Hence, we have the following property:

Property 5. For the function ϕ_n obtained in Construction 1,

$$\mathrm{nl}(\phi_n) \geq \begin{cases} 4\displaystyle\sum_{i=0}^{\lceil\frac{n-1}{2}\rceil-2}\binom{n-2}{i} & n = 2k, \\ 2\displaystyle\sum_{i=0}^{\lceil\frac{n}{2}\rceil-2}\binom{n-1}{i} & n = 2k+1. \end{cases} \tag{10}$$

We compute $\mathrm{nl}(\phi_n)$ for $n \leq 14$, and list them in Table 5. It can be observed that the value of $\mathrm{nl}(\phi_n)$ in Table 5 all reach the low bound of formula (10).

Table 5: Values of $\mathrm{nl}(\phi_n)$ for $n \leq 14$

n	3	4	5	6	7	8	9	10	11	12	13	14
$\mathrm{nl}(\phi_n)$	2	4	10	20	44	88	186	372	772	1544	3172	6344

5 Conclusion

In this paper, we proposed a first order recursive construction of Boolean function with optimum algebraic immunity. It's the first one of such constructions. By the construction, we obtained (n+1)-variable Boolean function with optimum AI from n-variable ones. The construction has sense to study the relation between the odd variables Boolean functions and even variables Boolean functions. We also analyzed other cryptographic properties of the constructed Boolean functions, which showed that they're balanced and have good algebraic degrees.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (Grant No. 60673082), and the Special Funds of Authors of Excellent Doctoral Dissertation in China (Grant No. 200084).

References

1. N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback[A]. Advances in Cryptology-Eurocrypt 2003[C], Berlin: Springer-Verlag, 2003, 345-359
2. W. Meier, E. Pasalic, and C. Carlet. Algebraic attacks and decomposition of Boolean functions[A]. Advances in Cryptology-Eurocrypt 2004[C], Berlin: Springer-Verlag, 2004, 474-491
3. F. Armknecht and M. Krause. Algebraic Attacks on Combiners with Memory[A]. Advances in Cryptology-Crypto 2003[C], Berlin: Springer-Verlag, 2003, 162-175
4. N. Courtois. Algebraic Attacks on Combiners with Memory and Several Outputs[A]. Information security and cryptology 2004 (ICISC 2004), LNCS 3506, 2005, 3-20
5. N. Courtois. Cryptanalysis of SFINKS[A]. Information Security and Cryptology 2005 (ICISC 2005)[C]. Berlin: Springer-Verlag, 2006, 261-269
6. L. M. Batten. Algebraic Attacks over GF(q) [A]. Progress in Cryptology-Indocrypt 2004[C], Berlin: Springer-Verlag, 2004, 84-91
7. J. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases[A]. Advances in Cryptology-Crypto 2003[C], Berlin: Springer-Verlag, 2003, 44-60
8. F. Armknecht. On the Existence of low-degree Equations for Algebraic Attacks[EB/OL]. http://eprint.iacr.org/2004/185
9. N. Courtois, A. Klimov, J. Patarin, *et al.* Efficient algorithms for solving overdefined systems of multivariate polynomial equations[A]. Advances in Cryptology-Eurocrypt 2000[C], Berlin: Springer-Verlag, 2000, 392-407
10. A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization[A]. Advances in Cryptology-Crypto'99, Berlin: Springer-Verlag, 1999, 19-30
11. William W Adams, Philippe Loustaunau. An introduction to gröbner bases[M]. USA: AMS, 1994
12. N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback[A]. Advances in Cryptology-Crypto 2003[C], Berlin: Springer-Verlag, 2003, 176-194
13. F. Armknecht. Improving fast algebraic attacks[A]. Fast Software Encryption2004[C], Berlin: Springer-Verlag, 2004, 65-82
14. C. Carlet, D. K. Dalai, K. C. Gupta, *et al.* Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction[J]. IEEE Transactions on Information Theory, 2006, 52(7): 3105-3121
15. D. K. Dalai, K. C. Gupta and S. Maitra. Cryptographically Significant Boolean functions: Construction and Analysis in terms of Algebraic Immunity[A]. Fast Software Encryption 2005 (FSE05) [C], Paris, France, 2005, 98-111
16. A. Braeken and B. Preneel. On the algebraic immunity of symmetric Boolean functions[A]. Progress in Cryptology-Indocrypt 2005[C], Berlin: Springer-Verlag, 2005, 35-48

17. D. K. Dalai, S. Maitra and S. Sarkar. Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity[J]. Design, Codes and Cryptography, 2006, 40(1): 41-58
18. C. Carlet. A method of construction of balanced functions with optimum algebraic immunity[EB/OL]. http://eprint.iacr.org/2006/149
19. C. Carlet, X. Zeng, C. Li, *et al.* Further properties of several classes of Boolean functions with optimum algebraic immunity[EB/OL]. http://eprint.iacr.org/2007/370
20. F. Armknecht, C. Carlet, P. Gaborit, *et al.* Efficient Computation of Algebraic Immunity for Algebraic and Fast Algebraic Attacks[A]. Advances in Cryptology-Eurocrypt 2006[C], Berlin: Springer-Verlag, 2006, 147-164
21. N. Li and W. Qi. Construction and analysis of Boolean functions of $2t+1$ variables with maximum algebraic immunity[A]. Advances in Cryptology-Asiacrypt 2006[C], Berlin: Springer-Verlag, 2006, 84-98
22. N. Li and W. Qi. Boolean function of an odd number of variables with maximum algebraic immunity[J]. Science in China, Ser. F, 2007, 50(3): 307-317
23. N. Li, L. Qu, W. Qi, *et al.* On the construction of Boolean functions with optimal algebraic immunity[J]. IEEE Transactions on Information Theory, 2008, 54(3): 1330-1334
24. C. Carlet and K. Feng. An Infinite Class of Balanced Functions with Optimal Algebraic Immunity, Good Immunity to Fast Algebraic Attacks and Good Nonlinearity[A]. Advances in Cryptology-Asiacrypt 2008[C], Berlin: Springer-Verlag, 2008, 425-440
25. D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions[A]. Progress in Cryptology-Indocrypt 2004[C], Berlin: Springer-Verlag, 2004, 92-106
26. L. Qu, C. Li, and K. Feng. A note on symmetric Boolean functions with maximum algebraic immunity on odd number of variables[J]. IEEE Transactions on Information Theory, 2007, 53(8): 2908-2910
27. L. Qu and C. Li. On the 2m-variable symmetric Boolean functions with maximum algebraic immunity[J]. Science in China, Ser. F, 2008, 51(2): 120-127
28. F. Liu and K. Feng. On the 2^m-variable symmetric Boolean functions with maximum algebraic immunity 2^{m-1} [J]. To be published in Designs, Codes and Cryptography
29. L. Qu and C. Li. Weight support technique and the symmetric Boolean functions with maximum algebraic immunity on even number of variables[A]. Procedding of Information Security and Cryptology 2007[C], Berlin: Springer-Verlag, 2007, 270-281
30. A. Canteaut and M. Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5[A]. Advances in Cryptology-Eurocrypt 2000[C]. Berlin: Springer-Verlag, 2000, 573-588
31. C. Ding, G. Xiao, and W. Shan. The Stability Theory of Stream Ciphers[M]. Lecture Notes in Computer Science (vol.561). Berlin: Springer-Verlag, 1991
32. M. Lobanov. Tight bound between nonlinearity and algebraic immunity[EB/OL]. http://eprint.iacr.org/2005/441