On Existence and Invariant of Algebraic Attacks

Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario N2L 3G1, CANADA
Email. ggong@calliope.uwaterloo.ca

Abstract. In this paper, we investigate the existence and invariant of algebraic attacks, which have been recently shown as an important cryptanalysis method for symmetric-key cryptographical systems. For a given boolean function f in n variables and two positive integers d and e, we observe that the sufficient condition $d + e \geq n$, shown in [8] or [9], cannot guarantee the existence of a function g with $deg(g) \leq d$ such that $deg(fg) \leq e$ where $fg \neq 0$. Based on this observation, we find a sufficient and necessary condition for the existence of such a multiplier g, which also yields an algorithm to construct them. The algorithm is more efficient when the polynomial basis is employed for linearization than the boolean basis is employed. We then introduce the concept of *invariants* of algebraic attacks in terms of the algebraic security criterion, proposed by Courtois and Meier in 2003, and characterize these invariants. Applying this criterion to the hyper-bent functions, we derive that for a randomly selected boolean function g, the probability of the degree of fg is greater than or equal to $deg(f) = n/2$ is close to 1 where f is a given hyper-bent function in n variables. The tool for establishing our assertions in this paper is to use the (discrete) Fourier transform of boolean functions in terms of technics of analysis of pseudo-random sequences.
Key words. Algebraic attacks, low degree approximation, linearization, (discrete) Fourier transform, invariant, and hyper-bent function.

1 Introduction

Algebraic attacks have been shown as an important cryptanalysis method for symmetric-key cryptographical systems in recent work by Shamir, Patrin, Courtois and Klimov [20], Courtois and Meier [8], and Courtois [7] [9]. Especially, it is significantly effective to attack stream cipher systems in which key streams are generated by filtering function generators, i.e., applying a boolean function on a subset of the tap positions of an linear feedback shift register (LFSR) and the output of the boolean function is a key stream. In general, if one of the following conditions is satisfied, then the cipher system could be broken by applying linearization to find the initial state of the system, which is served as a seed for key stream generation: (a) a boolean function f employed in a stream cipher system has a low degree or it is probabilistically close to a low degree polynomial; (b) there exists some specific function g such that the product $fg = 0$ (or

$(f+1)g = 0$) or $fg \neq 0$ has a low degree or fg is probabilistically close to a zero polynomial or a low degree polynomial.

In order to make algebraic attacks effective, if f is not of low degree, one has to find a boolean function g, as a multiplier, such that either $fg = 0$ (or $(f+1)g = 0$) or $fg \neq 0$ which has a low degree or the probability that fg has a low degree is close to 1. For a given boolean function f in n variables, and two positive integers d and e, Courtois and Meier investigated how to construct a boolean function g with degree at most d such that the degree of fg is at most e. They presented a sufficient condition for existence of such a g in [8], which is $d = \lceil \frac{n}{2} \rceil$ and $e = \lceil \frac{n+1}{2} \rceil$ where $\lceil x \rceil$ denotes the smallest integer that is greater or equal to x , and in [9] which is $d + e \geq n$, respectively. This year, in [17], Meier, Pasalic and Carlet discussed the case for finding g such that $fg = 0$, and presented an algorithm to construct such g if it exists.

For design of boolean functions with applications in stream ciphers and block ciphers, the existing work in the literature is to design boolean functions with resistant to various types of correlation attacks. The popular design criteria for achieving this goal are as follows:

1. Large degree.
2. High nonlinearity: measuring how far from all affine functions.
3. Good correlation immunity and resiliency (Siegenthaler, 1984): k-order correlation immunity/resiliency is to measure whether a boolean function f in n variables is independent of all variables in any k-subset when they are considered as random variables over the binary field $GF(2) = \{0, 1\}$. If so, it is said the boolean function has *k-order correlation immunity* or *k-order resiliency property* if f is balanced. These concepts also correspond to *kth-order invariants* of boolean functions studied by Golomb as early as 1959 [13].
4. Strict avalanche criterion (SAC) [10] and the propagation property [23], more general, low additive autocorrelation, i.e., considering the difference between an output of a boolean function and the resulting output by changing some inputs of the boolean function.

In [8], Courtois and Meier proposed a new criterion for design of boolean functions with applications in symmetric-key cryptographical systems. We recite this criterion with a slight modification.

> *Algebraic Security Criterion*: Assume that f is not of low degree, and for any boolean function g, if $fg \neq 0$, then the degree of the product fg should be greater than or equal to the degree of f.

In this case, we say that f is *invariant of algebraic attacks* (we will formally definite that later). In the literature, there are many constructions for boolean functions which satisfy one or more criteria 1-4 listed above, but not all, since there are some trade-offs among these criteria. The well-known trade-off was given by Siegenthaler appeared in his paper [21] (1984) for introducing correlation immunity/resiliency, i.e., for a boolean function f in n variables, the sum

of its degree and the order of its correlation immunity/resiliency is less than or equal to n or $n-1$. Thus, one can not obtain a boolean function with degree much larger than $n/2$ in order to achieve a certain order of correlation immunity/resiliency. After Siegenthaler's work, there exist a great number of papers in the literature along these lines for example, [22] [10] [19] [2], [3] [15], just listed a few. Some trade-offs or bounds between nonlinearity and correlation immunity/resiliency of boolean functions have been discussed in recent years by a number of researchers [18] [24] [4]. The nonlinearity and propagation property are related by the well-known convolution law which was wisely used in [23].

In this paper, we investigate the existence and invariant of algebraic attacks for the case of finding g such that $fg \neq 0$ which has low degree. We call such g a *low degree approximation* of f. It is worth to point out that such multipliers are necessary in order to launch the fast algebraic attacks proposed in [9]. For a given boolean function f in n variables and two positive integers d and e, we observed that the sufficient condition $d+e \geq n$, shown in [8] or [9], cannot guarantee the existence of a function g with $deg(g) \leq d$ such that $deg(fg) \leq e$ where $fg \neq 0$. Pursuing this approach, we find a sufficient and necessary condition for the existence of such a multiplier g. From this sufficient and necessary condition, we obtain an algorithm to construct these multipliers. We then introduce the concept of *invariant* of algebraic attacks in terms of the algebraic security criterion (see above). Hyper-bent functions are introduced in [25], and satisfy the design criteria 2 and 4. We show for a given hyper-bent function, by randomly selecting a boolean function g, the probability of the degree of fg is greater than or equal to $n/2$ is close to 1. The tool for establishing our assertions in this paper is to use the (discrete) Fourier transform of boolean functions in terms of the methods frequently used in design and analysis of pseudo-random sequences.

This paper is organized as follows. In Section 2, we introduce the (discrete) Fourier transform of boolean functions through their polynomial representations, and bases of the linear space as linearization of nonlinear boolean functions. In Section 3, we first present a sufficient condition for the existence of low degree approximations, then a sufficient and necessary condition, and last, an algorithm to construct such low degree approximations. In Section 4, we introduce the concept of invariants of algebraic attacks. We show that any hyper bent function is invariant of algebraic attacks with probability 1 in Section 5. Section 6 is for conclusions and some discussions on trace-offs between polynomial representations and boolean representations.

2 Discrete Fourier Transform of Boolean Functions and Linearization

We denote $\mathbb{F}_2 = GF(2)$ and $\mathbb{F}_2^m = \{(x_0, x_1, \cdots, x_{m-1}) \,|\, x_i \in \mathbb{F}_2\}$ where m is a positive integer. A *boolean function* f in n variables is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. An algebraic normal form of f is given by

$$f(x_0, \cdots, x_{n-1}) = \sum a_{i_1,\cdots,i_t} x_{i_1} \cdots x_{i_t}, a_{i_1,\cdots,i_t} \in \mathbb{F}_2$$

where the sum runs through all the t-subset $\{i_1, \cdots, i_t\} \subset \{0, 1, \cdots, n-1\}$. The *degree* of the boolean function f is the largest t for which $a_{i_1,\cdots,i_t} \neq 0$.

In this paper, the notation $H(s)$ represents the Hamming weight of s, i.e., the number of nonzeros in s, where s could be a positive integer represented in a binary form, or s a k-dimensional binary vector, or s a boolean function in n variables represented as a binary vector of $(f(\mathbf{x}_0), f(\mathbf{x}_1), \cdots, f(\mathbf{x}_{2^n-1}))$ where $\mathbf{x}_i, i = 0, \cdots, 2^n - 1$ constitutes all elements in $\mathbb{F}_2^n$.

Note that $\mathbb{F}_2^n$ is isomorphic to the finite field $GF(2^n)$, regarded as a linear space over $GF(2)$ of dimension n. For simplicity, we denote $GF(2^n)$ by $\mathbb{F}_{2^n}$. Any boolean function can be represented by a polynomial function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$. For a polynomial function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, say $f(x) = \sum_{i=0}^{2^n-1} d_i x^i, d_i \in \mathbb{F}_{2^n}$, the *algebraic degree* of f is given by the largest t such that $d_i \neq 0$ for which the Hamming weight of the binary representation of i is equal to t, i.e., $H(i) = t$. In this paper, the degree of a function f from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$ always means the degree of its boolean form or equivalently, the algebraic degree of its polynomial form, denoted by $deg(f)$.

2.1 Trace Representation of Boolean Functions, Polynomial Functions, and Sequences

Using the Lagrange interpolation, we may define the (discrete) Fourier transform of boolean functions through their polynomial forms. Let f be a boolean function in n variables in a polynomial form. The *(discrete) Fourier Transform (DFT)* of f is defined as

$$A_k = \sum_{x \in \mathbb{F}_{2^n}^*} [f(x) + f(0)] x^{-k}, \ k = 0, 1, \ldots, 2^n - 1. \tag{1}$$

The inverse formula of (1) is given as follows:

$$f(x) = \sum_{k=0}^{2^n-1} A_k x^k, \quad x \in \mathbb{F}_{2^n}^*. \tag{2}$$

Fact 1 *$A_k \in \mathbb{F}_{2^n}$, and $A_{2^i k} = A_k^{2^i}, i = 0, 1, \cdots, n-1$.*

In the following, we show the trace representation of boolean functions in terms of their Fourier transforms. For doing so, we need the concept of cyclotomic cosets. A *(cyclotomic) coset* C_s modulo $2^n - 1$ is defined by

$$C_s = \{s, s \cdot 2, \cdots, s2^{n_s-1}\},$$

where n_s is the smallest positive integer such that $s \equiv s2^{n_s} \pmod{2^n - 1}$. The subscript s is chosen as the smallest integer in C_s, and s is called the *coset leader* of C_s. For example, for $n = 4$, the cyclotomic cosets modulo 15 are:

$$C_0 = \{0\}, C_1 = \{1, 2, 4, 8\}, C_3 = \{3, 6, 12, 9\}, C_5 = \{5, 10\}, C_7 = \{7, 14, 13, 11\}$$

where $\{0, 1, 3, 5, 7\}$ are coset leaders modulo 15.

We may group monomial terms according to Fact 1, which results in the following representation of a boolean function by the sum of trace terms, whose proof is omitted here.

Proposition 1. (Trace Representation of Functions) *Any non-zero function $f(x)$ from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$ can be represented as*

$$f(x) = \sum_{k \in \Gamma(n)} Tr_1^{n_k}(A_k x^k) + A_{2^n-1} x^{2^n-1}, A_k \in \mathbb{F}_{2^{n_k}}, A_{2^n-1} \in \mathbb{F}_2, x \in \mathbb{F}_{2^n} \quad (3)$$

where $\Gamma(n)$ is the set consisting of all coset leaders modulo $2^n - 1$, $n_k \,|\, n$ the size of the coset C_k, and $Tr_1^{n_k}(x)$ the trace function from $\mathbb{F}_{2^{n_k}}$ to $\mathbb{F}_2$.

Recall that $H(k)$ denotes the Hamming weight of a positive integer k. If f is a boolean function with degree $deg(f) = r < n$, from Proposition 1, the trace representation of f can be given by

$$f(x) = \sum_{H(k) \le r} Tr_1^{n_k}(A_k x^k), \quad (4)$$

where $A_k \neq 0$ for at least one $k \in \Gamma(n)$ such that $H(k) = r$.

Next we investigate the relationship of the Fourier transforms between a boolean function and its associated binary sequence. Let α be a primitive element in $\mathbb{F}_{2^n}$. We assume that $f(0) = 0$ (if $f(0) \neq 0$, we replace f by $g = f - f(0)$ where $g(0) = 0$). We associate f with a binary sequence $\{a_t\}$ whose elements are given by

$$a_t = f(\alpha^t), t = 0, 1, \cdots, 2^n - 2. \quad (5)$$

Then the period of $\{a_t\}$, say N, is a factor of $2^n - 1$. Usually, the Fourier transform of $\{a_t\}$ is defined as $B_k = \sum_{t=0}^{N-1} a_t \beta^{-tk}, 0 \le k < N$ where β is an element in $\mathbb{F}_{2^n}$ with order N, and its inverse formula is given by $a_t = \sum_{k=0}^{N-1} B_k \beta^{kt}, 0 \le t < N$. If we choose $\beta = \alpha^v$ where $v = (2^n - 1)/N$, then we have

$$B_k = A_k, 0 \le k < N, \text{ and } A_{iN+j} = A_j, 0 \le i < v, 0 \le j < N.$$

Thus (3) is the trace representation of both f and the associated sequence $\{a_t\}$. $\{A_k\}$ is also called a *Fourier spectral sequence* of f or equivalently $\{a_t\}$. We list some frequently used results on Fourier spectra into the following fact.

Fact 2 *With the above notation,*

(a) $\{A_k\}$ *is of period* N.

(b) $A_k = 0$ *if* $k \not\equiv 0 \pmod v$, *and* $A_{jv} = \sum_{t=0}^{N-1} a_t \alpha^{-tvj}, 0 \le j < N$ *where there is at least* $j : 0 < j < N$ *such that* $A_{jv} \neq 0$ *if* $\{a_t\}$ *is not a constant sequence.*

(c) Let $a(x) = \sum_{t=0}^{2^n-2} a_t x^t$. *Then* $A_k = a(\alpha^{-k}), 0 \le k < 2^n - 2$.

Therefore, any function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, or equivalently a boolean function in n variables, corresponds a binary sequence with period $N \mid 2^n - 1$. They have the same Fourier spectral sequence. This leads to an efficient linearized method to find low degree approximations, which will be shown below.

For simplicity, sometimes, we denote $q = 2^n$ and $\mathcal{F}$ the set consisting of all functions from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, or equivalently, all boolean functions in n variables.

2.2 Bases of $\mathcal{F}$

Note that $\mathcal{F}$ can be considered as a linear space of dimension q over $\mathbb{F}_2$ when each function in $\mathcal{F}$ is represented by a binary vector of dimension q. For $f \in \mathcal{F}$, there are two common ways to represent f as a binary vector of dimension q.

Method 1. f is of the boolean representation. We list the elements in $\mathbb{F}_2^n$ in the same order as the truth table of f. Thus,

$$\begin{aligned} f(x_0, x_1, \cdots, x_{n-1}) = (f(\mathbf{t}_0), f(\mathbf{t}_1), \cdots, f(\mathbf{t}_{q-1})), \\ \mathbf{t}_i = (t_{i,0}, t_{i,1}, \cdots, t_{i,n-1}),\ t_{ij} \in \mathbb{F}_2, \quad \text{with} \\ i = t_{i,0} + t_{i,1}2 + \cdots + t_{i,n-1}2^{n-1}, 0 \le i < q. \end{aligned}$$

For $\mathbf{x} = (x_0, \cdots, x_{n-1})$ and $\mathbf{c} = (c_0, \cdots, c_{n-1})$ in $\mathbb{F}_2^n$, we denote $\mathbf{x}^{\mathbf{c}} = x_0^{c_0} x_1^{c_1} \cdots x_{n-1}^{c_{n-1}}$. Then, the basis of $\mathcal{F}$, regarded as a linear space over $\mathbb{F}_2$, consists of all monomial terms:

$$\Delta = \{\mathbf{x}^{\mathbf{c}} \mid \mathbf{c} \in \mathbb{F}_2^n\}.$$

This basis is referred as a *boolean basis* of $\mathcal{F}$.

Method 2. f is of the polynomial form. We use the cyclic group of $\mathbb{F}_q$, i.e.,

$$f(x) = (f(0), f(1), f(\alpha), \cdots, f(\alpha^{q-2})). \tag{6}$$

Let

$$\Pi_k = \{Tr_1^{n_k}(\beta(\alpha^i x)^k) \mid, i = 0, 1, \cdots, n_k - 1\}, \beta \in \mathbb{F}_{2^{n_k}}, \tag{7}$$

where n_k is the size of the coset consisting k. Note that $\{\alpha^{ik} \mid i = 0, 1, \cdots, n_k - 1\}$ is a basis of $\mathbb{F}_{2^{n_k}}$ over $\mathbb{F}_2$. Together with the trace representation of a function in $\mathcal{F}$ given by Proposition 1, we obtain that all monomial trace terms in Π_k when k runs through all coset leaders modulo $q-1$ constitute a basis of $\mathcal{F}$. In other words, the following set is a basis of $\mathcal{F}$:

$$\Pi = \cup_{k \in \Gamma(n)} \Pi_k \tag{8}$$

where $\Gamma(n)$ is the set consisting of all coset leaders modulo $q-1$, which is referred to as a *polynomial basis* of $\mathcal{F}$.

2.3 Efficient Computation of the Polynomial Basis of $\mathcal{F}$

Let $\mathbf{a} = \{a_i\} = a_0, a_1, \cdots$ be a binary sequence with period N. Let L be the (left) shift operator of a sequence $\mathbf{a}$, i.e., $L\mathbf{a} = a_1, a_2, \cdots$, $L^i\mathbf{a} = a_i, a_{i+1}, \cdots$. A

k-decimation of $\mathbf{a}$ is the sequence $\mathbf{a}^{(k)} = \{a_{ki}\}$, where the indices are reduced by modulo N. In the following, we assume that $a_i = Tr_1^n(\beta\alpha^i)$ where $\beta \in \mathbb{F}_{2^n}^*$. If $\mathbf{a}^{(k)} = \mathbf{0}$, then we may choose r such that the k-decimation of $L^i\mathbf{a}$ is a zero sequence for $i = 0, 1, \cdots, r-1$, and the k-decimation of $L^r\mathbf{a}$ is not a zero sequence. For simplicity, we still denote such a decimation as $\mathbf{a}^{(k)}$. Therefore, the elements of $\mathbf{a}^{(k)}$ is given by $a_{ki} = Tr_1^{n_k}(\beta\alpha^{ki}), i = 0, 1, \cdots$, where n_k is the size of the coset C_k. We denote

$$P_k = \begin{bmatrix} 0, & \mathbf{a}^{(k)} \\ 0, & L\mathbf{a}^{(k)} \\ \vdots & \\ 0, & L^{n_k-1}\mathbf{a}^{(k)} \end{bmatrix} \tag{9}$$

where $L^i\mathbf{a}^{(k)}$'s are regarded as binary vectors of dimension $q-1$, and each row corresponds to a function in Π_k. Thus, for each k, a coset leader modulo $q-1$, we only need to compute $\mathbf{a}^{(k)}$, the rest of rows in P_k can be obtained by the shift operator which has no cost. Furthermore, $|\Gamma(n)|$, the number of the coset leaders modulo $q-1$, is equal to the number of the irreducible polynomials over $\mathbb{F}_2$ of degrees dividing n. Thus, for computing the polynomial basis of $\mathcal{F}$, one only need to compute $|\Gamma(n)|$ decimation sequences from $\mathbf{a}$, approximately, q/n decimation sequences from $\mathbf{a}$, in order to get the polynomial basis of $\mathcal{F}$. Compared it with the boolean basis, we have to compute $q = 2^n$ vectors of dimension q since there is no computational saving for these evaluations.

3 Characteristic of Low Degree Approximations

Assume that a boolean function f employed in a cipher system is not low degree.

Definition 1. *Let f be a boolean function in n variables. If there exists some specific function g such that the product $fg \neq 0$ has a low degree in the sense that $deg(fg) < def(f)$ or fg is probabilistically close to a non-zero low degree polynomial in the sense that $Prob\{fg = h\} = 1$ where $deg(h) < deg(f)$, then g is called a* low degree approximation *of f or a* probabilistically low degree approximation.

In [8] [9], Courtois and Meier showed that for given positive integers d and e with $d + e \geq n$ there exists some g with degree at most d such that fg has the degree at most e. In this section, we will show how to characterize the case that $fg \neq 0$ for given d and e. We first present a sufficient condition for the existence of low degree approximations. Following this approach, we establish a sufficient and necessary condition for the existence of low degree approximations, which also yields an algorithm to construct them.

Let S_d be the set consisting of all those functions in the polynomial basis with $H(k) \leq d$, or equivalently, all boolean monomial terms of degrees less than or equal to d. Then

$$|S_d| = \sum_{i=0}^{d} \binom{n}{i}.$$

Notice that any function in $\mathcal{F}$ of degree d is a linear combination of functions in S_d over $\mathbb{F}_2$. For a given $f \in \mathcal{F}$, we denote

$$fS_d = \{f(x) \cdot g(x) | g \in S_d\}.$$

Note that $f(x) \cdot g(x)$ is a dot product when both f and g are represented as binary vectors of dimension $q = 2^n$.

Theorem 1. *With the above notation, for a given $f \in \mathcal{F}$ and two positive integers d and e with $1 \leq d, e < n$, if fS_d contains at least*

$$2^n + 1 - |S_e|$$

different functions and all those functions are linear independent over $\mathbb{F}_2$, then there exists a function $g \in \mathcal{F}$ with degree at most d such that $deg(fg) \leq e$.

Proof. Let $r = |fS_d|$ and $s = |S_e|$. Since $r + s > 2^n$ and the dimension of $\mathcal{F}$ is 2^n, the vectors in $fS_d \cup S_e$ are linear dependent over $\mathbb{F}_2$. We list the elements in fS_d as $fg_1, \cdots, fg_r$, and the elements in S_e as $h_1, \cdots, h_s$. Then there exist $c_i \in \mathbb{F}_2, i = 1, \cdots, r + s$, which are not all zeros, such that

$$\sum_{i=1}^{r} c_i fg_i + \sum_{i=1}^{s} c_{r+i} h_i = 0. \tag{10}$$

Note S_e is linear independent over $\mathbb{F}_2$. From the assumption, the elements in fS_d are linear independent over $\mathbb{F}_2$. Thus, there are some $j : 1 \leq j \leq r$ and $i > r$ such that both $c_j \neq 0$ and $c_i \neq 0$. We write $g = \sum_{i=1}^{r} c_i g_i$ and $h = \sum_{i=1}^{s} c_{r+i} h_i \neq 0$. Hence $fg = h$ where $deg(g) \leq d$ and $deg(h) \leq e$.

□

Remark 1. The condition in Theorem 1 implies that $d + e \geq n$. Otherwise $|S_d| + |S_e| \leq 2^n \Longrightarrow |fS_d| + |S_e| \leq 2^n$, which is a contradiction.

Note that it is possible that $|fS_d| < |S_d|$ and the elements in fS_d are linear dependent over $\mathbb{F}_2$. In this case, possibly, there is no function g with $deg(g) \leq d$ such that $fg \neq 0$ and $deg(fg) \leq e$.

Example 1. Let $f(x) = Tr_1^3(\alpha^5 x + \alpha^6 x^3)$ be a function from $\mathbb{F}_{2^3}$ to $\mathbb{F}_2$ where α is a primitive element in $\mathbb{F}_{2^3}$ with $\alpha^3 + \alpha + 1 = 0$. Let $d = 2$ and $e = 1$, then $d + e = n$. The set S_2 contains the following seven functions

$$\begin{aligned}
\text{constant function c} &= 11111111 \\
Tr_1^3(x) &= 01001011 \\
Tr_1^3(\alpha x) &= 00010111 \\
Tr_1^3(\alpha^2 x) &= 00101110 \\
Tr_1^3(x^3) &= 01110100 \\
Tr_1^3((\alpha x)^3) &= 01101001 \\
Tr_1^3((\alpha^2 x)^3) &= 01010011
\end{aligned}$$

On the other hand, we have

$$f(x) = 00100001.$$

Thus the elements of fS_2 are

$$\begin{aligned} fTr_1^3(x) = fTr_1^3(\alpha x) = fTr_1^3((\alpha^2 x)^3) &= 00000001 \\ fTr_1^3(\alpha^2 x) = fTr_1^3(x^3) &= 00100000 \\ fc = fTr_1^3((\alpha x)^3) &= 00100001 \end{aligned}$$

Thus $|fS_2| = 3$. However, $00100001 = 00000001 + 00100000$. Thus the elements of fS_2 are linear dependent. The maximal linear independent set in fS_2 is given by $D = \{00000001, 00100001\}$. It is easy to see that $D \cup S_1$ is a linear independent set. So there is no function g with degree 2 such that $deg(fg) = 1$ with $fg \neq 0$. This shows that the condition $d + e \geq n$ in Theorem 6.0.1 given in [8] and Theorem 7.2.1 in [9] (also recited in [17]) cannot guarantee that the existence of some function $g \neq 0$ with $deg(g) \leq d$ such that $fg \neq 0$ with $deg(fg) \leq e$, since there exists a degenerated case for which $|fS_d| < |S_d|$.

Note that the boolean form of f is given by $f(x_0, x_1, x_2) = x_0x_1 + x_0x_2 + x_1x_2 + x_1$. We may use the boolean form of f to show the above result using a similar technics as above.

From the proof of Theorem 1, it is not necessary to require that $r + s > 2^n$. In the following, we characterize the existence of a low degree approximation.

Theorem 2. *(*Existence of Low Degree Approximations*) With the above notation, for a given $f \in \mathcal{F}$ and $1 \leq d, e < n$, let D be a maximal linear independent set of fS_d. Then there exists a function $g \in \mathcal{F}$ with degree at most d such that $deg(fg) \leq e$ if and only if $D \cup S_e$ is linear dependent over $\mathbb{F}_2$.*

Proof. A proof for the sufficient condition is similar as we did for Theorem 1. We denote $|D| = t$ and $|S_e| = s$. Since $D \cup S_e$ is linear dependent over $\mathbb{F}_2$, there exist $c_i \in \mathbb{F}_2, i = 1, \cdots, t + s$ such that

$$\sum_{i=1}^{r} c_i fg_i + \sum_{i=1}^{s} c_{t+i} h_i = 0$$

where $fg_i \in D$ and $h_i \in S_e$. Since both D and S_e are linear independent over $\mathbb{F}_2$, so there exist some i with $1 \leq i \leq t$ and j with $1 \leq j \leq s$ such that $c_i \neq 0$ and $c_{t+j} \neq 0$, respectively. Thus $g = \sum_{i=1}^{t} c_i g_i \neq 0$ with $deg(g) \leq d$, and $fg = h = \sum_{j=1}^{s} c_{t+j} h_j \neq 0$ with $deg(h) \leq e$, which establishes the assertion.

Conversely, assume that there exists some $g \in \mathcal{F}$ with $deg(g) \leq d$ such that $fg \neq 0$ and $deg(fg) \leq e$. Since any function with degree less than or equal to d is a linear combination of functions in S_d, we may write $g = \sum_{i=1}^{l} d_i g_i$ where $d_i \in \mathbb{F}_2$ and $l = |S_d|$. For simplicity, we may assume that $g_i, i = 1, \cdots, t$ are functions in S_d such that $\{fg_i \,|\, i = 1, \cdots, t\}$ is a maximal linear independent set of fS_d. Thus we have

$$fg = f \cdot \sum_{i=1}^{l} d_i g_i = \sum_{i=1}^{l} d_i (fg_i) = \sum_{i=1}^{t} e_i (fg_i) \neq 0, e_i \in \mathbb{F}_2.$$

Consequently, there exists some i with $1 \leq i \leq t$ such that $e_i \neq 0$. On the other hand, since $h = fg$ with degree $deg(h) \leq e$, then $h \in S_e$. Therefore, h is a linear combination of the functions in S_e, say $h = \sum_{i=1}^{s} k_i h_i \neq 0, k_i \in \mathbb{F}_2$. This shows that there is some $k_i \neq 0$ with $1 \leq i \leq s$. Therefore, we have

$$h = fg \Longleftrightarrow \sum_{i=1}^{t} e_i f g_i + \sum_{i=1}^{e} k_i h_i = 0$$

where e_i's and k_i's are not all zero. Thus, $D \cup S_e$ is linear dependent over $\mathbb{F}_2$, which completes the proof.

□

From the proof of Theorem 2, we can easily establish the solution to the case where $fg = 0$, which we present it in the following corollary.

Corollary 1. *For a given f and $0 < d < n$, there exists some $g \neq 0$ with $deg(g) \leq d$ such that $fg = 0$ if and only if $|D| < |S_d|$.*

In the following, using Theorem 2, we provide an algorithm to determine whether there exists a function g with $deg(g) \leq d$ such that $fg \neq 0$ with $deg(fg) \leq e$ for a given f and two integers d and e with $1 \leq d, e < n$. If there exists such a multiplier g, the algorithm outputs g and fg. Otherwise, it outputs $g = 0$ and $fg = 0$. For simplicity, in the algorithm, if it is needed, a subset of $\mathcal{F}$ is also regarded as a matrix in which each row is a function in the set, represented as a 2^n-dimensional binary vector.

Algorithm 1 An Algorithm for Finding a Low Degree Approximation

Input: $f \in \mathcal{F}$, *a function from* $\mathbb{F}_{2^n}$ *to* $\mathbb{F}_2$;
$1 \leq d, e < n$; *and*
$t(x) = x^n + \sum_{i=0}^{n-1} t_i x^i, t_i \in \mathbb{F}_2$, *a primitive polynomial over* $\mathbb{F}_2$ *of degree* n.

Output: $g \in \mathcal{F}$ *with* $deg(g) \leq d$ *and* $h = fg$ *with* $h \neq 0$ *and* $deg(h) \leq e$ *if there exist such* g *and* h. *Otherwise, outputs* $g = 0$ *and* $h = 0$.

Procedure

1. *Randomly select an initial state* $(a_0, a_1, \cdots, a_{n-1}), a_i \in \mathbb{F}_2$, *and compute*

$$a_{n+i} = \sum_{j=0}^{n-1} t_j a_{j+i}, i = 0, 1, \cdots, 2^n - 1 - n.$$

(Note $\mathbf{a} = (a_0, \cdots, a_{2^n-2})$ *is a binary m-sequence of degree n [13].)*

2. *Compute k, each coset leader modulo $2^n - 1$, and n_k, the size of C_k. Set $I = \{(k, n_k) | k \in \Gamma n\}$ ($\Gamma(n)$ consists of all coset leaders modulo $2^n - 1$).*
3. *Set $m = max\{d, e\}$. Pack S_m as follows:*
   $P_0 = (1, 1, \cdots, 1)$;
   for $0 \neq k$ *in* $\Gamma(n)$ *with* $H(k) \leq m$ **do**
   Compute $\mathbf{a}^{(k)} = (a_0, a_k, \cdots, a_{k(2^n-2)})$, a k-decimation of $\mathbf{a}$, then apply the shift operator to the decimated sequence.
   Pack P_k, defined by (9), as a n_k by 2^n sub-matrix of S_m for all k with $0 \leq H(k) \leq m$.
4. *Using the Gauss elimination, find the rank of fS_d, represented in a $|fS_d|$ by 2^n matrix, and a maximal linear independent set of fS_d, say D.*
5. *Apply the Gauss elimination to the following matrix whose entries are given by*
$$\begin{bmatrix} D \\ S_e \end{bmatrix}.$$
   If the rank of the above matrix is equal to $t + s$ where $|D| = t$ and $|S_e| = s$, set $g = 0$ and $h = 0$, then go to Step 6. Otherwise, find $c_i, i = 1, \cdots, t$ such that
$$\sum_{i=1}^{t} c_i f g_i + \sum_{i=1}^{s} c_{t+i} h_i = 0.$$
   Set $g = \sum_{i=1}^{t} c_i g_i$ and $h = \sum_{i=1}^{s} c_{t+i} h_i$.
6. *Return g and h.*

Remark 2. The algorithm can also be applied to f of a boolean representation. However, packing S_m for the polynomial basis is more efficient than packing S_m using the boolean function basis:

$$1, x_1, \cdots, x_n, x_1x_2, \cdots, \mathbf{x}^{\mathbf{c}}, \cdots, x_{i_1}x_{i_2}\cdots x_{i_m}, \forall \mathbf{c} \in \mathbb{F}_2^n \text{ with } H(\mathbf{c}) \leq m,$$

due to Step 3 which needs only the shift operation for a group of $n_k - 1$ rows of S_m for each $k \in \Gamma(n)$ with $H(k) \leq m$.

Remark 3. The Algorithm can be applied to find g such that $fg = 0$ where Step 5 is replaced by Step 5': If $|D| = |S_d|$, then return $g = 0$ which represents no annihilators with degree at most d. Otherwise, finding c_i such that $\sum_{i=1}^{t} c_i f g_i = 0$, set $g = \sum_{i=1}^{t} c_i g_i$, and return g.

When Algorithm 1 is restricted to compute annihilators of f, the computational complexity of this algorithm is approximately the cost to execute the Gaussian elimination algorithm to a $|S_d|$ by 2^n matrix.

Remark 4. In [17], Meier, Pasalic and Carlet proposed the following algorithm (Algorithm 1 in [17]) for finding g such that $fg = 0$ where $deg(g) \leq d$ for given f and d. Let $S_d = \{g_i\}$. Then any boolean function of degree at most d can be represented by $g = \sum_i c_i g_i, c_i \in \mathbb{F}_2$. Form a system of the equations:

$$g(x) = 0 \text{ for } x \in \mathbb{F}_{2^n} \text{ such that } f(x) = 1 \tag{11}$$

which is a system of linear equations in unknowns c_i's. If (11) has solutions, then there exist such annihilators. The second algorithm in [17] improved the above algorithm by taking account of the Hamming weight structure of $x = (x_0, x_1, \cdots, x_{n-1})$. For both algorithms in [17], the computational cost consists of two parts. One part is the cost to form the equations (11) which cannot be ignored. The other part is the cost to apply the Gaussian elimination algorithm to the $H(f)$ by $|S_d|$ coefficient matrix of (11).

Algorithm 1 in [17] (also Algorithm 2 in [17]) can be extended to compute low degree approximations in the following way. We write $S_e = \{h_i\}$. Then h, a boolean function with degree at most e, can be represented as $h(x) = \sum_i e_i h_i, e_i \in \mathbb{F}_2$. For given f, d and e, to find g such that $fg = h \neq 0$ is to solve the following system of linear equations in unknowns c_i's and e_i's:

$$g(x) = h(x) \quad \text{for } x \in \mathbb{F}_{2^n} \text{ such that } f(x) = 1. \tag{12}$$

Similarly, the computational cost consists of two parts. One part is the cost to form (12), and the other is the cost to execute the Gaussian elimination algorithm to the $H(f)$ by $|S_d| + |S_e|$ coefficient matrix of (12). If one ignores the cost to form the equations (11) or (12), then the algorithm described above are more efficient that Algorithm 1, since $H(f) < 2^n$. However, the algorithm that we proposed here is to easily characterize the invariant property of algebraic attacks, which will be given in the next section.

4 Invariants of Algebraic Attacks

In this section, we define the concept of invariants of algebraic attacks in terms of the algebraic security criterion, introduced by Courtois and Meier in [8] (2003), and characterize these invariants.

Definition 2. *Assume that $f \in \mathcal{F}$ is not of low degree. f is said to be* invariant of algebraic attacks *if $\forall g \in \mathcal{F}$,*

$$deg(fg) \geq deg(f) \text{ with } fg \neq 0 \text{ or } fg = 0/(f+1)g = 0 \text{ with } deg(g) \geq \deg(f). \tag{13}$$

*If (13) is only true for those g with $deg(g) \leq k$, then $f(x)$ is said to be k-*order invariant *of algebraic attacks.*

For convenience, we say that g is an *annihilator* of f if either $fg = 0$ or $(f+1)g = 0$. Thus, if f is invariant of algebraic attacks, then f cannot have any low degree approximation or f cannot have any annihilator with degree less than the degree of f. Thus, it is resistant to algebraic attacks. From Theorem 2, we have the following condition to determine whether a given function is invariant of algebraic attacks.

Theorem 3. *Let $f \in \mathcal{F}$ with degree r, and D_d be a maximal linear independent set of fS_d.*

1. *f is invariant of algebraic attacks if and only if $D_d \cup S_e$ is linear independent over $\mathbb{F}_2$ for all $e : 1 \leq e < r$ and $d : 1 \leq d \leq n$ or $|D_d| = |S_d|$ for all $d : 1 \leq d < r$.*
2. *f is k-order invariant of algebraic attacks if and only if $D_k \cup S_e$ is linear independent over $\mathbb{F}_2$ for all $e : 1 \leq e \leq r-1$ or $|D_d| = |S_d|$ for all $d : 1 \leq d < min\{r, k\}$.*

Proof. Note that f being invariant of algebraic attacks is equivalent to f being n-order invariant of algebraic attacks. So, in the following, we state only for the k-order invariant. Also, it suffices to show that it is necessary.

Case 1. If there exists some e such that $D_k \cup S_e$ is linear dependent over $\mathbb{F}_2$, according to Theorem 2, there is some $g \in \mathcal{F}$ with $deg(g) \leq k$ such that $deg(g) \leq e < r$.

Case 2. If there is some d: $1 \leq d < min\{r, k\}$ such that $|D_d| < |S_d|$. According to Corollary 1, there is an annihilator g of f such that $fg = 0$ with $deg(g) \leq d < r$.

Both cases show that f is not invariant of algebraic attacks. Thus, the conditions are necessary.

□

From Theorem 3, we know that Algorithm 1 can be used in two ways. One is to verify whether a given function is invariant of algebraic attacks. In other words, for a given function of degree r, we may run Algorithm 1 (for both cases) for all pairs of (d, e) where $1 \leq d \leq n$ and $1 \leq e < r$ (here we include the case $d = n$ in Algorithm 1). If the outputs are zero for each pair of (d, e), then f is invariant of algebraic attacks. Note that we only need to pack S_m in Algorithm 1 once. The complexity for verifying whether a given function is invariant of algebraic attacks is approximate to $n(r-1)$ times of the complexity of the Gauss reduction involved in Algorithm 1. The other is for search of such invariants.

Example 2. Let $\mathbb{F}_{2^4}$ be defined by a primitive polynomial $t(x) = x^4 + x + 1$ and α a root of $t(x)$. Let $f(x) = Tr(\alpha x^3)$ where $Tr(x) = x + x^2 + x^4 + x^8$ is the trace function from $\mathbb{F}_{2^4}$ to $\mathbb{F}_2$. Then $f(x)$ is a bent function (see the definition in the next section). From Proposition 2, any function $g(x) \in \mathcal{F}$ with $g(0) = 0$ can be written as

$$g(x) = Tr(bx) + Tr(cx^3) + Tr_1^2(dx^5) + Tr(ex^7) + wx^{15}, b, c, e \in \mathbb{F}_{2^4}, d \in \mathbb{F}_{2^2}, w \in \mathbb{F}_2$$

where $Tr_1^2(x) = x + x^2$ is the trace function from $\mathbb{F}_{2^2}$ to $\mathbb{F}_2$. Multiplying f by each monomial trace term in g, we have

$$\begin{aligned}
Tr(bx)Tr(\alpha x^3) &= Tr(b^4\alpha^4 x) + Tr_1^2((b^2\alpha + b^8\alpha^4)x^5) + Tr((b\alpha^2 + b^4\alpha)x^7) \\
Tr(cx^3)Tr(\alpha x^3) &= Tr((c^2\alpha^4 + c^4\alpha^2 + c^8\alpha^8)x^3) + Tr(c\alpha^4) \\
Tr_1^2(dx^5)Tr(\alpha x^3) &= Tr(d^2\alpha^2 x + d^2\alpha^4 x^7) \\
Tr(ex^7)Tr(\alpha x^3) &= Tr((e^4\alpha^4 + e\alpha^8)x) + Tr_1^2((e^2\alpha^2 + e^8\alpha^8)x^5) + Tr(e^4\alpha^8 x^7).
\end{aligned}$$

In the follows, we consider $w = 0$. It is similar for the case $w = 1$. From the above identities, we have the expansion of $f(x)g(x)$ as follows

$$f(x)g(x) = Tr(Ax + Bx^3 + Dx^7) + Tr_1^2(Cx^5) + E$$

where

$$\begin{aligned} A &= b^4\alpha^4 + d^2\alpha^2 + e^4\alpha + e\alpha^8 \\ B &= c^2\alpha^4 + c^4\alpha^2 + c^8\alpha^8 \\ D &= b\alpha^2 + b^4\alpha + d^2\alpha^4 + e^4\alpha^8 \\ C &= b^2\alpha + b^8\alpha^4 + e^2\alpha^2 + e^8\alpha^8 \\ E &= Tr(c\alpha^4). \end{aligned}$$

Considering that $fg \neq 0$, then $deg(fg) = 1$ if and only if

$$B = C = D = 0.$$

It can be verified that the system of those equations has no solutions for any choices of b, c, d and e. Thus $deg(fg) \geq 2$. Therefore, f is invariant of algebraic attacks. Alternatively, we can directly apply Algorithm 1 to f to obtain this result. Note that the boolean form of $f(x)$ is given by $x_2 + x_0x_1 + x_0x_2 + x_0x_3 + x_1x_2 + x_1x_3 + x_2x_3$. However, in order to verify whether f is invariant of algebraic attacks, using the polynomial form of f is much easier than using the boolean form.

5 Possibility of Invariants of Algebraic Attacks for Hyper-bent Functions

A bent function is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ whose Hadamard transform has constant magnitude, i.e.,

$$\widehat{f}(\lambda) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{Tr(\lambda x) + f(x)} = \pm\sqrt{2^n}, \ \forall \lambda \in \mathbb{F}_{2^n}. \tag{14}$$

In this case, n is even. We write $n = 2m$. A bent function is called a *hyper-bent function* [25] if

$$\hat{f}(\lambda, c) = \pm 2^m, \forall \lambda \in \mathbb{F}_{2^n}, c : 0 < c < 2^n - 1, gcd(c, 2^n - 1) = 1$$

where

$$\hat{f}(\lambda, c) = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{f(x) + Tr(\lambda x^c)}.$$

$\hat{f}(\lambda, c)$ is so called the *extended Hadamard transform* introduced in [14]. There are two general constructions for bent functions. One is the Maiorana-McFarland construction, and the other is the Dillon construction [11]. We introduce the Dillon construction as follows. Let α be a primitive element of $\mathbb{F}_{2^n}$, $d = 2^m + 1$,

$v = 2^m - 1$, $\mathbf{a} = (a_0, a_1, \cdots, a_{d-1})$ not a constant vector in $\mathbb{F}_2^d$. We associate $\mathbf{a}$ with a function $f(x)$ which is defined by

$$f(0) = 0, f(\alpha^{id+j}) = a_j, 0 \leq i < v, 0 \leq j < d. \tag{15}$$

If the Hamming weight of $\mathbf{a}$ is equal to 2^{m-1}, then $f(x)$ is bent. (Note. The construction given above is taken from [25], which used a completely different approach from Dillon's original construction [11].) Carlet had several excellent generalizations of both constructions of bent functions in [5].

Youssef and Gong proved that the bent functions, as shown above, are hyper-bent (see [25]), and determined that they all have degree m, which is maximal among all bent functions. Hyper-bent functions have the largest distance from all affine boolean functions and all bijective monomial trace terms from $\mathbb{F}_q$ to $\mathbb{F}_2$.

Remark 5. Some New Observations for Hyper-bent Functions: Hyper-bent functions can also be defined by the Hadamard transforms of a family of functions. In a detail, let $f_c(x) = f(x^c)$. Then f is hyper-bent if and only if $f_c(x)$ is bent for each $c : 0 < c < 2^n - 1$ with $gcd(c, 2^n - 1) = 1$. Let $\mathcal{DB}$ be the set consisting all bent functions from the Dillon's construction. Note that if $\mathbf{a}$ is the associated sequence of f, then $\mathbf{a}^{(c)}$, the c-decimation of $\mathbf{a}$ (see Section 2. 3 for the definition of decimation of sequences), is associated with $f_c(x)$. Since $gcd(c, 2^n - 1) = 1$, then c is relatively prime to d. Therefore, $H(\mathbf{a}) = H(\mathbf{a}^{(c)})$. Thus $f \in \mathcal{DB}$, then $f_c(x) \in \mathcal{DB}$, i.e., the composition of monomial bijective map with a bent function in $\mathcal{DB}$ is still a bent function in $\mathcal{DB}$. This is a short proof for that any bent function in $\mathcal{DB}$ is hyper-bent.

From now on, when we talk about the hyper-bent functions, we mean that they are constructed by the Dillon construction. In the following proposition, we show the Fourier transform for a general class of functions constructed by (15), which include the hyper-bent functions as a subset.

Proposition 2. *Let $f(x)$ be constructed by (15). Then the Fourier transform of $f(x)$ satisfies*

$$A_k = 0, \forall k \not\equiv 0 \pmod{v}, \quad \textit{and } f(x) = \sum_t Tr_1^{n_t}(A_{tv} x^{tv}) \tag{16}$$

where $n_t \,|\, n$ is the size of the coset containing tv modulo $2^n - 1$, and there is at least one $t : 0 < t < d$ such that $A_{tv} \neq 0$. Furthermore, the degree of f is m.

Proof. Since $\{a_t\}$ has a period d, from Proposition 1 and Fact 2 - (b), the first assertion follows immediately. The second assertion is due to [25].

□

Therefore, any hyper bent function in $\mathcal{DB}$ can be represented by (16), so f has degree m. We are now in a position to establish the following result.

Theorem 4. *Let $f \in \mathcal{DB}$, a hyper-bent function in n variables, for a randomly selected g in $\mathcal{F}$, then*
(1) $Prob\{fg = 0\} = \epsilon_0$ where ϵ_0 is negligible when $n > 6$.
(2) If $fg \neq 0$, then

$$Prob\{deg(fg) \geq deg(f) = n/2\} = 1 - \epsilon$$

where ϵ is negligible when $n > 12$.

The assertion (1) is a special case of the result on annihilators established by Meier, Pasalic and Carlet in [17]. So, we only need to show the assertion (2). For doing so, we need the following some preparations. Let $h(x) \in \mathcal{F}$, and $c_i = h(\alpha^i), i = 0, \cdots, 2^n - 2$. We arrange $\mathbf{c} = (c_0, c_1, \cdots, c_{2^n-2})$ into a v by d array $C = (c_{ij})$ where $c_{ij} = c_{id+j}, 0 \leq i < v, 0 \leq j < d$, i.e.,

$$C = \begin{bmatrix} c_0 & c_1 & \cdots & c_{d-1} \\ c_d & c_{d+1} & \cdots & c_{d+d-1} \\ \vdots & & & \\ c_{(v-1)d} & c_{(v-1)d+1} & \cdots & c_{vd-1} \end{bmatrix} = [\mathbf{C}_0, \mathbf{C}_1, \cdots, \mathbf{C}_{d-1}]$$

where $\mathbf{C}_j$'s are column vectors of C.

Lemma 1. *With the above notation, let $\{C_k\}$ be the Fourier spectral sequence of h, and let*

$$I_0 = \{j \,|\, 0 \leq j < d, H(\mathbf{C}_j) \equiv 0 \,(\mathrm{mod}\, 2)\}$$

$$I_1 = \{j \,|\, 0 \leq j < d, H(\mathbf{C}_j) \equiv 1 \,(\mathrm{mod}\, 2)\}.$$

(a) The Fourier spectra $C_{tv} = 0$ for all t with $0 \leq t < d$ if $\mathbf{c}$ satisfies either $|I_0| = d$ (equivalent to $|I_1| = 0$) or $|I_1| = d$.
(b) If $0 < |I_1| < d$, then there exists some t with $0 < t < d$ such that $C_{tv} \neq 0$.

Proof. Let

$$c(x) = \sum_{k=0}^{2^n-2} c_k x^k.$$

Then we have

$$c(x) = \sum_{0 \leq i < v} \sum_{0 \leq j < d} c_{id+j} x^{id+j}.$$

From Fact 2 -(c), the Fourier transform of h can be given by

$$C_k = c(\alpha^{-k}), k = 0, 1, \cdots, 2^n - 2.$$

Therefore,

$$C_{tv} = c(\alpha^{-tv}) = \sum_{j=0}^{d-1} \left(\sum_{i=0}^{v-1} c_{id+j} \alpha^{-tivd} \right) \alpha^{-jtv}$$

$$= \sum_{j=0}^{d-1} \left(\sum_{i=0}^{v-1} c_{id+j} \right) \alpha^{-jtv} \ (\text{ using } \alpha^{-tivd} = 1)$$
$$= \sum_{j=0}^{d-1} e_j \alpha^{-jtv} \ (\text{ setting } e_j = \sum_{i=0}^{v-1} c_{id+j}). \tag{17}$$

Note that $e_j = 1$ if the Hamming weight of $\mathbf{C}_j = \{c_{id+j}\}_{i=0}^{v-1}$, the jth column vector of C, is odd. Otherwise $e_j = 0$. If $|I_0| = d$ or $|I_1| = d$, then $\mathbf{e}$ is a constant sequence, i.e., $\mathbf{e} = (c, c, \cdots, c)$ where $c = 0$ for $|I_0| = d$ and $c = 1$ for $|I_1| = d$, respectively. Substituting the values of $e_j's$ into (17), we have

$$C_{tv} = c \sum_{j=0}^{d-1} \alpha^{-jtv} = \begin{cases} 0 \ \text{ if } c = 0 \\ \sum_{j=0}^{d-1} \alpha^{-jtv} = 0 \ \text{ for all } \ t : 0 < t < d, \text{ if } \ c = 1. \end{cases}$$

If $0 < |I_1| < d$, then $\mathbf{e}$ is not a constant sequence. We associate $\{e_j\}$ with a function $t(x)$ in $\mathcal{F}$ in the way of (15). Then $\{e_j\}$ can be regarded as a binary sequence of period d. From Fact 2 - (b) and (17), C_{tv} becomes the Fourier spectrum of $t(x)$ or equivalently $\{e_j\}$ at tv, so there is at least one t with $1 \le t < d$ such that $C_{tv} \neq 0$ which completes the proof.

□

We need the following fact which was observed in [17].

Fact 3 *For a boolean function f in n variables, $A(f)$, the set consisting of all annihilators of f, is defined by*

$$A(f) = \{g \in \mathcal{F} \mid fg = 0\}.$$

Then

$$|A(f)| = 2^{2^n - 1 - H(f)}$$

where $H(f)$ is the Hamming weight of f. In particular, if f is a hyper-bent function in $\mathcal{DB}$, then

$$|A(f)| = 2^{v(2^{m-1}+1)}.$$

Lemma 2. *For a given hyper-bent function f in $\mathcal{DB}$, let*

$$S_0(f) = \{g \in \mathcal{F} \mid fg \neq 0, deg(fg) < deg(f)\}.$$

Then

$$|S_0(f)| \le \left(2^{(v-1)2^{m-1}} - 1\right) 2^{v(2^{m-1}+1)}.$$

Proof. We arrange $\{a_i\}$ into a v by d array $A = (a_{ij}), a_{ij} = a_{id+j}$. From the construction of f, the column vectors of A are constant vectors of zero or one. Let $T = \{j \mid a_j = 1, 0 \le j < d\}$. For $g(x) \in \mathcal{F}$, let $b_i = g(\alpha^i), i = 0, 1, \cdots, 2^n - 2$. We arrange $\{b_i\}$ into a v by d array $B = (b_{ij})$ where $b_{ij} = b_{id+j}$, and denote $\mathbf{B}_j$

the jth column vector of B. Let $h(x) = f(x)g(x)$, $c_i = a_i b_i, i = 0, 1, \cdots, 2^n - 2$, $C = (c_{ij})_{v \times d}$ where $c_{ij} = c_{id+j}$, and $\mathbf{C}_j$ the jth column vector of C. Then

$$\mathbf{C}_j = \begin{cases} 0 & \text{if } a_j = 0 \\ \mathbf{B}_j & \text{if } a_j = 1. \end{cases}$$

From Proposition 2, the degree of f is equal to m. Applying Lemma 1 to h, the case $deg(h) < m$ may happen only when $H(\mathbf{B}_j)$ is even for all $j \in T$. Therefore, we may choose the column vector $\mathbf{B}_j$ having even weight for $j \in T$. Thus, there are $\sum_{i=0}^{v-1} \binom{v}{2i} = 2^{v-1}$ choices for such a $\mathbf{B}_j, j \in T$ including the case $\mathbf{B}_j = \mathbf{0}$. The rest of columns can be chosen as arbitrary binary vectors of dimension v. Under this construction of g, if $h = fg \neq 0$, then it is possible that $deg(h) < m$. Thus, there are $2^{(v-1)2^{m-1}} - 1$ choices for $\mathbf{B}_j, j \in T$ and $2^{v(2^{m-1}+1)}$ choices for the rest of the columns of B for which $fg \neq 0$ and possibly, $deg(fg) < m$. Hence

$$|S_0(f)| \leq \left(2^{(v-1)2^{m-1}} - 1\right) 2^{v(2^{m-1}+1)}.$$

□

Proof of Theorem 4. The assertion (1) is a consequence of Fact 3 where $\epsilon_0 \leq \frac{1}{2^{2^{2m-1}}}$ which is negligible when $2m > 6$. For the assertion (2), considering a randomly selected $g \in \mathcal{F}$, we have

$$Prob\{deg(fg) \geq m\} = 1 - Prob\{deg(fg) < m\}. \tag{18}$$

Applying Fact 3 and Lemma 2,

$$\begin{aligned} Prob\{deg(fg) < m\} &= \frac{|S_0(f)| + |A(f)|}{2^{2^{2m}-1}} \leq \frac{2^{(v-1)2^{m-1}} 2^{v(2^{m-1}+1)}}{2^{2^{2m}-1}} \\ &= \frac{2^{2^{2m}-2^{m-1}-1}}{2^{2^{2m}-1}} = \frac{1}{2^{2^{m-1}}}. \end{aligned}$$

Combining with (18), it follows that

$$Prob\{deg(fg) \geq m\} = 1 - \epsilon,$$

where $\epsilon = \frac{1}{2^{2^{m-1}}}$ which is negligible when $m > 6$. □

Remark 6. The result in Theorem 4 can not guarantee there are no low degree approximations for an arbitrary hyper-bent function f, because there may exist some function g in $S_0(f)$ such that $deg(fg) < m$. We have done some experiments for this problem. For example, for the case $n = 4$, so $m = 2$, the function $f(x) = Tr(\alpha x^3)$, given in Example 2, is a hyper-bent function, which is invariant of algebraic attacks. But for $n = 6$ and $n = 8$ there do exist some g such that $deg(fg) < m$ for $fg \neq 0$. We continue to investigate this case for the patterns of such hyper-bent functions.

6 Conclusions and Discussions

Courtois and Meier showed the effective algebraic attacks on several concrete stream cipher systems proposed in the literature when a boolean function employed in the system happens in the following cases: (a) the function itself has a low degree, (b) the function is not of low degree, but one can find another function to multiply the function used in the system such that the resulting product has a low degree or the product is zero (annihilator case), (c) the probabilities that the cases (a) and (b) happen are close to 1, respectively. We have characterized these multipliers for the case of nonzero products. We also derived an algorithm to construct such multipliers. In order to prevent algebraic attacks, Courtois and Meier also introduced the algebraic security criterion for design of boolean functions for cryptographical applications, i.e., a boolean function used in a cryptographical system should not have any low degree approximations or annihilators. If so, we say the function is invariant of algebraic attacks. Our algorithm also provides a method to find such boolean functions which do not have any low degree approximations or annihilators. In light of search for invariants of algebraic attacks, we revealed that the probability that any hyper-bent function is invariant of algebraic attacks is close to 1.

All results obtained in this paper are due to use the (discrete) Fourier transform, which gives a polynomial or trace representation of a boolean function in terms of technics of analysis of pseudo-random sequences. We would like to point out that some trade-offs between the polynomial representation of a function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$ and the boolean representation of the function. In this paper, we use the polynomial representation, which have advantages in analysis of cryptographical properties of functions. Another advantage is to prevent finding a low degree approximation probabilistically, i.e., the case (c) as shown above. In general, for a boolean representation, if the number of monomial terms with high degrees are small, then one can easily remove these terms to obtain a low degree function for which the probability that these two functions are equal is close to 1. However, if a function is in a polynomial form, the degree of the function is governed by the Hamming weights of exponents in monomial trace terms. Removing one or more monomial trace terms from the expression may result in a large change of the distance between the function and the resulting function, since it is equivalent to calculate the distance of two codewords of a cyclic code. So, the probability that these two functions are equal is not close to 1, possibly, close to 1/2. A similar argument is also applied to the case of probabilistically low degree approximations. However, boolean forms are easy to analyze the correlation immunity/resiliency and propagation property, and they can be efficiently implemented at hardware level.

References

1. Frederik Armknecht and Matthias Krause, Algebraic attacks on combiners with memory, *Advances in Cryptology-Crypto'2003*, Lecture Notes in Computer Science,

No. 2729, pp. 162-175, Springer-Verlag, 2003.
2. P. Camion, Claude Carlet, Pascale Charpin and N. Sendrier, On correlation-immune functions, *Advances in Cryptology-Crypto'91*, Lecture Notes in Computer Science, No. 576, Springer-Verlag, 1991, pp. 86-100.
3. A. Canteaut, Claude Carlet, Pascale Charpin and C. Fontaine, On cryptographic properties of the cosets of $R(1, m)$, *IEEE Trans. on Inform. Theory*, vol.47, No. 4, May 2001, pp.1491-1513.
4. C. Carlet, A larger class of cryptographic Boolean functions via a study of the Maiorana-McFarland construction, *Advances in Cryptology-Crypto'2002*, *Lecture Notes in Computer Science*, No. 2442, pp. 549-564, Springer-Verlag, 2002.
5. C. Carlet, More correlation-Immune and resilient functions over Galios fields and Galios rings, *Advances in Cryptology-Eurocrypt'97*, *Lecture Notes in Computer Science*, No. 1233, pp. 422-433, Springer-Verlag, 1997.
6. Nicolas Courtois and Josef Pieprzyk, Cryptanalysis of block ciphers with overdefiend systems of equations, *Asiacrypt'2002*, *Lecture Notes in Computer Science*, No. 2501, Springer-Verlag, 2002.
7. Nicolas Courtois, Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt, *5th International Conference on Information Security and Cryptology (ICISC) 2002*, *Lecture Notes in Computer Science*, No. 2587, pp. 549-564, Springer-Verlag, 2002.
8. Nicolas Courtois and Willi Meier, Algebraic attacks on stream ciphers with lienar feedback, *Advances in Cryptology-EuroCrypt'2003*, Lecture Notes in Computer Science, No. 2656, pp. 345-359, Springer-Verlag, 2003.
9. Nicolas Courtois, Fast algebraic attacks on stream ciphers with lienar feedback, *Advances in Cryptology-Crypto'2003*, Lecture Notes in Computer Science, No. 2729, pp. 176-194, Springer-Verlag, 2003.
10. M.H. Dawson and Stafford E. Tavares, An expanded set of s-box design criteria based on information theory and its relation to differential-like attacks, *Advances in Cryptology-Eurocrypt'91*, Lecture Notes in Computer Science, No. 0547, Springer-Verlag, Berlin, 1991, pp.352-367.
11. J. F. Dillon, Elementary Hadamard difference sets, in *Proc. Sixth S-E Conf. Comb. Graph Theory and Comp.*, 237-249, F. Hoffman et al. (Eds), Winnipeg Utilitas Math (1975).
12. H. Dobbertin, Construction of bent functions and balanced Boolean functions with high nonlinearity, *Proceedings of Fast Software Encryption, Second International Workshop*, Springer-Verlag, 1995, pp. 61-74.
13. S.W. Golomb, On the classification of Boolean functions, *IEEE Trans. on Inform. Theory*, vol.IT-5, pp. 176-186, May 1959. Also appears in S.W. Golomb, *Shift Register Sequences*, Chapter VIII.
14. G. Gong and S.W. Golomb, Transform Domain Analysis of DES, *IEEE Trans. on Inform. Theory*, vol. 45, No.6, September 1999, pp. 2065-2073.
    Design of SAC/PC(l) of Order k Boolean Functions and Three Other Cryptographic Criteria
15. Kaoru Kurosawa and Takashi Satoh Takashi Satoh, Design of SAC/PC(l) of Order k Boolean Functions and Three Other Cryptographic Criteria, *Advances in Cryptology-Eurocrypt'97*, *Lecture Notes in Computer Science*, No. 1233, pp. 434-449, Springer-Verlag, 1997.
16. R.L. McFarland, *A family of Noncyclic Difference Sets* , Journal of Comb. Th. (Series A) 15, pp. 1-10, 1973.

17. Willi Meier, Enes Pasalic and Claude Carlet, Algebraic attacks and decomosition of boolean functions, *Advances in Cryptology-Eurocrypt'2004*, C. Cachin and J. Camenisch Eds., Lecture Notes in Computer Science, No. 3027, Springer-Verlag, 2004, pp. 474-491.
18. S. Maitra and P. Sarkar, Nonlinearity bounds and constructions of resilient boolean functions, *Advances in Cryptology-Crypto'2000*, Lecture Notes in Computer Science, No. 1880, Springer-Verlag, 2000, pp. 515-532.
19. K. Nyberg, Perfect nonlinear S-boxes, *Advances in Cryptology-Eurocrypt'91*, Lecture Notes in Computer Science, No. 0547, Springer-Verlag, Berlin, 1991, pp.378-386.
20. Adi Shamir, Jacques Patarin, Nicolas Courtois, and Alexander Klimov, Efficient algorithms for solving overdefined systems of multivariate polynomial equations, *Advances in Cryptology-EuroCrypt'2000*, Lecture Notes in Computer Science, No. 1807, pp. 392-407, Springer-Verlag, 2000.
21. T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Trans. Inform. Theory*, vol. 30, No. 5, September 1984, pp. 776-780.
22. Guo-zhen Xiao and J.L. Massey, A spectral characterization of correlation-immune combining functions, *IEEE Trans. Inform. Theory*, vol. 34, No. 3, May 1988, pp. 569-571.
23. Y. Zheng and X.M. Zhang, Relationships between bent functions and complementary plateaued functions, *Information and Communications Security'99*, Lecture Notes in Computer Science, No. 1787, pp. 60-75, Berlin, Springer-Verlag, 1999.
24. Y. Zheng and X.M. Zhang, Improved upper bound on the nonlinearity of high order correlation immune functions, *Proceedings of Selected Areas in Cryptography 2000*, Lecture Notes in Computer Science, No. 2012, Springer-Verlag, 2001, pp. 262-274.
25. A. M. Youssef and G. Gong, Hyper-Bent Functions, *Advances in Cryptology, Eurocrypt'2001, Lecture Notes in Computer Science*, Brigit Pfitzmann (Ed). Berlin, Springer-Verlag, 2001, vol. 2045, pp.406-419.