Upper Bounds on Algebraic Immunity of Boolean Power Functions
(Extended Abstract)

Yassir Nawaz[1], Guang Gong[1] and Kishan Chand Gupta[2]

[1]Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, N2L 3G1, CANADA
[2]Centre for Applied Cryptographic Research
University of Waterloo
Waterloo, ON, N2L 3G1, CANADA
ynawaz@engmail.uwaterloo.ca, G.Gong@ece.uwaterloo.ca,
kgupta@math.uwaterloo.ca

Abstract. Algebraic attacks have received a lot of attention in studying security of symmetric ciphers. The function used in a symmetric cipher should have high algebraic immunity ($\mathcal{AI}$) to resist algebraic attacks. In this paper we are interested in finding $\mathcal{AI}$ of Boolean power functions. We give an upper bound on the $\mathcal{AI}$ of any Boolean power function and a formula to find its corresponding low degree multiples. We prove that the upper bound on the $\mathcal{AI}$ for Boolean power functions with Inverse, Kasami and Niho exponents are $\lfloor\sqrt{n}\rfloor + \lceil\frac{n}{\lfloor\sqrt{n}\rfloor}\rceil - 2$, $\lfloor\sqrt{n}\rfloor + \lceil\frac{n}{\lfloor\sqrt{n}\rfloor}\rceil$ and $\lfloor\sqrt{n}\rfloor + \lceil\frac{n}{\lfloor\sqrt{n}\rfloor}\rceil$ respectively. We also generalize this idea to Boolean polynomial functions. All existing algorithms to determine $\mathcal{AI}$ and corresponding low degree multiples become too complex if the function has more than 25 variables. In our approach no algorithm is required. The $\mathcal{AI}$ and low degree multiples can be obtained directly from the given formula.

Keywords: Algebraic attacks, Algebraic immunity, Inverse exponent, Kasami exponent, Polynomial functions, Power functions, Niho exponent.

1 Introduction

The idea behind the algebraic attacks is to express the cipher as a system of multivariate equations whose solution gives the secret key. The complexity of the attack depends on the degree of these equations. Therefore the existence of low degree equations, to express a cipher, is crucial for algebraic attacks.

The algebraic attacks on stream ciphers composed of LFSR(s) and a nonlinear combining function f were proposed by Courtois and Meier in [8, 9]. The authors presented several scenarios under which low degree equations exist for ciphers using a combining function f with small number of inputs. These low degree equations are obtained by producing low degree multiples of f, i.e., by multiplying f with a low degree function g such that fg is of low degree. In [26] Meier, Pasalic and Carlet reduced the scenarios (given in [8, 9]) under which low degree equations may exist to two and showed that existence of low degree equations is equivalent to the existence of low degree annihilators of f or $f+1$.

Krause and Armknecht extended algebraic attacks to combiners with memory in [2]. They proved that algebraic equations always exist for such combiners and also gave an upper bound on the degree of such equations in terms of their input size and memory. In [10] Courtois further extended these attacks to combiners with memory and several outputs and provided an upper bound on the degree of equations for such combiners in terms of the size of their input, output and memory. An improvement on the algebraic attacks called fast algebraic attacks was presented in [7]. These attacks have been further examined in [3, 21].

The first algebraic attack on a block cipher was discussed in [30]. In [11] Courtois and Pieprzyk showed that AES can be attacked by solving a system of quadratic equations. This is possible because the only nonlinear component in AES, i.e., the S-box, can be expressed as a system of quadratic Boolean equations. They also introduced a definition of the resistance of the S-box to algebraic attacks based on the number and type of equations that describe the S-box. Low degree equations that describe the S-box are again essential for the low complexity of the algebraic attack. Cheon and Lee [6] used this definition to determine the resistance of S-boxes (based on Gold, Kasami and inverse exponents) against algebraic attacks. However their results have been disputed by Courtois et al. in [12]. Another algebraic attack on AES was given in [27, 28].

Since the existence of low degree equations for simple combiners, combiners with memory, and S-boxes is important for algebraic attacks, Armknecht combined the three cases in [1]. He showed that finding low degree equations for simple combiners, combiners with memory, and S-boxes can be reduced to the same problem of finding low degree annihilators.

In other direction there is increasing interest in the construction of Boolean functions with highest $\mathcal{AI}$. So far there are only three known constructions [5, 14, 15] that can acheive maximum possible $\mathcal{AI}$ $\lceil \frac{n}{2} \rceil$, where n is the number of inputs to the function. But the constructed function lacks certain cryptographic properties making it unsuitable to be used in a cryptosystem.

Except [27] all other techniques for finding the low degree equations have been developed from the theory of Boolean functions. Even, functions and S-boxes designed over finite field $\mathbb{F}_{2^n}, n > 2$ are analyzed according to this approach. We can refer to them as polynomial functions or mappings. For example the filter function $f : \mathbb{F}_{2^{16}} \rightarrow \mathbb{F}_2$ used in the stream cipher SFINKS [4] is a component of the inverse mapping in $\mathbb{F}_{2^{16}}$. S-box used in AES [13] and stream cipher SNOW [22] consists of inverse mapping in $\mathbb{F}_{2^8}$. A power mapping from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^n}$ can be decomposed into n component functions, from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, called Boolean power functions. In other words a Boolean power function is a monomial trace function which will be introduced later.

In this paper we use the theory of polynomial functions to analyze the $\mathcal{AI}$ of Boolean power function. This approach allows us to obtain meaningful results that are very difficult to obtain from the theory of Boolean functions. For example we derive upper bounds on the $\mathcal{AI}$ of many Boolean power functions that are much lower than the optimal upper bound presented in literature. We show that $\mathcal{AI}$ of functions based on inverse, Kasami and Niho exponents decreases drastically as n increases. More over the existing algorithms to determine the $\mathcal{AI}$ (and finding low degree equations) of functions are very slow and are not practical for $n > 25$. Our approach has no such limitation. The $\mathcal{AI}$ of any Boolean power function can be obtained directly from the formula regardless of the value of n. Similarly the low degree equations required for the algebraic attack are also obtained directly from the formula.

We believe that besides determining the $\mathcal{AI}$ of polynomial functions, the approach presented in this paper can be used to analyze and design polynomial functions with high $\mathcal{AI}$ along with other cryptographic properties.

2 Definitions and Preliminaries

In this section we provide the necessary preliminary material required in the later sections.

2.1 Polynomial Functions

Let $\mathbb{F}_2$ be the finite field of two elements. We consider the domain of an n-variable Boolean function to be the vector space $(\mathbb{F}_2^n, +)$ over $\mathbb{F}_2$, where $+$ is used to denote the addition operator over both $\mathbb{F}_2$ and the vector space $\mathbb{F}_2^m$.

The Hamming weight of an integer i is the number of nonzero coefficients in the binary representation of i and is denoted by $\mathrm{H}(i)$.

For a binary string, λ consecutive ones (1's) preceded by zero and followed by zero is called a run of ones of length λ. We are only interested in the total number of runs of ones in a given binary string and not their length λ. Furthermore we consider our runs of ones to be cyclic. For example 1100011110011111 has two (not three) cyclic runs of ones.

Any n variable Boolean function h: $\mathbb{F}_2^n \to \mathbb{F}_2$, can be uniquely represented as a multivariate polynomial over $\mathbb{F}_2$, called the *algebraic normal form*,

$$h(x_1, \ldots, x_n) = a_0 + \sum_{1 \le i \le n} a_i x_i + \sum_{1 \le i < j \le n} a_{i,j} x_i x_j + \ldots + a_{1,2,\ldots,n} x_1 x_2 \ldots x_n,$$

where the coefficients $a_0, a_i, a_{i,j}, \ldots, a_{1,2,\ldots,n} \in \mathbb{F}_2$. The degree of the Boolean function h, denoted by $deg(h)$, is the same as the degree of the multivariate polynomial.

An (n, m) S-box (or vectorial function) is a map F: $\mathbb{F}_2^n \to \mathbb{F}_2^m$ and has component functions $f_1, \cdots, f_m$.

Let $\mathbb{F}_{2^n}$ be the finite field with 2^n elements. A Trace function Tr: $\mathbb{F}_{2^n} \to \mathbb{F}_{2^m}$, is given by [24, page 51]

$$Tr_m^n(x) = \sum_{i=0}^{n/m-1} x^{2^{mi}}, x \in \mathbb{F}_{2^n}.$$

A Cyclotomic coset C_s modulo 2^n-1 is defined as [25, page 104]

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

where n_s is the smallest positive integer such that $s \equiv s2^{n_s} (\bmod\ 2^n-1)$. The subscript s is chosen as the smallest integer in C_s, and s is called the coset leader of C_s. For example 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.

Any non-zero polynomial function f: $\mathbb{F}_{2^n} \to \mathbb{F}_2$, can be represented as a sum of trace functions [20, page 178]:

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

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

If $f(x)$ is balanced , we have [20]

$$f(x) = \sum_{k\in\Gamma(n)} Tr_1^{n_k}(A_k x^k), A_k \in \mathbb{F}_{2^{n_k}}, x \in \mathbb{F}_{2^n}. \tag{1}$$

The algebraic degree of f, denoted by $deg(f)$, is given by the largest w such that $A_k \neq 0$ and $H(k) = w$. There is a natural correspondence between Boolean functions h and polynomial functions f [20, page 334]. Let $\{\alpha_0, \ldots, \alpha_{n-1}\}$ be a basis for $\mathbb{F}_{2^n}$, then this correspondence is given by:

$$h(x_0, \ldots, x_{n-1}) = f(\alpha_0 x_0 + \ldots \alpha_{n-1}x_{n-1}).$$

A monomial or single trace term function f is a function that can be represented by a single trace term, $f(x) = Tr_1^n(\beta x^t)$ where $\beta \in \mathbb{F}_{2^n}$ and t is the coset leader of C_t.

2.2 Algebraic Immunity $\mathcal{AI}$

A Boolean function f is said to admit an annihilating function g, if $f * g = 0$. In [26] $\mathcal{AI}$ of f, denoted by $\mathcal{AI}(f)$, is defined as the minimum value of d such that f or $f+1$ admits an annihilating function of degree d.

Proposition 1 of [26] states that existence of the relations of form $f * g = h$, where g and h have degree at most d, means the existence of annihilating function g' of degree at most d (as $f*(g+h) = 0$). Therefore if for f we can find a function g such that degree of $f * g$ is d then we can say that $\mathcal{AI}(f) \leq d$.

Fact 1. [8, Theorem 6.0.1] *Let f be any Boolean function with n inputs. Then there is a Boolean function $g \neq 0$ of degree at most $\lceil \frac{n}{2} \rceil$ such that f * g is of degree at most $\lceil \frac{n}{2} \rceil$.*

Fact 1 shows that the upper bound on the $\mathcal{AI}$ of any Boolean function is $\lceil \frac{n}{2} \rceil$. To establish upper bound on the $\mathcal{AI}$ of a polynomial function f we will try to find multipliers g such that the degree of $f * g$ is less than $\lceil \frac{n}{2} \rceil$.

2.3 Monomial Trace Functions and Power Mappings

Monomial Trace functions are represented by a single trace term in polynomial form. There are several compelling reasons to study the $\mathcal{AI}$ of monomial trace functions. Any polynomial function can be expressed as a sum of monomial trace functions. Therefore, for a constant multiplier g, the $\mathcal{AI}$ of any function f is upper bounded by the maximum $\mathcal{AI}$ of any monomial trace function in its polynomial representation. This bound may not always be tight but in certain cases it can reveal the weakness of a function against algebraic attacks. Another important class of functions are the Boolean power functions that can be represented as monomial trace functions. These functions are of interest as they can be used as combining or filtering functions in stream ciphers [4]. These functions can easily be constructed from a power mapping in a finite field. Power mappings can be represented as $F : x \to x^a$ in $\mathbb{F}_{2^n}$ and are classified based on exponent a. Some famous exponents that have been studied for use in S-boxes are Inverse, Gold, Kasami, Welch and Niho [16–18, 23]. These mappings can easily be decomposed into Boolean power functions or monomial trace functions:

Let $\{\alpha_0, ..., \alpha_{n-1}\}$ and $\{\beta_0, ..., \beta_{n-1}\}$ be the dual basis [25, page 117] of $\mathbb{F}_{2^n}$. Then an S-box based on power mapping ($F : x \to x^a$) can be represented as $F(x) = \sum_{j=0}^{n-1} Tr_1^n(\beta_j x^{2^i t})\alpha_j,\ x \neq 0$ and its component functions can be represented as monomial trace functions of the form $f_j(x) = Tr_1^n(\beta_j x^{2^i t})$, where $a \in C_t$. It is conventional to represent monomial trace functions in the form $Tr_1^n(\beta x^t)$ where t is a coset leader of C_t. Note that we can write $a = 2^{-i}t$ for some i. So for any exponent a we can write the monomial trace function in the standard forms as

$$f(x) = Tr_1^n(\beta x^a) = Tr_1^n(\beta x^{2^{-i}t}) = Tr_1^n(\beta^{2^i} x^t),$$

since $Tr_1^n(x) = Tr_1^n(x^2)$. Next we find the $\mathcal{AI}$ of monomial trace functions.

3 Algebraic Immunity of Monomial Trace Functions

We provide the following proposition that will be used to derive upper bound on the $\mathcal{AI}$ of monomial trace functions.

Proposition 1. *Let $f(x) = Tr_1^n(\beta x^t)$ and $g(x) = Tr_1^m(\gamma x^r)$ be monomial trace functions, where $x \in \mathbb{F}_{2^n}$, t and r are the coset leaders of cosets C_t and C_r. The sizes of the cosets C_t and C_r are n and m respectively, $m|n$, and $\beta \in \mathbb{F}_{2^n}$, $\gamma \in \mathbb{F}_{2^m}$. Then*

$$deg(f(x)g(x)) = \max_{0 \leq i < m} H(r + t2^{-i})$$

Proof. Note both $f(x)$ and $g(x)$ are n variable Boolean functions. From the definition of trace function we can write (see also [19])

$$f(x)g(x) = \sum_{j=0}^{n-1} (\beta x^t)^{2^j} \sum_{l=0}^{m-1} (\gamma x^r)^{2^l} = \sum_{j=0}^{n-1} \sum_{l=0}^{m-1} \beta^{2^j} \gamma^{2^l} x^{t2^j + r2^l}$$

$$= \sum_{k=0}^{m-1} Tr_1^n(\gamma\beta^{2^k} x^{r+t2^k}),$$

where the algebraic degree of $f(x)g(x)$ is given by the largest w such that $\gamma\beta^{2^k} \neq 0$ and $H(r+t2^k) = w$. Let $k = m-i$, we have $t2^k \equiv t2^{m-i} \equiv t2^{-i} \bmod 2^n - 1$. Therefore

$$deg(f(x)g(x)) = \max_{0 \leq k < m} H(r+t2^k) = \max_{0 \leq i < m} H(r+t2^{-i})$$

$\square$

Proposition 1 shows that we only need to add r to the members of coset C_t, and the highest hamming weight of the resulting integers gives the maximum possible degree of $f(x)g(x)$.

In the following Theorem we derive an upper bound on the $\mathcal{AI}$ of monomial trace functions based on a property of the exponent t, i.e., the number of runs of 1's in the binary representation of t.

Theorem 1. *Let $l = \lfloor \sqrt{n} \rfloor$, $k = n - \lfloor \frac{n}{l} \rfloor l$ and $f(x) = Tr_1^n(\beta x^t)$, where $\beta \in \mathbb{F}_{2^n}$ and t is the coset leader of C_t. Let $g(x) = Tr_1^m(x^r)$, where* $m = \begin{cases} l, & k = 0; \\ n, & 0 < k < l \end{cases}$, *and* $r = \begin{cases} 1 + \sum_{i=1}^{\frac{n}{l}-1} 2^{il} & , k = 0 \\ 1 + 2^k + \sum_{i=1}^{\lfloor \frac{n}{l} \rfloor - 1} 2^{il+k} & , 0 < k < l \end{cases}$.

Then

$$deg(f(x)g(x)) \leq ul + \left\lceil \frac{n}{l} \right\rceil - 1, \tag{2}$$

where u is the number of runs of 1s in the binary representation of t.

To prove Theorem 1 we need the following two lemmas.

Lemma 1. *r is a coset leader modulo $2^n - 1$.*

Proof. The above can be established by examining the binary representation of r.

Case: $k = 0$

$$
\begin{array}{ll}
r & = \overbrace{\overbrace{00\cdots01}^{l}\,\overbrace{00\cdots01}\ldots\ldots\overbrace{00\cdots01}^{l}}^{n} \\
2r & = 00\cdots10\,00\cdots10\ldots\ldots00\cdots10 > r \\
\cdot & \cdot \qquad\qquad \cdot \\
2^{l-1}r & = 10\cdots00\,10\cdots00\ldots\ldots10\cdots00 > r \\
2^{l}r & = 00\cdots01\,00\cdots01\ldots\ldots00\cdots01 = r
\end{array}
$$

Therefore r is the coset leader modulo $2^n - 1$.

Case: $0 < k < l$

$$\begin{aligned}
r &= \overbrace{\overbrace{00\cdots01}^{l}\,\overbrace{00\cdots01}\ldots\ldots\overbrace{00\cdots01}^{l}\,\overbrace{00\cdots01}^{k}}^{n} \\
2r &= 00\cdots10\,00\cdots10\ldots\ldots00\cdots10\,00\cdots10 > r \\
\cdot &\quad \cdot \qquad\qquad \cdot \\
2^l r &= \overbrace{00\cdots01}^{l}\,\overbrace{00\cdots01}\ldots\ldots\overbrace{00\cdots01}^{k}\,\overbrace{00\cdots01}^{l} > r \\
\cdot &\quad \cdot \qquad\qquad \cdot \\
2^n r &= \overbrace{00\cdots01}^{l}\,\overbrace{00\cdots01}\ldots\ldots\overbrace{00\cdots01}^{l}\,\overbrace{00\cdots01}^{k} = r
\end{aligned}$$

Therefore r is the coset leader modulo $2^n - 1$. □

Note: For $k = 0, |C_r| = l$ and for $0 < k < l, |C_r| = n$, where $|C_r|$ is the size of the coset C_r.

In Lemma 1 $\overbrace{\quad}$ is used to indicate the size of a segment in bits. From here onwards we will use $\overbrace{\quad}$ to represent the size of the segment as before and $\underbrace{\quad}$ to represent the number of 1's in the segment

Lemma 2. *$H(r + t2^{-i}) \leq ul + \lceil \frac{n}{l} \rceil - 1$, $0 \leq i \leq m - 1$ and $r, t, u,$ and l are as defined in Theorem 1.*

Proof. Consider the binary representations of r and t. In Lemma 1, r consists of $\lfloor \frac{n}{l} \rfloor$ identical l bit segments when $k = 0$. If $k \neq 0$, then k least significant bits of r form an additional segment. All $\lceil \frac{n}{l} \rceil$ segments have hamming weight 1. We can segment t in the same way as r however these segments may or may not be identical. We will represent a segment of r and t as r' and t' respectively. Now let us consider the addition of r' and t'. Initially we will restrict ourselves to the case where the binary representation of t' has at most one run. Now consider all possible transitions in t' with and without carry.

Case 1: 1 → 0 transition

$$\begin{aligned}
t' &= \overbrace{111\cdots1}^{j\ \curvearrowright}0\cdots00 \\
r' &= 000\cdots00\cdots01 \\
&+ - - - - - - - \\
&\ \ 111\cdots10\cdots01
\end{aligned}
\qquad\qquad
\begin{aligned}
t' &= \overbrace{111\cdots1}^{j\ \curvearrowright}0\cdots00 \qquad \leftarrow carry \\
r' &= 000\cdots00\cdots01 \\
&+ - - - - - - - \\
&\ \ 111\cdots10\cdots10
\end{aligned}$$

$H(r' + t') = j + 1, j < l - 1$ $\qquad\qquad$ $H(r' + t') = j + 1, j < l - 1$

$$\begin{array}{rl} & \overset{j}{\overbrace{111\cdots11}}\,0 \\ t' = & 111\cdots110 \\ r' = & 000\cdots001 \\ + & -------- \\ & 111\cdots111 \end{array}$$

$H(r'+t') = j+1, j = l-1$

$$\begin{array}{lcl} carry \leftarrow & \overset{j}{\overbrace{111\cdots11}}\,0 & \leftarrow carry \\ t' = & 111\cdots110 & \\ r' = & 000\cdots001 & \\ + & -------- & \\ & 000\cdots000 & \end{array}$$

$H(r'+t') = 0, j = l-1$

Case 2: 0 → 1 transition

$$\begin{array}{rl} t' = & \overbrace{000\cdots0}^{j}\,1\cdots11 \\ r' = & 000\cdots00\cdots01 \\ + & -------- \\ & 000\cdots10\cdots00 \end{array}$$

$H(r'+t') = 1, j < l$

$$\begin{array}{rll} t' = & \overbrace{000\cdots0}^{j}\,1\cdots11 & \leftarrow carry \\ r' = & 000\cdots00\cdots01 & \\ + & -------- & \\ & 000\cdots10\cdots01 & \end{array}$$

$H(r'+t') = 2, j < l$

Case 3: No transition

$$\begin{array}{rl} t' = & 000\cdots00 \\ r' = & 000\cdots01 \\ + & ------ \\ & 000\cdots01 \end{array}$$

$H(r'+t') = 1$

$$\begin{array}{rll} & & \leftarrow carry \\ t' = & 000\cdots00 & \\ r' = & 000\cdots01 & \\ + & ------ & \\ & 000\cdots10 & \end{array}$$

$H(r'+t') = 1$

$$\begin{array}{ll} carry \leftarrow & \\ t' & = 111\cdots11 \\ r' & = 000\cdots01 \\ & + ------ \\ & 000\cdots00 \end{array}$$

$H(r'+t') = 0$

$$\begin{array}{lll} carry \leftarrow & & \leftarrow carry \\ t' & = 111\cdots11 & \\ r' & = 000\cdots01 & \\ & + ------ & \\ & 000\cdots01 & \end{array}$$

$H(r'+t') = 1$

From the above cases it is clear that $H(r'+t')$ achieves maximum value l for $1 \rightarrow 0$ transition (no carry case). For $0 \rightarrow 1$ transition the maximum value is 2 and when there is no transition it is 1.

Now consider the following complete binary representation of r and t.

$$\begin{array}{l} t = \overbrace{**\cdots**}^{l}\ldots\ldots\overbrace{1110\cdots00}^{l}\ldots\ldots\overbrace{**\cdots**}^{l} \\ r = 0\,0\cdots0\,1\ldots\ldots0000\cdots01\ldots\ldots0\,0\cdots0\,1 \end{array}$$

where * can be either 1 or 0.

First we assume that each segment t' has at most one run. To get the maximum possible value of $H(r+t2^{-i})$ each segment with a $1 \rightarrow 0$ transition must contribute l number of 1's to the sum. Since there are u number of $1 \rightarrow 0$ transitions this adds up to ul number of 1's. For u number of $1 \rightarrow 0$ transitions there are at most u segments with $0 \rightarrow 1$ transitions, each contributing a maximum of 2 number of 1's to the sum, i.e., $2u$ number of 1's. All the remaining $\lceil \frac{n}{l} \rceil - 2u$ segments with no transitions can contribute at most single 1 to the sum. So the total contribution is $ul + 2u + \lceil \frac{n}{l} \rceil - 2u = ul + \lceil \frac{n}{l} \rceil$. Now consider the first segment from right to left that contains a $0 \rightarrow 1$ transition (right most segment with a $0 \rightarrow 1$ transition). For this segment to contribute 2 number of 1's to the sum it must receive a carry, otherwise it will contribute a single 1 (see case 2). Since the right most segment never receives a carry, the only way a carry can be generated between the right most segment and the right most segment with a $0 \rightarrow 1$ transition is due to the presence of a segment with all 1's (see all cases without carry). However this all 1's segment contributes zero number of 1's to the sum. Therefore we subtract 1 from $ul + \lceil \frac{n}{l} \rceil$. Therefore the maximum possible value of $H(r + t2^{-i})$ is $ul + \lceil \frac{n}{l} \rceil - 1$.

Suppose a segment t' has more than one runs. Then it must contain a $1 \rightarrow 0$ transition. In our analysis of segments with single runs, we assumed that each segment with a $1 \rightarrow 0$ transition must contribute l number of 1's to the sum. As the size of the segment is l the contribution of a segment with more than one runs must be less than or equal to the contribution of the segment with one run. Therefore $H(r + t2^{-i})$ is upper bounded by $ul + \lceil \frac{n}{l} \rceil - 1$

$\square$

Proof of Theorem 1.
The assertion follows directly from Lemma 1, Lemma 2 and Proposition 1.

$\square$

Remark 1. The significance of Theorem 1 is two fold. It gives the upper bound on the $\mathcal{AI}$ of f, i.e., $ul + \lceil \frac{n}{l} \rceil - 1$ and it also gives the low degree multiplier g ($deg(g) = \lceil \frac{n}{l} \rceil$). In Theorem 1 we give only one multiplier however we can get $2^m - 1$ distinct non zero multipliers by taking $g(x) = Tr_1^m(\beta x^r)$, where $\beta \in \mathbb{F}_{2^m}$. Note only m of them are linearly independent. Also note that these multipliers can be computed directly from the formula in Theorem 1 with almost no effort.

Remark 2. From Fact 1 we know that $\mathcal{AI}$ of any function is at most $\lceil \frac{n}{2} \rceil$. Let v be the degree of $f(x)$, then, to obtain a meaningful upper bound on $\mathcal{AI}$, $deg(f(x)g(x)) \leq \min\left(v, \lceil \frac{n}{2} \rceil\right)$. So we have the following condition on u,

$$u \leq \min\left(\frac{v - \lceil \frac{n}{l} \rceil + 1}{l}, \frac{\lceil \frac{n}{2} \rceil - \lceil \frac{n}{l} \rceil + 1}{l}\right). \tag{3}$$

For many cryptogaphically useful power mappings, u is very small. For example, $u = 1$ for inverse, and $u = 2$ for Kasami, Gold, Welch and Niho. Therefore Theorem 1 can give very useful bounds for these mappings. In fact in most cases using the proof technique of Theorem 1 and exploiting the specific binary form of each exponent, we can further improve this bound. Since functions with Gold and Welch exponents have very small degrees (2 and 3 respectively) we will only consider inverse, Kasami and Niho exponents in this paper.

4 Inverse Exponent

Inverse mappings $x \to x^{-1}$ in $\mathbb{F}_{2^n}$ can be decomposed in to n monomial trace functions of the form $Tr_1^n(\beta x^{-1})$. The degree of the monomial trace functions with inverse exponents is $n-1$. The inverse exponent consists of a single run of 1's. From Theorem 1 its $\mathcal{AI}$ is upper bounded by $l + \lceil \frac{n}{l} \rceil - 1$. However in Theorem 2 we show that for inverse function this bound can be improved to $l + \lceil \frac{n}{l} \rceil - 2$.

Lemma 3. *Let $t = 2^{n-1} - 1$. Then $H(r + t2^{-i}) \le l + \lceil \frac{n}{l} \rceil - 2$, $0 \le i \le m-1$ and r is defined as in Theorem 1.*

Proof. Consider the binary representation of r and t.

$$\begin{array}{rl}
t = & \overbrace{01\cdots 11}^{l}\,\overbrace{11\cdots 11}\ldots\ldots\overbrace{11\cdots 11}^{l}\,\overbrace{11\cdots 11}^{l} \\
r = & 00\cdots 01\,00\cdots 01\ldots\ldots 00\cdots 01\,00\cdots 01 \\
+ & ----------------- \\
 & \underbrace{10\cdots 01}_{2}\,\underbrace{00\cdots 01}_{1}\ldots\ldots\underbrace{00\cdots 01}_{1}\,\underbrace{00\cdots 00}_{0}
\end{array}$$

$H(r + t) = \lceil \frac{n}{l} \rceil$.
We can see that $H(r + t2^{-i})$ is maximized when $i = l - 2$

$$\begin{array}{rl}
t2^{-(l-2)} & = \overbrace{11\cdots 01}^{l}\,\overbrace{11\cdots 11}\ldots\ldots\overbrace{11\cdots 11}^{l}\,\overbrace{11\cdots 11}^{l} \\
r & = 00\cdots 01\,00\cdots 01\ldots\ldots 00\cdots 01\,00\cdots 01 \\
 & + ----------------- \\
 & \quad\underbrace{11\cdots 11}_{l}\,\underbrace{00\cdots 01}_{1}\ldots\ldots\underbrace{00\cdots 01}_{1}\,\underbrace{00\cdots 00}_{0}
\end{array}$$

$H(r + t2^{-(l-2)}) = l + \lceil \frac{n}{l} \rceil - 2$.

Therefore $H(r + t2^{-i}) \le l + \lceil \frac{n}{l} \rceil - 2$, $0 \le i \le n - 1$.

$\square$

Theorem 2. *Let $f(x) = Tr_1^n(\beta x^{-1})$ and $g(x) = Tr_1^m(x^r)$. Then*

$$deg(f(x)g(x)) = l + \left\lceil \frac{n}{l} \right\rceil - 2 \tag{4}$$

where β, m, r and l are the same as defined in Theorem 1.

Proof. From Lemma 3, $t = 2^{n-1} - 1$. Since $Tr_1^n(x) = Tr_1^n(x^2)$, we have

$$f(x) = Tr_1^n(\beta x^{-1}) = Tr_1^n(\beta x^{2t}) = Tr_1^n(\beta^{2^{n-1}} x^t). \tag{5}$$

From Lemma 1, r is a coset leader and from Lemma 3, Eqn.(5) and Proposition 1, $deg(f(x)g(x)) \le l + \lceil \frac{n}{l} \rceil - 2$. $\square$

This bound on $\mathcal{AI}$ is much less than theoretical optimal value $\lceil \frac{n}{2} \rceil$ for higher values of n (see Table 1).

5 Kasami Exponents

An n variable monomial trace function with Kasami exponent, $f(x)$, can be defined as $f(x) = Tr_1^n(\beta x^e)$, where $e = 2^{2s} - 2^s + 1$, gcd(n, s)=1 and $1 \leq s \leq \frac{n}{2}$ [23]. Note that $f(x)$ is not balanced for all values of n. The algebraic degree of $f(x)$ is $s + 1$. We will only consider Kasami exponents that give the highest algebraic degree. The Kasami exponent consists of 2 runs of 1's in its binary representation. From Theorem 1 its $\mathcal{AI}$ is upper bounded by $2l + \lceil \frac{n}{l} \rceil - 1$. However in Theorem 3 we show that this bound can be improved to $l + \lceil \frac{n}{l} \rceil$.

Theorem 3. *Let $f(x) = Tr_1^n(\beta x^e)$, where e is the Kasami exponent that gives highest algebraic degree, i.e., $\frac{n+1}{2}$ for n odd, $\frac{n}{2}$ for $n \equiv 0$ mod 4 and $\frac{n}{2} - 1$ for $n \equiv 2$ mod 4. Then*

$$deg(f(x)g(x)) \leq l + \left\lceil \frac{n}{l} \right\rceil$$

where β, l and $g(x)$ are defined in Theorem 1.

Proof. Let $t = e2^{-s}$, then

$$f(x) = Tr_1^n(\beta x^e) = Tr_1^n(\beta x^{t2^s}) = Tr_1^n(\beta^{2^{n-s}} x^t), \quad (6)$$

since $Tr_1^n(x) = Tr_1^n(x^2)$. The assertion follows directly from Lemma 5 (Proof is given in appendix), Lemma 1, Eqn.(6) and Proposition 1. □

Remark 3. Though in Theorem 3 bound on $\mathcal{AI}$ is proved for Kasami exponent that gives the highest algebraic degree, it is easy to prove that this bound holds for any Kasami exponent. The proof is given in [29]. This bound is much lower than the optimal bound $\lceil \frac{n}{2} \rceil$ for large n (see Table 1).

6 Niho Exponent

An n variable monomial trace function with Niho exponent [6, 17], $f(x)$, can be defined as $f(x) = Tr_1^n(\beta x^e)$, where $e = 2^s + 2^{\frac{s}{2}} - 1$, $n = 2s + 1$ when s is even and $e = 2^s + 2^{\frac{3s+1}{2}} - 1$, $n = 2s + 1$ when s is odd. The degree of Niho function in n variables is $\frac{n+3}{4}$ for $n \equiv 1$ mod 4 and $\frac{n+1}{2}$ for $n \equiv 3$ mod 4.

The Niho exponent consists of 2 runs of 1's in its binary representation. From Theorem 1 its $\mathcal{AI}$ is upper bounded by $2l + \lceil \frac{n}{l} \rceil - 1$. However Theorem 4 show that this bound can be improved to $l + \lceil \frac{n}{l} \rceil$. The proof of the theorem is very similar to the proof of Kasami case, so we provide Theorem 4 without proof (proof is given in [29]).

Theorem 4. *Let $f(x) = Tr_1^n(\beta x^e)$, where e is a Niho exponent. Then*

$$deg(f(x)g(x)) \leq l + \left\lceil \frac{n}{l} \right\rceil$$

where β, l and $g(x)$ are defined in Theorem 1.

Table 1 shows how the upper bound on the $\mathcal{AI}$ of monomial trace functions with Inverse, Kasami and Niho exponents decreases as n increases. *Also note that in [4], $\mathcal{AI}$ of inverse function for $n = 16$ is given as 6 which is confirmed by our bound. This shows that our bound is tight for $n = 16$.*

f	n	$deg(f)$	Bound from Fact 1 ($\lceil \frac{n}{2} \rceil$)	Our bound on $\mathcal{AI}$
Inverse	16	15	8	6
	36	35	18	10
	100	99	50	18
Kasami	16	8	8	8
	36	18	18	12
	100	50	50	20
Niho	15	8	8	8
	35	18	18	12
	99	50	50	20

Table 1. $\mathcal{AI}$ bounds for Inverse, Kasami and Niho functions

7 Generalization to Polynomial Functions

Any balanced polynomial function $f(x)$ can be represented by Eqn.(1) which is reproduced here as

$$f(x) = \sum_{k \in \Gamma(n)} Tr_1^{n_k}(A_k x^k), A_k \in \mathbb{F}_{2^{n_k}}, x \in \mathbb{F}_{2^n},$$

which is simply a sum of monomial trace functions. If we only consider monomial trace functions as multipliers the result in Theorem 1 can be generalized to all balanced polynomial functions. Let u_k be the number of runs of 1's in the binary representation of k and $u = \max_{k \in \Gamma(n)}\{u_k\}$ such that $A_k \neq 0$. Let $g(x)$ be a monomial trace function defined in Theorem 1. Then

$$deg(f(x)g(x)) \leq ul + \left\lceil \frac{n}{l} \right\rceil - 1.$$

To obtain a meaningful bound on $\mathcal{AI}$ of f, u must satisfy Eqn.(3). The above result implies that $\mathcal{AI}$ of f is upper bounded by the maximum $\mathcal{AI}$ of the single trace functions in the polynomial representation of f.

8 Conclusions

In this paper we use the theory of polynomial functions to provide an upper bound on $\mathcal{AI}$ of Boolean power functions. The low degree multiples are also obtained directly from the formula for any n. This is particularly useful as there are no existing algoritms to find the $\mathcal{AI}$ of a function with more than 25 variables. We improve the $\mathcal{AI}$ bound on inverse, Kasami and Niho functions and show that their $\mathcal{AI}$ is very low. We also generalize our results to polynomial functions.

References

1. F. Armknecht, On the Existence of Low-degree Equations for Algebraic Attacks, *Cryptology ePrint Archive, Report 2004/185*, http://eprint.iacr.org/, 2004.

2. F. Armknecht, Algebraic Attacks on Combiners with Memory, *Advances in Cryptology-CRYPTO 2003*, LNCS 2729, pp. 162-176, Springer-Verlag, 2003.
3. F. Armknecht, Improving Fast Algebraic Attacks *Fast Software Encryption 2004*, LNCS 3017, pp. 65-82, Springer-Verlag, 2003.
4. A. Braeken, J. Lano, N. Mentens, B. Preneel and I. Verbauwhede, SFINKS: A Synchronous Stream Cipher for Restricted Hardware Environments, *eSTREAM Project report 2005/026*, Available at http://www.ecrypt.eu.org/stream/.
5. A. Braeken and B. Preneel, On the Algebraic Immunity of Symmetric Boolean Functions,*To appear in Indocrypt 2005.*
6. J. Cheon and D. Lee, Resistance of S-Boxes Against Algebraic Attacks, *Fast Software Encryption 2004*, LNCS 3017, pp. 83-94, Springer-Verlag, 2004.
7. N. Courtois, Fast Algebraic Attacks on Stream Ciphers with Linear Feedback, *Advances in Cryptology-CRYPTO 2003*, LNCS 2729, pp. 176-194, Springer-Verlag, 2003.
8. N. Courtois and W. Meier, Algebraic Attacks on Stream Ciphers with Linear Feedback, *Advances in Cryptology-EUROCRYPT 2003*, LNCS 2656, pp. 346-359, Springer-Verlag, 2003.
9. N. Courtois and W. Meier, Algebraic Attacks on Stream Ciphers with Linear Feedback, Extended version of [8], available at http://cryptosystem.net/stream
10. N. Courtois, Algebraic Attacks on Combiners with Memory and Several Outputs, *ICISC 2004*, LNCS 3506, pp. 3-20, Springer-Verlag, 2004.
11. N. Courtois and Pieprzyk J., Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, *Advances in Cryptology-ASIACRYPT 2002*, LNCS 2501. Springer-Verlag, 2002.
12. N. Courtois, B. Debraize and E. Garrido, On Exact Algebraic [Non]Immunity of S-boxes Based on Power Functions, *Cryptology ePrint Archive, Report 2005/203*, http://eprint.iacr.org/, 2005.
13. J. Daemen, and V.Rijmen, *The Design of Rijndael*, Springer-Verlag, 2002.
14. D. K. Dalai, K. C. Gupta and S. Maitra, Cryptographically Significant Boolean Functions: Construction and Analysis in Terms of Algebraic Immunity. *Fast Software Encryption 2005*, LNCS 3557, pp. 98-111, Springer-Verlag, 2005.
15. D. K. Dalai, S. Maitra and S. Sarkar, Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity. *To appear in Designs, Codes and Cryptography.*
16. H. Dobbertin, Almost Perfect Nonlinear Power Functions on $GF(2^n)$: The Welch Case. *I*EEE Transactions on Information Theory, Vol. 45, No. 4, pp. 1271-1275, 1999.
17. H. Dobbertin, Almost Perfect Nonlinear Power Functions on $GF(2^n)$: The Niho Case. *I*nformation and Computation, Vol. 151, pp. 57-72, 1998.
18. R. Gold, Maximal Recursive Sequences with 3 valued cross-correlation function, *IEEE Transactions on Information Theory*, Vol. 14, pp. 154-156, 1968.
19. S. W. Golomb and G. Gong, Hyper-Cyclotomic Algebra, *Sequences and their Applications, SETA'01*, Discrete Mathematics and Theoretical Computer Science, Springer, 2001, pp. 154-165. CORR 2001-33.
20. S. W. Golomb, and G. Gong, *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*, Cambridge University Press, ISBN 0521821045, 2005.
21. P. Hawkes, G. Rose, Rewriting Variables: The Complexity of Fast Algebraic Attacks on Stream Ciphers, *Advances in Cryptology-CRYPTO 2004*, volume LNCS 3152, pp. 390-406, Springer-Verlag, 2004.
22. P. Ekdahl, and T. Johansson, SNOW-A New Version of the Stream Cipher SNOW, *Selected Areas in Cryptography, 2002*, LNCS 2595, pp. 47-61, Springer-Verlag 2003.

23. T. Kasami, The Weight Enumerators for Several Classes of Subcodes of the Second Order Binary Reed-Muller Codes, *Infor. Contr.*, Vol. 18, pp. 369-394, 1971.
24. R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications.* Cambridge University Press, 1994.
25. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes.* North Holland, 1986.
26. W. Meier, E. Pasalic, and C. Carlet, Algebraic Attacks and Decomposition of Boolean Functions, *Advances in Cryptology EUROCRYPT 2004*, LNCS 3027, pp.474-491, Springer-Verlag, 2004.
27. S. Murphy and M. Robshaw, Essential Algebraic Structure within AES, *Advances in Cryptology Crypto 2002*, LNCS 2442, pp.1-16, Springer-Verlag, 2002.
28. S. Murphy and M. Robshaw, Comments on the Security of the AES and the XSL Technique, *Electronic Letters*, Vol. 39, pp. 26-38, 2003.
29. Y. Nawaz, G. Gong and K. Gupta, Upper Bounds on Algebraic Immunity of Boolean Power Functions, *Preprint.*
30. I. Schaumuller-Bichl, Cryptanalysis of the Data Encryption Standard by the Method of Formal Coding, *Advances in Cryptology EUROCRYPT-1982*, LNCS 149, pp.235-255, Springer-Verlag, 1983.

A Appendix

In order to study the binary representation of e we need the following result.

Lemma 4. *Let $n > 4$ be any integer and $f(x) = Tr_1^n(\beta x^e)$, where $e = 2^{2s} - 2^s + 1$, gcd(n, s)=1 and s is the highest value less than $\frac{n}{2}$. Then*

1. *If $n \equiv 0 \bmod 4$, then gcd(n, s)=1 where $s = \frac{n}{2} - 1$*
2. *If $n \equiv 2 \bmod 4$, then gcd(n, s)=1, where $s = \frac{n}{2} - 2$*
3. *If n is odd gcd(n, s)=1, where $s = \frac{n-1}{2}$*

Proof. Let n be even and 2 divides $\frac{n}{2}$. Let $k > 1$ be an integer such that k divides both n and $\frac{n}{2} - 1$. As $\frac{n}{2}$ is even, $\frac{n}{2} - 1$ is odd, and so divisor k must be odd. Since k divides n, and k is odd, it must also divide $\frac{n}{2}$ and hence it can not divide $\frac{n}{2} - 1$. Hence the contadiction. So $\gcd(n, \frac{n}{2} - 1)$=1 and hence $s = \frac{n}{2} - 1$.
The proves of the other two cases are similar.

□

Lemma 5. *Let $t = e2^{-s}$ where e and s are defined in Lemma 4. Then $H(r + t2^{-i}) \leq l + \lceil \frac{n}{l} \rceil$, $0 \leq i \leq m - 1$ and r is defined in Theorem 1.*

Proof. From Lemma 4 the binary representation of t is :

$$t = 2^{\frac{n-1}{2}} + 2^{\frac{n+1}{2}} - 1 = \overbrace{00\cdots01}^{s}0\overbrace{11\cdots11}^{s}, \quad n \text{ is odd.}$$

$$t = 2^{\frac{n}{2}-1} + 2^{\frac{n}{2}+1} - 1 = \overbrace{00\cdots01}^{s}00\overbrace{11\cdots11}^{s}, \quad n \equiv 0 \bmod 4.$$

$$t = 2^{\frac{n}{2}-2} + 2^{\frac{n}{2}+2} - 1 = \overbrace{00\cdots01}^{s}0000\overbrace{11\cdots11}^{s}, \quad n \equiv 2 \bmod 4.$$

Let us consider the addition of r and t. We only consider the case where $2 \mid \frac{n}{2}$. The other two cases are similar. The binary representation of t has 2 runs of 1's out of which one run consists of single 1. This 1 can only contribute a single 1 in any segment of $r+t$. Now we can see that $H(r + t2^{-i})$ is maximized when $i = l - 1$.

$$
\begin{array}{ll}
t2^{-(l-1)} & = \overbrace{11\cdots10}^{l}\,\overbrace{00\cdots00}\ldots\ldots\overbrace{0\cdots100}^{l}\,\overbrace{001\cdots11}^{l}\ldots\ldots\overbrace{11\cdots11} \\
r & = 00\cdots01\;00\cdots01\ldots\ldots0\cdots001\;000\cdots01\ldots\ldots00\cdots01 \\
 & + - \\
 & \;\underbrace{11\cdots11}_{l}\,\underbrace{00\cdots01}_{1}\ldots\ldots\underbrace{0\cdots101}_{2}\,\underbrace{010\cdots01}_{2}\ldots\ldots\underbrace{00\cdots00}_{0}
\end{array}
$$

$H(r + t2^{-(l-1)}) = l + \lceil \frac{n}{l} \rceil$.

Therefore $H(r + t2^{-i}) \leq l + \lceil \frac{n}{l} \rceil$, $0 \leq i \leq n-1$

$\square$