Higher Order Algebraic Attacks on Stream Ciphers

Qichun Wang * and Thomas Johansson †

Abstract

In this paper we introduce a new type of algebraic attacks, called higher order algebraic attacks, with applications towards cryptanalysis of stream ciphers. Its efficiency is described through the new concept r-order algebraic immunity of a Boolean function. We show that the 2-order algebraic immunity of the following two classes of Boolean functions is equal to 1, which gives very efficient attacks on them (substantially and greatly outmatching previously known attacks):

(a) The class of Carlet-Feng functions, proposed at Asiacrypt 2008, having optimum algebraic degree, optimum algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity.

(b) The class of rotation symmetric Boolean functions, an interesting and well studied class of Boolean functions, which can be implemented efficiently.

Keywords: Stream ciphers, Boolean functions, higher order algebraic attacks, higher order algebraic immunity.

1 Introduction

Recently, algebraic attacks and fast algebraic attacks have received a lot of attention in the cryptographic community. They might be efficient against LFSR-based stream ciphers as well as on block ciphers [6, 7, 8, 9, 32]. To measure the resistance against algebraic attacks, the notion of algebraic immunity has been proposed by Courtois and Meier: for a given Boolean function f, any Boolean function $g \neq 0$ such that $f \cdot g = 0$ or $(f+1) \cdot g = 0$ should have high algebraic degree. It is known that the algebraic immunity of an n-variable Boolean function is upper bounded by $\lceil \frac{n}{2} \rceil$. To resist fast algebraic attacks, a large algebraic immunity is not sufficient. It should also have a *high degree product*, HDP [15]: for a given Boolean function f, for any non-annihilating function g, $\deg(g) + \deg(f \cdot g)$ should be high.

In another direction, Rϕnjom and Helleseth recently found a new kind of algebraic attack on filter generators [16], which is very efficient. Its time complexity is roughly $O(D)$, where $D = \Sigma_{i=1}^{d} \binom{n}{i}$ and d is the degree of the Boolean function. But it needs $O(D)$ keystream bits, which is much more than classical algebraic attacks. The filter function should have very high algebraic degree to resist this attack.

In the search for lower degree relations, Fischer and Meier investigated augmented functions in S-boxes [13]. They gave a definition of algebraic immunity of an S-box: Let $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$. Given

*School of Computer Science, Fudan University, Shanghai 200433, PR China. Email: 032018023@fudan.edu.cn

†Department of Electrical and Information Technology, Lund University, Lund, Sweden 221 00 Email: thomas@eit.lth.se

some fixed output y, let d be the minimum degree of a non-zero conditional equation $F_y(x) = 0$ which holds for all $x \in S^{-1}(y)$. Then the algebraic immunity of S is defined by the minimum of d over all $y \in \mathbb{F}_2^m$. S-boxes should have good immunity against algebraic attacks on augmented functions.

There are many classes of Boolean functions achieving optimum algebraic immunity [2, 4, 5, 6, 11, 12, 14, 15], and a few of them achieve also a good nonlinearity, let alone achieving a high order of HDP at the same time. At Asiacrypt 2008 [5], Carlet and Feng proposed an infinite class of balanced functions with optimum algebraic degree, optimum algebraic immunity and a much better nonlinearity than all the previously obtained infinite classes of functions. Moreover, they checked that the functions also have a good immunity against fast algebraic attacks. In [6], they generalized these functions to a vectorial form. The cryptographic properties of these functions are known to be very good, but a practical problem is its implementation. We will show that our new attacks apply to these functions.

The class of rotation symmetric Boolean functions is an interesting choice for a Boolean function and this class has been studied in a number of papers [1, 18, 19, 21, 23, 24, 29, 30]. They can be represented in a very compact way, both in their algebraic normal forms and in their truth table form [3], and they can be implemented efficiently [17]. Previous work has demonstrated that rotation symmetric Boolean functions is a class of functions, which contains Boolean functions with excellent cryptographic properties [18, 22, 25, 26, 27, 28]. However, as we will show, they are vulnerable to higher order algebraic attacks.

In this paper, we propose a generalization of the traditional algebraic attacks as well as fast algebraic attacks. The new approach builds a low degree equation using r different initial equations coming from evaluating the Boolean function in r different points. We call this approach higher order algebraic attacks. This also leads to a generalization of the concept of algebraic immunity to the new notion of *r-order algebraic immunity*. The notion comes from the fact that classical algebraic attacks only consider one equation of the form $f(x) = c$ whereas the new approach considers r different equations, and e.g. seeks an annihilator to a function involving r equations. The approach is a generalization of algebraic attacks; it is in general more complex to compute an annihilator, but it can give much more powerful results.

To show its usefulness, we give very efficient attacks on the class of Carlet-Feng functions and the class of rotation symmetric Boolean functions used in a filter generator. The new attacks substantially and greatly outmatch all previously known attacks. As a consequence, we observe that previously constructed functions with good cryptographic properties may not be resistant to a higher order algebraic attack. In order to construct a secure stream cipher, the new notions of higher order algebraic immunities of Boolean functions should also be considered, as an additional criteria.

The paper is organized as follows. Some basics on Boolean functions are introduced in Section 2. We assume a filter generator in our analysis and some of its properties are introduced in Section 3. In Section 4, we then introduce higher order algebraic attacks. In Section 5 we apply the new attacks on Carlet-Feng functions and in Section 6 we apply them on rotation symmetric Boolean functions. We end in Section 7 with a few conclusions.

2 Preliminaries

We start with some basics on Boolean functions. Let $\mathbb{F}_2^n$ be the n-dimensional vector space over the finite field $\mathbb{F}_2$. A Boolean function of n variables is a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. We denote by B_n the set of all n-variable Boolean functions.

Any $f \in B_n$ can be uniquely represented as a multivariate polynomial

$$f(x_1, ..., x_n) = \sum_{K \subseteq \{1,2,...,n\}} a_K \prod_{k \in K} x_k,$$

which is called its algebraic normal form (ANF). The algebraic degree of f, denoted by $\deg(f)$, is the number of variables in the highest order term with nonzero coefficient.

A Boolean function is affine if there exists no term of degree strictly greater than 1 in the ANF and the set of all affine functions is denoted by A_n.

Let

$$1_f = \{x \in \mathbb{F}_2^n | f(x) = 1\},\ 0_f = \{x \in \mathbb{F}_2^n | f(x) = 0\}.$$

The cardinality of 1_f, denoted by $wt(f)$, is called the Hamming weight of f. The Hamming distance between two functions f and g, denoted by $d(f, g)$, is the Hamming weight of $f + g$. We say that an n-variable Boolean function f is balanced if $wt(f) = 2^{n-1}$.

Let $f \in B_n$. The nonlinearity of f is its distance from the set of all n-variable affine functions, i.e.,

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

For any $f \in B_n$, a nonzero function $g \in B_n$ is called an annihilator of f if $f \cdot g = 0$, and the algebraic immunity of f, denoted by $\mathcal{AI}(f)$, is the minimum value of d such that f or $f + 1$ admits an annihilator of degree d.

Let $\mathbb{F}_{2^n}$ denote a finite field with 2^n elements. It can be viewed as an n-dimensional vector space over its subfield $\mathbb{F}_2$. Every function $f : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ can be uniquely represented as a polynomial $\sum_{i=0}^{2^n-1} a_i x^i$ (called its univariate representation), where $a_i \in \mathbb{F}_{2^n}$, and f is a Boolean function if and only if $\sum_{i=0}^{2^n-1} a_i x^i \in \mathbb{F}_2$ for any $x \in \mathbb{F}_{2^n}$. Given a basis $(\beta_1, \beta_2, \cdots, \beta_n)$, we can identify any element $x = \sum_{i=1}^{n} x_i \beta_i \in \mathbb{F}_{2^n}$ with the n-tuple of its coordinates $(x_1, x_2, \cdots, x_n) \in \mathbb{F}_2^n$. f can then be represented as an n-variable polynomial over $\mathbb{F}_2$. It is easy to see that $x^{2^j} = x_1 \beta_1^{2^j} + \cdots + x_n \beta_n^{2^j}$, for any $1 \leq j \leq n$. Hence the algebraic degree of the function equals the maximum $w_2(i)$ such that $a_i \neq 0$, where $w_2(i)$ equals the number of 1's in the binary expansion of i. Let $tr(x)$ denote the trace function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, $tr(x) = x + x^2 + \cdots + x^{2^{n-1}}$. Clearly, $\deg(tr(x)) = 1$. In what follows, we denote an element of $\mathbb{F}_2^n$ by a column vector.

3 Filter Generators and Their Cyclic Groups

As our cryptographic application we consider a stream cipher built from a filter generator, where the filter generator consists of a length n linear feedback shift register that generates an m-sequence and a Boolean function that combines state bits from the shift register. This is a standard model used in numerous papers. Clearly, most results in this paper can be generalized to other similar generators.

A typical filter generator uses an m-sequence $\mathbf{s} = s_0, s_1, s_2, \ldots$ and a filter function $f \in B_n$. Denote the output of the filter generator by $c_0, c_1, c_2, \ldots$. Any term s_t of the m-sequence $\mathbf{s}$ is uniquely determined by a linear function of the initial state $(s_0, s_1, \ldots, s_{n-1})$. Let

$$f(s_t, s_{t+1}, \ldots, s_{t+n-1}) = c_t, t = 0, 1, 2, \ldots$$

Then we have

$$f(L^t(s_0, s_1, \ldots, s_{n-1})) = c_t, t = 0, 1, 2, \ldots,$$

where L^t are vectorial Boolean functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ and all components are linear functions.

An algebraic attack is an approach to solve this system of equations efficiently. If $\mathcal{AI}(f)$ is low and there exists a $g \in B_n$ of low degree such that $fg = 0$. Then

$$f(L^t(s_0, s_1, \ldots, s_{n-1}))g(L^t(s_0, s_1, \ldots, s_{n-1})) = c_t g(L^t(s_0, s_1, \ldots, s_{n-1})) = 0, t = 0, 1, 2, \ldots.$$

Therefore, each time $c_t = 1$, we have $g(L^t(s_0, s_1, \ldots, s_{n-1})) = 0$, and many equations of low degree are derived. They can be solved more efficiently than the initial system. The complexity of the standard algebraic attack is roughly $O(D^3)$ in time and $O(D)$ in data, where $D = \sum_{i=0}^{\mathcal{AI}(f)} \binom{n}{i}$.

To resist fast algebraic attacks, a high algebraic immunity is not sufficient. Assume that we can find g of low degree and h of reasonable degree such that $fg = h$. Then

$$h(L^t(s_0, s_1, \ldots, s_{n-1})) = c_t g(L^t(s_0, s_1, \ldots, s_{n-1})), t = 0, 1, 2, \ldots.$$

Then there exists a linear combination of the first $\sum_{i=0}^{\deg(h)} \binom{n}{i}$ equations that sum the left hand side to 0. We find this by using Berlekamp-Massey algorithm or through an explicit algebraic calculation [31]. After summing up we arrive at one equation of degree at most $\deg(g)$. The fast algebraic attack has a pre-computation step of complexity $O(E \log^3 E + En \log^2 n)$ and an online complexity of $O(D^3 + 2DE \log E)$ [31], where $D = \sum_{i=0}^{\mathcal{AI}(f)} \binom{n}{i}$ and $E = \sum_{i=0}^{\deg(h)} \binom{n}{i}$. Note however that fast algebraic attacks need more data than standard ones [7].

Let the sequence $\mathbf{s}$ obey the recursion

$$\sum_{j=0}^{n} m_j s_{t+j} = 0, \quad m_j \in \mathbb{F}_2,$$

where $m_0 = m_n = 1$. Clearly, $m(x) = m_0 + m_1 x + \cdots + m_{n-1} x^{n-1} + x^n$ is its generator polynomial, and it is primitive. The (transpose) companion matrix M (we call it the generator matrix of the sequence) is

$$M = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & & & & \\ 0 & 0 & 0 & \ldots & 1 \\ m_0 & m_1 & m_2 & \ldots & m_{n-1} \end{pmatrix}.$$

Let α be a zero of $m(x)$, which is a primitive element in $\mathbb{F}_{2^n}$.

Let $(s_t, s_{t+1}, \ldots, s_{t+n-1})$ denote the state of the register at time t. Then the next state is determined by $(s_{t+1}, s_{t+2}, \ldots, s_{t+n})^T = M(s_t, s_{t+1}, \ldots, s_{t+n-1})^T = M^{t+1}(s_0, s_1, \ldots, s_{n-1})^T$. Let

$$S = \{(s_t, s_{t+1}, \ldots, s_{t+n-1})^T | t = 0, 1, ..., 2^n - 2\}.$$

We define a multiplication " $*$ " on the set S as follows:

$$M^i(1,0,...,0)^T * M^j(1,0,...,0)^T = M^{i+j}(1,0,...,0)^T.$$

Then S will be a cyclic group of order 2^n-1 and $M(1,0,\ldots,0)^T=(0,\ldots,0,1)^T$ is its generator. Let $\alpha=(0,\ldots,0,1)^T$. Then $\alpha^i=M^i(1,0,\ldots,0)^T$.

We now give some observations related to previous work [16]. Let the filter function be written in univariate representation,

$$f(x)=\sum_{i=0}^{2^n-1} a_i x^i,$$

where $a_i\in\mathbb{F}_{2^n}$, and let the initial state of the register be α^r. Then we have

$$\begin{aligned} f(\alpha^r) &= \sum_{i=0}^{2^n-1} a_i\alpha^{ir}=c_0,\\ f(\alpha^{r+1}) &= \sum_{i=0}^{2^n-1} a_i\alpha^{i(r+1)}=c_1,\\ &\vdots \quad\quad \vdots\\ f(\alpha^{r+2^n-2}) &= \sum_{i=0}^{2^n-1} a_i\alpha^{i(r+2^n-2)}=c_{2^n-2}. \end{aligned}$$

Since $m_0+m_1\alpha+...+m_n\alpha^n=0$, for $0\le i\le n$, multiplying the $(i+1)$th equation by m_i, and summing up the first $n+1$ equations, we cancel the terms $a_1\alpha^{r+j}$, where $0\le j\le n$. In general, the minimal polynomial of α^i is $\prod_{k=0}^{n-1}(x+\alpha^{i\cdot 2^k})$. Using the same method, we can cancel the terms $a_i\alpha^{i(r+j)}$. Let $p(x)=\prod_{wt(l)\ge 2}(x+\alpha^l)\in\mathbb{F}_2[x]$. Then $p(\alpha^l)=0$, for $wt(l)\ge 2$. Therefore, multiplying the $(i+1)$th equation with the coefficient of x^i in $p(x)$, and summing up the first $\sum_{k=2}^{\deg(f)}\binom{n}{k}$ equations, we can cancel all the terms $a_i\alpha^{i(r+j)}$, for $wt(i)\ge 2$, where $0\le j\le\sum_{k=2}^{\deg(f)}\binom{n}{k}$, and a linear equation on α^r is obtained. This comes down to the result of Rϕnjom and Helleseth [16]. Our arguments above are clear and simple, and can be an alternative to Lemma 1, 2 and 3 of [16], which is more complicated.

Furthermore, assume that a fast algebraic attack results in

$$h(L^t(s_0,s_1,\ldots,s_{n-1}))=c_t g(L^t(s_0,s_1,\ldots,s_{n-1})), t=0,1,2,\ldots$$

where $\deg(g)=d$, $\deg(h)=e$ and $e>d$. By summing up equations according to the polynomial $\prod_{wt(l)=d+1}^{e}(x+\alpha^l)$, we can cancel the terms with degree more than d of h, rather than summing up equations that sums the left hand side to 0 using e.g. Berlekamp-Massey algorithm as in the standard fast algebraic attack. The new way of summing equations gives a resulting equation of degree at most d, the same as before, but it requires slightly less keystream symbols. This is a slight improvement of the classical fast algebraic attack.

4 Introducing Higher Order Algebraic Attacks

Assume a filter generator with the generator matrix M and a filter function f with a very high algebraic immunity. Our underlying thoughts are as follows. If we add the first equation and the

ith equation, $i \neq 1$, i.e., we consider $f(x) + f(M^{i-1}x)$, then there may exist some $g \in B_n$ of low degree such that $(f(x) + f(M^{i-1}x))g = 0$ or $(f(x) + f(M^{i-1}x) + 1)g = 0$. This motivates the following definition of higher order algebraic immunity relative to M, and its related definition for arbitrary M.

Definition 1. *Let $f \in B_n$ and M be an $n \times n$ generator matrix for some sequence. The r-order algebraic immunity ($\mathcal{AI}_r$) of f relative to M is equal to*

$$\min_{h \in B_r} AI(h(f(x), f(M^{i_1}x), ..., f(M^{i_{r-1}}x))),$$

where $\deg(h) \geq 1$ *and* $1 \leq i_1 < i_2 < ... < i_{r-1} \leq P < 2^n - 1$*, where P is a secure parameter that can be very large. Furthermore, we define*

$$\mathcal{AI}_r(f) = \max_M \{\mathcal{AI}_r(f) \text{ relative to } M\}.$$

Let us explain the underlying attack if $\mathcal{AI}_r(f)$ is low. For example, if $r = 2$ and $\mathcal{AI}_2(f)$ is low, there exists a $g \in B_n$ of low degree such that $(f(x) + f(M^{i-1}x))g(x) = 0$. Then

$$(f(M^t s) + f(M^{t+i-1}s))g(M^t s) = (c_t + c_{t+i-1})g(M^t s) = 0, t = 0, 1, 2, \ldots$$

Therefore, each time t satisfies $c_t + c_{t+i-1} = 1$, we have $g(M^t s) = 0$, and many equations of low degree are derived, which can be solved more efficiently than the initial system. The time complexity and data complexity of this attack depend on r and $\mathcal{AI}_r(f)$. With a resonable value of r and a small value of $\mathcal{AI}_r(f)$, a very efficient attack can be given.

Remark 1: The classical algebraic attack is the special case $r = 1$, and the Rϕnjom-Helleseth attack is a special case using $r \leq \Sigma_{i=1}^{d} \binom{n}{i}$, where $d = \deg(f)$. Moreover, we can combine the Rϕnjom-Helleseth attack and the classical algebraic attack. For example, using the first $\binom{n}{d}$ equations, we can get a new f' of degree $d - 1$, and $\mathcal{AI}(f')$ might be low (though $\deg(f)$ may be close to n and $\mathcal{AI}(f)$ may be high). This can give an efficient algebraic attack to f'.

Remark 2: The security parameter P used in the definition determines an interval where we can select equations, and this affects the required keystream length. A suitable choice could be either of the same order as the maximum keystream length that is allowed, or close to the number of keys (if no maximum length is given). We do not allow P as large as $2^n - 1$ (the period), as then we have trivial attacks.

To resist higher order algebraic attacks in a specific design, the r-order algebraic immunity of the filter function f relative to the generator matrix M should be high. Otherwise, there would be a $g(x)$ of low degree such that $h(f(x), f(M^{i_1}x), ..., f(M^{i_{r-1}}x)) \cdot g(x) = 0$ or $(h(f(x), f(M^{i_1}x), ..., f(M^{i_{r-1}}x)) + 1) \cdot g(x) = 0$. Also, if $\mathcal{AI}_r(f)$ is low, we can regard f as useless. Sometimes, we denote $\mathcal{AI}_r(f)$ relative to M also with $\mathcal{AI}_r(f)$, assuming no ambiguity.

Fast algebraic attacks can be generalized to a higher order fast algebraic attack, in a similar way. For example, if one can find g of low degree and h of reasonable degree such that $(f(x) + f(M^{i-1}x))g = h$, then similar to the classical fast algebraic attacks, many equations of degree at most $\deg(g)$ can be obtained.

We now give two examples showing the usefulness of this definition.

Example 1: Let the generator polynomial be $m(x) = x^5 + x^2 + 1$, and the filter function be

$$f(x) = x_1x_2x_3x_4 + x_1x_2x_4x_5 + x_1x_2x_5 + x_1x_3x_5 + x_2x_4x_5 + x_1x_3 + x_1x_5 + x_2x_3 + x_2x_5 + x_3x_4 + x_4x_5 + x_2 + 1.$$

It is a 5-variable Carlet-Feng function and has a good immunity to fast algebraic attacks [5], $\deg(f) = 4$, $\mathcal{AI}(f) = 3$ and $nl(f) = 12$. We have

$$\begin{aligned} f(Mx) &= x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_2x_3 + x_1x_2x_4 + x_2x_3x_4 + x_2x_3x_5 \\ &\quad + x_1x_3x_5 + x_1x_2 + x_1x_3 + x_1x_5 + x_2x_4 + x_3x_4 + x_4x_5 + 1, \end{aligned}$$

and

$$\begin{aligned} f(x) + f(Mx) &= x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 \\ &\quad + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_1x_2 + x_2x_3 + x_2x_4 + x_2x_5 + x_2. \end{aligned}$$

Now it can be verified that $(f(x) + f(Mx))(x_2 + 1) = 0$ and hence the second order algebraic immunity is only 1, $\mathcal{AI}_2(f) = 1$.

Example 2: Let the generator polynomial be $m(x) = x^5 + x^2 + 1$, and the filter function be

$$\begin{aligned} f(x) &= x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_3x_4x_5 + x_2x_3x_4x_5 + x_1x_2x_3 + x_1x_2x_4 \\ &\quad + x_1x_2x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5. \end{aligned}$$

It is a 5-variable majority function, $\deg(f) = 4$, $\mathcal{AI}(f) = 3$ and $nl(f) = 10$. Again, we have

$$\begin{aligned} f(Mx) &= x_1x_2x_3x_4 + x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_3x_4x_5 + x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_4 \\ &\quad + x_1x_3x_5 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_4x_5 + x_3x_4x_5 + x_2x_3 + x_3x_4 + x_3x_5, \end{aligned}$$

and

$$f(x) + f(Mx) = x_2x_3x_4x_5 + x_2x_3 + x_3x_4 + x_3x_5.$$

We verify that $(f(x) + f(Mx))(x_3 + 1) = 0$ and $\mathcal{AI}_2(f) = 1$.

5 A 2-order Algebraic Attack on the Carlet-Feng Functions

After the introduction of higher order algebraic attacks we now demonstrate its usefulness by attacking two classes of Boolean functions where previous attack methods were not successful. This section presents powerful attacks on the Carlet-Feng functions.

Let $f \in B_n$ and α a primitive element of the field $\mathbb{F}_{2^n}$. The Carlet-Feng function $f \in B_n$ satisfies $1_f = \{0, 1, \alpha, ..., \alpha^{2^{n-1}-2}\}$. It has optimum algebraic immunity $\lceil \frac{n}{2} \rceil$, optimum algebraic degree $n - 1$, good immunity to fast algebraic attacks and good nonlinearity.

Lemma 1 (*5, Theorem 2*)**.** *The univariate representation of the function f equals*

$$1 + \sum_{i=1}^{2^n-2} \frac{\alpha^i}{(1 + \alpha^i)^{1/2}} x^i,$$

where $u^{1/2} = u^{2^{n-1}}$.

Let M be the generator matrix of the m-sequence of period $2^n - 1$. Recall the cyclic group as defined in Section 3. Let $\beta = (0, ..., 0, 1)^T$, and $\alpha = \beta^s$, i.e., $\alpha = M^s(1, 0, \ldots, 0)^T$, where $1 \le s \le 2^n - 2$ and $(s, 2^n - 1) = 1$. Let the initial state of the register be $\alpha^r = \beta^{rs} = (x_1, x_2, \ldots, x_n)^T$. Then the

state of the register at time s is $\beta^{rs+s} = \alpha^{r+1} = M^s(x_1, x_2, \ldots, x_n)^T$. Then we have the following equations:

$$f(\beta^{rs+j}) = 1 + \sum_{i=1}^{2^n-2} \frac{\alpha^i}{(1+\alpha^i)^{1/2}} (\beta^{rs+j})^i = c_j,$$

where $j = 0, 1, 2, \ldots$.

Theorem 1. *Let the security parameter P of the definition be at least 2^{n-1}. Then the 2-order algebraic immunity of f is equal to 1.*

Proof. Let

$$g(\alpha^r) = 1 + a\alpha^{r+1} + a^2\alpha^{2(r+1)} + \cdots + a^{2^{n-1}}\alpha^{2^{n-1}(r+1)},$$

where $tr(a) = tr(a^2\alpha^{-1}) = 1$. By Lemma 1, we have

$$f(\alpha^r) + f(\alpha^{r+1}) = \sum_{i=1}^{2^n-2} \frac{\alpha^i}{(1+\alpha^i)^{1/2}} \alpha^{ir}(1+\alpha^i) = \sum_{i=1}^{2^n-2} \alpha^{i+ir}(1+\alpha^{i\cdot 2^{n-1}}).$$

Since

$$\begin{aligned} tr(a) &= tr(a^2\alpha^{-1}) = 1, \\ \sum_{i=1}^{2^n-2} \alpha^{i+ir} \cdot \alpha^{1+r} &= \sum_{i=1}^{2^n-2} \alpha^{i+ir} + 1 + \alpha^{1+r}, \end{aligned}$$

and

$$\sum_{i=1}^{2^n-2} \alpha^{i+ir+i\cdot 2^{n-1}} \cdot \alpha^{3+2r} = \sum_{i=1}^{2^n-2} \alpha^{i+ir+i\cdot 2^{n-1}} + 1 + \alpha^{3+2r},$$

we have

$$\sum_{i=1}^{2^n-2} \alpha^{i+ir} \cdot g(\alpha^r) = 1 + tr(a\alpha^{r+1}),$$

and

$$\sum_{i=1}^{2^n-2} \alpha^{i+ir+i\cdot 2^{n-1}} \cdot g(\alpha^r) = 1 + tr(a^2\alpha^{2(r+1)}).$$

Therefore, $(f(\alpha^r) + f(\alpha^{r+1})) \cdot g(\alpha^r) = 0$, and the result follows. *QED*

Note that for any generator matrix M, the 2-order algebraic immunity of f relative to M is equal to 1.

The number of $g(\alpha^r)$ satisfying $tr(a) = tr(a^2\alpha^{-1}) = 1$ is equal to 2^{n-2}. Therefore, we can get 2^{n-2} linear equations if $c_t + c_{t+s} \neq 0$. The solution of these equations is $\beta^{rs+t} = \alpha^{-1} = \beta^{-s}$ or $\beta^{rs+t} = \alpha^{2^{n-1}-2} = \beta^{(2^{n-1}-2)s}$, and the exact value of β^{rs+t} can be determined from c_t:

$$\beta^{rs+t} = \begin{cases} \beta^{-s}, & \text{if } c_t = 0, \\ \beta^{(2^{n-1}-2)s}, & \text{if } c_t = 1. \end{cases}$$

The attack is then described as follows.

Algorithm for the attack.
Given a filter generator, β and $\alpha = \beta^s$ are known.
1. Pre-compute s, where $\alpha = \beta^s$.
2. Find the first t such that $c_t + c_{t+s} \neq 0$.
3. Determine the value of β^{rs+t} from c_t, which is the state of the filter generator at time t:

$$\beta^{rs+t} = \begin{cases} \beta^{-s}, & \text{if } c_t = 0, \\ \beta^{(2^{n-1}-2)s}, & \text{if } c_t = 1. \end{cases}$$

We note that this attack can be very efficient, and the state even can be recovered by hand, after a pre-computation with complexity corresponding to the DLP in $\mathbb{F}_{2^n}$. In the best cases, this attack needs only two bits of keystream c_t and c_{t+s} on the condition that $s = 1$ or $2^n - 2$ and $c_t + c_{t+s} \neq 0$. In the worst cases, this attack needs about 2^{n-1} bits of keystream ($s = 2^{n-1}$ or $s = 2^{n-1} - 1$).

For a filter generator using f as the filter function, the time complexity of the Rϕnjon-Helleseth attack is roughly $O(2^n)$, after a pre-computation with complexity $O(n^3 \cdot 2^n)$. and it needs almost all bits of keystream. The time complexity of the classical algebraic attack is roughly $O(2^{3(n-1)})$, and it needs about $\sum_{i=1}^{\lceil \frac{n}{2} \rceil} \binom{n}{i}$ bits of keystream. No nonzero function g and no function h were observed such that $fg = h$, when $\deg(g) + \deg(h) < n - 2$ [5]. If this is true for all n, then the time complexity and keystream needed of the fast algebraic attack is not much better than the classical algebraic attack and the Rϕnjon-Helleseth attack.

Example 3: Let the generator polynomial

$$m(x) = x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^6 + x^2 + x + 1,$$

and the filter function $f(x)$ be the 16-variable Carlet-Feng function with $\alpha = \beta^2$, where $\beta = (0, ..., 0, 1)^T$. Let the m-sequence be

$$(00110110000000000000000100111...).$$

Then the first 6 bits of keystream is $(c_0, c_1, \ldots, c_5) = (010100)$.

Suppose we receive the first 6 bits of the keystream. Clearly, $s = 2$ and $c_t + c_{t+s} = 1$ when $t = 3$. Therefore we can determine the value of β^{2r+3} from $c_3 = 1$:

$$\beta^{2r+3} = \beta^{2(2^{n-1}-2)} = \beta^{-3} = (1011000000000000)^T,$$

which is the state of the register at time $t = 3$.

As a comparison, the classical algebraic attack needs at least $2^{15} + \frac{1}{2}\binom{16}{8} - 1$ bits of keystream, and the Rϕnjom-Helleseth attack needs $2^{16} - 3$ bits of keystream. The keystream needed in our attack here is only 6 bits.

The time complexity of the classical algebraic attack is roughly $O(2^{45})$, and the Rϕnjom-Helleseth attack has complexity roughly $O(2^{16})$. For any Carlet-Feng function of n-variable, it was conjectured in [5] that no nonzero function g and no function h exist such that $fg = h$, when $(\deg(g), \deg(h)) = (1, n-3)$ for n even. If this is true, then the time complexity of the fast algebraic attack is roughly $O(2^{16})$ and it needs about $2^{16} - 19$ bits of keystream. Our attack can recover the state by hand very easily.

6 A 2-order Algebraic Attack on Rotation Symmetric Boolean Functions

The second common class of Boolean functions we attack is rotation symmetric Boolean functions.

Define (see [18])

$$\rho_n^k(x_i) = \begin{cases} x_{i+k}, & \text{if } i+k \le n, \\ x_{i+k-n}, & \text{if } i+k > n. \end{cases}$$

Let $x = (x_1, \ldots, x_n)^T$. We can extend the definition of ρ_n^k on tuples as follows:

$$\rho_n^k(x) = (\rho_n^k(x_1), \ldots, \rho_n^k(x_n))^T.$$

An n-variable Boolean function f is called rotation symmetric Boolean function (RSBF) if for each input $x \in \mathbb{F}_2^n$, $f(\rho_n^i(x)) = f(x)$ for $1 \le i \le n-1$.

Theorem 2. *Let f be an n-variable rotation symmetric Boolean function. Then the 2-order algebraic immunity of f is equal to 1.*

Proof. For any filter generator, let $m(x) = m_0 + m_1 x + ... + m_{n-1}x^{n-1} + x^n$ be its generator polynomial, and M be its generator matrix. We have

$$\begin{aligned} f(Mx) &= f(x_2, \ldots, x_n, x_1 + m_1 x_2 + \cdots + m_{n-1}x_n) \\ &= f(x_2, \ldots, x_n, x_1) + (m_1 x_2 + \cdots + m_{n-1}x_n)(f(x_2, \ldots, x_n, 0) + f(x_2, \ldots, x_n, 1)) \\ &= f(x_1, x_2, \ldots, x_n) + (m_1 x_2 + \ldots + m_{n-1}x_n)(f(0, x_2, \ldots, x_n) + f(1, x_2, \ldots, x_n)). \end{aligned}$$

Therefore,

$$f(x) + f(Mx) = (m_1 x_2 + \cdots + m_{n-1}x_n)(f(0, x_2, \ldots, x_n) + f(1, x_2, \ldots, x_n)),$$

and $m_1 x_2 + \cdots + m_{n-1}x_n + 1$ is an annihilator of it. *QED*

Again, note that for any generator matrix M, the 2-order algebraic immunity of f relative to M is equal to 1.

If $c_0 + c_1 \neq 0$, we get $m_1 x_2 + \cdots + m_{n-1}x_n = 1$ and $f(0, x_2, \ldots, x_n) + f(1, x_2, \ldots, x_n) = 1$. From the latter equation, the value of x_1 can be obtained in many cases. In general, if $c_t + c_{t+1} \neq 0$, the linear equations can easily be obtained from the generator polynomial. Particularly, for the majority function [12, 20]:

$$f(x_1, x_2, ..., x_n) = \begin{cases} 0, & \text{if } wt(x) < \lceil \frac{n}{2} \rceil, \\ 1, & \text{otherwise.} \end{cases}$$

If $c_0 + c_1 \neq 0$, we have $f(0, x_2, \ldots, x_n) = 0$ and $f(1, x_2, \ldots, x_n) = 1$. Therefore, we can get the following two linear equations: $m_1 x_2 + \ldots + m_{n-1}x_n = 1$ and

$$x_1 = \begin{cases} 0, & \text{if } f(x_1, x_2, ..., x_n) = 0, \\ 1, & \text{if } f(x_1, x_2, \ldots, x_n) = 1. \end{cases}$$

Algorithm for the attack.
1. Find some t such that $c_t + c_{t+1} \neq 0$.
2. Deduce some linear equations at these time t, which can easily be obtained from the generator polynomial.
3. Solve these linear equations.

This attack is very efficient, and the time complexity of the attack is roughly $O(n^3)$. While the classical algebraic attacks, the Rϕnjom-Helleseth attack and the fast algebraic attacks need exponential time on n in most cases. Moreover, this attack needs very few bits of keystream, which can be seen from the following examples (one example with a RSBF will be given in the Appendix and here we provide an example with a symmetric Boolean function).

Example 4: Let the generator polynomial $m(x) = x^8 + x^7 + x^6 + x + 1$, the initial state be $(00000001)^T$, and the filter function $f(x) = \sigma_1^{(8)} + \sigma_3^{(8)} + \sigma_4^{(8)} + \sigma_6^{(8)} + \sigma_8^{(8)}$ ($\deg(f) = 8$, $\mathcal{AI}(f) = 4$ and $nl(f) = 70$ [1]), where $\sigma_i^{(8)}$ denotes 8-variable homogeneous symmetric Boolean function which contains of all terms of degree i, for $i = 1, 3, 4, 6, 8$. Then the sequence is

$$(00000001101101010001001011110 0...)$$

and the first 18 bits of keystream is (100111000111110101).
Suppose we receive the first 18 bits of keystream. Then we have

$$\begin{aligned}
&x_2 + x_7 + x_8 = 1 \text{ (from } c_0 + c_1 = 1),\\
&x_2 + x_3 + x_4 + x_8 = 1 \text{ (from } c_2 + c_3 = 1),\\
&x_2 + x_4 + x_5 + x_6 + x_7 + x_8 = 1 \text{ (from } c_5 + c_6 = 1).\\
&x_3 + x_4 + x_5 + x_7 + x_8 = 1 \text{ (from } c_8 + c_9 = 1).\\
&x_3 + x_4 + x_5 + x_6 + x_8 = 1 \text{ (from } c_{13} + c_{14} = 1).\\
&x_1 + x_2 + x_4 + x_5 + x_6 + x_8 = 1 \text{ (from } c_{14} + c_{15} = 1).\\
&x_1 + x_3 + x_5 + x_6 + x_8 = 1 \text{ (from } c_{15} + c_{16} = 1).\\
&x_1 + x_4 + x_6 + x_8 = 1 \text{ (from } c_{16} + c_{17} = 1).
\end{aligned}$$

Solving these equations, we recover the initial state $(00000001)^T$.

In comparison, the classical algebraic attack would need $\sum_{i=1}^{4} \binom{8}{i} = 162$ bits of keystream, the fast algebraic attacks need more than 162 bits of keystream and the Rϕnjom-Helleseth attack needs about $\sum_{i=1}^{8} \binom{8}{i} = 255$ bits of keystream. The keystream needed in our attack is only 18 bits.

To recover the initial state, the classical algebraic attack needs to solve the linear equations in 162 variables, the Rϕnjom-Helleseth attack and the fast algebraic attacks need many calculations to get the linear equations in 8 variables. Our attack can get linear equations in 8 variables easily.

Ending this section, we consider the case when the register is of length n and f is an l-variable RSBF. If there are j gaps, i.e., the filter function is

$$f'(x) = f(x_1, \ldots, x_{k_1}, x_{k_1+r_1}, \ldots, x_{k_2}, x_{k_2+r_2}, \ldots, x_{k_j}, x_{k_j+r_j}, \ldots, x_n),$$

then we have the following.

Theorem 3. *The 2-order algebraic immunity of f' relative to M is at most $j + 1$.*

Proof.

$$\begin{aligned}
f'(Mx) &= f(x_2, \ldots, x_{k_1+1}, x_{k_1+r_1+1}, \ldots, x_{k_j+1}, x_{k_j+r_j+1}, \ldots, x_n, x_1 + h(x))\\
&= f(x_1, \ldots, x_{k_1+1}, x_{k_1+r_1+1}, \ldots, x_{k_j+1}, x_{k_j+r_j+1}, \ldots, x_n) + h(x)g(x),
\end{aligned}$$

where $h(x) = m_1x_2 + \cdots + m_{n-1}x_n$ and $g(x)$ is an $(l-1)$-variable function. Therefore,

$$\begin{aligned} f'(x) + f'(Mx) &= f_1 \cdot (x_{k_1+1} + x_{k_1+r_1}) + f_2 \cdot (x_{k_2+1} + x_{k_2+r_2}) + \cdots + f_j \cdot (x_{k_j+1} + x_{k_j+r_j}) \\ &\quad + f_{12} \cdot (x_{k_1+1}x_{k_2+1} + x_{k_1+r_1}x_{k_2+r_2}) + \cdots + f_{j-1,j} \cdot (x_{k_{j-1}+1}x_{k_j+1} + x_{k_{j-1}+r_{j-1}}x_{k_j+k_j}) \\ &\quad + \ldots \\ &\quad + f_{12\ldots j} \cdot (x_{k_1+1} \cdots x_{k_j+1} + x_{k_1+r_1} \cdots x_{k_j+r_j}) + h(x)g(x), \end{aligned}$$

where f_i are $(l-j)$-variable functions on $x_1, \ldots, x_{k_1}, x_{k_1+r_1+1}, \ldots, x_{k_j}, x_{k_j+r_j+1}, \ldots, x_n$. Clearly,

$$(h(x)+1)\prod_{i=1}^{j}(x_{k_i+1} + x_{k_i+r_i} + 1)$$

is an annihilator of $f'(x) + f'(Mx)$, and the result follows. *QED*

There are many RSBFs $f \in B_l$ with optimum algebraic immunity $\lceil \frac{l}{2} \rceil$. However, consider the first two equations, we can get an equation of low degree if the number of gaps is small. To resist the 2-order algebraic attack, there should be enough gaps (at least close to $\lceil \frac{l}{2} \rceil$). Moreover, these gaps should satisfy some conditions to ensure that $\mathcal{AI}(f'(x) + f'(M^i x))$ are all high, for $i \geq 1$. However, even if $\mathcal{AI}_2(f'(x))$ is high, there are likely to be a small r such that $\mathcal{AI}_r(f'(x))$ is low. It seems to be hard to construct secure stream ciphers using RSBFs as filter functions.

7 Conclusion

In this paper, we introduced higher order algebraic attacks. In particular, we computed the 2-order algebraic immunity of the Carlet-Feng functions and the rotation symmetric Boolean functions which is equal to 1. This led to very efficient attacks on filter generators using functions from these classes, that greatly outmatch all previously known attacks. In fact, many other functions with good cryptographic properties may be also vulnerable to high order algebraic attacks. It is left as an open problem to investigate higher order algebraic attacks on other classes of Boolean functions. In particular, functions that use a subset of the state variables, determined e.g. by a difference set.

To construct a secure stream cipher, higher order algebraic immunities of Boolean functions must be considered.

References

[1] A. Braeken, *Cryptographic properties of Boolean functions and S-boxes*, PhD thesis, Katholieke University, 2006, Available: http://www.cosic.esat.kuleuven.be/publications/thesis-129.pdf

[2] A. Braeken and B. Preneel, *On the algebraic immunity of symmetric Boolean functions*, Indocrypt 2004, volume 3797 of Lecture Notes in Computer Science, pages 35-48. Springer-Verlag, 2005.

[3] A. Canteaut and M. Videau, *Symmetric Boolean functions*, IEEE Transactions on Information Theory, 2005, 51(8): 2791-2811.

[4] C. Carlet, D. K. Dalai, K. C. Gupta and S. Maitra, *Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction*, IEEE Transactions on Information Theory, 2006, 52(7): 3105-3120.

[5] C. Carlet and K. Feng, *An Infinite Class of Balanced Functions with Optimal Algebraic Immunity, Good Immunity to Fast Algebraic Attacks and Good Nonlinearity*, ASIACRYPT 2008, volume 5350 of Lecture Notes in Computer Science, pages 425-440, Springer-Verlag, 2008.

[6] C. Carlet and K. Feng, *An Infinite Class of Balanced Vectorial Boolean Functions with Optimum Algebraic Immunity and Good Nonlinearity*, IWCC 2009, volume 5557 of Lecture Notes in Computer Science, pages 1-11, Springer-Verlag, 2009.

[7] N. Courtois, *Fast Algebraic attacks on stream ciphers with linear feedback*, CRYPTO 2003, volume 2729 of Lecture Notes in Computer Science, pages 176-194. Springer-Verlag, 2003.

[8] N. Courtois and G. Bard, *Algebraic Cryptanalysis of the Data Encryption Standard*, IMA Int. Conf, volume 4887 of Lecture Notes in Computer Science, pages 152-169. Springer-Verlag, 2007.

[9] N. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, EUROCRYPT 2003, volume 2656 of Lecture Notes in Computer Science, pages 345-359. Springer-Verlag, 2003.

[10] N. Courtois and J. Pieprzyk, *cryptanalysis of block ciphers with overdefined systems of equations*, ASIACRYPT 2002, volume 2501 of Lecture Notes in Computer Science, pages 267-287. Springer-Verlag, 2002.

[11] D. K. Dalai, K. C. Maitra and S. Maitra, *Cryptographically significant Boolean functions: Construction and analysis in terms of algebraic immunity*, FSE 2005, volume 3557 of Lecture Notes in Computer Science, pages 98-111. Springer-Verlag, 2005.

[12] D. K. Dalai, S. Maitra and S. Sarkar, *Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity*, Des. Codes Cryptogr. 2006, 40(1), 41-58.

[13] S. Fischer, W. Meier, *Algebraic Immunity of S-Boxes and Augmented Functions*, FSE 2007, volume 4593 of Lecture Notes in Computer Science, pages 366-381. Springer-Verlag, 2007.

[14] N. Li, W. Qi, *Construction of Boolean functions with maximum possible annihilator immunity*, ASIACRYPT 2006, volume 4284 of Lecture Notes in Computer Science, pages 84-98. Springer-Verlag, 2006.

[15] E. Pasalic, *Almost Fully Optimized Infinite Classes of Boolean Functions Resistant to (Fast) Algebraic Cryptanalysis*, ICISC 2008, volume 5461 of Lecture Notes in Computer Science, pages 399-414. Springer-Verlag, 2009.

[16] S. Rϕnjom and T. Helleseth, *A New Attack on the Filter Generator*, IEEE Transactions on Information Theory, 2007, 53(5): 1752-1758.

[17] I. Wegener, *The Complexity of Boolean Functions*, New York: Wiley, 1987.

[18] P. Stănică and S. Maitra, *Rotation symmetric Boolean functions-count and cryptographic properties*, Discrete Mathematics and Applications, 2008, 156(10), 1567-1580.

[19] S. Sarkar and S. Maitra, *Construction of Rotation Symmetric Boolean Functions on Odd Number of Variables with Maximum Algebraic Immunity*, AAECC 2007, volume 4851 of Lecture Notes in Computer Science, pages 271-280. Springer-Verlag, 2007.

[20] F. Armknecht, C. Carlet, P. Gaborit, S. Künzli, W. Meier and O. Ruatta, *Efficient Computation of Algebraic Immunity for Algebraic and Fast Algebraic Attacks*, EUROCRYPT 2006, volume 4004 of Lecture Notes in Computer Science, pages 147-164. Springer-Verlag, 2006.

[21] J. Pieprzyk and C. X. Qu, *Fast Hashing and Rotation-Symmetric Functions*, Journal of Universal Computer Science, 1999, 5(1): 20-31.

[22] P. Stănică, S. Maitra and L. Clark, *Results on Rotation Symmetric Bent and Correlation immune Boolean functions*, FSE 2004, volume 4004 of Lecture Notes in Computer Science, pages 161-177. Springer-Verlag, 2004.

[23] A. Maximov, M. Hell and S. Maitra, *Plateaued rotation symmetric boolean functions on odd number of variables*, First Workshop on Boolean Functions: Cryptography and Applications, pages 83-104. Rouen, France, 2005.

[24] S. Kavut, S. Maitra and M.D. Yücel, *Search for Boolean functions with excellent profiles in the rotation symmetric class*, IEEE Transactions on Information Theory, 2007, 53(8): 1743-1751.

[25] N. Li and W. Qi: *Symmetric Boolean functions depending on an odd number of variables with maximum algebraic immunity*, IEEE Transactions on Information Theory, 2006, 52(5): 2271-2273.

[26] L. Qu, C. Li and K. Feng, *A note on symmetric Boolean functions with maximum algebraic immunity in odd number of variables* IEEE Transactions on Information Theory, 2007, 53(8): 2908-2910.

[27] J. Clark, J. Jacob, S. Maitra and P. Stănică, *Almost Boolean Functions: The Design of Boolean functions by Spectral Inversion*, CEC 2003, volume 3 in the proceedings, pages 2173-2180, IEEE Press, 2003.

[28] E. Filiol and C. Fontaine, *Highly nonlinear balanced Boolean Functions with a good correlation-immunity*, EUROCRYPT'98, volume 1403 of Lecture Notes in Computer Science, pages 475-488. Springer-Verlag, 1998.

[29] H. Kim, S. Park and S. G. hahn *On the weight and nonlinearity of homogeneous rotation symmetric Boolean functions of degree 2*, Discrete Applied Mathematics, 2009, 157(2): 428-432.

[30] T. W. Cusick and P. Stănică, *Fast Evaluation, Weights and Nonlinearity of Rotation-Symmetric Functions*, Discrete Mathematics, 2002, 258(1-3): 289-301.

[31] P. Hawkes and G. G. Rose, *Rewriting Variables: The Complexity of Fast Algebraic Attacks on Stream Ciphers*, CRYPTO 2004, volume 3152 of Lecture Notes in Computer Science, pages 390-406. Springer-Verlag, 2004.

[32] F. Armknecht and M. Krause, *Algebraic Attacks on Combiners with Memory*, CRYPTO 2003, volume 2729 of Lecture Notes in Computer Science, pages 162-176. Springer-Verlag, 2003.

[33] Q. Wang, J. Peng, H. Kan and X. Xue, *Constructions of Cryptographically Significant Boolean Functions Using Primitive Polynomials*, IEEE Transactions on Information Theory, 2010, 56(6): 3048-3053.

8 Appendix

Example 5: Let the generator polynomial $m(x) = x^5 + x^2 + 1$ and the initial state be $(00110)^T$. Let

$$G(x) = \begin{cases} 1, & \text{if } wt(x) \leq 2, \\ 0, & \text{if } wt(x) > 2, \end{cases}$$

$$\begin{aligned} O_1 &= \{(1,0,0,1,0),(0,1,0,0,1),(1,0,1,0,0),(0,1,0,1,0),(0,0,1,0,1)\} \\ O_2 &= \{(1,0,0,1,1),(1,1,0,0,1),(1,1,1,0,0),(0,1,1,1,0),(0,0,1,1,1)\}, \end{aligned}$$

and the filter function

$$R(x) = \begin{cases} G(x)+1, & \text{if } x \in O_1 \cup O_2, \\ G(x), & \text{otherwise.} \end{cases}$$

$R(x)$ is a RSBF with $\deg(f) = 4$, $\mathcal{AI}(f) = 3$ and $nl(f) = 12$ (see [19]). Then a full period of the sequence is

$$(11100011011101010000010010110011)$$

and the first 16 bits of keystream is

$$(c_0, c_1, ..., c_{15}) = (1000100000111110).$$

In fact, let the initial state be $(x_1, x_2, x_3, x_4, x_5)^T$. Then the sequence is

$$(x_1, x_2, x_3, x_4, x_5, x_1+x_3, x_2+x_4, x_3+x_5, x_1+x_3+x_4, x_2+x_4+x_5, ...).$$

Suppose we receive the first 16 bits of keystream. Then we have

$$\begin{aligned} &x_3 = 1 \text{ (from } c_0 + c_1 = 1), \\ &x_1 + x_3 = 1 \text{ (from } c_3 + c_4 = 1), \\ &x_2 + x_4 = 1 \text{ (from } c_4 + c_5 = 1), \\ &x_1 + x_2 + x_3 = 1 \text{ (from } c_9 + c_{10} = 1), \\ &x_1 + x_2 + x_4 + x_5 = 1 \text{ (from } c_{14} + c_{15} = 1). \end{aligned}$$

Solving these equations, we recover the initial state $(x_1, x_2, x_3, x_4, x_5)^T = (00110)^T$.

In comparison, the classical algebraic attack would need $\sum_{i=1}^{3} \binom{5}{i} = 25$ bits of keystream, the fast algebraic attacks need more than 25 bits of keystream and the Rϕnjom-Helleseth attack needs about $\sum_{i=1}^{4} \binom{5}{i} = 30$ bits of keystream. The keystream needed in our attack is much less.

To recover the initial state, the classical algebraic attack needs to solve the linear equations in 25 variables, the Rϕnjom-Helleseth attack and the fast algebraic attacks need many calculations to get the linear equations in 5 variables. Our attack can get linear equations in 5 variables easily, and is much more efficient than theirs.