Algebraic Attack on LFSR-Based Multi-Output Stream Ciphers and Research on Algebraic Immunity for Multi-Output Boolean Functions

Xiao Zhong[1,2] and Mingsheng Wang[3]

1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
2. Graduate School of Chinese Academy of Sciences, Beijing 100190, China
3. State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing, China
zhongxiao456@163.com
mingsheng_wang@yahoo.com.cn

Abstract. This paper focuses on the LFSR-based multi-output stream ciphers consisting of a linear feedback shift register (LFSR) and a multi-output filter boolean function. Our contribution is twofold. First, to fill in the blank, for the first time, we propose a general algebraic attack framework on the multi-output stream ciphers, put forward four attack scenarios and explain how to use the low degree multiples. Moreover, we mount our method on the LFSR-based stream ciphers of single-output boolean function with optimum algebraic immunity via the augmented function to transform the model into multi-output. Using our approach, we identify equation system with degree less than the optimum algebraic immunity of the single-output boolean function, which suggests a better way to apply algebraic attack on this kind of model.

Our second contribution involves providing novel definitions to give the algebraic descriptions of the algebraic immunity for multi-output boolean functions. We demonstrate that our definitions are equivalent to the definition presented by Armknecht and Krause, while being more convenient and understandable. Armed with our new description, it is easier to derive the low degree equation system when algebraic attack is applied on multi-output case. Finally, we study the general properties and various characterizations of the algebraic immunity of multi-ouput boolean functions and obtain some meaningful results.

Key words: Algebraic attacks, Stream Ciphers, Multi-output boolean functions, Single-output boolean functions, Algebraic immunity, Augmented functions.

1 Introduction

Algebraic attacks are kinds of efficient methods on stream ciphers which adapt LFSR-based keystream generators with nonlinear filter generator or the combiner as one of their components. The basic idea is to recover the initial state of the LFSR by solving a system of nonlinear equations of low degree. Courtois and Meier made a major breakthrough in algebraic attacks on stream ciphers [1,2]. They proposed a powerful method to obtain low degree equations in the initial state bits by multiplying the filter function f by a well chosen boolean function g. Hence given some keystream bits, the attack tries to recover the initial state by solving systems of the overdefined polynomial equations. Further, the concept of algebraic immunity(AI) for boolean functions was introduced in [4] as a measure for stream ciphers based on linear feedbacks against algebraic attacks. In general it should be as high as possible and the maximum is $\lceil \frac{m}{2} \rceil$ where m is the number of variables of boolean functions.

Since then, a large body of literature has emerged on the field of algebraic immunity (AI) [5,6,7,8,9]. Carlet. C and Feng. K proposed an infinite class of balanced boolean functions with optimum algebraic immunity [10]. This class of functions also achieves optimum algebraic degree and a much better nonlinearity than all the previously obtained infinite classes of functions.

On the other hand, with the development of the stream ciphers, some modern stream ciphers are multi-output, the encryption and decryption speed of which is much faster than the single-output ones. There are two basic designs for such stream ciphers based on the linear feedback shift register (LFSR) [16]. One kind is the filter function generator where m bits are extracted from one LFSR as inputs to the filter function $f(x)$ to produce n-bit ($n \geq 1$) keystream. The other one is the combiner generator consisting of m LFSRs and a multi-output boolean function $f(x)$. One bit is trapped from each LFSR as an input bit for $f(x)$ to produce n-bit keystream each clock.

Similar with the single-output stream ciphers, it is necessary to consider the algebraic attack on the multi-output case. In this paper, we propose a general algebraic attack framework on the multi-output stream ciphers, put forward four attack scenarios and explain how to use the low degree multiples by listing Case 1, Case 2 and Case 3 systematically. To the best of our knowledge, it is the first time that we give a general method to apply the algebraic attack on multi-output case by using the attack scenarios given in advance. Moreover, we mount our method on the LFSR-based single-output stream ciphers by using augmented function to transform them into multi-output case. Several examples show that we can get equation system of degree less than the optimum algebraic immunity of the filter functions.

Armknecht and Krause presented a definition of the algebraic immunity for multi-output boolean functions in [11], and investigated some construction methods of multi-output boolean functions with maximal algebraic immunity. In this paper, based on the algebraic attack scenarios mentioned before, we promote novel definitions to give the algebraic description for the algebraic immunity of multi-output boolean functions, and we prove that the new definitions are equivalent to the definition in [11]. While the new definitions are more convenient and understandable for us to derive the low degree equation system when we apply algebraic attack on multi-output case. We study the general properties and various possible characterizations of the algebraic immunity of multi-output boolean functions, and get some meaningful results.

The paper is organized as follows: In Section 2, we present some preliminaries which will be used throughout this paper. We promote a general framework for the algebraic attack on the LFSR-based multi-output stream ciphers in Section 3, give four attack scenarios and explain how to use the low degree multiples. We also give an example to show how to use our method on the LFSR-based multi-output stream ciphers. In Section 4, we promote new definitions to give the algebraic description for the algebraic immunity of the multi-output boolean functions, and rigorously prove that the new definitions are equivalent to the definition of algebraic immunity of the multi-output boolean functions. Furthermore, we give some meaningful theoretical results on the algebraic immunity of the multi-output boolean functions. In Section 5, we extend our attack proposed in Section 3 to the LFSR-based single-output stream ciphers by using augmented function, and get better results than the conventional algebraic attack presented in [1]. Section 6 gives the conclusion for this paper.

2 Preliminaries

A boolean function on m variables is a mapping from $\mathbb{F}_2^m$ into $\mathbb{F}_2$. We denote the ring of boolean functions in m variables by $\mathbb{B}_m$. Let $X_1, \ldots, X_m$ be m indeterminates, then we may represent $\mathbb{B}_m$ as $\mathbb{B}_m = \mathbb{F}_2[X_1, \ldots, X_m]/I$, where $I = \text{Ideal}(X_1^2 - X_1, \ldots, X_m^2 - X_m)$ is the ideal generated by $X_1^2 - X_1, \ldots, X_m^2 - X_m$ in $\mathbb{F}_2[X_1, \ldots, X_m]$.

Hence every $h \in \mathbb{B}_m$ can be represented as $h = h_1 \bmod I$, where $h_1 \in \mathbb{F}_2[X_1, \ldots, X_m]$. We denote $X_i \bmod I$ by x_i, thus an element in $\mathbb{B}_m$ is denoted by $h(x_1, \ldots, x_m)$, while its corresponding element in $\mathbb{F}_2[X_1, \ldots, X_m]$ is denoted by $h(X_1, \ldots, X_m)$.

Thus every boolean function f of m variables may be written as

$$f(x_1, \ldots, x_m) = \bigoplus_{I \subseteq \{1, \ldots, n\}} a_I \Pi_{i \in I} x_i.$$

Where $a_I \in \mathbb{F}_2$. The terms $\Pi_{i \in I} x_i$ are called monomials. The algebraic degree $\deg(f)$ of a boolean function f equals to the maximal degree of those monomials with nonzero coefficients.

Let $f \in \mathbb{B}_m$, we would like to introduce the following notations that will be used throughout the paper.

(i) For $g = (g_1, \ldots, g_n) \in \mathbb{B}_m^n$, $f = (f_1, \ldots, f_n) \in \mathbb{B}_m^n$, $g \circ f$ denotes $\sum_{i=1}^n g_i f_i$, which is in $\mathbb{B}_m$. In particular, for $c = (c_1, \ldots, c_n) \in \mathbb{F}_2^n$, $c \circ g = \sum_{i=1}^n c_i g_i$.

(ii) Let $b = (b_1, \ldots, b_n) \in \mathbb{F}_2^n$, and $f = (f_1, \ldots, f_n) \in \mathbb{B}_m^n$, $\text{Ideal}(f + b)$ denotes the ideal generated by $f_1 + b_1, \ldots, f_n + b_n$ in $\mathbb{B}_m$, and $\text{Ann}(f + b) = \{g \in \mathbb{B}_m^n | g \circ (f + b) = \sum_{i=1}^n (f_i + b_i) g_i = 0\}$.

(iii) For a nonzero $g = (g_1, \ldots, g_n) \in \mathbb{B}_m^n$, define $\deg_1(g) = \max_{0 \neq c \in \mathbb{F}_2^n} \{\deg(c \circ g) | c \circ g \neq 0\}$, $\deg_2(g) = \min_{0 \neq c \in \mathbb{F}_2^n} \{\deg(c \circ g) | c \circ g \neq 0\}$. It is easy to know $\deg_1(g) = \max_i \{\deg(g_i) | g \neq 0\}$.

3 Algebraic Attack on LFSR-Based Multi-Output Stream Ciphers

Courtois and Meier [1,2] presented algebraic attack on stream ciphers with LFSR, and they focused on the single-output case. They observed that for a boolean function $f \in \mathbb{B}_m$, the following cases may arise:

(1) There exists a low degree $0 \neq g \in B_m$ such that $h = fg \neq 0$ is of low degree.

(2) There exists a low degree $0 \neq g \in B_m$ such that $h = fg = 0$.

(3) There exists $g \in B_m$ such that $h = fg \neq 0$ is of low degree.

The above cases can be used to mount algebraic attack on LFSR-based stream ciphers, such as LiLi-128, Toyocrypt, and E_0 [1,3]. The algebraic immunity of boolean functions quantifies the resistance to the algebraic attack of stream ciphers based on LFSR filtered by a boolean function. Let us recall the following important definition:

Definition 1. *[4] Let $f \in \mathbb{B}_m$, $AI(f) = \min\{deg(g) | g \neq 0, gf = 0 \text{ or } g(f+1) = 0\}$ is called algebraic immunity of f.*

By Courtois and Meier's theorem [1], $AI(f) \leq \lceil \frac{m}{2} \rceil$. In general $AI(f)$ should be as large as possible in order to resist algebraic attack.

Now we extend the definition of algebraic immunity to multi-output boolean functions. Let S be a subset of $\mathbb{F}_2^m$, define $I(S) = \{g \in \mathbb{B}_m | g(s) = 0, \forall s \in S\}$.

Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, Armknecht and Krause [11] introduced the following definition:

Definition 2. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$. $AI(f) = \min\{deg(g) | 0 \neq g \in I(f^{-1}(a)), a \in \mathbb{F}_2^n\}$ is called algebraic immunity of f.*

Let d be the least integer such that $\sum_{i=0}^{d} \binom{m}{d} > 2^{m-n}$, then $AI(f) \leq d$ [11]. Feng. K etc. proved that d can be reached by suitable boolean functions [9].

In this section, we will give a detailed observation on the filter function of LFSR-based multi-output stream ciphers and explore the algebraic attack scenarios for the multi-output case, which has not been done before.

Similar with the conventional algebraic attack in [1], we consider only synchronous stream ciphers. The target cipher systems are multi-output stream ciphers.

Let the length of the linear feedback shift register be m. L is the "connection function" of the LFSR, and it is linear. The LFSR generator polynomial is a primitive polynomial $p(x) = p_0 + p_1 x + ... + p_{m-1} x^{m-1} + x^m$. It generates an m-sequence which is filtered by a nonlinear multi-output boolean function $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$. Let the initial state of the LFSR is $s^0 = (s_0, s_1, ..., s_{m-1})$, then the state of the LFSR at time t is

$$s^t = (s_t, s_{t+1}, ..., s_{t+m-1}) = L^t(s_0, s_1, ..., s_{m-1}).$$

Denote the output of the filter generator by $C_0, C_1, C_2, ...$, where $C_i \in \mathbb{F}_2^n$, then we can get the following equation system:

$$\begin{cases} C_0 = f \quad\ (s_0, s_1, ..., s_{m-1}) \\ C_1 = f(L\ (s_0, s_1, ..., s_{m-1})) \\ C_2 = f(L^2(s_0, s_1, ..., s_{m-1})) \\ \vdots \end{cases}$$

Our problem is to recover the initial state $s^0 = (s_0, s_1, ..., s_{m-1})$ from some keystream bits $C_0, C_1, C_2, ..$, where $C_i \in \mathbb{F}_2^n$.

Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, we give the following attack scenarios:

Algebraic Attack Scenarios on Multi-Output Case

S1: There exists low degree $g = (g_1, \ldots, g_n) \in \mathrm{Ann}(f)$, that is, $f \circ g = \sum_{i=1}^n f_i g_i = 0$.

S2: There exists $g = (g_1, \ldots, g_n) \in \mathbb{B}_m^n$ such that $f \circ g = \sum_{i=1}^n f_i g_i = h$ is of low degree.

S3: There exists low degree $g = (g_1, \ldots, g_n) \in \mathbb{B}_m^n$ such that $g \in \mathrm{Ann}(f + b)$ for some nonzero $b \in F_2^n$.

S4: There exists low degree $h \in \mathrm{Ideal}(f + b)$ for some nonzero $b \in F_2^n$.

How to Use the Low Degree Multiples and Related Definitions

There are three cases that we can mount algebraic attack by using the attack scenarios $S1$, $S2$, $S3$ and $S4$.

Case 1: $S1$ **and** $S2$

Using $S1$ and $S2$, we can obtain low degree equations:

(a) For scenario S1, if $f(v) = u \neq 0$, then for a low degree $g = (g_1, \ldots, g_n) \in \mathrm{Ann}(f)$, we obtain a low degree equation: $0 = g(v) \circ f(v) = g(v) \circ u = \sum_{i=1}^n u_i g_i(v)$.

(b) For scenario S2, if $f(v) = 0$, then for a low degree $h = g \circ f \in \mathrm{Ideal}(f)$, we have a low degree equation: $h(v) = 0$.

These low degree equations can be used to mount algebraic attack to stream ciphers. Thus the following definition should be significant.

Definition 3. *Let* $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$*, define* $AI_j(f) = \min\{\deg_j(g), \deg(h) | 0 \neq g \in \mathrm{Ann}(f), 0 \neq h \in \mathrm{Ideal}(f)\}$ *for* $j = 1, 2$.

From Definition 3 we can see that when $n = 1$, $AI_1(f) = AI_2(f) = AI(f)$, which is corresponding to the single-output case. For arbitrary n, $AI_j(f)$ should be large in certain sense.

However, when $n > 1$, only considering $AI_j(f)$ is not enough. For example, if $v \in \mathbb{F}_2^m$, $b \in \mathbb{F}_2^n$ such that $f(v) = b \neq 0$, and there is no low degree function in $Ann(f)$, but there exists a low degree function in $Ideal(f + b)$; this is possible, for $Ann(f)$ does not generally equal to $Ideal(f + b)$ when $n > 1$. Hence we also consider the following case:

Case 2: $S3$ **and** $S4$

Using $S3$ and $S4$, we can get the following attack:

(c) For scenario S3, if $f(v) = a \neq b$, then for a low degree $g \in \mathrm{Ann}(f + b)$, we get a low degree equation $0 = g(v) \circ (f(v) + b) = \sum_{i=1}^n (a_i + b_i) g_i(v)$.

(d) For scenario S4, if $f(v) = b$, then for a low degree $h \in \mathrm{Ideal}(f + b)$, we derive a low degree equation $h(v) = 0$.

Enlightened by **(a)**, **(b)**, **(c)** and **(d)**, we give the following definition:

Definition 4. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map. $AI_j(f+b) = \min\{\deg_j(g), \deg(h) | 0 \neq g \in \mathrm{Ann}(f + b), 0 \neq h \in \mathrm{Ideal}(f + b)\}$ for $j = 1, 2$.*

It is necessary that $\min_{b\in\mathbb{F}_2^n} AI_j(f + b)$ should be large in certain sense. Thus the following problem seem to be worthy of consideration:

Problem 2.1. What is the relationship of $AI(f)$ and $\min_{b\in\mathbb{F}_2^n} AI_j(f + b)$?

We would solve Problem 2.1 in the next section.

Similar with the case $n = 1$, low degree relations among f_1+b_1, f_2+b_2,...,f_n+b_n may also produce the low degree equation. We make use of the attack scenarios $S1$ and $S3$ to mount an attack similar with **(a)** and **(b)** mentioned above.

Case 3: $S1$ and $S3$

Using $S1$ and $S3$, we can mount the following attack:

(e) For scenario $S1$, if $f(v) \neq 0$, then we have similar low degree equation as in **(a)** using the low degree boolean function from $\mathrm{Ann}(f)$.

(f) For scenario $S3$, if $f(v) = 0$, then for a low degree g as in $S3$, we obtain $0 = g(v) \circ (f(v) + b) = \sum_{i=1}^{n} b_i g_i(v)$.

Corresponding to **(e)** and **(f)**, the following definition should be considered:

Definition 5. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$, define $ANI_j(f) = \min\{\deg_j(g) | 0 \neq g \in \cup_{b\in\mathbb{F}_2^n} \mathrm{Ann}(f + b)\}$ for $j = 1, 2$.*

By definition of $ANI_j(f)$, for any $b \in \mathbb{F}_2^n$, we have $ANI_j(f) = ANI_j(f + b)$. Thus a natural problem is as follows:

Problem 2.2. How to relate $AI(f)$ with $ANI_j(f)$ for $j = 1, 2$?

We will solve Problem 2.2 in the next section.

According to the above analysis, if we can find boolean functions that satisfy any of the three cases, then we can construct low degree equation system with the initial state bits as the variables. Moreover, if we can find function g that satisfies **S1** or **S3**, then we need at most $n \sum_{i=0}^{AI(f)} \binom{m}{i}$ keystream bits in order to obtain a complete saturated system solvable by linearization. Otherwise, the number of required data should be $n2^n \sum_{i=0}^{AI(f)} \binom{m}{i}$.

Similar with the analysis in [1], Table 1 gives the estimation of the complexity for the algebraic attack on multi-output stream ciphers. Here $D = \sum_{i=0}^{AI(f)} \binom{m}{i}$, m is the number of variables for f, n is the length of the keystream bits output by the multi-output boolean function, w is the exponent of the Gaussian reduction, and $w \leq 2.376$ [17].

When we apply algebraic attack on the LFSR-based multi-output stream ciphers, we look for the boolean functions that satisfy Case 1, Case 2 or Case 3 by computing Gröbner basis, resulting to equation system of the lowest degree with the initial state bits as the variables. Solve the equation system and we get the initial state value. In the following, we will give an attack example on the LFSR-based multi-output stream cipher.

Table 1. Estimation of the Complexity

Memory	Complexity	Data(best)	Data(worst)
$O(D^2)$	$O(D^w)$	$O(nD)$	$O(n2^nD)$

Example 1. Let the generator polynomial be $p(x) = x^7+x+1$, which is primitive. The filter function is a 7-variable and 2-output function $F = (f_1, f_2) : \mathbb{F}_2^7 \rightarrow \mathbb{F}_2^2$, where $f_1 : \mathbb{F}_2^7 \rightarrow \mathbb{F}_2$, $f_2 : \mathbb{F}_2^7 \rightarrow \mathbb{F}_2$.
$f_1 = x_1x_2x_3x_4x_5x_6+x_1x_2x_3x_4x_5x_7+x_1x_2x_3x_4x_5+x_1x_2x_3x_4x_6x_7+x_1x_2x_3x_4x_7+x_1x_2x_3x_5x_6x_7+x_1x_2x_3x_5x_6+x_1x_2x_3x_6x_7+x_1x_2x_3x_6+x_1x_2x_4x_6x_7+x_1x_2x_4x_6+x_1x_2x_4x_7+x_1x_2x_4+x_1x_2x_6x_7+x_1x_2x_7+x_1x_3x_4x_6+x_1x_3x_5x_6x_7+x_1x_3x_5x_7+x_1x_3x_5+x_1x_3x_6x_7+x_1x_3x_7+x_1x_4x_5x_6x_7+x_1x_4x_5x_7+x_1x_4x_6x_7+x_1x_4x_6+x_1x_4x_7+x_1x_4+x_1x_5x_6+x_1x_5x_7+x_1x_7+x_2x_3x_4x_5x_6x_7+x_2x_3x_4x_5x_6+x_2x_3x_4x_6x_7+x_2x_3x_4x_7+x_2x_3x_5x_7+x_2x_3x_5+x_2x_4x_5x_7+x_2x_4x_6+x_2x_5x_7+x_2x_5+x_2x_6x_7+x_3x_4x_5x_6x_7+x_3x_4x_6+x_3x_5x_7+x_3x_6+x_4x_5x_7+x_4x_7+1.$
$f_2 = x_1x_2x_3x_4x_5+x_1x_2x_3x_4x_7+x_1x_2x_3x_4+x_1x_2x_3x_5x_6x_7+x_1x_2x_3x_5x_7+x_1x_2x_3x_5+x_1x_2x_3+x_1x_2x_4x_5x_6x_7+x_1x_2x_4x_5+x_1x_2x_4x_6x_7+x_1x_2x_4x_7+x_1x_2x_5x_6x_7+x_1x_2x_5x_7+x_1x_2x_5+x_1x_2x_6x_7+x_1x_3x_4x_5x_6x_7+x_1x_3x_4x_6+x_1x_3x_4x_7+x_1x_3x_5x_6+x_1x_3x_5x_7+x_1x_3x_5+x_1x_3x_6+x_1x_3x_7+x_1x_3+x_1x_4x_5x_6x_7+x_1x_4x_6x_7+x_1x_4x_7+x_1x_4+x_1x_5x_7+x_1x_5+x_1x_6x_7+x_1x_6+x_1x_7+x_1+x_2x_3x_4x_5x_7+x_2x_3x_4x_5+x_2x_3x_4x_6x_7+x_2x_3x_4x_6+x_2x_3x_4x_7+x_2x_3x_4+x_2x_3x_5x_6+x_2x_3x_5x_7+x_2x_3x_5+x_2x_3x_6x_7+x_2x_3x_6+x_2x_3+x_2x_4x_5x_6x_7+x_2x_4x_5x_6+x_2x_4+x_2x_5x_6+x_2x_5x_7+x_2x_6x_7+x_2x_6+x_2x_7+x_2+x_3x_4x_5x_6+x_3x_4x_5x_7+x_3x_4x_5+x_3x_4x_6x_7+x_3x_4x_6+x_3x_4x_7+x_3x_4+x_3x_5x_6x_7+x_3x_5+x_3x_6x_7+x_3x_7+x_3+x_4x_5x_6x_7+x_4x_5x_6+x_4x_5x_7+x_4x_5+x_4x_6+x_4+x_5x_6x_7+x_5x_6+x_5x_7+x_5+x_6x_7+x_6+x_7.$

Here f_1 and f_2 are both of optimum algebraic immunity, $AI(f_1) = AI(f_2) = 4$.

Using Case 1, Case 2 and Case 3, we find the following equation:

$f_1(x_1x_5x_7+x_3x_5x_7+x_1x_7+x_3x_7+x_5x_7+x_7)+f_2(x_2x_3x_7+x_1x_5x_7+x_2x_5x_7+x_3x_5x_7+x_4x_5x_7+x_2x_6x_7+x_1x_7+x_3x_7+x_4x_7+x_5x_7+x_7) = 0.$

In the next section, we first prove some basic properties involving in the invariants defined in this section, and give a complete and strict proof that Definition 4 and Definition 5 are both equivalent to Definition 2, which provides great convenience for computing the algebraic immunity of multi-output boolean functions thanks to the detailed algebraic description in Definition 4 and Definition 5 .

4 Research on the Algebraic Immunity of the Multi-Output Boolean Function

We begin to prove the following lemma which shows that $\min_{0\neq c\in F_2^n}(AI(c \circ f))$ is the biggest among all these invariants:

Lemma 1. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map, then $ANI_1(f) \leq \min_{0 \neq c \in F_2^n}(AI(c \circ f))$, $ANI_2(f) \leq \min_{0 \neq c \in F_2^n}(AI(c \circ f))$.*

Proof. Let $0 \neq c = (c_1, \ldots, c_n) \in \mathbb{F}_2^n$ such that $AI(c \circ f) = \min_{e \neq 0}(AI(e \circ f))$. By definition of $AI(c \circ f)$, there exists a boolean function g such that $AI(c \circ f) = \deg(g)$, and $g(c \circ f + b) = 0$, where $b = 0$ or 1. Let $c_{i_1} = \cdots = c_{i_s} = 1$, other terms are 0. Then $c \circ f = f_{i_1} + \cdots + f_{i_s}$. Let $\overline{g}$ be a vector of boolean functions such that i_j position is g for $j = 1, \ldots, s$, other positions are 0. Hence $\overline{g} \circ (f + \overline{b}) = 0$, where $\overline{b}$ is a vector of length n with b in the i_1 position and 0 otherwise. Thus $ANI_1(f) \leq \deg_1(\overline{g}) = \deg(g)$. Similar discussion shows that $ANI_2(f) \leq \deg(g)$. □

Lemma 2. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a surjective map, then $AI(f) \leq ANI_2(f)$.*

Proof. Let $ANI_2(f)$ be achieved by $g = (g_1, \ldots, g_n)$. That is, $ANI_2(f) = \deg_2(g) = \deg(b \circ g)$ for a nonzero $b \in \mathbb{F}_2^n$, and $g \circ (f + a) = 0$ for some $a \in \mathbb{F}_2^n$.

For $v = (v_1, \ldots, v_m) \in \mathbb{F}_2^m$ such that $f(v) = a + b$, we have $f(v) + a = b$. Hence $0 = g(v) \circ (f(v) + a) = b \circ g(v) = (b \circ g)(v)$. Hence by definition of $AI(f)$, we have $AI(f) \leq \deg(b \circ g)$. □

By Lemma 1 and Lemma 2, the following corollary can be directly obtained:

Corollary 1. *Let $f = (f_1, \ldots, f_n) : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a surjective map, then $AI(f) \leq \min_{0 \neq c \in \mathbb{F}_2^n}(AI(c \circ f))$.*

Remark 1. Corollary 1 indicates that the algebraic immunity of a vector boolean function is less than or equal to that of its component functions. For a multi-output boolean function $f = (f_1, f_2, ..., f_n) \in \mathbb{B}_m^n$, we choose $c = (1, 0, 0, ..., 0) \in \mathbb{F}_2^n$, then we can see $AI(f) \leq AI(f_1)$. Similarly, we can get that $AI(f) \leq AI(f_i)$, $i \in \{2, ..., n\}$. The following example shows that the " = " can be reached.

Example 2. Let $f = (f_1, f_2) : \mathbb{F}_2^6 \to \mathbb{F}_2^2$ be a surjective map.
$f_1 = x_1x_2x_3x_4x_5 + x_1x_2x_3x_4x_6 + x_1x_2x_3x_4 + x_1x_2x_4x_5x_6 + x_1x_2x_4x_5 + x_1x_2x_4x_6 + x_1x_2x_5x_6 + x_1x_2x_5 + x_1x_2x_6 + x_1x_2 + x_1x_3x_4 + x_1x_3x_5 + x_1x_3x_6 + x_1x_4x_5 + x_1x_4 + x_1 + x_2x_3x_4x_5x_6 + x_2x_3x_4x_6 + x_2x_3x_6 + x_2x_4x_5 + x_2x_4x_6 + x_2x_4 + x_2x_5x_6 + x_2x_5 + x_3x_4x_6 + x_3x_4 + x_3x_6 + x_5 + x_6$.
$f_2 = x_1x_2x_3x_4x_5 + x_1x_2x_3x_5x_6 + x_1x_2x_3x_5 + x_1x_3x_4x_5x_6 + x_1x_3x_4x_5 + x_2x_3x_4x_5x_6 + x_1x_2x_3x_6 + x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4x_6 + x_1x_3x_5 + x_1x_4x_5 + x_2x_4x_5 + x_1x_3x_6 + x_2x_3x_6 + x_1x_4 + x_2x_4x_6 + x_2x_4 + x_3x_4x_6 + x_3x_4 + x_2x_5x_6 + x_2x_5 + x_1 + x_3x_6 + x_5 + x_6$.
We can compute that $AI(f) = 1$, $AI(f_1) = AI(f_2) = 3$, and $AI(f_1 + f_2) = 1$.

Example 2 verifies that $AI(f) \leq \min_{0 \neq c \in \mathbb{F}_2^n}(AI(c \circ f))$, and the " = " can be reached.

In order to investigate the problems in Section 2, we introduce a new invariant:

Definition 6. *Let $f = (f_1, \ldots, f_n) : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map. Define* $\mathrm{DI}(f) = \min\{\deg(g) | 0 \neq g \in \cup_{a \in Im(f)} \mathrm{Ideal}(f + a)\}$*, where $Im(f)$ is the image set of f.*

With the above notation, the following lemma provides an algebraic description of $AI(f)$, which seems to be obvious from the view of algebraic geometry and the definition of $AI(f)$. Since it is a basis of our subsequent discussions, we give here a rigorous proof.

Lemma 3. *Let $f = (f_1, \ldots, f_n) : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map, then $AI(f) = \mathrm{DI}(f)$.*

Proof. Let $\overline{\mathbb{F}}_2$ be the algebraically closure of $\mathbb{F}_2$. Let $f_i(X)$ be the polynomial obtained from the boolean function $f_i(x)$ by replacing the boolean variables x_i by the polynomial variables X_i for $i = 1, \ldots, m$, where $x = (x_1, \ldots, x_m)$, and $X = (X_1, \ldots, X_m)$.

Let J be the ideal generated by $f_1(X)+a_1, \ldots, f_n(X)+a_n, X_1^2 - X_1, \ldots, X_m^2 - X_m$ in $\mathbb{F}_2[X_1, \ldots, X_m]$, and $J_1 = J\overline{\mathbb{F}}_2[X_1, \ldots, X_m]$ is the extension of J in $\overline{\mathbb{F}}_2[X_1, \ldots, X_m]$. For $a \in \mathbb{F}_2^n$, it is easy to know $f^{-1}(a) = \{v \in \overline{\mathbb{F}}_2^m | \forall h \in J_1, h(v) = 0\}$.

Let $0 \neq g \in \mathbb{B}_m$ such that g vanishes on $f^{-1}(a)$ with $\deg(g) = AI(f)$, where $f^{-1}(a) \neq \emptyset$. Let $g_1 \in \mathbb{F}_2[X_1, \ldots, X_m]$ be any polynomial corresponding to g, i.e., $g_1 \bmod I = g$, where $I = \mathrm{Ideal}(X_1^2 - X_1, \ldots, X_m^2 - X_m)$. Then g_1 vanishes on $f^{-1}(a)$. Hence by Hilbert's Nullstellensatz, $g_1^r \in J_1$ for some positive integer r. Therefore, $g_1^r \in J_1 \cap \mathbb{F}_2[X_1, \ldots, X_m] = J$ ([13], page 212).

Note that $g_1^r \bmod I = (g_1 \bmod I)^r = g^r = g = g_1 \bmod I$, we have $g_1^r - g_1 \in I$. By $I \subseteq J$, we have $g_1 \in J$. Hence there exists the following equation in $\mathbb{F}_2[X_1, \ldots, X_m]$:

$g_1 = \sum_{i=1}^n h_i(X_1, \ldots, X_m)(f_i(X_1, \ldots, X_m) + a_i) + h$, where $h \in I$.

Passing to $\mathbb{B}_m$ by modulo I, we get $g = \sum_{i=1}^n h_i(x_1, \ldots, x_m)(f_i(x_1, \ldots, x_m) + a_i)$, thus $g \in \mathrm{Ideal}(f + a)$ in $\mathbb{B}_m$. Hence $\deg(g) \geq \mathrm{DI}(f)$, i.e., $AI(f) \geq \mathrm{DI}(f)$.

On the other hand, let $0 \neq h \in \cup_{a \in Im(f)} \mathrm{Ideal}(f + a)$ such that $\deg(h) = \mathrm{DI}(f)$. Thus for some $a \in Im(f)$ and boolean functions g_i, $h = \sum_{i=1}^n g_i(f_i + a_i)$. Thus for any $v \in f^{-1}(a)$, we have $h(v) = 0$. Hence $\mathrm{DI}(f) = \deg(h) \geq AI(f)$. Thus we have proved $AI(f) = \mathrm{DI}(f)$. □

The following theorem gives the relationship of $AI(f)$ and $ANI_2(f)$.

Theorem 1. *Let $f = (f_1, \ldots, f_n) : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map. Then $AI(f) \geq ANI_2(f)$. In particular, when f is surjective, $AI(f) = ANI_2(f)$.*

Proof. By Lemma 3, $AI(f) = \mathrm{DI}(f)$. Hence there exists a nonzero $g \in \mathrm{Ideal}(f + a)$ for some $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ such that $AI(f) = \deg(g)$.

Therefore there exists $h_i \in \mathbb{B}_m$ for $i = 1, \ldots, n$ such that

$$g = h_1(f_1 + a_1) + h_2(f_2 + a_2) + \cdots + h_n(f_n + a_n). \tag{1}$$

Since $g \neq 0$, there exists some i with $h_i \neq 0$. Without loss of generality, we may assume $i = 1$. Multiplying two sides of (1) by $f_1 + a_1$, we get:

$$g(f_1 + a_1) = h_1(f_1 + a_1) + h'_2(f + a_2) + \cdots + h'_n(f_n + a_n), \tag{2}$$

where $h'_i = h_i(f_i + a_i)$ for $i = 2, \dots, n$.

Combining (1) and (2), we obtain:

$$g(f_1 + a_1 + 1) + w_2(f_2 + a_2) + \cdots + w_n(f_n + a_n) = 0, \tag{3}$$

where $w_i = h_i + h'_i$ for $i = 2, \dots, n$.

From (3), we get $g' = (g, w_2, \dots, w_n) \in \mathrm{Ann}(f + a')$, where $a' = (a_1 + 1, a_2, \dots, a_n)$. Hence $ANI_2(f) \le \deg_2(g') \le \deg(g) = AI(f)$.

In particular, when f is surjective, by Lemma 2, $AI(f) \le ANI_2(f)$. Thus we have $AI(f) = ANI_2(f)$. □

Until now, we have solved Problem 2.2.

We would like to give an example to verify that when f is surjective, $AI(f) = ANI_2(f)$.

Example 3. Let $f = (f_1, f_2) : \mathbb{F}_2^5 \to \mathbb{F}_2^2$ be a surjective map.
$f_1 = x_1x_2x_3x_5 + x_1x_2x_5 + x_1x_2 + x_1x_3x_4x_5 + x_1x_3x_4 + x_1x_3x_5 + x_1x_4x_5 + x_1x_4 + x_2x_3 + x_2x_4x_5 + x_2x_5 + x_3x_4 + x_4x_5 + 1$.
$f_2 = x_1x_2x_3x_4 + x_1x_2x_3 + x_1x_2x_4x_5 + x_1x_2x_4 + x_1x_2x_5 + x_1x_3x_5 + x_1x_3 + x_1x_5 + x_2x_3x_4x_5 + x_2x_3x_5 + x_2x_4x_5 + x_2x_5 + x_3x_4 + x_3 + x_4x_5 + 1$.
We can compute that $AI(f) = ANI_2(f) = 2$, which completes our verification.

In order to give a characterization of $AI(f)$ in view of the minimum of $AI_2(f + a)$ and $AI_1(f + a)$, we note the following lemma:

Lemma 4. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map. Then $AI(f) \le AI_2(f)$.*

Proof. Let $g = (g_1, \dots, g_n) \in \mathrm{Ann}(f)$ such that $\deg_2(g) = \min\{\deg_2(h) | 0 \ne h \in \mathrm{Ann}(f)\}$.

Assume that $\deg_2(g) = \deg(b \circ g)$ for some nonzero $b \in \mathbb{F}_2^n$. Then $\forall v \in f^{-1}(b)$, we have $f(v) = b$. Hence $0 = g(v) \circ f(v) = (b \circ g)(v)$, which shows that $AI(f) \le \deg(b \circ g)$.

On the other hand, let $0 \ne h \in \mathrm{Ideal}(f)$ such that $\deg(h)$ reaches the minimum. Thus $h = \sum_{i=1}^n g_i f_i$ for some boolean functions $g_1, \dots, g_n$. $\forall v \in f^{-1}(0)$, we have $f(v) = (f_1(v), \dots, f_n(v)) = 0$. Hence $h(v) = 0$, which shows that $AI(f) \le \deg(h)$. Thus we have proved $AI(f) \le AI_2(f)$. □

The following theorem reveals the relations among $AI(f)$ and $AI_j(f + b)$ for $b \in \mathbb{F}_2^n$ and $j = 1, 2$, which can be thought as a solution to Problem 2.1.

Theorem 2. *Let $f : \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a map. Then $AI(f) = \min_{b \in \mathbb{F}_2^n} AI_2(f + b) = \min_{b \in \mathbb{F}_2^n} AI_1(f + b)$.*

Proof. By Lemma 4, we have $AI(f) \le AI_2(f)$. Hence for any $b \in \mathbb{F}_2^n$, we have $AI(f) = AI(f + b) \le AI_2(f + b)$. Thus $AI(f) \le \min_{b \in \mathbb{F}_2^n} AI_2(f + b) \le \min_{b \in \mathbb{F}_2^n} AI_1(f + b)$.

On the other hand, $\min_{b \in \mathbb{F}_2^n} AI_1(f + b) \le \min_{b \in \mathbb{F}_2^n} \{\min \deg(g) | 0 \ne g \in \mathrm{Ideal}(f + b)\}$. The latter is $AI(f)$ by Lemma 3. Hence $\min_{b \in \mathbb{F}_2^n} AI_1(f + b) \le AI(f)$. Thus we have proved the conclusion. □

Remark 2. The results in this section are very useful when we apply algebraic attack on LFSR-based stream ciphers.
(1)Theorem 2 and Lemma 3 verify that Definition 4 and Definition 6 are both equivalent to Definition 2. Particularly, Definition 4 gives a precise algebraic description of the algebraic immunity, which is more general and understandable when we mount algebraic attack on the multi-output model. Compared with Definition 2, our new definition provides a more convenient and intuitive target when we study the properties of the algebraic immunity for multi-output boolean functions.
(2) Given a multi-output filter function, according to Corollary 1, we may find equation system with degree less than the optimum algebraic immunity of the component functions.

Inspired by Corollary 1, we conjecture that for the LFSR-based single-output stream ciphers, maybe we can use our method in Section 3 to derive equation system of degree lower than the optimum algebraic immunity of the filter function. In the next section, we apply our attack on single-output stream ciphers by using augmented function to transform the model into multi-output.

5 Algebraic Attacks on LFSR-Based Single-Output Stream Ciphers by Using Augmented Function

Qichun Wang and Thomas Johansson presented a method called higher order algebraic attack on stream ciphers [14]. They also apply their attack on augmented functions. However, they did not give the detailed and systematic attack scenarios on how to look for the low degree equations.

In this section, we would like to apply the method presented in Section 3 on the LFSR-based single-output stream ciphers. When the boolean function is single-output, we can construct augmented function to transform it into multi-output case.

First, let us give the model of single-output stream ciphers. The LFSR is the same with the one described in Section 3. While the filter function $f : \mathbb{F}_2^m \rightarrow \mathbb{F}_2$ is single-output. The algebraic immunity of f is optimum, which means that the lowest degree of the equation system we can find equals to the algebraic immunity of f when we apply the conventional algebraic attack given in [1] on this model.

Let the initial state of the LFSR be $s^0 = (s_0, s_1, ..., s_{m-1})$, then the state of the LFSR at time t is

$$s^t = (s_t, s_{t+1}, ..., s_{t+m-1}) = L^t(s_0, s_1, ..., s_{m-1}).$$

Denote the output of the filter generator by $c_0, c_1, c_2, ...$, where $c_i \in \mathbb{F}_2$, then we can get the following equation system:

$$\begin{cases} c_0 = f \quad (s_0, s_1, ..., s_{m-1}) \\ c_1 = f(L\ (s_0, s_1, ..., s_{m-1})) \\ c_2 = f(L^2(s_0, s_1, ..., s_{m-1})) \\ \vdots \end{cases}$$

When we apply the conventional algebraic attack on this model, we can get equations of degree no less than $\lceil \frac{m}{2} \rceil$. While according to Corollary 1, we may get equations of degree less than $\lceil \frac{m}{2} \rceil$ if we use the augmented function. First, let us recall the definition of augmented function:

Definition 7. *[16] For a nonlinear filter function $f : \mathbb{F}_2^m \to \mathbb{F}_2$, the n-th augmented function of f is defined as $F^n : \mathbb{F}_2^{m+n-1} \to \mathbb{F}_2^n$:*

$$F^n(x_1, ..., x_{n+m-1}) = (f(x_1, ..., x_m), f(x_2, ..., x_{m+1}), ..., f(x_n, ..., x_{n+m-1})).$$

From Definition 7, we can define the augmented function for the LFSR-based stream cipher described in this section as follows:

Definition 8. *For the nonlinear filter function $f : \mathbb{F}_2^m \to \mathbb{F}_2$, the n-th augmented function of f is defined as $F^n : \mathbb{F}_2^{n+m-1} \to \mathbb{F}_2^n$:*
$F^n(s_0, s_1, ..., s_{n+m-1}) = (f(s_0, s_1, ..., s_{m-1}), f(s_1, s_2, ..., s_m), ..., f(s_n, s_{n+1}, ..., s_{n+m-1}))$

The generator polynomial is $p(x) = p_0 + p_1 x + ... + p_{m-1}x^{m-1} + x^m$, which is primitive. Then we can get the generator matrix of the sequence generated by LSFR:

$$M = \begin{pmatrix} 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ 0 & 0 & 0 & ... & 1 \\ p_0 & p_1 & p_2 & ... & p_{m-1} \end{pmatrix} \tag{4}$$

Then Definition 8 can also be written as:

Definition 9. *For the nonlinear filter function $f : \mathbb{F}_2^m \to \mathbb{F}_2$, the n-th augmented function of f is defined as $F^n : \mathbb{F}_2^m \to \mathbb{F}_2^n$:*

$$F^n(s_0, s_1, ..., s_{m-1}) = (f(s_0, s_1, ..., s_{m-1}), f(M(s_0, s_1, ..., s_{m-1})^T), ..., f(M^n(s_0, s_1, ..., s_{m-1})^T)).$$

According to Corollary 1, we derive that the algebraic immunity of the augmented function $AI(F^n) \leq AI(f)$. It means that although $AI(f)$ is optimum, we may find equation system of degree less than $AI(f)$ by targeting the augmented function F^n.

Here we would like to give examples to show that when the single-output filter function f has optimum algebraic immunity, our attack on augmented function case can find equation system of degree less than $AI(f)$.

Example 4. Let the generator polynomial is $p(x) = x^5+x^2+1$. The filter function is a 5-variable Carlet-Feng function $f(x_1, x_2, ..., x_5) = x_1x_2x_3x_5 + x_1x_2x_4x_5 + x_1x_2x_4 + x_1x_3x_4x_5 + x_1x_3 + x_1x_4x_5 + x_2x_3x_4 + x_2x_3x_5 + x_2x_3 + x_2 + x_3x_5 + x_4x_5 + x_4 + 1$.

Then we know that $AI(f) = 3$, $deg(f) = 4$. So if we apply the conventional algebraic attack on this model, we can get equation system with degree no less than $AI(f) = 3$.

In the following, we will show the results when we apply our method in Section 3 on the augmented function, which is constructed by the expressions of the keystream bits of different clocks. In this case, we construct the augmented function $F^2 : \mathbb{F}_2^5 \rightarrow \mathbb{F}_2^2$ as follows:

$$F^2 = (f_1, f_2) = (f(x_1, x_2, ..., x_5), f(M(x_1, x_2, ..., x_5)^T).$$

Where M is the companion matrix of $p(x)$.

$$M = \begin{pmatrix} 0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 1\,0\,1\,0\,0 \end{pmatrix} \tag{5}$$

Similar with the attack on multi-output boolean function, we mount algebraic attack on this model and get the following equation:

$$f_1(x_1 + x_3 + x_4 + x_5 + 1) + f_2(x_1x_4 + x_2 + x_3 + 1) = 0.$$

$AI(F^2) = 1 < AI(f) = 3$. Then we can get equation system of degree less than 3.

Remark 3. This example indicates that the algebraic immunity of the multi-output boolean function is less than or equal to that of the component functions, which is consistent with Corollary 1.

We would like to give another example:

Example 5. Let the generator polynomial is $p(x) = x^{12}+x^7+x^6+x^5+x^3+x+1$. The filter function is a 12-variable function f, where $1_f = \{M^i\beta|0 \leq i < 2^{12-1}\}$, $\beta = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T$, M is the companion matrix of $p(x)$. According to [15], f is of optimum algebraic immunity, $AI(f) = 6$. So if we apply the conventional algebraic attack on this model, we can get equation system with degree no less than $AI(f) = 6$. the complexity of the standard algebraic attack is roughly $\mathcal{O}(E^w)$ in time and $\mathcal{O}(E)$ in data, where $E = \sum_{i=0}^{6} \binom{m}{i}$, m is the number of variables for f, w is the exponent of the Gaussian reduction, and $w \leq 2.376$ [17].

Here we set $n = 2$, then the augmented function:

$$F^2 = (f_1, f_2) = (f(x_1, x_2, ..., x_{12}), f(M(x_1, x_2, ..., x_{12})).$$

Where M is the companion matrix of $p(x)$.

$$M = \begin{pmatrix} 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&1&0&1&0&1&1&1&0&0&0&0 \end{pmatrix} \tag{6}$$

Using our method, we can find the following relation:

$$(f_1 + f_2)(x_7 + x_{12}) = 0.$$

We get a linear equation, which can greatly decrease the computational complexity.

According to Definition 7, we can learn that if the LFSR-based stream cipher is equidistant, there is no need to involve all the initial state bits of the LFSR in the augmented function. For instance, assume the length of the LFSR is m and the filter function is of k-variable ($m > 10k$), if we adapt 2-th augmented function, then we only need to consider $k + 1$ variables instead of the whole m initial state bits. Hence the number of variables involved in the augmented function can be reduced in a large extent, which cuts down the cost of deriving the low degree boolean functions that satisfy Case 1, Case 2 or Case 3. To illustrate the method intuitively, we would like to give the following example.

Example 6. Let the length of the LFSR be $m = 128$. The generator polynomial is $p(x) = x^{11} + x^2 + 1$. The filter function is a 11-variable Carlet-Feng function. The support of f is $\{0, 1, \alpha, \alpha^2, ..., \alpha^{2^{11-1}-2}\}$, where α is a primitive element of the finite field F_2^{12}. The stream cipher is equidistant.

If we use the whole initial state bits as the variables of the augmented function, then the number of variables is 128, which is almost impossible for us to derive the low degree functions by using computer.

We take advantage of the equidistant feature of the cipher and adapt the augmented function $F^2 := (f_1, f_2)$, where $f_1 = f, f_2 = f(x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})$. We can check that $AI(f_1) = AI(f_2) = 6$.

Then the number of variables for F^2 is 12, less than $m = 128$. We search the low degree boolean functions quickly and get the following equation of degree 2:

$$(f_1 + f_2)(x_1x_{12} + x_1 + x_{12} + 1).$$

In the same way, we can find low degree equations with the other initial state bits as the variables.

Remark 4. The three examples in this section are very meaningful. They suggests another way to mount algebraic attack on the LFSR-based single-output stream ciphers. In particular, when the filter function is of optimum algebraic immunity, we may use the augmented function model to find equation system of degree less than the algebraic immunity.

6 Conclusion

This paper makes extensive efforts on the LFSR-based multi-output stream ciphers. To sum up, we have made two main contributions. First, a systematic and standardized framework of algebraic attack on multi-output stream ciphers is proposed, which is unprecedented. We put forward four attack scenarios and provide interpretation on how to leverage the low degree multiples in Case 1, Case 2 and Case 3. Moreover, our approach is applied on the LFSR-based stream ciphers with single-output boolean functions by using augmented function to change them into multi-output models. we can find equation system of degree no greater than the optimum algebraic immunity of the original single-output filter boolean function, which suggests a better way than the conventional algebraic attack in [1] to attack the LFSR-based stream ciphers with single-output boolean functions.

Secondly, this paper provides novel definitions as the algebraic descriptions of the multi-output boolean functions, and rigorously proves that the descriptions are equivalent to the definition given by Armknecht and Krause [11]. Our definitions are more convenient and understandable for us to look for the low degree multiples. We also study the general properties and various possible characterizations of the invariants presented in Section 3.

Further study should focus on the detailed relationship between the algebraic immunity and number of variables for the multi-output boolean functions. Research on how to find the low degree multiples more effectively is also very meaningful.

References

1. Courtois, N., Meier, W.: Algebraic attacks on stream ciphers with linear feedback, Eurocrypt 2003, Lecture notes in computer science 2656, pp. 345-359, 2003.
2. Courtois, N., Meier, W.: Algebraic attacks on stream ciphers with linear feedback, extended version of the Eurocrypt 2003 paper.
3. F. Armknecht and M. Krause: Algebraic attacks on combiners with memory. In D. Boneh, editor, CRYPTO, volume 2729 of Lecture Notes in Computer Science, pages 162-175. Springer, 2003.
4. Meier, W., Pasalic, E., Carlet, C.: Algebraic attacks and decomposition of Boolean functions. Eurocrypt 2004, Lecture notes in computer sciences vol. 3027, pp. 474-491, 2004.
5. Braeken, A., Preneel, B.: On the algebraic immunity of symmetric Boolean functions. INDOCRYPT 2005. lncs, vol.3797, pp. 35-48. Springer, Heidelberg, 2005.

6. Carlet, C., Dalai, D.K., Gupta, K. C., Maitra, S.: Algebraic immunity for cryptographically significant Boolean functions: analysis and construction. IEEE Trans. Inform. Theory 52(7), 3105-3121, 2006.
7. Claude C., Xiangyong Z., Chunlei L. and Lei H.: Further Properties of several classes of Boolean functions with optimum algebraic immunity. Designs, Codes and Cryptography, volume 52, vumber 3, pp. 303-338, 2009.
8. Deepak K. D., Subhamoy M. and Sumanta S.: Basic Theory in Construction of Boolean Functions with Maximum possible Annihilator Immunity. Designs, Codes and Cryptography, pp. 41-58, 2006.
9. Feng, K., Liao, Q., Yang, J.: Maximum Values of Gneralized Algebraic Immunity. Designs, Codes and Cryptography,volume 50, number 2, pp. 243-252, 2009.
10. Carlet, C., Feng, K.: An Infinite Class of Balanced Functions with Optimal Algebraic Immunity, Good Immunity to Fast Algebraic Attacks and Good Nonliearity. ASIACRYPT, LNCS 5350, pp. 425-440, 2008.
11. Armknecht, F., Krause, M.: Constructing single-and multi-output boolean functions with maximal algebraic immunity. LNCS. 4052, pp. 180-191, 2006.
12. R. Anderson: Searching for the Optimum Correlation Attack. Fast Software Encryption-Leuven'4, LNCS 1008, Springer-Verlag, pp.137-143, 1995.
13. Becker, E., Weispfenning, V.: Gröbner bases, A computational approach to commutative algebra. Graduate texts in Mathematics 141, Springer, 1993.
14. Qichun Wang, Thomas Johansson: Higher Order Algebraic Attacks on Stream Ciphers. http://eprint.iacr.org/2012/013.pdf, 2012.
15. Q.Wang, J. Peng, H. Kan, and X. Xue: Constructions of cryptographically significant Boolean functions using primitive polynomials. IEEE Trans. Inf. Theory, vol. 56, no. 6, pp. 3048C3053, 2010.
16. R. A. Rueppel: Analysis and Design of Stream Ciphers. Berlin, Germany, Springer-Verlag, 1986.
17. Don Coppersmith, Shmuel Winograd: Matrix multiplication via arithmetic progressions. J.Symbolic Computaion, 9, pp.251-280, 1990.