UDC 512.624.95

On algebraic attacks on some keystream generators

Sergey Chekanov, Evgeny Rassolov
Far Eastern Federal University, RUSSIA
stepltd@mail.ru
sky.04@mail.ru

Abstract. Many stream ciphers are used to encrypting the data stream at high speed. These stream ciphers are based on keystream generators. The keystream generator produces an output stream of arbitrary length, called a keystream, after initialization with a secret value, called a key. The legal receiver can produce the same keystream with the secret key and decrypt the encrypted data stream. The main problem of any algebraic attack is the need to solve a nonlinear system of algebraic equations. We used two approaches for solving these systems a linearization method and a method based on the Gröbner basis [1, 2]. All explored in the work generators that are based on linear feedback shift registers (LFSR). We supposed that an adversary knows everything about the generator beside the secret key. As result where build up a few generators and algebraic attacks on them.

Key words: keystream generators, algebraic attacks, Gröbner basis.

Let $K = (k_0, \dots, k_{n-1})$ be the unknown initial state and L be the known feedback matrix (see [1]). The secret key K and matrix L would produce a keystream $k_0, k_1, k_2, \dots$ where $(k_t, \dots, k_{k+n-1}) = KL^t$. If an adversary knows the values $k_{t_1}, \dots, k_{t_m}$ of the keystream elements at the m clocks $t_1, \dots, t_m$ he can set up the following system of linear equations:

$$\begin{gathered} k_{t_1} = KL^{t_1}P, \\ \dots \\ k_{t_m} = KL^{t_m}P, \end{gathered} \tag{1}$$

where $P = (1, 0, \dots, 0)^t$. The linearity of the system is a real treat for a generator of keystream. Thus, to strengthen LFSR-based keystream generators, one has to incorporate some kind of non-linearity. To apply, for example, a non-linear function to the outputs from several LFSRs and to output the result.

Definition 1. Let F be a finite field. A (l, m) −combiner consists of the following components:

1) s LFRSs with lengths $n_1, \dots, n_s$ and feedback matrices $L_1, \dots, L_s$;
2) an internal state $S \in F^m \times F^n$ where $n = n_1 + \dots + n_s$;
3) a matrix L over F of size $n \times n$, defined by
$$L := \begin{pmatrix} L_1 & & 0 \\ & \ddots & \\ 0 & & L_s \end{pmatrix};$$
4) a (projection) matrix P over F of size $n \times l$;
5) a non-linear next memory state function $\Psi: F^m \times F^l \to F^m$;
6) an output function $f: F^m \times F^l \to F$.

The definition supposes the combiner consist of s LFSRs and use outputs of them and the memory for generation the final outputs.

Let start with simple generator over field F_2. It uses three LFSRs lengths three, four and five, respectively. Their initial states are denoted by $A_0 = (a_0, a_1, a_2), B_0 = (b_0, b_1, b_2, b_3), C_0 = (c_0, c_1, c_2, c_3, c_4)$. The minimal polynomial of LFSR A is $m_a(x) = x^3 + x + 1$ and the sequence (a_t) produced by LFRS A fulfills the recursion $a_{t+3} = a_{t+1} + a_t$. The minimal polynomial of B is $m_b(x) = x^4 + x^3 + 1$ and the recursion $b_{t+4} = b_{t+3} + b_t$. The third minimal polynomial is $m_c(x) = x^5 + x^2 + 1$

with the recursion $c_{t+5} = c_{t+2} + c_t$. In this case $K := (a_0, a_1, a_2, b_0, b_1, b_2, b_3, c_0, c_1, c_2, c_3, c_4)$. The feedback matrix L and the projection matrix are defined as

$$L := \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \text{ and } P := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

At each clock t, the keystream element z_t is generated by the function $z_t = f(K_t) = f(a_t, b_t, c_t) = a_t b_t + a_t c_t$. Hereby, a_t, b_t, c_t, z_t are the outputs from LFRSs A, B, C and the keystream at the clock t, respectively.

Let $A_0 = (0, 0, 1), B_0 = (0, 0, 0, 1), C_0 = (1, 0, 1, 1, 1)$ than the generated keystream we may see in the table 1.

Table 1

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
a_t	**0**	**0**	**1**	0	1	1	1	0	0	1	0	1	1	1	0	0	1	0	1	1
b_t	**0**	**0**	**0**	**1**	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1
c_t	**1**	**0**	**1**	**1**	**1**	0	1	0	1	0	0	0	0	1	0	0	1	0	1	1
z_t	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0

On the base of the Table 1 we can construct the system of non-linear equations. For a successful algebraic attack need to solve the system.

1) $a_0 b_0 + a_0 c_0 = 0,$
2) $a_1 b_1 + a_1 c_1 = 0,$
3) $a_2 b_2 + a_2 c_2 + 1 = 0$
4) $a_0 b_3 + a_1 b_3 + a_0 c_3 + a_1 c_3 = 0$
5) $a_1 b_0 + a_1 b_3 + a_2 b_0 + a_1 c_4 + a_2 c_4 = 0$
6) $a_2 b_1 + a_2 b_0 + a_2 b_3 + a_0 b_1 + a_0 b_0 + a_0 b_3 + a_1 b_1 + a_1 b_0 + a_1 b_3 + a_2 c_0 + a_2 c_2 + a_0 c_0 + a_0 c_2 + a_1 c_0 + a_1 c_2 + 1 = 0$
7) $a_0 b_2 + a_0 b_1 + a_0 b_0 + a_0 b_3 + a_2 b_2 + a_2 b_1 + a_2 b_0 + a_2 b_3 + a_0 c_1 + a_0 c_3 + a_2 c_1 + a_2 c_3 = 0$
8) $a_0 b_2 + a_0 b_1 + a_0 b_0 + a_0 c_2 + a_0 c_4 = 0$
9) $a_1 b_2 + a_1 b_1 + a_1 b_3 + a_1 c_3 + a_1 c_0 + a_1 c_2 = 0$
10) $a_2 b_2 + a_2 b_0 + a_2 c_4 + a_2 c_1 + a_2 c_3 = 0$
11) $a_0 b_1 + a_0 b_3 + a_1 b_1 + a_1 b_3 + a_0 c_0 + a_0 c_4 + a_1 c_0 + a_1 c_4 = 0$
12) $a_2 b_2 + a_2 b_0 + a_2 b_3 + a_1 b_2 + a_1 b_0 + a_1 b_3 + a_2 c_1 + a_2 c_1 + a_2 c_0 + a_2 c_2 + a_1 c_1 + a_1 c_0 + a_1 c_2 + 1 = 0$

13) $a_0b_1 + a_0b_0 + a_1b_1 + a_1b_0 + a_2b_1 + a_2b_0 + a_0c_2 + a_0c_1 + a_0c_3 + a_1c_2 + a_1c_1 + a_1c_3 + a_2c_2 + a_2c_1 + a_2c_3 = 0$
14) $a_2b_2 + a_2b_1 + a_0b_2 + a_0b_1 + a_2c_3 + a_2c_2 + a_2c_4 + a_0c_3 + a_0c_2 + a_0c_4 + 1 = 0$
15) $a_0b_2 + a_0b_3 + a_0c_4 + a_0c_3 + a_0c_0 + a_0c_2 = 0$
16) $a_1b_0 + a_1c_0 + a_1c_2 + a_1c_4 + a_1c_1 + a_1c_3 = 0$
17) $a_2b_1 + a_2c_1 + a_2c_3 + a_2c_0 + a_2c_4 + 1 = 0$
18) $a_0b_2 + a_1b_2 + a_0c_4 + a_0c_1 + a_0c_0 + a_1c_4 + a_1c_1 + a_1c_0 = 0$
19) $a_1b_3 + a_2b_3 + a_1c_0 + a_1c_1 + a_2c_0 + a_2c_1 = 0$
20) $a_2b_0 + a_2b_3 + a_0b_0 + a_0b_3 + a_1b_0 + a_1b_3 + a_2c_1 + a_2c_2 + a_0c_1 + a_0c_2 + a_1c_1 + a_1c_2 = 0$
21) $a_0b_1 + a_0b_0 + a_0b_3 + a_2b_1 + a_2b_0 + a_2b_3 + a_0c_3 + a_0c_2 + a_2c_3 + a_2c_2 + 1 = 0$
22) $a_0b_2 + a_0b_1 + a_0b_0 + a_0b_3 + a_0c_4 + a_0c_3 = 0$
23) $a_1b_2 + a_1b_1 + a_1b_0 + a_1c_0 + a_1c_2 + a_1c_4 = 0$
24) $a_2b_2 + a_2b_1 + a_2b_3 + a_2c_1 + a_2c_3 + a_2c_0 + a_2c_2 = 0$
25) $a_0b_2 + a_0b_0 + a_1b_2 + a_1b_0 + a_0c_2 + a_0c_4 + a_0c_1 + a_0c_3 + a_1c_2 + a_1c_4 + a_1c_1 + a_1c_3 = 0$
26) $a_2b_1 + a_2b_3 + a_1b_1 + a_1b_3 + a_2c_3 + a_2c_0 + a_2c_4 + a_1c_3 + a_1c_0 + a_1c_4 = 0$
27) $a_0b_2 + a_0b_0 + a_0b_3 + a_1b_2 + a_1b_0 + a_1b_3 + a_2b_2 + a_2b_0 + a_2b_3 + a_0c_4 + a_0c_1 + a_0c_0 + a_0c_2 + a_1c_4 + a_1c_1 + a_1c_0 + a_1c_2 + a_2c_4 + a_2c_1 + a_2c_0 + a_2c_2 = 0$
28) $a_2b_1 + a_2b_0 + a_0b_1 + a_0b_0 + a_2c_0 + a_2c_1 + a_2c_3 + a_0c_0 + a_0c_1 + a_0c_3 = 0$
29) $a_0b_2 + a_0b_1 + a_0c_1 + a_0c_2 + a_0c_4 = 0$

It has been proven that finding a solution of a system quadratic equations is an NP-hard problem. This means that there is probably no polynomial time algorithm for solving general systems of non-linear equations over finite fields. We used two approaches for solving non-linear systems: a linearization method and the Gröbner bases method.

The theory of Gröbner bases has been initiated by Bruno Buchberger as a tool to solve the ideal membership problem. The main useful result is the following theorem.

Theorem. Consider a system of equations $f_1 = 0, \dots, f_n = 0$ where $f_1, \dots, f_n \in F_q[x_1, \dots, x_n]$ and F_q is a finite field. We define the ideal $I =< f_1, \dots, f_n, x_1^q - x_1, \dots, x_n^q - x_n >$. Then, for any term ordering, the reduced Gröbner basis of *I* is equal to

1. $\{x_1 - x_1^0, \dots, x_n - x_1^0\}\ if\ (x_1^0, \dots, x_n^0)$ is the unique solution of the system of equations.
2. { 1 } if the system has no solution.

We used the program WOLFRAM MATHEMATIC for construct a Gröbner basis.

The key idea of linearization method is to re-write the system of non-linear equation from n unknowns as a new system of linear equations with increased number of unknowns. This allows us to solve a system of linear equations, which is a much simpler task.

For future computing experiments it was written the program “Generator” in C++ language for generation a key stream generator. Where tested a few generators and attacked by both linearization and Gröbner basis methods.

The most difficult and interesting problem is to estimate the complexity of Gröbner basis method, especially to get a theoretical border. It is obviously that the linearization method is more effective in a small dimension tasks, if the dimension of the task grows, it is hard to say which approach is preferable.

Obviously, any experiment is useful and productive in some way, it is much more important to get a general theoretical base of a stream ciphers resistance.

Definition 2. Let $f(x_1, \dots, x_n)$ be Boolean function. Then algebraic immunity is defined as

$$AI(f) = \min\{deg(g) \mid fg \equiv 0 \ or \ (f \oplus 1)g \equiv 0\} .$$

We intend to explore the relationship between the cipher resistance and the algebraic immunity of Boolean functions used.

1. Armknecht F. Algebraic attacks on certain stream ciphers / F. Armknecht // In C. Wolf, S. Lucks, and P. Yau, editors, WEWORC, volume P–74 of LNI, , 2005.- 217 c.
2. Courtois N., Meier W. Algebraic Attacks on Stream Ciphers with Linear Feedback // Proceedings of Eurocrypt 2003, Lecture Notes in Computer Sciences. 2003. V. 2656. P. 345 – 359.