Trends in Mathematics
Information Center for Mathematical Sciences
Volume 8, Number 1, June, 2005, Pages 133–143

ALGEBRAIC ATTACKS ON STREAM CIPHERS (SURVEY)

DONG HOON LEE

Abstract. Most stream ciphers based on linear feedback shift registers (LFSR) are vulnerable to recent algebraic attacks. In this survey paper, we describe generic attacks: existence of algebraic equations and fast algebraic attacks. The generic attacks only states the existence and gives the upper bound of the complexity. Thus we should find good algebraic equations, case by case, in order to apply the attack to a specific stream cipher. We also review some stream ciphers attacked: Toyocrypt, LILI-128, E_0, and summation generator.

1. Introduction

Algebraic attacks are to find the secret key in cryptosystems by solving the underlying algebraic equations. Algebraic attacks were first applied to public key cryptosystems [13, 4] as linearization and re-linearization. Their idea was extended to XL (Extended Linearization) algorithm to solve overdefined systems of polynomial equations [8]. XL algorithm has gathered much attention since it was applied to try finding the secret key of AES [10].

The purpose of algebraic attacks on block ciphers is to find algebraic *quadratic* equations[1] relating inputs and outputs of S-box. However recent researches show that XL algorithm may not work well as it was claimed [21, 11]. On the other hand, the resistance of S-boxes from well-known polynomials against algebraic attacks was analyzed in [3].

The successful application of algebraic attacks was an application to stream ciphers. It was done on Toyocrypt and was soon extended to LILI-128 [5, 9]. Stream ciphers that utilize memory were first thought to be much more resistant to these attacks, but soon it was shown that even these cases were subject to algebraic attacks [2, 6].

In this survey paper, we classify algebraic attacks can into two types. The first type is generic attacks. This kind of attacks is theoretically applied to general stream ciphers based on linear feedback shift registers (LFSRs) [9, 2, 6, 7, 1, 12]. It states the existence of algebraic equations relating the initial key bits and output bits and gives upper bound of complexity of algebraic attacks. The second type is concrete attacks that were applied to concrete stream ciphers [5, 9, 1, 15]. In

[1]The equations of higher degree are useless to find the secret key since there are so many unknown variables.

fact, we should find good (low degree) algebraic equations, case by case, in order to apply to concrete stream ciphers.

The rest of this paper is organized as follows: We recall some classical criteria of designing Boolean functions in the following section. Boolean functions are very important building blocks for designing classical stream ciphers. In order to design secure stream ciphers, there are several criteria that Boolean functions should satisfy. Algebraic attacks and a new criterion are described in Section 3. We explain generic algebraic attacks including fast algebraic attacks in Section 4. We present a few examples that were attacked by algebraic attacks in Section 5 and present our conclusion in Section 6.

2. Criteria of Designing Boolean Functions

Classical stream ciphers consist of a linear part that produces a sequence of a large period, usually composed of one or several LFSRs, and a nonlinear part that produces the output, given the state of the linear part. There are two classical model of nonlinear parts based on LFSRs: nonlinear combiners and nonlinear filters.

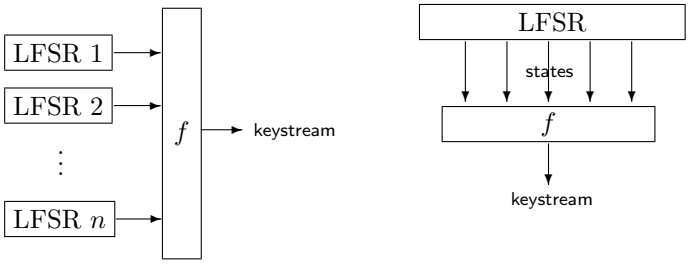

Figure 1. Classical nonlinear combiner and filter model

A function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called a *Boolean function* where $\mathbb{F}_2^n$ is an n-dimensional vector space over $\mathbb{F}_2$ and $\mathbb{F}_2$ is the finite field with two elements $\{0, 1\}$.

In the both model, a Boolean function f is used to generate the key stream bits. In nonlinear combiner model, one bit input is extracted from each LFSR and then all the bits are combined using the Boolean function. In nonlinear filter model, several bits are generated from a single LFSR and these are then combined using the Boolean function.

We recall some classical cryptographic criteria for a Boolean function f to design secure stream ciphers.

- **Balancedness:** f is *balanced* if the cardinality of the set $\{x \mid f(x) = 0\}$ is equal to that of the set $\{x \mid f(x) = 1\}$. f should be balanced to prevent the system from leaking statistical information on the plaintext when the ciphertext is known.
- **High algebraic degree:** A Boolean function can be represented as a multivariate polynomial over $\mathbb{F}_2$. When the representation is reduced form, the polynomial is called the *algebraic normal form* (ANF) of f. The *algebraic*

degree or briefly *degree* of f is defined to be the degree of the algebraic normal form of f. A function of degree 1 is called an *affine* function.

- **High linear complexity:** Every finite or periodic infinite sequence of $\{0,1\}$ can be emulated with a single LFSR, that is, there exists an LFSR with a initial states that produces the given sequence. The shortest length of LFSR that produces the same key stream is called the *linear complexity*.
- **High nonlinearity:** The *distance* between two functions f and g is the cardinality of $\{x \in \mathbb{F}_2^n \mid f(x) + g(x) = 1\}$. The *nonlinearity* $\mathcal{N}(f)$ of f is defined by the minimal distance between f and the set of all affine functions. Any Boolean function has the nonlinearity at most $2^{n-1} - 2^{n/2-1}$. It is trivial that any affine function has nonlinearity of zero.
- **High correlation immunity:** A Boolean function has correlation immunity of order m if $f(x) + (a_0x_0 + a_1x_1 + \cdots + a_{n-1}x_{n-1})$ is balanced for all $1 \le wt(a) \le m$ where a is a vector $(a_0, a_1, \cdots, a_{n-1})$ and $wt(a)$ is the number of nonzero a_is. A balanced and correlation immune function is called a *resilient* function.

These criteria have trade-off relationship. For example, Siegenthaler inequality [20] states that any correlation immune function of order m on n-variables has the algebraic degree at most $n - m$, that any resilient function of order $m < n - 1$ has the algebraic degree at most $n - m - 1$, and that any $(n-1)$-resilient function is affine.

Sarkar and Maitra showed in [19] that the distance between any m-resilient function and any affine function is divisible by 2^{m+1}. Thus we have the upper bound of the nonlinearity of f,

$$\mathcal{N}(f) \le \begin{cases} 2^{n-1} - 2^{m+1} & \text{if } n/2 < m+2, \\ 2^{n-1} - 2^{m+1} - 2^{n/2-1} & \text{if } n \text{ is even and } n/2 \ge m+2, \\ 2^{n-1} - 2^{m+1}\lceil 2^{n/2-m-2} \rceil & \text{if } n \text{ is odd and } n/2 \ge m+2. \end{cases}$$

3. Algebraic Attacks and a New Criterion

Until recently, a high algebraic degree, a high nonlinearity, a high linear complexity, and a high resiliency were sufficient criteria for designing a Boolean function used in a stream cipher as a combining function or a filtering function. (Of course, this is not easy!)

However the advent of the algebraic attacks makes the situation more complex. Algebraic attacks are to find the secret key by solving the underlying algebraic equations. The purpose of algebraic attacks is to find algebraic equations involving the secret key bits and the output bits of f. If such a equation of *low* degree is found, then algebraic attacks are very efficient.

Low degree equations are obtained by multiplying the Boolean function f by a well chosen low degree g such that $h = fg$ has also low degree [9]. If there exists such a equation, we have the following two scenarios:

(1) If $g = h$, then $(f+1)g = 0$. That is g is an annihilator of $(f+1)$

(2) Otherwise, we have $fh = f(fg) = fg = h$ since $f^2 = f$. Thus $(f+1)h = 0$ or $f(g+h) = 0$.

Let $(s_0, s_1, \cdots, s_{n-1})$ be the initial states of LFSRs. The output bit z_i is given by

$$z_i = f(L^i(s_0, s_1, \cdots, s_{n-1})),$$

for a linear transformation L. L might be the recurrence relation of LFSRs. Then the algebraic attack is mounted according to following steps.

(1) Assume that there exists an annihilator of degree $d \ll n/2$.
   - If $fg = 0$, we have $g(L^i(s_0, s_1, \cdots, s_{n-1})) = 0$ when $z_i = 1$.
   - If $(f+1)h = 0$, we have $h(L^i(s_0, s_1, \cdots, s_{n-1})) = 0$ when $z_i = 0$.

(2) There are $T \approx \binom{n}{d}$ monomials of degree at most d with n variables s_js. Therefore we have a system of equations of degree d with n variables if we know sufficiently large number of the output bits.

(3) Consider each monomial as a new variable. Then we have a system of linear equations with T variables.

(4) If we are given $R > \binom{n}{d}$ equations, we can find the initial states $(s_0, s_1, \cdots, s_{n-1})$ by solving the system of linear equations using Gauss elimination method.

Therefore the complexity of the algebraic attacks is related to the degree of an annihilator. The *algebraic immunity* of f is defined by the minimum value of d such that f or $f+1$ admits an annihilator of degree d.

In [17], Meier and others proposed an algorithm to decide whether the given Boolean function admits an annihilator of degree $\leq d$. But the algorithm is infeasible for $d \geq 6$ when $n \geq 32$. It is still open problem to compute the algebraic immunity of a given Boolean function. They also showed that a random Boolean function likely has the maximal algebraic immunity.

4. Generic Attacks

In the previous section, we explain that the purpose of algebraic attacks is to find low degree annihilators. In this section, we describe generic algebraic attacks that guarantee the existence of an algebraic equation between the initial secret key bits and the output key stream bits.

4.1. **Stream Ciphers with LFSRs.**

Courtois and Meier proposed a method of generating low degree equations by multiplying the initial Boolean function by well chosen multivariate polynomials.

Theorem 1 ([9]). *Let f be any Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$. Then there is a Boolean function g of degree at most $\lfloor n/2 \rfloor$ such that fg is of degree at most $\lceil n/2 \rceil$.*

Proof. Let A be set of all the monomials of degree up to $\lceil n/2 \rceil$ and B be set of all the monomials of degree up to $\lfloor n/2 \rfloor$ multiplied by f. Then we have

$$|A| + |B| = \sum_{i=0}^{\lceil n/2 \rceil} \binom{n}{i} + \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{n}{i} = \sum_{i=0}^{n} \binom{n}{i} + \binom{n}{\lfloor n/2 \rfloor} > 2^n.$$

Since the rank of Boolean functions with n variables is n, there exist some linear dependencies. □

The above theorem shows that for any stream cipher with linear feedback, for which the nonlinear filter uses n variables, it is possible to generate an equation of degree at most $\lceil n/2 \rceil$. But it states the worst case. For a specific cipher, there might exist an equation of lower degree than the upper bound given in Theorem 1. (See Section 5)

4.2. **Stream Ciphers with Memories.** As explained before, there is a tradeoff between classical criteria for a Boolean function. Thus one of the solution for overcoming tradeoffs is to use a stateful function instead of a (memoryless) Boolean function.

A (k, l)-combiner consists of k input bits from k LFSRs, l memory bits, and one output bit. The memory bits are initialized to c^1. For each clock $t \geq 1$, the LFSRs produce k parallel bits and the combiner produces the t-th key stream bit z^t and changes the inner state of memory to c^{t+1}. (See Figure 2)

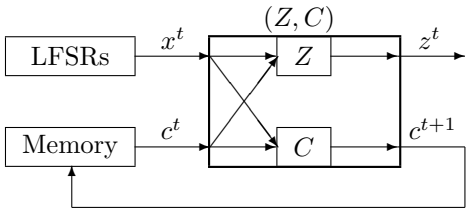

FIGURE 2. A (k, l)-combiner

Stream ciphers that utilize memories were at first thought to be much more resistant to algebraic attacks, but soon Armnecht and Krause showed that even these cases were subject to these attacks.

Theorem 2 ([2]). *Let $\mathcal{C} = (Z, C)$ be a (k, l)-combiner. Then for each $r > l$ there is a z-relation of degree $\lceil (k(l+1))/2 \rceil$ for $\mathcal{C}$ for some $z \in \{0, 1\}^r$.*

We note that Theorem 2 is extended to the case of m outputs in [6].

4.3. **Fast Algebraic Attacks.** The previous two sections focus on finding a system of algebraic equations in the initial secret key bits and the output bits. This step can be proceeded as a precomputation, that is the attacker must find these equations before the attacking the target cipher.

If enough low degree equations and known output key bits are given, then the initial secret key can be recovered by solving the system of equations in the next step. Usually, if we find lower algebraic equations, then the attack runs faster. Thus the purpose of algebraic attacks is to find low degree equations.

Courtois introduced fast algebraic attacks in [7] to decrease the degree of the system of equations obtained in the previous step. The fast algebraic attack proceeds as follows:

1. Assume that we already find *ad-hoc* algebraic equations. Write it in the following form:
$$\mathsf{Left}^t(\mathcal{K}) = \mathsf{Right}^t(\mathcal{K}, \mathcal{Z}),$$
where $\mathcal{K}$ is the initial secret key bits and $\mathcal{Z}$ is output key stream bits. The leftside $\mathsf{Left}^t(\mathcal{K})$ of the equation is only dependent on the initial secret key. Let $d = \deg(\mathsf{Left})$ and $e = \deg(\mathsf{Right})$ with respect to $\mathcal{K}$. Assume $e < d$.
2. Assume that the attacker knows a vector $\alpha = (\alpha_0, \cdots, \alpha_{D-1}) \in \{0,1\}^D$ such that
$$\sum_{i=0}^{D-1} \alpha_i \mathsf{Left}^{t+i}(\mathcal{K}) = 0. \tag{1}$$
3. We get an equation of lower degree $e < d$ in $\mathcal{K}$ and $\mathcal{Z}$ as follows:
$$\sum_{i=0}^{D-1} \alpha_i \mathsf{Right}^{t+i}(\mathcal{K}, \mathcal{Z}) = 0. \tag{2}$$
4. Apply the general algebraic attacks to the above equation.

4.3.1. *How to find the coefficients?* There are two problems to apply fast algebraic attacks. The first problem is how to find a vector α satisfying Equation (1). Courtois proposed to eliminate all high degree monomials independent of the output key stream bits using the Berlekamp-Massey algorithm [7]. At first, choose a random key $\hat{\mathcal{K}}$ and compute $\hat{z}^t = L^t(\hat{\mathcal{K}})$ for $t = 1, \cdots, 2D$. Then apply the Berlekamp-Massey algorithm to find a vector α such that
$$\sum_{i=0}^{D-1} \alpha_i \mathsf{Left}^{t+i}(\hat{\mathcal{K}}) = 0. \tag{3}$$

The Berlekamp-Massey algorithm finds a vector with the smallest value of D satisfying Equation (3). In general, the exact value of D is not known but an upper bound is the maximum number of different monomials occurring. Thus $D \leq \sum_{i=0}^{d} \binom{n}{i}$. The complexity of the Berlekamp-Massey algorithm is $O(D^2)$.

However, this is not justified in the original paper [7]. Armknecht proved the correctness of the algorithm under *reasonable* assumptions [1]. The assumptions apply to a large class of LFSR-based stream ciphers. For example, let $m_i(x) \in \mathbb{F}_2[x]$ $(1 \leq i \leq k)$ be the minimal polynomials of the used LFSRs such that the roots are all pairwise distinct and non-zero. If the degrees of the minimal polynomials are pairwise co-prime, then the assumptions are satisfied.

4.3.2. *The complexity of the substitution step.* The second problem is to make a system of equations by substituting output key stream bits into the obtained equation (2) in the previous step. In [7] and [1], the complexity of this step is underestimated as only $O(DE)$ where $E(\approx \sum_{i=0}^{e} \binom{n}{i})$ is the size of the second system of equations (2). However simple substitution would require a complexity of $O(DE^2)$. In some cases, the complexity of the fast algebraic attack using simple substitution is rather larger than that of the original algebraic attack since the complexity of the substitution is relatively high.

Hawkes and Rose showed that the complexity of the substitution step can be decreased to $O(ED \log_2 D)$ by applying the fast Fourier transform (FFT) [12]. Hence the fast algebraic attack would be *faster* than the original attack.

5. Attacked Stream Ciphers

In this section, we review some stream ciphers attacked by the algebraic attack.

5.1. **Toyocrypt.** Toyocrypt was a stream cipher submitted to the Japanese government Cryptrec call for cryptographic primitives. It is a nonlinear filter generator with a single LFSR of length 128. The Boolean function is of the form:

$$f(s_0, \cdots, s_{127}) = s_{127} + \sum_{i=0}^{62} s_i s_{\alpha_i} + s_{10}s_{23}s_{32}s_{42} + s_1 s_2 s_9 s_{12} s_{18} s_{20} s_{23} s_{25} s_{26} s_{28} s_{33} s_{38} s_{41} s_{42} s_{51} s_{53} s_{59} + \prod_{i=0}^{62} s_i,$$

with $\{\alpha_0, \cdots, \alpha_{62}\}$ being some permutation of the set $\{63, \cdots, 125\}$.

We can see that the parts of degree 4, 17 and 63 contain a common factor $s_{23}s_{42}$. Hence $f \cdot (s_{23} + 1)$ is an equation of degree 3 since the monomials divisible by s_{23} of f cancel out. In the same manner, $f \cdot (s_{42} + 1)$ is an equation of degree 3. This means that the algebraic immunity of f is at most 3.

Now we can apply a simple linearization attack described in Section 3. For each output key stream bit, we obtain two equations of degree 3 in the initial secrete key bits. The number of monomials of degree at most 3 is $T \approx \binom{128}{3} \simeq 2^{18.4}$. Hence the attack works if $T/2 \approx 2^{17.4}$ output key stream bits are given. The time complexity of the attack is roughly $T^w \approx 2^{52}$ where $w = \log_2 7$.

5.2. **LILI-128.** LILI-128 was a stream cipher submitted to Nessie European call for cryptographic primitives.[2] It is a nonlinear filter generator with two LFSRs. The first LFSR of length 39 is used to clock the second LFSR of length 89. (See Figure 3.)

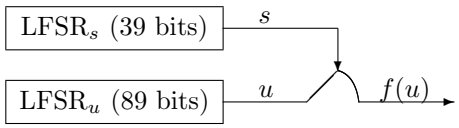

Figure 3. The structure of LILI-128

The output function f of LILI-128 has a degree 6 with 10 variables. By Theorem 1, there is an equation g of degree at most 5 such that the degree of fg is at most 5.

In fact, the part of degree 5 and 6 of f can be factored as follows:
$x_7x_9(x_3x_8x_{10} + x_4x_6x_8 + x_4x_6x_{10} + x_4x_8x_{10} + x_5x_6x_8 + x_5x_6x_{10} + x_4x_6x_8x_{10} + x_5x_6x_8x_{10})$.

[2]Nessie project did not select any stream ciphers as their portfolio of cryptographic primitives in 2003.

Hence the degree of $f \cdot (x_7 + 1)$ or $f \cdot (x_9 + 1)$ is 5. Furthermore the part of degree 5 and 4 of $f \cdot (x_9 + 1)$ can still be factored as follows:
$x_{10}(x_3x_7x_8x_9 + x_5x_7x_8x_9 + x_3x_7x_8 + x_3x_8x_9 + x_4x_7x_9 + x_4x_8x_9 + x_5x_7x_8 + x_5x_7x_9 + x_6x_7x_9)$.

This means that the degree of $f \cdot (x_9+1)(x_{10}+1)$ is only 4. In [9], Courtois and Meier have searched all low degree polynomials g such that fg is also of low degree as the following table 1.

TABLE 1. The number of linearly independent g such that fg is of low degree

Degree of g	10	1	2	3	4	10
Degree of fg	3	4	4	4	4	4
# of g	0	0	4	8	14	14

The first column of Table 1 means that there are no g of degree at most 10 such that fg is of degree at most 3. According to Table 1, the algebraic immunity of f is 4.

In order to apply the algebraic attack to LILI-128, we should guess the initial state of the clocking LFSR. Thus the complexity is multiplied by 2^{39}. Then we can obtain 14 multivariate equations of degree 4 in the initial secret key bits for each output key stream bit. Following Section 3, we have $T \approx \binom{89}{4} \simeq 2^{21.2}$ monomials, and we need $m = T/14 \simeq 2^{18}$ key stream bits. The time complexity of the attack is roughly $T^w \approx 2^{98.6}$ where $w = \log_2 7$.

5.3. E_0 **in Bluetooth.** E_0 is the stream cipher as a part of the Bluetooth encryption scheme used for wireless communication. It can be considered as a modified summation generator using 4 LFSRs and 4 memory bits, that is, E_0 is a (4,4)-combiner.

Let $x^t = (x_1^t, x_2^t, x_3^t, x_4^t)$ be the output of the 4 LFSRs and $c^t = (q^t, p^t, q^{t-1}, p^{t-1})$ be the memory bits at time t. Then the output key stream bit of E_0 is as follows:

$$\begin{aligned} z^t &= x_1^t \oplus x_2^t \oplus x_3^t \oplus x_4^t \oplus p^t \\ q^{t+1} &= s_1^{t+1} \oplus q^t \oplus p^{t-1} \\ p^{t+1} &= s_0^{t+1} \oplus q^{t-1} \oplus p^t \oplus p^{t-1} \end{aligned}$$

where

$$(s_1^{t+1}, s_0^{t+1}) = \left\lfloor \frac{x_1^t + x_2^t + x_3^t + x_4^t + 2 \cdot q^t + p^t}{2} \right\rfloor .$$

Hence there exists an equation of degree 10 by Theorem 2. So the complexity of the naïve algebraic attack is greater than that of the exhaustive search attack when $n = 128$. Armknecht and Krause showed that there exists an equation of degree 4

using 4 consecutive output key stream bits as follows [2].

$$\begin{aligned}0 =& 1 \oplus z^{t-1} \oplus z^t \oplus z^{t+1} \oplus z^{t+2} \\ & \oplus S_1^t(z^t z^{t+2} \oplus z^t z^{t+1} \oplus z^t z^{t-1} \oplus z^{t-1} \oplus z^{t+1} \oplus z^{t+2} \oplus 1) \\ & \oplus S_2^t(1 \oplus z^{t-1} \oplus z^t \oplus z^{t+1} \oplus z^{t+2}) \oplus S_3^t z^t \oplus S_4^t \\ & \oplus S_1^{t-1} \oplus S_1^{t-1} S_1^t (1 \oplus z^t) \oplus S_1^{t-1} S_2^t \\ & \oplus S_1^{t+1} z^{t+1} \oplus S_1^{t+1} S_1^t z^{t+1} (1 \oplus z^t) \oplus S_1^{t+1} S_2^t z^{t+1} \\ & \oplus S_2^{t+1} \oplus S_2^{t+1} S_1^t (1 \oplus z^t) \oplus S_2^{t+1} S_2^t \\ & \oplus S_1^{t+2} \oplus S_1^{t+2} S_1^t (1 \oplus z^t) \oplus S_1^{t+2} S_2^t\end{aligned}$$

where S_i^t is the symmetric Boolean function of degree i at clock t.

Hence we can obtain one equation for each four consecutive output key stream bits. Following Section 3, there are $T \approx \binom{128}{4} \simeq 2^{23.4}$ monomials, and we need $(T+3) \simeq 2^{23.4}$ consecutive key stream bits. The time complexity of the attack is roughly $T^w \approx 2^{65.5}$ where $w = \log_2 7$.

5.4. **Summation Generators.** The summation generator proposed by Ruepel [18] is a nonlinear combiner with memory. We consider a summation generator that uses n binary LFSRs (See Figure 4).

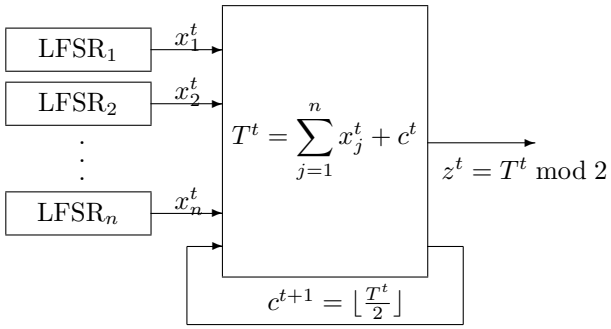

FIGURE 4. Structure of summation generators

The output of the j-th LFSR at time t is denoted by $x_j^t \in \{0,1\}$. It is known that the generator produces sequences whose period and correlation immunity are maximum, and whose linear complexity is conjectured to be close to the period. Hence it serves as a good building block for stream ciphers.

However, a correlation attack on the summation generator that uses two LFSRs was presented in [16], even though it is also stated in [16] that this attack is not plausible if there are more than two LFSRs in use. Another well known attack on summation generator is given by [14] and points in the opposite direction. It uses feedback carry shift registers (FCSR) to simulate the summation generator and indicates that for a fixed initial key size, breaking them into too many LFSRs will add to its weakness.

In [15], it was showed that for a summation generator that uses n LFSRs, an algebraic equation relating the key stream bits and LFSR output bits can be made

to be of degree less than or equal to $2^{\lceil \log_2 n \rceil}$, using $\lceil \log_2 n \rceil + 1$ consecutive key stream bits. This is much lower than the upper bound on the degree of algebraic equations that is guaranteed by the general works [2, 6]. It was also showed that the techniques of [7] can be applied to summation generators using 2^k LFSRs to reduce the effective degree of the equation further.

6. Conclusion

In this paper, we survey the recent progress of algebraic attacks on stream ciphers. Algebraic attacks are to find the secret key by solving algebraic equations involving the secret key and output key stream bits. Algebraic attacks are most powerful attacks on stream cipher base on linear feedback shift registers, because of the state update function is linear. For example, Toyocrypt, LILI-128, E_0 and summation generators are attacked. LFSR-based stream ciphers might be (potentially) vulnerable to algebraic attacks.

Recently several new non-LFSR-based stream ciphers are proposed. But they are hard to analyze the characteristic of the ciphers such as period. (This kind of strategy of designing is called *wild* strategy.) To design a good non-LFSR-based stream cipher is one of challenging problems.

References

[1] F. Armknecht, Improving fast algebraic attacks, *Fast Software Encryption (FSE) 2004*, LNCS 3017, Springer-Verlag, pp. 65–82, 2004.

[2] F. Armknecht and M. Krause, Algebraic attacks on combiners with memory, *Advances in Cryptology - Crypto 2003*, LNCS 2729, Springer-Verlag, pp. 162–175, 2003.

[3] J. Cheon and D.H. Lee, Resistance of S-boxes against algebraic attacks, *Fast Software Encryption (FSE) 2004*, LNCS 3017, Springer-Verlag, pp.83–94, 2004.

[4] N. Courtois, The security of Hidden Field Equations (HFE), *CT-RSA 2001*, LNCS 2020, Springer-Verlag, pp. 266–281, 2001.

[5] N. Courtois, Higher order correlation attacks, XL algorithm and Cryptanalysis of Toyocrypt, *ICISC 2002*, LNCS 2587, Springer-Verlag, pp. 182–199, 2002.

[6] N. Courtois, Algebraic attacks on combiners with memory and several outputs, E-print archive, 2003/125.

[7] N. Courtois, Fast algebraic attack on stream ciphers with linear feedback, *Advances in Cryptology - Crypto 2003*, LNCS 2729, Springer-Verlag, pp. 176–194, 2003.

[8] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, Solving overdefined multivariate equations, *Advances in Cryptology - Eurocrypt 2000*, LNCS 1807, Springer-Verlag, pp. 392–407, 2000.

[9] N. Courtois and W. Meier, Algebraic attacks on stream ciphers with linear feedback, *Advances in Cryptology - Eurocrypt 2003*, LNCS 2656, Springer-Verlag, pp. 345–359, 2003.

[10] N. Courtois and J. Pieprzyk, Cryptanalysis of block ciphers with overdefined systems of equations, *Asiacrypt 2002*, LNCS 2501, Springer-Verlag, pp. 267–287, 2002.

[11] K. Diem, The XL-algorithm and a conjecture from commutative algebra, *Asiacrypt'04*, LNCS, Springer-Verlag, to be published 2004.

[12] P. Hawkes and G. Rose, Rewriting variables: the complexity of fast algebraic attacks on stream ciphers, *Advances in Cryptology - Crypto 2004*, LNCS 3152, Springer-Verlag, pp.390–406, 2004.

[13] A. Kipnis and A. Shamir, Cryptanalysis of the HFE public key cryptosystem by relinearization, *Advances in Cryptoloy - Crypto'99*, LNCS 1666, Springer-Verlag, pp. 19–30, 1999.

[14] A. Klapper and M. Goresky, Cryptanalysis based on 2-adic rational approximation, *Advances in Cryptology - Crypto '95*, LNCS 963, Springer-Verlag, pp. 262–273, 1995.

[15] D.H. Lee, J. Kim, J. Hong, J. Han, and D. Moon, Algebraic attacks on summation generators, *Fast Software Encryption (FSE) 2004*, LNCS 3017, Springer-Verlag, pp. 34–48, 2004.

[16] W. Meier and O. Staffelbach, Correlation Properties of Combiners with Memory in Stream Cipher, *Journal of Cryptology*, vol.5, pp. 67–86, 1992.

[17] W. Meier, E. Pasalic, and C. Carlet, Algebraic attacks and decomposition of Boolean functions, *Advances in Cryptology - Eurocrypt 2004*, LNCS 3027, Springer-Verlag, pp.474–491, 2004.

[18] R.A. Rueppel, Correlation immunity and the summation generator, *Advances in Cryptology - Crypto'85*, LNCS 219, Springer-Verlag, pp. 260–272, 1985.

[19] P. Sarkar and S. Maitra, Nonlinearity bounds and constructions of resilient Boolean functions, *Advances in Cryptology - Crypto 2000*, LNCS 1880, Springer-Verlag, pp.515–532, 2000.

[20] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Trans. on Infor. Theory*, vol.IT-30, no.5, pp.776–780, 1985.

[21] B. Yang and J. Chen, Theoretical analysis of XL over small fields, *Information Security and Privacy (ACISP) 2004*, LNCS 3108, Springer-Verlag, pp. 277–288, 2004.

National Security Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea

E-mail address: dlee@etri.re.kr