Algebraic Countermeasure to Enhance the Improved Summation Generator with 2 Bit Memory

Md. Iftekhar Salam [a], Hoon-Jae Lee [b]

[a] Dept. of Ubiquitous IT, Graduate School of Design and IT,

Dongseo University, San 69-1 Jurye-2-dong, Sasang-gu, Busan 617-716, Korea

[b] Div. of Information Network Eng., School of Internet Engineering,

Dongseo University, San 69-1 Jurye-2-dong, Sasang-gu, Busan 617-716, Korea

E-mail: iftekharsalam@gmail.com, hjlee@dongseo.ac.kr

Corresponding Author: Hoon-Jae Lee

Abstract

Recently proposed algebraic attack has been shown to be very effective on several stream ciphers. In this paper, we have investigated the resistance of PingPong family of stream ciphers against algebraic attacks. This stream cipher was proposed in 2008 to enhance the security of the improved summation generator against the algebraic attack. In particular, we focus on the PingPong-128 stream cipher's resistance against algebraic attack in this paper. In our analysis, it is found that an algebraic attack on PingPong family of stream ciphers require much more operations compare to the exhaustive key search on the internal state of the LFSRs. It will be shown that due to the irregular and mutual clock controlling in PingPong stream cipher the degree of the generated equation tends to grow up with each successive clock which in turn increases the overall complexity of an algebraic attack. Along with the PingPong 128 stream cipher the other instances of PingPong family stream ciphers are also investigated against the algebraic attack. Our analysis shows that, PingPong family stream ciphers are highly resistant against the algebraic attack due to their mutual and irregular clocking function.

Keywords: Algebraic attack, cryptography, stream cipher, PingPong keystream generator

1. Introduction

Cryptography deals with the secrecy of transmitted data in a communication system. A secret key cryptosystem encrypts the original message into a ciphertext depending on the value of secret key. Linear feedback shift register (LFSR) based stream ciphers are one such secret key cryptosystem where the output from several LFSRs are combined by a nonlinear Boolean function to generate the keystream bits. The original message is encrypted by performing bit by bit XOR operation of the keystream and original message. The secret key is used to determine the initial state of most of the LFSR based stream cipher. According to the principle of kerckhoff, the secrecy of a cryptosystem depends on the secrecy of the key. A cryptosystem should build in such way so that an adversary (who has the knowledge about the cryptosystem) should not be able to determine the secret key faster than trying all possible keys (exhaustive search/ brute-force attack). Several analysing/ attacking methods exist in literature to recover the secret key of a cipher. Recently, a new type of structure dependent attack known as algebraic attack has been proposed which attempts to recover the initial internal state of a cryptographic system. The algebraic attack exploits a number of observed keystream bits to solve an over-defined system of multivariate equations to recover the secret

key of the cipher [1]. It is currently the fastest attack against many well-known LFSR based stream ciphers [2, 3]. Algebraic attack on stream cipher includes attacks on nonlinear filter generator [2], attacks on summation generator [4], attacks on combiners with memory [2], attacks on mutually clocked shift register [5], attacks on improved summation generator [6].

The idea of the algebraic attack is to recover the secret key bits by solving a system of nonlinear equation which relates the output keystream bits with the secret key of the cipher. Algebraic attack on LFSR based stream ciphers build up a valid relationship between the output keystream bits and the internal state. Based on the nonlinear combining function the attacker generates a system of nonlinear equation relating the internal states and the output keystream bits; and these nonlinear equations are then solved in an efficient way to determine the internal state of the LFSR. Generally, solving such a system of nonlinear equation is considered to be NP-complete even if all the equations are of degree 2. Therefore; it is difficult to find an efficient solver for solving these systems of equations. However, the circumstances will change dramatically if the system is over-defined. An over-defined system is one where there are more equations then the number of variables and such a system can be solved by method called linearization [7]. In the linearization method, the over-defined system of equation can be solved in polynomial time. However, this method requires knowledge about large number of keystream bits. Another method for solving the nonlinear equation is Gröbner Bases. There exist several other methods to solve such system of nonlinear equation, however; to date the complexity of the solution can be computed only for the linearization approach. The complexity of an algebraic attack can be reduced significantly if the degree of the generated system of equation is low. Therefore, it is necessary to find low degree equation for feasible algebraic attack. The fast algebraic attack was proposed for this which works in a similar way as the standard algebraic attack but with a reduced degree of system of equation [8]. There exist several algorithms to reduce the degree of the equation [2, 9]. The steps of an efficient algebraic attack is summed up below

- Set up a valid relationship between the internal state of the cipher and the output keystream bits for all time instances.
- The system of equations generated should be of low degree for a successful algebraic attack. Attacker should try to reduce the degree of the generated equations as far as possible. The reduction of degree will significantly reduce the complexity of solving the system of equation and hence makes it easier to recover the internal state of the cipher.
- Solve the generated system of equation efficiently. For solving the system of equation method like linearization can be used.

The objective of this paper is to analyse the resistance strength of PingPong family of stream ciphers against algebraic attack. In 2000 Lee et al. proposed an improved summation generator with 2 bit memory [10] to provide resistance against the correlation attack. However, it was found that the improved summation generator was still susceptible to several attacks including the algebraic attack [6, 11]. PingPong keystream generator [12] was proposed to overcome these security weaknesses of the improved summation generator. This is an irregularly and mutually clocked stream cipher which exploits two LFSR output bits and two memory output bits with a nonlinear combiner to produce the keystream bits. It will be shown that, due to the irregular and mutual clocking of the LFSR's the degree of generated algebraic equations tend to increase which increases the complexity of solving such system of equations and makes it infeasible to make an attack based on the algebraic analysis.

The rest of the paper is organized as follows. Section 2 provides an overview of PingPong family of stream ciphers. Section 3 provides the algebraic analysis of PingPong 128. Resistance analysis of other instances of PingPong stream cipher is discussed in section 4.

Section 5 discusses the approach of direct recovery of the key bits based on algebraic analysis. Finally, section 6 concludes this paper.

2. Background of PingPong Family of Stream Ciphers

PingPong family of keystream generator is based on two mutually clock controlled linear feedback shift registers LFSR A (L_a) and LFSR B (L_b), and two memory bits. The output key stream is generated by combining the output of the two LFSR sequences and the memory sequences with a nonlinear combining function. The two LFSRs used in the PingPong generator are of length l bits and m bits respectively. Along with the memory bits, PingPong family of stream cipher has an internal state of $l + m$ bits. It uses two primitive polynomials, $P_a(x)$ and $P_b(x)$ which defines the tap connection of LFSR A (L_a) and LFSR B (L_b).

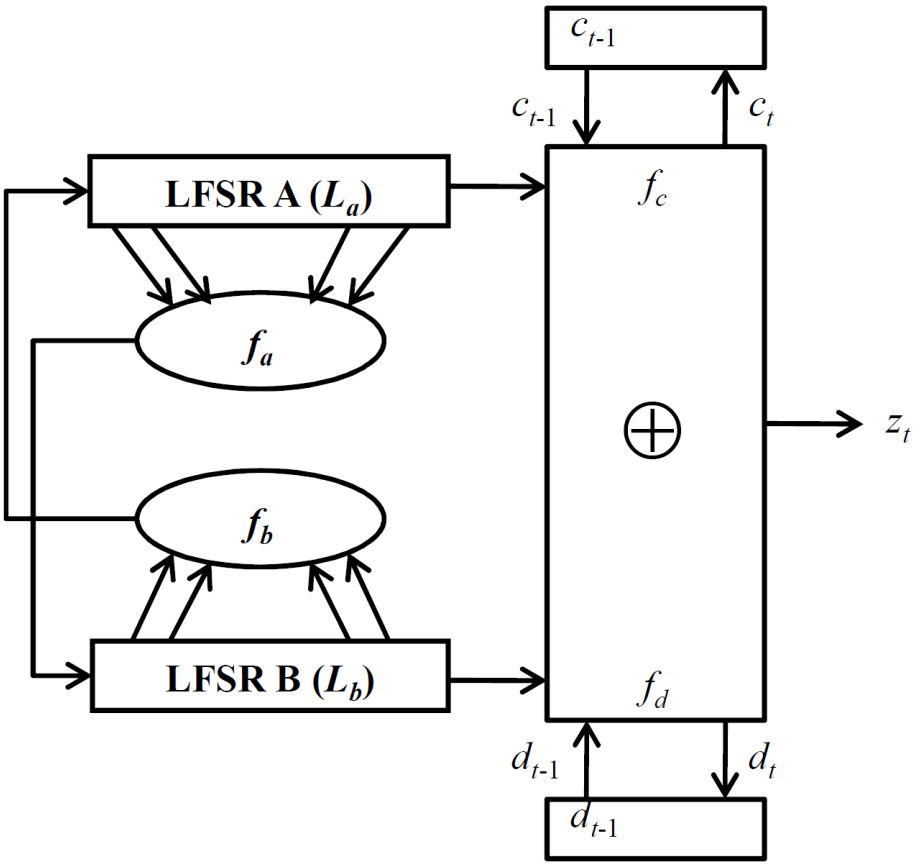

Figure 1: Key generation in PingPong family of stream ciphers

The working principle of PingPong keystream generator is shown in figure 1. At time instant t, the output of the LFSR A (L_a) and LFSR B (L_b) are denoted by a_t and b_t respectively while c_t and d_t represents the memory bit. The memory bits are defined by the function f_c and f_d respectively and at time t these functions are defined as

$$c_t = f_c(a_t, b_t, c_{t-1}) = a_t b_t \oplus (a_t \oplus b_t) c_{t-1} \quad (1)$$

$$d_t = f_d(a_t, b_t, d_{t-1}) = b_t \oplus (a_t \oplus b_t) d_{t-1} \quad (2)$$

The output of the keystream generator is obtained by combining the output of the LFSR sequences and the memory bit sequences. The output sequence at time t is denoted by z_t, and defined as

$$z_t = a_t \oplus b_t \oplus c_{t-1} \oplus d_{t-1} \quad (3)$$

In PingPong family of stream ciphers, two linear feedback shift register LFSR A (L_a) and LFSR B (L_b) are mutually clock controlled by the functions $f_b(L_b)$ and $f_a(L_a)$ respectively. The mutual clock controlled structure is used to provide irregular clocking of the LFSRs which increases the nonlinearity of the output key stream. The clock controlling mechanism of LFSR A (L_a) is defined by a function $f_b(L_b)$ which takes input from two random register's value of LFSR B (L_b) at time instant t. Clock controlling of LFSR B (L_b) is also performed in a similar manner where the function $f_a(L_a)$ takes input from two random register's value of LFSR A (L_a). Depending on the value of the registers at time instant t, the LFSR's are clocked between 1 to 4 times.

For the initial key loading process k bit key and k bit initial vector (IV) are used to determine the initial state of the LFSRs. The generator is used twice to determine the initial internal state of the LFSRs. The starting state of the l bits for LFSR A is obtained by simply XOR-ing the k bit binary string with the k bit IV. The starting state of the m bits LFSR B is obtained by embedding the k bit key in a (k+1) bit word and shifting 1 bit left, and then XOR-ing that with the IV embedded in a (k+1) bit word with a leading zero. The generator is then run to produce an output string of length l+m bits. For the second iteration, the first l bits of the output is used to fill up the contents of LFSR A and the rest m bits are used to fill up the contents of LFSR B. The cipher is then run again for second time to produce an output string of length l+m bits. Similar, to the first iteration, the first l bits of the output is used to fill up the internal initial state of LFSR A and the rest m bits are used to fill up the internal initial state of LFSR B. The generator then can be used to produce keystream bits.

2.1 PingPong-128 Keystream Generator

PingPong-128 [12] is a member of the PingPong family keystream generator which uses LFSR A (L_a) and LFSR B (L_b) of size 127 and 129 bits respectively and has a key size of 128 bits. These 128 bits of key and an initial vector of 128 bits are combined to fill up the 256 bits internal state. PingPong-128 uses two primitive polynomials, $P_a(x)$ and $P_b(x)$ which are given below

$$P_a(x) = x^{127} \oplus x^{109} \oplus x^{91} \oplus x^{84} \oplus x^{73} \oplus x^{67} \oplus x^{66} \oplus x^{63} \oplus x^{56} \oplus x^{55} \oplus x^{48} \oplus x^{45} \oplus x^{42} \oplus x^{41} \oplus x^{37} \oplus x^{34} \oplus x^{30} \oplus x^{27} \oplus x^{23} \oplus x^{21} \oplus x^{20} \oplus x^{19} \oplus x^{16} \oplus x^{13} \oplus x^{12} \oplus x^{7} \oplus x^{6} \oplus x^{2} \oplus x^{1} \oplus 1 \quad (4)$$

$$P_b(x) = x^{129} \oplus x^{125} \oplus x^{121} \oplus x^{117} \oplus x^{113} \oplus x^{109} \oplus x^{105} \oplus x^{101} \oplus x^{97} \oplus x^{93} \oplus x^{89} \oplus x^{85} \oplus x^{81} \oplus x^{77} \oplus x^{73} \oplus x^{69} \oplus x^{65} \oplus x^{61} \oplus x^{57} \oplus x^{53} \oplus x^{49} \oplus x^{45} \oplus x^{41} \oplus x^{37} \oplus x^{33} \oplus x^{29} \oplus x^{25} \oplus x^{21} \oplus x^{17} \oplus x^{13} \oplus x^{9} \oplus x^{5} \oplus 1 \quad (5)$$

The feedback connection of LFSR A (L_a) and LFSR B (L_b) is determined by the primitive polynomials, $P_a(x)$ and $P_b(x)$ respectively. Since primitive polynomial is used for the feedback connection, both of the LFSRs generate maximal length sequence. LFSR A (L_a) has a period of $2^{127}-1$ and LFSR B (L_b) has a period of 2^{129} - 1. The two clock controlled functions of PingPong 128 are defined as follows

$$f_a(L_a) = 2L_{a42}(t) + L_{a85}(t) + 1 \quad (6)$$

$$f_b(L_b) = 2L_{b43}(t) + L_{b86}(t) + 1 \quad (7)$$

As can be seen from equation (6), the clock controlling function $f_a(L_a)$ takes input from the 42^{nd} and 85^{th} register of LFSR A (L_a) at time instant *t*. Similarly, equation (7) describes that the clock controlling function $f_b(L_b)$ takes input from the 43^{rd} and 86^{th} register of LFSR B (L_b) at time instant *t*.

2.2 PingPong-192 Keystream Generator

PingPong-192 keystream generator uses two LFSRs defined as LFSR A (L_a) and LFSR B (L_b) of size 191 and 193 bits respectively and has a key size of 192bits. These 192 bits of key are combined with the initial vector to fill up the internal states of the LFSR. Two primitive polynomials are used for the feedback connection which generates maximal length sequence for both LFSRs. LFSR A (L_a) has a period of 2^{191} - 1 and LFSR B (L_b) has a period of 2^{193} - 1. The two clock controlled functions of PingPong-192 are defined as follows

$$f_a(L_a) = 2L_{a64}(t) + L_{a127}(t) + 1 \tag{8}$$

$$f_b(L_b) = 2L_{b65}(t) + L_{b128}(t) + 1 \tag{9}$$

As can be seen from equation (8), the clock controlling function $f_a(L_a)$ takes input from the 64^{th} and 127^{th} register of LFSR A (L_a) at time instant *t*. Similarly, equation (9) describes that the clock controlling function $f_b(L_b)$ takes input from the 65^{th} and 128^{th} register of LFSR B (L_b) at time instant *t*.

2.3 PingPong 256 Keystream Generator

The two LFSRs used for PingPong-256 keystream generator are defined as LFSR A (L_a) and LFSR B (L_b) and has a size of 255 bits and 257 bits respectively. A key size of 256 bits is combined with the initial vector to fill up the internal state of these two LFSR. Two primitive polynomials are defined for the feedback connection which generates maximal length sequence for both of the LFSRs. LFSR A (L_a) has a period of $2^{255}-1$ and LFSR B (L_b) has a period of $2^{257}-1$. The two clock controlled functions of PingPong-256 are defined as follows

$$f_a(L_a) = 2L_{a85}(t) + L_{a171}(t) + 1 \tag{10}$$

$$f_b(L_b) = 2L_{b86}(t) + L_{b173}(t) + 1 \tag{11}$$

As can be seen from equation (10), the clock controlling function $f_a(L_a)$ takes input from the 85^{th} and 171^{st} register of LFSR A (L_a) at time instant *t*. Similarly, equation (11) describes that the clock controlling function $f_b(L_b)$ takes input from the 86^{th} and 173^{rd} register of LFSR B (L_b) at time instant *t*.

3. Algebraic Analysis of PingPong-128

In PingPong 128, the output of the two mutually clock controlled LFSR's are combined with two memory bits to compute the keystream bit. In order to provide an analysis based on algebraic attack we need to get rid of the memory bits. By using equation (1) and (2) the update functions of the two memory bits c_t and d_t can be represented as

$$c_t = a_t b_t \oplus a_t c_{t-1} \oplus b_t c_{t-1} \tag{12}$$

$$d_t = b_t \oplus a_t d_{t-1} \oplus b_t d_{t-1} \tag{13}$$

In the following steps, we will show how to get rid of the memory bits and generate a relationship between the internal states and the output keystream bits. Adding equation (12) and (13) we get,

$$c_t \oplus d_t = a_t b_t \oplus b_t \oplus (a_t \oplus b_t)(c_{t-1} \oplus d_{t-1}) \tag{14}$$

By using equation (3), the memory bits at time instant *t*-1 can be written in terms of the output bits of the two LFSR and the keystream bit at time instant *t*.

$$c_{t-1} \oplus d_{t-1} = a_t \oplus b_t \oplus z_t \tag{15}$$

Substituting equation (15) into equation (14) we get,

$$c_t \oplus d_t = a_t b_t \oplus b_t \oplus (a_t \oplus b_t)(a_t \oplus b_t \oplus z_t) \tag{16}$$

At time instant *t*+1, the output of the keystream bits can be written as follows by using equation (3)

$$z_{t+1} = a_{t+1} \oplus b_{t+1} \oplus c_t \oplus d_t \tag{17}$$

Then substituting equation (16) into equation (17) we get

$$z_{t+1} = a_{t+1} \oplus b_{t+1} \oplus a_t b_t \oplus b_t \oplus (a_t \oplus b_t)(a_t \oplus b_t \oplus z_t) \tag{18}$$

It can be seen from equation (18), that a relationship has been formed between the keystream bits and the internal state of the LFSR without involving the memory bits. The equation has a degree of 2 and in this case, the total number of monomials expected to appear in the system is $M = \sum_{i=1}^{2} \binom{256}{i} \approx 2^{15}$. According to Strassen's algorithm [13] it requires at most 2^{45} operations to solve these system of equations. However, the irregular clocking of the system has not been taken into account in the above equation. The number of generated equations and the overall degree of the system increases drastically due to the mutual irregular clocking in PingPong-128. The following steps describe how to incorporate the clocking mechanism used in the PingPong into one equation.

As stated earlier, the clock controlling in PingPong is mutual and irregular. The two clock controlling functions of LFSR A (L_a) and LFSR B (L_b) is defined by equation (6) and (7) respectively. Both of these LFSRs are clocked between 1 to 4 times depending on the contents of the clock controlling bits at time instant *t*. A relationship can be obtained between the internal state of LFSR B (L_b) and the output of PingPong-128 by incorporating the clock controlling bits from LFSR A (L_a) as variables into LFSR B (L_b). Similarly, we need to incorporate the clock controlling bits from LFSR B (L_b) as variables into LFSR A (L_a). Based on equation (6) and (7), table 1 illustrates the clocking scheme by a binary truth table of all the clocking possibilities

Table 1: Clock control description of PingPong 128

LFSR B (L_b)			**LFSR A (L_a)**		
a_{42}^t	a_{85}^t	Number of clocking	b_{43}^t	b_{86}^t	Number of clocking
0	0	1	0	0	1
0	1	2	0	1	2
1	0	3	1	0	3
1	1	4	1	1	4

Incorporating the relationship between the clock controlling bits (a_{42}^t, a_{85}^t) as shown in table 1, the relationship between the number of clocking and the state of i^{th} register in LFSR B (L_b) can be represented in an algebraic expression as follows

$$B_i^{t+1} = \begin{cases} B_{i-1}^t & a_{42}^t = 0, a_{85}^t = 0 \\ B_{i-2}^t & a_{42}^t = 0, a_{85}^t = 1 \\ B_{i-3}^t & a_{42}^t = 1, a_{85}^t = 0 \\ B_{i-4}^t & a_{42}^t = 1, a_{85}^t = 1 \end{cases}$$

In this case, the internal state of the i^{th} register in LFSR B (L_b) can be represented as

$$B_i^{t+1} = (a_{42}^t \oplus 1)(a_{85}^t \oplus 1)B_{i-1}^t \oplus (a_{42}^t \oplus 1)a_{85}^t B_{i-2}^t \oplus a_{42}^t (a_{85}^t \oplus 1)B_{i-3}^t \oplus a_{42}^t a_{85}^t B_{i-4}^t \quad (19)$$

Similarly, the clock controlling bits (b_{43}^t, b_{86}^t) are incorporated to build a relationship between the number of clocking and the state of i^{th} register in LFSR A (L_a).

$$A_i^{t+1} = \begin{cases} A_{i-1}^t & b_{43}^t = 0, b_{86}^t = 0 \\ A_{i-2}^t & b_{43}^t = 0, b_{86}^t = 1 \\ A_{i-3}^t & b_{43}^t = 1, b_{86}^t = 0 \\ A_{i-4}^t & b_{43}^t = 1, b_{86}^t = 1 \end{cases}$$

In this case, the internal state of the i^{th} register in LFSR A (L_a) can be represented as

$$A_i^{t+1} = (b_{43}^t \oplus 1)(b_{86}^t \oplus 1)A_{i-1}^t \oplus (b_{43}^t \oplus 1)b_{86}^t A_{i-2}^t \oplus b_{43}^t (b_{86}^t \oplus 1)A_{i-3}^t \oplus b_{43}^t b_{86}^t A_{i-4}^t \quad (20)$$

It can be seen from equation (19) and (20) that the degree of the equation increases with each successive clock. Substituting the new representation of the outputs of LFSR A (L_a) and LFSR B (L_b) into equation (18) will result an equation of degree 6 for the first output bit. However, the degree keeps increasing with each successive clock because of the mutual clocking function of the LFSR. For a standard algebraic attack on mutually clock controlled linear feedback shift register [5] with size of l and m, the degree of the generated equation will be $l+m$ with a maximum of $\sum_{i=1}^{l+m} \binom{l+m}{i}$ monomials.

According to Strassen's algorithm, the complexity of solving such a system of equation with the linearization approach is at most $\left[\sum_{i=1}^{l+m} \binom{l+m}{i}\right]^{\omega}$, where $2.807 \le \omega \le 3$.

Following the above mentioned property, maximum possible degree of the generated system of equations for PingPong 128 is $d = l+m = 256$. In such case the maximum number of monomial expected to appear in the generated equation is $M = \sum_{i=1}^{d=l+m} \binom{256}{i} \approx 2^{256}$ and according to Strassen's algorithm the attack complexity of solving such system of equation can be computed as approximately $M^3 = 2^{768}$. Clearly, the attack complexity is much worse than the exhaustive key search and is an infeasible solution for practical scenario.

In the following we will discuss about the effectiveness of an attack by guessing the contents of one LFSR. It is worth to note that if the content of one LFSR is guessed in the current clocking mechanism, then the degree of the initially generated equation will be 3. However, this degree also tends to increase because of the nonlinear update of the clock control

function and can have a maximum degree of size equal to the length of the other LFSR size. In the guessing approach, the internal state of one LFSR is guessed and the internal state of the other LFSR is recovered by solving the low degree equation. For a mutually clock controlled stream cipher, guessing k bits of the internal state will reduce the overall degree of equations by k and the complexity of such algebraic attacks using guessing approach will be $(2^k - 1)\left[\sum_{i=1}^{(l+m-k)} \binom{l+m-k}{i}\right]^{\omega}$,where $2.807 \le \omega \le 3$. Following this, the attack complexity of PingPong 128 for such divide and conquer style attack is listed down in table 2.

Table 2: Attack complexity with the guessing approach

Guessed LFSR	Maximum degree of equations	Required keystream	Total attack complexity (approx.)
LFSR A (L_a)	d=129	2^{129}	$2^{127} \times 2^{387} = 2^{514}$
LFSR B (L_b)	d=127	2^{127}	$2^{129} \times 2^{381} = 2^{510}$

Since, both of the LFSRs are almost equal size, guessing either LFSR will therefore result in similar attack complexity. As illustrated in table 2, guessing the contents of LFSR A (L_a) will require solving equations with 2^{129} unknowns, while guessing LFSR B (L_b) ends up with equations having 2^{127} unknowns. Solving such system of equations requires a huge number of operations and moreover we need to consider the number of operations required for the guessing approach as well. In total, 2^{514} operations are required to recover the internal initial state when the contents of LFSR A (L_a) are guessed, whereas guessing LFSR B (L_b) requires 2^{510} operations. The guessing approach reduces the total attack complexity significantly compared to the standard algebraic attack; however still the attack complexity is much worse than the exhaustive key search attack. As well as it requires a huge amount of keystream bits. There, might exist some low degree multiples or annihilators to reduce the overall degree of the generated equations. However, the degree of these multiples also increases with time due to the nonlinear clock controlling function. Moreover, currently there exist no suitable algorithm to find low degree multiples for a system of equation having large number of variables.

Another alternative attack can proceed for the mutual clock controlling LFSR by preventing the degree accumulation as illustrated in [5]. In such case, the degree accumulation is prevented by introducing new variables for the register state at every clock. Since PingPong 128 is a mutually clock controlled generator the same method can be applied for the prevention of degree accumulation. For PingPong 128 it is required to introduce $256^2 = 65536$ number of variables and the same number of equation in the system. The degree of the monomials in such a system of equation will be 6. These equations can be solved by Gröbner based methods; however the complexity of solving such system of equation is unknown.

4. Resistance Analysis to Algebraic Attack for Other Instances of PingPong Family Stream Cipher

The basic structure and working principle for all of the PingPong family stream cipher is same. The main difference between the members of the PingPong family stream cipher is the length of the LFSR size. In this section, we analyze the resistance of other instances (e.g. PingPong 192, PingPong 256) of PingPong family stream cipher against algebraic attack.

Since the internal mechanism of all PingPong family stream cipher is same, therefore; we can use similar method described in section 3 to incorporate the algebraic representation for PingPong 192 and PingPong 256. For PingPong 192 the internal state of the i^{th} register for LFSR B (L_b) and LFSR A (L_a) can be represented by the following equations. Here, (a_{64}^t, a_{127}^t) represents the clock controlling bits for LFSR B (L_b) and (b_{65}^t, b_{128}^t) represents the clock controlling bits for LFSR A (L_a) in PingPong 192.

$$B_i^{t+1} = (a_{64}^t \oplus 1)(a_{127}^t \oplus 1)B_{i-1}^t \oplus (a_{64}^t \oplus 1)a_{127}^t B_{i-2}^t \oplus a_{64}^t (a_{127}^t \oplus 1)B_{i-3}^t \oplus a_{64}^t a_{127}^t B_{i-4}^t \quad (21)$$

$$A_i^{t+1} = (b_{65}^t \oplus 1)(b_{128}^t \oplus 1)A_{i-1}^t \oplus (b_{65}^t \oplus 1)b_{128}^t A_{i-2}^t \oplus b_{65}^t (b_{128}^t \oplus 1)A_{i-3}^t \oplus b_{65}^t b_{128}^t A_{i-4}^t \quad (22)$$

Similarly, for PingPong 256 the internal state of the i^{th} register for LFSR B (L_b) and LFSR A (L_a) can be represented by incorporating the corresponding clock controlling registers (a_{85}^t, a_{171}^t) and (b_{86}^t, b_{173}^t). The algebraic representation of the internal state of the LFSRs for PingPong 256 is shown in the following equations.

$$B_i^{t+1} = (a_{85}^t \oplus 1)(a_{171}^t \oplus 1)B_{i-1}^t \oplus (a_{85}^t \oplus 1)a_{171}^t B_{i-2}^t \oplus a_{85}^t (a_{171}^t \oplus 1)B_{i-3}^t \oplus a_{85}^t a_{171}^t B_{i-4}^t \quad (23)$$

$$A_i^{t+1} = (b_{86}^t \oplus 1)(b_{173}^t \oplus 1)A_{i-1}^t \oplus (b_{86}^t \oplus 1)b_{173}^t A_{i-2}^t \oplus b_{86}^t (b_{173}^t \oplus 1)A_{i-3}^t \oplus b_{86}^t b_{173}^t A_{i-4}^t \quad (24)$$

As seen from the above equations, the degree of the generated equations increases by one or two for each successive clock. For the first output bit, the degree of the overall generated equation is 6. However, this degree tends to increase due to the mutual irregular clock controlling function. Table 3 illustrates the attack complexity for mounting an attack on PingPong 192 and PingPong 256 based on the algebraic attack with guessing approach.

Table 3: Attack complexity with the guessing approach for different instances of PingPong stream ciphers

	Guessed LFSR	Maximum degree of equations	Required keystream	Total attack complexity (approx.)
PingPong 192	LFSR A (L_a)	d=193	2^{193}	$2^{191} \times 2^{579} = 2^{770}$
	LFSR B (L_b)	d=191	2^{191}	$2^{193} \times 2^{573} = 2^{766}$
PingPong 256	LFSR A (L_a)	d=257	2^{257}	$2^{255} \times 2^{771} = 2^{1026}$
	LFSR B (L_b)	d=255	2^{255}	$2^{257} \times 2^{765} = 2^{1022}$

As shown in table 3, the complexity of a divide and conquer style attack increases with the increase of the LFSR length. For both PingPong 192 and PingPong 256, the complexity of the attack is too high and infeasible in a practical scenario. To recover the internal state of the LFSR with such an attack will require thousands of years. It is also found that the complexity of the attack with guessing approach is much higher compare to the exhaustive key search attack and as the length of the LFSRs are increased the complexity also increases.

5. Direct Recovery of the Key Bits

In most of the modern stream ciphers the initial secret key is expanded (possibly incorporating with an initial vector (IV)) to fill up a comparatively large size of internal state. For instance, in PingPong 128 a key size of 128 bits are used with an IV of 128 bits to fill up the internal state of 256 bits. If the variables represented in the abovementioned algebraic attack represent the key bits instead of the initial internal state bits then there will be fewer

number of variables required to be considered. Following equation (19) and (20) it can be seen that if the variables in the equation represents the key bits rather than the initial internal state of the cipher, then the generated system of equations will have k numbers of variable (where k is the size of the key) with a maximum degree of k and a maximum number of $\sum_{i=1}^{k}\binom{k}{i}$ monomials. In order to recover the key directly, the attacker needs to compute the initial internal state in a pre-computation step where the internal state of the cipher will be represented in terms of the key k. Since, the degree of the output equation increases with each clock, therefore the internal contents of the register will be high degree functions of all the key bits. In the following, we first describe the procedure for algebraic attack against PingPong keystream generator to recover the key, k directly and then discuss about the attack complexity for such an attack.

Pre-computation Steps

Goal: To find the initial contents of the LFSRs in terms of the key bit variables
Input to the starting state of LFSR A: XOR [k, IV]
Input to the starting state of LFSR B: XOR [(0|k)<<1, (0|IV)]

Step 1: Run the generator to produce l+m bits of output string. Input the first l bits of the obtained output into LFSR A and the rest m bits into LFSR B.
Step 2: Run the generator second time to produce l+m bits of output string which defines the initial internal state of the cipher. Input the first l bits of the obtained output as the initial contents of the registers in LFSR A and the rest m bits as the initial contents of the LFSR B.

After Computation Steps

Goal: Recover the secret key of the cipher

Step 1: Generate $\sum_{i=1}^{k}\binom{k}{i}$ number of equations in terms of the initial state obtained in the pre-computation step.
Step 2: Insert the observed keystream bits into the corresponding identifier of each equation.
Step 3: Solve the generated system of equations using linearization approach to recover the secret key.
Step 4: Output the secret key bits

As illustrated in the above procedure, in the pre-computation phase the initial contents of the registers will be computed which generally are high degree function of the input key bit variable. Once the internal initial states of the LFSRs are defined, then the after-computation steps are similar to the procedure as discussed in the previous sections. It is noted that in the pre-computation stage the information for the IV are used which is known publicly. Since, the use of the information for IV and the key bits reduces the total number of variable in the

system of equations; therefore it will reduce the attack complexity for this approach as well. In the following, table 4 lists down the required attack complexity to recover the secret key bits directly for different instances of PingPong family stream cipher.

Table 4: Attack complexity for the direct recovery of the key bits of PingPong keystream generator

	Maximum degree	Required Keystream	Required Memory	Attack Complexity
PingPong 128	128	2^{128}	2^{256}	2^{384}
PingPong 192	192	2^{192}	2^{384}	2^{576}
PingPong 256	256	2^{256}	2^{512}	2^{768}

From table 4 it can be seen that the attack complexity has been significantly reduced compared to the algebraic attack mentioned in the previous sections. This is because in this scenario the generated system of equations can have a maximum degree of k whereas for the attack mentioned in the previous section can have a maximum degree of $l+m$. However, for this scenario as well the attack complexity is much worse than the exhaustive key search attack.

6. Conclusion

In this paper, we have analyzed the PingPong family of stream ciphers against algebraic attack. It is shown that, PingPong stream cipher is highly resistant to algebraic attack because of the mutual and irregular clock controlling. For PingPong stream ciphers, a standard algebraic attack with the linearization approach requires much more operations compare to the exhaustive key search attack. This is because the degree of the generated equations increases with each clocking due to the irregular mutual clock control function of the cipher. There might exist some low degree multiples to reduce the degree of the equations; however degree of such multiples also tends to increase. Moreover, currently there is no suitable algorithm to find low degree multiples for a system having large number of variables. Guessing some of the content of the LFSR is an approach which has been successfully used to attack on some of the stream ciphers. The feasibility of such an attack with the guessing approach is also examined for the PingPong stream cipher. The guessing approach reduces the complexity compare to the standard algebraic attack. On the other hand, if the algebraic attack uses the information of the initial vector to recover the secret key bits directly, then the complexity of the attack procedure can be further reduced; however it is still much worse than the exhaustive key search attack. Overall, we have shown that PingPong family of stream ciphers have enhanced the security of improved summation generator and provide high resistance against the algebraic attack due to their mutual irregular clock controlling function.

References

[1] N. Courtois, J. Pieprzyk, Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, in: Advances in Cryptology — ASIACRYPT 2002, Lecture Notes in Computer Science, vol. 2501, Springer Verlag, London, 2002, pp. 267-287.

[2] N. Courtois, W. Meier, Algebraic attacks on stream ciphers with linear feedback, in: Advances in Cryptology— EUROCRYPT 2003, Lecture Notes in Computer Science, vol. 2656, Springer, Berlin, 2003, pp. 345–359.

[3] F. Armknecht, M. Krause, Algebraic attacks on combiners with memory, in: Advances in Cryptology—CRYPTO 2003, Lecture Notes in Computer Science, vol. 2729, Springer, Berlin, 2003, pp. 162–175.

[4] D. H. Lee, J. Kim, J. Hong, J. W. Han, D. Moon, Algebraic attacks on summation generators, in: Fast Software Encryption 2004, Lecture Notes in Computer Science, vol. 3017, Springer Verlag, 2004, pp 34–48.

[5] S. Al Hinai, L. M. Batten and B. Colbert, Mutually clock-controlled feedback shift registers provide resistance to algebraic attacks, in: Information Security and Cryptology, Lecture Notes in Computer Science, vol. 4990, Springer Verlag, Berlin, 2008, pp. 201-215.

[6] D. Han and M. Lee, An algebraic attack on the improved summation generator with 2-bit memory, Information Processing Letters 93 (2005) 43 – 46.

[7] F. Armknecht, Algebraic Attacks on Stream Ciphers, in: Fourth European Congress on Computational Methods in Applied Sciences and Engineering, Finland, 24-28 July 2004.

[8] N. Courtois, Fast algebraic attacks on stream ciphers with linear feedback, in: Advances in Cryptology—CRYPTO 2003, Lecture Notes in Computer Science, vol. 2729, Springer, Berlin, 2003, pp. 176–194.

[9] X. Zhang, J. Pieprzyk, Y. Zheng, On algebraic immunity and annihilators, in: Information Security and Cryptology – ICISC 2006, Lecture Notes in Computer Science, vol. 4296, Springer, Berlin, 2006, pp. 65-80.

[10] H.J. Lee, S.J. Moon, On an improved summation generator with 2-bit memory, Signal Process. 80 (2000) 211–217.

[11] J. C. Mex-Perera, S. J. Shepherd, Cryptanalysis of a summation generator with 2-bit memory, Signal Process. 82 (2002) 2025-2028.

[12] H. J. Lee, K. Chen, PingPong-128, A New Stream Cipher for Ubiquitous Application, in: International Conference on Convergence Information Technology—ICCIT 2007, pp. 1893-1899.

[13] V. Strassen, Gaussian elimination is not optimal, Number. Math. 13 (1969) 254–356.