Analysis a Class of Stream Ciphers

Qasim M. hussein[1] , AKhalid F. Jassim[2]
[1] *Dept .of computer science, College of computer and Mathematics , Tikrit University , Tikrit , Iraq*
[2] *Dept. of Computer Science, Cihan University , Erbil, Iraq*
E-mail: kassimalshamry@yahoo.com
E-mail: Khalid.jassim@yahoo.com
(**Received:** 27/ 10 / 2010 ---- **Accepted:** 12 / 6 / 2012)

Abstract:
Stream cipher systems are widely used in secure sensitive information cryptography, since they are fast and lower error propogation. The designing of some of these systems depend on using more than one LFSRs with different length and combined function with different feedback polynomials. They are varing in security. This paper produces an anaysis about the standards to be considered in designing the LFSRs stream cipher systems that concerned in choosing shift registers and their linear complexity. Also it produces a method that is used to attack this class of stream cipher systems.

Keywords: Stream Cipher, Linear Complexity, Correlation Attack, Cryptanalysis.

1.Introduction

Stream ciphers are an important class of encryption algorithms. They encrypt individual characters of a plaintext message one at a time, using an encryption transformation which varies with time [4]. By contrast, block ciphers tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation [4].

Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate, and in some cases mandatory in some telecommunications applications, when buffering is limited or when characters must be individually processed as they are received. Because they have limited or no error propagation, stream ciphers may also be advantageous in situations where transmission errors are highly probable [6]. In section 2 , this paper explain the type of stream cipher, in section 3 introduces a linear complexity, section 4 concentrates on linear feedback shift register (LFSR). Section 5 introduces cryptanalysis of stream cipher with LFSRs.

2.Types of Stream Ciphers

Stream ciphers can be either symmetric-key or public-key. The focus of this research is symmetric-key stream ciphers.

2.1 One Time-Pad Cipher

Vernam cipher over the binary alphabet is defined by

$$C_i = m_i \oplus Z_i \quad \text{for } i = 1, 2, 3 \ldots, \qquad (1)$$
$$C, m, Z \in [0,1]$$

where m_1, m_2, m_3,… are the plaintext digits, Z_1, Z_2, Z_3,… are the key stream, C_1, C_2, C_3,… are the cipher text digits. Decryption is defined by

$$m_i = C_i \oplus Z_i \quad \text{for } i = 1, 2, 3 \ldots, \qquad (2)$$

If the key stream digits are generated independently and randomly, the Vernam cipher is called a one-time pad, and is unconditionally secure against a cipher text-only attack [8].

Definition of Entropy: Let Y be a random variable which takes on a finite set of values $y_1, y_2, \ldots, y_n$ with probability $P(Y = y_i)=p_i$,
where $0 \le p_i \le 1$ for each i, $1 \le i \le n$, and where $\sum_{i=1}^{n} p_i = 1$. The entropy of Y is a mathematical measure of the amount of information provided by an observation of Y. Equivalently, it is the uncertainty about the outcome before an observation of Y. The entropy or uncertainty of Y is defined as follows [6]:

$$H(Y) = \sum_{i=1}^{n} p_i \log (1 / p_i). \qquad (3)$$

Let M, C, and K are random variables respectively denoting the plaintext, ciphertext, and secret key, and H(Y) denotes the entropy function. Shannon proved that a necessary condition for a symmetric-key encryption scheme to be unconditionally secure is that $H(K) \ge H(M)$. That is, the uncertainty of the secret key must be at least as great as the uncertainty of the plaintext. If the key has bit length Len, and the key bits are chosen randomly and independently, the $H(K) = Len$, and Shannon's necessary condition for unconditional security becomes $Len \ge H(M)$. The one-time pad is unconditionally secure regardless of the statistical distribution of the plaintext [6]. An obvious drawback of the one-time pad is that the key should be as long as the plaintext, which increases the difficulty of key management. This motivates the design of stream ciphers [7] where the key stream is pseudo randomly generated from a smaller secret key, with the intent that the key stream appears random to a computationally bounded adversary. Such stream ciphers do not offer unconditional security (since $H(K) << H(M)$), but the hope is that they are computationally secure [8].

Stream ciphers are commonly classified as being synchronous or self-synchronous.

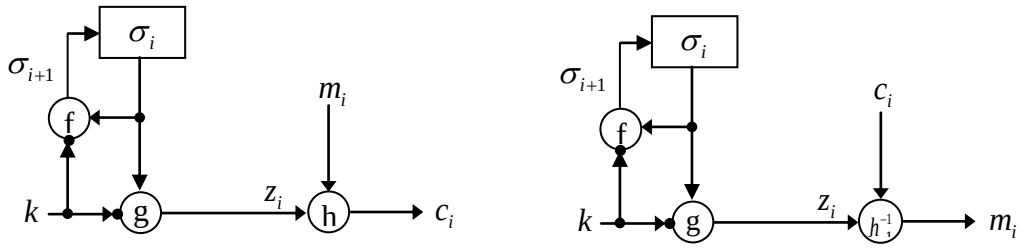
2.2 Synchronous Stream Ciphers

A synchronous stream cipher is one in which the key stream is generated independently of the plaintext message and of the cipher text [6]. The encryption process of a synchronous stream cipher can be described by the equations:

$$\sigma_{i+1} = f(\sigma_i, k) \qquad (4)$$
$$Z_i = g(\sigma_i, k) \qquad (5)$$
$$C_i = h(Z_i, m_i) \qquad (6)$$

where σ_0 is the initial state and may be determined from the key k, f is the next-state function, g is the

function which produces the key stream Z_i, and h is the output function which combines the key stream and plaintext m_i to produce cipher text C_i. The encryption and decryption process are shown in Figure 1.



(a) Encryption. **(b) Decryption.**

Figure 1: Synchronous Stream Cipher.

2.1 Properties of Synchronous Stream Ciphers

1. Synchronization requirements. In a synchronous stream cipher, both the sender and receiver must be synchronized using the same key and operating at the same position within that key to allow for proper decryption. If synchronization is lost due to cipher text digits being inserted or deleted during transmission, then decryption fails and can only be restored through additional technique for resynchronization.
2. No error propagation. A ciphertext digit that is modified (but not deleted) during transmission does not affect the decryption of other ciphertext digits.
3. Active attacks. As a consequence of property(1), the insertion, deletion, or replay of ciphertext digits by an active adversary causes immediate loss of synchronization, and hence might possibly be detected by the decrypter. As a consequence of property (2), an active adversary might possibly be able to make changes to selected ciphertext digits, and know exactly what affect these changes have on the plaintext.

2.3 Self Synchronizing Stream Ciphers

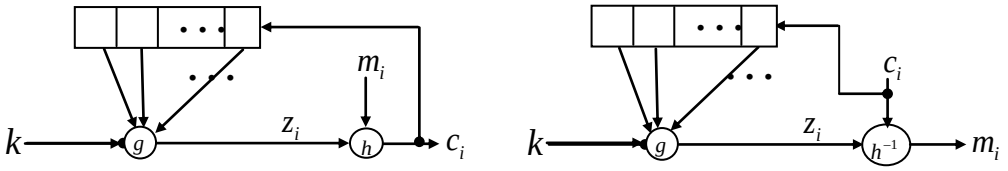
A self synchronizing or asynchronous stream cipher is one in which the key stream is generated as a function of the key and a fixed number of previous ciphertext digits[6]. The encryption function of a self synchronizing stream cipher can be described by the equations:

$$\sigma_i = (C_{i-t}, C_{i-t+1}, \ldots, C_{i-1}) \quad (7)$$

$$Z_i = g(\sigma_i, k) \quad (8)$$

$$C_i = h(Z_i, m_i) \quad (9)$$

where $\sigma_0 = (C_{-t}, C_{-t+1}, \ldots, C_{-1})$ is the (non secret) initial state, k is the key, g is the function which produces the key stream Z_i, and h is the output function which combines the key stream and plaintext m_i, to produce ciphertext C_i. The encryption and decryption processes are shown in Figure 2.



(a) Encryption. **(b) Decryption.**

Figure 2: Self synchronizing Stream Cipher.

2.3.1 Properties of Self Synchronizing Stream Ciphers [4]

Self synchronization. Self synchronization is possible if ciphertext digits are deleted or inserted, because the decryption mapping depends only on a fixed number of preceding ciphertext characters.

1. Limited error propagation. Suppose that the state of a self synchronization stream cipher depends on t previous ciphertext digits. If a single ciphertext digit is modified during transmission, then decryption of up to t subsequent ciphertext digits may be incorrect, after which correct decryption resumes.
2. Active attacks. property (2) implies that any modification of ciphertext digits by an active adversary causes several other ciphertext digits to be decrypted incorrectly, thereby improving the likelihood of being detected by the decryptor. As a consequence of property (1), it is more difficult to detect insertion, deletion, or replay of ciphertext digits by an active adversary.
3. Diffusion of plaintext statistics. Since each plaintext digit influences the entire following ciphertext, the statistical properties of the plaintext are dispersed through the ciphertext. Hence, self synchronizing stream ciphers may be more resistant than synchronous stream ciphers against attacks based on plaintext
4. redundancy.

3. Linear Complexity

The linear complexity of a finite binary sequence S^n, denoted $L(S^n)$, is the length of the shortest LFSR that generates a sequence having S^n as its first n terms [7]. Let S denotes an infinite sequence whose terms are $S_0,S_1,S_2, \ldots$ and S^n denotes a finite sequence of length n whose terms are $S_0,S_1, \ldots ,S_{n-1}$. An LFSR is said to generate a sequence S if there is some initial state for which the output sequence of the LFSR is S. Similarly, an LFSR is said to generate a finite sequence S^n if there is some initial state for which the output sequence of the LFSR has S^n as its first n terms [7]. Then the linear complexity of an infinite binary sequence S, denoted L(S), is defined as follows[7]:

- If S the zero sequence S = 0,0,0, … , then L(S) = 0 ;
- If no LFSR generates S, then $L(S) = \infty$;
- Otherwise, L(S) is the length of the shortest LFSR that generates S.

3.1 Properties of Linear complexity [2]

Let S and a be binary sequences. Then

- For any $n \geq 1$, the linear complexity of the subsequence S^n satisfies $0 \leq L(S^n) \leq n$.
- $L(S^n) = 0$ if and only if S^n is the zero sequence of length n.
- $L(S^n) = n$ if and only if $S^n = 0,0,0, \ldots, 1$.
- If S is periodic with period N, then $L(S) \leq N$.
- $L(S \oplus a) \leq L(S) + L(a)$, where $(S \oplus a)$ denotes the bitwise XOR of S and a.

4. Stream Ciphers with LFSRs

The Linear Feedback Shift Registers (LFSRs) are widely used in key stream generators because they are well suited for hardware implementation, produce sequences with large periods and good statistical properties, and are readily analyzed using algebraic techniques[1,2]. Unfortunately, the output sequences of LFSRs are also easily predictable as follows. Suppose that the output sequence S of an LFSR has linear complexity L. The connection polynomial C(D) of an LFSR of length L which generates S can be efficiently determined using the Berlekamp-Massey algorithm from any short subsequence t of S having length at least n = 2L. Having determined C(D), the LFSR (L, C(D)) can then be initialized with any substring of t having length L, and used to generate the remainder of the sequence S [7].

4.1 Use of LFSRs in Key stream Generators

Since a well designed system should be secure against known-plaintext attacks, an LFSR should never be used by itself as a keystream generator. Nevertheless, LFSRs are desirable because of their very low implementation costs. Three general methodologies for destroying the linearity properties of LFSRs are used [6]:

- Using a nonlinear combining function on the outputs of several LFSRs.
- Using a nonlinear filtering function on the contents of a single LFSR.
- Using the output of one (or more) LFSRs to control the clock of one (or more) other LFSRs.

4.2 Properties of LFSR-Based Keystream Generators

For essentially all possible secret keys, the output sequence of an LFSR-based keystream generator should have the following properties [8]:

- Large period.
- Large linear complexity.
- Good statistical properties.

It is emphasized that these properties are only necessary conditions for a keystream generator to be considered cryptographically secure. Since mathematical proofs of security of such generators are not known, such generators can only be deemed computationally secure after having withstood sufficient public security.

4.3 Nonlinear Combination Generators

One general technique for destroy the linearity inherent in LFSRs is to use several LFSRs in parallel. The keystream is generated as a nonlinear function F of the outputs of the component LFSRs, this construction is shown in Figure 3. Such keystream generators are called nonlinear combination generators, and F is called the combining function, it is non linear function, [1,6].

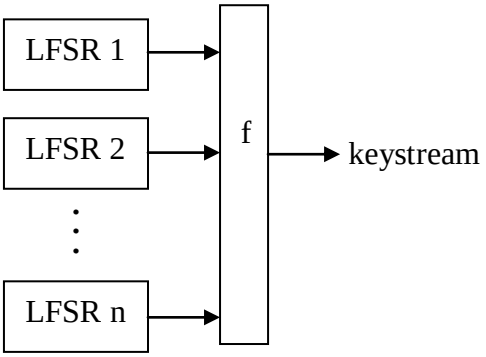


Figure 3: A nonlinear combination generator.

4.3.1 Properties of Combining Function

The function F must satisfy several criteria in order to withstood certain particular cryptographic attacks. A product of m distinct variables is called an m^{th} order product of the variables. Every Boolean function $F(X_1, X_2, \dots, X_n)$ can be written as a modulo 2 sum of distinct m^{th} order products of its variables, $0 \leq m \leq n$; this expression is called the algebraic normal form of F. The nonlinear order of F is the maximum of the order of the terms appearing in its algebraic normal form[8]. For example, the Boolean function:

$F(X_1, X_2, X_3, X_4, X_5) = 1 + X_2 + X_3 + X_4X_5 + X_1X_3X_4X_5$

has nonlinear order 4. Note that the maximum possible nonlinear order of a Boolean function in n variables is n. Suppose that n maximum length LFSRs, whose lengths $L_1, L_2, \dots, L_n$ are pairwise distinct and greater than 2, are combined by a nonlinear function $F(X_1, X_2, \dots, X_n)$, as in Figure 3, which is expressed in algebraic normal form. Then the linear complexity of the keystream is $F(L_1, L_2, \dots L_n)$. For example, the Geffe generator is defined by three maximum length LFSRs whose length L_1, L_2, L_3 are pairwise relatively prime with nonlinear combining function:

$F(X_1, X_2, X_3) = X_1X_2 + (1+X_2)X_3 = X_1X_2 + X_2X_3 + X_3$.

The keystream generated has period $(2^{L1} - 1) . (2^{L2} - 1).(2^{L3} - 1)$ and linear complexity equal to $L = L_1L_2 + L_2L_3 + L_3$ [8].

5. Cryptanalytic Attack of Stream Ciphers With LFSRs

5.1 Description of the Cryptanalytic Attack

Suppose that n maximum-length LFSRs $R_1, R_2, \dots, R_n$ of lengths $L_1, L_2, \dots, L_n$ are employed in a nonlinear combination generator.If the connection polynomials of the LFSRs and the combining function F are public knowledge, then the number of difference keys of the generator is $\prod_{i=1}^{n} (2^{Li} - 1)$. A key consists of the initial states of the LFSRs. Suppose that there is a correlation between the keystream and the output sequence of R_1, with correlation probability $p > \frac{1}{2}$ [5] . If a sufficiently long segment of the keystream is known the initial state of R_1 can be deduced by counting the number of coincidences between the keystream and all possible shifts of the output sequence of R_1, until this number agrees with the correlation probability p. Under these conditions, finding the initial state of R_1 will take at most $(2^{L1} - 1)$ trials. In the case where there is a correlation between the keystream and the output sequences of each of $R_1, R_2, \dots, R_n$, the secret initial state of each LFSR can be determined independently in a total of about $\sum_{i=1}^{n} (2^{Li} - 1)$ trials; this number is far smaller than the total number of different keys [3].

5.2 Steps of the Cryptanalytic Attack

The major steps of the cryptanalytic attack include the following:

1. Create simulation program for the cryptographic system under attack.
2. Compute the correlation probability between the keystream and each of the LFSRs, and in some cases we compute the correlation between the cipher stream and each of the LFSRs. In the Tow cases we fix the threshold values for the correlation.
3. The statistical properties of plaintext, ciphertext, keystream, LFSRs, and combining function must be computed and saved in separated files.
4. Initialize LFSRi with initial values and run the simulation program, count the number of coincidences between the keystream and all possible shifts of the sequence generated by LFSRi. Also in some cases we count the coincidences between the cipher stream and the sequence generated by LFSRi.
5. If the number of coincidence agrees with the threshold values then the secret initial state of LFSRi is determined.
6. In the same manner the secret initial state of each LFSR can be deduced.

6. Conclusions

This paper introduced a studied and analysis of stream ciphers systems based on LFSRs, since LFSRs are the basic building block in most stream ciphers. The general technique for destroying the linear properties of LFSRs is to use a non linear combining function on the outputs of several LFSRs. The combining function must include several conditions in order to withstand against cryptanalytic attacks. Thus the combining function should be carefully selected to ensure that there is no statistical dependence between any small subset of the n LFSR sequences and the keystream.

References:

[1] F. Armknecht and M. Krause: Algebraic Attacks on Combiners With Memory, proc. crypto 2003, vol. 2729 of LNCS, pages 162-175, Springer 2003.

[2] N. Courtois: Fast Algebraic Attacks on Stream Ciphers With Linear Feedback, In D. Boneh, editor, proc. crypto 2003, vol. 2729 of LNCS, pages 176-194, Springer 2003.

[3] Matthias Krause: BDD-Based Cryptanalysis of Keystream Generators, http://eprint.

[4] Wenbo Mao: Modern Cryptography: Theory and practice, Prentice Hall PTR, 2003.

[5] Rolf Oppliger, "Contemporary cryptography", Artech house inc. 2005.

[6] Menezes, P. Van Oorschot and S. Vanstone: Handbook of Applied Cryptography, CRC Press, 1996.

[7] Alan. Konheim, Coputer Security And Cryptography ", published by John Wiley & Sons, Inc., Hoboken, New Jersey, 2007

[8] Schneier: Applied Cryptography, second edition, John Wiley & Sons, 1996.

تحليل نوع من التشفير الانسيابي

خالد فاضل جاسم ، قاسم محمد حسين

[1]*قسم علوم الحاسوب ، جامعة جيهان ، اربيل ، العراق*

[2]*قسم علوم الحاسوب ، كلية علوم الحاسوب والرياضيات ، جامعة تكريت ، تكريت ، العراق*

(تاريخ الاستلام: 27 / 10 / 2010 ---- **تاريخ القبول:** 12 / 6 / 2012)

الملخص

تستخدم انظمة التشفير الانسيابي بشكل واسع في تشفير المعلومات الحساسة (المهمة) وذلك لسرعتها وقلة نشرها للاخطاء. ان تصميم أنظمة التشفير الانسيابي يعتمد على استخدام مسجل تزحيف خطي باطوال محتلفة ، ودوال ربط تستخدم متعددات حدود للتغذية العكسية مختلفة. يقدم هذا البحث تحليل للمعايير التي تؤخذ بنظر الاعتبار عند تصميم نظام التشفير الأنسيابي فيما يتعلق باختيار مسجلات التزحيف والتعقيد الخطي. كما يقدم طريقة تستخدم في مهاجمة هذا النوع من انظمة التشفير.