Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(3): 24-29

Research Article ISSN: 2394 - 658X

Enhance LFSR Cipher

Louay A Hussein Al-Nuamy

Department of computer science, Oman college of Management and Technology, Barka, Sultanate of Oman
loay.alneimy@omancollege.edu.om

ABSTRACT

Encryption software executes an algorithm that is designed to encrypt computer data in such a way that it cannot be recovered without access to the key. Software encryption is a fundamental part of all aspects of modern computer communication and files protection. The purpose of encryption is to prevent third parties from recovering the original information. This is particularly important for sensitive data like credit card numbers. In this paper a new stream cipher, called LFSR key position, is proposed. The design of the proposed cipher is quite simple, composing two important cipher algorithms LFSR and Key position ciphers. The design goal of producing the LFSR key position stream cipher significantly depend on eliminating the weakness of both LFSR and key position ciphers through combine the two randomize factors characteristics of character position and the plain text context. The paper also explains the importance of the stream ciphers as modern class of encryption algorithms.

Key words: Cryptography, key position cipher, LFSR cipher, LFSR key position cipher

---

INTRODUCTION

Cryptography is where security engineering meets mathematics. It provides us with the tools that underlie most modern security protocols. It is probably the key enabling technology for protecting distributed systems [2]. The basic terminology is that cryptography refers to the science and art of designing cryptographic algorithms (ciphers); cryptanalysis to the science and art of breaking them; while cryptology, often shortened to just crypto, is the study of both. The input to an encryption process is commonly called the plaintext, and the output the cipher text [16]. Cryptographic algorithms play a crucial role in the information society. When we use our ATM or credit card, call someone on a mobile phone, get access to health care services, or buy something on the web, cryptographic algorithms are used to offer protection. These algorithms guarantee that nobody can steal money from our account, place a call at our expense, eavesdrop on our phone calls, or get unauthorized access to sensitive health data [4]. It is clear that information technology will become increasingly used and we expect to see more of e-government, e-voting, e-commerce, and m-commerce. These new environments and applications will present new security challenges, and there is no doubt that cryptographic algorithms and protocols will form part of the solution [4]. Long distance communication allows information being intercepted much easier than ever. To protect the confidentiality of information, encryption is widely used in military, intelligence and diplomatic services. Cryptography starts to play an important role in daily life. Modern cryptography is developed to protect information confidentiality, integrity and provide authentication [12].

Cryptography systems can be broadly classified into symmetric key systems such as DES, AES and RC4 that use a single key that both the sender and recipient have to encrypt and decrypt respectively. Public key or asymmetric systems such as RSA, ElGamal and Elliptic curve cryptography that use two keys, a public key known to everyone and a private key that only the recipient of messages uses [10, 13]. Symmetric key cryptosystems are an important type of modern cryptosystem [11]. Symmetric key systems are cryptosystems where the same key is used for both encryption and decryption. This class of cryptosystems is important in modern cryptography because, in general, symmetric key cryptosystems are much faster than public key cryptosystems [11].

Symmetric cryptosystem itself also is usually subdivided into block ciphers and stream ciphers. Block ciphers tend to simultaneously encrypt groups of characters; whereas stream ciphers operate on individual characters of a plaintext message one at a time [10]. Stream ciphers are always faster than block ciphers but due to the nature of random number generators which have been used in well-known stream ciphers, there are confronting with many threatening problems that permits unauthorized persons to easily access on public privacy. On the other hand, it is

impossible to have infinite state random number generator to generate a truly random sequence, since the finiteness forces the random sequence to be periodic [3]. In this paper, two public stream cipher algorithms (LFSR and Key Position) are explained briefly, and then a new stream cipher algorithm has designed. The key position cipher dose not needs pseudorandom number generation and so the resulting algorithm (LFSR Key Position) so its pass one of the most stream cipher problems.

STREAM CIPHER

Stream ciphers are a very important class of encryption algorithms. Stream cipher algorithms are being used in a wide range of information processing applications. This kind of cryptography is symmetric encryption primitives which are widely applied for providing the confidentiality of different networks [3]. Practically all studies in the cryptosystem of block ciphers is focused on DES, and nearly all the proposed block ciphers are based in some way on the perceived design goals of DES. There is no algorithm occupying an equivalent position in the field of stream ciphers. There are a huge variety of alternative stream cipher designs [14]. The area of stream cipher encryption has been very active recently due to growing interest from academic and industry research, standardization efforts like AES, NESSIE, CRYPTREC, and ESTREAM, as well as due to ease of some governments control over export of cryptography. A stream cipher is a type of cryptographic system that usually strengthen internet network and wireless networks more security. It is one of most active directions in published field of cipher algorithm and symmetric key encryption [8].

Stream ciphers differ from block ciphers in several aspects: they always contain a secret “state” (i.e. memory) which evolves with time during the encryption; they usually produce streams of keys rather than blocks. Thus the two main parts of a stream cipher are: state transition function (which given an old state computes a new state), and a filter (which given the state produces the output of the stream cipher). The output of a stream cipher (i.e. a random looking) stream of digits is typically merged to the plaintexts resulting in cipher texts. Thus stream ciphers can be viewed as computational analogy of a one-time pad (OTP) cipher, replacing a long secret key by a short secret seed and pseudo-randomly generated stream of digits, computationally indistinguishable from a stream of random digits [5]. In addition to the plaintext (P), cipher text and key spaces (K), stream cipher are endowed with a key stream alphabet L and a key stream generator F = {f1, f2 …}, where fi : K × Pi-1 → L. A key stream l1, 12 … is generated and used to encrypt a plaintext X = x1x2… xm. According to the rule E (l1, x1) E (l2, x2)…, Where li = fi (K, x1… xi-1) For each l ∈ L the encryption and decryption function El, Dl satisfy El(Dl (p)) = p for all p ∈ P [16].

A stream cipher is synchronous if the key stream is independent of the plain text, i.e. key stream is generated only as a function of the key (K is called the seed) [16]. The key component of the stream cipher is the key stream generator. Figure one shows the process of stream cipher key generation. Various types of stream cipher can be generated based on the choice of the combining function and the linear feedback shift registers (LFSR) [15].

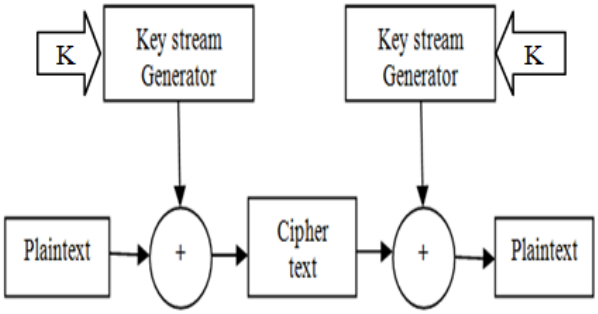

Fig. 1 Stream cipher key generation

LFSR CIPHER

Linear Feedback Shift Register (LFSR) is used to generate pseudo random numbers. LFSR has two main parts. These are shift register and feedback function. A shift register’s identifying function is shifting its contents into adjacent positions within the register or, in the case of the position on the end, out of the register. The position on the other end is left empty unless some new content is shifted into the register. The contents of a shift register are usually thought of as being binary, that is, ones and zeros. In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit. By definition, the selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift, filling the position that is emptied as a result of the shift [17].

LFSRs (linear-feedback shift registers) find extensive use in cryptography. For example, the cryptographic algorithms in the GSM (Global System for Mobile communications) mobile-phone system rely on the use of LFSRs.

An LFSR comprises a register containing a sequence of bits and a feedback function. In general, this function is an XOR (exclusive-OR) operation on certain bits in the register [9]. LFSR is suitable for speech because speech is continuous streaming data. They encrypt individual character (usually binary digits) of a plaintext message one at a time, using an encryption transformation which varies with time. Stream cipher which used LFSR is algorithm that encrypts plaintext one bit at a time. Key stream generator generates outputs stream of bits k_1, k_2, ..., k_n. Cipher text is obtained by XORing this key stream bits with plain text bits p_1, p_2, ..., p_n (i.e. $c_i = p_i \oplus k_i$) [17].

One advantage of LFSR synchronous stream ciphers is that they do not propagate transmission errors. If a bit is garbled during transmission only, that bit is decrypted incorrectly. If any active attacker inserts a bit into the cipher text stream and then it can be detected, the cipher text cannot be decrypted correctly after that inserted bit [17]. In this paper we will use Auto Key LFSR cipher which encrypt one character at a time instead of one bit, and also it does not need random number generator because it use the auto key generator from the plaintext itself. In this method the key stream starts with secret key K, and regenerated with the auto key function which shifts the plaintext characters by one position and cipher text is obtained by adding current character modulo to M, where M represent the plaintext representation code size (i.e. 256 for ASCII code and 65536 for Unicode), and the encryption formals as follows in equation number one [16]:

$$k_1 = K,\ k_i = xi_{-1}$$
$$E_k(x) = x + k \bmod M$$
$$D_k(y) = y - k \bmod M \quad (1)$$

Where K: Encryption Decryption secret key, k_i: Encryption Decryption key for character xi, x_i: plain text character with sequence i, and M: represent the plaintext representation code size.

To good understanding for how auto key LFSR cipher works we will apply it for the plaintext "WELCOME", with M = 26, and K = 20; as follows:

i	1	2	3	4	5	6	7
Plain text (x_i)	W (22)	E (4)	L (11)	C (2)	O (14)	M (12)	E (4)
Key (k_i)	20	22	4	11	2	14	12
Cipher text (y_i)	Q (16)	A (0)	P (15)	N (13)	Q (16)	A (0)	Q (16)

So E(WELCOME) = "QAPNQAQ".

KEY POSITION CIPHER

To generate a unique encryption key for each character of the plaintext, the character position within the plaintext can be used for that purpose [7]. It is clear that characters positions will be changed in an incremental sequence in the plaintext. Normally the first character will be at the position one of the plaintext, the second character will be at the position two of the plaintext, …, and the last character will be at the position N (N represents the number of character in the plaintext); so on with the rest of the characters in the plaintext [1]. The character position within the plaintext is not a mystery; any intruder can use it in the process of decrypt the cipher text. To use the character position as a secret encryption key which cannot be accessed by the hackers; the character position number can be entered into a mathematical function to get a new number that differs from the character position number and used as the encryption key. The mathematical function that used in the process the conversion can be any simple or complex equation which is proposed by the person doing the encryption process [1]. The encryption decryption formals with an example of quadratic conversion equation showed as follows in equation number two [7]:

$$k_i = (a * i^2 + b * i + c) \bmod M$$
$$E_k(x) = x + k \bmod M$$
$$D_k(y) = y - k \bmod M \quad (2)$$

Where i: The character position in the text, k_i: Encryption Decryption key for character xi, a, b, and c: three secret keys(seed key), x_i: plain text character with sequence i, and M: represent the plaintext representation code size.
To good understanding for how key position cipher works we will apply it for the plaintext "WELCOME", with M = 26, a = 4, b = 5, and c = 20; as follows:

i	1	2	3	4	5	6	7
Plain text (x_i)	W (22)	E (4)	L (11)	C (2)	O (14)	M (12)	E (4)
Key (k_i)	3	20	19	0	15	12	17
Cipher text (y_i)	Z (25)	Y (24)	E (4)	C (2)	D (3)	Y (24)	V (21)

So E(WELCOME) = "ZYECDYV".

LFSR KEY POSITION CIPHER

The Linear Feedback Shift Register (LFSR) has been one of the most popular encryption techniques widely used in communication [17]. But the main disadvantage of LFSR based structure is its vulnerability to attack due to inherent linearity in the structure [6]. LFSR based stream ciphers mainly employ two different methods to spoil this linearity.

In the first method, nonlinearity is introduced by using a suitable cryptographic Boolean function. Combination generators and filter generators are the structures built using Boolean function. In the second method, the LFSR is irregularly clocked to effect non-linearity. The fundamental structures based on this method are step1-step2 generators, alternating step generators, shrinking and self-shrinking generators [6]. In this work we suggest a third method to resolve the LFSR disadvantage through combining the LFSR with another stream cipher which bridge the problem of the linearity. The key position stream cipher can use quadratic conversion function (to generate the keys stream) which solve the problem of linearity [7]. The combining of key stream generation formals for both LFSR and key position can satisfy our goal, one suggestion for this sort of combination (equation one and equation two) can be given by:

From:

$$k_1 = K,\ k_i = x_{i-1} \quad (3)$$

$$k_i = (a * i^2 + b * i + c) \text{ Mod } M \quad (4)$$

We can get:

$$k_1 = K,$$

$$k_i = (x_{i-1} * i^2 + x_{i-1} * i + x_{i-1}) \text{ Mod } M \quad (5)$$

The final encryption decryption formals for the suggested cipher (LFSR Key Position) can be given by equation number three:

$$k_1 = K,$$

$$k_i = (x_{i-1} * i^2 + x_{i-1} * i + x_{i-1}) \text{ Mod } M$$

$$E_k(x) = x + k \text{ mod } M$$

$$D_k(y) = y - k \text{ mod } M \quad (6)$$

Where

Where i: The character position in the text, k_i: Encryption Decryption key for character xi, x_i: plain text character with sequence i, and M: represent the plaintext representation code size.

To good understanding for how auto key LFSR cipher works we will apply it for the plaintext "WELCOME", with M = 26, and K = 20; as follows:

$k_1 = K = 20,$

$k_2 = (22 * 2^2 + 22 * 2 + 22) \text{ Mod } 26 = 24,$

$k_3 = (4 * 3^2 + 4 * 3 + 4) \text{ Mod } 26 = 0,$

$k_4 = (11 * 4^2 + 11 * 4 + 11) \text{ Mod } 26 = 23,$

$k_5 = (2 * 5^2 + 2 * 5 + 2) \text{ Mod } 26 = 10,$

$k_6 = (14 * 6^2 + 14 * 6 + 14) \text{ Mod } 26 = 4,$ and

$k_7 = (12 * 7^2 + 12 * 7 + 12) \text{ Mod } 26 = 8.$

i	1	2	3	4	5	6	7
Plain text (x_i)	W (22)	E (4)	L (11)	C (2)	O (14)	M (12)	E (4)
Key (k_i)	20	24	0	23	10	4	8
Cipher text (y_i)	Q (16)	C (2)	L (11)	Z (25)	Y (24)	Q (16)	M (12)

So E(WELCOME) = "QCLZYQM", and the decryption process will be as shown below:

i	1	2	3	4	5	6	7
Cipher text (y_i)	Q (16)	C (2)	L (11)	Z (25)	Y (24)	Q (16)	M (12)
Key (k_i)	20	24	0	23	10	4	8
Plain text (x_i)	W (22)	E (4)	L (11)	C (2)	O (14)	M (12)	E (4)

So E(QCLZYQM) = "WELCOME".

TESTS FOR LFSR KEY POSITION CIPHER

There are several tests that can be used to quantify the strength of stream ciphers. Standard tests that are independent of the algorithm are statistical tests, correlation attack and linear complexity profile [15]. In statistical tests a binary sequence is said to be random if there is no obvious relationship between the individual bits of the sequence. Since the sequence generated by the LFSR is periodic with a period p, then it is not considered a true random sequence but is referred as a pseudorandom sequence or a p_n (pseudonoise) sequence [15]. For the LFSR key position cipher a set of statistical tests are applicable frequency test, serial test, poker test, autocorrelation test and runs test.

Frequency test calculates the number of ones and zeroes of the binary sequence and checks if there are no large differences [15]. For LFSR key position cipher the key stream depends mainly on the character position which changes one bit at a time starting from 1 to the plain text size. For example by using 32 bits position sequence variable; the values will be $(00000001)_{16}$, $(00000002)_{16}$ … $(FFFFFFFF)_{16}$; and it is obvious that this sequence has equal number of ones and zeros.

Serial test represents the transition characteristics of a sequence such as the number 00, 01, 10 and 11 are evaluated. Ideally, it should be uniformly distributed within the sequence [15]. For LFSR key position cipher the random numbers are only used to generate the initial key which is used only one time to encrypt and decrypt the first character of the text, and then the remainder key stream will depend on the internal structure of the plaintext which is random by its nature.

In poker test an N length sequence is segmented into blocks of M bits and the total number of segments is N/M. Within each segment, the integer value can vary from 0 to m = 2M-1. The objective of this test is to count the frequency of occurrence of each M length segment. Ideally, all the frequency of occurrences should be equal [15]. In LFSR key position cipher the key stream generation process depends on the context of the plain text which is effect the cipher text generation; which produce different cipher text segments.

The autocorrelation test performs the autocorrelation of the sequence and compares the value of the maximum peak with the value in the origin, the principal-secondary lobe. The worst result of this test is when there is a large peak because many of bits shifted will reflect the same behaviours as the originals. It is preferably having many reasonable middle peaks than the few high peaks, also due to the correlation immune attack [15]. As the LFSR key position cipher encryption key stream generation depends on two factors (the character position and the plain text context); so the correlation between the cipher text characters will be more varied than using one random number generator source.

In runs test a sequence is divided into contiguous stream of 1's that is referred as blocks and contiguous stream of 0's that is referred as gaps. If r0i is the number of gaps of length I, then half of the gaps will have length 1 bit, a quarter with length 2 bits, and an eighth with length 3 bits. If r1i is the number of blocks of length I, then the distribution of blocks is similar to the number of gaps [15]. The distribution of blocks and gaps in the LFSR key position cipher text will be randomly depend on three factors which are the secret initial key, character positions, and the plain text context.

A correlation attack is a widely applicable type of attack which might be used with success on generators which attempt to combine the output from several cryptographically weak key stream generators [14]. A correlation attack exploits the weakness in some combining function which allows information about individual input sequences to be observed in the output sequence. In such a case, there is a correlation between the output sequence and one of the internal sequences. This particular internal sequence can then be analyzed individually before attention is turned to one of the other internal sequences. In this way the whole generator can be deduced - this is often called a divide-and-conquer attack [14]. With the LFSR key position cipher the code size play a big role in the key stream generation process because the number of bits represent each character in the plain text will be enter as a parameter in the generation formula of the key stream, and this will affect the correlation between the plain text as input to the cipher and cipher text as its output.

Every sequence s_0s_1 … of period p satisfies a linear recurrence of length p, namely $s_i + p = s_i$ for all $i \geq 0$. A sequence may additionally satisfy a shorter recurrence that is each bit of the sequence can be defined using some linear expression which involves bits that are less than p bits away. The length of the shortest recurrence is defined to be the linear complexity (or linear span) of the sequence [14]. The linear recurrence in LFSR key position cipher is limited according to the fact that the character position (which is does not recurrence) enter as one of the important factor in key stream generation.

Table -1 LFSR Key Position Cipher Tests Result

Test	Result
Frequency test	98/100
Serial test	96/100
Poker test	97/100
Autocorrelation test	99/100
Runs test	96/100
Correlation attack test	18/20 Immune
Linear complexity profile test	4875

Basically, the strength of stream cipher correlates with the size of key used. Although the algorithms are not good but with the large key used and excellent key management scheme, it could increase their performance. In practice, long messages are not transmitted to avoid Possibility of Intercept (POI) and on average, data format for military standard is around 4000 bits [3]. According to National Institute of Standards and Technology (NIST) [3], all important cryptography tests (Frequency test, Serial test, Poker test, Autocorrelation test, Run test, a Correlation attack test, and Linear complexity profile test) have applied on LFSR key position stream cipher algorithm. All of tests passed successfully as shown in the following table (Table no. 1) which presents the number of keys that passed for each statistical test. Based on 95% significant level, a good generator should passes at least 95 over 100 initial conditions or keys.

CONCLUSION

While there is no single algorithm which acts as a focus for cryptanalysis in the field of stream ciphers, the impression left by many reviews of stream cipher techniques is that an overwhelming interest has been paid to shift register based schemes [14]. In this paper we introduce two important stream ciphers LFSR and Key position, and then the paper has presented the new stream cipher LFSR key position. A complete description of LFSR key position and an overview of possible attacks have been given. The designed algorithm has passed all of cryptographic tests in NIST standard successfully. Several properties of LFSR key position have not yet been considered, for example hardware implementations, hand optimized code in software, and a more exhaustive treatment of the security. We hope to fill some of these gaps in the near future.

REFERENCES

[1] Al-Nuamy, Data Security Using the Key Position Method, *Girin University Scientific Journal*, **2001**, 5(2), November, Lybia.
[2] Anderson, Security Engineering: *A Guide to Building Dependable Distributed System*, John Wiley Inc., 2nd edition, USA, **2001**.
[3] Bakhtiari Maarof, An Efficient Stream Cipher Algorithm for Data Encryption, *IJCSI Issues*, **2011**, 8(3).
[4] Bart, The NESSIE Project: Towards New Cryptographic Algorithms, *Katholieke University Press*, Kasteelpark Arenberg 10, B-3001, Belgium, **2003**.
[5] Biryukov, Block Ciphers and Stream Ciphers: The State of the Art, *Katholieke University Press*, Kasteelpark Arenberg 10, B-3001, Belgium, **2003**.
[6] Deepthi Sathidevi, Hardware Stream Cipher Based on LFSR and Modular Division Circuit, *World Academy of Science Press*, **2008**, 46.
[7] Ghwanmeh and Abhath Al-Yarmouk, Enhanced Data Security Scheme using Key Position Algorithm Applied on Arabic Language, *J. Basis Sci. Eng*., **2008**, 17,165-176.
[8] Lan, Q and Rong, A Study to the Symmetric Ciphers and Protocols Weigh in Bayesian Mode, *International Journal of Computational Cognition*, **2010**, 8(3).
[9] Lillo Motta, LFSR Provides Encryption, Brandeis University, *Design Ideas Journal*, www.ednmag.com, **2001.**
[10] Patrik Thomas, *SNOW: A New Stream Cipher*, Lund University Press (www.it.lth.se), Sweden, **2001**.
[11] Pfleeger C L, *Security in Computing*, Prentice Hall., 2nd edition, USA, **2003**.
[12] Preneel, *Crypt Analysis and Design of Stream Ciphers*, Katholieke University Press. 1st edition, Belgium, **2008**.
[13] Ramesh K Murali, On Linear Complexity of Binary Sequences, *IJDPS*, **2010,** 1 (2).
[14] Robshaw, Stream Ciphers: Technical Report, *RSA Laboratories a Division of RSA Data Security Inc*., 2nd edition, USA, **1995**.
[15] Sidek Sha'ameri, Comparison Analysis of Stream Cipher Algorithms for Digital Communication, *Universiti Teknologi , Jurnal Teknologi*, 46(D), Malaysia, **2007**.
[16] Stallings, *Cryptography and Network Security: Principles and Practice*, Prentice Hall. USA, 2nd edition, USA, **2006**.
[17] Win Kyaw, Speech Encryption and Decryption Using LFSR, *World Academy of Science, Engineering and Technology Journal*, **2008**, 48.