NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 2, ISSUE 8, AUG-2015

CRYPTOSYSTEM KEY GENERATOR WITH MIXED OPERATIONS AND DYNAMIC SHIFT REGISTERS

Dr. Qasim Mohammed Hussein
Assistant Prof.
Computer sciences department, computer sciences and mathematic college -Tikrit university, Tikrit – Iraq.

ABSTRACT

The aim of this paper design and implement a key generator of cryptosystem that produces high security and random keys. The proposed generator includes many linear feedback shift registers. Each shift register has a shifting number in each encryption process that varies from other registers. A specific number of them are used in each encryption/decryption process. The selected registers are choosing depends on using tables that construction in the 2^3 field. To reduce the initial key and get random initial, the convolution multiplication is used to fill the initial values of registers. The generator uses in combining functions two logical operations, XOR and XNOR. From implementation of key generator and show the results of statistical test of randomness of key sequences samples, we show that the generator has good security.

KEYWORDS: Stream cipher, GF(2), Convolution multiplication, key generator, shift register, encryption.

INTRODUCTION

Cryptology has two main branches: cryptography and cryptanalysis. The cryptographic is the mathematical functions uses for encryption and decryption. The aims of cryptography are to secure the sensitive information communications that transmit from adversaries. The cryptanalysis is used mathematical techniques for attempting to defeat cryptographic techniques. In good cryptosystem, the retrieve information about the plaintext from the cipher text without knowledge its key is very hard [1].
Cryptographic algorithms are either symmetric (secret key) or Asymmetric (public key) encryption. Symmetric encryptions use one key to encrypt and decrypt data. Secret key algorithms can categories into either Stream or Block ciphers [2] .
Various types of stream cipher are implemented based on the choice of the combining function and the linear feedback shift registers (LFSR). [3][4][5][6][7]
The aim of this paper introduces key generator cryptosystem that contains different LFRSs for each encryption process, and using dual operations in the combining functions, XOR and XNOR. And every shift register has different from other registers in the generator and vary in each encryption process .Also to reduce the amount of key exchange, the convolution multiplication is used to fill the initial values of LFSRs.

1. Stream Cipher

Stream cipher is a symmetric key cryptosystems; the same key is used for both encryption and decryption. Stream cipher contains two components: a key stream and a mixing function [8]. In stream ciphers, many linear feedback shift registers (LFSRs) and combined function for the output are used. An LFSR of length *n* is an *n* stage register with a linear feedback function. To get a secure key stream with high randomness a non-linear Boolean function are used to destroy the linearity of the LFSR(s) output. Encryption is performed by doing a simple operation between the key stream with the plaintext, usually with the bitwise XOR operations.

There are two types of stream ciphers synchronous stream ciphers and self synchronizing stream ciphers. In synchronous stream ciphers the key stream is generated from the key independently of the plaintext and the cipher text. While self-synchronizing stream ciphers the key stream is generated from the key as well as a fixed number of previous cipher text digits [9].

The good key generator must have the following properties [10][11]:

1) Has maximum period of key sequence (e.g. 2^n-1).
2) High linear complexity
3) Produce sequences with good statistical properties.
4) Can be readily analyzed using algebraic techniques.
5) Easy to implement in hardware.

There are a number of statistical tests that used to determine whether the key sequences have specific characteristics of sequence randomness [12]

2. Finite Field

The finite field, call Galois field (GF), is a field containing a set of elements with two arithmetic operations, addition and multiplication, have been defined and it has the properties: closure, associativity, commutativity, distributivity. Also it has both additive and multiplicative inverses. The number of elements in the field is called order of a finite field. So the order of GF(2^n) is 2^n.

In cryptographic algorithms, GF(2^n) are widely used. For example, AES and elliptic curve are based on arithmetic operations in finite fields [1].

The modular arithmetic is an integer arithmetic which reduce all numbers to the set [0,1, …, n-1] for a number n. Any integer outside of this range is reduced to number in this set by taking the remainder after division by n. A polynomial can be used to represent the LFSR function by using the form:

$$L(x) = a_n x^n + a_{n-1} x^{n-1} \dots a_1 x^1 + a_0 \quad (1)$$

Where a_n is either 0 or 1, where n is the length of register length. a_0 must be 1 in all LFSRs.

3. Convolution multiplication

It is a cyclic convolution of two polynomials. It is calculated using equation (2)[13] [14]:

$$c_k = \sum_{i=0}^{k}{}' a_i.b_{k-1} + \sum_{i=k+1}^{N-1}{}' a_i \,.b_{N+k-i} = \sum_{i+j=k\ (mod N)}{}' a_i \,.b_j \qquad (2)$$

Where $a = a_0 + a_1x^1 + \ldots + a_nx^n$ and $b = b_0 + b_1x^1 + \ldots + b_nx^n$

For example, the convolution multiplication for the two polynomials of with degree 5, [1 0 1 0 3] and [0 2 0 1 0] is $(1 + x^2 + 3x^4) * (2x + x^3)$.

$$= 2x + x^3 + 2x^3 + x^5 + 6x^5 + 3x^7$$
$$= 2x + 3x^3 + 7x^5 + 3x^7$$
$$= 7 + 2x + 3x^2 + 3x^3$$

$= 1 + x^2 + x^3$ *by taking mod2* So, the resulting polynomial can be represented as SR with the values [1 0 1 1].

4. The proposed key generator Description

5.1 The contents of the generator

The proposed key generator consists of the following:

1) There is a set of maximal LFSRs including (79) SRs with different lengths each one has a combining function that give maximum period [15][16]. There are two operations, the first 40 registers use XOR operations in its combining function, while the rest (39 register) is used XNOR operation. For each encryption / decryption process, eight of them are selected to contribute in key generation. Four of them use XOR operation where as the other four use XNOR operation. The generator uses nonlinear combining functions on moving depending on chosen one from four nonlinear tables:3,4,5,6.

2) The generator uses 4 tables that are prepared with size 8x8 in the GF(2^3), as shown in tables 3, 4,5, and 6. The number of tables is: 00, 01, 10, and 11.The function of tables determines the number of shifting in each SR. In each encryption process, one of them is selected in each encryption process and one row of it determines the number of shift for LFSRs; the first number determine the shifting in LFSR1, the second number for LFSR2,…etc.

3) There are two tables; each one contains the 6 bits representation of each English language letters, as shown in table 1 and table 2. This Representation insures the balance between 0 and 1.

4) There is a control register (RC) with length 31that it work with maximum period. The function of it to determine the LFSRs that are used in encryption process, the table and a row in it to control the shifting for each LFSR.

5) Another initial register, IR, with length 31 is used to help us in filling LFSRs uses two linear function with the same tapping which give maximum period in each case: XOR and XNOR.

5.2 Key generation

To generate the key stream, the following steps are done:

1) RC fills with the code of 5 characters from the key, each character represented by 6 bits from table 1. The position 31 of it is fill with 1. After filling the control register (RC) , it move 31 times and then select content of specific positions in it as following:

a) 2 bits are used to determine one of the four tables. The 2 bits decide the number of the chosen table from {00, 01, 10, 11}.

b) 3 bits to determine the row number in the selecting table to specify the motion of each eight register, {0,1, .. . , 7}. The values in the selected row are taken in sequence for the selected registers.

c) 6 bits to select 8 SRs that are used in encryption/decryption process from the registers set. The decimal number of the 6 bits represents the number of the first register. If the number grater that 39 , then mod 39 is done. Taking in the account, no duplication in selecting the register in the same encryption. The rest 7 registers will be found by add 10 to the first register number respectively, and take mod 79 for numbers. For example, if the 1st register of XOR set is 36, the second register 36+10 =46. 3rd register is 46+10=56, 56+10= 66, 76+10= 86 mod 79= 7 ,. The others are 17, 27. The condition in choosing, no two registers with same length.

2) IR fills in the same manner of filling the RC, but it depends on table 2 . The content of stage 31 in this SR is 1.

3) To fill the first register (SR1), IR is shifted 31, it use to generate a sequence of 0 and 1, to fill SR1 starting from the stage n-1 toward stage 1. And then the convolution multiplication between RC and SR1, the result represent the initial values of SR1.

The same manner is used to fill the other 7 registers that were selected, excepting they depend on using different register by using convolution multiplication as shown bellow. The resulting polynomial is truncated with the length of the SR that will filling, and then making modulo of 2 to the coefficients that yield from the multiplication as Follow:

$$SR3 = SR1 * SR2; \qquad SR4 = SR2 * SR3$$

$$SR5 = SR3 * SR4; \qquad SR6 = (SR2 \gg 2) * (SR5 \ll 3)$$

$$SR7 = (SR5 * SR6); \qquad SR8 = (SR7 \ll 4) * SR5$$

Where * represents convolution multiplication, << and >> represent shift left and shift right respectively.

4) The registers outputs are combined with using XOR function to produce the key stream bits of the generator.

Table 1: 6-bit coding

Symbol	Binary code	symbol	Binary code	symbol	Binary code	symbol	Binary code
U	000000	T	010000	Q	100000	N	110000
I	000001	6	010001	B	100001	M	110001
C	000010	7	010010	0	100010	G	110010
I	000011	M	010011	L	100011	I	110011
K	0000100	9	010100	H	100100	U	110100
J	0000101	S	010101	G	100101	O	110101

	0000110	3	010110	J	100110	K	110110
E	0000111	@	010111	B	100111	Z	110111
R	0001000	5	011000	V	101000	T	111000
E	0011001	8	011001	D	101001	Y	111001
O	0011010	4	011010	F	101010	D	111010
Q	0011011	Y	011011	P	101011	S	111011
B	0011100	X	011100	W	101100	A	111100
C	0011101	R	011101	N	101101	Z	111101
W	0011110	2	011110	V	101110	X	111110
A	0011111	F	011111	E	101111	L	111111

Table 2: 6-bit coding

Symbol	Binary code	symbol	Binary code	symbol	Binary code	symbol	Binary code
B	000000	K	010000	A	100000	U	110000
D	000001	X	010001	0	100001	I	110001
S	000010	T	010010	W	100010	W	110010
Q	000011	F	010011	X	100011	8	110011
P	0000100	T	010100	M	100100	R	110100
U	0000101	4	010101	N	100101	7	110101
E	0000110	H	010110	6	100110	T	110110
I	0000111	E	010111	D	100111		110111
C	0001000	3	011000	V	101000	@	111000
9	0011001	K	011001	A	101001	N	111001
5	0011010	J	011010	R	101010	1	111010
J	0011011	Y	011011	G	101011	S	111011
B	0011100	L	011100	Z	101100	Z	111100
F	0011101	E	011101	O	101101	H	111101
V	0011110	O	011110	M	101110	Y	111110
A	0011111	C	011111	G	101111	2	111111

Table 3

3	1	4	6	7	5	0	2
4	7	5	0	3	2	1	6
0	2	3	5	1	7	6	4
6	4	1	2	0	3	7	5
5	6	7	1	4	0	2	3
2	3	0	7	6	4	5	1
7	5	6	3	2	1	4	0
1	0	2	4	5	6	3	7

Table 4

1	2	5	0	7	4	9	3
0	3	6	1	5	7	2	4
5	4	7	2	6	1	3	0
2	1	4	6	0	3	5	7
6	5	2	3	4	0	7	1
3	6	0	7	1	2	4	5
7	0	3	4	2	5	1	6
4	7	1	5	3	6	0	2

Table 5

7	4	0	3	1	5	2	6
5	7	2	4	6	0	3	1
1	5	3	2	7	4	6	0
6	1	3	5	0	2	4	7
0	2	5	6	3	1	7	4
4	6	7	1	2	3	0	5
3	0	4	7	5	6	1	2
2	1	6	0	4	7	5	3

Table 6

1	0	3	2	4	5	7	6
7	6	5	4	3	2	1	0
3	2	1	0	7	6	5	4
0	1	2	3	4	5	6	7
6	7	4	5	2	3	0	1
4	5	6	7	0	1	2	3
2	3	0	1	6	7	4	5
5	4	7	6	1	0	3	2

EXPERIMENTAL RESULTS

The simulation of the proposed generator , which implemented using visual studio 2010, is illustrated the key stream that generated in this generator have a good statistical properties , and it pass the statistical test as shown in figures 1,2,3,4,5 for sample contains 1000 bits. Also, a high linear complexity is found using Berlekamp – Massey algorithm. As well as this generator insure high immunity against the correlation attack through using suitable combining nonlinear function.

$$The\ complexity\ of\ the\ proposed\ system = \binom{79}{8} x2^{4}\ x2^{44}$$

Where L is the sum of selected shift register length.
Also, the generator generates different keys, even if the same initially keys are used .because the registers that use within the generator will vary each time

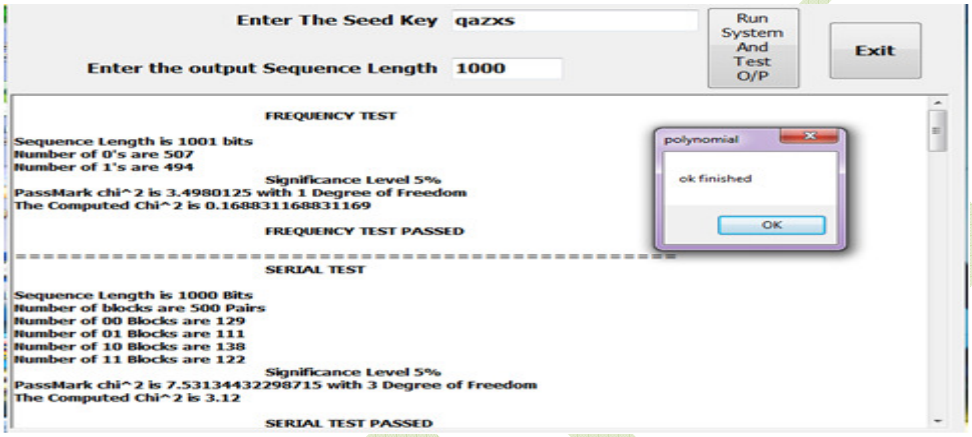

figure 1: Result of serail test

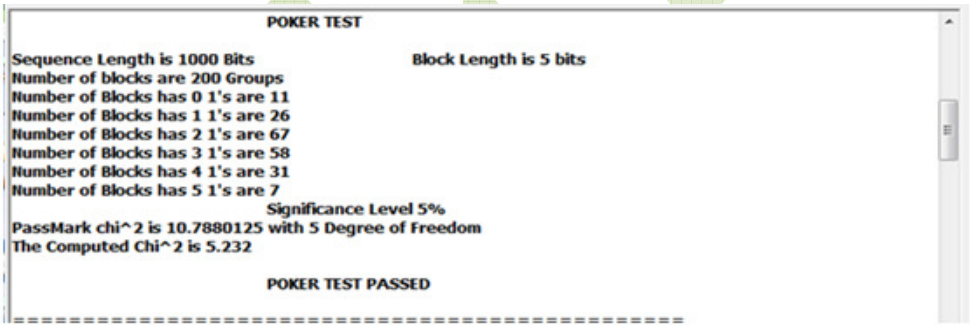

Figure 2 : Result of Poker test

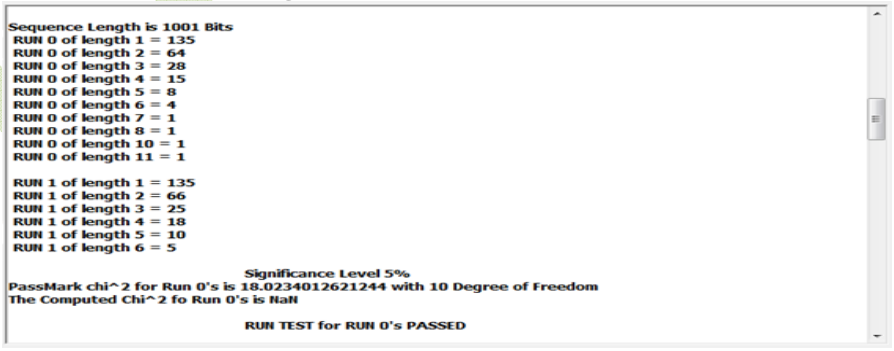

Figure 3: Result of run test

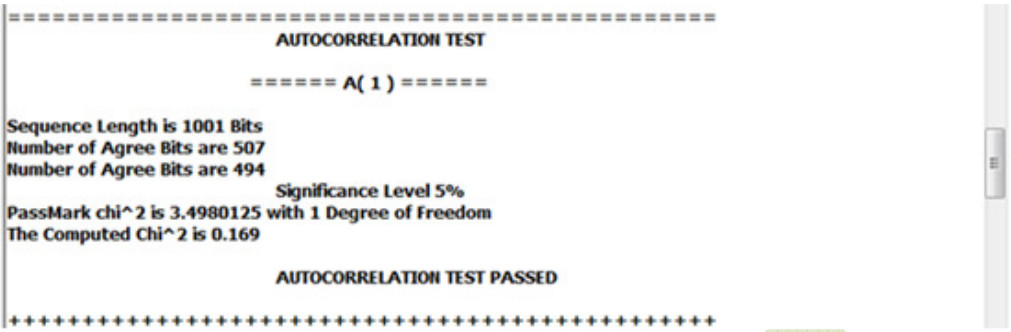

Figure 4: Result of autocorrelation test

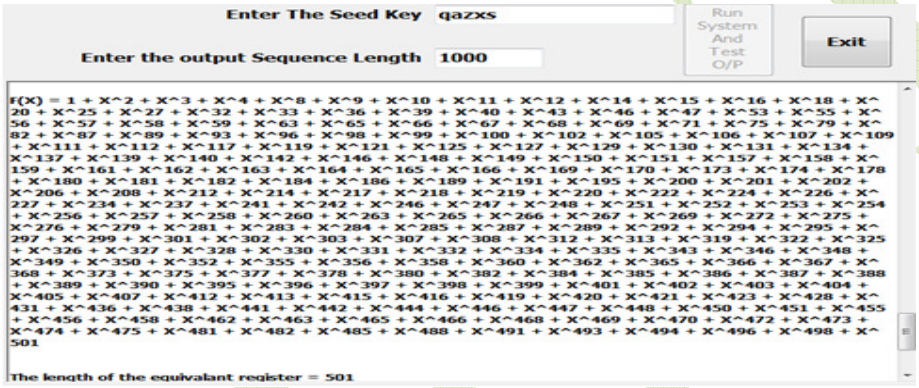

Figure (5) : The polynomial of a register tapping (local sequence)

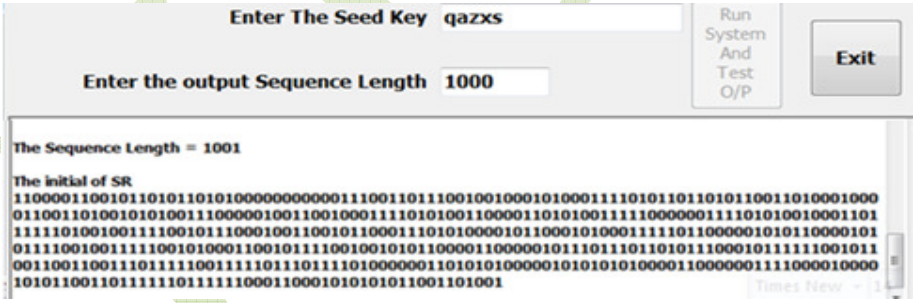

Figure (6): Initial of generated sequence

CONCLUSION

This paper presents an efficient key generator of a stream cipher system that implemented in manner that enable the register system to work in a non linear way as well as its function as a linear system, by using a set of non linear tables and dual operations and the convolution multiplication. The initial key that is needed to fill the register is limited in compared with the total initial key for the registers.

It uses 8 LFRSs from a set of LFSRs that contains 79 SRs in each encryption process. Each LFSR works with maximum length, and each has a private shifting vary in each

process depending on row of one table from 4 tables with size 8x8. The generator use dual Boolean operations in combining functions, XOR and XNOR. The experimental results illustrated that the key generator output pass the tests. So, experiments result proved that the key generator security and can be relied upon. All the encryption process is different from the other in terms of the length of the Registers, and amount shifting in each register, and the operation (XOR or XNOR) implemented. The generator generates different keys, even if the same initially keys are used because the registers that use within the generator will vary each time

REFERENCES

[1] Stallings, W.,"Cryptography and Network Security", *New Jersey: Prentice Hall, p95-129, 2011*

[2] Martin Hell, "*On the Design and Analysis of Stream Ciphers", PhD. Theses, Lund University, Sweden,ISBN: 91-7167-043-2,ISRN: LUTEDX/TEIT-07/1039-SE , 2007.*

[3] Abid Rahim Mat & Ahmad Zurisha, " *Comparision analysis of stream cipher algorithms for digital communication", Jurnal Teknologi,© Universiti Teknologi Malaysia, 2007.*

[4] C.S. Lamba, *"Design and Analysis of Stream Cipher for Network Security", Second International Conference on Communication Software and Networks, 2010.*

[5] Mohammed Abumuala, Othman Khalifa and Aisha-Hassan A. Hashim, *" A New Method for Generating Cryptographically Strong Sequences of Pseudo Random Bits for Stream Cipher " . International Conference on Computer and Communication Engineering (ICCCE 2010), Kuala Lumpur, Malaysia, 978-1-4244-6235-3/10/$26.00 ©2010 IEEE. 2010.*

[6] Mike Thomson, " *Linear Feedback Shift Registers, Galois Fields, and Stream Ciphers ", Cryptography II , 2012*.

[7] Kevin D. Bowers, and Alina Opreea, *Efficient Software Implementations of Large FiniteFields GF(2n) for Secure Storage Applications, JIANQIANG LUO,Wayne State University.RSA Laboratories,LIHAO XU,Wayne State University, 2010*

[8] Suhaila Orner Sharif, S.P. Mansoor, "*Performance analysis of Stream and Block cipher algorithms",3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 2010*

[9] D. Bucerzan, M. Cr˘aciun, V. Chi¸s, C. Ra¸tiu, "*Stream Ciphers Analysis Methods", Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol., No. 4, pp. 483-489. 2010.*

[10] willi.meier,2012,*Stream Ciphers, a Perspective AfricaCrypt 2012, Ifrane, willi.meier@fhnw.ch*

[11] Andreas Klein," *Stream Ciphers", Springer-Verlag, London, 2013*.

[12] A. Menezes, P. van Oorschot and S. Vanstone, "*Handbook of Applied Cryptography", CRC CRC Press, 1997.*

[13] Qasim Muhammad Hussein,, , *"Dynamic Stream Cipher Key Generation with Mixed Operations", the first internal scientific conference of cihan university, Iraq, 2014.*

[14] Qasim Muhammed Husein," *Recover the Private Keys of NTRU Cryptosystem from Known Public Information and Public Key", PhD. Theses, Iraq, University of Technology, 2009*

[15] Roy Ward, Tim Molteno, *Table of Linear Feedback Shift Registers, 2007.*

[16] Maria George and Peter Alfke ,2007*, Linear Feedback Shift Registers in Virtex Devices, XAPP210 (v1.3) , 2007.*

AUTHOR

Qasim Mohammed Hussein has PhD degree in computer sciences from Technology University, and currently Assist. Prof.at Tikrit University. He published 14 journal articles in the fields of cryptography and computer security, He participated in writing (4) books in computer science, registering in Iraqi national library – the house of books and documentations. He evaluated dozens of scientific research to scientific upgrade in Iraq universities or for publication in the scientific journals, and a member of the preparation and scientific committees for scientific conferences. He member of the editorial board of Journal of Pure Science at the University of Tikrit, and expert in many committees, and contributed to a number of scientific and preparation committees of scientific conferences in Iraq. He attended many conferences and scientific symposia, inside and outside of Iraq, and has a number of lectures and field studies in the field of computer.