S-RADG: A Stream Cipher RADG Cryptography

Salah A.K. Albermany[1], Duha Amer[2], and Sawsan Kamal[2]

Abstract This paper is attempt to develop keyless cryptography existing algorithm called RADG algorithm into symmetric key cryptography algorithm called S-RADG (Stream RADG {Reaction Automata Direct Graph}) algorithm . The result from S-RADG are different cipher texts with same plaintext. The key is generated randomly by using one of stream cipher algorithms , which is LFSR(Linear Feedback Shift Register) method. The new algorithm uses to encrypt data in many environments like a cloud computing environment..

Index Terms— Stream cipher, LFSR, RADG, Cryptography, Security.

1 INTRODUCTION

CRYPTOGRAPHY is an essential tool for security. Previously was the purpose of cryptography is to hide text messages during the war. But it has recently become necessary to secure the transfer of information between online networks with complete secrecy. Cryptography is the science that used for encryption and decryption, so that the information be secured and the phenomenon for the sender and the addressee only. Cryptography goals are to satisfy confidentiality, integrity , authentication. In modern cryptography, there are two types, symmetric key cryptography which use one key for encryption and decryption, and asymmetric key (public key) cryptography which use two different keys, one for encryption is called public key, another for decryption is called private key [5,6]. In symmetric key there are two types of ciphering, stream cipher and block cipher [5]. In this paper, will introduce stream cipher. Stream cipher encrypts one bit at time, see Figure 1. The new method is merged between stream cipher and existing method is called Reaction Automata Direct Graph (RADG) method [2] , by use LFSR method to generate key of new method, see Figure 2, converted keyless RADG method into symmetric key method with keeping RADG design properties.

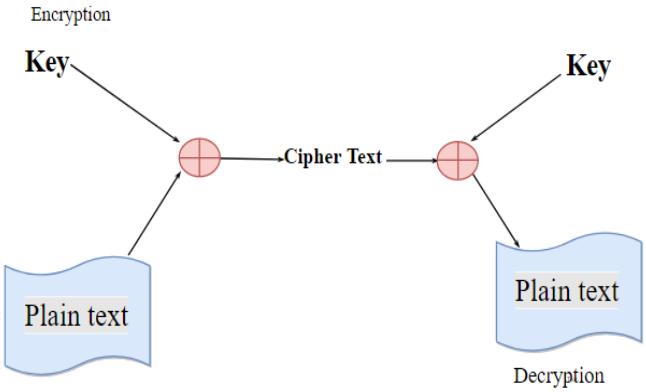

Figure 1: Encryption/Decryption in steam cipher

This attempt increase efficient of encryption , analysis of security for a new method show this by satisfy confidentiality and integrity.

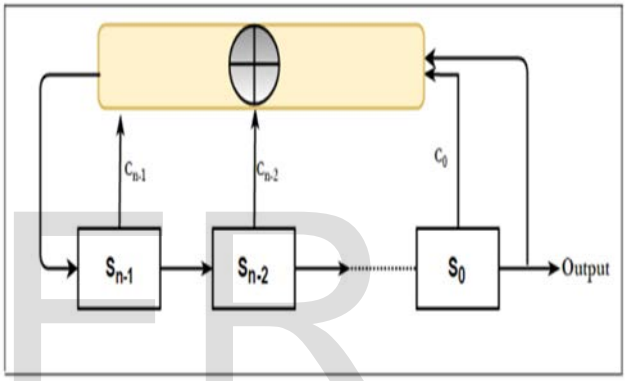

Figure 2 : Basic diagram of LFSR

2. STREAM CIPHER RADG(S-RADG)

RADG is algorithm that represent by set of tuples $\{R, Q, \Sigma, \Psi, J, T\}$, where R is reaction set that have n of states , which have λ of values, Q is standard design set that have m of states , also have λ of values , Σ represent input data , Ψ represent output transition, J is jump state which is subset of Q state that have K of states , which is haven't value , just transmit from one state to another in Q set, T represent transition function[2]. The following example in Figure 3 illustrated how RADG is work, where $n = 2, m = 3, k = 1$, suppose $\lambda = 2$. Let state number 4, 5 number are R states and state number 0, 1, 2 and 3 are Q states.

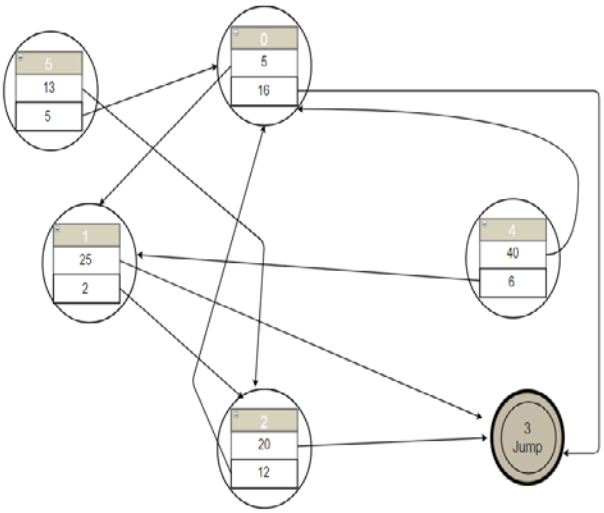

Figure 3 : Implementation RADG algorithm

Encryption in RADG algorithm which is :
Suppose the message to be encrypted or plain text is 0111

T(0,0)= (1,5)
T(1,1)= (2,2)
T(2,1)= (0,12)
T(0,1)= (4,16)
The cipher text is 5,2,12 and 16 .

The paper tried to design a new algorithm based on original RADG. The new algorithm is called S-RADG (stream RADG) , that used linear feedback shift register to generate key, see Figure 4 . Therefore the S-RADG algorithm is key scheme on the contrary of RADG algorithm which is keyless scheme. S-RADG contain 6 tuples as original RADG which are $\{R, Q, \Sigma, \Psi, J, T\}$. Each single state in R or Q states have λ of values , the value either 0 or 1 in each state. The proposed algorithm used message text in stream cipher and also use key generated by linear feedback shift register (LFSR) as a S-RADG key. The current state in R and Q is transmitted by certain function is called transition function. In S-RADG design the transitions of states based on the transitions of designed examples.

Key Generation

S-RADG key is sequence that generated by LFSR algorithm .LFSR is linear function , in equation 1[1,5].

$$f(\vec{S}) = \sum_{i=0}^{n-1} C_i S_i \qquad (1)$$

.

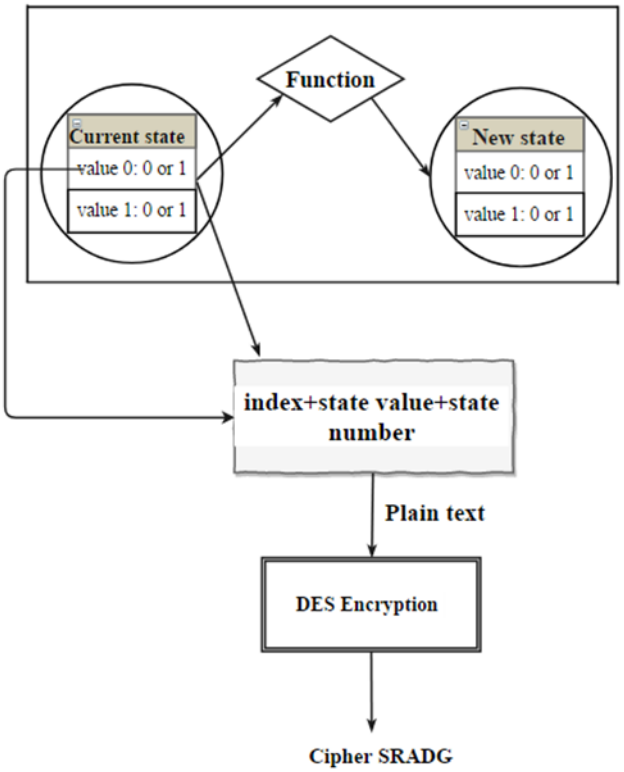

Figure 4: S-RADG algorithm

When the initial value generated, is called seed, then used to determine the output. The input sequence with length n is $S_{n-1}, S_{n-2}, \ldots S_0$. The equation 2 that used for linear function to determine output is[1,5,3]:

$$S_n + L = \sum_{i=0}^{L-1} C_i S_n , \quad \forall n \geq 0 \quad (2)$$

Table 1: LFSR implement notations

Notations	Details
L	The length of the stages
S_n	The sequences of the output
C_i	The initial value

Table1 explain the coefficients of equation 2. known number of stage by the primitive polynomial.
Suppose that, LFSR with degree $= 4$, the path shown in Figure 5 [1,5].

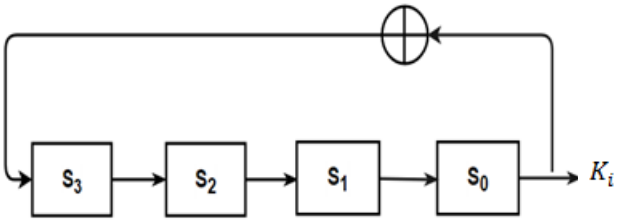

Figure 5: LFSR with degree 4

The sequence of inner state is S_i , it is shifted one bit to right , the leftmost is the output. The output is computed by *XOR* for summation of previous stage. If we suppose seed = 1001 , Table2 illustrate how LFSR sequence is generated

Table 2: Sequence of stages of LFSR

T	S_3	S_2	S_1	S_0
0	1	0	0	1
1	0	1	0	0
2	0	0	1	0
3	1	0	0	1

B. Determine Index Of The State

The message text converted into binary bits then the generated key XORed with message to determine the index of state. See Figure 6 .

Figure 6: S-RADG Design

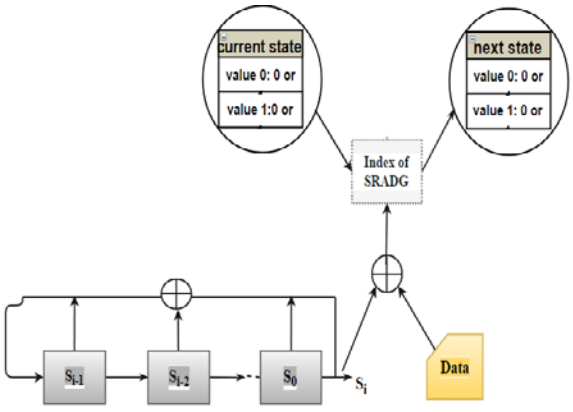

Figure 6: S-RADG Design

3 S-RADG IMPLEMENTATION

Implementation include provide algorithm of key generation , encryption and decryption.

A . KEY GENERATION

Step1: *seed*← `random initial bit`
Step2: *shift all value over one place and dropping the last value*
Step3: *set the computed value in first position*
Step 4: *repeat process*
Step5: *k sequences* $\leftarrow k_i$

B. ENCRYPTION ALGORITHM

Input: Sequence of message $M = \{m_0, m_1, \ldots m_{|M|}\}$, Sequence of key $K = \{K_0, K_1 \ldots . K_{|M|}\}$
Output: sequence of cipher blocks $C \leftarrow \{C_0, C_1, \ldots . \}$

Step1: $C \leftarrow \emptyset, q_{current} = q_{rand}, L \leftarrow 0$
Step2: $while\ (L < |M|)$
Step3: $\text{index} \leftarrow F(m_L, K_L)$

Step4: $(\text{State value})_L \leftarrow getvalue(q_{old}, inde$
Step5: $b_n \leftarrow block(index, state\ value, q_{old})$

Step6: $C_1 \leftarrow DES\ cipher[block]$
Step7: $\text{Add}\ (C_1, C_1(L))$

Step8: $\text{if}(q_{old} \in Jump)$
$q_{new} \leftarrow R$
Step9: *else*
Step10: $q_{new} \leftarrow q_{old}$
Step11: $return\ C$

C. DECRYPTION ALGORITHM

Input : sequence of cipher text $C \leftarrow \{C_0, C_1, \ldots . , C_{|L|}\}$, sequence of key $K \leftarrow \{K_0, K_1, \ldots . , K_{|C|}\}$
Output : sequence of plain text $M \leftarrow \{m_0, m_1, \ldots . \}$
Step1: $block_L \leftarrow decipher[C_L], L \leftarrow |C|$

Step2: $Search\ (q_{old}, Q)$
Step3: $while\ (L <= 0)$

Step4: $m_L \leftarrow F(index, K_L)$

Step5: $Add(P, m_L)$

Step6: $reverse\ sequence(m_0, m_1, \ldots . m_{|L|})$

Step7: $return\ M$

[1] College of Computer Science and mathematics, University of Kufa, Iraq, salah.albermany@uokufa.edu.iq ,
[2] College of Sciences, Computer Science department Al-Nahrain University, Iraq , [stcs-dam16][skt]@sc.nahrainuniv.edu.iq,

5 SECURITY AND PERFORMANCE ANALYSIS

The security and performance analysis of S-RADG clarified by implement example of 12 states , with $n = 4$, $m = 6$, $k = 2$, all states in Figure 6 , have λ=2.
Where $n = |R|$, $m = |Q|$, $k = |J|$.

A. SECURITY ANALYSIS

- **CONFIDENTIALITY**

Confidentiality prevent wrong people to access to information , that mean , it equally to privacy of information [8]. Encryption and decryption can be used for confidentiality in the example below.

***Example*:** The message to be encrypted is *hi*, and the key is generated by LFSR algorithm, which used for both encryption and decryption , so at first, computed the sequences of random key as illustrated in Table 3. Let suppose the primitive polynomial is $P(x) = x^4 + x + 1$. The stages of LFSR illustrated in Figure 7 .

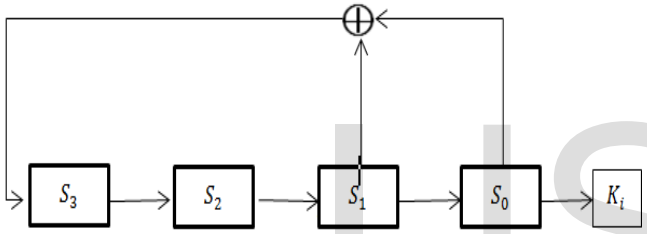

Figure 7: LFSR length

Table 3: sequences of LFSR

t	S_3	S_2	S_1	S_0
0	1	0	1	0
1	1	1	0	1
2	1	1	1	0
3	1	1	1	1

From Table 3 the key sequences are :
K_i = 1010 1101 1110 1111

a. ENCRYPTION DATA

First, converted message into binary value.
hi: 0110 1000 0110 1001

It's important to see appendix I and appendix II together to understand how the encryption is done. A column block in Table 4 , include $8\ bits$ binary data , the left two bits are the index and state value and the others bits are address which is a state number after converted into binary values. Then encrypted the blocks data by using DES algorithm to result a cipher text of S-RADG. DES algorithm encrypt 64 bits at time with *56 bits* of key (the cipher key is normally given as a *64-bits* key in which *8 extra bits* are the parity bits, which are dropped before the actual key-generation process), so if the data size to be encrypted is greater than 64 bits , the DES will encrypt the first *8 blocks* then return to encrypt other blocks so if the other blocks is less than 8 blocks, needed to extended it by using padding DES. The DES encrypt the first *64-bits* then repeat it work for other 64-bits , by use same S-RADG key.

b. DECRYPTION DATA

A decryption process in a receiver side begin backward to result plain text. It start form cipher text, to decrypt it by using DES algorithm, from the $8\ bits$ result , the receiver obtain state number. Then by XORed index with the key to get original message that sent by a sender. See appendix I and appendix II together to understand how exactly decryption process is done.

- **INTEGRITY**

Integrity means detect unauthorized user in writing (i.e., modification of data)[8]. In S-RADG algorithm , integrity is satisfy by notice that any modification on cipher text can detect easily. If the hacker modifies cipher block marked by unauthorized user, the decryption process of the cipher text will failed because the search process is failed where not found the value which must give the correct plain text and in this case cannot be continue to complete decryption process.

B.PERFORMANCE ANALYSIS

Entropy is used to measured uncertainty in the system [7]. Using entropy to analysis performance of S-RADG method, S-RADG is run *100* times to show how it performs in terms of entropy for different cipher text values. Figure 8 show the entropy between individual cipher text for message length of *16 bits*.

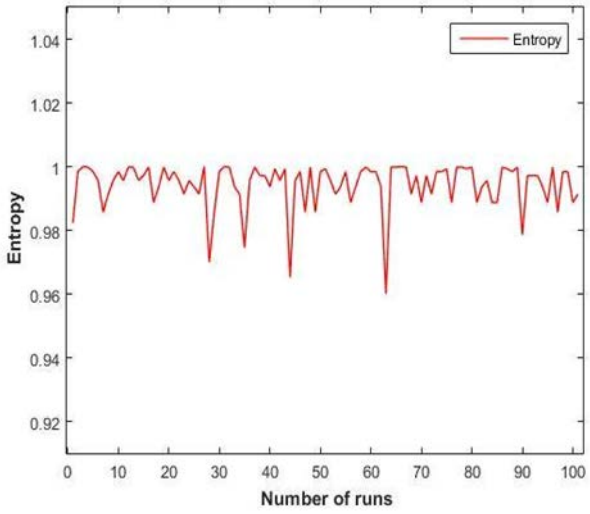

Figure 8: Entropy for message length of 128 bits

6. *CONCLUSION*

The proposed algorithm improved RADG implementation level , makes cryptography stronger since use stream cipher to generate a good keystream based on LFSR to generate random sequenc-

es of key. The LFSR satisfy security since add binary sequences to data, this add noise to information and makes it difficult to detect by unauthorized users. LFSR method have highly long sequences may reach to 10^{50},[4] this makes S-RADG key strong and the hacker need long time to detect. The security analysis prove that S-RADG successfully provides confidentiality, Integrity , which are aspects of security. The results obtained from performance analysis have shown that for the same plain message have the different cipher text ,this proves the strong cryptography of S-RADG where the process of breaking the code within large systems requires a more effort compared to the RADG scheme that not require any key.

REFERENCES

[1] A.Klein, *Stream cipher*. Springer, 2013.
[2] A.Safdar, S.A.A.a.G., *Keyless security in Wireless network.* Springer, 2014.
[3] B.Dharma Teja, V.S., D.Lokesh,lokesh. K.V.R.L.Prasad, *Design and Analysis of a 128 Bit Linear Feedback Shift Register Using VHDL*. International Journal of Advanced Research in Science, Engineering and Technology, 2016. **3**(2).
[4] Brock, T.B., *Linear Feedback Shift Registers in SAGE*. The Art of Computer Programming., 2006.
[5] C.paar, J.P., *understanding cryptography.* springer, 2010.
[6] ghosh, s.s.s., *stream cipher cryptosystem based on linear feedback shift register.* international journal of mathmaticl 2012.
[7] Munkhbayar Bat-Erdene, T.K., Hyundo Park and Heejo Lee, *Packer Detection for Multi-Layer Executables Using Entropy Analysis.* Symbolic Entropy Analysis and Its Applications, 2017. **19**(3).
[8] Stamp, M., *INFORMATION SECURITY Principles and Practice.* Canada , Hoboken, New Jersey., 2011.

Appendix I: **Example of S-RADG (Encryption/Decryption)**

Table 4: S-RADG Encryption

i	State number	Message	Key	Messge⊕Key	Jump	Block	Cipher text
0	3	0	1	1	-	10000011	10011110
1	1	1	0	1	-	11000001	01100101
2	10	1	1	0	J	00001010	01001111
3	8	0	0	0	-	00001000	00101011
4	2	1	1	0	-	01000010	00101010
5	10	0	1	1	J	10001010	10001111
6	2	0	0	0	-	01000010	10100110
7	11	0	1	1	J	10001011	11111001
8	1	0	1	1	-	11000001	01010110
9	9	1	1	0	J	01001001	11100111
10	1	1	1	0	-	00000001	11011110
11	2	0	0	0	-	01000010	00101100
12	10	1	1	0	J	00001010	10001111
13	8	0	1	1	-	10001000	11101010
14	2	0	1	1	-	11000010	11011000
15	10	1	1	0	J	00001010	10001100

Table 5: S-RADG Decryption

i	Cipher text	Block	State number	State value	T^{-1} or search()	Index	Key	Message
15	10001100	00001010	10	0	search	0	1	1
14	11011000	11000010	2	1	T^{-1} is True	1	1	0
13	11101010	10001000	8	0	T^{-1} is True	1	1	0
12	10001111	00001010	10	0	T^{-1} is False then search in R	0	1	1
11	00101100	01000010	2	1	T^{-1} is True	0	0	0
10	11011110	00000001	1	0	T^{-1} is True	0	1	1
9	11100111	01001001	9	1	T^{-1} is False then search in R	0	1	1
8	01010110	11000001	1	1	T^{-1} is True	1	1	0
7	11111001	10001011	11	0	T^{-1} is False then search in R	1	1	0
6	10100110	01000010	2	1	T^{-1} is True	0	0	0
5	10001111	10001010	10	0	T^{-1} is False then search in R	1	1	0
4	00101010	01000010	2	1	T^{-1} is True	0	1	1
3	00101011	00001000	8	0	T^{-1} is True	0	0	0
2	01001111	00001010	10	0	T^{-1} is False then search in R	0	1	1
1	01100101	11000001	1	1	T^{-1} is True	1	0	1
0	10011110	10000011	3	0	T^{-1} is True	1	1	0

Appendix II: **State value and Transitions**

Table6: Values of *Q* set

Number of states	First Value	Second value
1	0	1
2	1	1
3	1	0
4	-	4
5	-	5
6	0	1
7	1	0
8	0	0

Table7: Values of *R* set		
Number of states	**First Value**	**Second value**
9	1	1
10	0	0
11	1	0
12	0	1

Table8: The Transition		
Number of states	**First Value**	**Second value**
1	2	4
2	5	5
3	7	1
4	-	-
5	-	-
6	2	8
7	4	3
8	2	2
9	1	6
10	8	2
11	3	1
12	7	8