Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 6s (2024)

Double Encrypted Symmetric Cryptosystem Using adjacency matrix and Linear Feedback Shift Register (LFSR)

[1*]MPR Murthy, [2]CH. Suneetha
[1]Research Scholar, GSS, GITAM University, Visakhapatnam, India
[2] Associate Professor, Department of Mathematics, GSS, GITAM University, Visakhapatnam, India
[*]Author for Correspondence: mprmurthy75@gmail.com

Article History:

Received: *28-05-2024*

Revised: *16-07-2024*

Accepted: *26-07-2024*

Abstract:
Owing to an extensive and indiscriminate usage of social communicating networks nowadays, transfer of sensitive data genuine people has become hardship and Herculean task. Message encryption is the only tool to safeguard the original information that helps to protect the data from cyber-attacks in its journey via public channel. Graph theory is one of the important techniques used in the encryption process especially in block ciphers. Abundant research has been done in the field of cryptography using several graph theory concepts. This article explains a symmetric block cipher employing the concept of an adjacency matrix in an undirected graph. Here each block is encrypted at two different stages and three rounds at each stage. The first stage of encryption is done with an adjacency matrix and second stage using simple logical XOR operation in a special pattern. In this technique the adjacency matrix is public, the sender and receiver derive new matrices for encryption/decryption from the adjacency matrix using Linear Feedback Shift Register LFSR. LFSR polynomial acts as a secret key which is the agreement between the communicating parties.

Keywords: LFSR Polynomial, Adjacency Matrix, Logical XOR, Encryption, Decryption.

1. Introduction

In recent era graph theory occupies an important and adequate part in the history of Cryptography. Since path finding in Cayley graphs, Ramanujan graphs etc. is tough, these graphs are mostly used in quantum and classical computers. This paper explains a block cipher encrypted twice using adjacency matrix of undirected graph. At this juncture the original message gets encrypted in blocks and every block is encrypted for 3 rounds. Again in every round of encryption, there are two different stages using the concept of adjacency matrix and logical XOR operations. Logical XOR operation is employed on every character included in the message in a special pattern. In this case, the adjacency matrix acts as public key that is familiar to all users. Users generate new matrices from the secret key for each block and for each round. Here using Linear Feedback shift register LFSR acts as a layer of diffusion. Users before communication agree upon to use a LFSR as secret key (primary master key).

2. Literature Survey

Implementation of cryptosystems using several graph structures is a part in the history of cryptography. The concept of application of graph structures in encryption was explained by Natalia Tokareva [1]. In his work the author described the mobile application network using sparse graphs and bent functions. Wael Mahmoud Al Etaiwi [2] developed an innovative complex cipher with the help of minimal spanning trees in complete graphs. Amudha et al., [3] looked into a number of graph theory ideas that were used in cryptography with Euler graphs. A multiple enciphering strategy was presented by Yamuna et al., [4] employing the idea of the Hamilton path in the complete graph. Graph theory ideas were applied to mage encryption by Steve Lu et al., [5] by assigning nodes and edges to pictures. Charles et al.,[6] created a novel hash algorithm to prevent collisions using expander graphs. Using super singular isogeny graphs, Anmaria Costache et al., [7] looked at a number of quantum cryptography security concerns. Later, Hyungrok Jo [8] presented post quantum cryptography with cryptographic hash functions based on Charles expander graphs. Reviewing the literature, the current study provides a novel double encryption approach using an adjacency matrix and a simple logical XOR operation.

2.1 Adjacency Matrix:

If $\boldsymbol{v}$ is set of all vertices, $\boldsymbol{e}$ is set of all edges a graph g ($\boldsymbol{v}$, $\boldsymbol{e}$) is connection between vertices and edges. The path is cyclic if the starting and ending vertices are same

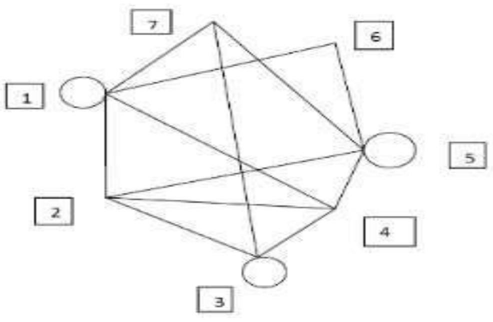
.Adjacency matrix is a symmetric matrix that represents a finite graph [9,10,11]. The adjacency matrix's elements are either 0 or 1.

A = { aij / aij = 0 or 1 }

aij = 1 if the vertices i,j are connected

= 0 otherwise.

The whole matrix is expressed just by considering principal diagonal entries and above elements to principal diagonal.



2.1 Adjacency Matrix Representation of a Graph

0 1 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1 1 1 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 0 1 0 0 1 0 1 0 1 0 0

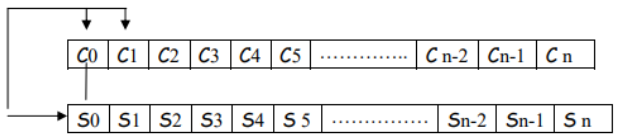
2.2 Linear Feedback Shift Register (LFSR)

An electronic system like ouroboros used for output responses is called Linear Feedback Shift Register (LFSR) [11]. In this efficient design output is fed back to the input, to form a cyclic sequence pattern. Here initial stage is called seed. As the LFSR function is periodic, seed must be non -zero state. An n – bit LFSR has maximum period length 2n-1. A primitive polynomial is selected as LFSR polynomial and seed sequence of bits (0 or 1). For the binary bits at LFSR polynomial positions logical XOR operation is applied. The resulting output is linearly shifted to right and fed to input. The leading bit is replaced by output of logical XOR operation. LFSR polynomial takes the form

1+C0x+c1x2+C2x3+ +Cnxn+1

Where Ci is either 0 or 1 i = o to n – 1

For the polynomial 1+x+x2 with p bit LFSR is



In electronics Linear Feedback Shift Register (LFSR) has several applications. For generating sequence of parallel pseudo random numbers LFSR is one of the tools in cryptography.

LFSR is mostly used in light weight stream ciphers, due to speedy encryption with low computation cost and required properties. At initial stages of cryptography LFSR was more popular. Some insecurity was noticed in LFSR due to its periodicity nature. Later on, many researches were conducted to avoid the periodicity of LFSR, found that repetition can be removed by considering large LFSR. Multiple large LFSR ‘s can increase the security of the cipher also . In the algorithm presented in this chapter LFSR is used as diffusion layer for generating encryption keys for different blocks from the primary master key.

3. Methodology

Here, the public key of an undirected graph is its adjacency matrix A.. Before the communication the legitimate communicators of the group share an irreducible polynomial as the secret key whose order is less than the adjacency matrix order. Each user has the option to share the secret key, or a reliable third party service may be used. The present cipher is block cipher. The original message is broken up into data blocks, each of which has a size equal to the square of the adjacency matrix's order. When dealing with a matrix of order l, block length is l2. As mentioned above the elements of entire adjacency matrix is denoted by diagonal elements and above diagonal elements. Example, elements of 5th order adjacency matrix are denoted as

[a11 a12 a13 a14 a15 a21 a22 a23 a24 a25 a31 a32 a33 a34 a35 a41 a42 a43 a44 a45 a51 a52 a53 a54 a55]

For instance adjacency matrix of 5th order is denoted by the stream 0101000010 .

This key stream pertains to matter key stream. LFSR is applied on master key stream to generate keys for three rounds of encryption of each block. LFSR polynomial acts as secret key. Original message characters are coded to ASCII decimal numbers and split into data blocks each having length square of order of adjacency matrix l2. All the decimal numbers of data blocks are arranged as l × l square matrices say m1 ,m2 ---- . Every block is enciphered at two stages using adjacency matrix at stage 1 and implementing XOR operation on each element at stage 2 with neighbouring elements of the matrix in a special pattern. Three round two stage encryption is done using distinct keys derived from the master key. For enhanced security of the cipher first stage encrypted characters are flipped before the second stage is (logical XOR) administered

3.1 Encryption:

Symbols and codes used in this technique are

C [(𝓅),(𝐪),(𝕤)] stands for cipher test for 𝓅th block , 𝐪 th stage and 𝕤th round A [(𝓅),(𝕤)] represents adjacency matrix used for 𝓅th block and 𝕤th round

R [(𝓅),(𝕤)] stands for a decimal number , the power of the adjacency matrix for 𝓅th

block and 𝕤th round

Where 𝓅 = 1,2,3 (block number)

= 1,2 (stage number)

= 1,2,3 (round number)

3.2 Encryption at stage 1:

At this point, the adjacency matrix A's binary stream serves as the seed for the LFSR. The bits are right shifted by replacing the first bit with a new bit and keeping the last bit of the seed in place with the help of the irreducible LFSR polynomial. Output binary number stream first state of LFSR represents elements for new adjacency matrix A [(1) (1)] . This adjacency matrix is used as key for first block matrix M1 encryption .

First block key A [(1) (1)] is raised to the power r [(1) (1)] . Here power r [(1) (1)] is obtained by converting first four bits of state are LFSR binary stream to decimal number . Since the keys for encrypting blocks are generated from the master key (agreed adjacency matrix) , by applying LFSR at each stage , this power to which the adjacency matrix raised is secret which cannot be broken unless the master key is compromised.

First data block matrix m1 is multiplied with A [(1) (1)]r[(1) (1)] to get the cipher matrix C [(1) (1)] at first

stage C [(1) (1)] = m1 * A [(1) (1)]r[(1) (1)]

The matrix C [(1) (1)] is flipped by interchanging rows to get C[(1)(1) F(1)]

3.3 Encryption stage 2:

At this stage the first stage cipher matrix C [(1) (1) F(1)] whose entries are Only the exclusive XOR operation with diagonal elements will be applied to decimal values. XOR procedure that replaces each element with a new resultant element by wrapping diagonal elements around each element in a unique way and operating counter clock judiciously.

Multiple XOR operations on the same number is prominent technique in cryptography that adds more strength to the cipher

For example, the element a33 is counter-clockwise XORed with a24, a22, a42, and a44, and the resultant new number replaces a33. ((((a33 XOR a24) XOR a22) XOR a42) XOR a44) is new a33 .

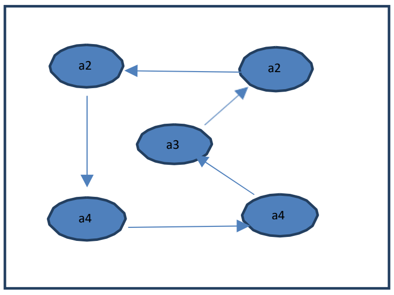


Fig.2.2: XOR Pattern for middle elements

Regarding the matrix's border elements, the corresponding last or first row / column elements are used for XORing. The first element a11 is XORed using the end elements as follows

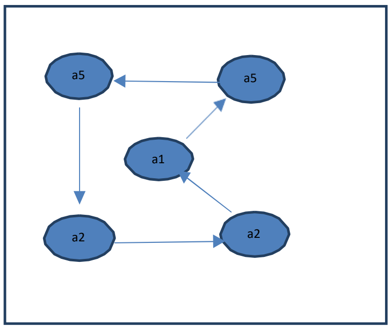


Fig. 2.3: XOR pattern for end elements

Logical XOR is applied on each element of the matrix C[(1)(1) F(1)] In the above explained pattern to get the outcome matrix C[(1) (2) (1)] same two stage encryption procedure repeated in two more rounds on C[(1) (2) (1)] using adjacency matrices A [(1) (2)] , A [(1) (3)] . A [(1) (2)] and A [(1) (3)] are adjacency matrices obtained by expanding binary streams of LFSR second and third states. Here is the algorithm for one block briefly depicted

C[(1)(1)(1)] $= M1 * A[(1)(1)]^{r[(1)(1)]}$ C[(1)(1)F(1)] $= Flip$C[(1)(1)(1)] C[(1)(2)(1)] $= XOR$ C[(1)(1)F(1)]	C[(1)(1)(2)] = C[(1)(2)(1)] * $A[(1)(2)]^{r[(1)(2)]}$ C[(1)(1)F(2)] $= Flip$C[(1)(1)(2)] C[(1)(2)(2)] $= XOR$ C[(1)(1)F(2)]	C[(1)(1)(3)] = C[(1)(2)(2)] * $A[(1)(3)]^{r[(1)(3)]}$ C[(1)(1)F(3)] $= Flip$C[(1)(1)(3)] C[(1)(2)(3)] $= XOR$ C[(1)(1)F(3)]

Two stage, 3 round encryption procedure is applied for all data block matrices. Key for each block and each round is different, generated from LFSR continuously

I ,II , III states of LFSR are keys (adjacency matrices and powers) for first block . IV , V , VI states are keys for second block and so on .

C[(1)(1)F(3)] C[(1)(1)F(2)] C[(1)(1)F(1)]

= XOR [(1)(2)(3)]   = XOR C[(1)(2)(2)]   = XOR C[(1)(2)(1)]

[(1)(1)(3)]   [(1)(1)(2)] C[(1)(1)(1)]

= $Flip$ C[(1)(1)F(3)]   = $Flip$ C[(1)(1)F(2)]   = $Flip$ C[(1)(1)F(1)]

[(1)(2)(2)]

= [(1)(1)(3)]

$*$ inv{A[(1)(3)]}r[(1)(3)] C[(1)(2)(1)]

= [(1)(1)(2)]

$*$ invA{[(1)(2)]}r[(1)(2)] M1

= [(1)(1)(1)]

$*$ invA{[(1)(1)]}r[(1)(1)]

LFSR is an interesting random number generator technique, but it is periodic and eventually coincides with seed. So, at same point the keys for different blocks will coincide. There may be a chance that two message blocks and keys for encrypting those two blocks coincide. In such a case the cryptanalyst becomes easy for an attacker. The cipher becomes susceptible to correlation attacks by differential and linear cryptanalysts. Selecting the largest seed possible is one way to prevent this. Another way to overcome this is after LFSR is exhausted (if any state coincides with the seed) the parity bit is applied on original LFSR. The elements of cipher block matrices C [(1) (2)(3)]C[(2)(2)(3)] are coded to ASCII Characters which is cipher text C.

3.4 Decryption:

As the present cipher is symmetric block cipher, decryption is just the reverse process of encryption. The receiver first divides the cipher test into cipher blocks of l2 length each. Each block's l2 characters are all coded to corresponding decimal numbers and arranged as order l * l square matrices.. Description from first block as follows.

3.5 Stage-1 Decryption:

At second stage encryption, logical XOR is applied on each entry with surrounding diagonal elements. New number of each entry is concatenated in XOR process of second entry row wise.

This special XOR pattern is implemented from first element row wise. So the decryption at this stage is just the logical XOR on each entry with surrounding diagonal elements in the opposite direction to encryption (clockwise direction).

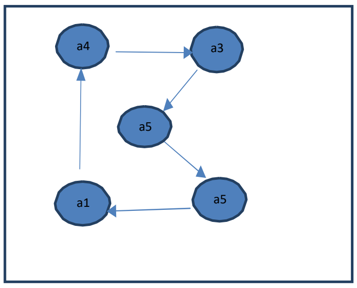


Fig.2.4 XOR Decryption pattern

$[(1)(1)F(3)] = XOR\ [(1)(2)(3)]$

$[(1)(1)(3)] = Flip\ [(1)(1)F(3)]$

The adjacency matrix's power is determined by the decimal equivalency of the first four bits of the LFSR third state. The adjacency matrix's inverse, raised to the preceding decimal place, is multiplied by [(1)(1)(3)]

$C[(1)(2)(2)] = C[(1)(1)(3)] * \mathrm{inv}\{A[(1)(3)]\}r[(1)(3)]$

The matrix M1 is obtained after two more rounds of the same process. The three round two-stage decryption is done for all the blocks and the decimals are coded to ASCII characters to get back the original message.

$C[(1)(1)F(3)] = XOR\ C[(1)(2)(3)]$ $C[(1)(1)(3)] = Flip\ C[(1)(1)F(3)]$ $C[(1)(2)(2)] = C[(1)(1)(3)] * \mathrm{inv}\{A[(1)(3)]\}^{r[(1)(3)]}$	$C[(1)(1)F(2)] = XOR\ C[(1)(2)(2)]$ $C[(1)(1)(2)] = Flip\ C[(1)(1)F(2)]$ $C[(1)(2)(1)] = C[(1)(1)(2)] * \mathrm{inv}A\{[(1)(2)]\}^{r[(1)(2)]}$	$C[(1)(1)F(1)] = XOR\ C[(1)(2)(1)]$ $C[(1)(1)(1)] = Flip\ C[(1)(1)F(1)]$ $M1 = C[(1)(1)(1)] * inv\mathrm{A}\{[(1)(1)]\}^{r[(1)(1)]}$

4. Implementation

The execution of the algorithm was performed using Intel core processor with speed 2.10 GHZ, using MATLAB 2020 trial version.

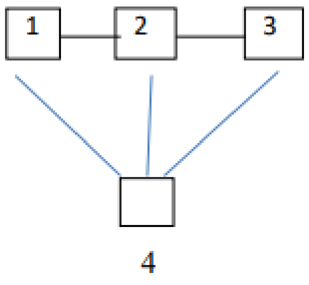


Fig.: 2.5 An undirected graph

The corresponding adjacency matrix is

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Know to all. 0101011010 is the representation of the entire matrix because it is symmetric. Two individuals who desire to exchange messages, agree upon to use LFSR polynomial x2+x+1. This is secret key shared either personally or taking the help of genuine third party. Seed and next three stated of LFSR are represented as1010101101, 1101010110, 0110101011 are adjacency matrices used for encryption of first block matrix in the rounds. The power of adjacency matrices are decimal equivalents of first four bits 9,13,6. If message is ‘SUNEETHA ‘consists of 8 letters. Block length is 16 [l2, 42].

0	1	0	1	0	1	1	0	1	0	Seed
1	0	1	0	1	0	1	1	0	1	0 I state
1	1	0	1	0	1	0	1	1	0	I II state
0	1	1	0	1	0	1	0	1	1	0 III state

To complete the block random characters maybe added. After two stage three round encryption the cipher text is ‘S! E3 A DC 3 Y J ‘.

In case of incomplete blocks, random characters are used to complete the block. While communications the cipher sends may add ###(of the cipher at the end) to mention the end of the cipher text . It represents other characters may be neglected. Again it repeated message ‘SSSSSSSSSS ‘is to be communicated, encrypted in the same process to set same as in file.

4.1 Security Analysis:

Here adjacency matrix is general key known to all. LFSR polynomial is secret key between the users. Adjacency matrix binary stream is seed for LFSR. Here LFSR acts as diffusion layer that adds in increased resistance against many well-known block cipher attacks. The exponents of adjacency matrices are also derived from a LFSR states. General attack on LFSR is the adversary tries to recover seed or state field. But the seed of LFSR is simply a tool to generate keys for encryption of blocks and also rounds. Even, this seed is public. So, correlation attack is not possible to execute here. It is safer to use a block cipher. It is safeguarded by a strict, well-defined mathematical framework. Here, the cipher's strength is provided by mathematical modelling that involves "multiplying each data block matrix with same power of adjacency matrix at first stage, XORing every entry of the matrix with surrounding diagonal elements in a special pattern at second stage". The concatenation of each element to the succeeding element in XOR process is addition security as it is tough to crack the cipher after three rounds. Since the keys are different for different rounds the repeated message is mapped to non – repeated cipher. So, linear cryptanalysis and differential cryptanalysis are hard to implement. English alphabet repetition for 0.1 Mb data is depicted in the below given histogram.

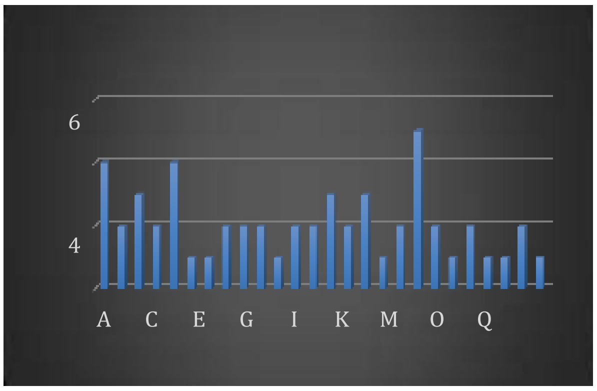


Fig 2.6: Graph Showing the Frequency of English Alphabets in 0. 1 Mb data

Performance Analysis: The encryption and decryption times are computed, tabulated, and plotted for varying data sizes in Figure 4.

Table 4.1. Execution Time

Size of the data in megabytes	Encryptio n time (sec)	Decryptio n time (sec)
0.001	0.465	0.514
0.002	0.974	1.025
0.003	1.401	1.384
0.004	1.745	1.770
0.005	2.467	2.681
0.006	3.133	3.203
0.007	3.429	3.525
0.008	4.125	4.344
0.009	4.433	4.505
0.01	4.937	5.023

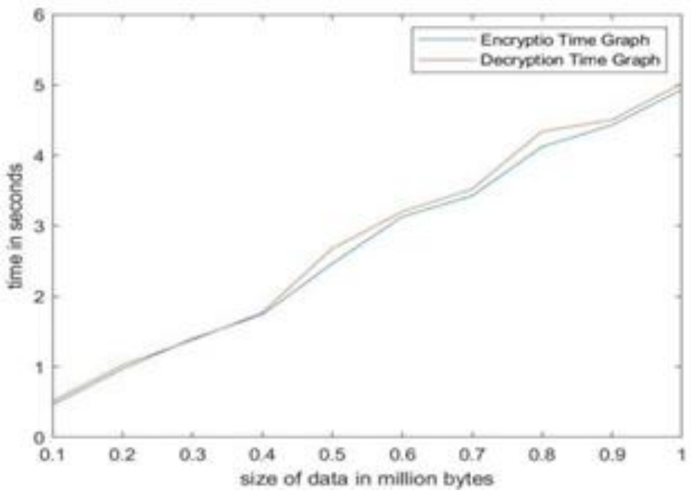


Fig. 2.7 Encryption/Decryption Time Graph

5. Conclusion

From the graph it is clear that time of execution is almost linear. The time is roughly 5 sec for encrypting the data of size 1 megabytes. This is almost nearer to conventional ciphers AES(5.66 sec), DES(4.1 sec) roughly (from the literature) . So, this paper describes more secure symmetric block cipher, competes with conventional blocks ciphers.

References

[1] Natalia Tokareva, “Connection between graph theory and cryptography G2C2:Graphs and groups, Cyclic and Coverings, Sep 2014, Novosibirsk Russia.

[2] Wael Mahmoud Al Etaiwi, “Encryption algorithm using graph theory”, Journal of scientific research and reports, 3(19) 2519-2527, 2014.

[3] P. Amudha, A.C. Charles Sagayaraj, A.C. shantha Sheela, “An application of graph theory in cryptography” , International journal of pure and applied mathematics IJPAM, 119(3), 2018, pp 375-383.

[4] Yamuna M, Meenal Gogia, Ashish sikka,Md.Jazib Hayat Khan, “Encryption using graph theory and linear algebra”, International journal of computer applications IJCA, 2012.

[5] S Lu, Rafael Ostrovsky, Daniel Manchala, “Visual cryptography on graphs”, Cite Seeix COCOON, 2008, 225-234.

[6] Denis X Charles, Eyal Z. Goren, Kristic E. Lauter, “Cryptographic hash functions from expander graphs”, Journal of Cryptology, 22, 93-113, 2009.

[7] Anmaria Costache, Brooke Feigon, Kristin Lauter, Maike Masierer, Anna Puskar, “Ramanujan graphs in cryptography”, arXiv:1806. 05709V2 [math.NT]18th December 2018.

[8] Hyungrok Jo, “Ramanujan graphs for post quantum cryptography”, http://www.researchgate.net/publication/337150777.

[9] Yvo Desmedt, Josef Piprzyk, Ron Sternfield, Xiaoming Sun and Tartary, “Graph Coloring Applied to Secure Computation in Non-Abilian Groups”, Journal of Cryptology, Springer, online from 5th September 2011

[10] C. Blundo, A. Santis, D.R. Stinson and U. Vaccaro “Graph decompositions and secret sharing schemes”, Journal of Cryptology, 1995, Vol.8, No.1, pgs 39-64.

[11] Connections between graph theory and cryptography http://en.wikipedia.org/wiki/Graph_theory