Frontiers in Health Informatics
ISSN-Online: 2676-7104
www.healthinformaticsjournal.com
2024; Vol 13: Issue 8
Open Access

Graph-Based Encryption and Decryption Algorithms in Symmetric Key Cryptography

Hemant Gena[a], Dr. Binny Gupta[b]
[a]Scholar,Department Of Mathematics, SKD University, Hanumangarh, Rajasthan, india
hemantgena1010@gmail.com
[b]Assistant professor, Department Of Mathematics, SKD University, Hanumangarh, raj.
binnygupta1519@gmail.com

Cite this paper as: Hemant Gena , Dr. Binny Gupta (2024). Graph-Based Encryption and Decryption Algorithms in Symmetric Key Cryptography. Frontiers in Health Informatics, 13 (8) 947-955

ABSTRACT

Modern society relies extensively on digital technology, which plays a critical role in various domains, including banking, e-commerce transactions and cyber security such as managing computer passwords. This widespread integration highlights the importance of secure, efficient digital systems in our daily lives. Therefore, safeguarding information during storage and sharing is essential. Cryptography is the study of secure communication that uses advanced mathematical algorithms to transform an original message into an unreadable format. Cryptography is the science of protecting information by transforming it into a secure format so that only authorized parties can understand it. It involves techniques and algorithms that convert plain data, or "plaintext," into an unreadable format, called "ciphertext," using encryption methods. Graph theory is utilized in cryptography since graphs can be easily transformed into matrices. This paper introduces a novel link between graph theory and symmetric cryptography to safeguard information from unauthorized access. The suggested method uses a matrix as the secret key, which increases the security of the cryptosystem. It transforms the plaintext into multiple graphs and displays these graphs in matrix form.

Keywords:
Multiple graph, Encryption, Matrix, Symmetric key cryptography.

1.Introduction

The field of cryptography is generally divided into two main branches: cryptography and cryptanalysis [5]. Cryptography refers to the science and practice of designing secure communication systems. Its primary goal is to create methods and algorithms that ensure the confidentiality, integrity, authentication, and non-repudiation of information. This includes encryption algorithms, digital signatures, hashing functions, key exchange protocols, and more. Cryptanalysis is the art and science of breaking cryptographic systems. It involves studying encryption methods in order to find weaknesses or vulnerabilities, allowing an adversary to decrypt data without the secret key or otherwise compromise the security of the system. Cryptanalysts often attempt to break cryptographic systems by discovering patterns or exploiting mathematical weaknesses. In essence, cryptography focuses on securing communication, while cryptanalysis focuses on breaking that security. Both fields work together to improve the strength and reliability of cryptographic system.

In a cryptographic system, the sender is the individual or system that creates and sends the original message (plaintext), while the receiver is the intended recipient who receives and interprets the message. The attacker is an unauthorized entity who tries to intercept, read, or manipulate the message in an attempt to compromise the system's security. To protect the message from being intercepted by attackers, the sender uses encryption, which converts the plaintext into an unreadable format (cipher text) using an encryption algorithm and a key. Once the encrypted message reaches the receiver, they use decryption to transform the ciphertext back into its original form (plaintext) using a decryption key, allowing them to read the message. In symmetric-key encryption, both the sender and receiver use the same key for both encryption and decryption, which must be kept secret between them. This type of encryption is efficient but requires a secure method for sharing the key between the communicating parties. Symmetric Key Cryptography and Asymmetric Key Cryptography are two fundamental types of encryption techniques used to secure communication and data.

1.1 Symmetric Key Cryptography

In symmetric key cryptography, both the sender and the receiver use the same key for both encryption and decryption. The key must remain secret between the two parties to ensure the security of the communication. This type of encryption is efficient and fast, making it suitable for encrypting large amounts of data. However, the challenge lies in securely sharing the key between the sender and receiver. If the key is intercepted by an attacker during transmission, they can decrypt the message. Common algorithms that use symmetric encryption include AES (Advanced Encryption Standard) and DES (Data Encryption Standard)

1.2 Asymmetric Key Cryptography

In asymmetric key cryptography (also known as public-key cryptography), two different keys are used: a public key and a private key. The public key is shared openly with anyone, while the private key is kept secret by the owner. In this system, the sender encrypts the message using the receiver's public key, and only the receiver can decrypt it using their private key. This eliminates the need to share a secret key ahead of time, solving the key distribution problem present in symmetric key cryptography. Asymmetric cryptography is slower than symmetric cryptography because of the more complex mathematical operations involved. Symmetric key cryptography uses the same key for both encryption and decryption, making it fast but requiring secure key sharing. Asymmetric key cryptography uses a pair of keys (public and private) to encrypt and decrypt messages, which provides more secure key exchange but is slower.

In [2], a new way of applying graph theory in cryptography is discussed. The original message is represented in terms of a graph, such as a Hamiltonian graph (a graph that contains a Hamiltonian cycle, meaning a cycle that visits each vertex exactly once). This can paths in the graph.This graph is then converted into several matrices and sends to the receiver.In [2], a novel approach to utilizing graph theory in cryptography is introduced. The original message is encrypted into a Euler graph, which is subsequently transformed into multiple matrices and transmitted to the receiver. In [4], a modified Affine cipher algorithm is presented. In this method, each character of the plaintext is transformed into a numeric value, which is then represented as points on a graph. This graphical representation is transmitted to the receiver. In [1,3],a novel encryption algorithm introduced based on graph theory. In this approach, the plaintext is divided into multiple blocks, which are then transformed into graphs. The graphical representations are subsequently transmitted to the receiver.

This paper proposes a novel algorithm for encrypting and decrypting messages using graph theory, specifically leveraging the properties of Hamiltonian graphs in symmetric key cryptography. The remaining sections are organized as follows: Section II presents the mathematical preliminaries essential to the proposed algorithm. The methodology is detailed in Section III, while Section IV provides the conclusion and final remarks.

2.Theoretical preliminaries

2.1 Graph-

A graph is a pair G= (V,E) ,where V={ v_1,v_2,v_3,….} is a set of vertices (also called nodes) and E={e_1,e_2,e_3……} is a set of edges that connect pairs of vertices. The edges can be directed or undirected, depending on whether the connections have a direction (like arrows) or not.

2.2 Complete Graph-

A complete graph is a type of graph in which every pair of distinct vertices is connected by a unique edge. In other words, a complete graph has an edge between every possible pair of vertices. A complete graph with n vertices is denoted by K_n. A complete graph with n vertices has n(n-1)/2 edges (in an undirected graph) [6].

2.3 Hamiltonian Graph-

A Hamiltonian graph is a graph that contains a Hamiltonian cycle. A Hamiltonian cycle is a cycle that visits each vertex of the graph exactly once and returns to the starting vertex. Every complete graph K_n (where $n \geq 3$) is a Hamiltonian graph.

2.4 Weighted Graph-

A weighted graph is a graph in which each edge is assigned a numerical value, called a weight. These weights typically represent costs, distances, or other metrics associated with traversing the edges between vertices [6].

2.5 Adjacency Matrix-

An adjacency matrix is a square matrix $D=(d_{ij})$ used to represent a graph G = (V,E), where each element indicates whether pairs of vertices are adjacent or not in the graph. It is especially useful for representing weighted and unweighted graphs. For an unweighted graph, the matrix contains 1s (or True) to indicate that an edge exists between two vertices, and 0s (or False) otherwise. For a weighted graph, the matrix contains the weight of the edge between the vertices if an edge exists, and 0 (or some other marker, like infinity) if no edge exists.

3. Proposed Methodology

Table-1 Encoding Table

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	g	h	i	j	k	l	m
26	27	28	29	30	31	32	33	34	35	36	37	38
n	o	p	q	r	s	t	u	v	w	x	y	z
39	40	41	42	43	44	45	46	47	48	49	50	51

3.1 Steps of Encryption–

1) Select a key (K) which is a square invertible matrix.
2) Define the values for a and b satisfying the conditions; $gcd(a, m) = 1$ and $0 \leq a, b \leq m$ where m=52.
3) Convert each character of the plain text in to a numeric value (x), using the encoding table(Table-1).
4) Obtain the letters(y), corresponding to the value $(ax + b) \; mod \; m$, using the encoding table.
5) Obtain the numeric values (z),where z is the UTF-8 (Decimal) value of y.
6) Obtain the letters (d), corresponding to the value $(z - r) \; mod \; m$,using the encoding table. Here r is the difference between maximum and minimum indexes of the encoding table.
7) Divide these letters into several blocks of size n-1 If block size<n-1, add padding characters to complete the block size.
8) Represent each character of the block as vertices. Connect each vertex with a weighted edge.
9) Make the graph complete by adding extra edges with random weights greater than m.
10) Identify a special character for each block. If it is the initial block, then the special character is the letter corresponding to the summation of elements in K. If it is not the initial block, then the special character is the last character of the previous block. Add this special character to the beginning of the block.
11) Construct the corresponding adjacency matrix (M).
12) C = M×K and send C to the receiver..

***3.2* Steps of Decryption–**

1) Receive several matrices as cipher text (C).
2) Calculate the matrix (M) where $M = C \times K^{-1}$
3) Draw the weighted graph whose adjacency matrix is M.

4) Identify the special character of the initial block by using K.
5) Convert all the vertices into letters using the encoding table and the edge weights
6) 6)Convert all the characters (ignore the special characters) into numeric values (s), using the encoding table.
7) Let p=s+(m×q) where $q = \begin{Bmatrix} 0; if\ s \geq 19 \\ 1; if\ s < 19 \end{Bmatrix}$
8) Obtain the letters (y) whose ASCII value is p+r.
9) Obtain the corresponding numeric value (x) of y using the encoding table.
10) Obtain the letters corresponding to the value $a^{-1}(x-b)$ which is the plaintext.

3.3 Example–

Suppose the plain text is “SeptembEr Last”.

Let a=19, b=43 and

determinent of exists. From m=52 , r=51-0 = 51,

K^{-1}

K = 8 is ≠0 .Hence the encoding table, from K, n=5.

Table-2 Encryption mechanism

plaintext	S	e	p	t	e	m	b	E	r	L	a	s	t
x	18	30	41	45	30	38	27	4	43	11	26	44	45
(19x+43)mod 52	21	41	42	14	41	37	36	15	28	44	17	47	14
y	V	p	q	O	p	l	k	P	c	s	R	v	O
z	86	112	113	79	112	108	107	80	99	115	82	118	79
(z-51)mod52	35	9	10	28	9	5	4	29	48	12	31	15	28
d	j	J	K	c	J	F	E	d	w	M	f	P	c

Table-2 shows the calculations that need to be done in order to obtain the first set of ciphertext.

$$block\ size = 4$$

$$\therefore Number\ of\ blocks = \frac{plaintext\ size}{block\ size} = \frac{13}{4} = 3.25 \cong 4$$

Convert each character of the block into vertices and connect these vertices with weighted edges. See Figure1 below.

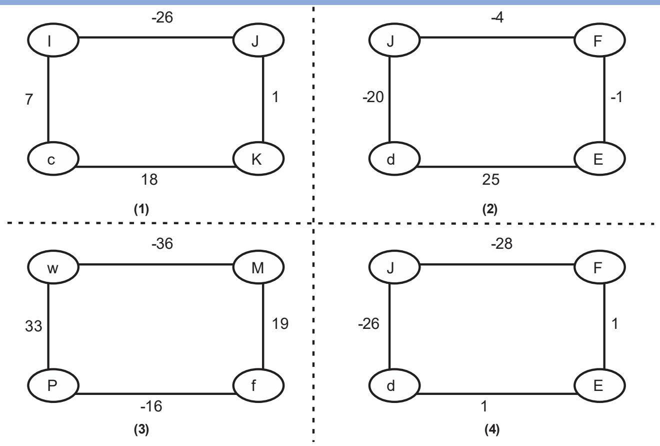


Figure1: Weighted graphs

The weights of the edges are given by using the encoding table.

For example, the weight of the edge iJ is calculated as follows:

Weight jJ =(index of J)-(index of j)

= 9-35

= (- 26)

Make these graphs complete by adding extra edges with random weights greater than m.

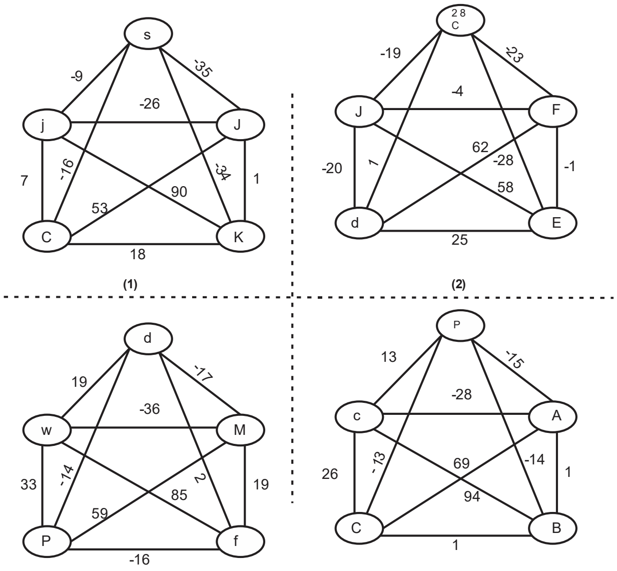
Next, identify the special character for each block.

Special character of the initial block= char((sumK)mod m)

=char(44 mod52)

=char(44) = s

Special characters of the other blocks are assigned to be the last character of the previous block. Add these special characters to the beginning of each block. Figure 2 shows the complete weighted graphs with the special characters



The adjacency matrices (M_i) of the resulted graph are multiplied with the key matrix (K). (Here i is the index of the block.)

$C_i = M_i * K$

$$\begin{bmatrix} -1 & 5 & 1 & 1 & -2 \\ 4 & 1 & 0 & 0 & 5 \end{bmatrix}$$

$$C1 = \begin{bmatrix} 0 & -9 & -35 & -34 & -16 \\ -9 & 0 & -26 & 90 & 7 \\ -35 & -26 & 0 & 1 & 53 \\ -34 & 90 & 1 & 0 & 18 \\ -16 & 7 & 53 & 18 & 0 \end{bmatrix} = \begin{bmatrix} -141 & -247 & -377 & -48 & -125 \\ -377 & 585 & 325 & 628 & 361 \\ 34 & -194 & 76 & -82 & 102 \\ 435 & -80 & 7 & -56 & 570 \\ 255 & 53 & 339 & -120 & 15 \end{bmatrix}$$

$$C2 = \begin{bmatrix} 0 & -19 & -23 & -28 & 1 \\ -19 & 0 & -4 & 58 & -20 \\ -23 & -4 & 0 & -1 & 62 \\ -28 & 58 & -1 & 0 & 25 \\ 1 & -20 & 62 & 25 & 0 \end{bmatrix} \begin{bmatrix} -1 & 5 & 1 & 1 & -2 \\ 4 & 1 & 0 & 0 & 5 \\ 5 & 0 & 5 & -4 & -2 \\ -3 & 7 & 5 & 6 & 3 \\ 2 & 0 & 2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} -105 & -215 & -253 & -77 & -130 \\ -215 & 311 & 211 & 365 & 160 \\ 134 & -126 & 96 & -91 & 209 \\ 305 & -82 & 17 & -49 & 423 \\ 154 & 160 & 436 & -97 & -151 \end{bmatrix}$$

$$C3 = \begin{bmatrix} 0 & 19 & -17 & 2 & -19 \\ 19 & 0 & -26 & 85 & 33 \\ -17 & -26 & 0 & 19 & 59 \\ 2 & 85 & 19 & 0 & 16 \\ 1 & -20 & 62 & 25 & 0 \end{bmatrix} \begin{bmatrix} -1 & 5 & 1 & 1 & -2 \\ 4 & 1 & 0 & 0 & 5 \\ 5 & 0 & 5 & -4 & -2 \\ -3 & 7 & 5 & 6 & 3 \\ 2 & 0 & 2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 117 & 33 & 57 & -37 & 10 \\ -338 & 690 & 380 & 600 & 368 \\ -26 & 22 & 196 & 38 & 138 \\ 465 & 95 & 129 & -90 & 431 \\ 398 & 50 & 356 & -159 & 133 \end{bmatrix}$$

$$C4 = \begin{bmatrix} 0 & 13 & -15 & -14 & 1 \\ 13 & 0 & -28 & 69 & 26 \\ -15 & -26 & 0 & 1 & 441 \\ -14 & 69 & 1 & 0 & 1 \\ 1 & 26 & 94 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 5 & 1 & 1 & -2 \\ 4 & 1 & 0 & 0 & 5 \\ 5 & 0 & 5 & -4 & -2 \\ -3 & 7 & 5 & 6 & 3 \\ 2 & 0 & 2 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 21 & -85 & -143 & -25 & 56 \\ -308 & 548 & 270 & 513 & 315 \\ 88 & -96 & 178 & -103 & 175 \\ 297 & -1 & -7 & -19 & 374 \\ 570 & 38 & 476 & -369 & -57 \end{bmatrix}$$

To decrypt, multiply the received ciphertexts (C_1) with K^{-1}

$$M_i = C_i \times K^{-1}$$

$$K^{-1} = \frac{1}{8} \begin{bmatrix} -280 & -220 & 224 & -204 & 268 \\ -294 & -236 & 238 & -212 & 284 \\ 238 & 188 & -190 & 172 & -228 \\ 242 & 192 & -194 & 176 & -232 \\ 220 & 180 & -180 & 156 & -212 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 0 & -9 & -35 & -34 & -16 \\ -9 & 0 & -26 & 90 & 7 \\ -35 & -26 & 0 & 1 & 53 \\ -34 & 90 & 1 & 0 & 18 \\ -16 & 7 & 53 & 18 & 0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & -19 & -23 & -28 & 1 \\ -19 & 0 & -4 & 58 & -20 \\ -23 & -4 & 0 & -1 & 62 \\ -28 & 58 & -1 & 0 & 25 \\ 1 & -20 & 62 & 25 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 19 & -17 & 2 & -19 \\ 19 & 0 & -26 & 85 & 33 \\ -17 & -26 & 0 & 19 & 59 \\ 2 & 85 & 19 & 0 & 16 \\ 1 & -20 & 62 & 25 & 0 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 0 & 13 & -15 & -14 & 1 \\ 13 & 0 & -28 & 69 & 26 \\ -15 & -26 & 0 & 1 & 441 \\ -14 & 69 & 1 & 0 & 1 \\ 1 & 26 & 94 & 1 & 0 \end{bmatrix}$$

Now the graphs of whole adjacency matrices are M_is. see Figure 3 below

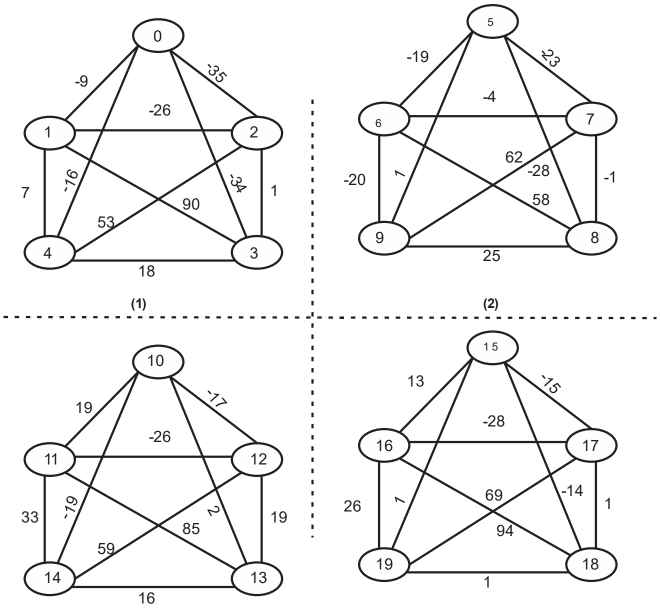


Figure 3 Simplified graph

We Know,

Special character of the initial block =char((sumK)mod m)
=char(44 mod52)
=char(44)
=s

i.e. 0th vertex = s

next, convert all the other vertices into letter using the encoding table.

For example, 1st vertex can be converted into a letter as follows:

1st vertex -0th vertex = (-9)

1st vertex -44 = -9; because from the encoding table s=44

1st vertex = -9+44=35

1st vertex =j; from the encoding table

Similarly, obtain the letter representation of the other vertices as well. Ignore the letters from special characters.

Table-3 Decryption Mechanism

Ciphertext	j	J	K	c	J	F	E	d	w	M	f	P	c
s	35	9	10	28	9	5	4	29	48	12	31	15	28
P=s+(52*q)	35	61	62	28	61	57	56	29	48	64	31	67	28
P+51	86	112	113	79	112	108	107	80	99	115	82	118	79
y	V	p	q	O	p	l	k	P	c	s	R	v	O
x	21	41	42	14	41	37	36	15	28	44	17	47	14
$(19)^{-1}(x-43)$ Mod52	18	30	41	45	30	38	27	4	43	11	26	44	45
plaintext	S	e	p	t	e	m	b	E	r	L	a	s	t

The calculation to be done in order to obtain the plaintext is in Table-3.
Ignore the last three characters as they are the padding characters.
Therefore,
Plaintext = SeptembErLast

Discussion-

In modern society, digital tools are integral to daily life, making the protection of shared information a critical task. Numerous algorithms have been developed to safeguard data from unauthorized access. An algorithm is considered secure if its ciphertext remains concealed even when an attacker possesses complete knowledge of the algorithm. Thus, the security of an algorithm depends on factors such as the difficulty of guessing the secret key and the challenge of deducing the plaintext from the ciphertext, even if the attacker has full access to it.

The proposed method produces a ciphertext larger than the plaintext, enhancing security. It employs a $n\times n$ matrix as the secret key, which significantly increases the difficulty of guessing the key. Ciphertext is generated using matrix multiplication, adding another layer of protection to the plaintext. Moreover, the method creates multiple matrices as ciphertext, reducing the likelihood of an attacker obtaining the complete ciphertext and strengthening the algorithm's resistance to cryptanalysis.

CONCLUSION

Symmetric key cryptography uses the same key for both encryption and decryption, making it fast but requiring secure key sharing.In symmetric key cryptography, both the sender and the receiver use the same key for both encryption and decryption. The key must remain secret between the two parties to ensure the security of the communication. This type of encryption is efficient and fast, making it suitable for encrypting large amounts of data. However, the challenge lies in securely sharing the key between the sender and receiver. This paper proposes a new methodology to overcome this difficulty using graph theory.The proposed method effectively demonstrates the integration of graph theory and cryptographic techniques, leveraging Hamiltonian graphs and matrix transformations to achieve secure encryption and decryption. By introducing a unique approach to convert graphs into matrices and incorporating a secret key, the methodology ensures robust security. The inclusion of a practical example illustrates the feasibility and efficiency of the proposed process. Future work could explore enhancing scalability, optimizing computational efficiency, and applying the technique to diverse data formats, paving the way for broader applications in secure communication systems.

References:

1] Wael Mahmoud Al Etaiwi. Encrypyion algorithm using graph theory. Journal of Scientific Research & Reports,3(19):2519-2527,2014.

[2] P. Amudha, A.C.Charles Sagayaraj , and A.C. Shantha Sheela. An application of graph theory in cryptography. International journal of Pure and Applied Mathematics, 119(13):375-383.

[3] Safaa Hrais and Wael Etaiwi. Symmetric encryption algorithm using graph representation. In 2017 8th International Conference on Information Technology (ICIT), pages 501-506.IEEE, 2017

[4] Manisha Kumari and V.B. Kirubanad. Data encryption and decryption using graph plotting. International Journal of Civil Engineering and Technology (IJCIET) Volume, 9:36-46, 2018.

[5] Christof Paar and Jan Pelzl. Understandin cryptography: a textbook for students and practitioners. Springer Science & Business Media,2009.

[6] Keijo Ruohonen, Graph Theory(2013).

[7] C. Vasudev. Graph theory with applications. New Age Internatinal, 2006.

[8] P.L.K. Priyadarsini, A Survey on some Applications of Graph Theory in Cryptography, Journal of Discrete Mathematical Sciences & Cryptography Vol. 18 (2015), No. 3, pp. 209–217.

[9] Srilekha Chowdhury , Promita Ghosh , Mayurakshi Jana, An Approach of Graph Theory for Solving Cryptographic Problem, BKGC SCHOLARS July -December 2020, Vol. 1 Issue. 2, PP. 64 – 68.

[10] P. Amudha, A.C. Charles Sagayaraj, A.C.Shantha Sheela, An Application of Graph Theory in Cryptography, International Journal of Pure and Applied Mathematics Volume 119 No. 13 2018, 375-383.