Encryption and Decryption Algorithms in Symmetric Key Cryptography Using Graph Theory

P.A.S.D.Perera
Department of Mathematics, University of Kelaniya, Kelaniya, Sri Lanka. dhanalishalini@gmail.com

G.S.Wijesiri
Department of Mathematics, University of Kelaniya, Kelaniya, Sri Lanka. sujeew@kln.ac.lk

ABSTRACT
The present-day society depends heavily on digital technology where it is used in many applications such as banking and e-commerce transactions, computer passwords, etc. Therefore, it is important to protect information when storing and sharing them. Cryptography is the study of secret writing which applies complex math rules to convert the original message into an incomprehensible form. Graph theory is applied in the field of cryptography as graphs can be simply converted into matrices There are two approaches of cryptography; symmetric cryptography and asymmetric cryptography. This paper proposes a new connection between graph theory and symmetric cryptography to protect the information from the unauthorized parties. This proposed methodology uses a matrix as the secret key which adds more security to the cryptosystem. It converts the plaintext into several graphs and represents these graphs in their matrix form. Also, this generates several ciphertexts. The size of the resulting ciphertexts are larger than the plaintext size.

Keywords:
Decryption, Encryption, Graph Theory, Symmetric Key Cryptography
Article Received: 18 October 2020, Revised: 3 November 2020, Accepted: 24 December 2020

I. Introduction

The field of cryptology is divided into two parts namely, cryptography and cryptanalysis [5]. Cryptography is the art of converting the original message into an incomprehensible form, with the intention of hiding the meaning, so that only the parties who are intended can read and process the information. Cryptanalysis is the science of breaking cryptosystems.

There are several components in a cryptosystem. Sender and receiver are the one who sends the message and the one who receives the message respectively. Attacker is the one who tries to get the message in an unauthorized way. The original message sent by the sender is known as plaintext and the secret message or the encrypted message is known as ciphertext. The process of converting the plaintext into ciphertext is called encryption while the reverse process is called decryption.

Symmetric key and asymmetric key cryptography are two main branches of cryptography. In asymmetric key cryptography, the sender encrypts the message using a key known as public key and sends the encrypted message to the receiver. The receiver then decrypts the ciphertext by using another key called private key. Yet in symmetric key cryptography, both the sender and the receiver use the same key to encrypt as well as to decrypt the message.

In [2], a new way of applying graph theory in cryptography is discussed. The original message is encrypted into a Euler Graph. This graph is then converted into several matrices and sent to the receiver. In [4], a modified version of Affine cipher is proposed. In this algorithm, each character of the plaintext is converted into a numeric value and these numeric values are plotted into a graph. The graph is then sent to the receiver. In [1,3], a new encryption algorithm is proposed using graph theory where the plaintext is divided into several blocks. These blocks are then converted into graphs and their graph representations are sent to the receiver. This paper proposes a new algorithm to encrypt and decrypt the message using graph theory in symmetric key cryptography.

The rest of the paper is organized as follows. Some mathematical preliminaries that are used in the proposed algorithm are introduced in section II. The proposed methodology is discussed in section III. Finally, in section IV, conclusions are stated.

II. Theoretical Preliminaries

2.1 Graph –

A graph G consists of a set of objects $V=\{v_1, v_2, v_3, \ldots\}$ called vertices and another set $E=\{e_1, e_2, e_3, \ldots\}$ called edges. Usually, a graph is denoted as G= (V, E)[7].

2.2 Complete Graph –

A graph G is said to be complete if every vertex in G is connected with every other vertex. A complete graph with n vertices is denoted by K_n. Then K_n has n(n-1)/2 edges[6].

2.3 Weighted Graph –

A weighted graph is a graph in which each edge is given a numerical weight[6].

2.4 Matrix Representation –

Adjacency Matrix –

The adjacency matrix of the graph G = (V, E) is a $n \times n$ matrix $D = (d_{ij})$, where n is the number of vertices in G, $V=\{v_1,v_2,v_3,\ldots\}$, $E = \{e_1, e_2, e_3,\ldots\}$. d_{ij} is 1 if there is an edge between v_i and v_j and d_{ij} is 0 if there is no edge between v_i and v_j[7].

III. 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**

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 $0 \le a, b \le m$ where *m=26*.
3) Convert each character of the plaintext into 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 ASCII 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 ciphertext. *(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) Convert all the characters (ignore the special characters) into numeric values (s), using the encoding table.
7) Let $p = s+(m \times q)$ where $q = \begin{cases} 1; if\ s \geq 14 \\ 2; if\ s < 14 \end{cases}$
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 plaintext is “DECEMBER FIRST “.
Let *a=17*, *b=23* and

$$K = \begin{bmatrix} 1 & 1 & 2 & 5 & 3 \\ 4 & 8 & 9 & 12 & 6 \\ 5 & 18 & 21 & 10 & 3 \\ 63 & 38 & 7 & 29 & 24 \\ 0 & 25 & 75 & 4 & 33 \end{bmatrix}_{5\times5}$$

$det|K| = -1.4781 \times 10^5 \neq 0$. Hence K^{-1} exists.

From the encoding table, $m = 26, r = 25 - 0 = 25$

From K, $n = 5$

Table -2 Encryption mechanism

Plaintext	**D**	**E**	**C**	**E**	**M**	**B**	**E**	**R**	**F**	**I**	**R**	**S**	**T**
x	**3**	**4**	**2**	**4**	**12**	**1**	**4**	**17**	**5**	**8**	**17**	**18**	**19**
(17x+23) mod 26	**22**	**13**	**5**	**13**	**19**	**14**	**13**	**0**	**4**	**3**	**0**	**17**	**8**
y	**W**	**N**	**F**	**N**	**T**	**O**	**N**	**A**	**E**	**D**	**A**	**R**	**I**
z	**87**	**78**	**70**	**78**	**84**	**79**	**78**	**65**	**69**	**68**	**65**	**82**	**73**
(z-25) mod 26	**10**	**1**	**19**	**1**	**7**	**2**	**1**	**14**	**18**	**17**	**14**	**5**	**22**
d	**K**	**B**	**T**	**B**	**H**	**C**	**B**	**O**	**S**	**R**	**O**	**F**	**W**

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 Figure 1 below.

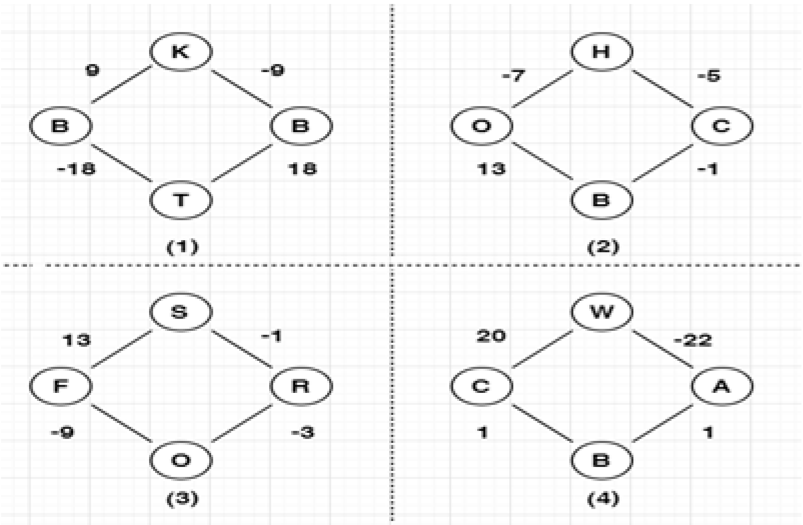

Figure 1: Weighted graphs

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

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

$$\begin{aligned} \textit{Weight}_{KB} &= (\textit{index of } B) - (\textit{index of } K) \\ &= 1 - 10 \\ &= (-9) \end{aligned}$$

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

Next, identify the special character for each block.

$$\begin{aligned} \textit{Special character of the initial block} &\equiv char((sum\ K) mod\ m) \\ &\equiv char(406\ mod\ 26) \\ &\equiv char(16) \\ &\equiv Q \end{aligned}$$

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.

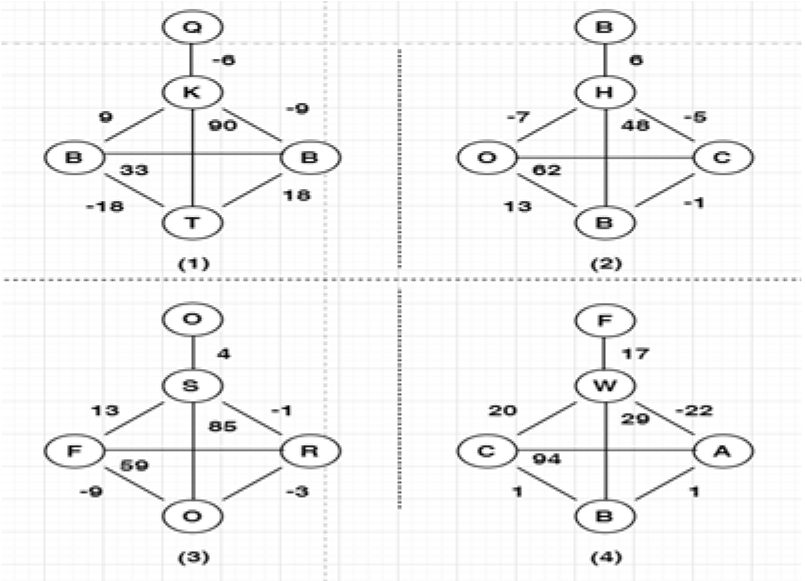

Figure 2: Final graph

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

$$C_i = M_i * K$$

$$C_1 = \begin{bmatrix} 0 & -6 & 0 & 0 & 0 \\ -6 & 0 & -9 & 90 & 9 \\ 0 & -9 & 0 & 18 & 33 \\ 0 & 90 & 18 & 0 & -18 \\ 0 & 9 & 33 & -18 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 2 & 5 & 3 \\ 4 & 8 & 9 & 12 & 6 \\ 5 & 18 & 21 & 10 & 3 \\ 63 & 38 & 7 & 29 & 24 \\ 0 & 25 & 75 & 4 & 33 \end{bmatrix} = \begin{bmatrix} -24 & -48 & -54 & -72 & -36 \\ 5619 & 3477 & 1104 & 2526 & 2412 \\ 1098 & 1437 & 2520 & 546 & 1467 \\ 450 & 594 & -162 & 1188 & 0 \\ -933 & -18 & 648 & -84 & -279 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 6 & 0 & 0 & 0 \\ 6 & 0 & -5 & 48 & -7 \\ 0 & -5 & 0 & -1 & 62 \\ 0 & 48 & -1 & 0 & 13 \\ 0 & -7 & 62 & 13 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 2 & 5 & 3 \\ 4 & 8 & 9 & 12 & 6 \\ 5 & 18 & 21 & 10 & 3 \\ 63 & 38 & 7 & 29 & 24 \\ 0 & 25 & 75 & 4 & 33 \end{bmatrix} = \begin{bmatrix} 24 & 48 & 54 & 72 & 36 \\ 3005 & 1565 & -282 & 1344 & 924 \\ -83 & 1472 & 4598 & 159 & 1992 \\ 187 & 691 & 1386 & 618 & 714 \\ 1101 & 1554 & 1330 & 913 & 456 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & -1 & 85 & 13 \\ 0 & -1 & 0 & -3 & 59 \\ 0 & 85 & -3 & 0 & -9 \\ 0 & 13 & 59 & -9 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 2 & 5 & 3 \\ 4 & 8 & 9 & 12 & 6 \\ 5 & 18 & 21 & 10 & 3 \\ 63 & 38 & 7 & 29 & 24 \\ 0 & 25 & 75 & 4 & 33 \end{bmatrix} = \begin{bmatrix} 16 & 32 & 36 & 48 & 24 \\ 5354 & 3541 & 1557 & 2527 & 2478 \\ -193 & 1353 & 4395 & 137 & 1869 \\ 325 & 401 & 27 & 954 & 204 \\ -220 & 824 & 1293 & 485 & 39 \end{bmatrix}$$

$$C_4 = \begin{bmatrix} 0 & 17 & 0 & 0 & 0 \\ 17 & 0 & -22 & 29 & 20 \\ 0 & -22 & 0 & 1 & 94 \\ 0 & 29 & 1 & 0 & 1 \\ 0 & 20 & 94 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 2 & 5 & 3 \\ 4 & 8 & 9 & 12 & 6 \\ 5 & 18 & 21 & 10 & 3 \\ 63 & 38 & 7 & 29 & 24 \\ 0 & 25 & 75 & 4 & 33 \end{bmatrix} = \begin{bmatrix} 68 & 136 & 153 & 204 & 102 \\ 1734 & 1223 & 1275 & 786 & 1341 \\ -25 & 2212 & 6859 & 141 & 2994 \\ 121 & 275 & 357 & 362 & 210 \\ 613 & 1890 & 2161 & 1209 & 426 \end{bmatrix}$$

To decrypt, multiply the received ciphertexts $(\mathrm{C_i})$ with K^{-1}

$$M_i = C_i * K^{-1}$$

$$K^{-1} = \begin{bmatrix} 1.7235 & -1.0643 & 0.3362 & 0.0294 & -0.0151 \\ -2.6007 & 1.4987 & -0.4535 & -0.0179 & 0.0182 \\ 1.6807 & -1.0095 & 0.3451 & 0.0100 & -0.0079 \\ 0.8767 & -0.4081 & 0.1616 & -0.0008 & -0.0196 \\ -1.9559 & 1.2084 & -0.4604 & -0.0091 & 0.0369 \end{bmatrix}$$

$$M_1 = \begin{bmatrix} 0 & -6 & 0 & 0 & 0 \\ -6 & 0 & -9 & 90 & 9 \\ 0 & -9 & 0 & 18 & 33 \\ 0 & 90 & 18 & 0 & -18 \\ 0 & 9 & 33 & -18 & 0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 0 & 6 & 0 & 0 & 0 \\ 6 & 0 & -5 & 48 & -7 \\ 0 & -5 & 0 & -1 & 62 \\ 0 & 48 & -1 & 0 & 13 \\ 0 & -7 & 62 & 13 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & -1 & 85 & 13 \\ 0 & -1 & 0 & -3 & 59 \\ 0 & 85 & -3 & 0 & -9 \\ 0 & 13 & 59 & -9 & 0 \end{bmatrix} \qquad M_4 = \begin{bmatrix} 0 & 17 & 0 & 0 & 0 \\ 17 & 0 & -22 & 29 & 20 \\ 0 & -22 & 0 & 1 & 94 \\ 0 & 29 & 1 & 0 & 1 \\ 0 & 20 & 94 & 1 & 0 \end{bmatrix}$$

Now draw the graphs whose adjacency matrices are M_is. See Figure 3 below.

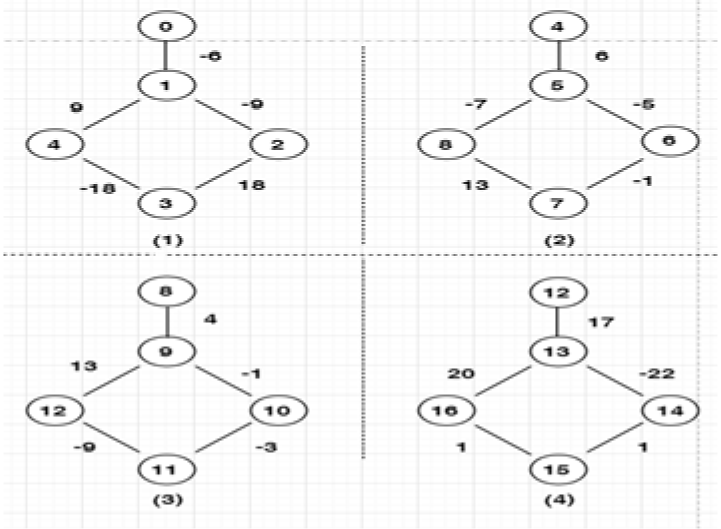

Figure 3:Final graph with extra edges

Ignore the edges having weights greater than m. See figure 4.

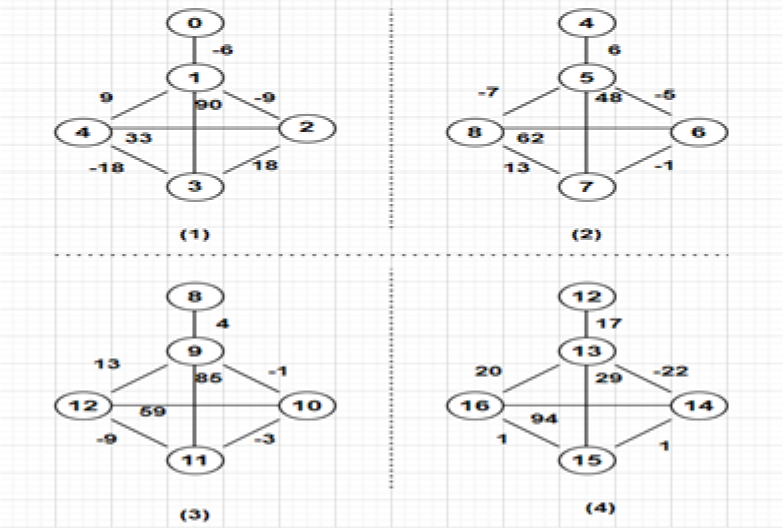

Figure 4:Simplified graph

We Know,

$$\begin{aligned} \textit{Special character of the initial block} &\equiv char\big((sum\ K) mod\ m\big) \\ &\equiv char(406\ mod\ 26) \\ &\equiv char(16) \\ &\equiv Q \end{aligned}$$

i.e.$0^{th}\ vertex \equiv Q$

Next, convert all the other vertices into letters using the encoding table.

For example, $1^{st}\ vertex$ can be converted into a letter as follows:

$1^{st}\ vertex - 0^{th}\ vertex \equiv (-6)$

$1^{st}\ vertex - 16 \equiv (-6); because\ from\ the\ encoding\ table\ Q \equiv 16$

1st vertex≡-6+16=10

$1^{st}\ vertex \equiv K; from\ the\ encoding\ table$

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

Table -3 Decryption Mechanism

Ciphertext	**K**	**B**	**T**	**B**	**H**	**C**	**B**	**O**	**S**	**R**	**O**	**F**	**W**	**A**	**B**	**C**
s	**10**	**1**	**19**	**1**	**7**	**2**	**1**	**14**	**18**	**17**	**14**	**5**	**22**	**0**	**1**	**2**
p=s+(26×q)	**62**	**53**	**45**	**53**	**59**	**54**	**53**	**40**	**44**	**43**	**40**	**57**	**48**	**52**	**53**	**54**
p+25	**87**	**78**	**70**	**78**	**84**	**79**	**78**	**65**	**69**	**68**	**65**	**82**	**73**	**77**	**78**	**79**
y	**W**	**N**	**F**	**N**	**T**	**O**	**N**	**A**	**E**	**D**	**A**	**R**	**I**	**M**	**N**	**O**
x	**22**	**13**	**5**	**13**	**19**	**14**	**13**	**0**	**4**	**3**	**0**	**17**	**8**	**12**	**13**	**14**
17^{-1}(x-23) mod 26	**3**	**4**	**2**	**4**	**12**	**1**	**3**	**17**	**5**	**8**	**17**	**18**	**19**	**7**	**4**	**1**
plaintext	**D**	**E**	**C**	**E**	**M**	**B**	**E**	**R**	**F**	**I**	**R**	**S**	**T**	**A**	**B**	**C**

The calculations 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,

$plaintext = DECEMBER\ FIRST$

3.4 Discussion–

The present society depends on digital tools in almost every activity in their day-to-day life. Therefore, protecting the information which are being shared is the most pivotal task today.

There are many algorithms that have been developed to safeguard the data from unauthorized parties. An algorithm is claimed to provide a strong protection if the ciphertext is held unrevealed even if the attacker has all the information of the algorithm. Therefore, the security of an algorithm depends on several factors, such as how hard it is to guess the secret key and how hard to guess the plaintext even though the whole ciphertext is obtained by the attacker.

The proposed methodology generates a ciphertext where its size is larger than the plaintext size. Also, this algorithm uses a matrix of order $(n\times n)$ as the secret key, which makes it hard to guess the secret key. The ciphertexts are obtained by applying matrix multiplication. This adds more security to the plaintext. Further, this methodology generates several matrices as the ciphertext which makes the algorithm strong against cryptanalysis as the probability of receiving the whole ciphertext is low.

IV.CONCLUSION

The symmetric key cryptography is used to encrypt large amounts of data as it is faster than asymmetric cryptography. In this approach, sharing the shared key via a secured channel is the most important thing. This paper proposes a new methodology to overcome this difficulty using graph theory. The proposed methodology first converts the original message into several graphs. And it further transforms these graphs into matrices. Finally, obtains the ciphertext by multiplying these matrices with the secret key. This paper explains the proposed encryption and decryption processes further by providing an example as well.

REFERENCES

[1] Wael Mahmoud Al Etaiwi. Encryption 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, 2018.

[3] Safaa Hraiz 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. *Understanding 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 International, 2006.