University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Text Encryption with Graph Theory Based Key Generation

Karrar Khudhair Obayes

Department of Computer Information Systems, University of Al-Qadisiyah, Iraq ,Al Diwaniyah, Email: : karrar.khudhair@qu.edu.iq

ARTICLE INFO

Article history:
Received: 11 /07/2024
Rrevised form: 29 /07/2024
Accepted : 5 /08/2024
Available online: 30 /09/2024

Keywords:

encryption, decoding

undirected graph

complete graph

adjacent matrix

Each keyword to start on a new line

ABSTRACT

This work presents a new encryption technique that ensures safe communication through two distinct phases: utilizing monoalphabetic substitution ciphers that rely on agreed-upon character-swapping arrangements between authorized individuals and applying an alphabetical encryption table. In addition, the research utilizes the Kurskal technique to calculate the minimum spanning tree, which improves security by creating intricate encrypted text using a shared key based on ideas drawn from graph theory.

MSC..

https://doi.org/10.29304/jqcsm.2024.16.31650

1. Introduction

Encryption plays a crucial role in protecting sensitive data when it is being sent or stored. It achieves this by using complex mathematical techniques to transform original communications into formats that cannot be understood. The origin of cryptography can be traced back more than two millennia, yet it was not until 1949 that Claude Shannon established the fundamental principles of modern cryptography [1]. As digital communications have progressed, encryption methods have developed to encompass solutions for ensuring confidentiality, privacy, and authentication. These solutions include passwords, digital signatures, digital identities, as well as electronic

∗Corresponding author

Email addresses:

Communicated by ‘sub etitor’

currencies. Encryption has become a crucial component of our everyday existence. Encryption is the process of transforming the original text into encrypted text, while decryption refers to the reverse process [2], [3], [4], [5], [6] . Within every encryption system, a designated key is employed to transform plaintext into encrypted text, which may subsequently be reversed into legible text using the decryption key. The key is an essential component for the process of encrypting and decrypting data, enabling authorized recipients to effortlessly decipher the encrypted communication. Graph structures were employed to bolster the encryption system, so enhancing the security of data transfer. Comprehending graph theory is essential for deciphering any encrypted text [7], [8], [9], [10]. A random graph is used to encode images by making the vertices public while keeping the edges secret. Afterward, the image contrast is adjusted [11]. A symmetric encryption algorithm transforms the original text into a series of graphs using a secret key, resulting in an increase in the size of the encrypted text compared to the original text [12], [13]. This study introduces a novel encryption method that operates securely across two phases: employing monoalphabetic substitution ciphers based on mutual character-swapping agreements between authorized parties and utilizing an alphabetical encryption table. Furthermore, the research employs the Kurskal algorithm to determine the minimal spanning tree, Additionally, the proposed approach incorporates a new technique for encoding repeated characters in the cipher text utilizing graph properties. The remaining parts of this paper are organized as follows: Section 2 provides key concepts of graphs along with theoretical steps for encoding and decoding signal text. Our proposed method is evaluated, and the results are presented in Section 3. Detailed discussions on the advantages and challenges of the proposed method are also included in this section. Finally, concluding remarks are made in Section 4.

2. Basic concepts

In this section, we discuss some important concepts of graph theory. We also discuss the definition of encryption and decryption. These concepts form the basis of the proposed encryption method.

Definition 2.1: A graph $G = (V, E)$ is an ordered pair of disjoint sets (V, E), where $V \neq \emptyset$ and E are a subset of unordered pairs of V. The elements $V = V\ (G)$, and $E = \ E\ (G)$ respectively vertices and edges of a graph (G).

Definition 2.2: cycle graph: A closed path starts and ends at the same node, and passes through each node only once without repetition, where the degree of each vertex equals 2.

Definition 2.3: A complete graph is a graph in which every vertex is adjacent to all vertices and denoted by K_n [14].

Definition 2.4: Suppose G is a graph and $V(G) = \{v_1, v_2, \dots, v_n\}$ is the set of vertices of G. In this case, the adjacency matrix of the graph G, which we display with the symbol $A(G)$ [14]. Is a $n \times n$ square matrix defined as $A(G) = [a_{ij}]$ is defined as :

$$a_{ij} = \begin{cases} 1 & if \quad \mathrm{u, v\ E\ (G)} \\ 0 & if \quad \text{otherwise} \end{cases}$$

Definition 2.5: Weighted Graph In the graph, each edge is characterized by a specific number, so the weighted graph is a numerical name of the graph and is symbolized by the symbol W_{ij} [14].

Encryption and decryption

To prevent unauthorized access or to protect the confidentiality of digital information, mathematical operations are employed to convert the original message into cipher text. This transformation renders the message unreadable to anyone except authorized individuals who possess the original data with the cipher text. Conversely, decryption involves reverting the cipher text to its original plain text form, which requires a password or private key accessible only to those authorized to decrypt the information.

Proposed encryption method

Our study introduces a method that utilizes a two-step encryption procedure, resulting in improved security and efficiency compared to previous research. At first, the letters in the original text are substituted with different random letters according to a mutually agreed upon substitution table. The encrypted letters are subsequently depicted as neighboring vertices in a network. This is achieved through the process of assigning consecutive weights to the vertices and incorporating edges until a graph without cycles is created. In this graph, all the letters of the alphabet are included in an encrypted table, with each letter being allocated a distinct integer. Afterwards, every vertex is linked to every other vertex, creating a graph that is complete. The recently inserted edges are allocated consecutive weights, beginning with the final index in the encrypted table. Subsequently, the adjacency matrix of the whole graph is computed, and the shortest path is ascertained utilizing the Kruskal algorithm. The multiplication of the two matrices is executed, and the resultant matrix is subsequently multiplied with the shared key matrix. The result is the ultimate matrix, which represents the encrypted data that is transmitted to the receiver.

This process ensures robust encryption by leveraging a comprehensive graph-based approach and matrix operations, significantly enhancing the method proposed security and efficiency Method A and B show our proposed method

Approach $\boldsymbol{A}$: Encryption step

Replace the original text with encrypted text using a randomized character substitution table agreed upon by both parties.

- Represent the text as a graph where each letter corresponds to a vertex.
- Create edges between every two consecutive letters in the original text to form a cyclic graph.
- Assign weights to these edges using a predetermined alphabetical encoding table.
- Transform the graph into a complete graph by connecting each vertex to all others. The weights of these new edges are determined as 1 plus the maximum weight from the encryption table.
- Introduce a designated starting vertex (A) crucial for determining the letter sequence during decoding. Construct the adjacency matrix (AK_1) of this complete graph.

- Utilize the Kurskal algorithm to find the shortest path with the minimum weight, subsequently obtaining the complete graph adjacency matrix (AK_2).
- Multiply the resultant matrices AK_1 and AK_2 to produce AK_3.
- Multiply 3M by the shared key K to generate the ciphertext.
- Format the cipher text as a linear combination of the C Matrix and AK_1 Matrix.

The following methods provide a systematic and secure way for encrypting text utilizing graph-based techniques and matrix operations, while also ensuring the feasibility of decoding.

Approach ***B***: Decryption step

- The recipient computes AK_3 matrix using the inverse of the shared key K.
- By employing the inverse of AK_1, AK_2 matrices can be determined.
- The cipher text characters are obtained using the alphabetical encryption table.
- The original text is retrieved using a mutually agreed random character substitution table.

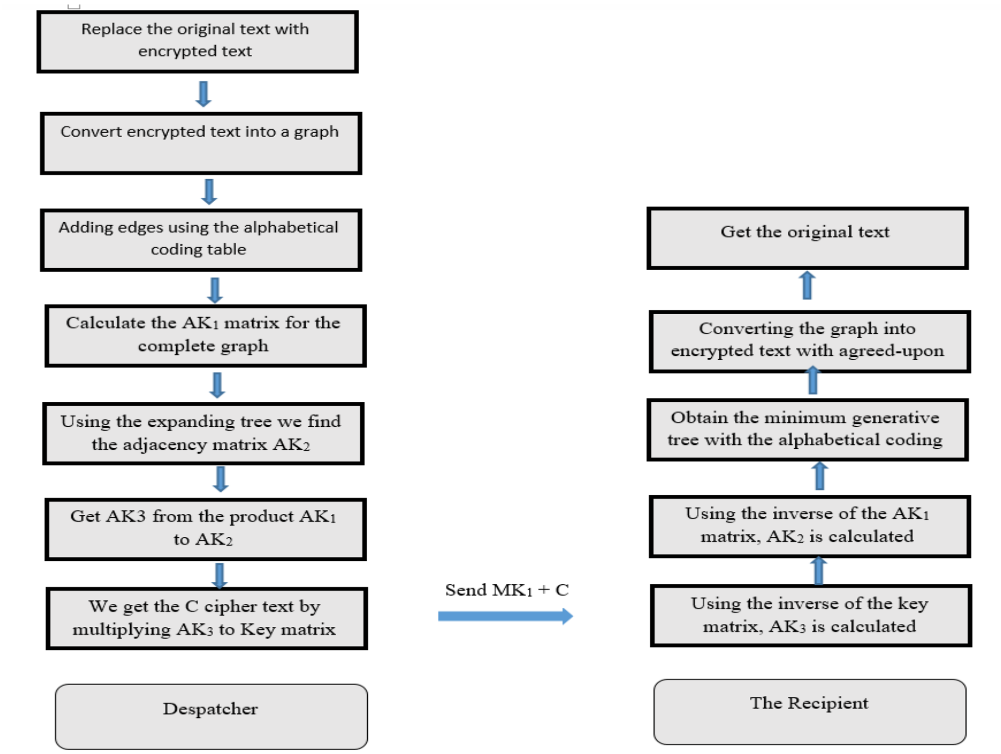

Replace the original text with encrypted text
Convert encrypted text into a graph
Adding edges using the alphabetical coding table
Calculate the AK₁ matrix for the complete graph
Using the expanding tree we find the adjacency matrix AK₂
Get AK3 from the product AK₁ to AK₂
We get the C cipher text by multiplying AK₃ to Key matrix
Despatcher
Send MK₁ + C
Get the original text
Converting the graph into encrypted text with agreed-upon
Obtain the minimum generative tree with the alphabetical coding
Using the inverse of the AK₁ matrix, AK₂ is calculated
Using the inverse of the key matrix, AK₃ is calculated
The Recipient

Fig. 1- An illustrative diagram of the text encryption and decryption process

3. Analyzing the Results

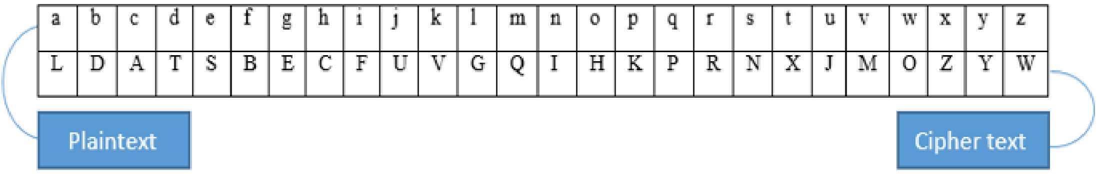
This section presents the encryption and decryption methods and the results obtained from the proposed algorithm. We explain the process of converting text into a graph, applying weights, and transforming it into a complete graph. We then discuss the methodology for finding the shortest path using the Kurskal algorithm. Finally, we illustrate how the matrices are manipulated and combined with the shared key to produce the cipher text, analyzing the approach's effectiveness and security using two encryption tables.

Assuming we have an encrypted message, "BOOST" sent to the recipient, it is important to note that in this study, a repeated character was selected in the cipher text. This text will be represented and processed using graph properties. The steps involved in this process are as follows:

The first step Replace the original text with encrypted text using a randomized character substitution

table agreed upon by both parties. As shown in (Fig.2).

Table 1- random coding table



a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
L	D	A	T	S	B	E	C	F	U	V	G	Q	I	H	K	P	R	N	X	J	M	O	Z	Y	W

Plaintext

Cipher text

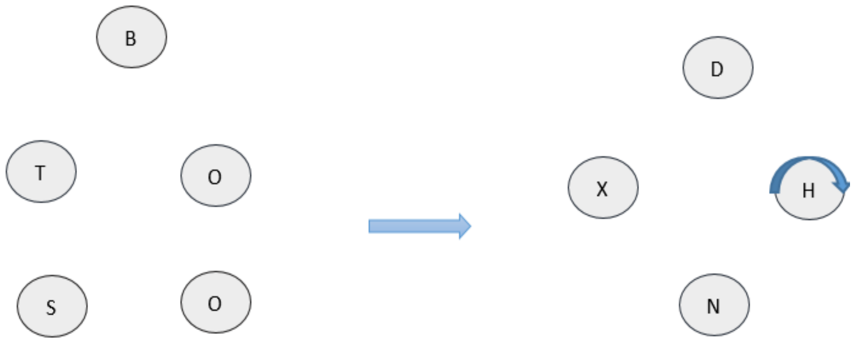


Fig. 2- Converting letters into vertices

The letters were converted into vertices and were randomly replaced according to the encoding table in (Fig. 2). Notably, each recurring letter in the original text was connected in the graph to form a loop. This approach offers a novel solution for handling repeated characters in the text to be encrypted.

Every two sequential letters are connected to form a cyclic graph, with each edge assigned a weight according to the following encryption table:

Table 2- Alphabetical Encoding

A	B	C	D	…	H	I	…	M	N	…	R	S	T	…	X	Y	Z
1	2	3	4		8	9		13	14		18	19	20		24	25	26

Therefore, the Distance between two vertices = $code(A) - code(B)$ where $A > B$

Distance between C and R = code (H) – code (D) = 8-4 = 4.

Similarly, the weights of the other vertices are calculated, resulting in the following weighted graph.

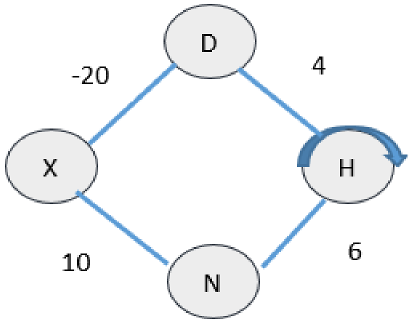


Fig. 3- A weighted histogram of cipher text characters

We continue adding edges to form a complete graph, with each new edge assigned a weight starting from the maximum weight in the encryption table. (27+1=28). As shown in the(fig.4).

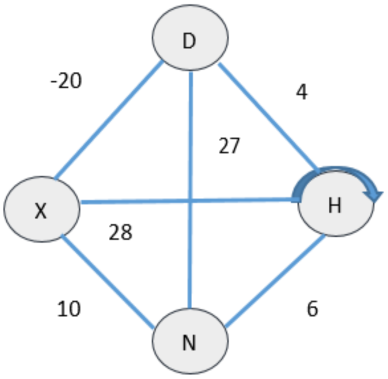


Fig. 4- Complete weighted graph of cipher text

Introduce a new character as a start indicator, placed before the first character in the cipher text. For instance, let us assume that the character (A) serves as the start indicator. As shown in the (fig. 5).

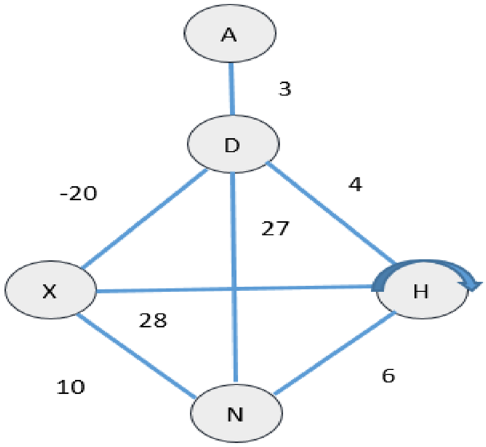


Fig. 5- Complete graph with the addition of one vertex of the original encrypted text

We determine the adjacency matrix (AK_1) of the complete graph shown in Fig. (5).

$$AK_1 = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 \\ 3 & 0 & 4 & 27 & -20 \\ 0 & 4 & 2 & 6 & 28 \\ 0 & 27 & 6 & 0 & 10 \\ 0 & -20 & 28 & 10 & 0 \end{bmatrix}$$

We find the minimum of the generative tree using the Kurskal algorithm

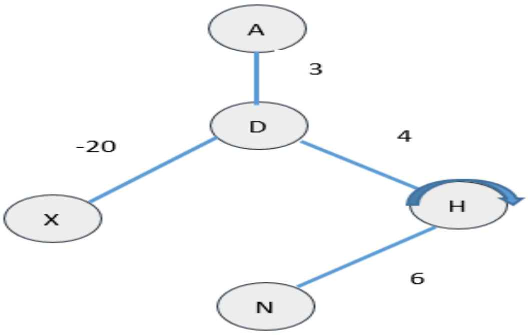


Fig. 6- Minimum generative tree

We determine the adjacency matrix (AK_2) in Fig. (6)

$$AK_2 = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 \\ 3 & 0 & 4 & 0 & -20 \\ 0 & 4 & 2 & 6 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & -20 & 0 & 0 & 0 \end{bmatrix}$$

We assume that vertex 1 corresponds to the letter A, which is added to the first vertex in the diagonal of the matrix. After this modification, the matrix is updated as follows:

$$AK_2 = \begin{bmatrix} 1 & 3 & 0 & 0 & 0 \\ 3 & 0 & 4 & 0 & -20 \\ 0 & 4 & 2 & 6 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & -20 & 0 & 0 & 0 \end{bmatrix}$$

Next, we perform the multiplication between matrices AK_1 and AK_2. This step is crucial for ensuring the proper alignment and integration of the data within the cryptographic framework.

$$AK_3 = AK_1.AK_2 = \begin{bmatrix} 9 & 0 & 12 & 0 & -60 \\ 3 & 425 & 170 & 24 & 0 \\ 12 & -552 & 56 & 12 & -80 \\ 81 & -176 & 120 & 36 & -540 \\ -60 & 112 & 36 & 168 & 400 \end{bmatrix}$$

To encrypt the AK_3 array, we utilize the shared key K. This encryption step is essential for securing the data and ensuring that only authorized parties with the correct key can access the information

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

To get the Cipher Matrix C by multiplying Key Matrix K with Matrix AK_3.

$$C = \begin{bmatrix} 9 & 0 & 12 & 0 & -60 \\ 3 & 425 & 170 & 24 & 0 \\ 12 & -552 & 56 & 12 & -80 \\ 81 & -176 & 120 & 36 & -540 \\ -60 & 112 & 36 & 168 & 400 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -39 & -48 & -48 & -60 & -60 \\ 622 & 619 & 194 & 24 & 0 \\ -552 & -564 & -12 & -68 & -80 \\ -479 & -560 & -384 & -504 & -540 \\ 656 & 716 & 604 & 568 & 400 \end{bmatrix}$$

At this point, the message is encrypted. We transmit the Cipher Matrix, Key Matrix K, and AK_1 Matrix to the recipient.

On the recipient's side, Matrix AK_3 is obtained by multiplying the received cipher text with the inverse of the shared keyK^{-1}.

$$AK_3 = CK^{-1} = \begin{bmatrix} -39 & -48 & -48 & -60 & -60 \\ 622 & 619 & 194 & 24 & 0 \\ -552 & -564 & -12 & -68 & -80 \\ -479 & -560 & -384 & -504 & -540 \\ 656 & 716 & 604 & 568 & 400 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 9 & 0 & 12 & 0 & -60 \\ 3 & 425 & 170 & 24 & 0 \\ 12 & -552 & 56 & 12 & -80 \\ 81 & -176 & 120 & 36 & -540 \\ -60 & 112 & 36 & 168 & 400 \end{bmatrix}$$

Subsequently, AK_2 is derived by multiplying AK_3 with the inverse of AK_1, denoted as AK_1^{-1}

$$AK_2 = AK_1^{-1}AK_3 = \begin{bmatrix} 1 & 3 & 0 & 0 & 0 \\ 3 & 0 & 4 & 0 & -20 \\ 0 & 4 & 2 & 6 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & -20 & 0 & 0 & 0 \end{bmatrix}$$

Consequently,(Fig.7) represents the final graph, which is utilized to reconstruct the Encrypted message

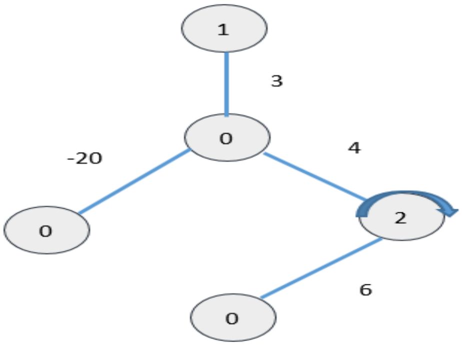


Fig. 7- Encrypted graph

We initially assumed that the number (1) corresponds to the letter (A) based on an alphabetical coding table. Thus, the second vertex represents vertex (1), and adding the weight of the second vertex gives us 1 + 3 = 4. According to the table, the letter (D) corresponds to this sum. Further, the second vertex, represented by the letter (D) and equivalent to the number 4, when added to another 4, results in 8, which corresponds to the letter (H) in the alphabetical coding table. According to the graph's properties, the vertex (H) connects to itself, indicating that this letter is repeated twice. We continue this process iteratively until we obtain the complete encrypted text (DHHNX). The final stage is to convert the encrypted text back to the original text by representing each character of the encrypted text within the random encryption table. Refer to Table (1). By doing so, we will have obtained the original text. (DHHNX) to (BOOST).

4. Conclusion and feature work

This paper introduces a novel encryption approach that employs two encryption tables. The original text is converted into cipher text using a predetermined random encryption table shared between the sender and the recipient. Subsequently, the cipher text is transformed into a graph using an alphabetical encryption table. Additionally, the proposed approach incorporates a new technique for encoding repeated characters in the cipher text utilizing graph properties.

Many future improvements can be made to reduce the size of the cipher text, such as dividing the array into smaller arrays that can be combined during decryption. Additionally, the proposed working method can be translated into algorithms in various programming languages, such as Python and C++, to reduce encryption time and enhance the overall performance of the algorithm.

Reference

[1] C. E. Shannon, "Communication theory of secrecy systems," The Bell system technical journal, vol. 28, pp. 656--715, 1949.

[2] S. G. Akl, "How to encrypt a graph," International Journal of Parallel, Emergent and Distributed Systems, vol. 35, pp. 668--681, 2020.

[3] Bai, Sen and Zhou, Longfu and Yan, Mingzhu and Ji, Xiaoyong and Tao, Xuejiao, "Image cryptosystem for visually meaningful encryption based on fractal graph generating," IETE Technical Review, vol. 38, pp. 130-141, 2021.

[4] S. A. a. A.-W. R. D. a. A. E. W. Abdul-Ghani, "Securing text messages using graph theory and steganography," Baghdad Science Journal, vol. 19, pp. 0189--0189, 2022.

[5] F. a. R. M. K. a. L. Q. a. W. S. Monrose, "Cryptographic key generation from voice," Proceedings 2001 IEEE Symposium on Security and Privacy. S\&P 2001, pp. 202--213, 2000.

[6] M. a. G. M. a. S. A. a. K. M. J. H. Yamuna, "Encryption using graph theory and linear algebra," International Journal of Computer Application, vol. 5, pp. 102--107, 2012.

[7] M. a. K. K. Yamuna, "Data transfer using bipartite graphs," International Journal of Advance Research in Science and Engineering, vol. 4, pp. 128--131, 2015.

[8] A. S. Naji, "key generation for text encryption using graph theory," Journal of Baghdad College of Economic sciences University, vol. 18, 2023.

[9] M. a. O. K. K. a. A. M. Alaeiyan, "Prediction nullity of graph using data mining," Results in Nonlinear Analysis, pp. 1-8, 2023.

[10] S. a. M. D. a. O. R. Lu, "Visual cryptography on graphs," Computing and Combinatorics: 14th Annual International Conference, COCOON 2008 Dalian, China, June 27-29, 2008 Proceedings 14, pp. 225-234, 2008.

[11] P. a. W. G. Perera, "Encryption and decryption algorithms in symmetric key cryptography using graph theory," Psychology and Education Journal, vol. 58, pp. 3420--3427, 2021.

[12] V. A. Ustimenko, "On graph-based cryptography and symbolic computations," Serdica Journal of Computing, vol. 1, pp. 131--156, 2007.

[13] C. Vasudev, Graph theory with applications, New Age International, 2006.