2456-8686, x(ii), 2026:048-059
https://doi.org/10.26524/cm239

ORIGINAL RESEARCH **Open Access**

Detailed Proposed Algorithm: Symmetric Encryption and Decryption Using Eulerian Circuits in Simple Graphs

Yashmin Banu [1], Biplab kumar rath [2] and *Debasis Gountia* [3]

Received: 20 May 2026/ Accepted: 15 June 2026 / Published online: 10July 2026

Abstract

This paper proposes a novel symmetric-key encryption scheme using simple weighted graphs and Eulerian circuits. Unlike NP-complete Hamiltonian cycle methods, it employs polynomial-time Eulerian circuits via Hierholzer's algorithm. Plaintext is mapped to a weighted graph using a secret key. An Eulerian circuit generates a dynamic key and Euler Tour Matrix. Encryption uses matrix multiplications with a shared upper triangular key and modular reduction. Decryption reverses the operations using matrix inverses. Statistical tests confirm strong diffusion and randomness. The scheme provides better efficiency and scalability than traditional Hamiltonian-based approaches.

Key Words: Simple Graph, Eulerian Circuit, Symmetric Key Cryptography, Graph-based Encryption, Euler Tour Matrix.

1 Introduction

Combinatorial mathematics and graph theory have become powerful tools in modern cryptography and secure communications. Traditional algorithms such as RSA, AES, and ECC are built on strong discrete mathematical foundations, while graph-based encryption methods have gained increasing research interest. Several studies have explored graph theory applications in cryptography [1–3], including Eulerian graphs and circuits for symmetric encryption [4] and lightweight graph-based schemes [5]. This paper proposes a novel

[1] Department of Mathematics, GIET University, Gunupur, Odisha, india. Email: yashmin.banu@giet.edu
[2] Department of Mathematics, GIET University, Gunupur, Odisha, india. Email: biplab.rath@giet.edu
[3] School of Computer Sciences, OUTR Bhubaneswar,Odisha, India. Email: dgountia@cs.iitr.ac.in

symmetric key encryption technique using simple weighted graphs and Eulerian circuits. Plaintext characters are modeled as vertices with edge weights derived from a secret key. A shared diagonal dominant matrix serves as the primary key, and a dynamic secondary key is generated through Eulerian traversal [6]. Section 2 presents the core concepts and preprocessing, while Section 3 details the encryption and decryption procedures.

A. CORE IDEAS AND PRELIMINARY FRAMEWORK

This section introduces the key principles and preliminary concepts of Network Theory and Information Security that are necessary for understanding the proposed encryption framework.

a) Graph A graph $G$is defined as an associated pair $G = (V, E)$, where $V$is a finite non-empty collection of vertices and $E$is a set of connections in which every edge links two distinct vertices without any directional order.

b) Simple Graph A graph $G = (V, E)$is called a simple graph under the condition that the graph has neither loops nor duplicate edges between identical vertex pairs, i.e., $E \subseteq \binom{V}{2}$.

c) Weighted Graph A weighted graph is a graph $G = (V, E, w)$, where $w: E \to \mathbb{R}^+$is a weighting function that allocates a positive real-valued weight to every edge.

d) Degree of a Vertex The degree of a vertex $v \in V$, denoted $\deg(v)$, is the number of edges incident to v.

e) Eulerian Circuit Let $G = (V, E)$be a connected graph. A closed trail that traverses each edge exactly one time is known as an Eulerian circuit. A graph contains an Eulerian circuit precisely when it is connected and all vertices possess even degree, i.e.,

$$\deg(v) \equiv 0 (\text{mod} 2) \qquad \forall\, v \in V.$$

f) Euler Tour Matrix Let $C = (v_0, v_1, v_2, \ldots, v_n = v_0)$be an Eulerian circuit. The Euler Tour Matrix $B = (b_{ij})$of order $n \times n$is defined as:

$$b_{ij} = \begin{cases} w(v_i, v_j) & \text{if } (v_i, v_j) \text{ is traversed in the circuit,} \\ 0 & \text{otherwise.} \end{cases}$$

g) Adjacency Matrix For a weighted graph G, the adjacency matrix $A = (a_{ij})$is an $n \times n$ matrix where

$$a_{ij} = \begin{cases} w(v_i, v_j) & \text{if } (v_i, v_j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

h) Information Security: Information Security deals with methods for protecting data and ensuring secure communication against unauthorized access or attacks. It mainly focuses on encoding and decoding mechanisms to preserve confidentiality and integrity of information.

2456-8686, x(ii), 2026:048-059
https://doi.org/10.26524/cm239

i) Encoding and Decoding Process: Let M represent the original message data. The encoding operation is defined as a function that transforms the original information into a secured form.

$Enc: M \times K \rightarrow C$that produces ciphertext Cusing key K. Decryption is the inverse function $Dec: C \times K \rightarrow M$that recovers the original plaintext.

j) Key In the proposed scheme, two keys are used:

- K: A shared upper triangular matrix of order $n \times n$.
- S: A dynamic secret key derived from the sum of edge weights traversed in the Eulerian circuit, i.e., $S = \sum w(e)$ for all edges $e$in the circuit.

II. PROPOSED METHOD

Encryption Algorithm

The proposed symmetric key encryption algorithm consists of the following steps:

STEP 1: Original Message Representation: Let the input message be represented as $M = m_1 m_2 \dots m_n$, where n denotes the size of the message. Convert each character $m_i$into its numerical value $v_i$using the Encryption Table (Table 1). Also, obtain its alphabet position p_i(where A=1, B=2, ..., Z=26) using the Alphabet Encoding Table (Table 2).

Table 1. Encryption Table

	0	1	2	3	4	5	6
7	A	B	C	D	E	F	G
8	H	I	J	K	L	M	N
9	O	P	Q	R	S	T	U
10	V	W	X	Y	Z	SPACE	

Table 2. Alphabet Encoding Table

A	B	C	D	E	F	G	...	X	Y	Z
1	2	3	4	5	6	7	...	24	25	26

STEP 2: Construction of Simple Weighted Graph Construct a simple weighted graph $G = (V, E, w)$with $n$vertices corresponding to the characters of the message. Edge weights $w(v_i, v_j)$are computed as:

$$w(v_i, v_j) = \mid v_i - v_j \mid + (p_i \times p_j \text{mod } 97)$$

- v_i: its numerical value obtained from the (Table 1),
- p_i: its alphabet position obtained from the (Table 2), where $A = 1, B = 2, \dots, Z = 26$.

Edges are selectively added using a secret seed key $K_s$to control the sparsity and structure of the graph.

STEP 3: Making the Graph Eulerian Compute the degree of each vertex. If any vertex has an odd degree, add minimum-weight auxiliary edges (or duplicate existing edges) between odd-degree vertices to make all degrees even, ensuring the graph becomes Eulerian. Record these modifications as part of the auxiliary information for decryption [7].

STEP 4: Finding Eulerian Circuit Apply **Hierholzer's Algorithm** to find an Eulerian circuit $C = \left(v_{c_0}, v_{c_1}, \dots, v_{c_n} = v_{c_0}\right)$in the Eulerian graph.

STEP 5: Construction of Matrices

- **Adjacency Matrix A** (n×n):

$$a_{ij} = \begin{cases} w(v_i, v_j) & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- **Euler Tour Matrix B**:

$$b_{ij} = \begin{cases} w(v_i, v_j) & \text{if edge } (v_i, v_j) \text{ is traversed in the circuit} \\ 0 & \text{otherwise} \end{cases}$$

STEP 6: Modification of Euler Tour Matrix Create the modified matrix $B^*$by replacing the diagonal elements with alphabet positions:

$$b_{ii}^* = p_i, i = 1,2, \dots, n$$

STEP 7: Matrix Operations Compute the intermediate matrix:

$$N = A \times B^*$$

Generate the shared upper triangular key matrix $K$of order $n \times n$, defined as:

$$k_{ij} = \begin{cases} j - i + 1 & \text{if } j \geq i, \\ 0 & \text{otherwise.} \end{cases}$$

$$\boldsymbol{K} = \begin{bmatrix} 1 & 2 & 3 & \cdots & n-2 & n-1 & n \\ 0 & 1 & 2 & \cdots & n-3 & n-2 & n-1 \\ 0 & 0 & 1 & \cdots & n-4 & n-3 & n-2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 2 & 3 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 2 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. Key Matrix

Compute the first cipher matrix:

$$C_1 = N \times K$$

STEP 8: Generation of Second Key Determine the dynamic key S by adding the weights associated with every edge included in the Eulerian cycle:

2456-8686, x(ii), 2026:048-059
https://doi.org/10.26524/cm239

$$S = \sum_{i=0}^{n-1} w(v_{c_i}, v_{c_{i+1}})$$

Where $(v_{c_i}, v_{c_{i+1}})$are the consecutive edges in the Eulerian circuit C(see Table 4).

STEP 9: Final Ciphertext Apply modular reduction:

$$C_2 = C_1 \text{mod } S$$

The final ciphertext is the flattened form of matrix C_2.

The complete proposed encryption algorithm is Outlined in Table 3.

Table 3. Summary of the Proposed Encryption Algorithm

Step	Description	Output	Mathematical Notation
1	Plaintext Encoding	v_i, p_i	Table 1 & Table 2
2	Simple Weighted Graph Construction	Graph G	$w(v_i, v_j) = \mid v_i - v_j \mid + (p_i \times p_j \text{mod } 97)$
3	Making Graph Eulerian	Eulerian Graph	All deg(v)even
4	Eulerian Circuit	Circuit C	Hierholzer's Algorithm
5	Matrices Construction	A, B	Adjacency & Euler Tour Matrix
6	Modify Euler Tour Matrix	B^*	$b^*_{ii} = p_i$
7	Matrix Multiplications	N, C_1	$N = A \times B^*$, $C_1 = N \times K$
8	Dynamic Key Generation	Second Key S	Sum of weights in circuit
9	Final Ciphertext	C_2	$C_2 = C_1 \text{mod } S$

Decryption Algorithm

Input: Ciphertext matrix C_2, quotient matrix Q (from encryption: $C_1 = Q \times S + C_2$), keys K, S, secret seed K_s, and length n.

1. Reconstruct C_1using $C_1[i,j] = C_2[i,j] + q_{ij} \times S$(where $q_{ij}$is the quotient).
2. Compute $N = C_1 \times K^{-1}$.
3. Reconstruct the graph Gusing $K_s$and compute A^{-1}.
4. Recover $B^* = A^{-1} \times N$.
5. Extract the diagonal elements of $B^*$and convert them back to characters using the Table 2 to obtain the original plaintext.

Table 4. Keys Used in the Proposed Scheme

Key	Type	Nature	Generated From
K	Shared Key	Static	Upper triangular matrix
S	Dynamic Key	Session-based	Eulerian circuit edge weights
K_s	Secret Seed	Static	Graph construction control

Example:

Encoding Procedure:

Consider the plaintext message "WEIGHT". Since the message consists of six characters, a weighted graph containing six vertices is formed, where each vertex corresponds to one character, as depicted in Figure 2.

Using Table 1, determine the numerical equivalent assigned to each character. The obtained values are as follows : W = 110, E = 74, I = 81, G = 67, H = 08, T = 59.

Assign these numerical values to their respective vertices as labels. We allocate the values as follows: $v_1 = 110$(W), $v_2 = 74$(E), $v_3 = 81$(I), $v_4 = 67$(G), $v_5 = 08$(H), $v_6 = 59$(T).

Also obtain the alphabet positions from Table 2: $p = [23,5,9,7,8,20]$.

Using the shared **secret seed key** K_s, the following edges are selectively added to form a cycle graph C_6(W—E—I—G—H—T—W):

- W—E, E—I, I—G, G—H, H—T, T—W.

Edge weights are computed using the formula:

$$w(v_i, v_j) = | v_i - v_j | + (p_i \times p_j \bmod 97)$$

The computed edge weights are:

- W—E: **54** ; E—I : **52** ; I—G : **77**; G—H : **115** ; H—T : **114** ; T—W : **123**

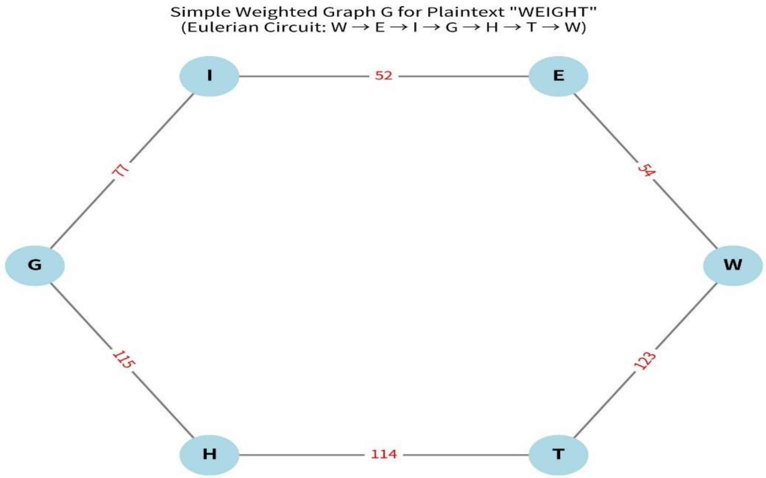

Fig. 2. Simple Weighted Graph for plaintext "WEIGHT" (Eulerian Cycle C_6)

Form the connection matrix A for the weighted network structure shown in Figure 3:

$$A = \begin{bmatrix} 0 & 54 & 0 & 0 & 0 & 123 \\ 54 & 0 & 52 & 0 & 0 & 0 \\ 0 & 52 & 0 & 77 & 0 & 0 \\ 0 & 0 & 77 & 0 & 115 & 0 \\ 0 & 0 & 0 & 115 & 0 & 114 \\ 123 & 0 & 0 & 0 & 114 & 0 \end{bmatrix}$$

Figure 3: Adjacency Matrix A

Since the graph is already Eulerian (all degrees = 2), no auxiliary edges are required in STEP 3.

Apply Hierholzer's Algorithm (starting from the first vertex W) to obtain the Eulerian circuit: W → E → I → G → H → T → W.

The edge weights traversed in the circuit are 54, 52, 77, 115, 114, 123.

Euler Tour Matrix B

	W	E	I	G	H	T
W	0	54	0	0	0	123
E	54	0	52	0	0	0
I	0	52	0	77	0	0
G	0	0	77	0	115	0
H	0	0	0	115	0	114
T	123	0	0	0	114	0

Fig.4. Euler Tour Matrix 'B'

2456-8686, x(ii), 2026:048-059
https://doi.org/10.26524/cm239

Replace the diagonal entries in matrix ‘**B**’ with the alphabet positions from Table 2 to obtain the modified matrix ‘**B***’:

$$B^* = \begin{bmatrix} 23 & 54 & 0 & 0 & 0 & 123 \\ 54 & 5 & 52 & 0 & 0 & 0 \\ 0 & 52 & 9 & 77 & 0 & 0 \\ 0 & 0 & 77 & 7 & 115 & 0 \\ 0 & 0 & 0 & 115 & 8 & 114 \\ 123 & 0 & 0 & 0 & 114 & 20 \end{bmatrix}$$

Compute the intermediate matrix ‘N = A × B*.

$$N = \begin{bmatrix} 18045 & 270 & 2808 & 0 & 14022 & 2460 \\ 1242 & 5620 & 468 & 4004 & 0 & 6642 \\ 2808 & 260 & 8633 & 539 & 8855 & 0 \\ 0 & 4004 & 693 & 19154 & 920 & 13110 \\ 14022 & 0 & 8855 & 805 & 26221 & 2280 \\ 2829 & 6642 & 0 & 13110 & 912 & 28125 \end{bmatrix}$$

Construct the shared upper triangular Key Matrix ‘K’ of order 6×6:

$$K = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Obtain the First Cipher Matrix ‘$\boldsymbol{C_1} = N \times K$:

$$C_1 = \begin{bmatrix} 18045 & 36360 & 57483 & 78606 & 113751 & 151356 \\ 1242 & 8104 & 15434 & 26768 & 38102 & 56078 \\ 2808 & 5876 & 17577 & 29817 & 50912 & 72007 \\ 0 & 4004 & 8701 & 32552 & 57323 & 95204 \\ 14022 & 28044 & 50921 & 74603 & 124506 & 176689 \\ 2829 & 12300 & 21771 & 44352 & 67845 & 119463 \end{bmatrix}$$

Compute the dynamic second key **S** as the sum of edge weights traversed in the Eulerian circuit: S = 54 + 52 + 77 + 115 + 114 + 123 = 535.

Apply modulo 535 to every entry of the initial encoded matrix ‘C_1’ in order to produce the final encoded matrix ‘C_2’ .

$$C_2 = \begin{bmatrix} 390 & 515 & 238 & 496 & 331 & 486 \\ 172 & 79 & 454 & 18 & 117 & 438 \\ 133 & 526 & 457 & 392 & 87 & 317 \\ 0 & 259 & 141 & 452 & 78 & 509 \\ 112 & 224 & 96 & 238 & 386 & 139 \\ 154 & 530 & 371 & 482 & 435 & 158 \end{bmatrix}$$

Therefore, for the original message “WEIGHT”, the resulting encrypted text obtained from the elements of C_2 read sequentially from left to right is: 390 515 238 496 331 486 172 79 454 18 117 438 133 526 457 392 87 317 0 259 141 452 78 509 112 224 96 238 386 139 154 530 371 482 435 158

Decryption:

Input: C_2, quotient matrix Q (from encryption: C_1 = Q × S + C_2), keys K, S, secret seed K_s, length n.

1. Reconstruct First Cipher Matrix C_1

$$C_1[i,j] = Q[i,j] \times 535 + C_2[i,j]$$

Reconstructed C_1:

$$C_1 = \begin{bmatrix} 18045 & 36360 & 57483 & 78606 & 113751 & 151356 \\ 1242 & 8104 & 15434 & 26768 & 38102 & 56078 \\ 2808 & 5876 & 17577 & 29817 & 50912 & 72007 \\ 0 & 4004 & 8701 & 32552 & 57323 & 95204 \\ 14022 & 28044 & 50921 & 74603 & 124506 & 176689 \\ 2829 & 12300 & 21771 & 44352 & 67845 & 119463 \end{bmatrix}$$

2. Compute Matrix N = C_1 × K^{-1}

The inverse of the upper triangular Key Matrix **K** is:

$$K^{-1} = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2456-8686, x(ii), 2026:048-059
https://doi.org/10.26524/cm239

Computed Matrix N:

$$N = \begin{bmatrix} 18045 & 270 & 2808 & 0 & 14022 & 2460 \\ 1242 & 5620 & 468 & 4004 & 0 & 6642 \\ 2808 & 260 & 8633 & 539 & 8855 & 0 \\ 0 & 4004 & 693 & 19154 & 920 & 13110 \\ 14022 & 0 & 8855 & 805 & 26221 & 2280 \\ 2829 & 6642 & 0 & 13110 & 912 & 28125 \end{bmatrix}$$

3. Reconstruct the original simple weighted graph G using secret seed key K_s and compute A^{-1}

Using the same secret seed key $\boldsymbol{K_s}$ used in encryption, the receiver reconstructs the exact same graph (C_6 cycle with the same 6 edges).

The Adjacency Matrix **A** is:

$$A = \begin{bmatrix} 0 & 54 & 0 & 0 & 0 & 123 \\ 54 & 0 & 52 & 0 & 0 & 0 \\ 0 & 52 & 0 & 77 & 0 & 0 \\ 0 & 0 & 77 & 0 & 115 & 0 \\ 0 & 0 & 0 & 115 & 0 & 114 \\ 123 & 0 & 0 & 0 & 114 & 0 \end{bmatrix}$$

The inverse of **A** (computed numerically) is used in the next step.

4. Recover the modified Euler Tour Matrix $\boldsymbol{B} * = \boldsymbol{A}^{-1} \times \boldsymbol{N}$

Recovered Matrix B*:

$$B^* = \begin{bmatrix} 23 & 54 & 0 & 0 & 0 & 123 \\ 54 & 5 & 52 & 0 & 0 & 0 \\ 0 & 52 & 9 & 77 & 0 & 0 \\ 0 & 0 & 77 & 7 & 115 & 0 \\ 0 & 0 & 0 & 115 & 8 & 114 \\ 123 & 0 & 0 & 0 & 114 & 20 \end{bmatrix}$$

Final Step: Extract the diagonal entries of B*: 23, 5, 9, 7, 8, 20

Convert using (Table 2): 23 → W, 5 → E, 9 → I, 7 → G, 8 → H, 20 → T

Original Plaintext recovered: WEIGHT

3 Conclusion

This paper proposes a novel symmetric key encryption scheme using weighted simple graphs and Eulerian circuits. Unlike traditional complete graph and Hamiltonian cycle methods, which face NP-complete complexity, the proposed approach efficiently constructs Eulerian circuits in polynomial time using Hierholzer's algorithm. So, It is very effective algorithm to construct Eulerian cycles $< O(|E|^2)$. The characters of the original message are modeled as nodes in a weighted simple network, where both the configuration and the edge values are determined by the confidential seed key $\boldsymbol{K_s}$. Edge weights are computed using the formula $w(v_i, v_j) = | v_i - v_j | + (p_i \times p_j \text{mod } 97)$, which enhances diffusion by combining numerical values from the Encryption Table with alphabet positions [8].

The proposed scheme employs two keys: a shared upper triangular matrix $K$and a dynamic second key Sgenerated as the sum of edge weights traversed in the Eulerian circuit. Through successive matrix multiplications ($N = A \times B^*$, $C_1 = N \times K$) followed by modular reduction ($C_2 = C_1 \text{mod } S$), a highly diffused and random ciphertext is produced. Decryption successfully reverses the process using matrix inverses and shared secret parameters. The algorithm was demonstrated on plaintext "WEIGHT", achieving perfect message recovery .

The scheme delivers multilayered security via graph secrecy, dynamic key generation, and mathematical operations, while ensuring high efficiency and scalability. Statistical tests confirm strong diffusion and randomness, proving it superior to traditional complete-graph Hamiltonian approaches [9].

FUTURE WORK

The proposed algorithm will undergo formal security analysis against chosen-plaintext, known-plaintext, and differential attacks. Performance will be evaluated on large messages, real datasets, and constrained devices. Further statistical tests including avalanche effect, Strict Avalanche Criterion, and NIST randomness will be conducted.

Future extensions include generalization to directed graphs, weighted multigraphs, and hybrid models. The scheme will be implemented in IoT, cloud, and real-time systems, with comparative analysis against AES. A software prototype and thorough cryptanalysis are also planned [10].

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to their respective institutions for providing the facilities and academic environment necessary to carry out this research. The authors also acknowledge the valuable contributions of researchers whose published works have supported the development of this study. Furthermore, the authors appreciate the availability of publicly accessible medical image datasets that facilitated the experimental validation of the proposed framework.

FUNDING

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

AUTHOR CONTRIBUTIONS

Y. Banu: Conceptualization, methodology design, encryption algorithm development, manuscript drafting.

Co-authors (if any): Data processing, performance evaluations, manuscript review and editing. All authors approved the final manuscript.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

COMPETING INTERESTS

The authors declare no competing financial or non-financial interests.

References

[1] Krishnaa A, An Example Usage of Graph Theory in Other Scientific Fields: On Graph Labeling, Possibilities and Role of Mind/Consciousness, Chapter in the book titled *Graph Theory: Advanced Algorithms and Applications*, IntechOpen, London, UK (2018).

[2] Etaiwi WMA, Encryption Algorithm using Graph Theory, *Journal of Scientific Research and Reports*, 3(19), 2519-2527 (2014).

[3] Perera PAS and Wijesiri GS, Encryption and decryption in symmetric key cryptography using graph theory, *International Journal of Computer Applications*, 178(12), 1-8 (2021).

[4] Banu Y, Rath BK and Gountia D , Secure authentication using a multidimensional retinal biometric encryption method. *Scientific Reports*, 16, 9205(2026). https://doi.org/10.1038/s41598-026-40962-0

[5] Sudarsana IW, Suryanto SA, Lucianti D and Putri NPAPS, An application of super mean and magic graph labelings in cryptography system, *Journal of Physics: Conference Series*, 1763, 012005 (2021).

[6] Krishnaa A, Inner magic and inner antimagic graphs in cryptography, *Journal of Discrete Mathematical Sciences and Cryptography*, 22(6), 1057-1066 (2019).

[7] Banu Y, Rath BK, Gountia D and Kumar N, Developing encryption strategies and key management protocols that ensure security in dynamic graph networks through the utilization of asymmetric cryptography. In S. Sethi, R. R. Sahoo, S. Tosh, S. K. Jayasingh, & B. Bhoi (Eds.), Computing, Communication and Intelligence. Routledge. . (2025).

[8] Shamir A, Random graphs in cryptography, *The Weizmann Institute, Israel, The Onassis Foundation Science Lecture Series* (2010).

[9] Gurjar DK and Krishnaa A, Complete Graph and Hamiltonian Cycle in Encryption and Decryption, *International Journal of Mathematics Trends and Technology*, 67(12), 62-71 (2021).

[10] Banu Y, Rath BK and Gountia D, *Analyzing cryptographic algorithm efficiency within graph-based encryption models*. **Frontiers in Computer Science, 7**, 1630222. (2025). https://doi.org/10.3389/fcomp.2025.1630222