

A Graph-Theoretic Encryption Framework: Laplacian Spectra And Chromatic SK Keys

R. Rajambigai, J. Ravi Sankar, T. Praveen*

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, Tamil Nadu - 632 014, India

**Corresponding author: praveen.t@vit.ac.in*

Abstract. Cryptography plays a critical role in securing digital communication by guarding data from unwanted access and cyber threats. This demand drives the development of creative and mathematically sound encryption methods. Various mathematical encryption techniques, especially using graph theory, have been introduced. In this paper, chromatic topological SK indices have been used for encryption methods for an improved data transfer security.

1. INTRODUCTION

The demand for safe and secured cryptographic techniques has been increased due to the growth of digital communication. Some traditional encryption techniques, even while they work, sometimes rely on proven arithmetic algorithms [1]. In order to maintain the originality and authenticity of cryptographic systems, studies have recently been investigating the use of mathematical structures such as graph theory. The application of graph theory in cryptography has drawn a lot of attention due to its possibility of creating safe and effective encryption methods, since graph-based parameters provide a comprehensive mathematical foundation that can be used to create novel cryptographic systems [2]. Topological indices are parameters that convert graphical structures into numerical terms [3,4]. They have wide applications in many fields such as QSPR analysis and so on [5,6]. By combining this field with coloring, these play a significant role in encryption [7,8]. Among these, chromatic topological indices have shown promising means of compactly securing structural information. In this work, we used chromatic SK indices, derived from applying the vertex coloring instead of the traditional degree-based SK indices, such that it captures both the structural and connecting features [9]. By utilizing coloring patterns with numerical representation, complex keys and encoding techniques that are invulnerable to straightforward attacks can

Received: Jun. 19, 2026.

2020 *Mathematics Subject Classification.* 94A60; 05C15; 05C90.

Key words and phrases. cryptography; ASCII code; XOR operation; coloring; chromatic SK indices.

be generated. The approach makes cryptographic systems more robust and unpredictable.

In this study, we integrate some basic computational operations such as binary number encoding, ASCII code, XOR operation, etc., to present a cryptographic framework derived from chromatic SK indices [10]. The binary and ASCII codes act as a smooth bridge between textual data and numerical structures from the graphs, and the application of the XOR operation guarantees computational efficiency and reversibility [11]. Therefore, these combinations provide a new approach to encode and decode secret messages.

The proposed study shows how a mathematical graph invariant can be utilized in a real-world cryptographical application, in addition to its theoretical analysis. This work advances the development of novel and mathematically grounded encryption techniques by using chromatic SK indices along with traditional encoding methods. In this study, all graphs considered are simple graphs, that is, graphs without loops or multiple edges. The color matrix $C(G)$ for a graph G is a diagonal square $n \times n$ matrix and is defined as

$$C(G) = (a_{ij}) = \begin{cases} c(v_i), & \text{if } i=j, \\ 0, & \text{otherwise.} \end{cases}$$

where $c(v_i)$ denotes the color assigned to the vertex v_i . The color adjacency matrix of a colored graph is a square $n \times n$ matrix, and it is defined as

$$A_C(G) = (a_{ij}) = \begin{cases} 1, & \text{if } u, v \text{ are adjacent and } c(u) \neq c(v), \\ -1, & \text{if } u, v \text{ are non-adjacent and } c(u) = c(v), \\ 0, & \text{otherwise.} \end{cases}$$

The Laplacian color matrix, also known as the graph Laplacian color admittance matrix, Kirchhoff color matrix, or color discrete Laplacian for a simple colored graph with n vertices is defined as (1.1),

$$L_c(G) = C(G) - A_c(G) \quad (1.1)$$

where $L_c(G)$ is the Laplacian color matrix, $C(G)$ is the color matrix and $A_c(G)$ is the color adjacency matrix. Encryption is the process of converting readable data into unreadable form using some algorithm and secret key. The main purpose of encryption is to protect information from unauthorized individuals. Decryption is the process of converting the received text to the original hidden text using some algorithm and secret key. The main purpose of decryption is to recover the original text. Plain text is the original readable data before any encryption is applied. It can be easily understood by anyone. Ciphertext is the encoded or messed-up version of plaintext. It is not understandable by everyone and needs a key to decode the original text. It is used to protect the secret message from any unauthorized access. Binary code is a way for computers to represent any text using only 0's and 1's. Standards like ASCII (American Standard Code for International Interchange) are used to assign a unique number to each character, and later the number is converted into binary code. The exclusive OR (XOR) operation is a Boolean logic function that

generates one output by comparing two inputs: if the bits are the same, then the result is 0; if the bits are different, then the result is 1. XOR operation is its own inverse.

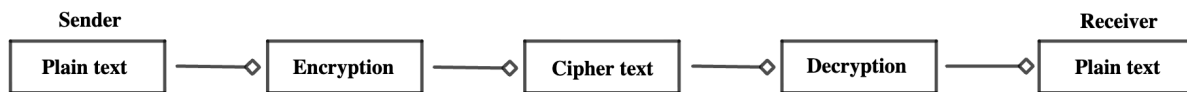
$$0 \text{ XOR } 0 = 0$$

$$0 \text{ XOR } 1 = 1$$

$$1 \text{ XOR } 0 = 1$$

$$1 \text{ XOR } 1 = 0$$

2. INFORMATION FLOW



Numerous cryptography algorithms utilizing graph theory have been introduced. In this work, we present a new encryption algorithm using a color matrix, color adjacency matrix, Laplacian color matrix, chromatic topological indices, and XOR operation. This approach is so strong because it has multi-layer security, highly sensitive to small changes, and are hard to reverse. The following table 1 shows the ASCII code and binary code of some characters .

3. CHROMATIC TOPOLOGICAL INDICES

Chromatic topological indices are numerical descriptors that capture more structural information by color partition and thus give a different perspective of the graph structure. These are extended versions of traditional topological indices by incorporating a coloring system. They have wide usage in drug discovery, network analysis, optimization problems, and pattern recognition in various fields. In this work, we have utilized coloring-based SK indices, which were originally discovered as degree-based indices by Shiegalli and Kanabur in the year 2016 [12].

$$SK^{\phi}(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u) + c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s] \quad (3.1)$$

$$SK_1^{\phi}(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u).c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} [ts\eta_{ts}] \quad (3.2)$$

$$SK_2^{\phi}(G) = \frac{1}{4} \sum_{uv \in E(G)} [c(u) + c(v)]^2 = \frac{1}{4} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s]^2 \quad (3.3)$$

where t and s denote the colors of the vertices u and v , and η_{ts} denotes the total number of edges whose vertices are colored with t and s (3.1),(3.2),(3.3).

Character	ASCII Code	Binary Number	Character	ASCII Code	Binary Number
blank space ()	32	0010 0000	^	94	0101 1110
!	33	0010 0001	_	95	0101 1111
“	34	0010 0010	‘	96	0110 0000
#	35	0010 0011	a	97	0110 0001
\$	36	0010 0100	b	98	0110 0010
A	65	0100 0001	c	99	0110 0011
B	66	0100 0010	d	100	0110 0100
C	67	0100 0011	e	101	0110 0101
D	68	0100 0100	f	102	0110 0110
E	69	0100 0101	g	103	0110 0111
F	70	0100 0110	h	104	0110 1000
G	71	0100 0111	i	105	0110 1001
H	72	0100 1000	j	106	0110 1010
I	73	0100 1001	k	107	0110 1011
J	74	0100 1010	l	108	0110 1100
K	75	0100 1011	m	109	0110 1101
L	76	0100 1100	n	110	0110 1110
M	77	0100 1101	o	111	0110 1111
N	78	0100 1110	p	112	0111 0000
O	79	0100 1111	q	113	0111 0001
P	80	0101 0000	r	114	0111 0010
Q	81	0101 0001	s	115	0111 0011
R	82	0101 0010	t	116	0111 0100
S	83	0101 0011	u	117	0111 0101
T	84	0101 0100	v	118	0111 0110
U	85	0101 0101	w	119	0111 0111
V	86	0101 0110	x	120	0111 1000
W	87	0101 0111	y	121	0111 1001
X	88	0101 1000	z	122	0111 1010
Y	89	0101 1001	{	123	0111 1011
Z	90	0101 1010		124	0111 1100
[	91	0101 1011	}	125	0111 1101
/	92	0101 1100	~	126	0111 1110
]	93	0101 1101			

TABLE 1. ASCII codes and corresponding binary representations of characters

4. METHODS

4.1. Encryption Process: First decide the plain text to be sent, and according to the plain text, consider a graph whose total number of vertices is the same as the number of letters in the plain text, including the spaces and symbols. Now, assign each letter in the plaintext with the vertices in ascending order. Then, assign colors to the vertices of the graph in such a way that the maximum number of independent vertices receive the smallest available color such that no two adjacent vertices receive the same color. Now, calculate the Laplacian color matrix according to the formula and calculate its eigenvalue. In this work, we have used python for calculating the Laplacian color matrix and its eigen values. Then add or subtract the eigenvalue (consider the nearest small integer if it is a decimal) with the ASCII code of each letter in the plaintext. Thus, we get a new ciphertext. Now, calculate the chromatic SK indices of the colored graph in order to find the secret key by using the formula (4.1).

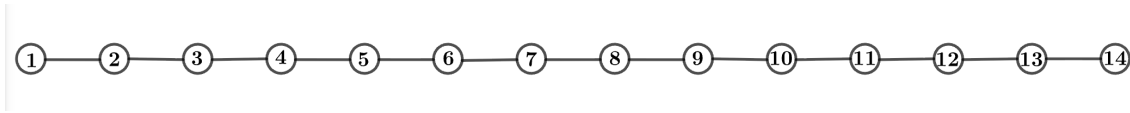
$$S = SK_i^\phi \pmod{M} \quad (4.1)$$

where S represents the secret key, SK_i^ϕ represents the chromatic SK indices where $i = 1, 2$, and M represents the total number of vertices in the graph. By applying the required value, we will get three secret keys according to each chromatic SK index. If we get a decimal number, consider the nearest small integer and find the binary code of each secret key. Now, find the ASCII code of the ciphertext and apply the XOR operation with each secret key. In this way, we get three different encrypted messages. Now, we will send the complement of the graph and this encrypted plaintext to the receiver.

4.2. Decryption process: With the graph received, find the complement of the graph and color the graph by giving the smallest possible color to the maximum number of independent vertices such that no two adjacent vertices receive the same color. Now, with the encrypted plaintext, choose any one chromatic SK index that is to be processed, and with the ciphertext of that chromatic SK index, assign each letter to the vertices in the graph in ascending order and also find that chromatic SK index. Now, with that value, find the secret key. Then, with the secret key, apply the XOR operation with the binary number of the secret key and the binary number of the received encrypted plaintext of the chosen chromatic SK index. From the resulting binary numbers, find the new ciphertext. Now, find the Laplacian color matrix according to the formula and find their eigenvalues using python. Then, subtract or add the eigenvalues with the ASCII code of the letters of the new message, and we finally get the meaningful secret plaintext sent by the sender. Therefore, the process used in encryption is applied in reverse in the decryption process. As a result, the plaintext will be reconstructed successfully.

5. MAIN RESULTS

5.1. Illustration-1. Consider the plaintext **VIT UNIVERSITY** and the graph in figure 1.

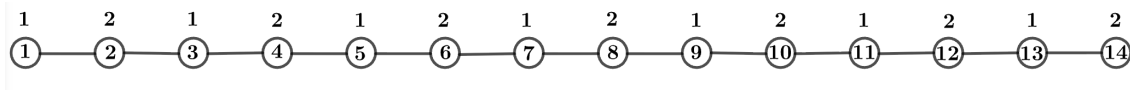
FIGURE 1. A path graph with 14 vertices (P_{14})

5.1.1. **Encryption. Step 1:** Assign each letters in the plaintext with the vertices in ascending order as in table 2.

Plaintext	V	I	T	()	U	N	I	V	E	R	S	I	T	Y
Vertices	1	2	3	4	5	6	7	8	9	10	11	12	13	14

TABLE 2. Representation of plaintext and corresponding vertices

Step 2: Apply proper coloring to the vertices of the graph as in figure 2.

FIGURE 2. Proper coloring of P_{14}

Step 3: Find the color matrix and color adjacency matrix to find the Laplacian color matrix and find the eigen values of the Laplacian color matrix using the following algorithm (1).

Algorithm 1 Computation of the Laplacian Color Matrix and its Eigenvalues for a graph G

- 1: Import NumPy library
- 2: Define the diagonal color matrix $matrix_C(G)$
- 3: Define the adjacency color matrix $matrix_Ac(G)$
- 4: Compute

$$Laplacian_matrix = matrix_C(G) - matrix_Ac(G)$$

- 5: Compute

$$eigenvalues = eigvals(Laplacian_matrix)$$

- 6: Display the Laplacian color matrix
 - 7: **for** $i = 1$ to n **do**
 - 8: Display Eigenvalue i
 - 9: **end for**
-

```

1 import numpy as np
2
3 # Matrix C(G): Diagonal Color Matrix
4 matrix_C = np.array([

```

```

5 [1,0,0,0,0,0,0,0,0,0,0,0,0,0],
6 [0,2,0,0,0,0,0,0,0,0,0,0,0,0],
7 [0,0,1,0,0,0,0,0,0,0,0,0,0,0],
8 [0,0,0,2,0,0,0,0,0,0,0,0,0,0],
9 [0,0,0,0,1,0,0,0,0,0,0,0,0,0],
10 [0,0,0,0,0,2,0,0,0,0,0,0,0,0],
11 [0,0,0,0,0,0,1,0,0,0,0,0,0,0],
12 [0,0,0,0,0,0,0,2,0,0,0,0,0,0],
13 [0,0,0,0,0,0,0,0,1,0,0,0,0,0],
14 [0,0,0,0,0,0,0,0,0,2,0,0,0,0],
15 [0,0,0,0,0,0,0,0,0,0,1,0,0,0],
16 [0,0,0,0,0,0,0,0,0,0,0,2,0,0],
17 [0,0,0,0,0,0,0,0,0,0,0,0,1,0],
18 [0,0,0,0,0,0,0,0,0,0,0,0,0,2]
19 ])

```

```

1 # Matrix A_c(G): Adjacency Color Matrix
2 matrix_Ac = np.array([
3 [0,1,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0],
4 [1,0,1,-1,0,-1,0,-1,0,-1,0,-1,0,-1],
5 [-1,1,0,1,-1,0,-1,0,-1,0,-1,0,-1,0],
6 [0,-1,1,0,1,-1,0,-1,0,-1,0,-1,0,-1],
7 [-1,0,-1,1,0,1,-1,0,-1,0,-1,0,-1,0],
8 [0,-1,0,-1,1,0,1,-1,0,-1,0,-1,0,-1],
9 [-1,0,-1,0,-1,1,0,1,-1,0,-1,0,-1,0],
10 [0,-1,0,-1,0,-1,1,0,1,-1,0,-1,0,-1],
11 [-1,0,-1,0,-1,0,-1,1,0,1,-1,0,-1,0],
12 [0,-1,0,-1,0,-1,0,-1,1,0,1,-1,0,-1],
13 [-1,0,-1,0,-1,0,-1,0,-1,1,0,1,-1,0],
14 [0,-1,0,-1,0,-1,0,-1,0,-1,1,0,1,-1],
15 [-1,0,-1,0,-1,0,-1,0,-1,0,-1,1,0,1],
16 [0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,1,0]
17 ])
18
19 resulting_Laplacian_matrix = matrix_C - matrix_Ac
20 eigenvalues = np.linalg.eigvals(resulting_Laplacian_matrix)
21
22 print("Laplacian Color Matrix")
23 print(resulting_Laplacian_matrix)
24
25 print("\nEigenvalues")
26 for i, val in enumerate(eigenvalues):
27     print(f"\u03BB{i+1}: {val}")

```

LISTING 1. Computation of Laplacian color matrix and its eigenvalues of P_{14} using Python

```

1 Laplacian Color Matrix
2 [[ 1 -1 1 0 1 0 1 0 1 0 1 0 1 -1]
3 [-1 2 -1 1 0 1 0 1 0 1 0 1 0 1]
4 [ 1 -1 1 -1 1 0 1 0 1 0 1 0 1 0]
5 [ 0 1 -1 2 -1 1 0 1 0 1 0 1 0 1]
6 [ 1 0 1 -1 1 -1 1 0 1 0 1 0 1 0]
7 [ 0 1 0 1 -1 2 -1 1 0 1 0 1 0 1]
8 [ 1 0 1 0 1 -1 1 -1 1 0 1 0 1 0]
9 [ 0 1 0 1 0 1 -1 2 -1 1 0 1 0 1]
10 [ 1 0 1 0 1 0 1 -1 1 -1 1 0 1 0]
11 [ 0 1 0 1 0 1 0 1 -1 2 -1 1 0 1]
12 [ 1 0 1 0 1 0 1 0 1 -1 1 -1 1 0]
13 [ 0 1 0 1 0 1 0 1 0 1 -1 2 -1 1]
14 [ 1 0 1 0 1 0 1 0 1 0 1 -1 1 -1]
15 [-1 1 0 1 0 1 0 1 0 1 0 1 -1 2]]
16 Eigenvalues
17 \u03BB_1: 9.56155281280883
18 \u03BB_2: 5.438447187191168
19 \u03BB_3: 2.370021284295309
20 \u03BB_4: 1.8434873025404346
21 \u03BB_5: 1.1693745320783886
22 \u03BB_6: -0.16937453207838932
23 \u03BB_7: -0.8434873025404341
24 \u03BB_8: -1.3700212842953077
25 \u03BB_9: -1.3700212842953061
26 \u03BB_10: -0.843487302540434
27 \u03BB_11: -0.16937453207838893
28 \u03BB_12: 2.3700212842953037
29 \u03BB_13: 1.1693745320783884
30 \u03BB_14: 1.8434873025404337

```

TABLE 3. Output (Laplacian color matrix and its eigenvalues of P_{14})

Step 4: Now, add or subtract the eigen values with the corresponding ASCII code of the letters in the plain text. In this we use subtraction and find the new cipher text as in the table 4.

New ciphertext: "MDU!UNIVCPRHSX"

Step 5: Calculate the chromatic SK indices of the graph and consider the nearest small integer.

$$(i) SK^{\phi}(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u) + c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s] = \frac{1}{2}[13(1 + 2)] = [19.5] = 19$$

$$(ii) SK_1^{\phi}(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u).c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} [ts\eta_{ts}] = \frac{1}{2}[13(1)(2)] = 13$$

$$(iii) SK_2^{\phi}(G) = \frac{1}{4} \sum_{uv \in E(G)} [c(u) + c(v)]^2 = \frac{1}{4} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s]^2 = \frac{1}{4}[13(1 + 2)^2] = [29.25] = 29$$

Ciphertext	ASCII	Eigen Value (λ)	ASCII- λ	ASCII	Plaintext
V	86	9	86-9	77	M
I	73	5	73-5	68	D
T	84	-1	84+1	85	U
()	32	-1	32+1	33	!
U	85	0	85+0	85	U
N	78	0	78+0	78	N
I	73	0	73+0	73	I
V	86	0	86+0	86	V
E	69	2	69-2	67	C
R	82	2	82-2	80	P
S	83	1	83-1	82	R
I	73	1	73-1	72	H
T	84	1	84-1	83	S
Y	89	1	89-1	88	X

TABLE 4. Decryption process using eigenvalues of the graph

Step 6: To find the secret key, the following formula is utilized,

$$S = SK_i^\phi \pmod{M}$$

$$S_1 = SK^\phi \pmod{M} = 19 \pmod{14} = 5$$

$$S_2 = SK_1^\phi \pmod{M} = 13 \pmod{14} = 13$$

$$S_3 = SK_2^\phi \pmod{M} = 29 \pmod{14} = 1$$

The binary codes of the obtained secret keys are $S_1 = 101$, $S_2 = 1101$, and $S_3 = 1$, respectively.

Step 7: The following table 5 shows the XOR operation of the binary codes of the new ciphertext with the binary codes of the secret keys.

Therefore, we got the encrypted ciphertext to be sent to the receiver, and they are listed below in table 6. Along with this, the sender sends the complement of the original graph to the receiver.

5.1.2. **Decryption.** The ciphertext and the graph received by the receiver are in the figure 3 and the table 6.

Step 1: Find the complement of the received graph and apply proper coloring as in figure 4.



FIGURE 4. Proper coloring of the decrypted graph

XOR operation			
Ciphertext	S_1	S_2	S_3
M-10011010	1001000-H	1000000-@	1001100-L
D-1000100	1000001-A	1001001-I	1000101-E
U-1010101	1010000-P	1011000-X	1010100-T
!-100001	100100-\$	101100-,	100000-O
U-1010101	1010000-P	1011000-X	1010100-T
N-1001110	1001011-K	1000011-C	1001111-O
I-1001001	1001100-L	1000100-D	1001000-H
V-1010110	1010011-S	1011011-[-	1010111-W
C-1000011	1000110-F	1001110-N	1000010-B
P-1010000	1010101-U	1011101-]	1010001-Q
R-1010010	1010111-W	1011111-_	1010011-S
H-1001000	1001101-M	1000101-E	1001001-I
S-1010011	1010110-V	1011110-^	1010010-R
X-1011000	1011101-)	1010101-U	1011001-Y

TABLE 5. XOR operation between ciphertext and secret keys

SK^ϕ	HAP\$PKLSFUWMV)
SK_1^ϕ	@IX,XCD[N]_E^U
SK_2^ϕ	LET TOHWBQSIRY

TABLE 6. Encrypted ciphertexts generated using different secret keys

SK^ϕ	HAP\$PKLSFUWMV)
SK_1^ϕ	@IX,XCD[N]_E^U
SK_2^ϕ	LET TOHWBQSIRY

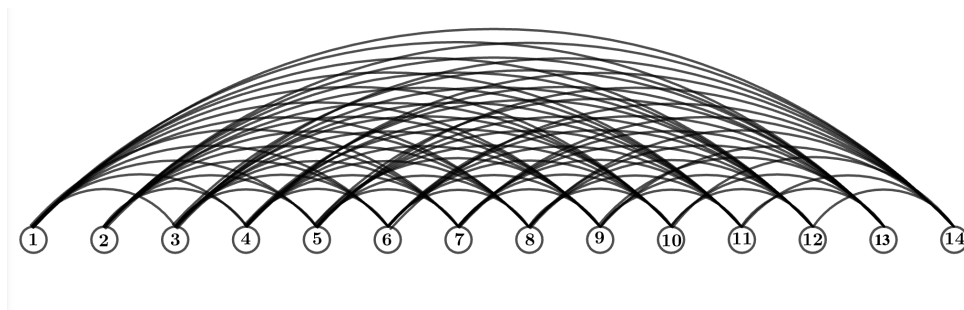


FIGURE 3. Cipher graph

Step 2: Calculate the chromatic SK indices and for each cipher text we assign the vertices in ascending order to each letter of the cipher text as in table 7. Then, calculate the chromatic SK indices and consider the nearest least integer.

Ciphertext 1	H	A	P	\$	P	K	L	S	F	U	W	M	V	)
Ciphertext 2	@	I	X	,	X	C	D	[	N	]	_	E	^	U
Ciphertext 3	L	E	T	(	T	O	H	W	B	Q	S	I	R	Y
Vertices	1	2	3	4	5	6	7	8	9	10	11	12	13	14

TABLE 7. Ciphertexts and corresponding vertex assignments

$$SK^\phi(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u) + c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s] = \frac{1}{2}[13(1 + 2)] = [19.5] = 19$$

$$SK_1^\phi(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u).c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} [ts\eta_{ts}] = \frac{1}{2}[13(1)(2)] = 13$$

$$SK_2^\phi(G) = \frac{1}{4} \sum_{uv \in E(G)} [c(u) + c(v)]^2 = \frac{1}{4} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s]^2 = \frac{1}{4}[13(1 + 2)^2] = [29.25] = 29$$

With the chromatic SK indices value, we can get the secret keys S_1, S_2, S_3 , and with the binary code of the secret keys, apply the XOR operation among the binary code of the secret keys and the binary code of the respective cipher texts, and we get a new cipher text as in table 8.

$$S_1 = SK^\phi \pmod{M} = 19 \pmod{14} = 5$$

$$S_2 = SK_1^\phi \pmod{M} = 13 \pmod{14} = 13$$

$$S_3 = SK_2^\phi \pmod{M} = 29 \pmod{14} = 1$$

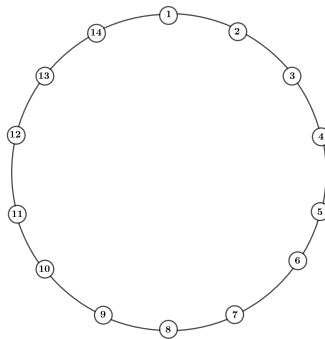
Therefore, the new cipher code is "MDU!UNIVCPRHSX".

Step 3: Find the Laplacian color matrix of the graph and its eigenvalues from the matrix 1.

Step 4: Now add these eigenvalues with the ASCII code of the new ciphertext to find the hidden message sent by the sender as in table 9.

Therefore, the hidden message sent by the sender is "VIT UNIVERSITY".

5.2. *Illustration-2.* Consider the plaintext **VIT UNIVERSITY** and the graph in figure 5.

FIGURE 5. Cycle graph with 14 vertices (C_{14})

XOR operation			
SK^ϕ	SK_1^ϕ	SK_2^ϕ	Ciphertext
H-1001000	@-1000000	L-1001100	10011010-M
A-1000001	I-1001001	E-1000101	1000100-D
P-1010000	X-1011000	T-1010100-	1010101-U
\$-100100	-101100	()-100000	100001-!
P-1010000	X-1011000	T-1010100	1010101-U
K-1001011	C-1000011	O-1001111	1001110-N
L-1001100	D-1000100	H-1001000	1001001-I
S-1010011	[-1011011	W-1010111	1010110-V
F-1000110	N-1001110-	B-1000010	1000011-C
U-1010101	]1011101	Q-1010001	1010000-P
W-1010111	_1011111	S-1010011	1010010-R
M-1001101	E-1000101	I-1001001	1001000-H
V-1010110	^1011110	R-1010010	1010011-S
)1011101	U-1010101	Y-1011001	1011000-X

TABLE 8. XOR operation performed between secret keys and ciphertext during decryption

Ciphertext	ASCII	Eigen Value (λ)	ASCII+ λ	ASCII	Plaintext
M	77	9	77+9	86	V
D	68	5	68+5	73	I
U	85	-1	85-1	84	T
!	33	-1	33-1	32	()
U	85	0	85-0	85	U
N	78	0	78-0	78	N
I	73	0	73-0	73	I
V	86	0	86-0	86	V
C	67	2	67+2	69	E
P	80	2	80+2	82	R
R	82	1	82+1	83	S
H	72	1	72+1	73	I
S	83	1	83+1	84	T
X	88	1	88+1	89	Y

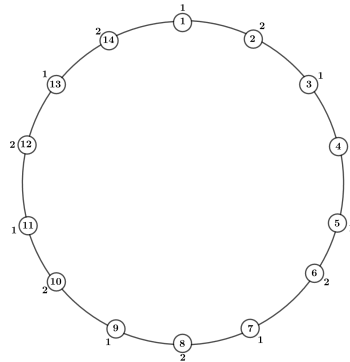
TABLE 9. Decryption process using eigenvalues of the graph

5.2.1. **Encryption. Step 1:** Assign the vertices to each letter in the plaintext in ascending order as in table 10.

Step 2: Apply proper coloring to the vertices of the graph and is shown in the figure 6.

Plaintext	V	I	T	()	U	N	I	V	E	R	S	I	T	Y
Vertices	1	2	3	4	5	6	7	8	9	10	11	12	13	14

TABLE 10. Representation of plaintext and corresponding vertices

FIGURE 6. Proper coloring of C_{14}

Step 3: Find the color matrix and color adjacency matrix to find the Laplacian color matrix and find the eigen values of the Laplacian color matrix using the algorithm 1.

```

1 Laplacian Color Matrix
2 [[ 1 -1 1 0 1 0 1 0 1 0 1 0 1 -1]
3 [-1 2 -1 1 0 1 0 1 0 1 0 1 0 1]
4 [ 1 -1 1 -1 1 0 1 0 1 0 1 0 1 0]
5 [ 0 1 -1 2 -1 1 0 1 0 1 0 1 0 1]
6 [ 1 0 1 -1 1 -1 1 0 1 0 1 0 1 0]
7 [ 0 1 0 1 -1 2 -1 1 0 1 0 1 0 1]
8 [ 1 0 1 0 1 -1 1 -1 1 0 1 0 1 0]
9 [ 0 1 0 1 0 1 -1 2 -1 1 0 1 0 1]
10 [ 1 0 1 0 1 0 1 -1 1 -1 1 0 1 0]
11 [ 0 1 0 1 0 1 0 1 -1 2 -1 1 0 1]
12 [ 1 0 1 0 1 0 1 0 1 -1 1 -1 1 0]
13 [ 0 1 0 1 0 1 0 1 0 1 -1 2 -1 1]
14 [ 1 0 1 0 1 0 1 0 1 0 1 -1 1 -1]
15 [-1 1 0 1 0 1 0 1 0 1 0 1 -1 2]]
16
17 Eigenvalues
18 \u03BB_1: 9.56
19 \u03BB_2: 5.43
20 \u03BB_3: 2.37
21 \u03BB_4: 1.84
22 \u03BB_5: 1.16
23 \u03BB_6: -0.16
24 \u03BB_7: -0.84
25 \u03BB_8: -1.37
26 \u03BB_9: -1.37
27 \u03BB_10: -0.84
28 \u03BB_11: -0.16
29 \u03BB_12: 2.37
30 \u03BB_13: 1.16
31 \u03BB_14: 1.84

```

TABLE 11. Output (Laplacian color matrix and its eigenvalues of C_{14})

Step 4: Now, add or subtract the eigen values with the corresponding ASCII code of the letters in the plain text. In this we use addition and find the new cipher text as in the table 12.

Ciphertext	ASCII	Eigen Value (λ)	ASCII+ λ	ASCII	Plaintext
V	86	9	86+9	95	_
I	73	5	73+5	78	N
T	84	2	84+2	86	V
()	32	1	32+1	33	!
U	85	1	85+1	86	V
N	78	0	78-0	78	N
I	73	0	73-0	73	I
V	86	-1	86-1	85	U
E	69	-1	69-1	68	D
R	82	0	82-0	82	R
S	83	0	83-0	83	S
I	73	2	73+2	75	K
T	84	1	84+1	85	U
Y	89	1	89+1	90	Z

TABLE 12. Decryption process using eigenvalues of the graph

New ciphertext: “_ NV!VNIUDRSKUZ”

Step 5: Calculate the chromatic SK indices of the graph and consider the nearest small integer

$$\begin{aligned}
 (i) SK^{\phi}(G) &= \frac{1}{2} \sum_{uv \in E(G)} [c(u) + c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts} [t + s] = \frac{1}{2} [14(1 + 2)] = 21 \\
 (ii) SK_1^{\phi}(G) &= \frac{1}{2} \sum_{uv \in E(G)} [c(u) \cdot c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} [ts \eta_{ts}] = \frac{1}{2} [14(1)(2)] = 14 \\
 (iii) SK_2^{\phi}(G) &= \frac{1}{4} \sum_{uv \in E(G)} [c(u) + c(v)]^2 = \frac{1}{4} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts} [t + s]^2 = \frac{1}{4} [14(1 + 2)^2] = [31.5] = 31
 \end{aligned}$$

Step 6: To find the secret key, the following formula is utilized,

$$\begin{aligned}
 S &= SK_i^{\phi} \pmod{M} \\
 S_1 &= SK^{\phi} \pmod{M} = 21 \pmod{14} = 7 \\
 S_2 &= SK_1^{\phi} \pmod{M} = 14 \pmod{14} = 0 \\
 S_3 &= SK_2^{\phi} \pmod{M} = 31 \pmod{14} = 3
 \end{aligned}$$

The binary code of the obtained secret keys are $S_1 = 111$, $S_2 = 0$ and $S_3 = 11$ respectively.

Step 7: The following table 13 shows the XOR operation of the binary codes of the new ciphertext with the binary codes of the secret keys.

XOR operation			
Ciphertext	S_1	S_2	S_3
_ -1011111	1011000-X	1011111-_	1011100-\
N-1001110	1001001-I	1001110-N	1001101-M
V-1010110	1010001-Q	1010110-V	1010101-U
!-100001	100110-&	100001-!	100010-"
V-1010110	1010001-Q	1010110-V	1010101-U
N-1001110	1001001-I	1001110-N	1001101-M
I-1001001	1001110-N	1001001-I	1001010-J
U-1010101	1010010-R	1010101-U	1010110-V
D-1000100	1000011-C	1000100-D	1000111-G
R-1010010	1010101-U	1010010-R	1010001-Q
S-1010011	1010100-T	1010011-S	1010000-P
K-1001011	1001100-L	1001011-K	1001000-H
U-1010101	1010010-R	1010101-U	1010110-V
Z-1011010	1011101-J	1011010-Z	1011001-Y

TABLE 13. XOR operation between ciphertext and secret keys

Therefore, we got the encrypted ciphertext to be sent to the receiver and they are listed below in the table 14. Along with this, the sender sends the complement of the original graph to the receiver.

SK^ϕ	XIQ&QINRCUTLR]
SK_1^ϕ	_NV!VNIUDRSKUZ
SK_2^ϕ	\MU"UMJVGQPHVY

TABLE 14. Encrypted ciphertexts generated using different secret keys

5.2.2. *Decryption*. The ciphertext and the graph received by the receiver are in the figure 7 and the table 14,

SK^ϕ	XIQ&QINRCUTLR]
SK_1^ϕ	_NV!VNIUDRSKUZ
SK_2^ϕ	\MU"UMJVGQPHVY

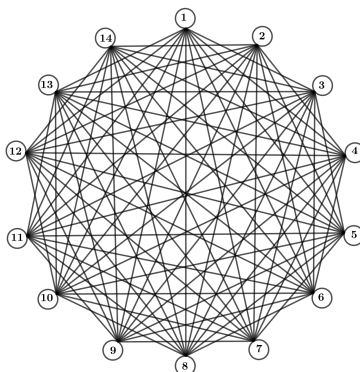


FIGURE 7. Complement of the cipher graph

Step 1: Find the complement of the cipher graph and apply proper coloring as in figure 8.

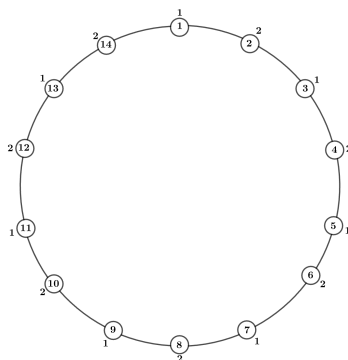


FIGURE 8. Proper coloring of the decrypted graph

Step 2: We assign the vertices in ascending order to each letter of the cipher text as in table 15. Then, calculate the chromatic SK indices and consider the nearest least integer find the secret key using the equation (4.1).

Ciphertext 1	X	I	Q	&	Q	I	N	R	C	U	T	L	R	J
Ciphertext 2	_	N	V	!	V	N	I	U	D	R	S	K	U	Z
Ciphertext 3	\	M	U	"	U	M	J	V	G	Q	P	H	V	Y
Vertices	1	2	3	4	5	6	7	8	9	10	11	12	13	14

TABLE 15. Representation of ciphertexts and associated vertices

$$SK^\phi(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u) + c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s] = \frac{1}{2} [14(1 + 2)] = 21$$

$$SK_1^\phi(G) = \frac{1}{2} \sum_{uv \in E(G)} [c(u) \cdot c(v)] = \frac{1}{2} \sum_{1 \leq t < s \leq \chi(G)} [ts\eta_{ts}] = \frac{1}{2}[14(1)(2)] = 14$$

$$SK_2^\phi(G) = \frac{1}{4} \sum_{uv \in E(G)} [c(u) + c(v)]^2 = \frac{1}{4} \sum_{1 \leq t < s \leq \chi(G)} \eta_{ts}[t + s]^2 = \frac{1}{4}[14(1 + 2)^2] = \lfloor 31.5 \rfloor = 31$$

With the chromatic SK_1 index value, we can get the secret key S_1, S_2, S_3 , and with the binary code of the secret key, apply the XOR operation among the binary code of the secret keys and the binary code of the respective ciphertext, and we get a new ciphertext in the table 16.

$$S_1 = SK_1^\phi \pmod{M} = 21 \pmod{14} = 7$$

$$S_2 = SK_1^\phi \pmod{M} = 14 \pmod{14} = 0$$

$$S_3 = SK_2^\phi \pmod{M} = 31 \pmod{14} = 3$$

XOR operation			
SK^ϕ	SK_1^ϕ	SK_2^ϕ	Ciphertext
X-1011000	-1011111	\-1011100	1011111-
I-1001001	N-1001110	M-1001101	1001110-N
Q-1010001	V-1010110	U-1010101	1010110-V
&-1001110	!-100001	“-100010	100001-!
Q-1010001	V-1010110	U-1010101	1010110-V
I-1001001	N-1001110	M-1001101	1001110-N
N-1001110	I-1001001	J-1001010	1001001-I
R-1010010	U-1010101	V-1010110	1010101-U
C-1000011	D-1000100	G-1000111	1000100-D
U-1010101	R-1010010	Q-1010001	1010010-R
T-1010100	S-1010011	P-1010000	1010011-S
L-1001100	K-1001011	H-1001000	1001011-K
R-1010010	U-1010101	V-1010110	1010101-U
J-1011101	Z-1011010	Y-1011001	1011010-Z

TABLE 16. XOR operation performed between secret keys and ciphertext during decryption

Therefore, the new cipher code is “_NV!VNIUDRSKUZ”.

Step 3: Find the Laplacian color matrix of the graph and its eigen values using the algorithm 1.

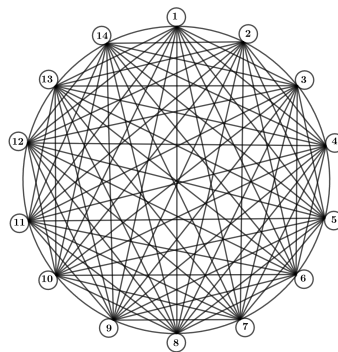
Step 4: Now subtract these eigen values with the ASCII code of the new ciphertext to find the hidden message sent by the sender as in the table 17.

Therefore, the hidden message sent by the sender is “VIT UNIVERSITY”.

Ciphertext	ASCII	Eigen Value (λ)	ASCII- λ	ASCII	Plaintext
_	95	9	95-9	86	V
N	78	5	78-5	73	I
V	86	2	86-2	84	T
!	33	1	33-1	32	()
V	86	1	86-1	85	U
N	78	0	78-0	78	N
I	73	0	73-0	73	I
U	85	-1	85+1	86	V
D	68	-1	68+1	69	E
R	82	0	82-0	82	R
S	83	0	83-0	83	S
K	75	2	75-2	73	I
U	85	1	85-1	84	T
Z	90	1	90-1	89	Y

TABLE 17. Decryption process using eigenvalues of the graph

5.3. **Illustration-3.** For the entire graph K_{14} in the figure 9, the encryption and decryption processes were performed in a similar manner using the identical plaintext "VIT UNIVERSITY". Vertex assignment, appropriate coloring, building the color matrix and color adjacency matrix, calculating the Laplacian color matrix and its eigenvalues, creating chromatic SK indices, creating secret keys, XOR operations, encryption, and decryption procedures were all carried out using the same methodology as the above illustration-1 and illustration-2. The specific computations are left out because the computational procedures are similar to those described. The recovered plaintext and the encrypted ciphertexts that were acquired confirm the accuracy and effectiveness of the suggested cryptographic method for the entire graph and are listed in the table 18.

FIGURE 9. Complete graph with 14 vertices (K_{14})

$$S_1 = SK^\phi \pmod{M} = 682 \pmod{14} = 10$$

$$S_2 = SK_1^\phi \pmod{M} = 2502 \pmod{14} = 10$$

$$S_3 = SK_2^\phi \pmod{M} = 5801 \pmod{14} = 5$$

The binary codes of the obtained secret keys are $S_1 = 1010$, $S_2 = 1010$, and $S_3 = 101$, respectively.

SK^ϕ	Z]\kX`QASW[Wn
SK_1^ϕ	Z]\kX`QASW[Wn
SK_2^ϕ	URS& dWQ`N\XTXa

TABLE 18. Encrypted ciphertexts generated using different secret keys

The new ciphertext after applying the XOR operation is "PWV# aRT[KY]Q]d". If we add the eigen values of the Laplacian color matrix with the ASCII code of the new cipher text we get the ASCII code of the plaintext "VIT UNIVERSITY" and from that we recover the original hidden text sent by the receiver.

5.4. Illustration-4. The suggested cryptographic approach is applicable to any linked random graph with the same plaintext "VIT UNIVERSITY" for example the figure 10. The encryption and decryption techniques are equivalent regardless of graph structure. Initially, the connected graph's vertices are assigned with the letters in the plaintext in ascending order. The appropriate coloring is then applied, and the associated color matrix, color adjacency matrix, and Laplacian color matrix are created. The Laplacian color matrix is used to calculate the graph's eigenvalues, which are then used in the encryption process. The secret keys are generated by computing the chromatic SK indices, followed by XOR operations to yield the encrypted ciphertexts as in the table 19.

On the recipient side, the complement of the transmitted graph is used to recreate the original graph. To recover the intermediate ciphertext, the same chromatic SK indices and secret keys are used, followed by an XOR operation. Finally, the respective eigenvalues are used to successfully recover the original plaintext "VIT UNIVERSITY".

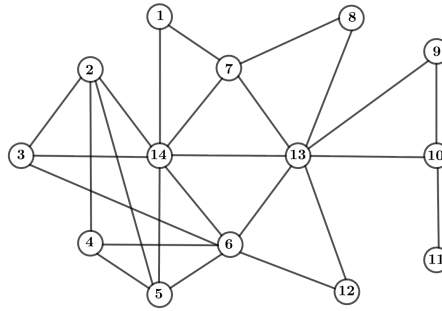
The encryption and decryption techniques for connected random graphs use the same methodology as in the previous illustrations, therefore extensive computations are omitted. The results show that the proposed graph-theoretic cryptography model may be effectively implemented on any connected random graph while maintaining the correctness, dependability, and security of the transmitted message. Let us take the following graph as an example.

$$S_1 = SK^\phi \pmod{M} = 46 \pmod{14} = 4$$

$$S_2 = SK_1^\phi \pmod{M} = 42 \pmod{14} = 0$$

$$S_3 = SK_2^\phi \pmod{M} = 96 \pmod{14} = 12$$

The binary codes of the obtained secret keys are $S_1 = 100$, $S_2 = 0$, and $S_3 = 1100$, respectively.

FIGURE 10. Connected graph with 14 vertices (G_{14})

SK^ϕ	YTV'PIH\CWPNPW
SK_1^ϕ	]PR#TMLXGSTJTS
SK_2^ϕ	Q\`XA@TK_XFX_

TABLE 19. Encrypted ciphertexts generated using different secret keys

The new ciphertext after applying the XOR operation is "]PR#TMLXGSTJTS". If we subtract the respective eigen values of the Laplacian color matrix with the ASCII code of the new cipher text we get the ASCII code of the plaintext "VIT UNIVERSITY" and from that we obtain the original hidden message sent by the receiver.

6. CONCLUSION

This paper combined XOR operations, binary number encoding, and ASCII transformation with graph-theoretic ideas to offer a unique cryptographic framework by using chromatic SK indices. The result shows that the technique effectively makes use of structural characteristics to produce safe and effective encryption. Notably, when different graphs are used, the same plaintext generates distinct ciphertexts under the same technique, underscoring the heavy reliance of this encryption process on graph structure. This variability improves strong security because of its unpredictability and susceptibility to cyber attacks. Therefore, the suggested method demonstrates that chromatic topological indices can serve as an effective tool for creating flexible and mathematically strong cryptographic techniques with potential for further research exploration using different graph parameters and developing advanced security analysis.

CRedit Author Contribution Statement: Rajambigai R: Conceptualisation, Methodology, Software, Resources, Validation; Praveen T: Supervision, Validation, Conceptualisation, Investigation; Ravi Sankar J: Conceptualisation, Supervision, Writing-Reviewing and Editing, Visualisation, Formal analysis, Validation.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

REFERENCES

- [1] M. Bellare, P. Rogaway, Introduction to Modern Cryptography, Lecture Notes, University of California San Diego, 2005. <https://cseweb.ucsd.edu/~mihir/cse207/classnotes.html>.
- [2] N. Deo, Graph Theory with Applications to Engineering and Computer Science, Dover Publications, Mineola, 2016.
- [3] C.J. Rayer, R.S. Jeyaraj, Wiener Index and Graph Energy of Zero Divisor Graph for Commutative Rings, Asian-Eur. J. Math. 16 (2023), 2350211. <https://doi.org/10.1142/S179355712350211X>.
- [4] J.R. Sankar, A. Felix, An Improved Fuzzy DEMATEL Technique and Its Applications, Int. J. Pharm. Technol. 8 (2016), 19122–19134.
- [5] B. Venkateswarlu, B. Mahaboob, C. Subbarami Reddy, J. Ravi Sankar, A Study on Technical Efficiency of a DMU (Review of Literature), IOP Conf. Ser.: Mater. Sci. Eng. 263 (2017), 042124. <https://doi.org/10.1088/1757-899X/263/4/042124>.
- [6] S. Kour, J. Ravi Sankar, Machine Learning Regression Models for Predicting Anti-Cancer Drug Properties: Insights from Topological Indices in QSPR Analysis, Contemp. Math. 5 (2024), 6515–6526. <https://doi.org/10.37256/cm.5420245826>.
- [7] S. Kour, J. Ravi Sankar, Characterization of Tricyclic Anti-Depressant Drugs Efficacy via Topological Indices, Sci. Rep. 15 (2025), 22853. <https://doi.org/10.1038/s41598-025-05045-6>.
- [8] E. Pandeewari, J. Ravi Sankar, Topological Insights into Breast Cancer Drugs: A QSPR Approach Using Resolving Topological Indices, Front. Chem. 13 (2025), 1710442. <https://doi.org/10.3389/fchem.2025.1710442>.
- [9] T.R. Jensen, B. Toft, Graph Coloring Problems, Wiley, New York, 1995. <https://doi.org/10.1002/9781118032497>.
- [10] R. Rajambigai, T. Praveen, J. Ravi Sankar, Structure–Efficiency Relationship of Access Group Antibiotics via SK Chromatic Descriptors, Front. Chem. 14 (2026), 1763932. <https://doi.org/10.3389/fchem.2026.1763932>.
- [11] P. Amudha, A.C.C. Sagayaraj, A.C.S. Sheela, An Application of Graph Theory in Cryptography, Int. J. Pure Appl. Math. 119 (2018), 375–383.
- [12] N.S. Ganapathi, H.R. Medini, S. D'Souza, Application of Graph Complements and Labeling in Secure Communication, Eng. Lett. 33 (2025), 4605–4611.
- [13] V. Shigehalli, R. Kanabur, New Version of Degree-Based Topological Indices of Certain Nanotube, J. Discret. Math. Appl. 9 (2024), 17–30.