A Novel Dual-Layer Graph Encryption Scheme Using Total Colouring and Matrix-Based Encoding

A. Punitha

Assistant Professor, Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamil Nadu, India.

Email id: punithasokan@gmail.com

Abstract

The increasing reliance on digital communication systems necessitates the development of secure and efficient encryption techniques. In this paper, a graph-theoretic cryptographic framework based on total coloring of path graphs is introduced. The proposed method represents plaintext as a path graph, where both vertices and edges are assigned colors under total coloring constraints, resulting in a dual-layer encoding structure. This representation is transformed into a matrix form and encrypted using an invertible key matrix. Unlike traditional graph labeling approaches, the proposed scheme enhances structural complexity by incorporating both vertex and edge dependencies. The correctness of the method is theoretically justified, and its practical applicability is demonstrated through an illustrative example. Furthermore, the scheme exhibits linear computational complexity with respect to input size, making it suitable for efficient implementation. The approach provides improved resistance to structural and brute-force attacks due to its combinatorial encoding mechanism.

Keywords

Cryptography, Graph Theory, Total Coloring, Path Graph, Encryption Algorithm, Secure Communication

AMS classification: 05C15, 94A60, 68P25

1. Introduction

Secure communication has become a critical requirement in modern digital systems, where sensitive data is continuously transmitted across open networks. Cryptographic techniques play a central role in safeguarding such information from unauthorized access. Conventional encryption methods are primarily based on number theory and algebraic structures; however, these approaches may encounter limitations in terms of scalability and adaptability to emerging data representations.

In recent years, graph theory has provided an alternative perspective for designing encryption mechanisms due to its inherent ability to model structural relationships. Several studies have explored graph labeling techniques for cryptographic applications, where numerical values are assigned to vertices or edges to encode information. While these methods offer interesting mathematical properties, they often rely on single-layer encoding, which may limit their resistance to advanced cryptanalytic attacks.

The present work addresses this limitation by introducing a cryptographic scheme based on total coloring of path graphs. Unlike labeling approaches, total coloring simultaneously assigns values to both vertices and edges under strict constraints, thereby increasing combinatorial complexity. This dual-layer encoding mechanism enhances the security of the system by introducing additional structural dependencies that are difficult to reconstruct without knowledge of the key.

The main contributions of this work are summarized as follows:

- Development of a total coloring-based encryption model
- Design of corresponding encryption and decryption algorithms

- Theoretical validation of correctness
- Analysis of computational efficiency and security aspects

The application of graph theory in cryptography has attracted increasing attention in recent years due to its flexibility in representing structured data. Early contributions in this direction primarily focused on graph labeling techniques, where numerical values are assigned to vertices or edges under specific constraints. These labeled structures have been utilized to encode information in a systematic manner.

Amudha et al. [1] proposed an encryption scheme based on graph labeling that employs adjacency matrices and spanning tree concepts for secure data transmission. While their method demonstrates the feasibility of graph-based encoding, it largely depends on edge labeling, which limits the depth of structural representation. In contrast, more generalized surveys, such as that of Gallian [2], provide an extensive overview of labeling techniques, including graceful and antimagic labeling, highlighting their theoretical significance and applicability in various domains. However, their direct application to cryptographic systems remains relatively constrained.

Further advancements were made by MacDougall et al. [3], who explored total labeling and total coloring of graphs. These approaches extend traditional labeling by incorporating both vertices and edges within a unified framework, thereby increasing combinatorial complexity. Despite their strong mathematical foundation, the potential of total coloring in cryptographic design has not been fully explored.

From a broader cryptographic perspective, Kahate [5] emphasized the importance of key management, computational efficiency, and algorithmic robustness in secure communication systems. Modern encryption techniques must not only ensure confidentiality but also adapt to diverse data representations and evolving attack strategies.

A careful examination of the existing literature reveals that most graph-based encryption methods rely on single-layer encoding mechanisms, where either vertices or edges are considered independently. This separation restricts the level of structural interdependence that can be achieved within the encoding process. Consequently, such methods may be more vulnerable to structural reconstruction attacks.

Motivated by this limitation, the present work introduces a cryptographic framework based on total coloring of path graphs. By simultaneously assigning values to both vertices and edges under strict coloring constraints, the proposed approach establishes a dual-layer encoding mechanism. This significantly enhances the structural complexity of the encrypted representation and provides an additional layer of security compared to conventional graph labeling techniques.

2. Preliminaries

Definition 1: A graph $G = (V, E)$ consists of a set of vertices V and edges E. Let $P = (v_1, v_2, \ldots, v_n)$ denote the plaintext message represented as a sequence of characters. Each character is mapped to a vertex in a path graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots v_n\}, E = \{v_i v_{i+1}; 1 \le i \le n\}$.

Definition 2: A total coloring assigns colors to both vertices and edges such that:

- Adjacent vertices receive different colors
- Adjacent edges receive different colors
- No edge shares the same color with its incident vertices

The colored graph is then represented as a matrix $M = (m_{ij})$, where:

$$m_{ij} = f(v_i),$$

$$m_{ij} = \begin{cases} f(e_{ij}), & if \ (v_i, v_j) \in E \\ 0, & otherwise \end{cases}$$

Encryption is performed using an invertible key matrix K, such that:

$$C = K \cdot M_1$$

Decryption is achieved by:

$$M_1 = K^{-1} \cdot C$$

3. Proposed Methodology

The fundamental step of the proposed method is to represent the given information in the form of a graph structure. Each character of the plaintext is considered as a vertex in the graph. Consecutive characters in the message are connected by edges, thereby forming a path graph. This representation preserves the sequential nature of the data. Next, a total coloring scheme is applied to the constructed path graph. In this process, colors are assigned to both vertices and edges such that no adjacent or incident elements share the same color. The vertex colors are determined based on the encoding table (for example, alphabetical positions), while the edge colors are assigned using the difference between the values of adjacent vertices. After assigning the total coloring, the graph is transformed into a matrix representation. Specifically, an adjacency matrix is developed in which diagonal entries represent vertex colors and off-diagonal entries correspond to edge colors. This matrix captures both structural and coloring information of the graph. Subsequently, a suitable invertible key matrix is selected. The matrix obtained from the total-colored graph is then multiplied by this key matrix to produce a new matrix. This resulting matrix represents the encrypted form of the original information. At the receiver's end, the decryption process is carried out by applying the inverse of the key matrix. The original matrix corresponding to the total-colored graph is recovered, and the plaintext is obtained by decoding the vertex values using the encoding table.

Thus, the proposed algorithm integrates graph representation, total coloring, and matrix operations to achieve a secure and efficient encryption mechanism.

3.1 Encryption Algorithm

1. Add a distinct symbol to represent the beginning of the plaintext (for instance, let it be A).
2. Represent each character of the plaintext as a vertex and construct a graph from these vertices.
3. Connect each pair of consecutive vertices by edges to form a path graph corresponding to the given data sequence.
4. Using the encoding table, assign numerical values to each vertex. Then compute edge values based on the difference between the values of adjacent vertices.
5. Apply total coloring to the graph by assigning colors to both vertices and edges such that no adjacent or incident elements share the same color.
6. Construct the adjacency matrix of the colored graph, where diagonal entries represent vertex colors and off-diagonal entries represent edge colors. Denote this matrix by M_1.
7. Arrange the vertex values in the diagonal positions of the matrix M_1while preserving the order of characters in the plaintext.
8. Define a suitable invertible key matrix K for encryption.

9. Multiply the matrix M_1 with the key matrix K to obtain a new matrix C.

10. The resulting matrix C represents the ciphertext of the given message.

11. Finally, transmit the encrypted information in the form of matrix C (and, if required, the coloring pattern or encoding scheme) to the receiver.

3.2 Decryption Algorithm

1. Receive the encrypted matrix C along with the necessary information regarding the encoding scheme (if shared).

2. Obtain the inverse of the shared key matrix K, denoted as K^{-1}.

3. Multiply the encrypted matrix C by the inverse key matrix K^{-1} to recover the original matrix:

$$M_1 = K^{-1} \times C$$

4. From the matrix M_1, identify the diagonal elements which represent the vertex values corresponding to the encoded characters.

5. Extract the off-diagonal elements to obtain the edge values, which reflect the relationships between consecutive characters.

6. Using the total coloring structure, reconstruct the original path graph by assigning the extracted vertex and edge values appropriately.

7. Decode the vertex values using the predefined encoding table to retrieve the corresponding characters.

8. Arrange the decoded characters in the correct sequence based on the path graph structure.

9. Remove the initial special symbol (used to indicate the beginning of the message) if it was introduced during encryption.

10. The resulting sequence of characters gives the original plaintext message.

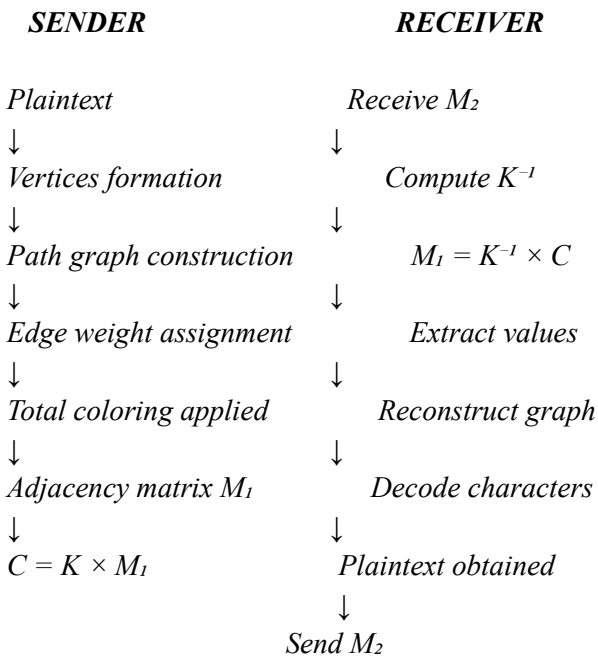

Fig 1: Steps in sender and receiver algorithm

3.3 Security Analysis

The security of the proposed cryptographic scheme relies on both structural complexity and key-based transformation. The use of total coloring introduces a dual-layer encoding mechanism, where both vertex and edge relationships must be correctly interpreted to recover the original message.

3.4 Key space analysis: The security strength depends on the selection of the invertible key matrix K. For an $n \times n$ matrix, the number of possible invertible matrices grows exponentially, making brute-force attacks computationally infeasible for sufficiently large n.

3.5 Resistant to structural attacks: Unlike traditional graph labeling methods, the proposed approach encodes information in both vertices and edges. This increases ambiguity for an attacker attempting to reconstruct the graph without knowledge of the coloring scheme.

3.6 Known plaintext attack: Even if partial plaintext information is known, recovering the key matrix requires solving a system of linear equations, which becomes increasingly difficult as matrix size increases.

3.7 Computational complexity: The efficiency of the proposed encryption scheme depends on both graph construction and matrix operations. The construction of the path graph from the plaintext requires a single pass through the input sequence, resulting in a time complexity of $O(n)$, where n denotes the number of characters in the message. Similarly, the total coloring process can be performed in linear time due to the simple structure of path graphs.

The dominant computational cost arises from the matrix multiplication involved in the encryption phase. For an $n \times n$ matrix, standard matrix multiplication requires $O(n^3)$ time. However, this cost can be reduced using optimized algorithms such as Strassen's method or parallel computation techniques.

Therefore, the overall complexity of the proposed scheme is primarily influenced by the matrix operation, while the graph-based preprocessing remains efficient and scalable.

Method	Technique	Limitation
Graph-Labeling	Vertex/Edges values	Single layer encoding
Adjacency matrix	Matrix only	Low structural complexity
Proposed method	Total coloring +Matrix	Higher complexity, better security

Table 1: comparison table

4. Illustration

Consider the message **"DATA"**, which is to be securely transmitted from sender to receiver using the proposed method.

Step 1: Conversion of Data into Graph

The given message is first represented in graphical form by assigning each character to a distinct vertex. This transformation helps in capturing the structure of the data in a systematic way.

Fig 2: Representation of the message "DATA" as individual vertices corresponding to each character.

Step 2: Construction of Path Graph

The vertices are then connected sequentially based on the order of characters in the message. This forms a path graph, ensuring that the original sequence of the data is preserved.

Fig 3: Path graph illustrating the sequential connection of vertices for the message "DATA"

Step 3: Encoding of Vertices

Each character is assigned a numerical value using a standard encoding scheme (A = 1, B = 2, …, Z = 26). Accordingly, the message is converted into numerical form as D = 4, A = 1, T = 20, and A = 1.

A	B	C	D	E	F	G	H	I	J	K	L	..	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	..	20	21	22	23	24	25	26

Table 2: Table to encode Alphabets

Fig 4: Path graph with encoded numerical values assigned to each vertex

Step 4: Edge Value Assignment

Weights are assigned to edges based on the difference between the values of adjacent vertices. This step introduces additional structure into the graph by capturing relationships between consecutive characters.

Distance among D and A = Code(A) – Code(D)

= 1-4 = -3

Distance among A and T = Code(T) – Code(A)

= 20-1 = 19

Distance among T and A = Code(A) – Code(T)

= 1-20 = -19

Fig 5: Weighted path graph showing edge values derived from differences between adjacent vertices

Step 5: Application of Total Coloring

A total coloring scheme is applied to the graph by assigning colors to both vertices and edges. The coloring is done in such a way that no adjacent or incident elements share the same color, ensuring proper distinction. A valid coloring can be:

Fig 6: Total colored path graph where vertex colors alternate and edges are assigned a distinct color satisfying total coloring conditions

(Vertices are colored alternately with 1 and 2, and edges are assigned color 3,4.)

Step 6: Matrix Representation

The colored graph is then converted into an adjacency matrix. In this matrix, diagonal elements represent vertex values, while off-diagonal elements represent edge weights.

$$M_1 = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & 1 & 19 & 0 \\ 0 & 19 & 20 & -19 \\ 0 & 0 & -19 & 1 \end{bmatrix}$$

Step 7: Encryption

An invertible key matrix K is selected, and the encryption process is performed by multiplying K with the matrix M_1. The result is the ciphertext matrix C, which is transmitted to the receiver.

$$C = K \times M_1$$

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad M_1 = \begin{bmatrix} 4 & -3 & 0 & 0 \\ -3 & 1 & 19 & 0 \\ 0 & 19 & 20 & -19 \\ 0 & 0 & -19 & 1 \end{bmatrix}$$

$$C = K \times M_1 = \begin{bmatrix} 1 & 17 & 20 & 18 \\ -3 & 20 & 20 & -18 \\ 0 & 19 & 1 & -18 \\ 0 & 0 & -19 & 1 \end{bmatrix}$$

This matrix $C$is transmitted to the receiver.

Step 8: Decryption

At the receiver's end, the inverse of the key matrix is applied to recover the original matrix. The diagonal entries are then decoded back into characters using the encoding table.

$$M_1 = K^{-1} \times C$$

$$K^{-1} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad M_1 = \begin{bmatrix} 1 & 17 & 20 & 18 \\ -3 & 20 & 20 & -18 \\ 0 & 19 & 1 & -18 \\ 0 & 0 & -19 & 1 \end{bmatrix}$$

$$M_1 = K^{-1} \times C = \begin{bmatrix} 4 & -3 & 0 & 36 \\ -3 & 1 & 19 & 0 \\ 0 & 19 & 20 & -19 \\ 0 & 0 & -19 & 1 \end{bmatrix}$$

The computational efficiency of the proposed method depends on both graph construction and matrix operations. The formation of the path graph and total coloring process require linear time $O(n)$, where n is the number of vertices (characters). However, the encryption and decryption steps involve matrix multiplication, as defined by $C = K \cdot M_1$ and $M_1 = K^{-1} \cdot C$, which require $O(n^3)$ time using standard algorithms.

The recovered values 4,1,20,1 correspond to the message **"DATA"**, confirming successful decryption.

5. Conclusion and work in future

This study introduced a graph-based encryption framework grounded in the concept of total coloring of path graphs. By combining graph-theoretic representation with matrix-based transformations, the proposed method establishes a dual-layer encoding mechanism in which both vertices and edges contribute to the encrypted structure. This integrated approach enhances the structural complexity of the ciphertext when compared with conventional graph labeling techniques that typically rely on a single layer of encoding.

From a computational perspective, the scheme remains efficient for practical implementation, particularly in environments where structured data representation is advantageous. The use of total coloring imposes stricter assignment

constraints, which in turn strengthens the encoding process by increasing interdependencies among graph components. In addition, the reliance on a symmetric key model ensures operational simplicity, although the overall security of the system is closely tied to the appropriate selection and management of the key matrix.

The framework also offers flexibility for implementation using widely adopted programming platforms such as C++, Java, and Python, supporting its applicability in real-world scenarios. At the same time, the current model presents opportunities for further refinement. For instance, extending the approach to more complex graph classes, including cyclic, tree-based, and hybrid structures, may provide additional layers of security and structural diversity.

Another direction for improvement lies in optimizing the size of the generated ciphertext matrix. Reducing storage requirements and transmission overhead would enhance the practicality of the method, particularly for large-scale data applications. Segment-based encryption strategies, where longer messages are divided and processed independently, may offer a feasible solution in this regard.

Future investigations could also consider integrating the proposed framework with public key cryptographic models to strengthen security guarantees and broaden its applicability. Additionally, optimizing matrix operations through advanced algorithms, as well as exploring parallel or distributed computing techniques, may significantly improve performance when handling large datasets.

In summary, the results of this work indicate that total coloring provides a promising foundation for the development of graph-based cryptographic systems. The proposed scheme demonstrates that incorporating structural and combinatorial properties of graphs can lead to encryption mechanisms that are both flexible and robust, making them suitable for evolving communication requirements.

Reference

1. Amudha, P., Jayapriya, J., & Gowri, J. (2021). *An algorithmic approach for encryption using graph labeling*. Journal of Physics: Conference Series, 1770.
2. Gallian, J. A. (2015). A dynamic survey of graph labeling. *Electronic Journal of Combinatorics*.
3. Hill, L. S. (1929). Cryptography in an Algebraic Alphabet, *American Mathematical Monthly*, 36(6), 306–312.
4. Kahate, A. (2011). *Cryptography and network security*.
5. MacDougall, J. A., Miller, M., Slamin, & Wallis, W. D. (2002). Total labeling of graphs.
6. Shanmugam, A., & Durairaj, M. (2018), Graph Theory Based Encryption Techniques for Secure Communication, *International Journal of Pure and Applied Mathematics*, 119(15), 2345–2354.
7. Singh, S., & Sharma, V. (2020), Applications of Graph Labeling in Cryptography, *Journal of Discrete Mathematical Sciences and Cryptography*, 23(4), 925–938.
8. Schneier, B. (1996), *Applied Cryptography* (2nd ed.). Wiley.
9. West, D. B. (2001). *Introduction to graph theory*.