Global Integrated Mathematics

https://gim.gospub.com/gim

Global Open Share Publishing

Article

Applications of Graph Transformations in Cryptography: A Secure Encoding Framework for Data Communication

Nasir Ali[1,*], Muhammad Touseef Haider[1], Mazhar Ahmad[1], Muhammad Irfan[1], Muhammad Imran Qureshi[1], Hafiz Muhammad Afzal Siddiqui[2]

[1]Department of Mathematics, COMSATS University Islamabad, Vehari Campus, Vehari, Pakistan

[2]Department of Mathematics, COMSATS University Islamabad, Lahore Campus, Lahore, Pakistan

[*]Corresponding author: Nasir Ali, nasirzawar@gmail.com

Abstract

In this paper, we introduce a novel and decentralized cryptographic approach that innovatively integrates concepts from graph theory specifically, the transformation of cycle graphs into path graphs alongside the strategic application of adjacency matrices, to ensure robust and secure data communication. The proposed method efficiently encodes textual data into graph-based representations, offering the capability to encrypt both individual words and complete sentences. This encoding not only preserves the original semantic structure but also enhances adaptability for various data types and formats commonly encountered in digital systems. A core component of this technique is its matrix-based encryption mechanism, which provides a highly secure framework resistant to a wide range of classical cryptographic attacks, including brute-force, known-plaintext, and frequency analysis methods. The encryption and decryption processes leverage the structural properties of graph matrices to conceal information effectively, ensuring confidentiality and integrity throughout the transmission. To substantiate the strength and reliability of the approach, we conduct both theoretical analysis and practical implementation, demonstrating that the scheme offers superior data protection and resilience against unauthorized access. Additionally, this work presents a fresh perspective on cryptographic systems by fusing graph theoretical principles with modern encryption techniques. This interdisciplinary synergy opens new pathways for solving contemporary cybersecurity problems, offering a promising direction for the development of future secure communication protocols in various real-world applications.

Keywords

Cyclic graph, Path graph, Adjacency matrix, Matrix encryption

Article History

Received: 30 May 2025 Revised: 29 July 2025

Accepted: 08 August 2025 Available Online: 03 September 2025

1. Introduction

Cryptography is the science of communicating and providing information in such a way that unauthorized access or alteration of that information is prevented. Cryptography is based on algorithms which transform data into unreadable format and guarantee (confidentiality, integrity, and authenticity). Today's connected world relies increasingly on the use of digital communication and exchange, especially of data and information, and cryptography has become essential to ensuring this information is being handled securely, particularly when it concerns financial transactions, a personal identity, and classified data owned by government [1]. Technologies like online banking, e-commerce, secure messaging, are all built on the assumption that they can take place in a safe and secure environment, and that includes both stopping cyber threats such as hacking, identity theft and breach of data [2].

Cryptography and network security have learnt very much from graph theory. Recent researches have investigated different ways in which even graphing algorithms may be used to safeguard communication networks, and particularly encryption methods [3]. One area which has been addressed to a large extent, is the application of complete graphs in encryption, providing new means for data protection [4]. Graph based encryption techniques also incorporate self-invertible matrices, creating these matrixes special to cryptographic systems [5,6]. These matrices are useful when the graph structures are complete graph structures for building encryption methods that are durable against different types of attacks [7,8]. Results of new solutions to protecting knowledge graph and other complex data forms [9] have arisen from adaption of graph theory with the ordinary cryptographic approaches. Graph based crypto is now largely being implemented in python, and therefore, it has become much easier for researchers to develop, and analyse new encryption techniques. Graph theory is used in modern cryptography, as basic encryption is not enough, they use this to assist with network security protocols, and other advanced encryption standards [10]. Moreover, graph labeling is being employed in recent cryptographical research to explore methods for encryption [11], making graph theory less applicable to cryptography and more generally applicable. For more applications readers are encouraged to read [12-19]. Cryptography has been the cornerstone for many years in securing trust in and privacy of the digital age and will continue to play a crucial role in solving ever evolving cybersecurity challenges to withstand threats.

In this research, we propose a novel cryptographic method using transformations from cycle graphs to path graphs with adjacency matrix techniques in the theory of graphs. Instead, this approach encodes data in graph structures that transparently encrypt each word as well as entire sentences, allowing the flexibility to deal with a variety of data formats. The methodology takes advantage of the structure of these graph representations and achieves strong encryption and decryption through matrix-based operations. This approach is shown to be effective both in theoretical analysis and practical experiments to enhance data security [20-22]. The results show that cycle graph transformations and adjacency matrices can lend themselves to cryptographic applications beyond their use in networks, adding a new and important perspective on secure data communication in modern networks.

A graph-based encryption technique provides a novel approach to secure communication by using the properties of graphs such as connectivity, structure and symmetry [23]. All these methods are resistant to classical cryptographic attacks and secure, while keeping design flexibility. This paper claims a contribution to this growing field by presenting algorithms based on the graph theory and on the matrix operations which provide the robust cryptographic solutions [24-27]. Besides being theoretically meaningful, these techniques have respective practical consequences for modern cryptography. This study integrated abstract mathematical concepts to advance cryptographic capability, while illustrating how real-world security challenges may be addressed. Some studies can be seen in [28-31].

1.1 Novelty

A revolutionary new cryptographic technique based on cyclic graph transformations and adjacency matrices allows the application of the periodic symmetry and adjacency properties of cyclic graphs to modern cryptography. Through the integration of graph transformations with matrix operations this method introduces a new dimension of complexity but sets up a secure framework which is resistant to classical cryptographic attacks. In addition to its theoretical contributions to graph- based cryptography, it is also of practical importance for providing secure communication systems. This approach simply connects the dots between abstract mathematical principle and real encryption challenge, thereby solving modern cryptographic needs using a scalable and innovative approach.

1.2 Problem Statement

As an innovative method of modern cryptography that addresses the practical problem of encrypting long messages that are often computationally expensive and susceptible to attack, an encryption method based upon cyclic graph transformation and adjacency matrices boasts the power of addressing shortcomings of multiple approaches in the area. Traditional encryption methods simply aren't feasible when large volumes of data are involved: they are constantly susceptible to attacks.

This problem is solved by the proposed technique by making use of the periodic symmetry and adjacency properties of cyclic graphs, which enables efficient encryption of long messages without degrading security. This method implements the conversion of complex graph relationships into secure and efficient cryptographic keys by integrating adjacency matrices to convert characters to numerical values from ASCII tables. Moreover, this approach is also helpful to the

advancement of modern cryptography and complements the process of encryption at the same time as it fortifies the encryption process and enables scalability to support security needs as the demand evolves.

The proposed encryption technique fills important gaps in traditional cryptography and it is particularly timely in the presence of large datasets, which are often computationally expensive and slow to encrypt. It implements high security, and efficient encryption, through the use of cyclic graph transformations and adjacency matrices. Even more efficient is the use the ASCII table to convert text to numerical values so that the process can function even faster. This method has good scalability, suited for large number of data, securely. That's the kind of thing you need, a robust, fast encryption solution, and it's better than what you've got. Together with the increasingly demanding needs for cryptography, its adaptability makes its use a secure and efficient approach to meeting modern cryptographic demands in the real world.

1.3 Preliminaries

We deal with improvements to encryption with the application of graph theory and algebraic concepts to these structures. First, we need to go back to some basic concepts on graph theory. Basic definitions are taken from source [19].

1.3.1 Graph

A graph $G = (V,E)$ consists of a set of vertices V and a set of edges E, where each edge connects two vertices, representing relationships between entities.

1.3.2 Cyclic Graph

A graph is cyclic if it has a cycle, a cycle is a path on the vertices for which we start and end at the same vertex, without repeating other vertices or edges. For example, in a graph with vertices $V = \{1,2,3\}$ and edges $E = \{(1,2),(2,3),(3,1)\}$ a cycle exists: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$.

1.3.3 Path Graph

A path graph is a graph where all the vertices and edges can be draw on a single line.

1.3.4 Adjacency Matrix

An adjacency matrix is a square matrix used to represent a graph, where each element a_{ij} is 1 if there's an edge between vertices v_i and v_j, and 0 otherwise.

2. Methodology

In this section, we illustrate using examples, how our proposed encryption technique can be applied. The below are examples of how the technique makes the security and efficiency of data encryption.

2.1 Encryption

Let "Graph" be the original message. Given that it contains 5 characters, we shall create a C_5 cyclic graph and place these characters as its nodes.

2.1.1 Convert the Plaintext to ASCII Values and Label the Edges

Now we use ASCII table to get the numerical values for each letter. We will obtain:

$$G = 71,\ r = 114,\ a = 97,\ p = 112,\ h = 104$$

These appropriate number values should be used to label the nodes. We assign:

$$V_1 = 71,\ V_2 = 114,\ V_3 = 97,\ V_4 = 112,\ V_5 = 104$$

These edges can be labeled as:

$$e_1 = |V_1\text{-}V_2| = |71\text{-}114| = 43$$
$$e_2 = |V_2\text{-}V_3| = |114\text{-}97| = 17$$
$$e_3 = |V_3\text{-}V_4| = |97\text{-}112| = 15$$
$$e_4 = |V_4\text{-}V_5| = |112\text{-}104| = 8$$
$$e_5 = |V_5\text{-}V_1| = |104\text{-}71| = 33$$

2.1.2 Graph Construction and Adjacency Matrix

Now we make the corresponding cyclic graph which is given below in Figure 1.

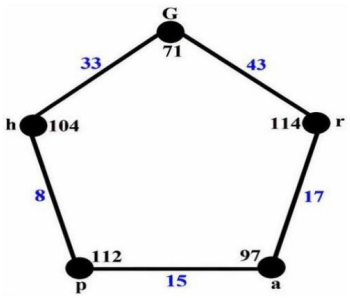

Figure 1. Cyclic graph derived from ASCII table to convert plain text.

Get a newly labeled adjacency matrix of the cyclic graph from above figure, which is denoted by M_1.

$$M_1 = \begin{bmatrix} 0 & 43 & 0 & 0 & 33 \\ 43 & 0 & 17 & 0 & 0 \\ 0 & 17 & 0 & 15 & 0 \\ 0 & 0 & 15 & 0 & 8 \\ 33 & 0 & 0 & 8 & 0 \end{bmatrix}$$

2.1.3 Path Construction and Updated Adjacency Matrix

Using the cyclic graph C_5, in Figure2, we obtain a path of length 5.

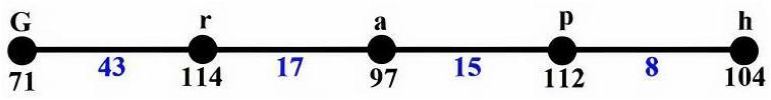

Figure 2. Path.

In the cyclic, take a matrix M_2. The cyclic graph matrix and the path matrix are both created using a similar method.

$$M_2 = \begin{bmatrix} 0 & 43 & 0 & 0 & 0 \\ 43 & 0 & 17 & 0 & 0 \\ 0 & 17 & 0 & 15 & 0 \\ 0 & 0 & 15 & 0 & 8 \\ 0 & 0 & 0 & 8 & 0 \end{bmatrix}$$

We modify the position numbers of the characters from ASCII Table by changing the diagonal elements in matrix M_2 from 0's to newly given values to the letters in the original message:

$$G = 71, r = 114, a = 97, p = 112, h = 104$$

We create a new matrix M_2^* (updated path matrix) by placing these values on the diagonal entries of matrix M_2 as illustrated below:

$$M_2^* = \begin{bmatrix} 71 & 43 & 0 & 0 & 0 \\ 43 & 114 & 17 & 0 & 0 \\ 0 & 17 & 97 & 15 & 0 \\ 0 & 0 & 15 & 112 & 8 \\ 0 & 0 & 0 & 8 & 104 \end{bmatrix}$$

2.1.4 Matrix Multiplication for Encryption

Construction of the new matrix M_3 as shown: $M_3 = M_1 \times M_2^*$

$$M_3 = \begin{bmatrix} 1849 & 4902 & 731 & 264 & 3432 \\ 3053 & 2138 & 1649 & 255 & 0 \\ 731 & 1938 & 514 & 1680 & 120 \\ 0 & 255 & 1455 & 289 & 832 \\ 2343 & 1419 & 120 & 896 & 64 \end{bmatrix}$$

Create the 'K' key matrix. Since there are 5 characters in the original message, the K matrix will be of order 5×5, as follows:

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

2.1.5 Cipher Matrix Construction

By multiplying Key Matrix by Matrix M_3, we generate Cipher Matrix C, $C = M_3 \times K$:

$$C = \begin{bmatrix} 1849 & 6751 & 7482 & 7746 & 11178 \\ 3053 & 5191 & 6840 & 7095 & 7095 \\ 731 & 2669 & 3183 & 4863 & 4983 \\ 0 & 255 & 1710 & 1999 & 2831 \\ 2343 & 3762 & 3882 & 4778 & 4842 \end{bmatrix}$$

Now this is an encrypted message. We send the Cipher matrix, key matrix and M_1 matrix to the receiver.

2.2 Decryption

The decryption process involves the following steps:

2.2.1 Matrix Construction from Encrypted Data

First, we compute the inverse of the key matrix K^{-1}.

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

Now obtain matrix M_3 by multiplying matrix C and the inverse of the key matrix K^{-1}, $M_3 = C \times K^{-1}$:

$$M_3 = \begin{bmatrix} 1849 & 4902 & 731 & 264 & 3432 \\ 3053 & 2138 & 1649 & 255 & 0 \\ 731 & 1938 & 514 & 1680 & 120 \\ 0 & 255 & 1455 & 289 & 832 \\ 2343 & 1419 & 120 & 896 & 64 \end{bmatrix}$$

2.2.2 Obtain the Original Matrix

To retrieve the original matrix, we first compute the inverse of matrix M_1.

$$M_1^1 = \begin{bmatrix} 68/21285 & 1/86 & -4/495 & -17/1290 & 1/66 \\ 1/86 & 495/1696 & 1/34 & -33/688 & -15/272 \\ -4/495 & 1/34 & 172/8415 & 1/30 & -43/1122 \\ -17/1290 & -33/688 & 1/30 & 187/3440 & 1/16 \\ 1/66 & -15/272 & -43/1122 & 1/16 & 215/2992 \end{bmatrix}$$

After multiplying M_1^1 by M_3, we obtain the final matrix M_2^*, $M_2^* = M_1^1 \times M_3$:

$$M_2^* = \begin{bmatrix} 71 & 43 & 0 & 0 & 0 \\ 43 & 114 & 17 & 0 & 0 \\ 0 & 17 & 97 & 15 & 0 \\ 0 & 0 & 15 & 112 & 8 \\ 0 & 0 & 0 & 8 & 104 \end{bmatrix}$$

2.2.3 Convert ASCII Values to Plaintext

Finally, we decode the values in M_2^* using ASCII Table: 71 = *G*, 114 = *r*, 97 = *a*, 112 = *p*, 104 = h.

Thus, the original message "Graph" is successfully retrieved.

2.3 Encryption

Let "I Play." be the original message. Given that it contain 7 characters, we shall create a C_7 cyclic graph and place these characters as its nodes.

2.3.1 Convert the Plaintext to ASCII Values and Label the Edges

Now we use ASCII table to get the numerical values for each letter. We will obtain: I = 73 , Space =32, P = 80, l = 108, a = 97, y = 121, dot = 46.

These appropriate number values should be used to label the nodes.

We assign $V_1 = 73$, $V_2 = 32$, $V_3 = 80$, $V_4 = 108$, $V_5 = 97$, $V_6 = 121$, $V_7 = 46$. These edges can be labeled as,

$$e_1 = |V_1\text{-}V_2| = |73\text{-}32| = 41$$
$$e_2 = |V_2\text{-}V_3| = |32\text{-}80| = 48$$
$$e_3 = |V_3\text{-}V_4| = |80\text{-}108| = 28$$
$$e_4 = |V_4\text{-}V_5| = |108\text{-}97| = 11$$
$$e_5 = |V_5\text{-}V_6| = |97\text{-}121| = 24$$
$$e_6 = |V_6\text{-}V_7| = |121\text{-}46| = 75$$
$$e_7 = |V_7\text{-}V_1| = |46\text{-}73| = 27$$

2.3.2 Graph Construction and Adjacency Matrix

Now we make the corresponding cyclic graph which is given below in Figure 3.

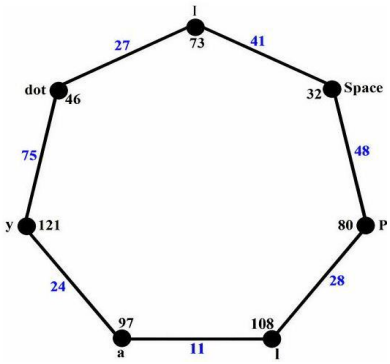

Figure 3. Cyclic graph.

Get a newly labeled adjacency matrix of the cyclic graph, which is denoted by M_1.

$$M_1 = \begin{bmatrix} 0 & 41 & 0 & 0 & 0 & 0 & 27 \\ 41 & 0 & 48 & 0 & 0 & 0 & 0 \\ 0 & 48 & 0 & 28 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 24 & 0 & 75 \\ 27 & 0 & 0 & 0 & 0 & 75 & 0 \end{bmatrix}$$

2.3.3 Path Construction and Updated Adjacency Matrix

Using the cyclic graph C_7, in Figure 4 we obtain a path of length 7.

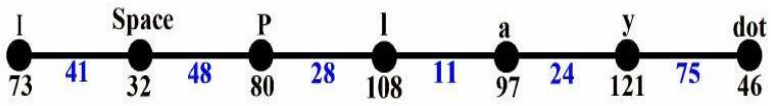

Figure 4. Path.

In the cyclic, take a matrix M_2. The cyclic graph matrix and the path matrix are both created using a similar method.

$$M_2 = \begin{bmatrix} 0 & 41 & 0 & 0 & 0 & 0 & 0 \\ 41 & 0 & 48 & 0 & 0 & 0 & 0 \\ 0 & 48 & 0 & 28 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 0 & 24 & 0 \\ 0 & 0 & 0 & 0 & 24 & 0 & 75 \\ 0 & 0 & 0 & 0 & 0 & 75 & 0 \end{bmatrix}$$

We modify the position numbers of the characters by changing the diagonal elements in matrix M_2 from 0 's to newly given values to the letters in the original message: $I = 73$, $Space = 32$, $P = 80$, $l = 108$, $a = 97$, $y = 121$, $dot = 46$.

We create a new matrix M_2^* (updated path matrix) by placing these values on the diagonal entries of matrix M_2 as illustrated below:

$$M_2^* = \begin{bmatrix} 73 & 41 & 0 & 0 & 0 & 0 & 0 \\ 41 & 32 & 48 & 0 & 0 & 0 & 0 \\ 0 & 48 & 80 & 28 & 0 & 0 & 0 \\ 0 & 0 & 28 & 108 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 97 & 24 & 0 \\ 0 & 0 & 0 & 0 & 24 & 121 & 75 \\ 0 & 0 & 0 & 0 & 0 & 75 & 46 \end{bmatrix}$$

2.3.4 Matrix Multiplication for Encryption

Construction of the new matrix M_3 as shown: $M_3 = M_1 \times M_2^*$,

$$M_3 = \begin{bmatrix} 1681 & 1312 & 1968 & 0 & 0 & 2025 & 1242 \\ 2993 & 3985 & 3840 & 1344 & 0 & 0 & 0 \\ 1968 & 1536 & 3088 & 3024 & 308 & 0 & 0 \\ 0 & 1344 & 2240 & 905 & 1067 & 264 & 0 \\ 0 & 0 & 308 & 1188 & 697 & 2904 & 1800 \\ 0 & 0 & 0 & 264 & 2328 & 6201 & 3450 \\ 1971 & 1107 & 0 & 0 & 1800 & 9075 & 5625 \end{bmatrix}$$

Create the 'K' key matrix. Since there are 7 characters in the original message, the K matrix will be of order 7×7, as follows:

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

2.3.5 Cipher Matrix Construction

By multiplying Key Matrix by Matrix M_3, we generate Cipher Matrix C, $C = M_3 \times K$:

$$C = \begin{bmatrix} 1681 & 2993 & 4961 & 4961 & 4961 & 6986 & 8228 \\ 2993 & 6978 & 10818 & 12162 & 12162 & 12162 & 12162 \\ 1968 & 3504 & 6592 & 9616 & 9924 & 9924 & 9924 \\ 0 & 1344 & 3584 & 4489 & 5556 & 5820 & 5820 \\ 0 & 0 & 308 & 1496 & 2193 & 5097 & 6897 \\ 0 & 0 & 0 & 264 & 2592 & 8793 & 12243 \\ 1971 & 3078 & 3078 & 3078 & 4878 & 13953 & 19578 \end{bmatrix}$$

Now this is an encrypted message. We send the Cipher matrix, key matrix and M_1 matrix to the receiver.

2.4 Decryption

The decryption process involves the following steps:

2.4.1 Matrix Construction from Encrypted Data

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

Now obtain matrix M_3 by multiplying matrix C and the inverse of the key matrix K^{-1}. $M_3 = C \times K^{-1}$:

$$M_3 = \begin{bmatrix} 1681 & 1312 & 1968 & 0 & 0 & 2025 & 1242 \\ 2993 & 3985 & 3840 & 1344 & 0 & 0 & 0 \\ 1968 & 1536 & 3088 & 3024 & 308 & 0 & 0 \\ 0 & 1344 & 2240 & 905 & 1067 & 264 & 0 \\ 0 & 0 & 308 & 1188 & 697 & 2904 & 1800 \\ 0 & 0 & 0 & 264 & 2328 & 6201 & 3450 \\ 1971 & 1107 & 0 & 0 & 1800 & 9075 & 5625 \end{bmatrix}$$

2.4.2 Obtain the Original Matrix

To retrieve the original matrix, we first compute the inverse of matrix M_1^{-1}

$$M_1^{-1} = \begin{bmatrix} -275/10332 & 1/82 & 275/12096 & -6/287 & -25/432 & 11/1148 & 1/54 \\ 1/82 & -63/11275 & 1/96 & 108/11275 & -7/264 & -9/2050 & 7/825 \\ 275/12096 & 1/96 & -11275/580608 & 1/56 & 1025/20736 & -11/1344 & -41/2592 \\ -6/287 & 108/11275 & 1/56 & -1296/78925 & 1/22 & 54/7175 & -4/275 \\ -25/432 & -7/264 & 1025/20736 & 1/22 & -7175/57024 & 1/48 & 287/7128 \\ 11/1148 & -9/2050 & -11/1344 & 54/7175 & 1/48 & -99/28700 & 1/150 \\ 1/54 & 7/825 & -41/2592 & -4/275 & 287/7128 & 1/150 & -287/22275 \end{bmatrix}$$

After multiply M_1^1 by M_3, we obtain the final matrix M_2^*: $M_2^* = M_1^1 \times M_3$

$$M_2^* = \begin{bmatrix} 73 & 41 & 0 & 0 & 0 & 0 & 0 \\ 41 & 32 & 48 & 0 & 0 & 0 & 0 \\ 0 & 48 & 80 & 28 & 0 & 0 & 0 \\ 0 & 0 & 28 & 108 & 11 & 0 & 0 \\ 0 & 0 & 0 & 11 & 97 & 24 & 0 \\ 0 & 0 & 0 & 0 & 24 & 121 & 75 \\ 0 & 0 & 0 & 0 & 0 & 75 & 46 \end{bmatrix}$$

2.4.3 Convert ASCII Values to Plaintext

Finally, we decode the values in M_2^* using ASCII Table: 73 = I, 32 = space, 80 = P, 108 = l, 97 = a, 121 = y, 46 = dot.

Thus, the original message "I Play." is successfully retrieved.

3. Critical Analysis

In this paper we propose cyclic graph transformations for an new encryption algorithm based on adjacency matrix, path construction and ASCII values ciphers. Due to the symmetry and structure of cyclic graph, the technique is especially well suited for the purpose of secure communication. We use adjacency matrices to represent the graph, and utilize the path construction process to effect efficient transformations during encryption and decryption. All characters fall on to their ASCII values, which is what further strengthens the encryption process. The complexity and randomness of the graph is due to the cyclic nature there by making the cryptographic security robust. The algorithm is also thoroughly analyzed economically in terms of resistance against different cryptographic algorithms to ensure the transmitted data's confidentiality and integrity. This method achieves an optimal tradeoff between security and computational performance, and thus is a promising solution to secure network communication in real applications.

3.1 Performance Review

Based on the proposed encryption technique using cyclic graph transformations and adjacency matrices, the solutions for secure communication become the best possible. These methods improve data security by exploiting the special algebraic properties favorable in cyclic graphs. Knowledge of adjacency matrices promotes security by facilitating efficient transformation during encryption and decryption operation. What this method brings is high complexity and randomness, which renders it immune to cryptographic attacks. Finally, our proposed encryption is scalable, able to process very large datasets whilst remaining efficient and stable. We now talk about advantages and limitations of this encryption and decryption methods.

3.2 Advantages and Drawbacks

In this, we will discuss the advantages and drawbacks of the proposed encryption and decryption schemes, Table 1 and Table 2 outlines the facts.

Table 1. Advantages of proposed encryption technique.

Advantage	Description
Improved Security	Using the natural structure of cyclic graphs to make some of their problems complex and random, we introduce complexity and randomness into our proposed technique that increases security by resisting common cryptographic attacks.
Expandability	Efficient scaling with larger datasets is achieved given by the technique's ability to increase expandability by utilizing cyclic graph transformations.
Efficient Resource Utilization	Also this technique combines cyclic graph transformations with adjacency matrices to achieve efficient resource use, in that no memory is wasted due to strong encryption.
Versatility and Robustness	The technique shows adaptability to a variety of encryption scenarios while being resilient to strong security and performance under a broad range of conditions.
Simplified Implementation	The technique benefits from the use of cyclic graphs and adjacency matrices, resulting in a straightforward implementation easier to add to existing secure communication systems.

Table 2. Drawbacks of proposed encryption technique.

Drawback	Description
Computational Overhead	Cyclic graph transformations may be more computationally expensive at encryption and decryption than more simple transformations.
Key Management Complexity	The algorithm's dependence on graph structures makes it more challenging than some of the other algorithms out there to manage and securely distribute the keys necessary for use of the algorithm.
Memory Requirements	The method is efficient but large-scale graph data structures might require lots of memory resources especially in case of very large input datasets.
Limited Standardization	The encryption method is new, and therefore the interoperability with existing cryptographic systems is limited, and as it is not widely supported and standardized, it may not be widely supported and not be supported at all.
Dependency on Graph Size	In real time communication scenarios the performance of the algorithm may deteriorate with the graph size.

3.3 Execution Efficiency

The execution efficiency of the proposed encryption technique must be assessed, it is especially important for real time applications as speed and resource usage are critical. In this study, the performance of a method that encrypts using cyclic graph transformations and adjacency matrices is tested in order to establish that the method can be securely used in communication.

The cyclic graph C_n based encryption technique proposed starts with encoding plaintext in ASCII table values. The adjacency matrix of the cyclic graph's numeric values are encrypted and decrypted. Effective transformations based on graph's inherent symmetry and structure enableIntroduction of randomness and complexity, enhanced security. Computational effort is required for matrix operations on the adjacency matrix but they are very important to the strength of the cryptography, although the resulting ciphertext is resilient against cryptographic attacks. Although there is a computational cost, the technique is efficient to the point of maintaining a tradeoff between high security and performance. You could say it is flexible; that is, it is able to work on different data sizes and communication scenarios, which make it a perfect solution for real-time applications, where speed and optimization of resources are essential. Overall, this technique is a scalable, secure, computationally efficient encryption technique and is hence a desired choice for secure communication systems.

3.4 Key governance Mechanisms

Analyze intrinsically a cryptography field with strong tie to good key management. The entire process includes key generation, exchange, storage, usage and finally decommissioning. Robust key management is especially critical for the particular characteristics of graph-based methodologies, in the context of graph-based encryption algorithms. Thus, secure and efficient key management remains a challenge, as these algorithms must balance (at the same time) data integrity security and speed.

3.4.1 Key Generation

Through the use of adjacency matrix techniques, transformations from cyclic graphs to path graphs are proposed as a method of key management in our proposed encryption technique. It ensures that keys are derived in a way reflecting graph transformations unique structure. The combinatorial properties of the cyclic and path graphs are used to handle key generation, storage, utilization, and decommissioning. We focus on integrity and structure of these graphs so that our key management process remains both secure and efficient when applied to graph-based encryption methods.

3.4.2 Key Transfer Protocols

Key control protocols are designed for our proposed encryption technique to ensure secure and efficient handling of cryptographic keys over the life cycle. Our cryptographic method and resulting protocols are founded around transformations from cyclic graphs into path graphs through use of adjacency matrix techniques. These graph transformations are closely tied to the key generation process tying it to high entropy and randomness for stronger security. Secure channels, such as Diffie-Hellman key exchange, or public key infrastructure (PKI), is used to allow safe key transmission from unauthorized parties in the key exchange method. Encrypting the keys is an important thing that is done with respect to key storage, either your symmetric encrypt encryption (e.g AES) or with hardware security modules (HRMs). Access control measures as leverage by the entity to ensure only authorized entities can do encryption and decryption with the keys during key utilization. Key decommissioning is the last but concludes with secure destruction protocols, keys are totally obliterated when not required and that this prevents misuse. These key control protocols, merged with our cryptography mechanism, are vital to ensure secure and efficient graph-based encryption system.

3.4.3 Access and the Key Storage

One of our proposed cryptographic technique's key concepts is that of key storage and access control that assures for the security and integrity of the cryptographic system. The keys created from the transformation from cyclic to path graphs are securely stored via such encryptions as symmetric encryption (Symmetric encryption e.g. AES) or hardware security module (HSM). Access control policies are put in place to limit key access so that only permitted entities, like the keys will have access to retrieve and use keys. To do this, we employ multi factor authentication (MFA) and role-based access control (RBAC) so only those people or processes that need access to the keys will have the allowed privileges to work with or manage the keys.

3.4.4 Key Expiry and Revocation

Our graph-based encryption method relies on key security so revocation and expiry protocols are important to prevent keys that are available to user for use after they are retired. In either case, the system's security requirements set if keys have to expire after a predetermined amount of time or have been used a given number of times (whichever comes first). Securely revocable keys can be revoked in case of potential compromise (e.g. using a certificate revocation list (CRL) or online certificate status protocol (OCSP). In this way, the encryption system is never susceptible to any unauthorized decryption attempts and therefore can never be using any expired key.

3.5 Comparative Analysis with Existing Techniques

To demonstrate the exclusive advantages and innovation of our proposed graph theory based encryption technique we compare the proposed solution with existing one. For instance, Chinthaka Weerarathna et al. [32] entitled' A Novel Cryptosystem Using Multipartite Graphs, RMVV Bandara et al. [33] titled' Symmetric encryption using snake graphs and supermagic covering, Nasir Ali et al. [34] titled' Secure communication in the digital age: a new paradigm with graph-based encryption algorithms and Karrar Khudhair Obayes [35] titled' Text Encryption with Graph Theory Based Key Generation. While our approach delivers improved security and computational complexity, trading a higher level of security for a more secure encryption framework than can be afforded by simpler techniques.

The second contribution is that we provide a complete key management system along the key lifespan from generation, distribution, storage, and revocation for which such systems have not been addressed in previous research. The anecdotal studies underlying these concerns are related to secure key generation and distribution, whereas we present key policies tailored to the algebraic properties of our encryption and avoid the risk of key compromise. Our method also scales well and is computationally efficient, which makes it suitable for real time applications where fast encryption and decryption are critical. With this, we bridge the gap between the theoretical idea and the practical application, in a shape that increases security whilst being more efficient. Real world examples and case studies of the applicability of our encryption technique in the real world are shown, and are important contributions to the field of cryptographic research. See Table 3 for comparative analysis

Table 3. Comparative analysis.

Feature/Aspect	Proposed Scheme	Chinthaka Weerarathna et al. [32]	Nasir Ali et al. [34]	RMVV Bandara et al. [33]	Karrar Khudhair Obayes [35]
Graph Type	Cyclic graph	Multipartite graph	Corona, complete bipartite and star graph	Snake, wheel and cyclic graph	Minimum spanning tree graph
Key Handling	Securer and flexible key management	Graph-based key management	Key security comprehensive protocols	Secret key pair management	Kruskal's algorithm provides key security
Encryption Reliability	Graph-based structure optimization	Using graph decomposition to enhance complexity	Very high, due to complex graph structures	Robustness is improved by key randomness	Strong security with graph based key generation
Execution Effectiveness	Efficient resource utilization	Graph transformations for optimized encryption	Speed and resource analyzed for Efficiency	Optimize graph labeling techniques	Analyzed, an efficient graph based encryption algorithm
Implementation contexts	Protecting sensitive information	Graph-based encryption for secure communication	It includes practical examples and case studies	Secure communication and data security	Security of communication and data protection
Uniqueness	Innovative graph-based encryption	Unique transformation using graph theory	Introduced novel graph-based methods	New supermagic labeling for encryption	Minimum spanning trees as novel graph-based encryption
Security Robustness	A further defense against traditional breaches	Resistant to cryptanalytic attacks	Be more secure from conventional attacks	Graph randomness and labeling techniques enhanced security	Graph complexity enhanced security

3.6 Time Complexity

We discuss the time complexity of our proposed encryption and decryption process which involves graph transformations, matrix operations, and key management in this section.

3.6.1 Encryption Process

Here the encryption occurs at multiple steps sequentially. Passing through each step will provide greater security of data and reliability.

Convert Plaintext to ASCII Values: Each ASCII code can correspond to only one character. The time complexity of O(m) where m is of length of plaintext.

Cyclic Graph to Path Graph Transformation: We represent the plaintext as a cyclic graph, and then restructure it into a path graph through graph traversal, edge reassignment, or graph structure restructuring. The cyclic graph has an O(V + E) time complexity.

Key Generation: We generate a unique encryption key for the graph by its properties i.e., its adjacency matrix. It maybe requires moving through the graph and some matrix operations. Depending on whether or not matrix operations can be utilized plus, we are getting a time complexity of O(V + E) or $O(n^3)$ where n is the number of vertices.

Apply Key to Encrypt Data: This is applied generally from the graph structure or the matrix operations to the plaintext data to create encryption key. If matrix-based encryption is used the time complexity is $O(n^3)$.

Encrypted Output: After encryption key, data is made secured for transmission or storage in encrypted form ciphertext. Combining all the steps and the overall time complexity of the encryption process is: O(m) + O(V + E) + $O(n^3)$.

3.6.2 Decryption Process

Here the decryption occurs at multiple steps sequentially. Passing through each step will provide greater security of data and reliability.

Retrieve Encrypted Data: It gets you the encrypted ciphertext you can decrypt and the time complexity is $O(1)$, as retrieving ciphertext was supposed to involve constant time operations.

Apply Decryption Key: The corresponding graph transformation and key generation process is used to derive the decryption key to be used on the ciphertext. Reverse of graph transformation and matrix manipulation during encryption. The time complexity is $O(n^3)$ if the decryption process is done with the help of matrix operations.

Reverse Matrix and Graph Transformations: To restore the original plaintext, the original plaintext path graph then gets 'back reverse' transformed (cyclic graph to path graph) and the matrix inversion(s) are taken in reverse if they are applicable. The time complexity is $O(n^3)$ and it's like encryption because it has matrix inversions and graph traversal.

Retrieve Original Plaintext: The plaintext is recovered after applying the decryption key and canceling transformations and the time complexity is $O(m)$, and in the decryption case, reading or reconstructing the plaintext, since m is the length of the plaintext. The overall time complexity of the whole decryption process is: O(1) + $O(n^3)$ + $O(n^3)$ + O(m).

3.6.3 Defense Against Advanced Attacks

Through adjacency matrix transformations of cyclic graphs into path graphs we offer a robust defense against advanced attacks. This method accomplishes high entropy and randomness by using this structural complexity and combinatorial graph theory properties to be resistant of brute force and statistical attack. The security is boosted by the adjacency matrices, which tie a complex relationship between input data points, so that small differences between the input can lead to noticeable differences in the output. There are strong safeguards against differential or linear attacks, which are two most common cryptanalytical attacks, that are based on key properties, such as the difficulty of the reversal of graph transformations without the correct decryption key or reliance on unique graph structures. Taken together, these represent a cryptographic framework able to resist advanced and adaptive attacks in today's cybersecurity environment.

3.6.4 Quantum-Induced Threats

Breaking traditional cryptographic systems based on factorization or discrete logarithms, which can be thought of as primary threats from quantum computing, can be solved at speeds far beyond classical computing, with the caveat that quantum computing is not yet here. These conventional methods are particularly susceptible to quantum-based attack, such as Shor's algorithm. Still, our approach makes use of the finer structural properties of graphs through transformations of cyclic graphs into path graphs by adjacency matrices. To exploit those, it is too high the combinatorial complexity and shared randomness of these transformations and matrix operations. With this approach, we establish robust defense against quantum induced threats, and offer secure and resilient alternative to traditional cryptographic systems.

(1) Quantum-resilient features.

Structural Complexity: The use of graph transformations from cyclic graphs to path graphs results in structural complexity high enough for quantum algorithms such as Shor's and Grover's to not be easily exploited.

Combinatorial Security: As a combinatorial problem, adjacency matrices encode intricate relationships between graph vertices and quantum attacks themselves become exponentially intractable with graph size.

Dynamic Key Generation: The technique derives encryption keys from graph specific properties, resulting in high entropy and unguessability keys that are immune to quantum based key search algorithms.

Non-reversible Transformations: The reversal of the cyclic-to-path graph transformations is computationally intensive without the right decryption key, increasing the quantum cryptanalysis security of the design by another layer.

Scalability in Graph Size: Raising the size of the graph leads to a dramatic increase in complexity in the space of transformations and operations, amplifying to a degree the resistance to quantum computational power.

Randomized Operations: Randomized adjacency matrix operations are incorporated such that quantum attacks are unable to efficiently learn patterns or reduce the problem space.

(2) Future adaptations.

Further application of post quantum cryptographic methods such as the lattice based, hash based, or code based will greatly enhance the quantum computation resistant capability of this graph-based encryption technique. Integrating the strength of both graph transformations and post quantum cryptography, security is a level higher. It guarantees that quantum attacks will never spoil the technique, and that it can still operate well in classical cryptographic applications.

3.6.5 External Channel Attacks

Physical attack methods are related to external channel threats by their inherent coupling to the exploitation of unintended physical phenomena to obtain sensitive information from our encryption technique. Analysis of such subtle values as timing variations, fluctuation on energy consumption or electromagnetic radiation during the encryption process forms the basis of these methods. Adversaries may view the strength of the cryptographic system simply as a mechanism for observing and interpreting these external signals to try and compromise the security of encrypted data. Nevertheless, our encryption technique is carefully crafted to take measures which minimize such leakage so that such external channel threats are adequately dealt with.

(1) Cryptographic countermeasures.

Resistance to External Channel Attacks: Given mechanisms that prevent external channel attacks such that the execution time is independent of the data being processed, we employ our encryption technique. This uniformity is attained by graph-based encryption and cryptographic primitives based on graph structures which avoid leak of any timing correlation information. As a result, external monitoring techniques have no means to time the encryption process and hence risk of timing analysis is mitigated thanks to this constant time approach.

Power Consumption Obfuscation: Our encryption scheme is designed to defend against power analysis attacks, e.g. masking or hiding. The system can substantially resist power analysis by randomizing intermediate values to be used while encrypting, and by reducing correlation between power consumption and data being processed. These techniques also work to hide power consumption patterns from leaking critical information, meaning it is very hard for an attacker to use changes in power consumption to help break around encryption.

Electromagnetic Emission Obfuscation: Finally, we further strengthen the defense against external channel threat by discussing electromagnetic analysis (EMA) attacks. Shielding and differential power analysis (DPA) countermeasures are employed by the system to hide electromagnetic emissions. We introduce noise and extra layers of protection on the signals that are emitted during encryption operations and that attackers cannot derive valuable information from from electromagnetic emissions.

Randomized Execution Paths: Our encryption technique incorporates controlled randomness into the execution flow of the last level, to fight against timing and power attacks. This injects unpredictability, which makes it hard for an attacker to infer timing, or power consumed by the system. Noting that randomness disrupts any existing attack models, this randomness greatly improves cryptographic security of the technique against a variety of external channel threats. Table 4 elaborates the fact.

Table 4. Resistance to advanced attacks.

Feature/Aspect	Proposed Scheme
Post-quantum resilience	The graph structures are complex, the algebraic properties are intricate, and the ability to be integrated with post-quantum algorithms make these large
Side-Channel Attack Resistance	Through implementation of secure hardware, masking and hiding techniques, as well as using constant time algorithms
Encryption performance	The computational efficiency is achieved by using optimized graph-based transformations and matrix operations
Extensibility and responsiveness	A flexible design which can fit several key sizes, configurations and which is compatible with current cryptographic protocols.

Randomized Execution Paths: Our encryption technique incorporates controlled randomness into the execution flow of the last level, to fight against timing and power attacks. This injects unpredictability, which makes it hard for an attacker to infer timing, or power consumed by the system. Noting that randomness disrupts any existing attack models, this randomness greatly improves cryptographic security of the technique against a variety of external channel threats.

(2) Functional implementation aspects.

By using hardware components specifically designed to resist side channel attacks we improve the security of our encryption method. Because of these robust protections, these specialized hardware elements feature noise generation to mask data leakage or shielding to minimize electromagnetic emissions. The measures they offer are quite strong against power and electromagnetic analysis attacks. The security hardware also provides protection of cryptographic keys and operations against the unauthorized. The integration of these hardware-based safeguards greatly increase the overall robustness of the encryption system against advanced attack techniques. In addition, because they prevent extraction of sensitive information through physical channels, the hardware countermeasures guarantee the security of the encryption even in the presence of sophisticated adversaries. The multiple layer approach used to secure this system makes the encryption technique very resistant to real world threats.

3.7 Space Complexity

Here we discuss the space complexity of encryption and decryption process:

3.7.1 Encryption Process

In this we will discuss the space complexity of Encryption process:

Plaintext Representation: It works by converting the plaintext to ASCII values that takes O(m) space, so m is the length of the plaintext. Here represents the data that will be encrypted first.

Graph Representation: Often an adjacency matrix representation of a cyclic graph is used, requiring $O(n^2)$ space in which V is the number of vertices in the graph. We use this graph structure to perform graph- based transformations.

Key Generation: A matrix of size $n \times n$ storing graph properties, is used to generate the key in $O(n^2)$ space. For encryption process, this key is used.

Matrix Operations: Matrix manipulations (such as multiplication or inversion) are performed during encryption, so storing them results in $O(n^2)$ space complexity.

Encrypted Data Storage: After finally encryption it is stored, which means that it takes O(m) space where m is the length of the plaintext. By combining all steps the overall space complexity is: $O(m) + O(V^2) + O(n^2)$.

3.7.2 Decryption Process

Now we discuss the space complexity of the decryption process:

Retrieve Encrypted Data: Decryption takes place on a ciphertext of length m, and we recover the encrypted ciphertext of which space complexity is O(m).

Apply Decryption Key: We obtain the decryption key from the graph transformation, then apply it to the ciphertext to achieve the decryption and its space complexity is $O(n^2)$ for such a key matrix n vertices.

Reverse Matrix and Graph Transformations: The encryption is reversed, by performing the graph transformations from path graph to cyclic graph and the matrix inversions to $O(n^2)$, to store the inverse matrices and intermediate results.

Retrieve Original Plaintext: We recover the original plaintext after applying decryption key to undo the transformations, as well as recover the space complexity of O(m) for storing the decrypted plaintext.

So the overall space complexity of decryption process is: $O(m) + O(n^2) + O(n^2) + O(m) = 2O(m + n^2)$.

3.8 Real-Life Implementations and Case Studies

In industries across the globe, graph-based encryption effectively keeps data and allows it to move securely.

3.8.1 Operational Validation and Testing

This implementation and extensive testing in real world scenarios within a small organizational context confirm the validity, and practicality of the proposed technique. The application of this encryption technique is demonstrated with a series of detailed case studies, and its effectiveness and security in practical use is verified.

3.8.2 Government Database Security

Setup: They applied encryption technique to sensitive government databases that store classified information.

Application: The records, as well as personal information of citizens, were encrypted during storage and transmission.

Outcome: We found that all data was fully encrypted and no data leakage or breaches occurred. Strong decryption system was inherited to protect the government database from inspection as per security regulation.

3.8.3 IoT Devices Security

Setup: We implemented the encryption technique in a network of IoT devices under industrial condition.

Application: Sensitive data was protected by encryption when communication occurred between IoT devices.

Outcome: The use of this technique also ensured security communication between the devices while ensuring safekeeping of the important industrial data. The encryption proved invalid, validation was done from testing, no security incidents record during testing.

3.8.4 Voice Over IP (VoIP) Communications

Setup: Secure voice communication was added to the organization's VoIP system by integrating the proposed encryption technique. Application: The voice data transmitted over the network was encrypted before transmission so the communication within the system is secure. Outcome: And there wasn't so much as a noticeable latency or loss of voice quality, and the communication was clear and uninterrupted. Calls sent into the encryption system successfully decrypted and the recipient's calls were received without affecting the service quality thanks to the secure communication.

3.8.5 E-Commerce Transactions

Setup: The technique was integrated to an e-commerce platform to secure the online payments. Application: Rather, before that transaction goes over the network, customer payment information such as the credit card details is encrypted.

Outcome: It encrypted all transactions of e-commerce in a way which no one can intercept the sensitive payment information. During payment processing, the encryption technique allowed no unauthorized access.

3.8.6 Employee Authentication System

Setup: The encryption technique proposed was used in an employee authentication system for secure logins into the system.

Application: Password and personal identification were encrypted before being sent on the network for verification.

Outcome: It successfully authenticated employees without having to get exposed to sensitive data, or secure login and minimize potential of a data breach. Security and robustness of an encryption technique were demonstrated, and it was successfully tested with no vulnerabilities.

The proposed encryption technique, as shown through the above examples, not only improves the security of the total system, but also makes it very easy to implement across all practical realizations, achieving a high performance and ease of use.

4. Conclusion

In this research we introduced a new method for encrypting data through the use of cyclic graph transformations and adjacency matrices to offer data security. Our approach makes use of the properties of graphs and matrix operations to guarantee strong encryption, high computational efficiency, and resistance to classical attacks. The method proposed is effective both at encrypting and decrypting data while ensuring data scalability for large datasets. We show that graph-based cryptography allows for secure and efficient alternatives to traditional methods. This technique can be further refined for future research, and also looked at for integration into real world security systems.

Availability of Data and Materials

The data is provided on request to the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest and all agree to publish this paper under academic ethics.

Fundings

The current work was assisted financially to the Dean of Science and Research at King Khalid University via the Large Group Project under grant number RGP. 2/482/45.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (project number RGP .2/ 482/45. Academic year 1444H).

Author's Contribution

All the authors equally contributed towards this work.

Generative AI Statement

The authors declare that no Gen AI was used in the creation of this manuscript.

References

[1] Lalitha M, Vasu S. A study on graph theory in cryptography using python. Journal of Emerging Technologies and Innovative Research, 2023, 10(4), 97-107.
[2] Subramani S, M S, A K, Svn SK. Review of security methods based on classical cryptography and quantum cryptography. Cybernetics and Systems, 2025, 56(3), 302-320. DOI: 10.1080/01969722.2023.2166261
[3] Acharya B, Rath GS, Patra SK, Panigrahy SK. Novel methods of generating self-invertible matrix for hill cipher algorithm. https://dspace.nitrkl.ac.in/dspace/handle/2080/620
[4] Bokhary SA, Kharal A, Samman FM, Dalam ME, Gargouri A. Efficient graph algorithms in securing communication networks. Symmetry, 2024, 16(10), 1269. DOI: 10.3390/sym16101269
[5] Xue Y, Chen L, Mu Y, Zeng L, Rezaeibagha F, Deng RH. Structured encryption for knowledge graphs. Information Sciences, 2022, 605, 43-70. DOI: 10.1016/j.ins.2022.05.015
[6] Raghavendran P, Gunasekar T, Gochhait S. Sustainable cryptographic solutions: Enhancing decision-making and security with the pourreza transform. In 2024 International Conference on Decision Aid Sciences and Applications (DASA), Manama, Bahrain, 2024, pp. 1-7. DOI: 10.1109/DASA63652.2024.10836613
[7] Yesodha K, Krishnamurthy M, Thangaramya K, Kannan A. Elliptic curve encryption-based energy-efficient secured ACO routing protocol for wireless sensor networks. The Journal of Supercomputing, 2024, 80(13), 18866-99. DOI: 10.1007/s11227-024-06235-1
[8] Sen A, Heng SH, Tan SC. A comprehensive review of cryptographic techniques in federated learning for secure data sharing and applications. IEEE Access, 2025, 13, 135138 - 135164. DOI: 10.1109/ACCESS.2025.3593953
[9] Singh AK, Siddiqui ZA, Singh S, Singh AK, Siddiqui TJ. Recent advances in computational intelligence and cyber security, 1st ed.; Taylor & Francis Group: London, UK, 2024.
[10] Amudha P, Jayapriya J, Gowri J. An algorithmic approach for encryption using graph labeling. Journal of Physics: Conference Series, 2021, 1770, 012072. DOI: 10.1088/1742-6596/1770/1/012072
[11] Islam MS, Rahman MA, Bin Ameedeen MA, Ajra H, Ismail ZB, Zain JM. Blockchain-enabled cybersecurity provision for scalable heterogeneous network: A comprehensive survey. CMES-Computer Modeling in Engineering & Sciences, 2024, 138(1), 43-123. DOI: 10.32604/cmes.2023.028687
[12] Beaula C, Venugopal P. Encryption using double vertex graph and matrices. Solid State Technology, 2021, 64(2), 2486-2493.
[13] Mohan P, Rajendran K, Rajesh A. An encryption technique using a complete graph with a self-invertible matrix. Journal of Algebraic statistics, 2022, 13(3), 1821-1826.
[14] Banoth R, Regar R. Security standards for classical and modern cryptography. In Classical and Modern Cryptography for Beginners; Springer, Cham, Switzerland, 2023, pp. 47-83.
[15] Sasikumar K, Nagarajan S. Comprehensive review and analysis of cryptography techniques in cloud computing. IEEE Access, 2024, 12, 52325-52351. DOI: 10.1109/ACCESS.2024.3385449
[16] Klima R, Klima RE, Sigmon N, Sigmon NP. Cryptology: classical and modern, 2nd ed.; Taylor & Francis Group: New York, USA, 2018, pp. 496.
[17] Gupta D, Chandra H, Soni L. An encryption and decryption technique using planar graph with self-invertible matrix. Mathematics in Engineering, Science & Aerospace (MESA), 2024, 15(4), 1335.
[18] Kaur S, Singh S, Kaur M, Lee HN. A systematic review of computational image steganography approaches. Archives of Computational Methods in Engineering, 2022, 29(7), 4775-4797. DOI: 10.1007/s11831-022-09749-0

[19] Kottarathil J, Naduvath S, Kureethara JV. Graph theory and decomposition, 1st ed.; Taylor & Francis Group: New York, USA, 2024, pp. 200. DOI: 10.1201/9781003391678
[20] Tripathi SL, Agarwal D, Verma SB, Dwivedi S, Prakash KB, Singh BK. Emerging trends in IoT and computing technologies, 1st ed.; Taylor & Francis Group: London, UK, 2023, pp. 338. DOI:10.1201/9781003350057
[21] Adeniyi AE, Jimoh RG, Awotunde JB. A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security. Computers and Electrical Engineering, 2024, 118, 109330. DOI: 10.1016/j.compeleceng.2024.109330
[22] Cusack B, Chapman E. Using graphic methods to challenge cryptographic performance. The Proceedings of 14th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 2016, pp.30-36. DOI: 10.4225/75/58a6991e71023
[23] Al Etaiwi WM. Encryption algorithm using graph theory. Journal of Scientific Research and Reports, 2014, 3(19), 2519-2527.
[24] Zainol ZN, Yatin SF, Sani MK. Improving the security and privacy in malaysia academic digital libraries. Journal of Information and Knowledge Management, 2025, 15(SI2),132-145. DOI: 10.24191/jikm.v15iSI2.7819
[25] Opiłka F, Niemiec M, Gagliardi M, Kourtis MA. Performance analysis of post-quantum cryptography algorithms for digital signature. Applied Sciences, 2024, 14(12), 4994. DOI: 10.3390/app14124994
[26] Qasem MA, Thabit F, Can O, Naji E, Alkhzaimi HA, Patil PR, et al. Cryptography algorithms for improving the security of cloud-based internet of things. Security and Privacy, 2024, 7(4), e378. DOI: 10.1002/spy2.378
[27] Silva C, Cunha VA, Barraca JP, Aguiar RL. Analysis of the cryptographic algorithms in IoT communications. Information Systems Frontiers, 2024, 26(4), 1243-1260. DOI: 10.1007/s10796-023-10383-9
[28] Radhakrishnan I, Jadon S, Honnavalli PB. Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors, 2024, 24(12), 4008. DOI: 10.3390/s24124008
[29] Zhang C, Liang Y, Tavares A, Wang L, Gomes T, Pinto S. An improved public key cryptographic algorithm based on chebyshev polynomials and RSA. Symmetry, 2024, 16(3), 263. DOI: 10.3390/sym16030263
[30] Ibrahim MM, Venkatesan R, Ali N, Qureshi MI, Siddiqui HM, Tolasa FT, et al. Enhanced image hash using cellular automata with sponge construction and elliptic curve cryptography for secure image transaction. Scientific Reports, 2025, 15(1), 14148. DOI: 10.1038/s41598-025-98027-7
[31] Zeng T, Ye Y, Chen Y, Zhu D, Huang Y, Huang Y, et al. Deep hashing and attention mechanism-based image retrieval of osteosarcoma scans for diagnosis of bone cancer. Journal of Bone Oncology, 2024, 49, 100645. DOI: 10.1016/j.jbo.2024.100645
[32] Weerarathna MD, Perera AA, Ranasinghe PG. A novel cryptosystem using multipartite graphs. Proceedings of the International Conference on Business Excellence, 2022, 590-594.
[33] Ranasinghe PG, Bandara RM, Athapaththtu AM. Symmetric encryption using snake graphs and supermagic covering. Journal of the National Science Foundation of Sri Lanka, 2025, 52(4), 435-440. DOI: 10.4038/jnsfsr.v52i4.12196
[34] Ali N, Sadiqa A, Shahzad MA, Imran Qureshi M, Siddiqui HM, Abdallah SA, et al. Secure communication in the digital age: A new paradigm with graph-based encryption algorithms. Frontiers in Computer Science, 2024, 6, 1454094. DOI: 10.3389/fcomp.2024.1454094
[35] Obayes KK. Text encryption with graph theory based key generation. Journal of Al-Qadisiyah for Computer Science and Mathematics, 2024, 16(3), 26-35. DOI: 10.29304/jqcsm.2024.16.31650