MATRIX-BASED GRAPH ENCRYPTION USING PAW GRAPH ADJACENCY AND INVERTIBILITY

T. Vatchala Rani[1] and K. Rajendran[2]

[1]Research Scholar(M.Phil.), Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies, Chennai.

[2]Associate Professor, Department of Mathematics, Vels Institute of Science, Technology and Advanced Studies, Chennai.

Vatcalarani67@gmail.com[1] and **gkrajendra59@gmail.com**[2]

ABSTRACT:

In contemporary times, message encryption algorithms serve as one of the most essential security measures for protecting communication and data. The rapid growth of networks and the internet has significantly accelerated the advancement of encryption technologies. This paper presents an encryption and decryption method based on the use of a self-invertible key matrix and the adjacency matrix of paw graphs to generate a complex cipher. Notably, because the key matrix is self-invertible—meaning its inverse always exists—there is no need to explicitly compute the matrix inverse during decryption. This characteristic effectively reduces computational complexity and enhances the efficiency of the decryption process.

Keywords: Graph-Based Encryption, Paw Graph, Self-Invertible Matrix, Adjacency Matrix.

1. Introduction:

Cryptography is a mathematical technique used to enhance the security of data transmission and protect communication, information, and images from unauthorized access. While both plaintext and ciphertext are typically composed of alphabetic characters, messages may also include special characters such as punctuation marks, digits, spaces, or other symbols. In this work, the message units are encrypted using the encoding table provided below.

A	B	C	D	E	F	G	…	…	…	X	Y	Z		.	?
1	2	3	4	5	6	7	…	…	…	24	25	26	27	28	29

Table 1.0. Encoded Table

We have developed a novel approach aimed at enhancing the encryption key and improving the overall security of the transmitted information. Our method utilizes the adjacency matrix derived from a complete graph of a cycle graph in combination with a self-invertible key matrix for both encryption and decryption processes [1,5-10]. Since the key matrix is self-invertible, there is no need to compute its inverse during

decryption, thereby simplifying the process. Furthermore, the complete key matrix is not disclosed, which further strengthens data security and reduces the risk of the key being compromised.

The approach proposed in this study is based on the concept of the adjacency matrix of a paw graph. A self-invertible matrix is employed as the key to encrypt and decrypt the original message units, aiming to enhance security and introduce a novel, efficient method. Recovering the original message is highly challenging without knowledge of the underlying procedure. The remainder of this paper is organized as follows: Section II describes the process for constructing a self-invertible key matrix. Section III outlines the proposed encryption technique. An implementation example is provided in Section IV. Finally, Section V presents the conclusion and offers suggestions for future research.

2. Generation of a Self-Invertible Key Matrix

A matrix K is considered **self-invertible** under addition modulo p if it satisfies the condition $K = K^{-1}$, or equivalently, $K = K^{-1}$, or $K \cdot K^{-1} = K^{-1} \cdot K = I$, where I is the identity matrix.

2.1 Procedure for Generating a Self-Invertible Key Matrix of Even Order

To construct a self-invertible matrix of even order n, where n is the order of the adjacency matrix, we begin by selecting a random submatrix K_{22} of size $\frac{n}{2} \times \frac{n}{2}$.

The remaining submatrices of the self-invertible matrix are then derived based on S_{22}, using the following relationships:

- $K_{11} + K_{22} = 0 \Rightarrow K_{11} = -K_{22}, K_{21} = I + K_{11}, K_{12} = I - K_{11}$.

Once the submatrices $K_{11}, K_{12}, K_{21}, K_{22}$ are computed, the final self-invertible matrix K is constructed as:

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} = \begin{bmatrix} k_{11}k_{12} & \cdots & \vdots & \cdots & \cdots & k_{1n} \\ k_{21}k_{22} & \cdots & \vdots & \cdots & \cdots & k_{2n} \\ \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ \cdots\cdots\cdots\cdots & \cdots & \vdots & \cdots\cdots & \cdots\cdots\cdots \\ \cdots & \cdots & \cdots\cdots & \vdots & \cdots & \cdots\cdots & \cdots \\ \cdots & \cdots & \cdots\cdots & \vdots & \cdots & \cdots\cdots & \cdots \\ k_{n1}k_{n2} & \cdots & \vdots & \cdots\cdots & \cdots & k_{nn} \end{bmatrix}$$

This block matrix structure ensures that K is self-invertible and suitable for use as an encryption key.

3. Paw graph based enciphering technique

The proposed method, outlined in this section, integrates the adjacency matrices of the paw graph with a self-invertible matrix, which serves as the encryption key.

Encryption Algorithm:

To perform encryption, the following steps are followed:

Step 1: Begin by constructing paw graphs from the sequential letters in the plaintext message units. The process starts with the special character 'A' as the first vertex in each paw graph.

Step 2: Convert the characters of the message units into their corresponding numerical values using the encoding table.

Step 3: Calculate the weights of each edge in the paw graph by finding the numerical difference between each pair of adjacent vertices.

Step 4: Apply addition modulo p to these edge weights to generate the adjacency matrix for the paw graph.

Step 5: Generate the self-invertible key matrix using the procedure outlined in Section 2.1, based on the shared input data.

Step 6: Encrypt the original plaintext message by multiplying the generated adjacency matrix with the self-invertible key matrix.

The resulting encrypted data, along with the order of the adjacency matrix and the matrix used to generate the self-invertible key, can be transmitted. These matrices may be sent in either row-wise or column-wise format.

Algorithm for the Proposed Decryption Technique:

Step 1: The receiver begins by reviewing the received data to determine the matrix order, the encrypted matrix, and the matrix used in generating the self-invertible key matrix.

Step 2: Using the procedures described in Sections 2.1, the receiver reconstructs the self-invertible key matrix.

Step 3: The encrypted matrix is then multiplied by the self-invertible key matrix.

Step 4: Apply addition modulo p to the resulting matrix from Step 3 to recover the adjacency matrix of the corresponding graph.

Step 5: Based on the reconstructed adjacency matrix, the receiver can recreate the associated paw graph, including its nodes and edge weights.

Step 6: The original message is retrieved by computing the weights and corresponding vertices. The first vertex v_1 is assigned the value of the character 'A' (with a numerical value of 1), and each subsequent vertex is calculated using the formula $v_i = v_{\{i-1\}} + e_{\{i-1\}}$, where $e_{\{i-1\}}$ represents the weight of the previous edge.

4. Implementation examples

4.1.1 Using Paw graph:

Assume that the sender intends to communicate the word "TOM" to another user using the method described in the above Algorithm, which makes use of Paw graph and its adjacency matrix as well as the self-invertible key matrix described in Section 2.1.

Encryption- The following techniques are used to encrypt data:

The first vertex of the supplied plaintext message units should be added as an add-on character A, and the message "$A\,TOM$" should then be converted into the vertices of paw graph. The specified message units' consecutive letters are used as the vertices to connect the vertices from vertex 1 to vertex 4.

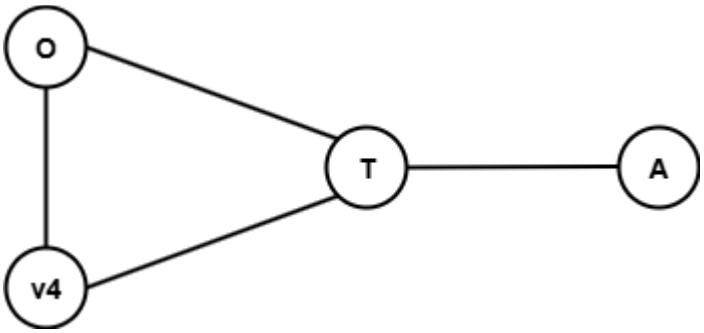

Figure 4.1 Paw graph of original message

We obtain, $A \to 1,\ T \to 20,\ O \to 15,\ M \to 13$by using the provided encoded table

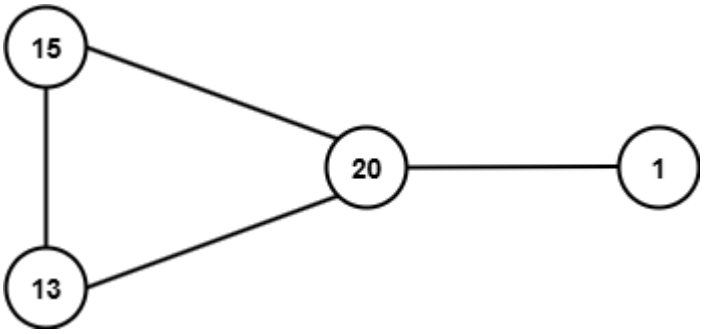

Figure 4.2. Encoded Paw graph

The weights of the edges in the paw graph are determined by calculating the numerical distance between the consecutive two linked vertices and adding them $modulo\ 29$, as we are using 29 characters in the encoded table provided, (Table 1.0).

$(e1 = Code\,T - Code\,A, e2 = Code\,O - Code\,T, \dots)$

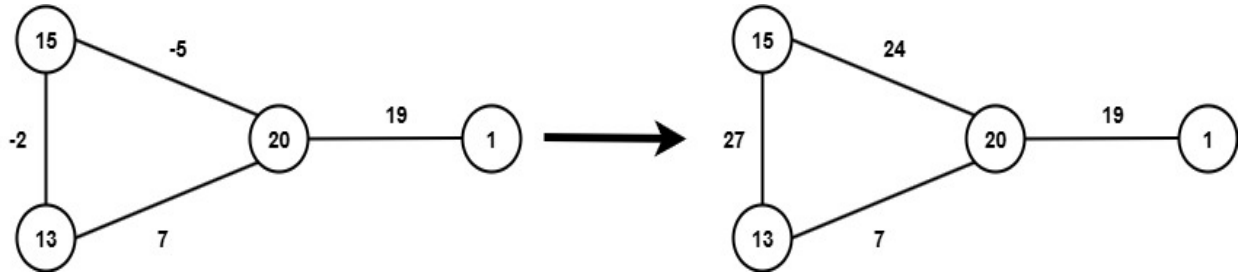

Figure 4.3. Paw graph with edge weights

The above graph's associated adjacency matrix, designated by the letter "M," has been calculated.

$$M = \begin{bmatrix} 0 & 19 & 0 & 0 \\ 19 & 0 & 24 & 7 \\ 0 & 24 & 0 & 27 \\ 0 & 7 & 27 & 0 \end{bmatrix}$$

Let the commonly shared $\frac{n}{2} \times \frac{n}{2}$ matrix $K_{22} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \end{bmatrix}$. Then the remaining $\frac{n}{2} \times \frac{n}{2}$matrices are,

$$K_{11} = \begin{bmatrix} 28 & 25 \\ 27 & 24 \end{bmatrix}, K_{12} = I - K_{11} = \begin{bmatrix} 2 & 4 \\ 2 & 6 \end{bmatrix}, K_{21} = I + K_{11} = \begin{bmatrix} 0 & 25 \\ 27 & 25 \end{bmatrix}$$

$$\therefore K = \begin{bmatrix} K_{11}K_{12} \\ K_{21}K_{22} \end{bmatrix} = \begin{bmatrix} 28 & 25 & 2 & 4 \\ 27 & 24 & 2 & 6 \\ 0 & 25 & 1 & 4 \\ 27 & 25 & 2 & 5 \end{bmatrix}$$

Finally, by multiplying M and K the encrypted matrix was computed

$$C = M \cdot K = \begin{bmatrix} 0 & 19 & 0 & 0 \\ 19 & 0 & 24 & 7 \\ 0 & 24 & 0 & 27 \\ 0 & 7 & 27 & 0 \end{bmatrix} \cdot \begin{bmatrix} 28 & 25 & 2 & 4 \\ 27 & 24 & 2 & 6 \\ 0 & 25 & 1 & 4 \\ 27 & 25 & 2 & 5 \end{bmatrix}$$

$$C = \begin{bmatrix} 513 & 456 & 38 & 114 \\ 721 & 1250 & 76 & 207 \\ 1377 & 1251 & 102 & 279 \\ 189 & 843 & 41 & 150 \end{bmatrix}$$

The encrypted matrix, the matrix which assists in constructing the self-invertible matrix, are converted into a row or column matrices and transmitted to the end user over any type of median.

$$[\,4,0,513,456,38,114,721,1250,76,207,1377,1251,102,279,189,843,41,150;1,4,2,5].$$

Decryption: The procedures below are used for decryption.

The receiver may determine the order of the matrix, the encrypted matrix, and the matrix that aids in creating the self-invertible key matrix using the information they have received.

$$C = \begin{bmatrix} 513 & 456 & 38 & 114 \\ 721 & 1250 & 76 & 207 \\ 1377 & 1251 & 102 & 279 \\ 189 & 843 & 41 & 150 \end{bmatrix}$$

With the process explained in Section 2.1, the receiver is also producing the self-invertible matrix.

$$K = \begin{bmatrix} K_{11}K_{12} \\ K_{21}K_{22} \end{bmatrix} = \begin{bmatrix} 28 & 25 & 2 & 4 \\ 27 & 24 & 2 & 6 \\ 0 & 25 & 1 & 4 \\ 27 & 25 & 2 & 5 \end{bmatrix}$$

$$C \cdot K = \begin{bmatrix} 513 & 456 & 38 & 114 \\ 721 & 1250 & 76 & 207 \\ 1377 & 1251 & 102 & 279 \\ 189 & 843 & 41 & 150 \end{bmatrix} \cdot \begin{bmatrix} 28 & 25 & 2 & 4 \\ 27 & 24 & 2 & 6 \\ 0 & 25 & 1 & 4 \\ 27 & 25 & 2 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 29754 & 27569 & 2204 & 5510 \\ 59527 & 55100 & 4432 & 11723 \\ 79866 & 73974 & 5916 & 14817 \\ 32103 & 29732 & 2405 & 6728 \end{bmatrix}$$

After taking addition $modulo$ 29 we obtain,

29754(mod 29) = 0, 27569(mod 29) = 19, 2204(mod 29) = 0, …, 6728 (mod 29) = 0.

$$\therefore C \cdot K = \begin{bmatrix} 0 & 19 & 0 & 0 \\ 19 & 0 & 24 & 7 \\ 0 & 24 & 0 & 27 \\ 0 & 7 & 27 & 0 \end{bmatrix} = M$$

For the aforementioned adjacency matrix, the corresponding paw graph was formed

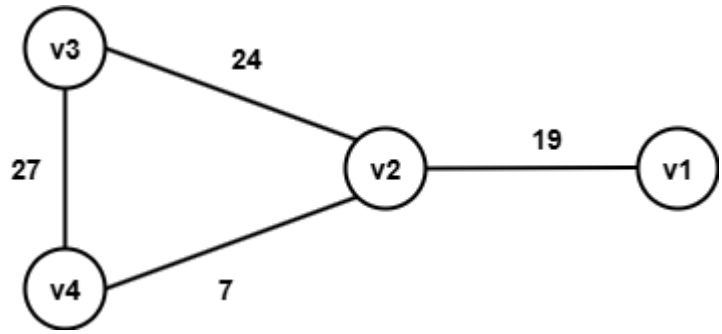

Figure 4.4. Paw graph of decrypted adjacency matrix.

The vertices of the above graph were constructed by adding the numerical equivalent values of vertex with corresponding edge. Since we are starting with an additional character A, we know that the first vertex must be 1, so the remaining vertices are found by letting $v_1 = 1$, so $v_2 = 1 + 19 = 20, v_3 = 20 + 24 = 44 = 15, v_4 = 15 + 27 = 42 = 13$.

∴ The vertices are $1, 20, 15, 13$.

∴The original message is $1 \rightarrow A$, $20 \rightarrow T, 15 \rightarrow O$, $13 \rightarrow M$. i.e., $A\,TOM$.

5. Conclusion:

In today's world, ensuring information security is crucial. To address this, many studies have employed symmetric encryption techniques such as the Caesar cipher, Hill cipher, graphical methods, and others. This research proposes a novel symmetric encryption approach that enhances data security by utilizing a self-invertible matrix as the key, combined with adjacency matrices of paw graphs. The proposed method is particularly effective for paw graphs and offers improved efficiency by eliminating the need for intermediate steps. It employs a simple yet secure encryption and decryption process. Importantly, the entire key matrix is not shared during key exchange, simplifying the process and making it more secure against potential attacks. Since the key matrix is self-invertible, there is no need to compute its inverse during decryption, further streamlining the procedure. In this study, a basic paw graph is used for both encrypting and decrypting messages. In the future, this approach will be extended to incorporate more complex concepts from graph theory and advanced encryption techniques, including applications in image and video encryption.

References:

1. Acharya, B., Rath, G.S., Patra, S.K., Panigrahy, S.K., A Novel methods of generating self-invertible matrix for Hill Cipher Algorithm, International Journal of Security(2007), pp.14-21.
2. Alastair Farrugia, Self-complementary graphs and generalisations: a comprehensive reference manual, University of Malta, 1999.
3. Arumugam S, Ramachandran S, Invitation to Graph theory, Scitech Publications, (2015).

4. Diffie, W., Hellman, M., New directions in Cryptography, IEEE Trans. Inf. Theory 22 (6), (1976), pp.644-654.
5. Mohan. P, Rajendran. K, Rajesh. A, An Encryption Technique using a Complete graph with a Self-invertible matrix, Journal of Algebraic statistics, Volume 13. No 3, (2022), https://publishoa.com/index.php/journal/article/view/816, pp.1821-1826.
6. Mohan P, Rajendran K, Rajesh A. A Hamiltonian Path-Based Enciphering Technique with the use of a Self-Invertible Key Matrix, Indian Journal of Science and Technology, 15(44) (2022), pp.2351-2355.
7. Mohan P, Rajendran K, Rajesh A. An encryption Technique using the adjacency matrices of certain graphs with a self-invertible key matrix, E3S Web of Conf,Volume 376, 01108(2023).
8. Mohan P, Rajendran K, Rajesh A. Enhancing Computational Performance of Minimal Spanning Tree of Certain Graphs Based Enciphering Technique Using Self-Invertible Key Matrix, Journal of Aeronautical Materials(1005-5053), Vol 43, Issue-01(2023), pp 359-371.
9. Mohan P, Suresh M.V, Periyasamy C. Enhancement of Minimum Spanning Tree Computing Performance in a Complete Graphic Based Enciphering Method Using Self-Invertible Key Matrix, Nanotechnology Perceptions(1660-6795), Vol 20, S12 (2024), pp 1436-1449.
10. Mohan P, Suresh M.V, Periyasamy C. A Graph Theoretic Encryption Using the Self-Invertible Key Matrices and the Adjacency Matrices of Complete Graph, Palestine Journal of Mathematics, (2219-5688), Vol 14(Special Issue II) 2025, pp 180-187.
11. Neal Koblitz, A course in Number Theory and Cryptography, second edition, Springer.
12. Weisstein, Eric W., Bull Graph, Math World.
13. Weal Mahmoud AI Etaiwi, Encryption algorithm using Graph theory, Journal of Scientific Research and Reports, 3(19), (2014), pp. 2519-2527.
14. Yamuna M, Meenal Gogia, Ashish Sikka, Md. Jazib Hayat Khan, Encryption Using Graph Theory and Linear Algebra, International Journal of Computer Application, ISSN:2250-1797,Issue 2 Vol 5(2012), pp.102-107.
15. Ziad E. Dawahdeh, Shahrul N. Yaakob, Rozmie Razif bin Othman, A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher, Journal of King Saud University - Computer and Information Sciences,30(3),(2018), pp.349-355.