An Application for Pair Sum Modulo Labeling in Cryptography

P. Amudha* and J. Jayapriya

Department of Mathematics, Sathyabama Institute of Science and Technology, Chennai, 600119, India

Revista Internacional
Métodos numéricos
para cálculo y diseño en ingeniería

RIMNI

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

In cooperation with CIMNE

INFORMATION

Keywords:
Encryption
decryption
graph labeling
pair sum modulo labeling
self-invertible matrix

DOI: 10.23967/j.rimni.2025.10.63629

SCIPEDIA
Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería
Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería RIMNI In cooperation with CIMNE

An Application for Pair Sum Modulo Labeling in Cryptography

P. Amudha* **and J. Jayapriya**

Department of Mathematics, Sathyabama Institute of Science and Technology, Chennai, 600119, India

ABSTRACT

In the dynamic realm of cybersecurity, it is important to create a strong cryptographic technique to protect sensitive data against advanced threats. This paper introduces an innovative encryption and decryption technique leveraging graph theory and matrix algebra, especially through the use of pair sum modulo (PSM) labeling of graphs, adjacency matrices, and self-invertible matrices. The PSM labeling for a simple undirected graph $\mathcal{G}(\mathrm{V_G}, \mathrm{E_G})$ with $|\mathrm{V_G}| = p$ and $|\mathrm{E_G}| = q$, is an 1-1 map $\mathcal{F}_\mathrm{G}\colon \mathrm{V_G} \to \{\pm 1, \pm 2, \ldots, \pm p\}$, and there is an induced edge labeling bijective function $g_\mathrm{G}\colon \mathrm{E_G} \to \{0, 1, 2, \ldots, (q-1)\}$ such that $g_\mathrm{G}(uv) = [\mathcal{F}_\mathrm{G}(u) + \mathcal{F}_\mathrm{G}(v)] \pmod q$ is distinct for each edge uv. A graph that satisfies PSM labeling is known as a PSM graph. Building on recent developments in graph labeling and matrix applications in cryptography, our method enhances security. It improves resistance to brute-force attacks by utilizing a large key space. Additionally, it leverages the complexity of matrix inversion to make cryptanalysis more difficult. The amalgamation of these mathematical groundworks reinforces the entropy and resistance to bit-flipping, thereby stimulating the ciphertext against statistical and cryptanalytic threats. We utilize the core principles of PSM labeling and algorithmic encryption methods, as defined in prior research, to develop an innovative cryptographic algorithm. The sustainability of the proposed method is verified through a thorough evaluation of its encryption efficacy, computational complexity, and a comparative study with existing cryptographic techniques. This work not only contributes to a new approach to the cryptographic domain but also opens avenues for further research into the integration of advanced mathematical structures in encryption algorithms.

OPEN ACCESS

Received: 20/01/2025

Accepted: 03/03/2025

Published: 07/04/2025

DOI
10.23967/j.rimni.2025.10.63629

Keywords:
Encryption
decryption
graph labeling
pair sum modulo labeling
self-invertible matrix

1 Introduction

In today's technological era, online security has become increasingly important because of the ongoing sharing of sensitive information. This concern affects multiple sectors, including finance, healthcare, military, and digital communications. Cryptography plays a vital role in ensuring the integrity, confidentiality, and authenticity of data. It serves as a key element in modern information security by enabling the secure transfer and protection of data, through encryption and decryption

*Correspondence: P. Amudha (amutha106@gmail.com).

methods. The rising complexity of dangers calls for the creation of encryption methods, in response to the evolving cyber threats we face today. Graph theory has proven to be a tool in the field of cryptography due to its presence in mathematics [1,2]. Moreover, the innovative concept of self-inverting matrices presents opportunities for crafting cryptographic systems.

Recently, researchers and professionals have been working hard to improve techniques to keep up with the increasing need for security and privacy protection [3]. The Hill Cipher is a standout encryption method known for its effective handling of text and image data. Originally introduced by Lester S. Hill in 1929, this cipher uses algebraic principles and matrix operations to convert plaintext into ciphertext and *vice versa* [4]. Over the years, many developments have been made to improve its security and efficiency, such as including self-invertible key matrices and elliptic curve cryptography. Utilizing random self-invertible matrices as keys in cryptographic systems bolsters security by allowing easy reversal of the key matrix during decryption, eliminating complex inverse calculations [5,6]. This reduces the decryption process and reinforces the encryption, making it more resistant to attacks. The complicated nature of these matrices adds an extra layer of security, making it harder for adversaries to breach the encryption [7]. Consequently, cryptographic systems utilizing random self-invertible matrices are considered highly secure and efficient for protecting sensitive information. Acharya et al. [8] came up with new ways to create self-invertible matrices for the Hill cipher. This helps fix the problem of key matrices that can't be inverted. It also makes decryption easier. Yunos et al. [9] looked at self-invertible 3×3 matrices for a Cipher Trigraphic Polyfunction cryptosystem. They found that using different encryption keys helped cut down the complexity of getting the inverses. Ching et al. [10] studied how self-invertible matrices improve the Cipher Hexagraphic Polyfunction cryptosystem. They pointed out that certain patterns in the matrices can boost security. Sundarayya et al. [11] discuss the application of self-invertible matrices in the Affine Hill cipher and Digraph Affine Hill cipher, emphasizing the advantages of removing the need for matrix inversion during decryption.

Yi [12] introduced machine learning methodologies aimed at utilizing the measurement of side channel attacks in the realm of post-quantum signatures. In addition to that, the author created a dedicated machine learning model specifically tailored for evaluating side channel attacks.

Recent developments in chaotic systems have significantly influenced the field of encryption, especially for image and audio data. For instance, Feng et al. [13] created a new multi-channel image encryption algorithm using pixel reorganization and hyperchaotic maps, which showed better encryption efficiency than existing methods. In a similar vein, Kumar et al. [14] created an image encryption algorithm that integrates a discrete memory-based logistic map combined with deep neural networks, achieving high security while maintaining image quality. Benaissi et al. [15] proposed a groundbreaking technique that combines three enhanced chaotic systems for key generation. Abdellatif et al. [16] examined the correlation characteristics and security aspects of chaotic keymaps, considering their practical application in cryptography. Dridi et al. [17] investigated the use of chaotic maps within block ciphers, emphasizing their potential to make better security and confusion properties of encryption processes. Vismaya et al. [18] examined the discrete dynamics of the Degn-Harrison model, exploring its usage in expounding oscillations in the respiration rates of Klebsiella aerogenes cultures. Their analysis focuses on transitions between periodic, quasiperiodic, and chaotic states using bifurcation theory and fixed-point analysis. The authors also investigated the model's behaviour in network topologies, identifying rich spatiotemporal patterns, including synchronization and chimera states, influenced by coupling strengths. Furthermore, Vismaya et al. proposed an image encryption method based on these chaotic signals and Convolutional Neural Network (CNN), giving robust security for image data transmission.

In the domain of audio encryption, Wanying Dai et al. [19] created an audio encryption algorithm that utilizes the Chen memristor chaotic system, transforming audio signals into color image data to improve security. Similarly, Abou El Qassime et al. [20] introduced a novel speech encryption method utilizing a concealed hyperchaotic attractor, exposing strong protection against various cryptographic attacks. These studies underscore the potential of chaotic systems to improve the security and efficiency of multimedia data encryption methods.

Many encryption methodologies have employed anti-magic labeled graphs to enhance data security, as discussed by Kumar Gurjar et al. [21]. Their method uses graph theory principles to develop sophisticated encryption algorithms that pose significant challenges for attackers. This article provides a novel encryption and decryption technique that utilizes Pair Sum Modulo (PSM) labeling of graphs, adjacency matrices of graphs, and self-invertible matrices. This approach is inspired by recent advancements in the field, including PSM labeling of graphs [22], algorithmic methods for encryption using graph labeling [23], and the incorporation of self-invertible matrices in cryptographic schemes [24].

The objective of including PSM labeling is to improve the security and resilience of cryptographic systems. It offers a new perspective on addressing key issues such as ciphertext indistinguishability, key management, and resistance to cryptanalysis. PSM labeling is ideal for cryptographic applications due to its injective and bijective properties, ensuring uniqueness and preventing collisions. Using modular arithmetic in edge labeling produces diffusion, increasing complexity and security. Additionally, the extensive labeling space boosts entropy, making the system resilient to attacks. Also, the foundation of graph theory provides mathematical rigor, ensuring the encryption process is robust and reliable. These features collectively make the PSM labeling a promising choice for cryptographic systems. By analysing the suggested method, for securing and revealing data and delving into its world application and effectiveness assessment practically, within this article's context aims to push forward the development of cryptographic methods and how they are used in real life scenarios.

In the subsequent sections, we delve deeper into the theoretical foundations, algorithmic intricacies, and real-world implications of PSM labeling in cryptography, shedding light on its potential to shape the future of secure communication and data protection.

2 Preliminaries

Definition 2.1: In cryptography, plaintext refers to text that is easily readable and understood before it is transformed into ciphertext through encryption or after it has been decrypted. It does not require any special processing or decryption to be interpreted.

Definition 2.2: Cipher text is the result of applying encryption to plain text using an algorithm (cipher) to make it unintelligible to anyone except those who have the key to decrypt it. It is the encrypted form of plain text, designed to protect the confidentiality of the information it contains.

Definition 2.3: In graph theory, labeling a graph entails assigning specific labels or values to its vertices or edges, adhering to predefined rules or constraints.

Definition 2.4: The PSM labeling for a simple undirected graph $\mathcal{G}(\mathrm{V_G},\ \mathrm{E_G})$ with $|\mathrm{V_G}| = p$ and $|\mathrm{E_G}| = q$, is an injective function $\mathcal{F}_{\mathrm{G}}\colon \mathrm{V_G} \rightarrow \{\pm 1, \pm 2, \ldots, \pm p\}$, there exists an induced edge labeling bijective function $g_{\mathrm{G}}\colon \mathrm{E_G} \rightarrow \{0, 1, 2, \ldots, (q-1)\}$ such that $g_{\mathrm{G}}(uv) = [\mathcal{F}_{\mathrm{G}}(u) + \mathcal{F}_{\mathrm{G}}(v)] \pmod q$ is distinct for each edge uv. A graph which is created to support PSM labeling is known as a PSM graph

Definition 2.5: A self-invertible matrix, also known as an involutory matrix, is a square matrix that equals its own inverse.

Algorithm for pair sum modulo labeling of path graphs

The following is a C++ like algorithm for giving PSM labeling to path graph P_n with n vertices. To assign PSM labels to the vertices of a path graph P_n with n vertices, we follow the approach outlined in Algorithm 1.

Algorithm 1: PSM Labeling Techniques Applied to Path Graphs

```
Step 1: Start
Step 2: Declare the variable int n
Step 3: Input n
Step 4: Declare the variables int v[n + 1], e[n] and let q = n − 1.
Step 5: For (i1 = 1 to n), $v[i1] \leftarrow 0, e[i1] \leftarrow 0$
Step 6: If (n is even) then {$v[1] \leftarrow 1 - n$ and For (i2 = 2 to n), $v[i2] \leftarrow i2 - 1$}
            Else
               For (i3 = 1 to $\frac{n+1}{2}$), $v[2(i3) - 1] \leftarrow \frac{n-1}{2} + i3$
               For (i4 = 1 to $\frac{n-1}{2}$), v [2∗(i4)] ← i4
Step 7: For (int i7 = 1 to n − 1){ e[i7] ← (v[i7] + v[i7 + 1]) mod(n − 1)}
Step 8: int k2 = 0;
         For (int r1 = 1 to (n − 1)){
            int k1←0;
            For (int r2 = (r1 + 1) to n){
               If (v[r1] ≠ v[r2]) then k1 ← k1 + 1;}
               If (k1 = (n-r1)) then k2←k2 + 1;}
Step 9: If (k2 = n − 1)
            int d ← 0;
            For (int a = 1 to (n − 2)){c ← 0;
               For (int b = (a + 1) to (n − 1)), {
                  If (e[a] ≠ e[b]) then c ← c + 1;}
               If (c = n − 1 − a) then d ← d + 1;}
Step 10: If (d = (n − 2) and k2 = (n − 1)) then {
            Print "Vertex labels: ";
            For (int t1 = 1 to n) {Print "v" t1 "=" v[t1] " ";}
            Print "Edge labels: ";
            For (int p2 = 1 to n − 1) {Print "e" p2 "=" e[p2] "   "}
            Print "Since all the vertex and edge labels are distinct, the given graph is a PSM graph."}
         Else
Print " Since all the vertex and edge labels are not distinct, the given graph is not a PSM graph."
Step 11: Stop
```

The flow chart illustrating the process for determining PSM labeling for P_n is presented in Fig. 1.

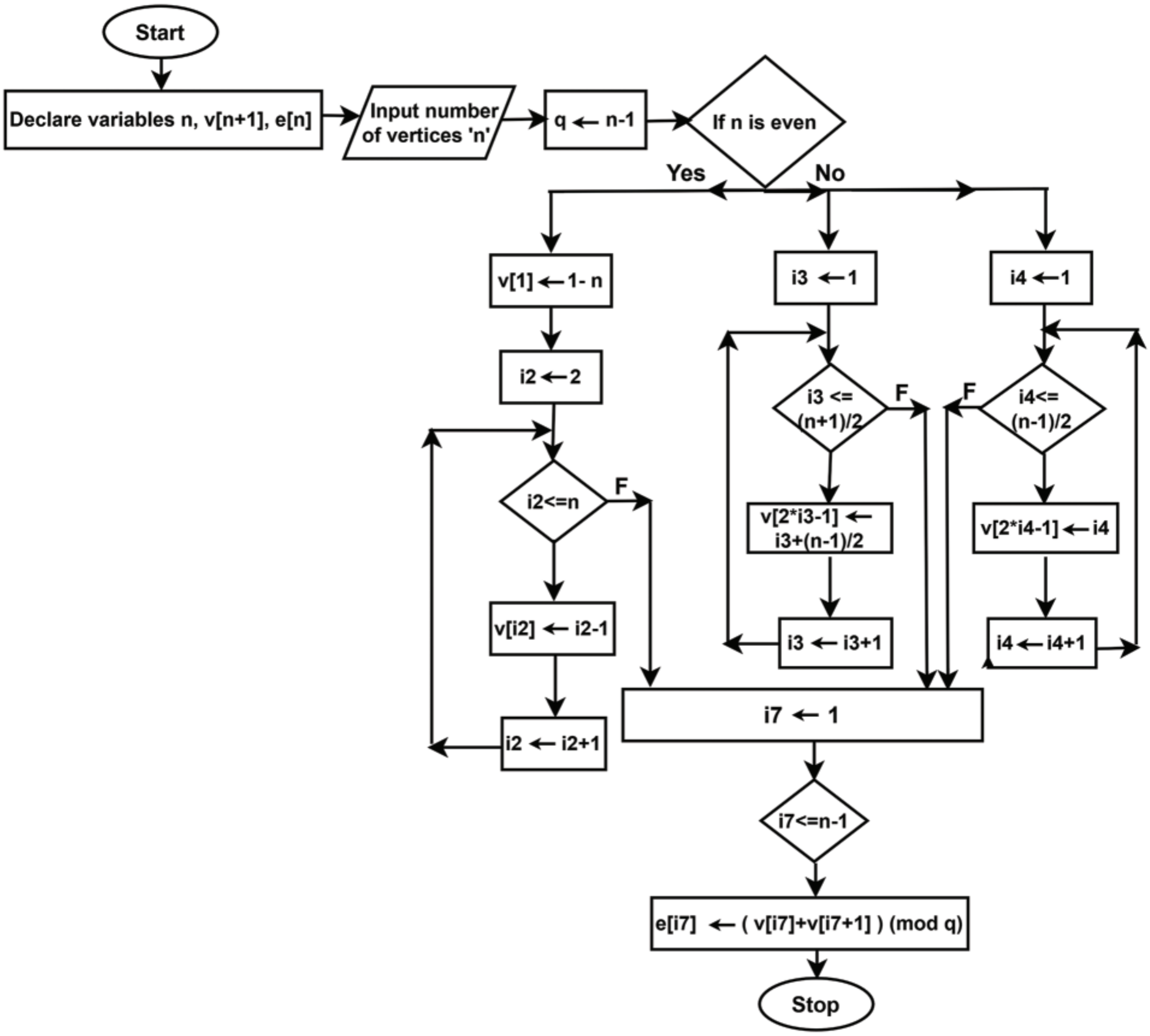

Figure 1: Diagram illustrating the process of determining PSM labeling for the vertices and edges of a path graph P_n, where n can be either even or odd

Example 2.1: The illustrations of PSM labeling for the path graphs P_5 and P_6 are presented in Figs. 2 and 3, respectively.

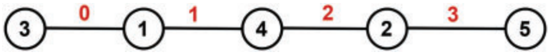

Figure 2: Diagram illustrating the PSM labeling along the path graph P_5. This figure demonstrates the application of the PSM labeling technique on the path graph with 5 vertices, showcasing the specific labeling pattern and how it adheres to the rules of the method for odd number of vertices

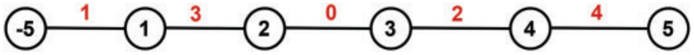

Figure 3: Diagram illustrating the PSM labeling along the path graph P_6. This figure highlights the application of the PSM labeling technique on the path graph with 6 vertices, demonstrating the specific labeling pattern and how it adheres to the rules of the method for even numbers of vertices

3 Methods

3.1 *The Proposed Crypto System*

The encoded table (Table 1) below is used to encode the message units of this operation.

Table 1: The correspondence between the character and integers. This encoded table is used to convert plaintext letters into numerical values for encryption

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P	Q	R	S	T	U	V	W	X	Y	Z	space	·	?	
16	17	18	19	20	21	22	23	24	25	26	27	28	29	

3.1.1 Methods to Generate Self-Invertible Key Matrix of Order n under Modulo φ

The analyses provided in this document for the creation of self-invertible matrices apply to matrices composed of positive integers, specifically those that represent the residues in modulo arithmetic concerning a prime number φ [5,11].

Let $M = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix}$ represents a self-invertible matrix of order n, structured as $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$.

Since M is a self-invertible matrix, $M^2 = I$, where I denotes the identity matrix modulo a prime number φ.

$\Rightarrow \mathrm{M}_{11}^2 + M_{12}M_{21} = I$;

$M_{11}M_{12} + M_{12}M_{21} = 0$;

$M_{21}M_{11} + M_{22}M_{21} = 0$; and

$M_{21}M_{12} + \mathrm{M}_{22}^2 = I$.

Method 1: When M is of even order

Let M_{11}, M_{12}, M_{21}, M_{22} are square matrices of order $\frac{n}{2}$.

Step 1: Choose any arbitrary matrix M_{22} of size $\frac{n}{2} \times \frac{n}{2}$.

Step 2: Find $M_{11} = -M_{22}$

Step 3: Take $M_{12} = c(I - M_{11})$ or $c(I + M_{11})$, c is a scalar constant.

Step 4: Then $M_{21} = \frac{1}{c}(I + M_{11})$ or $\frac{1}{c}(I - M_{11})$.

Step 5: Complete the formation of the matrix M.

Method 2: When M is of any order n

Let $M_{11} = [a_{11}]$, $M_{12} = [a_{12} \quad \dots \quad a_{1n}]$, $M_{21} = \begin{bmatrix} a_{21} \\ \vdots \\ a_{n1} \end{bmatrix}$, $M_{22} = \begin{bmatrix} a_{22} & a_{23} & \cdots & a_{2n} \\ a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$.

Step 1: Select a non-singular random matrix M_{22} of size $(n-1) \times (n-1)$ that possesses $(n-2)$ eigen values which can be either $+1$ or -1 or both.

Step 2: Calculate the other eigen value γ of M_{22}.

Step 3: Set $a_{11=} - \gamma$.

Step 4: Determine the consistent solution for all the elements of M_{12} and M_{21} solving the equation $M_{21}M_{12} + \mathrm{M}_{22}^2 = I$.

Step 5: Form the matrix M.

Example 3.1: The process of creating a self-invertible matrix of order 4 under modulo 17 is detailed here.

Let $M_{22} = \begin{bmatrix} 6 & 11 \\ 2 & 9 \end{bmatrix}$.

Then $M_{11} = -M_{22} = -\begin{bmatrix} 6 & 11 \\ 2 & 9 \end{bmatrix} \pmod{17} = \begin{bmatrix} 11 & 6 \\ 15 & 8 \end{bmatrix} \pmod{17}$.

If c = 1, $M_{12} = c(I - M_{11}) = \begin{bmatrix} 7 & 11 \\ 2 & 10 \end{bmatrix}$ (mod 17) and

$M_{21} = \frac{1}{c}(I + M_{11}) = \begin{bmatrix} 12 & 6 \\ 15 & 9 \end{bmatrix}$ (mod 17)

Therefore, the self-invertible matrix $M = \begin{bmatrix} 11 & 6 & 7 & 11 \\ 15 & 8 & 2 & 10 \\ 12 & 6 & 6 & 11 \\ 15 & 9 & 2 & 9 \end{bmatrix}$ (mod 17).

Example 3.2: The process of generating a self-invertible matrix of order 3 modulo 23 is explained here.

Let $M_{22} = \begin{bmatrix} 3 & 6 \\ 2 & 7 \end{bmatrix}$ which has eigen values 1,9.

So, $M_{11} = [9]$, and one of the consistent solutions of $M_{21}M_{12} + \mathrm{M}_{22}^2 = I$ is $M_{12} = \begin{bmatrix} 3 & 9 \end{bmatrix}$, $M_{21} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Therefore, $M = \begin{bmatrix} 14 & 3 & 9 \\ 1 & 3 & 6 \\ 1 & 2 & 7 \end{bmatrix}$ (mod 23).

3.1.2 Algorithm for the Suggested Encryption Method

The following steps are utilized to carry out the encryption:

Step 1: Identify initial letter and create path graph

- Begin by identifying the initial letter of the given message unit, using the special character A.
- Create the necessary path graph P_n by connecting the consecutive letters in the provided plain text message units.

Step 2: Convert message units using encoding table

- Utilize the encoding table (Table 1) to convert the message units into their corresponding numerical equivalents.

Step 3: Calculate weights of edges in path graph

- Calculate the weights of each edge in the path graph by determining the numerical variance between the two adjacent vertices.

Step 4: Generate adjacency matrix M_1

- Generates the adjacency matrix M_1 by applying addition modulo 29.

Step 5: Apply PSM labeling algorithm and generate adjacency matrix M_2

- Apply the PSM labeling algorithm to path graphs to determine the PSM labels for both the vertices and edges of P_n.
- Determine the adjacency matrix M_2 for the PSM labeled path graph.
- Obtain the matrix M_3 by adding M_1 and M_2 together.

Step 6: Generate random matrix K_{22} and construct self-invertible matrix K

- Generate a random matrix K_{22}, following the procedures outlined in Section 3.1.1.
- Construct a self-invertible matrix K under modulo 29 with the same procedure.
- Ensure that for the same plain text, a different matrix K_{22} is generated each time.

Step 7: Perform matrix multiplication and share encrypted information

- Perform matrix multiplication between M_3 and the recently generated self-invertible key matrix K to obtain the encrypted information for the initial plaintext.
- Share the encrypted matrix, the order of the adjacency matrix, and the matrix utilized to form the self-invertible matrix with another user via an insecure pathway in the form of either a row or column matrix.

3.1.3 Algorithm for the Suggested Decryption Method

Step 1: Establish order and matrices

- Backtrack through the information received to establish the order of the matrix n, the encrypted matrix C, and the matrix K_{22} that assists in the production of the self-invertible key matrix K.

Step 2: Generate self-invertible matrix K

- The recipient is required to generate the self-invertible key matrix K by utilizing the information provided in Section 3.1.1.

Step 3: Construct path graphs and apply PSM labeling

- The receiver is required to construct a path graph with n vertices based on the order of the matrix.
- Then determine the PSM labels of both the vertices and edges of the path graph using the Algorithm 1.
- Generate the adjacency matrix M_2 for the PSM labeled path graph.

Step 4: Calculate product to obtain matrix $\mathbf{M_3}$

- Calculate the product of the encoded matrix C with the self-invertible matrix K created in step 2 to obtain the resulting matrix M_3.

Step 5: Obtain adjacency matrix $\mathbf{M_1}$

- Obtain the adjacency matrix M_1 of the desired graph by subtracting M_2 from M_3 and then performing addition modulo 29 on the resulting matrix.

Step 6: Construct path graph from adjacency matrix $\mathbf{M_1}$

- From the adjacency matrix M_1, the recipient can construct the necessary path graph containing nodes and designated weights.

Step 7: Calculate the secret message from edge weights and vertices

- Calculate the message by summing the edge weight with its corresponding left end vertex.
- Note that vertex v_1 is denoted by the letter A and holds the value 1.
- Vertex v_2 is obtained by adding the weight e_1 to v_1, and so on.

4 Experiments and Results

Assume the sender intends to transmit the message “PSM GRAPH” to the recipient utilizing the method outlined in the preceding section. The self-invertible key matrix, which has been created utilizing Section 3, will be utilized for this purpose.

4.1 Encryption

Encryption involves a series of steps, starting with the addition of a unique character A at the start of the plain text message units. Subsequently, the message “APSM GRAPH” is transformed into vertices of a path graph consisting of 10 vertices. These vertices are connected in sequence from left to right by linking consecutive letters within the message units, as depicted in Fig. 4.

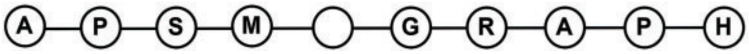

Figure 4: Path graph representation of the given plaintext, where each vertex corresponds to a character in the message and edges represent the sequential connections between characters

By utilizing the encoded table, the corresponding values for each letter are shown as follows and depicted in Fig. 5:

A→1, P→16, S→19, M→13, space→27, G→7, R→18, A→1, P→16, H→8.

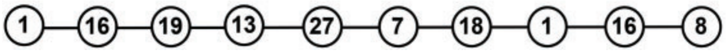

Figure 5: Encoded path graph derived from the plaintext, where each vertex is replaced by its corresponding integer from the encoded table

The weights of the edges in this graph are determined by calculating the numerical difference between each pair of adjacent vertices, followed by taking the sum modulo 29 due to the 29-character encoded table being utilized. For instance, $e_1 = (\text{Code G} - \text{Code A}) \bmod 29$, $e_2 = (\text{Code R} - \text{Code G}) \bmod 29$, and so on. Fig. 6 shows the encoded path graph with edge weights.

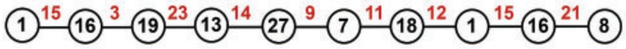

Figure 6: Encoded path graph with edge weights, where each edge is assigned a weight based on the numerical difference between each pair of adjacent vertices, followed by taking the sum modulo 29

The adjacency matrix M_1 for the graph mentioned earlier has been calculated.

$$M_1 = \begin{pmatrix} 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 23 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 23 & 0 & 14 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 & 0 & 12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 21 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 \end{pmatrix}$$

Now, utilizing the algorithm for PSM labeling of path graphs, we can determine the PSM labeling of graph P_{10} as depicted in Fig. 7.

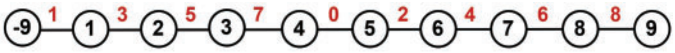

Figure 7: PSM labeling of the path graph P_{10}, showing the assignment of labels to each vertex according to the PSM labeling scheme

Determine the adjacency matrix M_2 for the path graph with PSM labeling.

$$M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

Perform the addition of matrices M_1 and M_2, and represent the resulting matrix as M_3.

$M_3 = M_1 + M_2$

$$\mathrm{M}_3 = \begin{pmatrix} 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 16 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 21 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 21 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Now that the calculation of the key matrix K is required, we utilize the matrix K_{22} to form the self-invertible key matrix K.

Let $\mathrm{K}_{22} = \begin{pmatrix} 2 & 3 & 11 & 5 & 5 \\ 21 & 8 & 3 & 1 & 7 \\ 9 & 25 & 2 & 9 & 0 \\ 7 & 0 & 5 & 28 & 1 \\ 5 & 9 & 13 & 3 & 16 \end{pmatrix}$.

Then $\mathrm{K}_{11} = -\mathrm{K}_{22}$ (mod 29)

$$= \begin{pmatrix} 27 & 26 & 18 & 24 & 24 \\ 8 & 21 & 26 & 28 & 22 \\ 20 & 4 & 27 & 20 & 0 \\ 22 & 0 & 24 & 1 & 28 \\ 24 & 20 & 16 & 26 & 13 \end{pmatrix}$$

$\mathrm{K}_{12} = \mathrm{c}\,(\mathrm{I} - \mathrm{K}_{11})$ (mod 29) Take $\mathrm{c} = 2$.

$$= 2 \begin{pmatrix} -26 & -26 & -18 & -24 & -24 \\ -8 & -20 & -26 & -28 & -22 \\ -20 & -4 & -26 & -20 & 0 \\ -22 & 0 & -24 & 0 & -28 \\ -24 & -20 & -16 & -26 & -12 \end{pmatrix} \text{(mod 29)}$$

$$= \begin{pmatrix} -52 & -52 & -36 & -48 & -48 \\ -16 & -40 & -52 & -56 & -44 \\ -40 & -8 & -52 & -40 & 0 \\ -44 & 0 & -48 & 0 & -56 \\ -48 & -40 & -32 & -52 & -24 \end{pmatrix} \text{(mod 29)}$$

$$= \begin{pmatrix} 6 & 6 & 22 & 10 & 10 \\ 13 & 18 & 6 & 2 & 14 \\ 18 & 21 & 6 & 18 & 0 \\ 14 & 0 & 10 & 0 & 2 \\ 10 & 18 & 26 & 6 & 5 \end{pmatrix}.$$

$$\mathrm{K}_{21} = \frac{\mathrm{I} + \mathrm{K}_{11}}{c} \ (\mathrm{mod}\ 29)$$

$$= \frac{1}{2}\begin{pmatrix} 28 & 26 & 18 & 24 & 24 \\ 8 & 22 & 26 & 28 & 22 \\ 20 & 4 & 28 & 20 & 0 \\ 22 & 0 & 24 & 2 & 28 \\ 24 & 20 & 16 & 26 & 14 \end{pmatrix} (\mathrm{mod}\ 29)$$

$$= \begin{pmatrix} 14 & 13 & 9 & 12 & 12 \\ 4 & 11 & 13 & 14 & 11 \\ 10 & 2 & 14 & 10 & 0 \\ 11 & 0 & 12 & 1 & 14 \\ 12 & 10 & 8 & 13 & 7 \end{pmatrix}$$

Therefore, $\mathrm{K} = \begin{pmatrix} \mathrm{K}_{11} & \mathrm{K}_{12} \\ \mathrm{K}_{21} & \mathrm{K}_{22} \end{pmatrix}$

$$= \begin{pmatrix} 27 & 26 & 18 & 24 & 24 & 6 & 6 & 22 & 10 & 10 \\ 8 & 21 & 26 & 28 & 22 & 13 & 18 & 6 & 2 & 14 \\ 20 & 4 & 27 & 20 & 0 & 18 & 21 & 6 & 18 & 0 \\ 22 & 0 & 24 & 1 & 28 & 14 & 0 & 10 & 0 & 2 \\ 24 & 20 & 16 & 26 & 13 & 10 & 18 & 26 & 6 & 5 \\ 14 & 13 & 9 & 12 & 12 & 2 & 3 & 11 & 5 & 5 \\ 4 & 11 & 13 & 14 & 11 & 21 & 8 & 3 & 1 & 7 \\ 10 & 2 & 14 & 10 & 0 & 9 & 25 & 2 & 9 & 0 \\ 11 & 0 & 12 & 1 & 14 & 7 & 0 & 5 & 28 & 1 \\ 12 & 10 & 8 & 13 & 7 & 5 & 9 & 13 & 3 & 16 \end{pmatrix}$$

The final step involved calculating the encrypted matrix through the multiplication of M_3 and K.

$\mathrm{C} = M_3{}^\circ\, \mathrm{K}$

$$= \begin{pmatrix} 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 16 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 21 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 21 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \circ \begin{pmatrix} 27 & 26 & 18 & 24 & 24 & 6 & 6 & 22 & 10 & 10 \\ 8 & 21 & 26 & 28 & 22 & 13 & 18 & 6 & 2 & 14 \\ 20 & 4 & 27 & 20 & 0 & 18 & 21 & 6 & 18 & 0 \\ 22 & 0 & 24 & 1 & 28 & 14 & 0 & 10 & 0 & 2 \\ 24 & 20 & 16 & 26 & 13 & 10 & 18 & 26 & 6 & 5 \\ 14 & 13 & 9 & 12 & 12 & 2 & 3 & 11 & 5 & 5 \\ 4 & 11 & 13 & 14 & 11 & 21 & 8 & 3 & 1 & 7 \\ 10 & 2 & 14 & 10 & 0 & 9 & 25 & 2 & 9 & 0 \\ 11 & 0 & 12 & 1 & 14 & 7 & 0 & 5 & 28 & 1 \\ 12 & 10 & 8 & 13 & 7 & 5 & 9 & 13 & 3 & 16 \end{pmatrix}$$

$$= \begin{pmatrix} 128 & 336 & 416 & 448 & 352 & 208 & 288 & 96 & 32 & 224 \\ 552 & 440 & 450 & 504 & 384 & 204 & 222 & 388 & 268 & 160 \\ 664 & 126 & 828 & 196 & 916 & 470 & 108 & 316 & 12 & 140 \\ 1064 & 532 & 1092 & 1106 & 273 & 714 & 966 & 714 & 630 & 105 \\ 588 & 117 & 585 & 129 & 696 & 312 & 27 & 309 & 45 & 87 \\ 268 & 323 & 313 & 416 & 260 & 363 & 266 & 273 & 67 & 136 \\ 342 & 201 & 341 & 316 & 156 & 170 & 439 & 175 & 209 & 65 \\ 295 & 176 & 460 & 245 & 470 & 483 & 128 & 153 & 604 & 133 \\ 210 & 42 & 294 & 210 & 0 & 189 & 525 & 42 & 189 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The encoded matrix can be converted into either a row or column matrix and can be transmitted to the recipient through any form of communication channel, while indicating the matrix's order and the matrix utilized in generating the self-invertible matrix.

[10, 128, 336, 416, 448, 352, 208, 288, 96, 32, 224, 552, 440, 450, 504, 384, 204, 222, 388, 268, 160, 664, 126, 828, 196, 916, 470, 108, 316, 12, 140, 1064, 532, 1092, 1106, 273, 714, 966, 714,630, 105, 588, 117, 585, 129, 696, 312, 27, 309, 45, 87, 268, 323, 313, 416, 360, 363, 266, 273, 67, 136, 342, 201, 341, 316, 156, ‘70, 439, 175, 209, 65, 295, 176, 460, 245, 470, 483, 128, 153, 604, 133, 210, 42, 294, 210, 0, 189, 525, 42, 189, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 11, 5 5, 21, 8, 3, 1, 7, 9, 25, 2, 9, 0, 7, 0, 5, 28, 1, 5, 9, 13, 3, 16].

4.2 Decryption

The decryption process involves several steps. Upon receiving the necessary information, the recipient can determine the size of the matrix (n), the encrypted matrix (C), and the matrix K_{22}, which is essential for generating the key matrix.

In this case, n = 10,

$$C = \begin{pmatrix} 128 & 336 & 416 & 448 & 352 & 208 & 288 & 96 & 32 & 224 \\ 552 & 440 & 450 & 504 & 384 & 204 & 222 & 388 & 268 & 160 \\ 664 & 126 & 828 & 196 & 916 & 470 & 108 & 316 & 12 & 140 \\ 1064 & 532 & 1092 & 1106 & 273 & 714 & 966 & 714 & 630 & 105 \\ 588 & 117 & 585 & 129 & 696 & 312 & 27 & 309 & 45 & 87 \\ 268 & 323 & 313 & 416 & 260 & 363 & 266 & 273 & 67 & 136 \\ 342 & 201 & 341 & 316 & 156 & 170 & 439 & 175 & 209 & 65 \\ 295 & 176 & 460 & 245 & 470 & 483 & 128 & 153 & 604 & 133 \\ 210 & 42 & 294 & 210 & 0 & 189 & 525 & 42 & 189 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$K_{22} = \begin{pmatrix} 2 & 3 & 11 & 5 & 5 \\ 21 & 8 & 3 & 1 & 7 \\ 9 & 25 & 2 & 9 & 0 \\ 7 & 0 & 5 & 28 & 1 \\ 5 & 9 & 13 & 3 & 16 \end{pmatrix}.$$

By referring to matrix order 10, the recipient can determine the adjacency matrix M_2 for the PSM labeled path graph P_{10} using the technique outlined in the algorithm for PSM labeling of path graphs.

$$\mathrm{M}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

The self-invertible matrix K is being created by the receiver following the steps outlined in Section 3.1.1.

$$\text{Hence, K} = \begin{pmatrix} \mathrm{K}_{11} & \mathrm{K}_{12} \\ \mathrm{K}_{21} & \mathrm{K}_{22} \end{pmatrix} = \begin{pmatrix} 27 & 26 & 18 & 24 & 24 & 6 & 6 & 22 & 10 & 10 \\ 8 & 21 & 26 & 28 & 22 & 13 & 18 & 6 & 2 & 14 \\ 20 & 4 & 27 & 20 & 0 & 18 & 21 & 6 & 18 & 0 \\ 22 & 0 & 24 & 1 & 28 & 14 & 0 & 10 & 0 & 2 \\ 24 & 20 & 16 & 26 & 13 & 10 & 18 & 26 & 6 & 5 \\ 14 & 13 & 9 & 12 & 12 & 2 & 3 & 11 & 5 & 5 \\ 4 & 11 & 13 & 14 & 11 & 21 & 8 & 3 & 1 & 7 \\ 10 & 2 & 14 & 10 & 0 & 9 & 25 & 2 & 9 & 0 \\ 11 & 0 & 12 & 1 & 14 & 7 & 0 & 5 & 28 & 1 \\ 12 & 10 & 8 & 13 & 7 & 5 & 9 & 13 & 3 & 16 \end{pmatrix}$$

$$\mathrm{C}^\circ\ \mathrm{K} = \begin{pmatrix} 128 & 336 & 416 & 448 & 352 & 208 & 288 & 96 & 32 & 224 \\ 552 & 440 & 450 & 504 & 384 & 204 & 222 & 388 & 268 & 160 \\ 664 & 126 & 828 & 196 & 916 & 470 & 108 & 316 & 12 & 140 \\ 1064 & 532 & 1092 & 1106 & 273 & 714 & 966 & 714 & 630 & 105 \\ 588 & 117 & 585 & 129 & 696 & 312 & 27 & 309 & 45 & 87 \\ 268 & 323 & 313 & 416 & 260 & 363 & 266 & 273 & 67 & 136 \\ 342 & 201 & 341 & 316 & 156 & 170 & 439 & 175 & 209 & 65 \\ 295 & 176 & 460 & 245 & 470 & 483 & 128 & 153 & 604 & 133 \\ 210 & 42 & 294 & 210 & 0 & 189 & 525 & 42 & 189 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\circ \begin{pmatrix} 27 & 26 & 18 & 24 & 24 & 6 & 6 & 22 & 10 & 10 \\ 8 & 21 & 26 & 28 & 22 & 13 & 18 & 6 & 2 & 14 \\ 20 & 4 & 27 & 20 & 0 & 18 & 21 & 6 & 18 & 0 \\ 22 & 0 & 24 & 1 & 28 & 14 & 0 & 10 & 0 & 2 \\ 24 & 20 & 16 & 26 & 13 & 10 & 18 & 26 & 6 & 5 \\ 14 & 13 & 9 & 12 & 12 & 2 & 3 & 11 & 5 & 5 \\ 4 & 11 & 13 & 14 & 11 & 21 & 8 & 3 & 1 & 7 \\ 10 & 2 & 14 & 10 & 0 & 9 & 25 & 2 & 9 & 0 \\ 11 & 0 & 12 & 1 & 14 & 7 & 0 & 5 & 28 & 1 \\ 12 & 10 & 8 & 13 & 7 & 5 & 9 & 13 & 3 & 16 \end{pmatrix}$$

$$= \begin{pmatrix} 40832 & 27392 & 47792 & 40832 & 35264 & 31088 & 29232 & 27276 & 15312 & 15312 \\ 60220 & 40542 & 66416 & 56840 & 51794 & 39266 & 41122 & 39614 & 29522 & 20010 \\ 73776 & 50872 & 68266 & 72180 & 44080 & 39266 & 51562 & 54114 & 33350 & 18734 \\ 114898 & 71050 & 130760 & 101703 & 100506 & 84245 & 72471 & 68005 & 61915 & 34307 \\ 57159 & 39846 & 52374 & 55701 & 34626 & 29067 & 40107 & 42021 & 26709 & 14442 \\ 42717 & 29754 & 48256 & 41557 & 37738 & 29783 & 29941 & 28072 & 17342 & 15254 \\ 37323 & 25636 & 43384 & 36018 & 35756 & 29707 & 24621 & 23419 & 19807 & 12818 \\ 52287 & 31929 & 52171 & 45124 & 40513 & 30392 & 30553 & 35728 & 35633 & 14297 \\ 23751 & 15834 & 29232 & 22533 & 22533 & 23142 & 14007 & 13419 & 14616 & 7917 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

After performing the addition modulo (29) operation, the result obtained is,

$40832 (\text{mod } 29) = 0$, $27392 (\text{mod } 29) = 16$, $47792 (\text{mod } 29) = 0, \ldots, 0 (\text{mod } 29) = 0$.

$$\text{Therefore, } C°K = \begin{pmatrix} 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 16 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 21 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 21 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = M_3$$

The adjacency matrix M_1 for the encoded path graph is derived by subtracting matrix M_2 from matrix M_3.

$M_3 - M_2 =$

$$= \begin{pmatrix} 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 16 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 28 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 28 & 0 & 21 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 21 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 23 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 23 & 0 & 14 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11 & 0 & 12 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15 & 0 & 21 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 21 & 0 \end{pmatrix}$$

$$= M_1$$

The path graph that corresponds to the matrix above has been created. Fig. 8 shows the path graph for the adjacency matrix M_1.

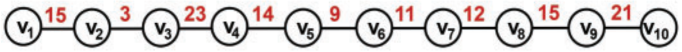

Figure 8: Graph representation of the decrypted adjacency matrix, illustrating the connections between vertices after the decryption process, with edges restored based on the decrypted numerical values

The vertices of the graph shown in Fig. 8 were determined by adding the numerical value of each vertex with its corresponding edge. To ensure that the first vertex is 1, a special character A was added at the beginning. The remaining vertices were calculated as follows:

Here $v_1 = 1$, so $v_2 = (1 + 15)\ (\text{mod } 29) = 16, v_3 = (16 + 3)\ (\text{mod } 29) = 19, v_4 = (19 + 23)(\text{mod } 29) = 13, v_5 = (13 + 14)(\text{mod } 29) = 27, v_6 = (27 + 9)(\text{mod } 29) = 7, v_7 = (7 + 11)(\text{mod } 29) = 18, v_8 = (18 + 12)(\text{mod } 29) = 1, v_9 = (1 + 15)(\text{mod } 29) = 16, v_{10} = (16 + 21)(\text{mod } 29) = 8$.

Thus, the vertices are 1, 16, 19, 13, 27, 7, 18, 1, 16, 8.

The corresponding message is 1→A, P→16, S→19, M→13, 27→ space, 7→G, 18→R, 1→A, 16→P, 8→H, which gives us APSM GRAPH.

4.3 Discussions

In the proposed method, to ensure the lossless nature of the encryption and decryption processes, all matrix operations are performed using modular arithmetic (modulo 29), which guarantees exact integer calculations. The ciphertext matrix consists of integer values, and since no floating-point or real-number arithmetic is involved, there is no risk of rounding errors. Both encryption and decryption are carried out in the finite field $\mathbb{F}_{29}$, ensuring that all operations are precise and deterministic, maintaining the integrity of the system in practical implementations.

The time complexity analysis of the proposed encryption method is $O(n^3)$, primarily due to the matrix multiplication of two $n \times n$ matrices. This time complexity is typical for matrix-based cryptographic schemes, where the encryption and decryption operations depend on matrix algebra. The space complexity of the encryption and decryption processes is $O(n^2)$, as it is determined by the size of the matrices involved. This means that the amount of memory required grows quadratically with the size of the matrices (or the parameter n).

To enhance security and ensure variability in the ciphertext, the matrix K_{22} is randomly generated during each encryption process. This random generation guarantees that K_{22} will have different elements every time, even when encrypting identical plaintexts. As a result, the corresponding key matrix K also changes with each encryption, ensuring that the ciphertext produced for the same plaintext will always be distinct. This approach effectively mitigates the risk of producing identical ciphertexts from repeated encryptions of the same input.

The security of the proposed encryption scheme is quantified based on the key space and its resistance to cryptographic attacks. The key space is determined by the self-invertible matrix K, which due to the modulo 29 operations, results in at least 29^{n^2} possible distinct keys. This large key space ensures strong brute-force resistance. The scheme provides approximately $4.857 \times n^2$ bits of security, comparable to AES-128, by leveraging the computational difficulty of matrix inversion and random key generation. These factors, combined with their resistance to known-plaintext and chosen-plaintext attacks, make the scheme robust against modern cryptanalytic methods.

The security of the proposed encryption scheme is enhanced by the integration of mathematical frameworks that provide resistance to brute-force attacks. This is primarily due to the complexity of matrix inversion and the large key space resulting from the use of randomly generated matrices. The scheme is also designed to resist predictability by utilizing randomly generated key matrices that ensure distinct ciphertexts even for identical plaintexts. These features provide a significant improvement over simple encryption schemes, especially by protecting against both known-plaintext and chosen-plaintext attacks.

The proposed encryption scheme is resistant to several attack vectors, including chosen-plaintext attacks, known-plaintext attacks, and ciphertext-only attacks. The large key space and complexity of matrix inversion make brute-force attacks infeasible. The randomness introduced by the key matrix K ensures that ciphertexts are distinct, even for identical plaintexts, preventing useful patterns in chosen-plaintext attacks. To assess the strength of the encryption scheme, we define and assess several formal security metrics. These metrics include entropy, which measures the randomness and unpredictability of the ciphertext. A high entropy value indicates that the ciphertext is effectively indistinguishable from random data, making it resistant to statistical analysis and ciphertext-only attacks. Additionally, the encryption scheme demonstrates strong bit-flipping resistance. A single bit flip in the plaintext results in a completely unpredictable change in the ciphertext, ensuring that even small alterations in the input significantly affect the output. Finally, we consider the key space size and brute-force resistance, demonstrating that the large key space created by the self-invertible matrix structure provides strong protection against exhaustive search attacks. These metrics confirm that the proposed encryption method exhibits strong cryptographic properties, ensuring a high level of security against a range of potential attacks.

5 Conclusions

In this article, we introduced a novel cryptographic method combining PSM labeling of path graphs, adjacency matrices, and self-invertible matrices. This method leverages the strengths of these mathematical concepts to provide a robust encryption and decryption framework. Our approach enhances security by utilizing graph structures and the unique properties of self-invertible matrices, offering a promising alternative to traditional cryptographic techniques.

Future developments of this model could include using larger finite fields (e.g., modulo 89) to enhance security and key space. Optimizing matrix generation for efficiency and exploring hybrid cryptographic approaches, such as combining with elliptic curve cryptography, could improve both security and performance. Additionally, incorporating post-quantum cryptography techniques would future-proof the system against quantum threats, while hardware acceleration and advanced key management solutions could make the encryption more practical and scalable for real-world applications. We will focus on optimizing the proposed method for practical implementation and exploring its applications in various domains, including secure communications and data protection. Additionally, further research will aim to refine the algorithm and evaluate its performance against emerging cryptographic standards and potential cyber threats.

By integrating advanced mathematical concepts into cryptography, we hope to contribute to the ongoing efforts to secure digital information in an increasingly connected world.

Acknowledgement: We thank the anonymous reviewers for their useful comments that helped to improve the paper.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: P. Amudha: Conceptualization, Methodology, Formal Analysis, Investigation, Writing—Original Draft. J. Jayapriya: Conceptualization, Writing—Review & Editing, Supervision. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data generated or analyzed during this study are included in this published article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Amudha P, Charles Sagayaraj AC, Shantha Sheela AC. An application of graph theory in cryptography. Int J Pure Appl Math. 2018;119(13):375–83.
2. Priyadarsini PLK. A survey on some applications of graph theory in cryptography. J Discrete Math Sci Cryptogr. 2015;18(3):209–17. doi:10.1080/09720529.2013.878819.
3. Supiyanto, Mandowen SA. Advanced Hill cipher algorithm for security image data with the involutory key matrix. J Phy: Conf Ser. 2021;1899(1):012116. doi:10.1088/1742-6596/1899/1/012116.
4. Christensen C. Lester hill revisited. Cryptologia. 2014;38(4):293–332. doi:10.1080/01611194.2014.915260.
5. Rajesh Kumar S, Periyasamy K, Manojkumar S, Poomani S, Karuppiah S. A new Hill cipher algorithm for Image encryption with self-invertible matrix. Int J Pure Appl Math. 2018;119(15):485–92.
6. Panigrahy SK, Acharya B, Jena D. Image encryption using self-invertible key matrix of hill cipher algorithm. In: 1st International Conference on Advances in Computing; 2008; Chikhli, India. vol. 10, p. 349–55.
7. Chen Y, Xie R, Zhang H, Li D, Lin W. Generation of high-order random key matrix for Hill Cipher encryption using the modular multiplicative inverse of triangular matrices. Wirel Netw. 2023;30(6):5697–5707. doi:10.1007/s11276-023-03330-8.
8. Acharya B, Rath GS, Patra SK, Panigrahy SK. Novel methods of generating self-invertible matrix for hill cipher algorithm. Int J Secur. 2007;1(1):14–21.
9. Yunos F, Zulkifli UZ, Ibrahim S, Asbullah MA, Basri W. Exploring self-invertible 3 × 3 matrices for cipher trigraphic polyfunction with distinct encryption keys. Menemui Matematik (Discovering Mathematics). 2024;46(3):30–41.

10. Ching SLP, Yunos F. Effect of self-invertible matrix on cipher hexagraphic polyfunction. Cryptography. 2019;3(2):15. doi:10.3390/cryptography3020015.
11. Sundarayya P, Vara Prasad MG, Satyam KP, Chari Paripurna P, Prasad V. Cryptanalysis of self-invertible key generation for affine hill cipher and digraph affine hill cipher. Int J Eng, Sci Mathem. 2017;6(8):153–62.
12. Yi H. Machine learning method with applications in hardware security of post-quantum cryptography. J Grid Comput. 2023;21(2):19. doi:10.1007/s10723-023-09643-4.
13. Feng W, Yang J, Zhao X, Qin Z, Zhang J, Zhu Z, et al. A novel multi-channel image encryption algorithm leveraging pixel reorganization and hyperchaotic maps. Mathematics. 2024;12(24):3917. doi:10.3390/math12243917.
14. Sakthi Kumar B, Revathi R. An efficient image encryption algorithm using a discrete memory-based logistic map with deep neural network. J Eng Appl Sci. 2024;71(1):41. doi:10.1186/s44147-023-00349-8.
15. Benaissi S, Chikouche N, Hamza R. A novel image encryption algorithm based on hybrid chaotic maps using a key image. Optik. 2023;272(1):17031. doi:10.1016/j.ijleo.2022.170316.
16. Abdellatif N, Manasreh A, Khrisat MS, Zaini HG, Alqadi ZA. Multiple rounds using mixed chaotic keys method for secure message cryptography. ARPN J Eng Appl Sci. 2023;18(8):967–82. doi:10.59018/0423128.
17. Dridi F, El Assad S, El Hadj Youssef W, Machhout M, Lozi R. Design, implementation, and analysis of a block cipher based on a secure chaotic generator. Appl Sci. 2022;12(19):9952. doi:10.3390/app12199952.
18. Vismaya VS, Muni SS, Panda AK, Mondal B. Degn-Harrison map: dynamical and network behaviours with applications in image encryption. Chaos Solit Fract. 2025;192(3):115987. doi:10.1016/j.chaos.2024.115987.
19. Wanying Dai, Xu X, Song X, Li G. Audio encryption algorithm based on chen memristor chaotic system. Symmetry. 2021;14(1):17. doi:10.3390/sym14010017.
20. Qassime AEl, Nhaila H, Bahatti L, Zayrit S. Advanced speech encryption method leveraging a hidden hyperchaotic attractor and its synchronization with robust adaptive sliding mode control. Nonlinear Dyn. 2025;64(8):821. doi:10.1007/s11071-025-10948-0.
21. Gurjar DK, Krishnaa A. Various antimagic labeled graphs from graph theory for cryptography applications. Int J Scient Res Mathem Statist Sci. 2022;9(3):11–8.
22. Amudha P, Jayapriya J. Pair sum modulo labeling of graph. Adv Mathem: Scient J. 2021;10(2):723–7. doi:10.37418/amsj.10.2.4.
23. Amudha P, Jayapriya J, Gowri J. An algorithmic approach for encryption using graph labeling. J Phy: Conf Ser. 2021;1770(1):012072O. doi:10.1088/1742-6596/1770/1/012072.
24. Dawahdeh ZE, Yaakob SN, Razif bin Othman R. A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher. J King Saud Univ-Comput Inf Sci. 2018;30(3):349–55. doi:10.1016/j.jksuci.2017.06.004.