Encryption of Data Using Distance Antimagic Labeling

Anjali Yadav and Minirani S.

***Abstract*—Applications for graph labeling can be explored across numerous engineering fields. Labeling can be efficiently utilised for encryption and decryption methods to provide secure data transfer in the domain of Cryptology. Data security is the process of shielding digital information against fraudulent and unauthorized users against cyber attack or data breach. The concealment of the original plain-text communication is termed encryption, yielding cipher-text, while the process of converting cipher-text back to the original plain-text is referred to as decryption. In contrast to conventional cryptographic techniques, the graph-based approaches employ the structural characteristics of graphs to accomplish strong encryption, offering increased resilience against corruption and assaults. This work uses distance antimagic labeling to formulate algorithms for encryption and decryption, thereby enhancing data transfer security.**

***Index Terms*—encryption, decryption, distance antimagic labeling, RSA algorithm, Hill cipher algorithm.**

I. Introduction

CRYPTOGRAPHY is a fundamental security technique used to protect organizational data and communications from cyber threats by encoding information. It relies on cryptographic algorithms, or ciphers, which are essential for maintaining data security. These algorithms support the generation of cryptographic keys and digital signatures, secure financial transactions, ensure safe web browsing, and verify the authenticity of messages.

At the heart of cryptography are the processes of encryption and decryption, which are vital for preserving the confidentiality and integrity of sensitive data in digital communications. Encryption transforms plaintext into unreadable ciphertext using a secret key and an encryption algorithm. Decryption, in turn, converts the ciphertext back into its original readable form using a decryption algorithm and the appropriate key.

There are two primary types of encryption: symmetric and asymmetric. Symmetric encryption, also known as secret key encryption, uses the same key for both encryption and decryption. It is fast and efficient, making it particularly suitable for encrypting large datasets. Asymmetric encryption, or public key encryption, employs a pair of keys: a public key for encryption and a private key for decryption. This method is especially effective for securing communications and key exchanges over open, untrusted networks, while also ensuring data integrity and authentication.

Magic and antimagic labelings have become valuable techniques in cryptography, introducing innovative approaches to encryption and decryption. By utilizing the distinctive mathematical characteristics of labeled graphs, these methods contribute to increased security and complexity within cryptographic systems. Their structured, yet inherently unpredictable nature makes them well-suited for developing secure communication protocols that uphold data confidentiality and integrity.

Graph labeling, first introduced by Kotzig and Rosa in 1970 [1], refers to the assignment of integers to the vertices, edges, or both, according to defined rules or conditions. In the context of cryptography, labeled graphs are used to represent interactions between various components, where the labels may encode critical information such as cryptographic keys, system states, or transition steps within cryptographic protocols.

The idea of distance magic labeling emerged as a result of study of magic squares. Distance magic labeling was introduced by Vilfred [2] in 1987. Let G be a graph on n vertices and consider a bijection $\chi\colon V(G) \to \{1, 2, \ldots, n\}$ such that there exists a positive integer m and the vertex weight $\varpi(a) = \sum_{c \in N(a)} \chi(c) = m$ for any vertex a in $V(G)$ where $N(a)$ is the set of vertices adjacent to vertex a in $V(G)$. Distance magic labeling naturally leads to distance antimagic labeling given by Kamatchi and Arumugam [3] in which the vertex weights $\varpi(a) \neq \varpi(b)$ for any pair of distinct vertices a and b in $V(G)$.

Definition 1 ([3]). *Consider a graph G with n vertices and let $\chi\colon V(G) \to \{1, 2, \ldots, n\}$ be a bijection. Define vertex weight $\varpi(a) = \sum_{c \in N(a)} \chi(c)$ for any vertex a in $V(G)$. If $\varpi(a) \neq \varpi(b)$ for any pair of distinct vertices a and b in $V(G)$, then χ is said to be distance antimagic. Any graph G that admits such a labeling is said to be distance antimagic.*

Krishnaa [4] employed inner magic and inner antimagic labeling for message encryption, both independently and in combination with the Triple DES algorithm. Gurjar et al. [5] utilized various antimagic labeled graphs, including path and web graphs, as encryption mechanisms. Jegan et al. [6] introduced a technique that uses super-edge antimagic total labeling of Bi-star networks to transform text into edge-weighted graphs for secure encryption. In [7], total edge bimagic mean labeling is applied to evaluate and enhance information security through the RSA algorithm. Furthermore, Vasuki et al. [8] implemented a hybrid encryption and decryption method using the Hill cipher combined with face antimagic labeling on double duplication graphs. Super mean and magic graph labeling is used to encrypt data on social media using Affine Cipher by Sudarsana et al. [9].

Manuscript received February 4, 2025; revised May 20, 2025.

Anjali Yadav is a Ph.D. research scholar at Department of Basic Sciences and Humanities, Mukesh Patel School of Technology and Management, Narsee Monjee Institute of Management Studies, Mumbai, India. (e-mail: anjalis.math26@gmail.com).

Minirani S. is an Associate Professor at Department of Basic Sciences and Humanities, Mukesh Patel School of Technology and Management, Narsee Monjee Institute of Management Studies, Mumbai, India. (e-mail: miniranis@yahoo.com).

This study introduces a set of algorithms designed to encrypt and decrypt twin numbers and secret messages using distance antimagic labeling across various classes of graphs. The first algorithm incorporates the RSA encryption scheme, applying distance antimagic labeling to the splitting graph of a path for secure data transmission. The second algorithm presents a novel combinatorial approach specifically for encrypting and decrypting twin numbers. The third method utilizes the Hill Cipher for secure communication, enabling the encryption and decryption of any confidential message. In the final algorithm, the message is converted into a matrix and encrypted through matrix transformations, enhancing security by leveraging the complexity of matrix operations. Additionally, Python programming is employed to validate the encryption and decryption processes based on the RSA and Hill Cipher algorithms.

II. Main Results

A. Method 1

This method introduces a novel cryptographic framework that integrates the well-established RSA algorithm with distance antimagic labeling applied to the splitting graphs of paths. The proposed approach combines the mathematical strengths of both techniques to create a robust encryption and decryption system. RSA forms the core of the asymmetric encryption process, relying on modular exponentiation and the computational difficulty of prime factorization. Meanwhile, distance antimagic labeling provides a unique vertex-weight distribution, ensuring that the weights of adjacent vertices differ by specific minimum distances.

When applied to the splitting graphs of paths, this hybrid model results in a cryptographic system with enhanced security features. Python is used as the implementation platform due to its rich set of mathematical libraries and its capability to efficiently handle large integers, which are essential in cryptographic computations. This study explores both the theoretical foundations and practical implementation aspects of the proposed method, offering a valuable contribution to the growing field of graph-theoretic approaches in cryptography.

1) Algorithm for Encryption of Secret Message: **Input:** A secret message K and a split graph of path $Sp(P_k)$ which is distance antimagic.

Output: Encrypted cipher text L.

1) Let $G = Sp(P_k)$ be the splitting graph of path on $2k$ vertices with vertex set $V(G) = \{c_i : 1 \leq i \leq k\} \cup \{d_i : 1 \leq i \leq k\}$ and edge set $E(G) = \{c_ic_{i+1} : 1 \leq i \leq (k-1)\} \cup \{d_ic_{i-1} : 2 \leq i \leq k\} \cup \{d_ic_{i+1} : 1 \leq i \leq k-1\}$.
2) Define a function $\chi\colon V(G) \to \{1, 2, \ldots, 2k\}$ such that

$$\chi(c_i) = 2i$$
$$\chi(d_i) = 2i - i$$

where $1 \leq i \leq k$. Clearly, χ is a bijection and the vertex weights are given by

$$\varpi(c_i) = \begin{cases} 7 & :i = 1 \\ 8i - 2 & :2 \leq i \leq k-1 \\ 4k - 5 & :i = k \end{cases}$$

$$\varpi(d_i) = \begin{cases} 4 & :i = 1 \\ 4i & :2 \leq i \leq k-1 \\ 2k - 2 & :i = k \end{cases}$$

Since the vertex weights are unique, the labeling is distance antimagic.

3) Assign the letters of the message K to vertices of the splitting graph of path $Sp(P_k)$.
4) Consider the largest two prime numbers m and n in the vertex weights of splitting graph of path $Sp(P_k)$.
5) Compute $r = m \times n$
$\phi(r) = (m-1) \times (n-1)$
Choose $e < \phi(r)$ such that $gcd(e, \phi(r)) = 1$.
Calculate d such that $d = e^{-1}(mod\phi(r))$.
6) Assign the numeric value of the letters of the message K from $0 - 25$, say $\{p_i, p_2, \ldots, p_{2k}\}$.
7) Compute $c_i = p_i^e(modr)$ for $1 \leq i \leq 2k$.
8) Add obtained c_i with corresponding vertex weights to get encrypted message $L = l_1l_2 \ldots l_{2k}$.

2) Algorithm for Decryption of Secret Message: **Input:** Encrypted message $L = l_1l_2 \ldots l_{2k}$

Output: Secret message $K = k_1k_2 \ldots k_{2k}$.

1) Subtract corresponding vertex weights from l_i to obtain c_i for $1 \leq i \leq 2k$.
2) Compute $p_i = c_i^d(modr)$ for $1 \leq i \leq 2k$.
3) Convert the values p_i to corresponding letters to get the secret message K.

3) Illustration: **Encryption:**

- Let the secret message be $ZACKCODY$.
Consider the distance antimagic splitting graph of path $Sp(P_4)$ and assign the letters of the message to its vertices as shown in Figure 1.
- The two prime numbers in vertex weights of $Sp(P_4)$ are $m = 7$ and $n = 11$.
$r = 7 \times 11 = 77$ and $\phi(r) = 60$.
We choose $e = 7$ such that $gcd(e, \phi(r)) = 1$.
For $d \times 7(mod60) = 1$, we obtain $d = 43$.
- $p_1 = 25$, $p_2 = 0$, $p_3 = 2$, $p_4 = 10$, $p_5 = 2$, $p_6 = 14$, $p_7 = 3$, $p_8 = 24$.
- $c_1 = 25^7(mod77) = 53$, $c_2 = 0^7(mod77) = 0$, $c_3 = 2^7(mod77) = 51$, $c_4 = 10^7(mod77) = 10$, $c_5 = 2^7(mod77) = 51$, $c_6 = 14^7(mod77) = 42$, $c_7 = 3^7(mod77) = 31$, $c_8 = 24^7(mod77) = 73$.
- After adding c_i with corresponding vertex weights, we get the encrypted message $L = 60 \quad 14 \quad 73 \quad 21 \quad 55 \quad 50 \quad 43 \quad 79$.

Decryption:

- Subtracting vertex weights from encrypted message L, we obtain c_i.
- Calculate $p_1 = 53^{43}(mod77) = 25$, $p_2 = 0^{43}(mod77) = 25$, $p_3 = 51^{43}(mod77) = 2$, $p_4 = 10^{43}(mod77) = 10$, $p_5 = 51^{43}(mod77) = 2$, $p_6 = 42^{43}(mod77) = 14$, $p_7 = 31^{43}(mod77) = 3$, $p_8 = 73^{43}(mod77) = 24$.
- Converting p_i to corresponding letters, we get the secret message $K = ZACKCODY$.

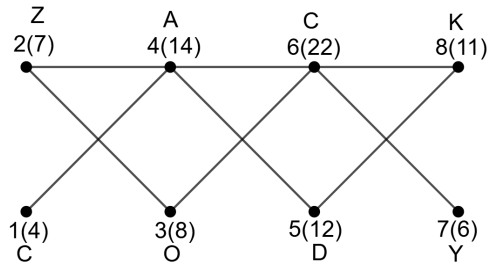

Fig. 1: Illustration of splitting graph of path $Sp(P_4)$ where usual font depicts the label and vertex weights are mentioned in brackets. The letters of the message $ZACKCODY$ are assigned to the vertices.

The Python program implements encryption and decryption of text using a distance antimagic labeling of the splitting graph of a path to create a numeric mapping and then applies the RSA algorithm for secure communication. The Python programming code for implementation of Method 1 is mentioned below:

```
import math
from sympy import isprime
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a
def mod_inverse(e, phi):
    for d in range(3, phi):
        if (d * e) % phi == 1:
            return d
    raise ValueError("Modular inverse does not exist")
def rsa_encrypt_decrypt_method1(message):
    # Step 1: Define the splitting graph of path Sp(P_k) and its distance antimagic labeling
    k = 4
    vertex_weights = {
        'c1': 7, 'c2': 14, 'c3': 22, 'c4': 11,
        'd1': 4, 'd2': 8, 'd3': 12, 'd4': 6
    }
    # Step 2: Find the two largest primes in vertex weights
    primes = [w for w in vertex_weights.values() if isprime(w)]
    m, n = sorted(primes, reverse=True)[:2]  # Largest two primes: m=7, n=11 X
    # Step 3: RSA key generation
    r = m * n
    phi_r = (m - 1) * (n - 1)
    e = 7  # Chosen such that gcd(e, phi_r) = 1
    d = mod_inverse(e, phi_r)
    numeric_values = [ord(ch) - ord('A') for ch in message]  # A=0, B=1, ..., Z=25
    # Assign vertex labels (order: c1, d1, c2, d2, ..., ck, dk)
    vertices = ['c1', 'c2', 'c3', 'c4', 'd1', 'd2', 'd3', 'd4']
    encrypted_message = []
    for i in range(len(numeric_values)):
        p_i = numeric_values[i]
        c_i = pow(p_i, e)% r
        l_i = c_i + vertex_weights[vertices[i]]
        encrypted_message.append(l_i)
        print("Encrypted Message:", encrypted_message)
    # Step 5: Decrypt the message
    decrypted_numeric = []
    for i in range(len(encrypted_message)):
        l_i = encrypted_message[i]
        c_i = l_i - vertex_weights[vertices[i]]
        p_i = pow(c_i, d) % r
        decrypted_numeric.append(p_i)
    decrypted_message = ''.join([chr(p + ord('A')) for p in decrypted_numeric])
    print("Decrypted Message:", decrypted_message)

message = input("enter the message: ")
rsa_encrypt_decrypt_method1(message)

enter the message: HELLOALL
Encrypted Message: [35, 74, 33, 22, 46, 8, 23, 17]
Decrypted Message: HELLOALL
```

Fig. 2: Implementation of Method 1 using Python program.

B. Method 2

This method demonstrates the encryption of two secret numbers using distance antimagic labeling of a centreless wheel graph. Utilizing two numbers in encryption enhances security by enabling the formation of more complex mathematical relationships. This dual-number approach increases the strength of the encryption by adding mathematical complexity and reducing vulnerability. The distance antimagic labeling of centreless wheel graphs is established in [10].

1) Algorithm for Encryption of Secret Twin Numbers:
Input: Two positive numbers P_1 and P_2 having digits k and m respectively where $k \geq m$.
Output: Encrypted labeled centreless wheel graph $CW_k^{'}$.

1) Let $G = CW_k^{'}$ be the centreless wheel graph on k vertices with vertex set $V(G) = \{c_i : 1 \leq i \leq k\} \cup \{d_i : 1 \leq i \leq k\}$ and edge set $E(G) = \{c_ic_{i+1} : 1 \leq i \leq (k-1)\} \cup \{c_kc_1\} \cup \{d_id_{i+1} : 1 \leq i \leq (k-1)\} \cup \{d_kd_1\} \cup \{c_id_i : 1 \leq i \leq k\}$.
2) Define a function $\chi\colon V(G) \to \{1, 2, \ldots, 2k\}$ such that

$$\chi(c_i) = i$$
$$\chi(d_i) = 2k + 1 - i$$

where $1 \leq i \leq k$. Clearly, χ is a bijection and the vertex weights are given by

$$\varpi(c_i) = \begin{cases} 3k+2 & :i = 1 \\ 2k+1+i & :2 \leq i \leq k-1 \\ 2k+1 & :i = k \end{cases}$$

$$\varpi(d_i) = \begin{cases} 3k+1 & :i = 1 \\ 4k+2-i & :2 \leq i \leq k-1 \\ 4k+2 & :i = k \end{cases}$$

Since the vertex weights are unique, the labeling is distance antimagic.

3) Consider the first number $P_1 = s_1 s_2 \dots s_k$ where $s_1, s_2, \dots s_k$ are the k digits of the number P_1.
4) Let the second number be $P_2 = t_1 t_2 \dots t_m$ where $t_1, t_2, \dots t_m$ are the m digits of the number P_2. If $m = k$, then $t_i \geq 0$ and in case $m < k$, then $t_{m+1}, t_{m+2}, \dots, t_k$ are considered blank.
5) Define a function $h : V(CW_k^{'}) \to \mathbb{N}$ such that

$$h(c_i) = \varpi(c_i) + s_i \quad :1 \leq i \leq k$$

$$h(d_i) = \begin{cases} \varpi(d_i) + t_i & :t_i \geq 0 \\ \varpi(d_i) & :t_i \text{ are blank} \end{cases}$$

2) Algorithm for Decryption of Secret Twin Numbers:
Input: Encrypted labeled centreless wheel graph $CW_k^{'}$ with secret numbers encrypted on vertices of the graph.
Output: Two positive numbers P_1 and P_2 .

1) Construct a matrix $A_{k\times 2}$ such that

$$a_{ij} = \begin{cases} h(c_i) & :j = 1, 1 \leq i \leq k \\ h(d_i) & :j = 2, 1 \leq i \leq k \end{cases}$$

2) Construct a weighted matrix $B_{k\times 2}$ where

$$b_{ij} = \begin{cases} \varpi(c_i) & :j = 1, 1 \leq i \leq k \\ \varpi(d_i) & :j = 2, 1 \leq i \leq k \end{cases}$$

3) Compute matrix $G_{k\times 2} = A_{k\times 2} - B_{k\times 2}$
4) Calculate the numbers P_1 and P_2 ignoring the zeros (if any) at the end in P_2.

$$P_1 = s_1 s_2 \dots s_k = g_{11} g_{21} \dots g_{k1}$$
$$P_2 = t_1 t_2 \dots t_k = g_{12} g_{22} \dots g_{k2}$$

3) Illustration: **Encryption:** Let $P_1 = 234765$ and $P_2 = 8921$ be two positive numbers.
Consider a centreless wheel $CW_6^{'}$. The labeling is defined on vertices of this graph as given in step 2 of Algorithm for Encryption which is distance antimagic.

- $k = 6$ and $s_1 = 2$, $s_2 = 3$, $s_3 = 4$, $s_4 = 7$, $s_5 = 6$, $s_6 = 5$.
- $m = 4$ and $t_1 = 8$, $t_2 = 9$, $t_3 = 2$, $t_4 = 1$. Here, t_5 and t_6 are blank.
- Function $h : V(CW_6^{'}) \to \mathbb{N}$ is defined as described step 5 of Algorithm for Encryption. We obtain $h(c_1) = 22$, $h(c_2) = 18$, $h(c_3) = 20$, $h(c_4) = 24$, $h(c_5) = 24$, $h(c_6) = 18$, $h(d_1) = 27$, $h(d_2) = 33$, $h(d_3) = 25$, $h(d_4) = 23$, $h(d_5) = 21$ and $h(d_6) = 26$.
- We obtain an encrypted centreless wheel graph $CW_6^{'}$ with secret numbers encrypted on vertices as shown in the figure 3.

Decryption:

- Consider the centreless wheel graph $CW_6^{'}$ which is distance antimagic and encrypted.
- Compute matrix $A_{6\times 2}$ as per step 1 in algorithm for decryption.

$$A_{6\times 2} = \begin{bmatrix} h(c_1) & h(d_1) \\ h(c_2) & h(d_2) \\ h(c_3) & h(d_3) \\ h(c_4) & h(d_4) \\ h(c_5) & h(d_5) \\ h(c_6) & h(d_6) \end{bmatrix} = \begin{bmatrix} 22 & 27 \\ 18 & 33 \\ 20 & 25 \\ 24 & 23 \\ 24 & 21 \\ 18 & 26 \end{bmatrix}$$

- Construct a weighted matrix $B_{6\times 2}$ as

$$B_{6\times 2} = \begin{bmatrix} \varpi(c_1) & \varpi(d_1) \\ \varpi(c_2) & \varpi(d_2) \\ \varpi(c_3) & \varpi(d_3) \\ \varpi(c_4) & \varpi(d_4) \\ \varpi(c_5) & \varpi(d_5) \\ \varpi(c_6) & \varpi(d_6) \end{bmatrix} = \begin{bmatrix} 20 & 19 \\ 15 & 24 \\ 16 & 23 \\ 17 & 22 \\ 18 & 21 \\ 13 & 26 \end{bmatrix}$$

- We obtain matrix $G_{6\times 2} = A_{6\times 2} - B_{6\times 2}$ as

$$G_{6\times 2} = \begin{bmatrix} 22 & 27 \\ 18 & 33 \\ 20 & 25 \\ 24 & 23 \\ 24 & 21 \\ 18 & 26 \end{bmatrix} - \begin{bmatrix} 20 & 19 \\ 15 & 24 \\ 16 & 23 \\ 17 & 22 \\ 18 & 21 \\ 13 & 26 \end{bmatrix} = \begin{bmatrix} 2 & 8 \\ 3 & 9 \\ 4 & 2 \\ 7 & 1 \\ 6 & 0 \\ 5 & 0 \end{bmatrix}$$

- $P_1 = 234765$ and $P_2 = 8921$ as per step 4 in algorithm for decryption.

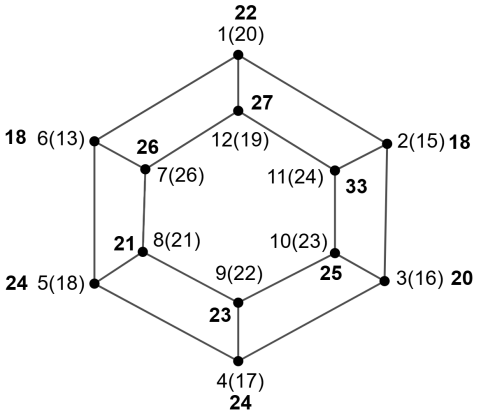

Fig. 3: Illustration of Centreless Wheel graph CW_6 where usual font depicts the label, vertex weights are mentioned in brackets and the encrypted numbers on the vertices are written in bold font.

C. Method 3

This method presents a hybrid encryption-decryption technique that combines the Hill Cipher with Distance Antimagic Labeling of a Wheel Graph. The Hill Cipher, developed by Lester S. Hill in 1929, is a symmetric key encryption method that operates as a polygraphic substitution cipher. It encrypts plaintext by dividing it into blocks of n letters and transforming these blocks using matrix operations. Widely used in fields such as military communications, banking, and computer security, the Hill Cipher relies on linear algebra over modular arithmetic to securely transform messages.

In this approach, additional complexity is introduced through the use of a wheel graph, which is labeled so that each vertex has a unique weight. These weights influence the cipher's structure, enhancing security through a combination of algebraic and graph-theoretic techniques. A Python-based implementation is used to validate the method, ensuring both accurate decryption and correct graph labeling. This integration of mathematical and computational elements makes the approach well-suited for secure cryptographic applications involving sensitive information.

1) Algorithm for Encryption of a Secret Message or Text:
Input: A secret message K with k letters and a wheel graph W_k which is distance antimagic.
Output: Encrypted message L.

1) Let $G = W_k$ be the wheel graph with k letters and vertex set $V(G) = \{c_i : 1 \leq i \leq k\} \cup c$ and edge set $E(G) = \{c_i c_{i+1} : 1 \leq i \leq (k-1)\} \cup \{c_k c_1\} \cup \{cc_i : 1 \leq i \leq k\}$.
2) Define a function $\chi\colon V(G) \to \{1, 2, \ldots, (k+1)\}$ such that

$$\chi(c_i) = i$$
$$\chi(c) = k+1$$

where $1 \leq i \leq k$. Clearly, χ is a bijection and the vertex weights are given by

$$\varpi(c_i) = \begin{cases} 2k+3 & :i = 1 \\ k+1+2i & :2 \leq i \leq k-1 \\ 2k+1 & :i = k \end{cases}$$

$$\varpi(c) = \frac{k(k+1)}{2}$$

Since the vertex weights are unique, the labeling is distance antimagic.
3) Assign each letter of the message K on the k vertices of the wheel graph W_k.
4) Assign values $0 - 25$ to alphabets $A - Z$ respectively on the vertices of the wheel graph W_k.
5) Add corresponding vertex weight and assigned value of the letter for each vertex.
6) Group the obtained values in pairs say M_1, M_2, …, $M_{k/2}$. Add a dummy value 0 at the end for pairing in case k is odd.
7) Find key E which is a 2×2 matrix obtained from smallest vertex weight and its adjacent weights such that E is invertible and its determinant is coprime to 26.
8) Compute $L_i = EM_i \text{mod } 26$ for $1 \leq i \leq k/2$.
9) We get the cipher text $L = L_1 L_2 \ldots L_{k/2}$.

2) Algorithm for Decryption of a Secret Message or Text:
Input: Encrypted message L.
Output: Secret message K.

1) Find modulus inverse E^{-1} of key E.
2) Compute $M_i = E^{-1} L_i \text{mod } 26$ for $1 \leq i \leq k/2$.
3) We obtain pair of values M_1, M_2, …, $M_{k/2}$.
4) Subtract the weight of the vertices from the values obtained in above step and convert the newly obtained values back into corresponding letters to form original secret message K.

3) Illustration: **Encryption:** Let the secret code message be $K = HSBQI$.

- Consider a wheel graph W_5. The labeling is defined on outer vertices of this graph as given in step 2 of Algorithm for Encryption which is distance antimagic. The calculated vertex weights are $\varpi(c_1) = 13$, $\varpi(c_2) = 10$, $\varpi(c_3) = 12$, $\varpi(c_4) = 14$, $\varpi(c_5) = 11$ and $\varpi(c) = 15$ as shown in figure 4.
- Converting the letters of the text to corresponding values from $0 - 25$, we get

H	S	B	Q	I
7	18	1	16	8

- Adding vertex weight to each of the letter of the message K, we obtain

H	S	B	Q	I
20	28	13	30	19

- Group the obtained values in pairs M_1, M_2 and M_3.

M_1	M_2	M_3
20 28	13 30	19 0

- We have the 2×2 matrix key $E = \begin{bmatrix} .15 & 10 \\ 12 & 13 \end{bmatrix}$.
Det$(E) = 75 \neq 0$, so E is invertible. It is easy to see that $E \times E^{-1} = 1$ as

$$E_{2\times 2}^{-1} = \begin{bmatrix} 13/75 & -10/75 \\ -12/75 & 15/75 \end{bmatrix}$$

$$E \times E^{-1} = \begin{bmatrix} 15 & 10 \\ 12 & 13 \end{bmatrix} \begin{bmatrix} 13/75 & -10/75 \\ -12/75 & 15/75 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Also, $GCD(DetE, 26) = GCD(75, 26) = 1$.

- Compute $L_i = EM_i \text{mod } 26$ for $1 \leq i \leq 3$.

$$L_1 = \begin{bmatrix} 15 & 10 \\ 12 & 13 \end{bmatrix} \begin{bmatrix} 20 \\ 28 \end{bmatrix} mod 26 = \begin{bmatrix} 580 \\ 604 \end{bmatrix} mod 26 = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 15 & 10 \\ 12 & 13 \end{bmatrix} \begin{bmatrix} 13 \\ 30 \end{bmatrix} mod 26 = \begin{bmatrix} 495 \\ 546 \end{bmatrix} mod 26 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$L_3 = \begin{bmatrix} 15 & 10 \\ 12 & 13 \end{bmatrix} \begin{bmatrix} 19 \\ 0 \end{bmatrix} mod 26 = \begin{bmatrix} 285 \\ 228 \end{bmatrix} mod 26 = \begin{bmatrix} 25 \\ 20 \end{bmatrix}$$

- Cipher text $L = L_1 L_2 L_3 = 8 \quad 6 \quad 1 \quad 0 \quad 25 \quad 20 = IGBAZU$

Decryption: The cipher text is $L = IGBAZU$.

- We have to first compute E^{-1} to convert cipher text to plain text.

$$E^{-1} = \frac{1}{DetE} mod 26 \begin{bmatrix} 13 & -10 \\ -12 & 15 \end{bmatrix}$$
$$= \frac{1}{75} mod 26 \begin{bmatrix} 13 & -10 \\ -12 & 15 \end{bmatrix}$$
$$\simeq \frac{1}{23} mod 26 \begin{bmatrix} 13 & -10 \\ -12 & 15 \end{bmatrix}$$

We calculate modulo inverse of 23 using naive method and observe that $23 \times 17 = 391 mod 26 = 1$. Therefore, $23^{-1} mod 26 = 17$.

$$E^{-1} = 17 \begin{bmatrix} 13 & -10 \\ -12 & 15 \end{bmatrix} mod 26$$
$$= \begin{bmatrix} 221 & -170 \\ -204 & 255 \end{bmatrix} mod 26 = \begin{bmatrix} 13 & 12 \\ 4 & 21 \end{bmatrix}$$

- Cipher text $L = IGBAZU = 8 \quad 6 \quad 1 \quad 0 \quad 25 \quad 20 = L_1 L_2 L_3$.
Compute $M_i = E^{-1} L_i \text{mod } 26$ for $1 \leq i \leq 3$.

$$M_1 = \begin{bmatrix} 13 & 12 \\ 4 & 21 \end{bmatrix} \begin{bmatrix} 8 \\ 6 \end{bmatrix} mod 26 = \begin{bmatrix} 20 \\ 2 \end{bmatrix} \sim \begin{bmatrix} 20 \\ 28 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 13 & 12 \\ 4 & 21 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} mod 26 = \begin{bmatrix} 13 \\ 4 \end{bmatrix} \sim \begin{bmatrix} 13 \\ 30 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 13 & 12 \\ 4 & 21 \end{bmatrix} \begin{bmatrix} 25 \\ 20 \end{bmatrix} mod 26 = \begin{bmatrix} 565 \\ 520 \end{bmatrix} mod 26 = \begin{bmatrix} 19 \\ 0 \end{bmatrix}$$

- We get the text $M_1M_2M_3 = 20 \quad 28 \quad 13 \quad 30 \quad 19.$
- Subtracting vertex weights from the numeric text obtained above in sequence, we get the secret message $K = 7 \quad 18 \quad 1 \quad 16 \quad 8 = HSBQI$

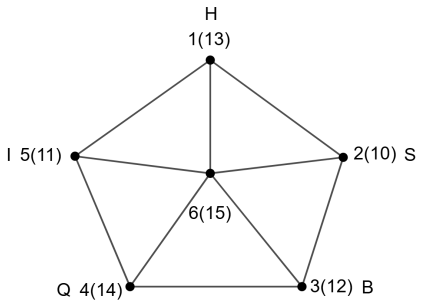

Fig. 4: Wheel graph W_5 where usual font depicts the label and vertex weights are mentioned in brackets. The letters of the message $HSBQI$ are assigned to the vertices.

```
import numpy as np
import math
def hill_cipher_method3(message):
    # Step 1: Define the wheel graph W_k and its distance antimagic labeling
    k = 5  # Example for W_5 as in the paper
    vertex_weights = {
        'c1': 13, 'c2': 10, 'c3': 12, 'c4': 14, 'c5': 11, 'c': 15
    }
    # Step 2: Encrypt the message (example: "HSBQI")
    numeric_values = [ord(ch) - ord('A') for ch in message]  # A=0, B=1, ..., Z=25
    # Add vertex weights (order: c1, c2, c3, c4, c5)
    weighted_values = [
        numeric_values[i] + vertex_weights[f'c{i+1}']
        for i in range(len(numeric_values))
    ]
    weighted_values.append(0)  # Dummy value for pairing (since k=5 is odd)
    # Step 3: Group into pairs for Hill Cipher
    M = [weighted_values[i:i+2] for i in range(0, len(weighted_values), 2)]
    # Step 4: Define the key matrix E (from smallest vertex weight and adjacent weights)
    E = np.array([[15, 10], [12, 13]], dtype=int)  # Ensure integer type
    det_E = int(round(np.linalg.det(E)))  # Explicitly round to integer
    # Check if the key matrix is invertible modulo 26
    if math.gcd(det_E, 26) != 1:
        raise ValueError("Key matrix is not invertible modulo 26")
    else:
        print("Key matrix is invertible modulo 26")
    # Step 5: Encrypt each pair
    L = []
    for pair in M:
        pair_matrix = np.array(pair, dtype=int).reshape(2, 1)
        encrypted_pair = np.dot(E, pair_matrix) % 26
        L.extend(encrypted_pair.flatten().tolist())
    cipher_text = ''.join([chr(int(l) + ord('A')) for l in L])
    print("Encrypted Message:", cipher_text)
    # Step 6: Decrypt the message
    # Compute E^{-1} mod 26 using integer arithmetic
    det_inv = pow(det_E, -1, 26)
    adjugate_E = np.array([[13, -10], [-12, 15]], dtype=int)  # Adjugate of E
    E_inv = (det_inv * adjugate_E) % 26  # Modular inverse of E
    decrypted_pairs = []
    for i in range(0, len(L), 2):
        l_pair = np.array(L[i:i+2], dtype=int).reshape(2, 1)
        decrypted_pair = np.dot(E_inv, l_pair) % 26
        decrypted_pairs.extend(decrypted_pair.flatten().tolist())

    # Subtract vertex weights and convert to letters
    decrypted_numeric = [
        (decrypted_pairs[i] - vertex_weights[f'c{i+1}']) % 26
        for i in range(len(message))
    ]
    decrypted_message = ''.join([chr(p + ord('A')) for p in decrypted_numeric])
    print("Decrypted Message:", decrypted_message)

message = input("enter the message: ")
hill_cipher_method3(message)
```

```
enter the message: HELLO
Key matrix is invertible modulo 26
Encrypted Message: YGXDLO
Decrypted Message: HELLO
```

Fig. 5: Implementation of Method 3 using Python program.

D. Method 4

In this algorithm, the process of encryption and decryption of secret message is demonstrated using matrix algorithm with the help of distance antimagic labeling of a product graph $P_n \boxtimes K_2$ proved in [10]. Matrices serve as an effective instrument in cryptography, facilitating secure communication via encryption, decryption, and key generation. The security and integrity of sensitive data is protected by the mathematical underpinnings that matrices offer for cryptographic operations.

1) Algorithm for Encryption of a Secret Message or Text:
Input: A secret message K with $2k$ letters and a product graph $G = P_k \boxtimes K_2$ which is distance antimagic.
Output: Encrypted matrix L.

1) The vertex weights for graph $G = P_k \boxtimes K_2$ is given below.
Let $V(P_k) = \{a_1, a_2, \ldots, a_k\}$ and $V(K_2) = \{b_1, b_2\}$. We denote the vertex (a_i, b_j) in $G = P_k \boxtimes K_2$ by c_{ij} . Define $\chi\colon V(G) \to \{1, 2, ..., 2k\}$ as

$$\chi(c_{i1}) = \begin{cases} i & \text{if } i \text{ is odd} \\ 2k+1-i & \text{if } i \text{ is even} \end{cases}$$

$$\chi(c_{i2}) = \begin{cases} 2k+1-i & \text{if } i \text{ is odd} \\ i & \text{if } i \text{ is even} \end{cases}$$

where $1 \leq i \leq k$. It is clear that χ is a bijection and vertex weights are calculated as

$$\varpi(c_{i1}) = \begin{cases} 4k+1 & i = 1 \\ 6k+3-i & i \text{ is odd, } 3 \leq i \leq k-1 \\ 4k+2+i & i \text{ is even, } 2 \leq i \leq k-1 \\ 3k+2 & i = k,\ n \text{ is odd} \\ 3k+1 & i = k,\ n \text{ is even} \end{cases}$$

$$\varpi(c_{i2}) = \begin{cases} 2k+2 & i = 1 \\ 4k+2+i & i \text{ is odd, } 3 \leq i \leq k-1 \\ 6k+3-i & i \text{ is even, } 2 \leq i \leq k-1 \\ 3k+1 & i = n,\ k \text{ is odd} \\ 3k+2 & i = n,\ k \text{ is even} \end{cases}$$

G is distance antimagic since each vertex weight is unique.

2) Assign letters of the message to vertices a_i and b_j of the graph $P_k \boxtimes K_2$.
3) Allocate numbers $1-26$ to alphabets $A-Z$.
4) Add vertex weight to corresponding number of the letter in the graph.
5) Define a matrix $B_{2k\times 2k}$ in which diagonal elements are the sum of vertex weight and number of the letter. The remaining elements of the matrix are the absolute difference between the weight of vertex and adjacent vertices.
6) Define a degree matrix $A_{2k\times 2k}$.
7) Compute $F = B \times A$.
8) For encryption of F, use a common key E.
9) The encrypted message in the form of matrix is $L = E \times F$.

2) Algorithm for Decryption of a Secret Message or Text:
Input: Encrypted matrix L.
Output: Secret message K.

1) Calculate $F = E^{-1} \times L$.
2) Compute $B = F \times A^{-1}$.
3) Select the diagonal elements of the matrix B and subtract the corresponding vertex weight from it.
4) Convert the obtained numbers back into alphabets to get secret message K.

3) Illustration: **Encryption:** Let the secret message be $K = HACKER$.

- Consider graph $P_3 \boxtimes K_2$. The labeling is defined on vertices of this graph as given in step 1 of Algorithm for Encryption which is distance antimagic. The calculated vertex weights are $\varpi(c_{11}) = 13$, $\varpi(c_{21}) = 16$, $\varpi(c_{31}) = 11$, $\varpi(c_{12}) = 8$, $\varpi(c_{22}) = 11$ and $\varpi(c_{32}) = 15$ as shown in figure 6.
- Converting the letters of the text to corresponding values from $1-26$, we get

H	A	C	K	E	R
8	1	3	11	5	18

- Adding vertex weight to each of the letter of the message K, we obtain

H	A	C	K	E	R
21	17	14	19	24	28

- Matrix $B_{6\times 6}$ is defined as

$$B = \begin{bmatrix} 21 & 3 & 0 & 5 & 6 & 0 \\ 3 & 17 & 5 & 8 & 3 & 6 \\ 0 & 5 & 14 & 0 & 8 & 1 \\ 5 & 8 & 0 & 19 & 9 & 0 \\ 6 & 3 & 8 & 8 & 24 & 9 \\ 0 & 6 & 1 & 0 & 9 & 28 \end{bmatrix}$$

- Degree matrix $A_{6\times 6}$ is given by

$$A = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

- $F = B \times A =$

$$\begin{bmatrix} 21 & 3 & 0 & 5 & 6 & 0 \\ 3 & 17 & 5 & 8 & 3 & 6 \\ 0 & 5 & 14 & 0 & 8 & 1 \\ 5 & 8 & 0 & 19 & 9 & 0 \\ 6 & 3 & 8 & 8 & 24 & 9 \\ 0 & 6 & 1 & 0 & 9 & 28 \end{bmatrix} \begin{bmatrix} 21 & 3 & 0 & 5 & 6 & 0 \\ 3 & 17 & 5 & 8 & 3 & 6 \\ 0 & 5 & 14 & 0 & 8 & 1 \\ 5 & 8 & 0 & 19 & 9 & 0 \\ 6 & 3 & 8 & 8 & 24 & 9 \\ 0 & 6 & 1 & 0 & 9 & 28 \end{bmatrix}$$

$$= \begin{bmatrix} 63 & 15 & 0 & 15 & 30 & 0 \\ 9 & 85 & 15 & 24 & 15 & 18 \\ 0 & 25 & 42 & 0 & 40 & 3 \\ 15 & 40 & 0 & 57 & 45 & 0 \\ 18 & 15 & 24 & 24 & 120 & 18 \\ 0 & 30 & 3 & 0 & 45 & 84 \end{bmatrix}$$

- For encryption of F, we use a common key E

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

- Encrypted message is $L = E \times F =$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 63 & 15 & 0 & 15 & 30 & 0 \\ 9 & 85 & 15 & 24 & 15 & 18 \\ 0 & 25 & 42 & 0 & 40 & 3 \\ 15 & 40 & 0 & 57 & 45 & 0 \\ 18 & 15 & 24 & 24 & 120 & 18 \\ 0 & 30 & 3 & 0 & 45 & 84 \end{bmatrix}$$

$$= \begin{bmatrix} 105 & 210 & 84 & 120 & 295 & 123 \\ 42 & 195 & 84 & 105 & 265 & 123 \\ 33 & 110 & 69 & 81 & 250 & 105 \\ 33 & 85 & 27 & 81 & 210 & 102 \\ 18 & 45 & 27 & 24 & 165 & 102 \\ 0 & 30 & 3 & 0 & 45 & 84 \end{bmatrix}$$

Decryption: Let the encrypted message be L.

- Compute $F = E^{-1} \times L$ where

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

$$F = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 105 & 210 & 84 & 120 & 295 & 123 \\ 42 & 195 & 84 & 105 & 265 & 123 \\ 33 & 110 & 69 & 81 & 250 & 105 \\ 33 & 85 & 27 & 81 & 210 & 102 \\ 18 & 45 & 27 & 24 & 165 & 102 \\ 0 & 30 & 3 & 0 & 45 & 84 \end{bmatrix}$$

$$= \begin{bmatrix} 63 & 15 & 0 & 15 & 30 & 0 \\ 9 & 85 & 15 & 24 & 15 & 18 \\ 0 & 25 & 42 & 0 & 40 & 3 \\ 15 & 40 & 0 & 57 & 45 & 0 \\ 18 & 15 & 24 & 24 & 120 & 18 \\ 0 & 30 & 3 & 0 & 45 & 84 \end{bmatrix}$$

- Calculate $B = F \times A^{-1}$ where

$$A^{-1} = \begin{bmatrix} 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 \end{bmatrix}$$

$$B = \begin{bmatrix} 63 & 15 & 0 & 15 & 30 & 0 \\ 9 & 85 & 15 & 24 & 15 & 18 \\ 0 & 25 & 42 & 0 & 40 & 3 \\ 15 & 40 & 0 & 57 & 45 & 0 \\ 18 & 15 & 24 & 24 & 120 & 18 \\ 0 & 30 & 3 & 0 & 45 & 84 \end{bmatrix} \begin{bmatrix} 1/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 \end{bmatrix}$$

$$B = \begin{bmatrix} 21 & 3 & 0 & 5 & 6 & 0 \\ 3 & 17 & 5 & 8 & 3 & 6 \\ 0 & 5 & 14 & 0 & 8 & 1 \\ 5 & 8 & 0 & 19 & 9 & 0 \\ 6 & 3 & 8 & 8 & 24 & 9 \\ 0 & 6 & 1 & 0 & 9 & 28 \end{bmatrix}$$

- Consider the diagonal elements of the matrix B. 21 17 14 19 24 28
- Subtracting vertex weights from the above numbers, we get 13 16 11 8 5 18
- Converting the numbers back into letters, we obtain the original message $K = HACKER$.

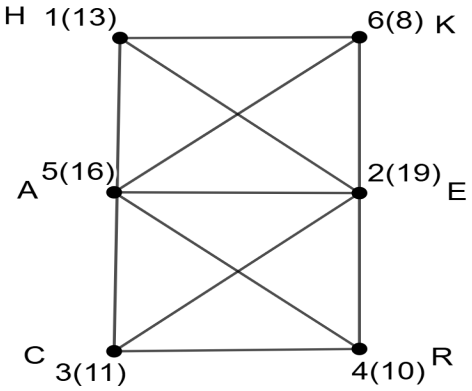

Fig. 6: Illustration of product graph $P_3 \boxtimes K_2$ where usual font depicts the label and vertex weights are mentioned in brackets. The letters of the message $HACKER$ are assigned to the vertices.

III. Conclusion

Our study demonstrates the effectiveness of distance antimagic labeling in developing encryption and decryption algorithms for splitting graphs of paths, cycle-related graphs, and product graphs. Specifically, we utilize distance antimagic labeling on the splitting graph of a path to generate unique numerical values, which serve as cryptographic keys in the RSA algorithm. This graph-based key generation enhances security by ensuring the uniqueness of the keys.

Using a matrix-based approach, we have developed an algorithm for encrypting twin secret numbers, while the Hill Cipher method is applied to encrypt secret messages. Additionally, we introduce a novel matrix method for data encryption, further strengthening the overall cryptographic framework.

The integration of distance antimagic labeling guarantees unique weight assignments, thereby improving the robustness and security of encrypted data. These findings contribute to the broader field of cryptography by presenting a novel approach to secure communication and offer promising directions for future research and practical applications in data protection.

References

[1] A. Kotzig and A. Rosa, "Magic valuations of finite graphs," *Canadian mathematical bulletin*, vol. 13, no. 4, pp. 451–461, 1970.

[2] V. Vilfred, "σ-labelled graph and circulant graphs," *Unpublished Ph. D. thesis, University of Kerala, Trivandrum, India*, 1994.

[3] N. Kamatchi and S. Arumugam, "Distance antimagic graphs," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 64, pp. 61–67, 2013.

[4] A. Krishnaa, "Inner magic and inner antimagic graphs in cryptography," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, no. 6, pp. 1057–1066, 2019.

[5] D. K. Gurjar and A. Krishnaa, "Lexicographic labeled graphs in cryptography," *Advances and Applications in Discrete Mathematics*, vol. 27, no. 2, pp. 209–232, 2021.

[6] R. Jegan, P. Vijayakumar, and K. Thirusangu, "Encrypting a word using super-edge antimagic and super-edge magic total labeling of extended duplicate graphs," *Indian Journal of Computer Science and Engineering*, vol. 13, no. 5, pp. 1559–1565, 2022.

[7] S. V. Shree and S. Dhanalakshmi, "Information security by employing rsa algorithm and graph labeling," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 12, pp. 2563–2568, 2024.

[8] B. Vasuki, L. Shobana, and B. Roopa, "Data encryption using face antimagic labeling and hill cipher," *Math. Stat.*, vol. 10, no. 2, pp. 431–435, 2022.

[9] I. W. Sudarsana, S. Suryanto, D. Lusianti, and N. Putri, "An application of super mean and magic graphs labeling on cryptography system," in *Journal of Physics: Conference Series*, vol. 1763, no. 1. IOP Publishing, 2021, p. 012052.

[10] A. Yadav and S. Minirani, "On distance antimagic labeling of some product graphs," *IAENG International Journal of Applied Mathematics*, vol. 54, no. 10, pp. 2092–2098, 2024.