Jurnal Sisfotek Global
ISSN (Online): 2721 - 3161, ISSN (Print): 2088 – 1762
DOI: http://dx.doi.org/10.38101/sisfotek.v16i2.15934
Vol. 16, No. 1, March 2026, pp. 12-18

Cryptography System Based on SEMT Labeling of Total Path and Its Application for Securing Image File by Using Android Studio

Trisha M A Januaviani[1], Bahrirrudin[2], Nikita[3], I Wayan Sudarsana [4], Nasria Nacong[5], Agusman Sahari[6], Hajar[7], Selvy Musdaifah[8], Moh Ali Akbar[9], Tri Prasetia Ningrum[10]

[1] Sains Data Study Program Tadulako University, Palu, Indonesia, 94118
[2,3,4,5,6,7,8,9,10] Mathematics Study Program Tadulako University, Palu, Indonesia, 94118
E-mail: [1]trishadelheid@gmail.com, [2]bahrirrudin21@gmail.com, [3]untadnikita@gmail.com, [4]sudarsanaiw1972@gmail.com, [5]nasrianacong@gmail.com, [6]agussh@yahoo.com, [7]hajar.20041990@gmail.com, [8]selvymusdalifah@yahoo.com, [9]alidgmatona@gmail.com , [10]thriprstya@gmail.com

ARTICLE HISTORY

Received : July 14, 2025
Revised : February 19, 2026
Accepted : March 27, 2026

KEYWORDS

Android
Cryptography
Image
SEMT Labelling
Total Path

ABSTRACT

Cryptography is one of the mathematical techniques used to address information security issues. The study of information security is essential due to the rapid development of information technology, which increases the risk of unauthorized access to sensitive data. Android smartphones are among the most popular communication devices, offering various features such as text messaging, image sharing, audio, video, and more. Given people use android widespread, data stored on Android devices is highly vulnerable to hacking attempts. Therefore, a cryptographic approach based on graph labeling will be applied to secure images on Android smartphones. This method utilizes the Super Edge Magic Total (SEMT) labeling for total path graphs to encrypt and decrypt image pixels, ensuring the security of images. Furthermore, this technique will be implemented to develop an Android-based application using Android Studio. The study results indicate that total path graphs satisfy the SEMT labeling properties and the developed application functions effectively on Android devices to secure images.

1. Introduction

The rapid development of information technology has driven the widespread use of Android smartphones, which provide various modern communication features such as video calls and instant [1]. In today's fast-paced digital era, the need to ensure data security has become increasingly crucial, as smartphones are highly vulnerable to interception and security threats. Therefore, a reliable mechanism is required to protect sensitive data on Android devices, one of which can be achieved through the implementation of cryptographic methods [2].

Cryptography involves techniques for securing data to maintain its confidentiality, integrity, and availability during transmission or storage. Among various cryptographic approaches, graph-based cryptography has recently emerged as a novel and mathematically enriched method. Previous studies, such as [3], have implemented super mean labeling and super edge-magic total (SEMT) labeling in cryptographic systems, demonstrating the feasibility of leveraging graph labeling for encryption. Other works [4][5] further highlighted the potential of graph labeling in enhancing cryptographic security. Additionally, [6] emphasized the need for secure image protection through cryptography and steganography, while [7]explored magic distinct labelling of graphs as a theoretical foundation for labeling-based cryptography systems. Research in mobile security, such as [8], has also underscored the importance of safeguarding data in Android applications.

Furthermore, recent developments in graph-based cryptography continue to demonstrate its theoretical depth and practical adaptability. Studies such as [9] explore the integration of anti-magic and magic-type labelling to enhance structural randomness in encryption schemes, thereby increasing resistance to brute-force and structural attacks. In addition, [10] proposes a graph-theoretic encryption framework utilizing adjacency matrices and vertex permutations to construct highly nonlinear cipher transformations. The work in [2] further strengthens this direction by analyzing super edge-magic total and super mean labelling techniques as mechanisms for generating complex key spaces derived from graph structures.

Beyond text-based encryption, research has expanded toward multimedia security. The study in [11] investigates image encryption mechanisms within Android environments, emphasizing computational efficiency and adaptability to mobile hardware constraints. Complementarily, [12] presents a comparative evaluation of cryptographic primitives executed on Android devices, highlighting performance-security trade-offs essential for practical

implementation. These findings collectively reinforce the importance of integrating mathematically rich graph labelling techniques with efficient mobile-oriented cryptographic frameworks to develop secure and scalable encryption systems.

However, although graph-based cryptography has been explored in both theoretical and system-based contexts, research specifically applying Super Edge Magic Total (SEMT) labeling for securing multimedia files on mobile platforms remains limited, particularly for image protection on Android devices. Most previous works focus either on the theoretical development of labeling techniques or the implementation of graph-based cryptography in general systems, without emphasis on Android-based applications.

Therefore, this study aims to secure image files by implementing SEMT labeling on total path graphs within the Android Studio environment. This work is intended to bridge the gap between mathematical graph labeling theory and practical mobile-based cryptographic applications, offering a novel approach for enhancing multimedia security on Android platforms.

2. Methods

In this research, the first step is to study the literature by studying matters relating to this research. The next step is to take the pixel value data from the image.

2.1 Super Edge Magic Total Labeling (SEMT)

Let G be a graph with the vertex set V and the edge set E. The edge magic total labeling of G on p vertices and q edges is a bijection **(Definition 1)** [13]

$$\lambda : V(G) \cup E(G) \rightarrow \{1, 2, 3, \cdots, p + q\}$$

Such that for any edge (xy) in G, there are a contant k which satisfy

$$k = \lambda(x) + \lambda(xy) + \lambda(y)$$

The constant k is called the magic constant of G and the graph G is called an edge magic total (EMT) graph (Kotzig and Rosa, 1970) [3]. The bijection λ is said to be the super edge magic total (SEMT) labeling if it has the property that each vertex obtains the smallest label, $\lambda(V) = \{1, 2, \cdots, p\}$. The graph which has a SEMT labeling is called SEMT graph[2].

If given graph G with the vertex set of $V(G)$ and the edge set $E(G)$, the total graph of G is denoted by $T(G)$ is a graph having the vertex $set\ V(G) \cup U(G)$, with $U(G)$ is the vertex set obtained from the addition of vertex from each side $e = v_i, v_j$ in graph G. In the total graph, two vertex are adjacent in T(G) if and only if v_i adjacent with v_j in graph G, $u_i \in U(G)$ adjacent with $u_j \in U(G)$ if edge e_i and edge e_j is incident at the same vertex in graph G, $v_i \in V(T(G))$ adjacent with $u_i \in U(G)$ if vertex v_i is incident with edge e_i . (Pramitasari, 2013) [4] **(Definition 2)**

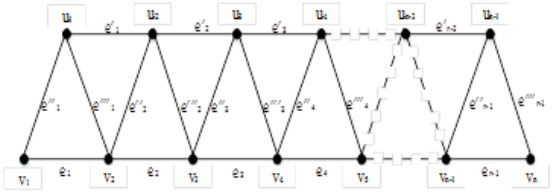

Figure 1: A construction of total path $T(P_n)$ on n vertices

Based on the figure above, it can be denoted the vertex set and the edge set of $T(P_n)$ in the following.

$$V(T(Pn)) = \{vi \,|1 \leq i \leq n\} \cup \{ui \,|1 \leq i \leq n - 1\}$$

$$E\big(T(Pn)\big) = \{ei, e''_i, e'''_i \,|1 \leq i \leq n - 1\} \cup \{e'_i \,|1 \leq i \leq n - 2\}, with\ e_i = v_i v(i + 1), 1 \leq i \leq n - 1, e''_i = v_i u_i, 1 \leq i \leq n - 1$$

2.2 The Super Edge Magic Total Labeling of Total Path

The Super Edge Magic Total Labeling of total path is established using the following theorem [14]:

Theorem 1. The total path $T(P_n)$ is SEMT [3] with the magic constant $k = 6n - 3, for\ n \geq 2$.

Proof: Label the vertices and the edges of $T(P_n)$ as follows:

$$f(v_i) = 2i - 1;\ 1 \leq i < n$$

$$f(u_i) = 2i;\ 1 \leq i < n - 1$$

$$f(e_i) = 6n - 4i - 3;\ 1 \leq i < n - 1$$

$$f(e'_i) = 6n - 4i - 5;\ 1 \leq i < n - 2$$

$$f(e''_i) = 6n - 4i - 2;\ 1 \leq i < n - 1$$

$$f(e'''_i) = 6n - 4i - 4;\ 1 \leq i < n - 1$$

By using the vertex and the edge labels above, we obtain that the magic constant is:

$$k = f(v_i) + f(e_i) + f(v_i + 1), 1 \leq i \leq n - 1$$
$$= f(u_i) + f(e'_i) + f(ui + 1), 1 \leq i \leq n - 2$$
$$= f(v_i) + f(e''_i) + f(u_i), 1 \leq i \leq n - 1$$
$$= f(v_i + 1) + f(e'''_i) + f(u_i), 1 \leq i \leq n - 1$$
$$k = 2i - 1 + 6n - 4i - 3 + 2(i + 1) - 1 = 6n - 3$$
$$= 2i + 6n - 4i - 5 + 2(i + 1) = 6n - 3$$
$$= 2i - 1 + 6n - 4i - 2 + 2i = 6n - 3$$
$$= 2(i + 1) - 1 + 6n - 4i - 4 + 2i = 6n - 3.$$

Thus the total path T(Pn) is a SEMT with the magic constant $k = 6n - 3, for\ n \geq 2$■

2.3 Encryption and Decryption

Data encryption is carried out by calculating the length of the graph using the formula $T(P_{m+2})$ where m is the number of pixels in the image. The next step is to create a random function using SMET labeling function, in this case using 3 random functions. The next step is the encryption process that aims to convert the original image into an encrypted image by using the **equation 1:**

$$C\begin{pmatrix}x_1\\y_1\\z_1\end{pmatrix}=\left(m.p\begin{pmatrix}x_1\\y_1\\z_1\end{pmatrix}+\begin{pmatrix}rand1\times rand2\\rand2\times rand3\\rand1\times rand3\end{pmatrix}\right)modN\ldots(1)$$

For the decryption process, the prime number properties from the following theorems are utilized [15][16]:

Theorem 2. Let P and C each denote plaintext and chipertext where the decimal characters $x \in P$ and $y \in C$. Rand1, Rand2, and Rand3 are a random function of the vertex and edge function of graph labeling. Rand1, Rand2, and Rand3 $\in \lambda$ and $y(mx + \lambda)mod\ N$. If m coprime with N then $x \equiv m^{-1}(y - \lambda)modN$.

Proof: $y \equiv (mx + \lambda)modN \leftrightarrow N\,|(y - (mx + \lambda))$, it means N divides $(y - (mx + \lambda))$. Thus, there is an original number l such that $y - (mx + \lambda) = lN$. Subsequently acquired $mx = y - \lambda - lN$. Because m coprime with N then there are m^{-1} such that $mm^{-1} = m^{-1}modN \equiv 1modN$. Therefore, $x - m^{-1}(y - \lambda) = m^{-1}lN$. Taking $n_1 = m^{-1}l$ is derived the existence of the original number n_1 to be able to declare N divide $x - m^{-1}(y-)$. Thus $xm^{-1}(y-)$modN.

Based on theorem 2, it can be confirmed that eq 1 has an inverse for the description of cipher image to encrypted image. [17] using the **equation 2** :

$$P\begin{pmatrix}x_1\\y_1\\z_1\end{pmatrix}=\left(m^{-1}\times\left(C\begin{pmatrix}x_1\\y_1\\z_1\end{pmatrix}-\begin{matrix}rand1\times rand2\\rand2\times rand3\\rand1\times rand3\end{matrix}\right)\right)modN \qquad (2)$$

The next step, create an image encryption and decryption program based on algorithms that have been compiled in the previous sections using android studio applications.

3. Result and Discussion

Pixel intensity data were extracted from the input image. Each pixel contains three primary color components Red, Green, and Blue (RGB) with each component having an intensity value ranging from 0 to 255. These RGB values represent the fundamental color information of the image and serve as the raw data for subsequent cryptographic operations. The extracted pixel data were then prepared for encryption and decryption, as outlined in the following section.

3.1 Image Encryption via Graph Labeling

3.1.1 Calculate the Length of Graph

To calculate the length of graph, we use n = m + 2 with the graph T(Pn) where m is number of pixels in the image [18]. Look at the following example Suppose that Figure 3 has a 2 × 2 resolution, here is a pixel extraction table of the image.

Table 1. pixel Extraction

266	266
16	16
16	16
266	266
16	16
16	16

Figure 2: Image Before Encryption

Because Figure 3 has a 2 × 2 resolution then the image contains 4 pixel or $m = 4$. So, the total path used is $T(P_{m+2}) = T(P_{4+2}) = T(P_6)$.

3.1.2 Creating Random from the Graph Labeling Function

To produce a good encryption result, a random function of the point and sides function is required. Here the author makes 3 random functions based on Theorem 1 as follows.

Table 2. Theorem 1

Random 1	*Random 2*	*Random 3*
1. $2i - 1$	1. $6n - 4i - 4$	1. $2i - 1$
2. 2i	2. 6n − 4i − 2	2. 6n − 4i − 3
3. 6n − 4i − 3	3. 6n − 4i − 5	3. 6n − 4i − 2
4. 6n − 4i − 5	4. 6n − 4i − 3	4. 2i
5. 6n − 4i − 2	5. 2i	5. 6n − 4i − 4
6. 6n − 4i − 4	6. 2i − 1	6. 6n − 4i − 5
7. Repeat point 1 and so on	7. Repeat point 1 and so on	7. Repeat point 1 and so on

3.1.3 Calculating the Value of the Random Labeling Graph Function

From the process of calculating the length of pixels obtained graph that will be used is graph $T(P_6)$. Then the value obtained from the random function is as follows.

Table 3. Value obtained from the random function

Random 1	*Random 2*	*Random 3*
$1.2 \times 1 - 1 = 1$	$1.\ 6 \times 6 - 4 \times 1 - 4 = 28$	$1.\ 2 - 1 = 1$
$2.2 \times 2 = 4$	$2.\ 6 \times 6 - 4 \times -2 = 26$	$2.\ 2.6 \times 6 - 4 \times 2 - 3 = 25$
$3.6 \times 6 - 4 \times 3 - 3 = 1$	$3.\ 6 \times 6 - 4 \times 3 - 5 = 19$	$3.\ 3.6 \times 6 - 4 \times 3 - 2 = 22$
$4.6 \times 6 - 4 \times 4 - 5 = 15$	$4.\ 6 \times 6 - 4 \times 4 - 3 = 17$	$4.\ 2 \times 4 = 17$
$5.6 \times 6 - 4 \times 5 - 2 = 14$	$5.\ 2 \times 5 = 10$	$5.\ 6 \times 6 - 4 \times 5 - 4 = 12$
$6.6 \times 6 - 4 \times 4 - 4 = 8$	$6.\ 2 \times 6 - 1 = 11$	$6.\ 6 \times 6 - 4 \times 6 - 5 = 7$

3.1.4 Encryption Process

The encryption process will convert the original image into an encrypted image by using **equation 1**

$$C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(m.p\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \begin{pmatrix} rand1 \times rand2 \\ rand2 \times rand3 \\ rand1 \times rand3 \end{pmatrix} \right)$$

$mod\ N; i1,2, \dots, n$

$C = Chiperteks$

$m = Prime\ Relative\ Numbers\ of\ N$

$x_1, y_1, z_1 = Pixel\ Number$

$Rand1\ Rand2, Rand3 =$
$Random\ of\ graph\ labeling\ function$

$$C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = new\ set\ of\ encrypted\ pixels$$

$N = 256$ (Sum of intensity of color)

In this case the relative prime number (m) is 171. The encryption process is as follows

$$C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(171 \times \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix} + \begin{pmatrix} 1 \times 28 \\ 28 \times 1 \\ 1 \times 1 \end{pmatrix} \right) mod\ 256$$

$$= \begin{pmatrix} 43633 \\ 2593 \\ 2566 \end{pmatrix} mod\ 256 = \begin{pmatrix} 133 \\ 33 \\ 6 \end{pmatrix}$$

$$C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(171 \times \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix} + \begin{pmatrix} 4 \times 26 \\ 26 \times 25 \\ 4 \times 25 \end{pmatrix} \right) mod\ 256$$

$$= \begin{pmatrix} 43709 \\ 3215 \\ 2665 \end{pmatrix} mod\ 256 = \begin{pmatrix} 189 \\ 143 \\ 105 \end{pmatrix}$$

$$C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(171 \times \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix} + \begin{pmatrix} 21 \times 29 \\ 19 \times 22 \\ 21 \times 22 \end{pmatrix} \right) mod\ 256$$

$$= \begin{pmatrix} 44004 \\ 2983 \\ 3027 \end{pmatrix} mod\ 256 = \begin{pmatrix} 228 \\ 167 \\ 211 \end{pmatrix}$$

$$C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(171 \times \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix} + \begin{pmatrix} 15 \times 17 \\ 17 \times 8 \\ 15 \times 8 \end{pmatrix} \right) mod\ 256$$

$$= \begin{pmatrix} 43860 \\ 2701 \\ 2685 \end{pmatrix} mod\ 256 = \begin{pmatrix} 84 \\ 141 \\ 125 \end{pmatrix}$$

From the encryption results obtained images as follows.

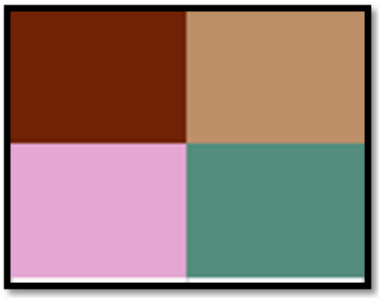

Figure 3. Image after Encryption

3.2.5 Decryption Process

Decryption process is the opposite of the encryption process, which is the process of restoring the encrypted image into the original image using **equation 2**:

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(m^{-1} \times \left(C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} - \begin{matrix} rand1 \times rand2 \\ rand2 \times rand3 \\ rand1 \times rand3 \end{matrix} \right) \right)$$

$mod\ N; i1,2, \dots, n$

$m^{-1} = inverse\ of\ m\ mod\ N$

$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$ = the set of original pixels obtained from the decryption process

For the decryption process used m^{-1} of m in the previous encryption process where $(171)^{-1}$ on modulo 256 is 3. Then obtained the decryption process as follows

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = 3 \times \left(\begin{pmatrix} 113 \\ 33 \\ 6 \end{pmatrix} - \begin{pmatrix} 1 \times 28 \\ 28 \times 1 \\ 1 \times 1 \end{pmatrix} \right) mod\ 256$$

$$= 3 \times \begin{pmatrix} 85 \\ 5 \\ 5 \end{pmatrix} = \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix}$$

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = 3 \times \left(\begin{pmatrix} 189 \\ 143 \\ 105 \end{pmatrix} - \begin{pmatrix} 4 \times 26 \\ 26 \times 25 \\ 4 \times 25 \end{pmatrix} \right) mod\ 256$$

$$= 3 \times \begin{pmatrix} 85 \\ 5 \\ 5 \end{pmatrix} = \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix}$$

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = 3 \times \left(\begin{pmatrix} 228 \\ 167 \\ 211 \end{pmatrix} - \begin{pmatrix} 21 \times 19 \\ 19 \times 22 \\ 21 \times 22 \end{pmatrix} \right) mod\ 256$$

$$= 3 \times \begin{pmatrix} 85 \\ 5 \\ 5 \end{pmatrix} = \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix}$$

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = 3 \times \left(\begin{pmatrix} 84 \\ 141 \\ 125 \end{pmatrix} - \begin{pmatrix} 15 \times 17 \\ 17 \times 8 \\ 15 \times 8 \end{pmatrix} \right) mod\ 256$$

$$= 3 \times \begin{pmatrix} 85 \\ 5 \\ 5 \end{pmatrix} = \begin{pmatrix} 255 \\ 15 \\ 15 \end{pmatrix}$$

From the decryption results obtained images as follows.

Figure 4: Image After Decryption

3.3 Creating an Image Encryption and Decryption Program

Algorithms are made to simplify the making of the program. The following is an encryption algorithm that is based on graph labeling to be implemented in android studio application.

a. Take a picture from the camera or gallery
b. Change plain image to bitmap type
c. Take pixel values from images (height, width, coordinates (x, y)).
d. Take the RGB value from pixels.
e. Counting total path $T(P_{m+2})$
f. Define SEMT labeling function on Total path $T(P_n)$.
   1. $f(v_i) = 2i - 1;\ 1 \le i \le n$
   2. $f(u_i) = 2i;\ 1 \le i \le n - 1$
   3. $f(e_i) = 6n - 4i - 3;\ 1 \le i \le n - 1$
   4. $f(e_i') = 6n - 4i - 5;\ 1 \le i \le n - 2$
   5. $f(e"_i) = 6n - 4i - 2;\ 1 \le i \le n - 1$
   6. $f(e_i''') = 6n - 4i - 4;\ 1 \le i \le n - 1$
g. Create a random of predefined point and side functions.

   Random 1
   7. $f(v_i) = 2i - 1;\ 1 \le i \le n$
   8. $f(u_i) = 2i;\ 1 \le i \le n - 1$
   9. $f(e_i) = 6n - 4i - 3; 1 \le i \le n - 1$
   10. $f(e_i') = 6n - 4i - 5; 1 \le i \le n - 2$
   11. $f(e"_i) = 6n - 4i - 2;\ 1 \le i \le n - 1$
   12. $f(e_i''') = 6n - 4i - 4; 1 \le i \le n - 1$
   13. repeat point 1 and so on

   Random 2
   14. $f(e_1''') = 6n - 4i - 4;\ 1 \le i \le n - 1$
   15. $f(e_i'') = 6n - 4i - 2; 1 \le i \le n - 1$
   16. $f(e_i') = 6n - 4i - 5; 1 \le i \le n - 2$
   17. $f(e_i) = 6n - 4i - 3; 1 \le i \le n - 1$
   18. $f(u_i) = 2i;\ 1 \le i \le n - 1$
   19. $f(v_i) = 2i - 1;\ 1 \le i \le n$
   20. repeat point 1 and so on

   Random 3
   21. $f(e_i''') = 6n - 4i - 4; 1 \le i \le n - 1$
   22. $f(e_i') = 6n - 4i - 5; 1 \le i \le n - 2$
   23. $f(u_i) = 2i;\ 1 \le i \le n - 1$
   24. $f(e_i'') = 6n - 4i - 2; 1 \le i \le n - 1$
   25. $f(v_i) = 2i - 1;\ 1 \le i \le n$
   26. $f(e_i) = 6n - 4i - 3; 1 \le i \le n - 1$
   27. repeat point 1 and so on
h. Decryption Process

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \left(m^{-1} \times \left(C\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} - \begin{matrix} rand1 \times rand2 \\ rand2 \times rand3 \\ rand1 \times rand3 \end{matrix} \right) \right) modN$$

i. Results

$$P\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_1, x_2, \dots, x_n \\ y_1, y_2, \dots, y_n \\ z_1, z_2, \dots, z_n \end{pmatrix}$$

j. Return the RGB and pixel values to an cipher image with bitmap type.
k. Recovery image
l. Done

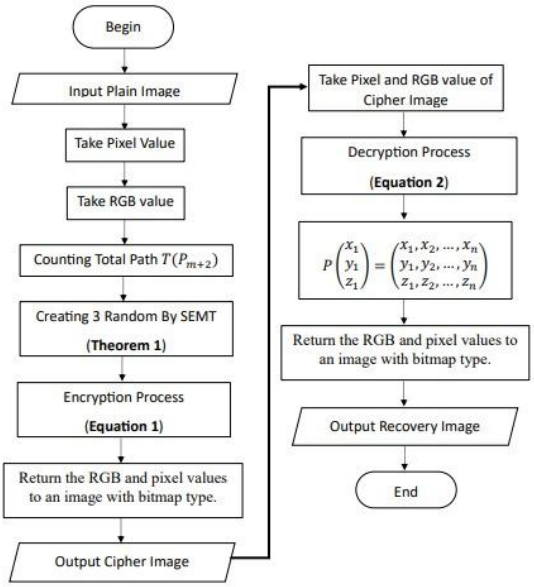

Figure 5. Flowchart Encryption and Decryption

Figure 5 shows the flowchart of the encryption and decryption process, starting from extracting pixel and

RGB values of the plain image, applying SEMT based encryption, and finally recovering the original image through the decryption stage.

3.4 Image Encryption and Decryption Implementation in Android Studio

The decryption program is based on algorithms that have been compiled in the previous sections using Android Studio applications. Initially, there is a plain image to be encrypted. After the encryption process, a cipher image is generated using the SEMT algorithm. This cipher image can then be decrypted to obtain a recovery image, which should closely resemble the original plain image.

Figure 6. Plain Image – Cipher Image – Recovery Image

Figure 6 illustrates the stages of the encryption and decryption process, starting from the plain image, which after encryption becomes the cipher image, and finally the recovery image that successfully resembles the original.

4. Conclusion

Based on the result of research, it can be concluded the total path $\boldsymbol{T(P_n)}$ is a SEMT graph with the magic constant $\boldsymbol{k = 6n - 3}$ for $\boldsymbol{n \geq 2}$. The application of graph labeling on cryptographic method for visual image security can be done by processing color pixel values, pixel colors successfully secured by encryption and decryption process with SEMT graph labeling on total path $(\boldsymbol{P_n})$. The completed application (Apps) based on the method successfully secures the image file.

References

[1] A. Abdul Aziz and A. B. Bawamohiddin, "The Development of Mobile Application Security Through Encryption," Journal of Technology and Humanities, vol. 3, no. 2, pp. 26–36, Dec. 2022, doi: 10.53797/jthkkss.v3i2.5.2022.

[2] H. Anwar, "Mathematical Sciences and Applications Algorithm of Encryption Using Graph Theory," vol. 2, no. 2, 2023.

[3] I. W. Sudarsana, S. A. Suryanto, D. Lusianti, and N. P. A. P. S. Putri, "An application of super mean and magic graphs labeling on cryptography system," J. Phys. Conf. Ser., vol. 1763, no. 1, p. 012052, Jan. 2021, doi: 10.1088/1742-6596/1763/1/012052.

[4] D. K. Gurjar and A. Krishnaa, "Complete Graph and Hamiltonian Cycle in Encryption and Decryption," International Journal of Mathematics Trends and Technology, vol. 67, no. 12, pp. 62–71, Dec. 2021, doi: 10.14445/22315373/ijmtt-v67i12p507.

[5] D. Kumar Gurjar and A. Krishnaa, "Various Antimagic Labeled Graphs from Graph Theory for Cryptography Applications," International Journal of Scientific Research in _____ Research Paper. Mathematical and Statistical Sciences, vol. 9, no. 3, pp. 11–18, 2022, doi: 10.26438/ijsrmss/v9i3.1118.

[6] P. Bagane et al., "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Securing Data in Images Using Cryptography and Steganography Algorithms," 2024. [Online]. Available: www.ijisae.org

[7] G. Xin, X. Xu, C. Zhang, and Y. Zhong, "On magic distinct labellings of simple graphs," J. Symb. Comput., vol. 119, pp. 22–37, Nov. 2023, doi: 10.1016/J.JSC.2023.02.005.

[8] M. Ahmad, V. Costamagna, B. Crispo, F. Bergadano, and Y. Zhauniarovich, "StaDART: Addressing the problem of dynamic code updates in the security analysis of android applications," Journal of Systems and Software, vol. 159, p. 110386, Jan. 2020, doi: 10.1016/J.JSS.2019.07.088.

[9] A. M. Qadir and N. Varol, "A review paper on cryptography," in 7th International Symposium on Digital Forensics and Security, ISDFS 2019, Institute of Electrical and Electronics Engineers Inc., Jun. 2019. doi: 10.1109/ISDFS.2019.8757514.

[10] A. Kareem Ridha, R. S. Jabar, and I. H. ALQinani, "Subject Review:Encryptionof Image in the Android Environment in Various Algorithms," International Journal of Engineering Research and Advanced Technology, vol. 07, no. 06, pp. 22–26, 2021, doi: 10.31695/ijerat.2021.3715.

[11] A. Ometov, K. Zeman, P. Masek, L. Balazevic, and M. Komarov, "A Comprehensive and Reproducible Comparison of Cryptographic Primitives Execution on Android Devices,"

IEEE Access, vol. 9, pp. 54625–54638, 2021, doi: 10.1109/ACCESS.2021.3069627.

[12] A. Ibrahim Hamid and W. Mustafa Abduallah, “StyleGAN2-Stego: Secure Coverless Image Steganography via Latent Space Encoding,” East Journal of Computer Science, vol. 1, no. 4, Aug. 2025, doi: 10.63496/ejcs.vol1.iss4.188.

[13] M. Sindhu and S. Chandra Kumar, “E-SUPER EDGE MAGIC LABELING ON SOME CLASSES OF GRAPHS.”

[14] A. S. Llado, T. Nakamigawa, and G. Ringel, “SUPER EDGE-MAGIC GRAPHS,” 1998.

[15] K. Rozman and P. Šparl, “Distance magic labelings of Cartesian products of cycles,” Discrete Math., vol. 347, no. 10, p. 114125, Oct. 2024, doi: 10.1016/J.DISC.2024.114125.

[16] D. Froncek, P. Paananen, and L. Sorensen, “Group-supermagic labeling of Cartesian products of two even cycles,” Discrete Math., vol. 347, no. 8, p. 113741, Aug. 2024, doi: 10.1016/J.DISC.2023.113741.

[17] D. McQuillan and J. M. McQuillan, “Strong vertex-magic and super edge-magic total labelings of the disjoint union of a cycle with 3-cycles,” Discrete Math., vol. 346, no. 9, p. 113482, Sep. 2023, doi: 10.1016/J.DISC.2023.113482.

[18] X. Zhang, S. Zhang, C. Ye, and B. Yao, “Graphic lattices made by graph felicitous-type labelings and colorings of topological coding,” Discrete Appl. Math. (1979)., vol. 336, pp. 37–46, Sep. 2023, doi: 10.1016/J.DAM.2023.03.023.