

Digital Image Encryption using Composition of RaMSH-1 Map Transposition and Logistic Map Keystream Substitution

Rama Dian Syah¹, Sarifuddin Madenda², Ruddy J. Suhatri³, Suryadi Harmanto⁴
Department of Computer Science, Universitas Gunadarma, Depok, Indonesia^{1,3}
Department of Information Technology, Universitas Gunadarma, Depok, Indonesia^{2,4}

Abstract—Digital communication of multimedia data (text, signal/audio, image, and video) through the internet network has an important role in the era of industrial revolution 4.0 and society 5.0. However, the easiness of exchanging personal and confidential digital information/data has a high risk of being hijacked by irresponsible people. The development of reliable and robust data encryption methods is a solution to this risk. This paper suggests combining modified Henon Map transposition encryption and Logistic Map keystream substitution encryption to create a novel data encryption method (called RaMSH-1). The proposed algorithm simultaneously transposes data positions and substitute data values randomly, as well as having encryption key combination or key space of 1.05×10^{670} . A few images with various sizes and variations of color features, object shapes, and textures have been tested. Based on the results of the analysis of randomness, key sensitivity, and visual, it is evident that the proposed encryption algorithm is resistant to differential attack, entropy attack and brute force attack.

Keywords—Encryption; Decryption; Digital Image; RaMSH-1 Map Transposition; Logistic Map Substitution

I. INTRODUCTION

Data security is an important issue in technological advances. Private data requires data security to protect the data from hacker or cracker [1]. Cryptography is a science related to data security. Cryptography is used to transform plaintext into ciphertext to make the meaning of message difficult to read [2]. Data can be in the form of digital images that can have many meanings. Data security methods such as encryption can be implemented on digital image data.

Transposition and substitution methods are techniques of data encryption. The transposition method in data security is used to randomize data by permuting the position of data sequence from the original data [3]. In the digital image encryption, this method is used to randomize the pixel position coordinate. The new pixel position coordinate is an encrypted image. The substitution methods are used to replace original data with other data [4][5]. This method is used to replace the pixel color intensity value to a different value [6]. The new pixel color intensity value is an encrypted image. These two methods can be combined to strengthen the security of encrypted image.

Transposition and substitution techniques have been used in numerous studies on digital image security. Ping's research

suggests applying Henon Map keystream substitution and transposition for encryption. Henon Map comes in two models which are transposition and generating keystream. Henon Map transposition encryption and decryption functions can only be used for square-size-image $N \times N$ (width = height) [7].

Research by Lone proposes encryption using Random Matrix Affine Cipher (RMAC) transposition, Henon Map transposition, Logistic Map keystream substitution. The encryption process is carried out sequentially, so it takes longer time in encryption process. Research by Lone has limitation which is that the encryption process cannot be implemented on images with size of $M \neq N$ (width \neq height) [8].

Research by Ratna proposes encryption using Cat Map transposition and Henon Map keystream substitution. The encryption process on image with size $M \neq N$ is carried out by adding dummy pixels, so the image size is larger than the original size. The limitation in Ratna's research is that the Cat Map transposition function can only carried out in the image with size $N \times N$ [9].

The Henon Map and Cat Map transposition functions have the advantage of being able to randomize the position of pixel coordinates based on the value of key variable, but these functions have limitations which can only be implemented on image with size $N \times N$. Padding the image with size $M \neq N$ (width \neq height) into $N \times N$ (width = height) can be done but it will enlarge the size of original image, so the encryption process times takes longer.

The Logistic Map functions has the advantage of being able to generate a keystream using two sensitive key variable which can be used to substitute pixel color intensity values, but to strengthen the security of encryption method, the substitution technique can be combined with transposition technique.

This paper proposes a combination of transposition and substitution encryption method called RaMSH-1 (Rama, Madenda, Suhatri, Harmanto). In the transposition encryption section, a modified Henon Map function is proposed, so that it can be applied to all image sizes without padding. Furthermore, the combination of the proposed transposition encryption function with the Logistic Map keystream substitution function aims to increase the encryption key combination so that the probability of finding the key will be smaller.

II. METHODS

This research proposes the RaMSH-1 Map transposition and Logistic Map keystream substitutions functions for encryption and decryption methods. The development of proposed method aims to encrypt and decrypt digital image with all size of image and has a high level of security. This study is focused on 8-bit RGB images.

The RaMSH-1 Map transposition is used to randomize the pixel position coordinates and the Logistic Map is used to generate a keystream that is used to replace the pixel color intensity values. The encrypted image is represented by new pixel position coordinates and color intensity values. To convert the encrypted image back to the original image, the pixel position coordinates, and color intensity values must be restored. The proposed methods for encryption and decryption are shown in Fig. 1.

A. RaMSH-1 Map

RaMSH-1 Map is modification of the Henon Map transposition function. Henon Map function has limitation which can only perform encryption and decryption on image with size $N \times N$ (width = height) because there are a crossing transposition in coordinates $\hat{x} \leftarrow y$ and $\hat{y} \leftarrow x$, so some pixels in the original image with different coordinates (x,y) are transposed to the same coordinate (\hat{x}, \hat{y}) [10]. This cause- the pixels cannot return to the original coordinates when decryption process is carried out. Eq. (1) and (2) are the encryption and decryption functions of Henon Map Transposition.

$$I(\hat{x}, \hat{y}) \leftarrow I(x, y) \text{ where } \begin{cases} \hat{x} = 1 - cx^2 + y \text{ mod } (N) \\ \hat{y} = (x + d) \text{ mod } (N) \end{cases} \quad (1)$$

$$I(x, y) \leftarrow I(\hat{x}, \hat{y}) \text{ where } \begin{cases} \hat{x} = 1 - cx^2 + y \text{ mod } (N) \\ \hat{y} = (x + d) \text{ mod } (N) \end{cases} \quad (2)$$

Modification of the Henon Map to RaMSH-1 Map is carried out by transposing $\hat{x} \leftarrow x$ and $\hat{y} \leftarrow y$. The modulo

process is also carried out according to the size of image at these coordinates. Eq. (3) and (4) are the encryption and decryption function of the RaMSH-1 Map transposition.

$$I(\hat{x}, \hat{y}) \leftarrow I(x, y) \text{ where } \begin{cases} \hat{x} = 1 - cy^2 + x \text{ mod } (M) \\ \hat{y} = (y + d) \text{ mod } (N) \end{cases} \quad (3)$$

$$I(x, y) \leftarrow I(\hat{x}, \hat{y}) \text{ where } \begin{cases} \hat{x} = 1 - cy^2 + x \text{ mod } (M) \\ \hat{y} = (y + d) \text{ mod } (N) \end{cases} \quad (4)$$

The variable keys of RaMSH-1 Map function are c and d with positive integer values. The width of the image is represented by M . The height of the image is represented by N [11]. The pixel coordinates are represented by (x, y) . $I(x,y)$ represents the original image as its value. The value of $I(\hat{x}, \hat{y})$ represents an encrypted image whose pixel position coordinates have been randomly generated. RaMSH-1 Map function can be applied in all image size.

B. Logistic Map

The chaotic function known as the logistic map is highly sensitive to the generation of keystream [12]. The original image's pixel color intensity values are substituted with the keystream value, changing the color intensity value. The Logistic Map function to generate keystream is defined in Eq. (5).

$$g_{n+1} = z \times g_n \times (1 - g_n) \quad (5)$$

The Logistic Map function has two key variables g_0 and z which are real number with range of value $0 < g_0 < 1$ and $3.5 < z \leq 4$. The keystream is calculated by Eq. (6). The keystream is represented as e_n with value from 0 to 255. The value of n is the numbers of pixels in the image where $n = (0, 1, \dots, T-1)$, $T = M \times N$. Keystream substitution is carried out by XOR process in Eq. (7).

$$e_n = \text{round}(|g_n \times 10000|) \text{ mod } 256 \quad (6)$$

$$\hat{I}(\hat{x}, \hat{y}) = e_n \oplus I(\hat{x}, \hat{y})_n \quad (7)$$

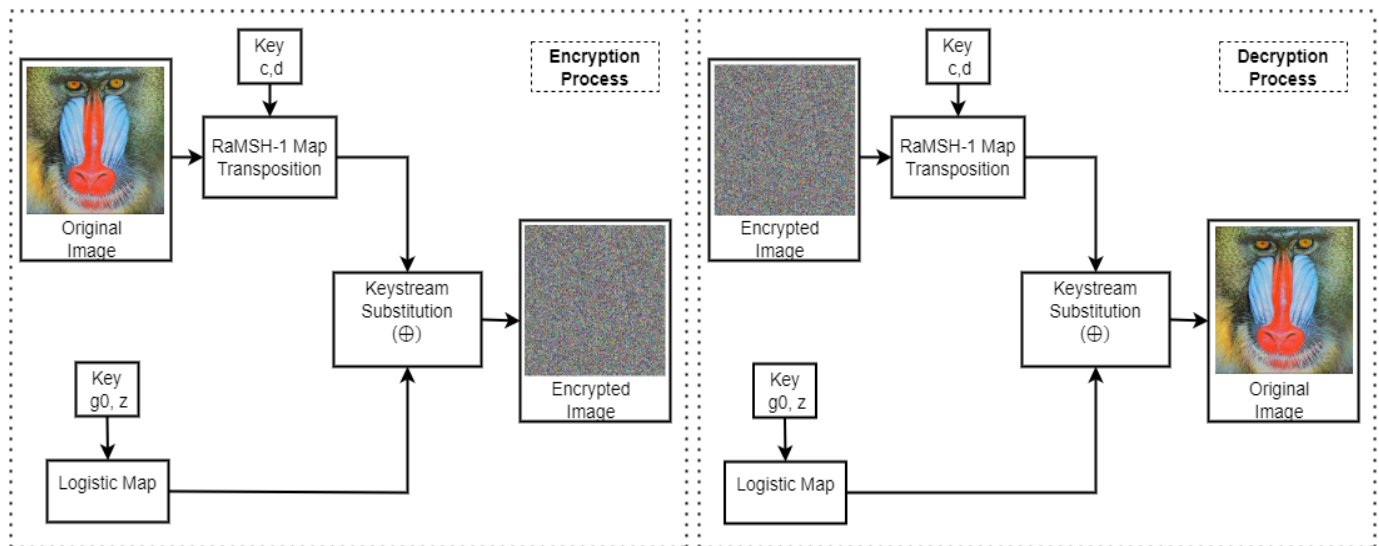


Fig. 1. Proposed methods for encryption and decryption.

C. Proposed Encryption Algorithm

RaMSH-1 Map transposition is used in Eq. (3) to randomize the pixel position coordinates during the encryption process. The encryption algorithm that is proposed is Algorithm 1. Algorithm 2 is used to modify the pixel color intensity values by keystream substitution.

Algorithm 1: Process for Proposed Encryption

Input: Variable c, d, g_0, z , Original Image $I(x, y)$

Output: Cipher Image $I'(x', y')$

$M \leftarrow$ width of image

$N \leftarrow$ height of image

$e_n \leftarrow$ Logistic(N, g_0, z)

```
n ← 0
for x ← 0 to M - 1 do
  for y ← 0 to N - 1 do
    x' ← (1 - cy2 + x) mod(M)
    y' ← (y + d) mod(N)
    I'(x', y') ← en ⊕ I(x, y)
    n ← n + 1
  end for
end for
```

Algorithm 1 step $e_n \leftarrow$ Logistic(N, g_0, z) is to call the Logistic Map function for generating keystream. Steps $x' \leftarrow (1 - cy^2 + x) \bmod(M)$ and $y' \leftarrow (y + d) \bmod(N)$ are randomization of pixel position coordinates by RaMSH-1 Map transposition. Step $I'(x', y') \leftarrow e_n \oplus I(x, y)$ is generating encrypted image using a keystream replacement to change the value of the pixel's color intensity. The encryption technique uses a single loop to carry out the transposition and substitution operations.

D. Logistic Map Keystream Algorithm

The Logistic Map function in Eq. (5) and (6) are applied in the keystream generation process. The Logistic Map keystream algorithm is used in Algorithm 2.

Algorithm 2: Keystream Generation of Logistic Map

```
function Logistic(M, N, g0, z)
T ← M × N
for n ← 0 to T - 1 do
  gn+1 ← z × gn × (1 - gn)
  en ← round(|gn+1 × 10000|) mod(256)
end for
end function
```

Algorithm 2 step $T \leftarrow M \times N$ is a calculation of pixels number in the image. Step g_{n+1} is the Logistic Map function. Step e_n is a function to generate keystreams Logistic Map. Keystreams were generated in an amount equal to the number of pixels in the image.

E. Proposed Decryption Algorithm

The decryption method uses the same procedure as the encryption process, but instead returns values for pixel position coordinates and color intensity. The proposed decryption algorithm is Algorithm 3.

Algorithm 3: Process for Proposed Decryption

Input: Variable c, d, g_0, z , Cipher Image $I'(x', y')$

Output: Original Image $I(x, y)$

$M \leftarrow$ width of image

$N \leftarrow$ height of image

$e_n \leftarrow$ Logistic(N, g_0, z)

```
n ← 0
for x ← 0 to M - 1 do
  for y ← 0 to N - 1 do
    x' ← (1 - cy2 + x) mod(M)
    y' ← (y + d) mod(N)
    I(x, y) ← en ⊕ I'(x', y')
    n ← n + 1
  end for
end for
```

The variable key used in the encryption and decryption processes is same. The distinction between the encryption and decryption processes is that the decryption process uses encrypted images as input, whereas step $I(x, y) \leftarrow e_n \oplus I'(x', y')$ of algorithm 3 restores encrypted images to their original forms.

III. RESULT AND DISCUSSION

The research experiment of proposed algorithm used image data with variety of colors, shape, and texture. The size of image is $M = N: 512 \times 512$ pixel, $M > N: 1500 \times 1000$ pixel, and $M < N: 2850 \times 3200$ pixel. Matlab R2020a, which runs on a computer system with an Intel (R) Core (TM) i7-4790 CPU, a processor clocked at 3.60 GHz, 16 GB of RAM, and the Microsoft Windows 10 operating system, is the software utilized in this study. The analyses used in the experimental are histogram, correlation, NPCR, UACI, decrypted image quality, key sensitivity, and key space.

A. Histogram Analysis

The distribution of pixel color intensity in an image is displayed using a histogram [13][14]. In this study, the pixel color intensity distribution was represented by a histogram, which can demonstrate the effectiveness of pixel randomization in the encrypted image produced by the proposed technique. If the histogram displays a particular pattern, the information in the image can be interpreted easily [15]. In column 3 of Table I, are encrypted images that visually differ from the original image. Columns 4 and 5 show how the histogram of every original image follows a certain pattern, while the histogram of every encrypted image follows a uniform pattern. This shows how the proposed algorithm can change the color intensity value and randomize all pixel coordinates within the encrypted image. It becomes

challenging to interpret the information in an encrypted image [16].

B. Correlation Analysis

The correlation parameter is used to determine how similar two neighboring pixels are to one another in an image's horizontal (I(x,y) and I(x+1,y)), vertical (I(x,y) and I(x,y+1)), and diagonal directions (I(x,y) and I(x+1,y+1)). When the correlation value is near to 1 or -1, two neighboring pixels are either closely associated or have comparable colors [17]. In contrast, if the correlation value is close to 0, either both adjacent pixels have a high degree of randomness or there is low color correlation between them [18]. Eq. (8) functions as the correlation formula, where A and B are similar images that represent two nearby pixels on all three directions. The average value of all the pixels in images A and B is represented by variables \bar{A} and \bar{B} , respectively.

$$r = \frac{\sum_x^M \sum_y^N (A_{xy} - \bar{A})(B_{xy} - \bar{B})}{\sqrt{(\sum_x^M \sum_y^N (A_{xy} - \bar{A})^2)(\sum_x^M \sum_y^N (B_{xy} - \bar{B})^2)}} \quad (8)$$

Columns 5, 6, and 7 of Table II display that the correlation absolute value of the encrypted image ranges from 0.00024 to 0.10269. In columns 2, 3, and 4, the encrypted image has a lower correlation value than the original image. This shows that the proposed algorithm can randomize each bit of information in the encrypted image, making it impossible to crack the data. The level of randomization in pixel color intensities is displayed in Table II. The curve lines of the original images can be seen moving diagonally in columns 2, 3, and 4, which indicates that the two neighboring pixels in the original image have a strong correlation. However, columns 5, 6, and 7 display the random movement of the curve lines of the encrypted image, indicating that the color correlation between

two neighboring pixels in the encrypted image is low. The proposed image encryption algorithm can randomize the position and color intensity of every pixel so that the messages contained in the encrypted image are difficult to read, according to the correlation value and correlation curve in Table II.

C. Decrypted Image Quality Analysis

Peak Signal to Noise Ratio (PSNR) and MSE (Mean Square Error) are used to evaluate the quality of encrypted images. Eq. (9) functions as the PSNR formula, and Eq. (10) is used to calculate the MSE value between the original image and the decrypted image [9]. The quality of the decrypted images is evaluated using an RGB image with a resolution of 2850x3200 pixels.

$$PSNR = 20 \text{Log}_{10} \frac{255}{MSE} \quad (9)$$

$$MSE = \frac{1}{M \times N} \sum_x^M \sum_y^N (I(x,y) - \hat{I}(x,y))^2 \quad (10)$$

The encryption algorithm used by the previous researcher is listed in Table III column 1. Visually, the original image, the encrypted image, and the decrypted image are shown in columns 2, 3, and 4. MSE value and PSNR value are displayed in columns 5 and 6, respectively. There are numerous encrypted image pixels that differ from the original image in rows 2 and 3. It implies that the Henon Map and Cat Map transposition cannot perform the encryption and decryption on an image with a size of $M \neq N$ (width \neq height), resulting in noise in the decrypted image, as seen by values of $PSNR \neq \infty$ and $MSE \neq 0$. The decrypted image in row 4 is identical to the original image with values of $MSE = 0$ and $PSNR = \infty$ [19]. It means that the proposed algorithm can be used to process images of all image size.

TABLE I. ORIGINAL AND ENCRYPTED IMAGE HISTOGRAM


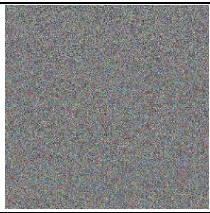
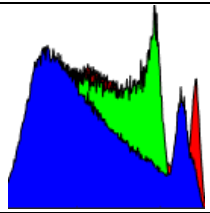
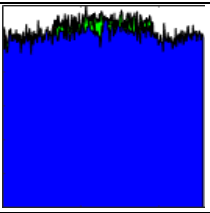


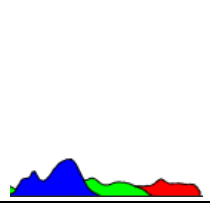
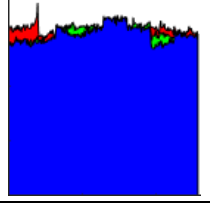
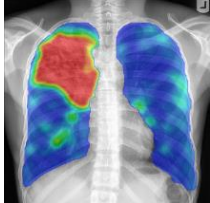

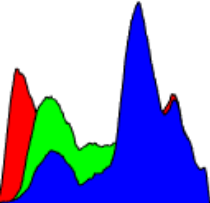
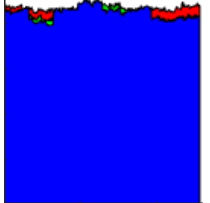
Name	Image		Histogram	
	Original	Encrypted	Original	Encrypted
Baboon (512x512)				
Cat (1500x100)				
Lung (2850x3200)				

TABLE II. ORIGINAL AND ENCRYPTED IMAGE CORRELATION

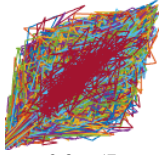
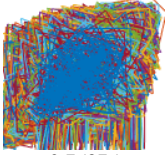
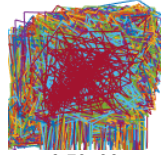
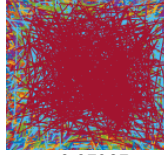
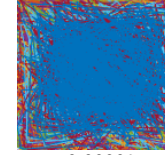
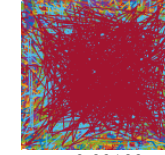
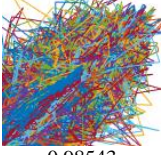
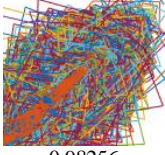
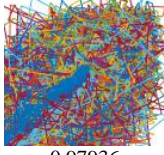
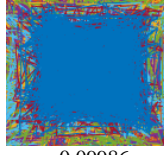
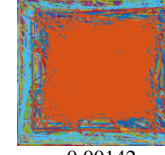
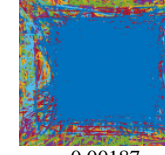
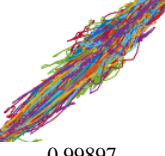
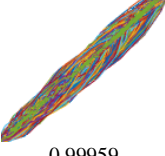
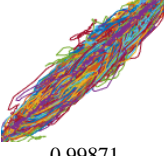


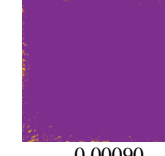
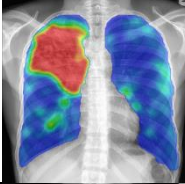
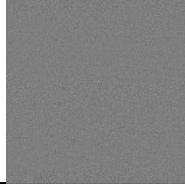
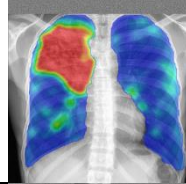
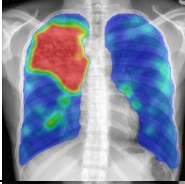
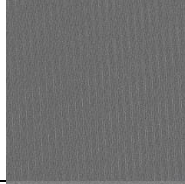
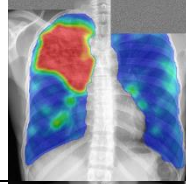
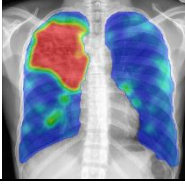


Name	Correlation of Original Image			Correlation of Encrypted Image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Baboon (512×512)	 0.86657	 0.75875	 0.72620	 -0.07237	 -0.00229	 -0.00100
Cat (1500×100)	 0.98543	 0.98256	 0.97936	 -0.09986	 -0.00142	 0.00187
Lung (2850×3200)	 0.99897	 0.99959	 0.99871	 -0.10269	 -0.00024	 -0.00090

TABLE III. DECRYPTED IMAGE ANALYSIS

Encryption Algorithm	Original Image	Encrypted Image	Decrypted Image	MSE	PSNR
Henon Map Transposition & Henon Map Keystream Substitution (Ping, 2018 [7])				1079.84	17.80 dB
Cat Map Transposition & Henon Map Keystream Substitution (Ratna, 2021 [9])				1322.47	16.92 dB
RaMSh-1 Map Transposition & Logistic Map Keystream Substitution (Proposed Algorithm)				0	∞

D. NPCR and UACI Analysis

The NPCR (Number of Pixel Change Rate) parameter measures how differently colored pixels in the original image affect the outcomes of the encrypted image [20][21]. The UACI parameter assesses the average pixel change percentage in the encrypted image. The security level of an algorithm's resistance to differential attack is shown by the NPCR and UACI value [7][22].

Eq. (11) and (12) are the formulas of NPCR and UACI where C1 and C2 are two encrypted images. The D_{xy} coefficient is the similarity value of pixel in same coordinate of both C1 and C2 images. The $C1_{xy}$ and $C2_{xy}$ are color intensity values of the C1 and C2 at the (x,y) position.

$$NPCR = \frac{1}{M \times N} \sum_x^M \sum_y^N D_{xy} \times 100\% \quad (11)$$

$$UACI = \frac{1}{M \times N} \sum_x^M \sum_y^N \frac{|C1_{xy} - C2_{xy}|}{255} \times 100\% \quad (12)$$

The largest UACI value is 33.37521%, while the largest NPCR value is 99.61014%, according to Table IV. The proposed algorithm is resistant to differential attack [8] and has sensitivity of pixel changes in original image effects to encrypted image results since both values are above the standard 99.6% of NPCR value and 33.3% of UACI value.

E. Entropy Analysis

The entropy parameter calculates the minimal average number of bits required to decode a symbol from a sequence of

bits. The entropy value displays the effectiveness of the algorithm's security level and the degree of unpredictability in the encrypted image [23]. Entropy is calculated using Eq. (13), where P_i is the chance that a pixel with the value of i will appear. Entropy can have a maximum value of 8 [8][24].

$$H = \sum_{i=0}^{255} P_i \log_2 \left(\frac{1}{P_i} \right) \quad (13)$$

The highest entropy value, according to Table IV, is 7.99980. It shows that the proposed algorithm can randomize the pixel coordinates and change the pixel color intensity value in the encrypted image and is resistant to entropy attacks [14].

TABLE IV. NPCR, UACI, ENTROPY OF ENCRYPTED IMAGE

Name	NPCR (%)	UACI (%)	Entropy
Baboon	99.61014	33.02773	7.99822
Cat	99.60851	33.29189	7.99964
Lung	99.60999	33.37521	7.99980

F. Key space Analysis

The proposed algorithm is composition of Logistic Map substitution and RaMSh-1 Map transposition. The key variables are c , d , g_0 , and z . The range of values $c, d \in \mathbb{Z}^+$. The range of values for z and g_0 are $3.5699 < r \leq 4$ and $0 < g_0 < 1$, respectively, where $g_0, r \in \mathbb{R}^+$ [8]. In Matlab, the largest integer number value is $2^{64} \approx 1.8 \times 10^{19}$ and the largest real number value is 1.8×10^{308} . The default mantissa value in floating-point is 10^{15} [23]. Table V shows the proposed algorithm's key space.

TABLE V. KEY SPACE

Function	Variable Key	Key Space
RaMSh-1 Map	$c, d \in \mathbb{Z}^+$	3.24×10^{38}
Logistic Map	$g_0 \in [0, 1]; z \in [3.5699, 4]$	3.24×10^{631}
Proposed Algorithm	$c, d \in \mathbb{Z}^+$ $g_0 \in [0, 1]; z \in [3.5699, 4]$	1.05×10^{670}

The proposed algorithm's key space is $(1.8 \times 10^{19}) \times (1.8 \times 10^{19}) \times (1.8 \times 10^{308}) \times (1.8 \times 10^{308}) \times (10^{15}) \approx 1.05 \times 10^{670}$. A computer that can perform 10^{24} computations in a second is used to simulate testing every key combination [22]. It would take 3.33×10^{639} years to complete the vast array of computations through the computer in a single year $(1.05 \times 10^{670}) \div (10^{24}) \times 365$ (days) $\times 24$ (hour) $\times 60$ (min) $\times 60$ (s) = 3.33×10^{639} . This amount of time for attempting every combination is sufficient to fend off a brute force attack [23][25].


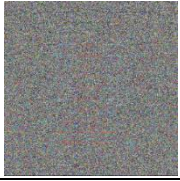

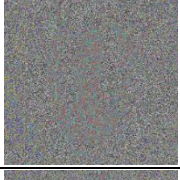

G. Key Sensitivity Analysis

The proposed algorithm using four variable keys. Key of c and d are positive integer numbers. Key of g_0 and z are positive real numbers. The decrypted key value with small changes on decrypted process will affect the decrypted image. The key sensitivity can be seen in Table VI.

Table VI row 2 shows the decrypted keys for decryption process are $c = 20, d = 30, g_0 = 0.1$, and $z = 3.6$. Each key is changed with a very small value change in key values. Rows 3,

4, 5, and 6 show the decrypted image still unreadable. In row 5 shows the key sensitivity of g_0 reached 10^{-16} which indicate the proposed algorithm is robust against brute force attack.

TABLE VI. KEY SENSITIVITY ANALYSIS

Key Sensitivity	Key Value	Decrypted Image
Decryption Key	$c = 20; d = 30$ $g_0 = 0.1; z = 3.6$	
Sensitivity of Decryption Key c	$c = 21; d = 30$ $g_0 = 0.1; z = 3.6$	
Sensitivity of Decryption Key d	$c = 20; d = 31$ $g_0 = 0.1; z = 3.6$	
Sensitivity of Decryption Key g0	$c = 20; d = 30$ $g_0 = 0.1 + 10^{-16};$ $z = 3.6$	
Sensitivity of Decryption Key z	$c = 20; d = 30$ $z = 3.6 + 10^{-15};$ $g_0 = 0.1$	

IV. CONCLUSIONS

The Henon Map transposition function is modified in the RaMSh-1 Map transposition. The proposed algorithm can be applied for all image size without padding. The key variable of proposed algorithm generated a key space of 1.05×10^{670} . The entropy values reached 7.99980 out of 8 as perfect entropy value. The correlation value for all encrypted images is very close to 0, indicating that there is no correlation between the color intensity of neighboring pixels. Entropy and correlation analyses show that the encrypted image's information is all scrambled, which makes it hard to decipher. The results of the key sensitivity reached 10^{-16} which indicate the proposed algorithm is very sensitive. The UACI value of 33.37521 and the NPCR value of 99.61014 respectively show that the proposed algorithm is resistant to differential attack.

V. FUTURE WORK

The proposed algorithm's weakness is that it only uses four variable keys. The algorithm's key space can be expanded. The data security can be improved by combining RaMSh-1 Map transposition, Logistic Map substitution, and Cat Map transposition.

REFERENCES

- [1] M. K. Hasan et al., "Lightweight Cryptographic Algorithms for Guessing Attack Protection in Complex Internet of Things Applications," *Complexity*, vol. 2021, pp. 1–9, 2021, doi: 10.1155/2021/5540296.
- [2] B. K. Yakti, S. A. Sudiro, S. Madenda, and S. Harmanto, "Hardware Implementation Method of Secret Data Security on Fpga Based on Zig-Zag Map Encryption and Stegano Algorithms," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 17, pp. 5572–5582, 2022.
- [3] K. Renuka and G. N. Harshini, "Analysis and Comparison of Substitution and Transposition Cipher," *Int. J. Res. Anal. Rev.*, vol. 6, no. 2, pp. 549–555, 2019.
- [4] P. Poonia and P. Kantha, "Comparative Study of Various Substitution and Transposition Encryption Techniques," *Int. J. Comput. Appl.*, vol. 145, no. 10, pp. 24–27, 2016, doi: 10.5120/ijca2016910783.
- [5] R. S. Navale, A. N. Jalgeri, and B. B. Jagadale, "Survey on various substitution techniques for Cryptography," *Int. J. Res. Dev. Technol.*, vol. 7, no. 4, pp. 613–616, 2017.
- [6] Y. P. K. Nkandeu and A. Tiedeu, "An image encryption algorithm based on substitution technique and chaos mixing," *Multimed. Tools Appl.*, vol. 78, no. 8, pp. 10013–10034, 2019, doi: 10.1007/s11042-018-6612-2.
- [7] P. Ping, F. Xu, Y. Mao, and Z. Wang, "Designing permutation substitution image encryption networks with Henon map," *Neurocomputing*, vol. 283, pp. 53–63, 2018, doi: 10.1016/j.neucom.2017.12.048.
- [8] P. N. Lone, D. Singh, and U. H. Mir, "A novel image encryption using random matrix affine cipher and the chaotic maps," *J. Mod. Opt.*, vol. 68, no. 10, pp. 507–521, 2021, doi: 10.1080/09500340.2021.1924885.
- [9] A. A. P. Ratna et al., "Chaos-based image encryption using Arnold's cat map confusion and Henon map diffusion," *Adv. Sci. Technol. Eng. Syst.*, vol. 6, no. 1, pp. 316–326, 2021, doi: 10.25046/aj060136.
- [10] R. D. Syah, S. Madenda, R. J. Suhatrik, and S. Harmanto, "Hybrid Digital Image Cryptography Using Composition of Henon Map Transposition and Logistic Map Substitution," in *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, 2022, pp. 1–6, doi: 10.1109/ICOSNIKOM56551.2022.10034926.
- [11] V. Tyagi, *Understanding Digital Image Processing*. CRC Press, 2018.
- [12] A. T. Ruslan, Marwan, and Q. Aini, "Behavior of logistic map and some of its conjugate maps," *AIP Conf. Proc.*, vol. 2641, no. 1, p. 20002, 2022, doi: 10.1063/5.0115103.
- [13] H. Kaur and N. Sohi, "A Study for Applications of Histogram in Image Enhancement," *Int. J. Eng. Sci.*, vol. 06, no. 06, pp. 59–63, 2017, doi: 10.9790/1813-0606015963.
- [14] A. Benlashram, M. Al-ghamdi, R. Altalhi, and P. Kaouter, "A novel approach of image encryption using pixel shuffling and 3D chaotic map A novel approach of image encryption using pixel shuffling and 3D chaotic map," in *Journal of Physics: Conference Series*, 2020, vol. 1447, doi: 10.1088/1742-6596/1447/1/012009.
- [15] N. Munir, M. Khan, A. Al Karim Haj Ismail, and I. Hussain, "Cryptanalysis and Improvement of Novel Image Encryption Technique Using Hybrid Method of Discrete Dynamical Chaotic Maps and Brownian Motion," *Multimed. Tools Appl.*, vol. 81, no. 5, pp. 6571–6584, 2022, doi: 10.1007/s11042-021-11810-2.
- [16] H. Gao and W. Zeng, "Image compression and encryption based on wavelet transform and chaos," *Comput. Opt.*, vol. 43, no. 2, pp. 258–263, 2019, doi: 10.18287/2412-6179-2019-43-2-258-263.
- [17] R. D. Syah and R. J. Suhatrik, "Digital Image Cryptography Using Combination of Arnold's Cat Map and Bernoulli Map Based on Chaos Theory," *Int. Res. J. Adv. Eng. Sci.*, vol. 4, no. 2, pp. 258–262, 2019, doi: 10.5281/zenodo.3153337.
- [18] S. Sabir and V. Guleria, "Multilayer color image encryption using random matrix affine cipher, RP2DFrHT and 2D Arnold map," *Multimed. Tools Appl.*, vol. 80, pp. 27829–27853, 2021, doi: 10.1007/s11042-021-11003-x.
- [19] M. T. Suryadi, M. Y. T. Irsan, and Y. Satria, "New modified map for digital image encryption and its performance," *J. Phys. Conf. Ser.*, vol. 893, no. 1, 2017, doi: 10.1088/1742-6596/893/1/012050.
- [20] Y. Liu and Y. C. Ko, "Image Processing Method Based on Chaotic Encryption and Wavelet Transform for Planar Design," *Adv. Math. Phys.*, vol. 2021, pp. 1–12, 2021, doi: 10.1155/2021/7511245.
- [21] Y. Chen, S. Xie, and J. Zhang, "A Hybrid Domain Image Encryption Algorithm Based on Improved Henon Map," *Entropy*, vol. 24, no. 2, pp. 1–28, 2022, doi: 10.3390/e24020287.
- [22] S. Kanwal et al., "An Effective Color Image Encryption Based on Henon Map, Tent Chaotic Map, and Orthogonal Matrices," *Sensors*, vol. 22, no. 12, p. 4359, 2022, doi: 10.3390/s22124359.
- [23] L. Zhang, L. Zhang, and L. Zhang, "Application Research of Digital Media Image Processing Technology Based on Wavelet Transform," *J Image Video Proc*, vol. 138, no. 2018, 2018, doi: 10.1186/s13640-018-0383-6.
- [24] Y. A. Hamza, "Highly Secure Image Steganography Approach Using Arnold's Cat Map and Maximum Image Entropy," in *Proceedings of the International Conference on Information and Communication Technology*, 2019, pp. 134–138, doi: 10.1145/3321289.3321323.
- [25] M. C. Alipour, B. D. Gerardo, and R. P. Medina, "A secure image encryption architecture based on pseudorandom number generator and chaotic logistic map," *ACM Int. Conf. Proceeding Ser.*, pp. 154–159, 2019, doi: 10.1145/3352411.3352436.