



Chaos-Based Image Encryption Using DNA Manipulation and a Modified Arnold Transform

Marwan A. Fetteha^{1(✉)}, Wafaa S. Sayed², Lobna A. Said¹,
and Ahmed G. Radwan^{2,3}

¹ Nanoelectronics Integrated Systems Center (NISC),
Nile University, Giza 12588, Egypt
M.Ahmed2129@nu.edu.eg

² Engineering Mathematics Department, Faculty of Engineering,
Cairo University, Giza 12613, Egypt

³ School of Engineering and Applied Sciences, Nile University, Giza 12588, Egypt

Abstract. Digital images, which we store and communicate everyday, may contain confidential information that must not be exposed to others. Numerous researches are interested in encryption, which protects the images from ending up in the hands of unauthorized third parties. This paper proposes an image encryption scheme using chaotic systems, DNA manipulation, and a modified Arnold transform. Both DNA manipulation and hyperchaotic Lorenz system are utilized in the substitution of the images' pixel values. An additional role of hyperchaotic Lorenz system is that it generates the random numbers required within the DNA manipulation steps. DNA cycling is implemented based on simple DNA coding rules and DNA addition and subtraction rules with modulus operation. The modified Arnold transform alters the pixels' positions, where it guarantees effective pixel permutation that never outputs the same input pixels arrangement again. The proposed design is simple and amenable for hardware realization. Several well established performance evaluation tests including statistical properties of the encrypted image, key space, and differential attack analysis were conducted for several images. The proposed scheme passed the tests and demonstrated good results compared to several recent chaos-based image encryption schemes.

Keywords: Arnold transform · Chaos · DNA · Image encryption

1 Introduction

Communication methods have undergone significant changes in the recent few decades due to the quick development of computer and network technology. The need for secure communication of media and exchanged information has gradually developed [1]. Specifically, image encryption has been the topic of numerous researches to protect the user's privacy [2]. The strong correlation and redundancy between neighbouring pixels of an image require devising new encryption schemes rather than the typical ones [3].

Chaotic systems are good candidates for image encryption systems because of their pseudorandomness, initial value sensitivity, parameter sensitivity, and unpredictability, among other qualities, which increase the security level [4–6]. Both Deoxyribonucleic acid (DNA) encoding and Arnold permutation have appeared in recent works as well. In [7], an image encryption algorithm based on bit-level Arnold transform and hyperchaotic maps was proposed. The algorithm divides the grayscale image into 8 binary images. Then, a chaotic sequence is used to shift the images. Afterwards, Arnold transform is applied. Finally, image diffusion is applied using the hyper chaotic map. The system requires image division, which increases the system’s complexity and may halt it from being optimized to applicable hardware design. In [6], Luo et al. used double chaotic systems, where two-dimensional Baker chaotic map is used to set the state variables and system parameters of the logistic chaotic map. In [8], Ismail et al. developed a generalised double humped logistic map, which is used in gray scale image encryption. In [9], a chaotic system and true random number generator were utilized for image encryption. The presence of both the chaotic system and true random number generator increases the system’s complexity making it less suitable for hardware implementation. In [1], a plaintext-related encryption scheme that utilises two chaotic systems and DNA manipulation was presented. The system depends on the values of some pixels for the encryption process, which threatens image restoration if they are changed.

This paper proposes an image encryption algorithm that uses hyperchaotic Lorenz system, an optimized DNA manipulation system and a new method for applying Arnold transform, which is more suitable for encryption applications. The rest of the paper is organized as follows: Sect. 2 provides a brief explanation of the utilized methods. Section 3 demonstrates the proposed encryption and decryption algorithms. Section 4 validates their good performance. Finally, Sect. 5 concludes the work.

2 Preliminaries

Generally, encryption systems require a source of randomness that can be regenerated in the decryption process. This section explains the main sources of randomness that are employed in the proposed scheme.

2.1 Hyperchaotic Lorenz System

Hyperchaotic Lorenz system [10] provides the randomness needed for encryption. The system is solved using Euler’s method:

$$x_{i+1} = x_i + h(a(y_i - x_i) + w_i), \quad (1a)$$

$$y_{i+1} = y_i + h(cx_i - y_i - x_iz_i), \quad (1b)$$

$$z_{i+1} = z_i + h(x_iz_i - bz_i), \quad (1c)$$

$$w_{i+1} = w_i + h(y_iz_i + rw_i), \quad (1d)$$

where $h = 0.01$, $a = 10$, $b = 8/3$, $c = 28$, and $r = -1$. Figure 1 shows the output results with initial conditions $x_0 = 0.23$, $y_0 = 0$, $z_0 = 0.7$, and $w_0 = 0.11$.

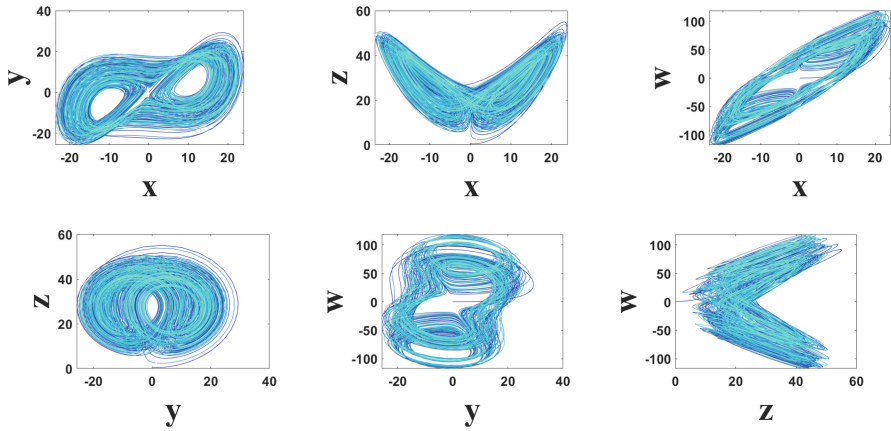


Fig. 1. Output of hyperchaotic Lorenz system.

Table 1. DNA binary codes

DNA base	Binary code
G	00
A	01
T	10
C	11

2.2 DNA Coding

DNA coding [11] is used to change the bit values according to some set of rules. This is done to enhance the security of the algorithm. DNA consists of 4 bases, which are Adenine (A), Thymine (T), Cytosine (C), and Guanine (G). The relation between these bases is that ‘A’ is complementary to ‘T’ and ‘G’ is complementary to ‘C’. Table 1 shows the used binary code for each DNA base.

Based on these relations, we can apply rules to manipulate the data as long as the relation between these bases does not change. Table 2 shows the list of all possible rules that are used in the encryption algorithm, where a random number is used to select the rule and then the two input bits are replaced with the corresponding DNA base. For example, if the chosen rule is 6 and the input is ‘T’, then the output will be ‘C’, which is equal to ‘11’.

Table 3 shows the results of DNA addition and subtraction, which can be done using simple operations on the DNA bases if the binary representation of Table 1 is used. The DNA sequence has a cyclic behavior, where each base is repeated every 4 cycles (i.e., T, C, G, A, T, C, ...). This enables performing ‘DNA cycling’ by dividing the number of cycles by 4 and then referring to Table 4.

Table 2. DNA encoding and decoding rules

	Rules							
	1	2	3	4	5	6	7	8
A	A	C	C	T	T	G	A	G
T	T	G	G	A	A	C	T	C
G	G	T	A	G	C	A	C	T
C	C	A	T	C	G	T	G	A

Table 3. DNA addition and subtraction rules

+	G	A	T	C	-	G	A	T	C
G	G	A	T	C	G	G	C	T	A
A	A	T	C	G	A	A	G	C	T
T	T	C	G	A	T	T	A	G	C
C	C	G	A	T	C	C	T	A	G

Table 4. DNA cycling

Number of cycles	T	C	G	A
4n+0	T	C	G	A
4n+1	C	G	A	T
4n+2	G	A	T	C
4n+3	A	T	C	G

2.3 Arnold Transform

Arnold transform [12] is used to permute the pixels positions of the image. Arnold transform and the inverse operation are defined as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \text{mod} \left(\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, M^2 \right), \tag{2a}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \text{mod} \left(\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}, M^2 \right). \tag{2b}$$

where (x, y) represents the original pixel position and (x', y') represents the new pixel position after applying the transform on an $M \times M$ image.

Arnold transform is a periodic transform [12], which means that at a specific iteration or cycle, the permuted image becomes the same as the original image. The period of the transform depends on M as shown in Table 5 [12].

If the number of cycles of Arnold transform is random, the image will not be permuted if this random number happens to be 0 or P , where P is the Arnold transform period of image dimensions $M \times M$ shown in Table 5. To overcome

Table 5. Arnold transform (2) period with different M

M	32	64	128	256	512
Period	24	48	96	192	384

this periodicity, we propose a modified Arnold transform, where the image will be permuted for any number of cycles chosen.

3 Proposed Algorithm

The proposed algorithm for encryption and decryption is shown in Fig. 2. The proposed modified Arnold transform is explained, after that the encryption and decryption process.

3.1 Modified Arnold Transform

To guarantee image permutation for any number of cycles, the number of cycles (*Cyc*) of the Arnold transform must not equal to 0 or P . Hence, we apply the following equation:

$$G = \text{mod}(Cyc, P - 2) + 1. \quad (3)$$

This will make the effective number of cycles G be in the range of $1 \rightarrow (P - 1)$, which avoids these two cases and eliminates the chances of periodicity.

3.2 Encryption Process

Step 1: The 4 input sub keys (K_1 , K_2 , K_3 , and K_4) are converted from hexadecimal to decimal representation to set the initial state of each variable of the hyperchaotic Lorenz system (1), x_0 , y_0 , z_0 , and w_0 . To make the initial conditions bounded by the chaotic system's basin of attraction, they are computed as:

$$x_0 = \left(\frac{K_1}{A/40} \right) - 20, \quad (4a)$$

$$y_0 = \left(\frac{K_2}{A/40} \right) - 20, \quad (4b)$$

$$z_0 = \left(\frac{K_3}{A/50} \right), \quad (4c)$$

$$w_0 = \left(\frac{K_4}{A/200} \right) - 100, \quad (4d)$$

where $A = 2^{52}$. Then, the 4 chaotic sequences x , y , z and w are generated with length equals $M^2 + 1000$.

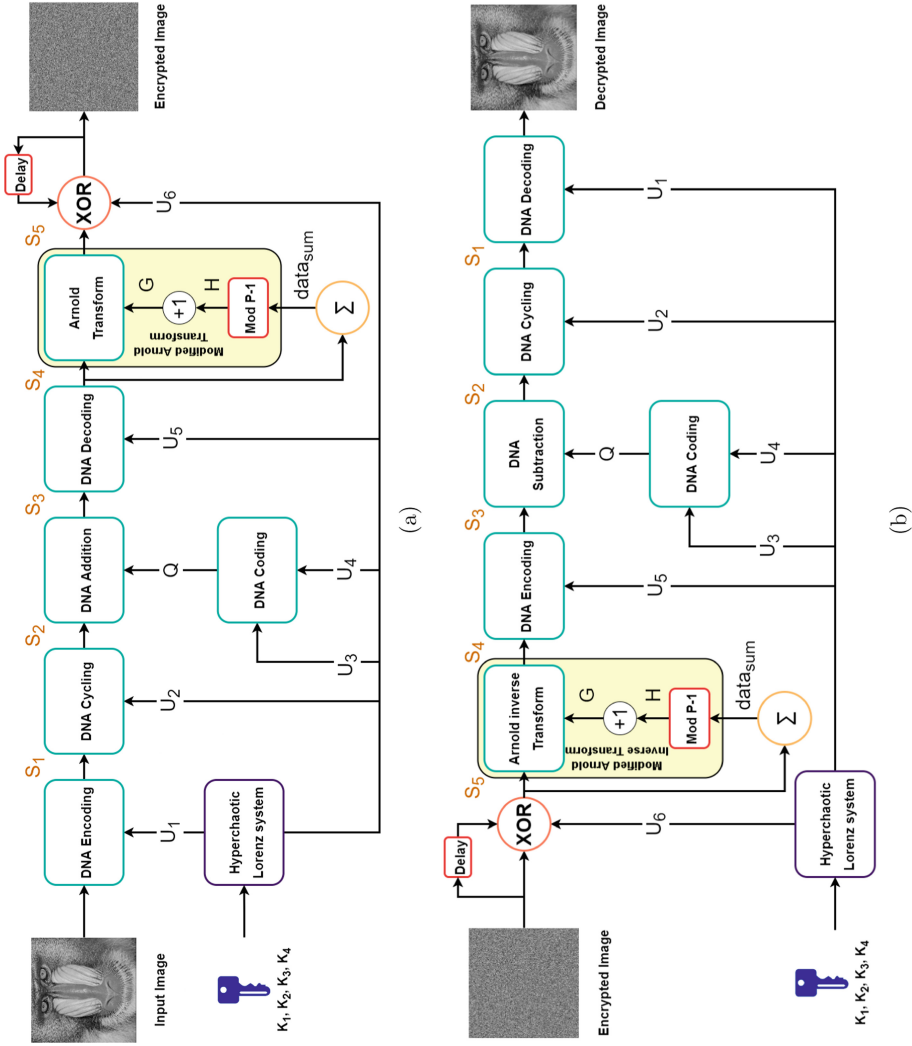


Fig. 2. (a) Encryption and (b) Decryption block diagrams.

Step 2: The first 1000 iterations are removed from the four chaotic sequences to generate X_h , Y_h , Z_h and W_h . Then, the vectors U_1 , U_2 , U_3 , U_4 , U_5 , and U_6 are generated by the following equations:

$$U_1 = \text{mod}(\lceil X_h \times 10^{13} \rceil, 8) + 1, \quad (5a)$$

$$U_2 = \text{mod}(\lceil (U - \lceil U \rceil) \times 10^{13} \rceil, M^2), \quad (5b)$$

$$U_3 = \text{mod}(\lceil W_h \times 10^{13} \rceil, 8) + 1, \quad (5c)$$

$$U_4 = \text{mod}(\lceil (X_h + Y_h) \times 10^{13} \rceil, 256) + 1, \quad (5d)$$

$$U_5 = \text{mod}(\lceil Y_h \times 10^{13} \rceil, 8) + 1, \quad (5e)$$

$$U_6 = \text{mod}(\lceil (W_h + Z_h) \times 10^{13} \rceil, 256), \quad (5f)$$

where $\lceil \cdot \rceil$ is the ceiling operator, and $U = [X_h, Y_h, Z_h, W_h]$.

Step 3: U_1 is used to select the DNA rule to encode the input image according to Table 2.

Step 4: U_2 is used to perform DNA cycling on S_1 . The result of $\text{mod}(U_2, 4)$ chooses how many times the data is shifted according to Table 4.

Step 5: U_3 is used to DNA encode U_4 to generate Q . Then, according to Table 3, the following equations are applied on S_2 :

$$q = Q(1) - Q(M^2), \quad (6a)$$

$$S_3(1) = S_2(1) + Q(1) + q, \quad (6b)$$

$$S_3(i) = S_2(i - 1) + S_2(i) + Q(i). \quad (6c)$$

Step 6: U_5 is used to select the rule for DNA decoding for S_3 according to Table 2.

Step 7: Every byte of S_4 is accumulated to calculate ‘ $data_{sum}$ ’. The proposed modified Arnold transform (3) is applied on S_4 to generate S_5 , where $cyc = data_{sum}$.

Step 8: S_5 is then XORed with the U_6 to generate the encrypted image.

3.3 Decryption Process

Steps 1 and 2: The same as the encryption process.

Step 3: The input encrypted image is XORed with U_6 .

Step 4: The same as step 7 in the encryption process. The only difference is using Arnold inverse transform (2b), instead of Arnold transform. This step is possible even though we are taking the ‘ $data_{sum}$ ’ before the Arnold inverse transform, which is not symmetric with the encryption process. This is because Arnold Transform does not change the pixels values, it only changes their positions.

Step 5: U_5 is used to select the DNA coding rule for S_4 .

Step 6: U_3 is used to DNA encode U_4 to generate Q . Then, according to Table 3, the following equations are applied:

$$q = Q(1) - Q(M^2), \quad (7a)$$

$$S_2(1) = S_3(1) - Q(1) - q, \quad (7b)$$

$$S_2(i) = S_3(i) - Q(i) - S_3(i - 1). \quad (7c)$$

Step 7: U_2 is used to cyclic shift S_2 , which is done by checking the result of $\text{mod}(U_2, 4)$ to choose how many times the data is shifted.

Step 8: U_1 is used to select the DNA decoding rule for S_1 to restore the original image.

4 Performance Evaluation

The proposed system is tested using the gray-scale ‘Lena’ (256×256), ‘Baboon’ (512×512), and ‘Pepper’ (512×512) images.

4.1 Encryption Quality Metrics

Figure 3 shows the histogram of the original and encrypted images, which indicate flat and uniform distribution. Mean Square Error (MSE) [13] and Peak Signal-to-Noise Ratio (PSNR) [14] are used to test encryption quality and are given by:

$$MSE = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M [O_{i,j} - E_{i,j}]^2, \quad (8a)$$

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE}, \quad (8b)$$

where $O_{i,j}$ and $E_{i,j}$ are the original and encrypted image at position (i, j) respectively and n is the number of bits per pixel. MSE and $PSNR \in [0, \infty]$, where high MSE and low PSNR values indicate huge difference between the original and encrypted images. Table 6 shows that the proposed system gives similar MSE and PSNR values compared to other researches.

4.2 Correlation Analysis

The correlation coefficient is given by:

$$\rho = \frac{Cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \quad (9a)$$

$$Cov(x, y) = \frac{1}{M^2} \sum_{i=1}^{M^2} (x_i - \frac{1}{M^2} \sum_{i=j}^{M^2} x_j) (y_i - \frac{1}{M^2} \sum_{i=j}^{M^2} y_j), \quad (9b)$$

$$D(x) = \frac{1}{M^2} \sum_{i=1}^{M^2} (x_i - \frac{1}{M^2} \sum_{i=j}^{M^2} x_j)^2, \quad (9c)$$

$$D(y) = \frac{1}{M^2} \sum_{i=1}^{M^2} (y_i - \frac{1}{M^2} \sum_{i=j}^{M^2} y_j)^2, \quad (9d)$$

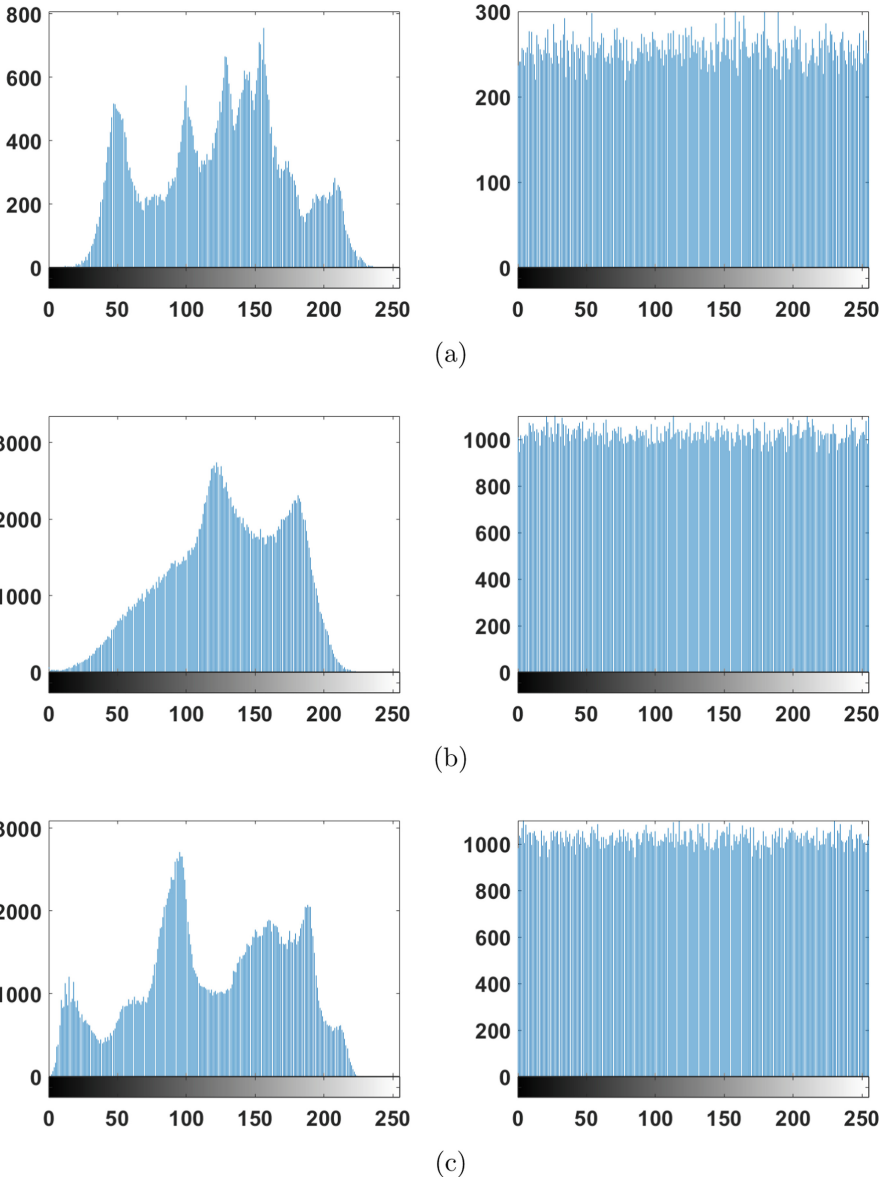
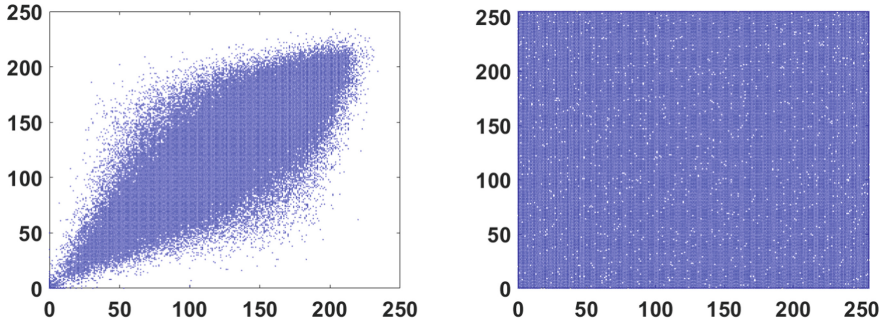
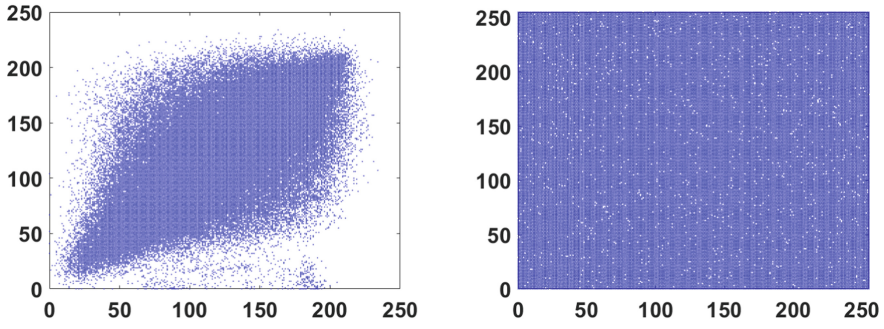


Fig. 3. Histogram of the original image (left), and encrypted image (right) for Lena, Baboon, and Pepper in (a), (b), and (c), respectively.

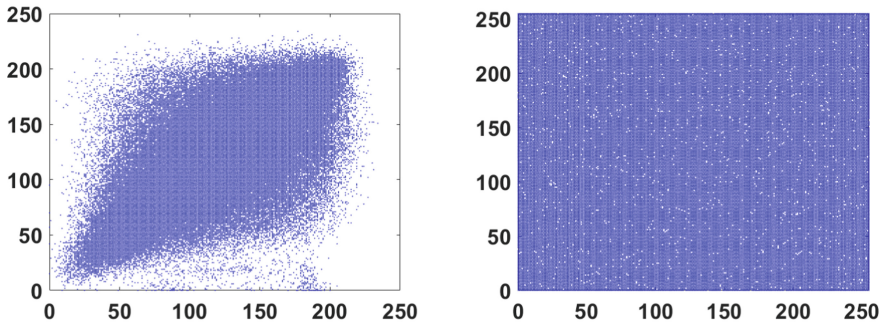
where $Cov(x, y)$ is the covariance between pixels x and y , and D is the standard deviation. The values of the correlation coefficients for the encrypted images must be close to 0, which means that even the neighbouring pixels are uncorrelated. The results in Table 6 show that the correlation coefficients are close to 0 and



(a)



(b)



(c)

Fig. 4. (a) Horizontal, (b) vertical, and (c) diagonal correlation of Baboon image (left) and encrypted Baboon image (right).

comparable to other works. Figure 4 further indicates that the original image pixel values are grouped in a region, which shows that they are correlated. On the contrary, the encrypted image pixel values are spread all over.

4.3 Information Entropy

Information entropy is the average amount of information conveyed by each pixel [14] and is given by:

$$Entropy = - \sum_{i=0}^{255} P(i) \log_2 P(i), \quad (10)$$

where $P(i)$ is the probability of occurrence of i . For an 8 bit image, the ideal value is 8, which means that the information is distributed uniformly over all pixel values. The results in Table 6 shows that the entropy of the encrypted images successfully approach 8.

4.4 Key Space and Sensitivity Analysis

The proposed system has a total number of 4 sub keys, each represented by 52 bits, where $K_1 = (\text{FF123FF0567EF})_{16}$, $K_2 = (\text{F655FF000FFFF})_{16}$, $K_3 = (\text{FFAB0957FFFFFF})_{16}$ and $K_4 = (\text{46FF0108F214F})_{16}$ are the values for the sub keys used. This results in a key space equals $2^{208} \approx 10^{63}$, which is large enough to resist brute force attacks [1, 15]. In addition, the key must have high sensitivity such that any slight change in the decryption key (single bit) prevents recovering the original image. Figure 5 shows the original image of ‘Baboon’ and the wrong decrypted image when changing the least significant bit of the first sub key.

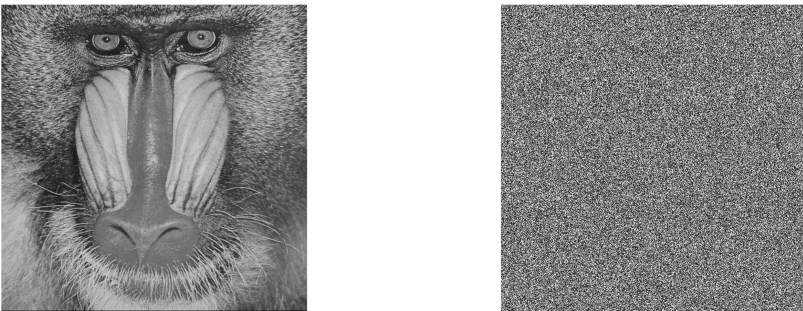


Fig. 5. Original Baboon image (left) and wrong decrypted image (right).

4.5 Robustness Against Differential Attacks

This test is done by changing the least significant bit of a random pixel in the original image and comparing the newly encrypted image to the original encrypted image using Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) [16], which are given by:

$$NPCR = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M DE(i, j) \times 100\%, \quad (11a)$$

$$UACI = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M \frac{|E_1(i, j) - E_2(i, j)|}{255} \times 100\%, \quad (11b)$$

$$DE(i, j) = \begin{cases} 0, & \text{if } E_1(i, j) = E_2(i, j), \\ 1, & \text{if } E_1(i, j) \neq E_2(i, j), \end{cases} \quad (11c)$$

where the difference between corresponding pixels in the encrypted versions of the original image $E_1(i, j)$ and the modified image $E_2(i, j)$ is $DE(i, j)$. The NPCR and UACI values are calculated as the average values of 50 iterations and given in Table 6. They are close to the ideal values 99.61% and 33.46%, respectively, [17] and comparable to recent works,

Table 6. Performance analysis

Ref.	Encrypted image	Encryption quality metrics		Correlation ($\times 10^2$)			Entropy	Robustness against differential attacks	
		MSE	PSNR	H	V	D		NPCR (%)	UACI (%)
This paper	Lena	7828	9.1943	0.42	0.12	0.01	7.9973	99.6042	33.4204
	Baboon	7289	9.5041	-0.18	-0.07	-0.01	7.9993	99.6091	33.4791
	Pepper	8390	8.8931	0.17	-0.08	0.39	7.9991	99.6086	33.4612
[1]	Lena	7793	9.21	-0.18	0.11	-0.09	7.9975	99.6147	33.4723
	Baboon	7285	9.52	0.19	-0.41	-0.99	7.9992	99.6063	33.4565
	Pepper	8436	8.86	-0.63	-0.06	-0.46	7.9993	99.6112	33.4776
[9]	Lena	-	9.2645	-0.03	-0.07	-0.01	7.9977	99.60	33.45
[7]	Lena	-	-	-0.06	-0.39	0.16	7.9978	-	-
	Baboon	-	-	-0.23	-0.00	-0.15	7.9982	99.6056	33.4282

5 Conclusion

This paper presented an encryption algorithm, utilizes hyperchaotic system, DNA manipulation, and a modified Arnold transform. The modified Arnold transform enhances the encryption process by eliminating the cases at which pixel permutation is cancelled. The performance evaluation for the proposed system shows that it is reliable for image encryption compared to recent similar schemes. The design is simple and amenable for real life application hardware realization. For future work, it can be applied on colored images for each channel separately rather than grayscale images only.

Acknowledgment. This paper is based upon work supported by Science, Technology, and Innovation Funding Authority (STIFA) under grant number (#38161).

References

1. Li, M., Wang, M., Fan, H., An, K., Liu, G.: A novel plaintext-related chaotic image encryption scheme with no additional plaintext information. *Chaos, Solitons Fractals* **158**, 111989 (2022)
2. Xian, Y., Wang, X.: Fractal sorting matrix and its application on chaotic image encryption. *Inf. Sci.* **547**, 1154–1169 (2021)
3. Li, T., Shi, J., Li, X., Wu, J., Pan, F.: Image encryption based on pixel-level diffusion with dynamic filtering and DNA-level permutation with 3D Latin cubes. *Entropy* **21**(3), 319 (2019)
4. Alawida, M., Samsudin, A., Teh, J.S., Alkhalwaldeh, R.S.: A new hybrid digital chaotic system with applications in image encryption. *Sig. Process.* **160**, 45–58 (2019)
5. Belazi, A., Abd El-Latif, A.A., Belghith, S.: A novel image encryption scheme based on substitution-permutation network and chaos. *Sig. Process.* **128**, 155–170 (2016)
6. Luo, Y., Yu, J., Lai, W., Liu, L.: A novel chaotic image encryption algorithm based on improved baker map and logistic map. *Multimed. Tools Appl.* **78**(15), 22023–22043 (2019). <https://doi.org/10.1007/s11042-019-7453-3>
7. Ni, Z., Kang, X., Wang, L.: A novel image encryption algorithm based on bit-level improved Arnold transform and hyper chaotic map. In: 2016 IEEE International Conference on Signal and Image Processing (ICSIP), pp. 156–160. IEEE (2016)
8. Ismail, S.M., Said, L.A., Radwan, A.G., Madian, A.H., Abu-Elyazeed, M.F.: Generalized double-humped logistic map-based medical image encryption. *J. Adv. Res.* **10**, 85–98 (2018)
9. Zhou, S., Wang, X., Zhang, Y., Ge, B., Wang, M., Gao, S.: A novel image encryption cryptosystem based on true random numbers and chaotic systems. *Multimed. Syst.* **28**(1), 95–112 (2022). <https://doi.org/10.1007/s00530-021-00803-8>
10. Wang, X., Wang, M.: A hyperchaos generated from Lorenz system. *Phys. A* **387**(14), 3751–3758 (2008)
11. Wu, J., Liao, X., Yang, B.: Image encryption using 2D Hénon-Sine map and DNA approach. *Sig. Process.* **153**, 11–23 (2018)
12. Wu, L., Zhang, J., Deng, W., He, D.: Arnold transformation algorithm and anti-Arnold transformation algorithm. In: 2009 First International Conference on Information Science and Engineering, pp. 1164–1167. IEEE (2009)
13. Mehra, I., Nishchal, N.K.: Optical asymmetric image encryption using gyrator wavelet transform. *Opt. Commun.* **354**, 344–352 (2015)
14. Kaur, M., Kumar, V.: A comprehensive review on image encryption techniques. *Arch. Comput. Methods Eng.* **27**(1), 15–43 (2020). <https://doi.org/10.1007/s11831-018-9298-8>
15. Ghebleh, M., Kanso, A., Noura, H.: An image encryption scheme based on irregularly decimated chaotic maps. *Sig. Process. Image Commun.* **29**(5), 618–627 (2014)
16. Wu, Y., Noonan, J.P., Aghaian, S., et al.: NPCR and UACI randomness tests for image encryption. *Cyber J. Multidisc. J. Sci. Technol. J. Sel. Areas Telecommuni. (JSAT)* **1**(2), 31–38 (2011)
17. Alghafis, A., Munir, N., Khan, M., Hussain, I.: An encryption scheme based on discrete quantum map and continuous chaotic system. *Int. J. Theor. Phys.* **59**(4), 1227–1240 (2020). <https://doi.org/10.1007/s10773-020-04402-7>