IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 22, 2022, accepted June 2, 2022, date of publication June 13, 2022, date of current version June 17, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3182329*

# Gray-Scale Image Encryption Using DNA Operations

**NADA H. SHARKAWY[1], YASMINE M. AFIFY[2], WALAA GAD[2], AND NAGWA BADR[2]**
[1]Institute-El-Shorouk Academy, Cairo 11837, Egypt
[2]Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt

Corresponding author: Nada H. Sharkawy (n.hossam@sha.edu.eg)

**ABSTRACT** Unsecured networks have recently become widely used for the transmission of confidential images. Consequently, cryptography is crucial for ensuring data confidentiality. Developing a key that is resistant to statistical and differential attacks has always been a challenging objective. In this paper, a novel model is proposed to boost image encryption while maintaining key strength. The proposed model adapts MD5 and SHA-256 hash functions to produce a key. It generates four matrices, X, Y, Z, and W, by using a memristor hyperchaotic system. Arnold's transform was applied to the original image once the key was created. The images were then scrambled using five chaotic maps. The image is then DNA-encoded, diffused using three matrices, and finally DNA-decoded. The proposed model was assessed using twelve performance measures on nine popular images. Compared to previous studies, the results of the proposed model indicate a promising improvement in performance. It achieves a better performance by expanding the key space and increasing its sensitivity.

**INDEX TERMS** Arnold's transform, chaotic maps, DNA, MD5, image encryption, memristor HCS, SHA-256.

## I. INTRODUCTION

In recent years, image data has become increasingly significant. As a result, researchers have begun to search for techniques to handle such data. Cryptography is a scientific field in which they can be applied. Different approaches were applied to the images. Many cryptography systems have recently relied on chaotic systems, Arnold's transformations, and DNA encoding.

Chaotic systems are used in many encryption systems owing to their main qualities, including sensitivity to the starting conditions and parameters, strong ergodicity, mixing capability, and highly intricate behavior. However, many chaotic encryption techniques are insecure and susceptible to cryptanalysis, preventing the use of purely chaotic systems in encryption [1].

Arnold's transformation function has several essential characteristics that lead to its use in cryptanalysis: it has a high degree of ergodicity and applies image cutting. However, they cannot be used in cryptography. The image histogram graph does not change because it changes the position of the pixels only, without changing their values.

Because of its advantages, DNA encoding is commonly employed in cryptanalysis. It has a great number of parallelisms, high information density, and it uses very little power. It does, however, have some limitations. DNA encryption rules can be predicted easily if they are combined with low-dimensional chaotic map. Another limitation is that the DNA rules should be related to the original image to improve the security of the encryption system [2].

These methods have recently been used in encryption. Some models have applied DNA encoding, phase-truncated fractional Fourier transform, Hyper-Chaos System, Arnold's transform, and SHA-256 hash function. The primary purpose of the encryption system is to generate the key. Extra encryption layers are required on the original image. Other models relied on chaotic maps only. Their main problem is to boost encryption levels. However, these models lacked a strong key, which leads to the risk of being easily cracked. Two MD5 generated sequences were used in other models to generate the encryption key, which was then used to scramble chaotic maps with the DNA-encoded image. These models require additional encryption layers.

The associate editor coordinating the review of this manuscript and approving it for publication was Wen Chen.

To overcome these shortcomings, we propose a novel model which raises the levels of encryption while increasing the security of the key. The proposed model starts by applying MD5 to the original image and its metadata in order to generate two keys. Both keys are then concatenated and run through SHA-256 to generate a final 256-bit hexadecimal integer, which is then fed into certain specific calculations to generate the secret key. The Arnold's transform algorithm settings, the Hyper-Chaos System beginning values, and the encryption and decryption procedures constitute the key. The Hyper-Chaos System is then utilized to construct four matrices, which are subsequently diffused with the image. The third step is to apply the Arnold's transform algorithm to the original image, and the resulting image is then successively entered into the five chaotic maps. Finally, the result is encoded to DNA, diffused with the three matrices formed by the Hyper-Chaos System, and DNA-encoded, with the resulting image being DNA-decoded and the cipher image is obtained.

To ensure that the model is efficient, it is evaluated on nine grayscale images using twelve evaluation measures. The key space and key sensitivity measure the key strength, while Histogram analysis, Chi-square test, Correlation Coefficient Adjacent (CCA) analysis, Information Entropy, Irregular Deviation, NPCR, UACI, and MSE measure the efficiency of the model. To measure the computational complexity of the model, computation analysis and time analysis are applied on the model.

Compared to previous studies, the results of the proposed model show a promising improvement in performance. Regarding the key analysis, the model assumes a wide key space and is highly affected by minor changes. The cipher images' histograms are eventually distributed which is proven in chi-square results. The difference between information entropy of the cipher image from the original image indicates great randomness, which is also visible in the correlation coefficient analysis of adjacent pixels of the cipher image and the irregular deviation analysis. The execution time of encryption and decryption processes vary since the key generation steps depend on the image size. It also appears in the computational complexity of the model. NPCR and UACI results show that even minor changes in the original image are causing great changes in the cipher image. MSE results reveal that different features in encryption layers expand the space between the original and encrypted images.

The contributions of this work consist of:

- Proposing a novel gray-scale image encryption model using DNA operations.
- Proposed model raises the encryption levels while enhancing the key security.
- Proposed model is comprehensively assessed using twelve performance measures on nine images.

The rest of the paper is structured as follows. Section 2 describes the related models in the literature and differences between them. Section 3 describes DNA sequences, how they work, the Memristor Hyper-Chaos System, how it performs, different Chaotic Maps, Arnold's transform, and Hashing Functions in depth. Section 4 goes into the specifics of the proposed model. Afterwards, we review the performance measures, and discuss the experimental results of the proposed model in section 5. Finally, section 6 concludes the work and presents the future directions.

## II. RELATED WORK

This section explores some of the most recent models that focus on image encryption. Details of these models and how they work, as well as their essential features, such as their strengths and weaknesses are presented.

Wang *et al.* [3] proposed a new model based on Logistic-Dynamics Coupled Map Lattices (LDCML) and DNA encoding. LDCML is used to create a chaotic sequence, which is subsequently applied to scramble the original image. The image is then DNA encoded and scrambled according to C shape. After that, the image is diffused using the chaotic sequence and subjected to operations like addition, subtraction, and XOR. Finally, the image is DNA-decoded. The results indicate that the key space is wide and that the key is sensitive to small changes. Unfortunately, the system is slightly vulnerable to statistical and differential attacks.

El-Khamy *et al.* [4] combined DNA encoding with Choquet's Fuzzy Integral sequences. Using a Logistic map, the original image is pixel confused. Afterwards, the image is DNA-encoded. The four DNA bases are used to create four coded images. In the meantime, a Choquet's fuzzy integral sequence is produced and DNA-encoded, yielding four sequences. Following that, using DNA XOR technique, the four Fuzzy-DNA sequences are diffused with the DNA-encoded images. Then, the wavelet fusion algorithm is applied to the generated images to create an encrypted image. The findings show that the system is very resistant to statistical and differential attacks, and that the key is extremely sensitive to tiny modifications. However, the key space is quite tiny and predictable.

Wang and Li [5] presented another model based on Multi-Objective Particle Swarm Optimization (MOPSO), DNA-encoding, and Logistic map. At first, PSO, the SHA-384 hash function, and a shuffle mark bit are used to create the key. Then, using a Logistic map and DNA-encoding, it generates random DNA-masked images. Finally, PSO is used to combine these images with the original image, and the best cipher result is returned. The results demonstrate that there is a considerable key space and that the key is sensitive to small modifications. Unfortunately, the system has a low level of statistical and differential attack protection.

In 2021, Zhang *et al.* [2] proposed a model based on phase-truncated fractional Fourier transform and DNA-level operations. It creates the key first, using the SHA-256 algorithm and HCS. The original image is then scrambled with Arnold's transform and encoded with ptFrFT into a noise-like intermediate. Finally, Hyper-Chaos System creates four matrices to diffuse the image. The model's testing results

revealed that it has a vast key space and is highly secure and robust.

Elamir *et al.* [6] proposed a model in the same year that entails masking patient information in a medical image using the least significant-bit approach, then diffusing the image with six chaotic maps: Chebyshev, Gauss, Logistic, Tent, Henon, and Piece-Wise maps. Finally, using DNA encoding principles, the image is DNA encoded. The chaotic maps' order is a part of the secret key. The results show that the system is resistant to statistical and differential attacks, but it has a limited key space and is less sensitive to slight changes.

Moreover in 2021, Xu *et al.* [7] presented a model for 2D and 3D images based on a discrete chaotic system. The hash value is first computed using the SHA-256 hash function and then used to alter the discrete system's initial conditions. The Arnold's matrix and DNA diffusion algorithm are then used to jumble the original image using the sequences generated by the operating chaotic system. The key space is large, and the key is particularly sensitive to minor modifications, according to the security analysis. In comparison to other models, the system is not powerful enough to prevent statistical and differential attacks.

Aouissaoui *et al.* [1] devised a model that relies on DNA, tent and logistic maps, and hash functions (MD5 and SHA-256). The key is first produced using the hash functions MD5 and SHA-256. After that, the two MSB bit-plans of the original image are rotated and permuted. Finally, the image is DNA encoded and diffused using a tent and logistic map. The results indicate that the system has a vast key space and is sensitive to small alterations. The system is not applied on many popular benchmark datasets which makes the comparison difficult.

Tian *et al.* [8] proposed a new model based on Coupled Map Lattices (CML) and Piece-Wise Linear Chaotic Map (PWLCM). To begin, the control parameters of the CML system and PWLCM map are calculated using the external keys and the original image hash value. The chaotic map is then constructed using a CML chaotic system based on the PWLCM map. The image is then pixel-based sorted and subjected to an XOR operation. It is then DNA-encoded and diffused. Finally, the picture is dispersed and DNA-decoded. The results are high, but not as good as others because the key space is tiny, less sensitive to slight changes, and provides less protection against statistical and differential attacks.

Some approaches, as demonstrated above, built a key with a very vast key space and a high sensitivity to slight changes, but they lacked the robustness against statistical and differential attacks. The models that imply high robustness, on the other hand, have a weak key that is easily predicted. Reaching both goals is a challenge.

## III. FUNDAMENTAL KNOWLEDGE (PRELIMINARIES)

This section describes DNA sequence and how to apply DNA-encoding and DNA-decoding in addition to DNA operations, memristor hyper-chaos system, Arnold's transform, and hashing functions.

### A. DNA SEQUENCE

DNA, or deoxyribonucleic acid, determines the traits and biological processes of all living beings. DNA is made up of two strands that create a helix structure. Each strand is made up of four nucleotides: adenine (A), thymine (T), cytosine (C), and guanine (G), all of which are connected by phosphodiester bonds. Hydrogen bonds are used to join the two strands. Adenine and Thymine are purines with two hydrogen bonds between them, whereas Cytosine and Guanine are pyrimidines with three hydrogen bonds between them, making Adenine and Thymine make a connection whereas Cytosine and Guanine make a connection [9]. The features of DNA have led to its use in cryptography in recent studies. It has rules in its connections that, if followed, will make cryptographic systems more secure.

#### 1) DNA ENCODING AND DECODING

Any type of data can be transformed to the shape of DNA using the properties of DNA. Each nucleotide can be represented in binary, for example, A = 00, C = 01, G = 10, and T = 11 due to the four different types of nucleotides. The number of potential permutations is eight, according to Watson and Crick's DNA structural model, which makes A and T complementary while C and G complementary. Table 1 [2], [10] shows how these permutations are stated in rules. The image is converted to binary form in DNA encoding, and then each two bits are converted to their respective nucleotide according to the rule chosen. Meanwhile, inverse actions are used to obtain DNA decoding.

For example, if the grayscale pixel value of an image is 125, then the correspondence binary value is "01111101". By using rule 5, the binary value can be encoded to the sequence "TGGT". Whereas decoding this sequence, we follow the decoding rule from Table 1. The encoding and decoding rules are obtained from the result of key generation operations.

**TABLE 1. Rules for DNA-encoding.**

| Rule | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 | Rule 7 | Rule 8 |
|---|---|---|---|---|---|---|---|---|
| 00 | A | A | T | T | C | C | G | G |
| 01 | G | C | G | C | T | A | T | A |
| 10 | C | G | C | G | A | T | A | T |
| 11 | T | T | A | A | G | G | C | C |

#### 2) DNA ADDITION, SUBTRACTION, AND XOR OPERATIONS

The encoding rule used on the data in addition, subtraction, and XOR operations is considered. These operations are used at the DNA level to perform diffusion between the image and other data in order to improve the system security [11]. For example, if the encoding rule is 1, besides having "AATT" and "CTAT," the result of addition is "CTTC", whereas the result of XOR is "CTTA". The matrices of addition,

**TABLE 2.** DNA addition operation.

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

**TABLE 3.** DNA subtraction operation.

| - | A | G | C | T |
|---|---|---|---|---|
| A | A | T | C | G |
| G | G | A | T | C |
| C | C | G | A | T |
| T | T | C | G | A |

**TABLE 4.** DNA XOR operation.

| XOR | A | G | C | T |
|---|---|---|---|---|
| A | A | G | C | T |
| G | G | A | T | C |
| C | C | T | A | G |
| T | T | C | G | A |

subtraction, and XOR operations employing rule 1 are shown in Table 2–4.

### B. MEMRISTOR HYPER-CHAOS SYSTEM (HCS)

The Hyper-Chaos system is a multi-dimensional chaotic map that generates several dependent chaotic sequences for cryptographic operations [2]. The chaotic system's initial parameters are frequently created from the original image, which makes them unique to each image and increases its randomness, hence improving the cryptosystem's security [12]. The memristor is a non-linear parameter that increases the randomness of the system [13].

The classical Liu chaotic system is the base of four-dimensional memristor hyper-chaotic system. Equation (1) is applied to generate the chaotic sequences [2]:

$$\begin{cases} x(1, i+1) = 10\,(y(1, i) - x(1, i)) \\ y(1, i+1) = (40x(1, i)) - (x(1, i)\, z(1, i)) \\ \qquad\qquad + (k\mathrm{W}(w(1, i))\, x(1, i)) \\ z(1, i+1) = (mz(1, i)) - (3y(1, i)) \\ \qquad\qquad + (30x(1, i)\, y(1, i)) \\ w(1, i+1) = x(1, i) \end{cases} \tag{1}$$

where $x$, $y$, $z$, and $w$ are system sequences, $\boldsymbol{i}$ is iterated from 1 to width of image multiplied by its height, $k$ is memristor parameter, and $m$ is the chaotic system parameter. W($w$) is the memductance of the memristor calculated by Equation (2):

$$\mathrm{W}(w(1, i)) = -a + (b * |w(1, i)|) \tag{2}$$

where $a$ and $b$ are memristor model parameters and $|\boldsymbol{w}(\boldsymbol{1,i})|$ is the absolute value of $\boldsymbol{w}(\boldsymbol{1,i})$. The system is applied to mod 256. The values of the parameters are $k = 1$, $m = -1$, $a = 10$, and $b = 1$ which make the system in hyper-chaotic state [2].

### C. ARNOLD'S TRANSFORM (ART)

Arnold's transform (Arnold's cat map) is a method of image splicing and cutting. Arnold's and Avez established it in 1968 based on ergodic theory [14], [15]. The original image is scrambled using the Arnold's transformation. It changes the pixels position. It is applied by Equation (3) [2]:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \left( \begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \right) \mathrm{mod}(N) \tag{3}$$

where $[x_n, y_n]^T$ and $[x_{n+1}, y_{n+1}]^T$ are the positions of the pixel before and after transformation, respectively, $\left( \begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \right) \mathrm{mod}\,(N)$ is the remainder of dividing $\begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}$ by $N$.

One of the key properties of Arnold's transform is that it resumes the image after a set number of iterations, which assists with image retrieval [15].

### D. HASHING FUNCTIONS

Hashing functions are used to map any-size data to fixed-size output [1]. The results are expressed as $n$-bit hexadecimal numbers. Because of its irreversibility, they can defeat plaintext and ciphertext attacks. We applied the Message-Digest algorithm (MD5) and the Secure Hash Algorithm-256 (SHA-256). The SHA-256 function generates hexadecimal numbers with a length of 256 bits [16], whereas the MD5 function generates hexadecimal numbers with a length of 128 bits. Hash functions have the advantage of being extremely sensitive to small changes in input values. For example, the outputs of MD5 function with Lena picture of size 256256 pixels as inputs and one-pixel difference between both inputs are the following 128-bit hexadecimal numbers:

**Image (1, 1) = 137,**
**D = "c09857783961d8c9c0be450a0e606342"**
**Image (1, 1) = 138,**
**D = "4a978f34763919f5fd2ea2cac231219c"**

The two numbers are absolutely different from one another, as shown above, making hashing functions efficient for generating security keys.

## IV. PROPOSED MODEL

In this section, the proposed model is presented along with details on its workflow and components. The encryption and decryption models are described in detail in the following sub-sections.

### A. IMAGE ENCRYPTION MODEL

Fig.1 shows how the proposed model works on M*N grayscale images. The proposed encryption model consists of

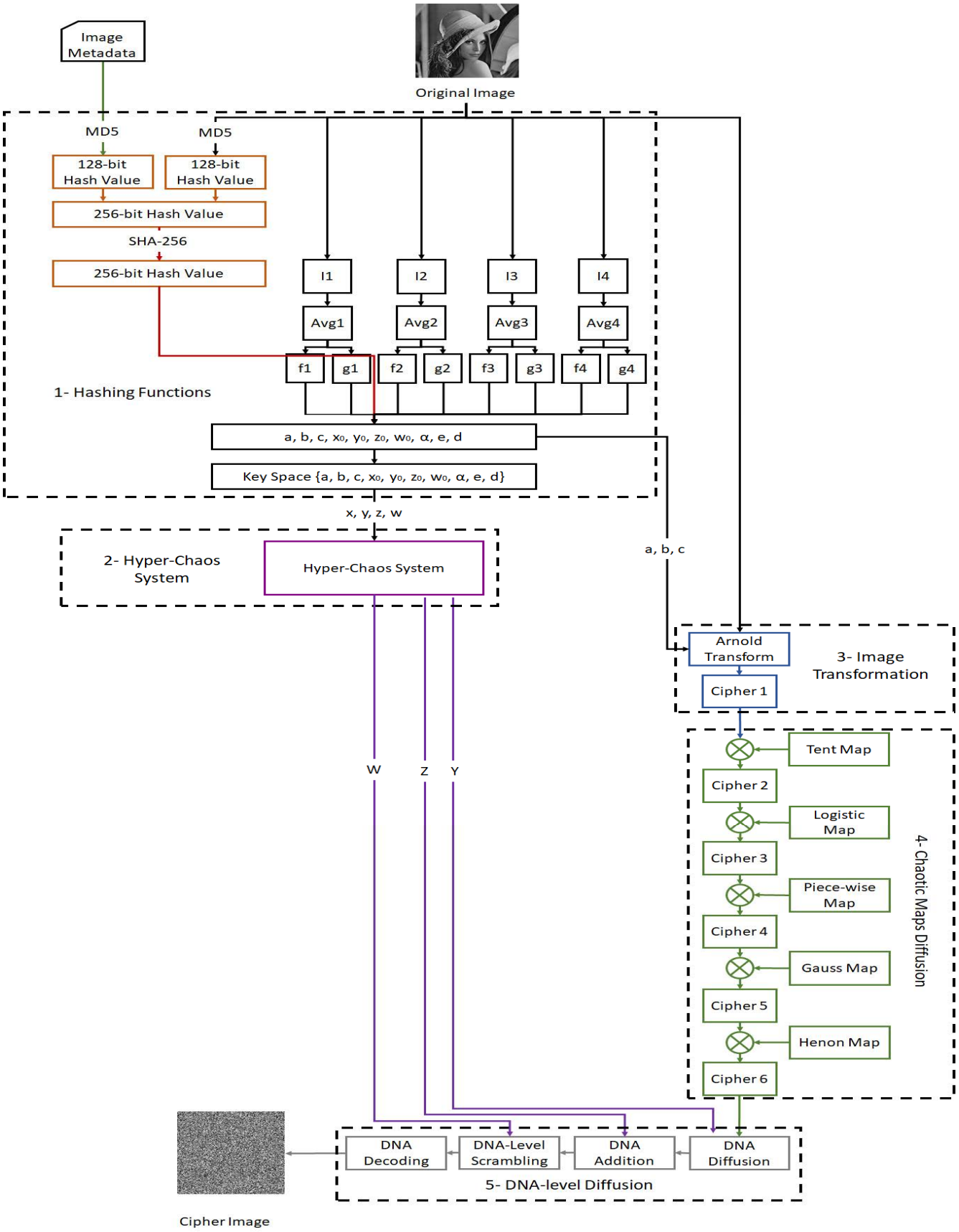


**FIGURE 1.** Proposed encryption model.

five phases: key generation using hashing functions, chaotic sequence generation, encryption using ART, encryption using chaotic maps, and encryption in DNA level. The original image and its metadata (image path) are first placed into Hashing Functions to get the secret key, which is then input into the Hyper-Chaos System to generate four matrices that are subsequently DNA encoded. The generated key and the original picture both go through the image transformation phase at the same time, after which the result is diffused with Chaotic Maps to create a new matrix, which is then encoded to DNA and employed in DNA operations with the four DNA-encoded matrices. The cypher image is the result after DNA-decoding. The model is described in detail in the following sub-sections.

#### 1) KEY GENERATION USING HASHING FUNCTIONS

The secret key is generated using the following steps:

*Step 1 (Applying MD5 Hash Function):* At first, the metadata is extracted from the original image (its path), then MD5 algorithm is run on the metadata and on the original image, to get two 128-bit hexadecimal numbers.

*Step 2 (Applying SHA-256 Hash Function):* The two sequences from the previous step are merged to create a single 256-bit hexadecimal number, which is then entered into the SHA-256 algorithm to produce the final 256-bit hexadecimal number $H$.

$$\boldsymbol{H} = [h_1, \boldsymbol{h_2}, \ldots, \boldsymbol{h_{64}}]$$

*Step 3 (Obtain $f$ and $g$):* The original image is partitioned into four $M/2^*N/2$ equivalent matrices. The average of each matrix is then calculated. Following that, for each average, two values are received: $f$ and $g$. The integer part is $f$, and the fractional part is $g$.

*Step 4 (Get Secret Key):* $a, b, c, x_0, y_0, z_0, w_0, e$, and $d$ are calculated in this step. The Arnold's transformation function's parameters are *a, b*, and *c*. The starting values of the hyper chaos system are $x_0, y_0, z_0$, and $w_0$. The encryption and decryption rules are $e$ and $d$, respectively. Equations (4) is used to calculate the starting values and control parameters:

$$\begin{cases} a = \text{round}\left(\text{mod}\left(\left(\text{hex2dec}\left(H\left(h1:h5\right)\right)/M\right), 10\right)+1\right) \\ b = \text{round}\left(\text{mod}\left(\left(\text{hex2dec}\left(H\left(h6:h10\right)\right)/N\right), 10\right)+1\right) \\ c = \text{round}\left(\text{mod}\left(\left(\text{hex2dec}\left(H\left(h11:h15\right)\right)/M\right), 10\right)+124\right) \\ x = \text{mod}\left(\left(\text{hex2dec}\left(H\left(h21:h25\right)\right)/(M+f1)\right), 0.234\right) \\ y = \text{mod}\left(\left(\text{hex2dec}\left(H\left(h26:h30\right)\right)/(N+f2)\right), 0.741\right) \\ z = \text{mod}\left(\left(\text{hex2dec}\left(H\left(h31:h35\right)\right)/(M+f3)\right), 0.508\right) \\ w = \text{mod}\left(\left(\text{hex2dec}\left(H\left(h36:h40\right)\right)/(N+f4)\right), 0.592\right) \\ e = \text{round}\left(\text{mod}\left(\left(\text{hex2dec}\left(H\left(h41:h45\right)\right)/M\right), 8\right)+1\right) \\ d = \text{round}\left(\text{mod}\left(\left(\text{hex2dec}\left(H\left(h46:h50\right)\right)/N\right), 8\right)+1\right) \end{cases} \tag{4}$$

where round (. . .) rounds value to the nearest integer, hex2dec (. . .) converts hexadecimal integer to the equivalent decimal number.

And then, $x_0, y_0, z_0$, and $w_0$ are calculated from *x, y, z*, and *w* using Equation (5):

$$\begin{cases} x_0 = x + g_1 \\ y_0 = y + g_2 \\ z_0 = z + g_3 \\ w_0 = w + g_4 \end{cases} \tag{5}$$

The secret key is {*a, b, c,* $x_0, y_0, z_0, w_0$, *e, d*}.

#### 2) CHAOTIC SEQUENCE GENERATION

The four sequences $X_2, Y_2, Z_2$, and $W_2$ are generated with one step, which is the following:

*Step 5 (Create Matrices):* By applying memristor HCS to the initial values $x_0$, $y_0$, $z_0$, and $w_0$, four sequences of size $M*N$ are received: *X, Y, Z*, and *W*. Then Equation (6) is used to determine $X_2, Y_2, Z_2$, and $W_2$:

$$\begin{cases} X_2 = \text{uint8}\left(\text{mod}\left(X, 1\right)\right) \\ Y_2 = \text{uint8}\left(\text{round}\left(\text{mod}\left(\left(Y \times 100000\right), 256\right)\right)\right) \\ Z_2 = \text{uint8}\left(\text{round}\left(\text{mod}\left(\left(Z \times 100000\right), 256\right)\right)\right) \\ W_2 = \text{uint8}\left(W * 100000\right) \end{cases} \tag{6}$$

where uint8 (. . .) converts value to unsigned 8-bit integer.

#### 3) ENCRYPTION USING ART

*Step 6 (Encryption using ART):* The original image is scrambled with Arnold's transform using the initial values, *a, b*, and iterated *c* times, to yield $I_2$.

#### 4) ENCRYPTION USING CHAOTIC MAPS

*Step 7 (Encryption Using Chaotic Maps):* In this stage, $I_2$ is changed in five iterations, each with a different chaotic map. The XOR operation is applied to $I_2$ and a map in sequence in each cycle. Tent, Logistic, Piecewise, Gauss, and Henon maps are the five maps employed in this step. Each function is firstly iterated 120 times (*N1*) before creating the map. As a part of the secret key, the sequence of maps is provided. Table 5 contains the equations of the five maps.

#### 5) ENCRYPTION IN DNA LEVEL

Regarding these steps, $I_2$ is encoded into DNA and diffused with $Y_2, Z_2$, and $W_2$ matrices:

*Step 8 (DNA Encoding):* First, the three sequences are reshaped into $M*N$ matrices: $Y_2, Z_2$, and $W_2$. The four matrices are then transformed to binary matrices: $I_2$, $Y_2$, $Z_2$, and $W_2$. $I_2$, $Y_2$, $Z_2$, and $W_2$ are then encoded to DNA matrices using rule $e$ which is calculated before.

*Step 9 (Applying DNA Operations):* $I_2$ and $Y_2$ are initially subjected to an XOR procedure using Equation (7):

$$\begin{cases} I_{3_1} = I_{3_1} \oplus I_{3_{4N}} \oplus Y_{2_1} & i = 1 \\ I_{3_i} = I_{3_{i-1}} \oplus I_{3_i} \oplus Y_{2_i} & i = 3, 5, 7 \ldots \\ I_{3_i} = I_{3_i} & i = 2, 4, 6 \ldots \end{cases} \tag{7}$$

where $\oplus$ is XOR operator.

$I_3$ will be the end result.

**TABLE 5. Chaotic maps equations [6].**

| Map | EQUATION | Initial Condition |
|---|---|---|
| Gauss map | $X_n = e^{-a(X_{n-1})^2} + b$ | $X_0 = 0.6$, $a = 4.7$, $b = 0.6$ |
| Henon map | $X_n = 1 - a(X_{n-1})^2 + Y_{n-1}$, $Y_n = bX_n$ | $X_0 = 0.6$, $Y_0 = 0.6$, $a = 1.4$, $b = 0.3$ |
| Logistic map | $X_n = hX_{n-1}(1 - X_{n-1})$ | $X_0 = 0.6$, $h = 2$ |
| Tent map | $\begin{cases} Y_n = uY_{n-1} & Y_{n-1} < 0.5 \\ Y_n = u(1 - Y_{n-1}) & Y_{n-1} \geq 0.5 \end{cases}$ | $Y_0 = 0.6$, $u = 1.9$ |
| Piece-wise map | $X_n = \frac{X_{n-1} - \frac{X_{n-1}}{q}}{q}$ | $X_0 = 1$, $q = 0.4$ |

Then, to achieve $I_4$, $Z_2$ is added to $I_3$. Finally, using Equation (8), $I_4$ is sorted based on $W_2$:

$$I_5 = \text{sort}(I_4, W_2) \tag{8}$$

where sort($I_4$, $W_2$) sorts first elements using second elements. The resulting matrix is $I_5$.

*Step 10 (Decode the Cipher Image):* To obtain the encrypted image, $C$, $I_5$ is DNA-decoded using rule $d$ in the final phase.

### B. IMAGE DECRYPTION MODEL

The decryption model is the reverse of the encryption model, as shown in Fig. 2. The cipher image is first DNA-encoded using the DNA rules. The reverse DNA operations are then applied. The original image is then obtained by decoding the matrix and applying chaotic maps, followed by an inverse Arnold's transform. The decryption model is described in detail in the following sub-sections:

#### 1) CHAOTIC SEQUENCE GENERATION

The four sequences $X_2$, $Y_2$, $Z_2$, and $W_2$ are generated with the following steps:

*Step 1 (Create Matrices):* By reapplying memristor HCS to the initial values $x_0$, $y_0$, $z_0$, and $w_0$, which are parts of the key, four sequences of size $M*N$ are received: $X$, $Y$, $Z$, and $W$. Then Equation (6) is reused to determine $X_2$, $Y_2$, $Z_2$, and $W_2$.

#### 2) DECRYPTION IN DNA LEVEL

The encrypted image $C$ is encoded and diffused with $Y_2$, $Z_2$, and $W_2$ using the following steps:

*Step 2 (Encode the Cipher Image):* The encrypted image, $C$, is DNA-encoded using rule $d$ to obtain $I_5$.

*Step 3 (Applying DNA Operations):* According to the order of sorting matrix $W_2$ using Equation (8), $I_5$ is resorted to get $I_4$. Afterwards, $Z_2$ is subtracted from $I_4$ to get $I_3$. $I_2$ is obtained from applying $I_3$ and $Y_2$ to an XOR procedure using Equation (7).

#### 3) DECRYPTION USING ART AND CHAOTIC MAPS

The following steps are applied to get the original image using Inverse Arnold's transform and chaotic maps:

*Step 4 (Decryption Using Chaotic Maps):* In this stage, the sequence of maps in the secret key is reversed in applying XOR operations with $I_2$ to get $I_1$. Each function is firstly iterated 120 times ($N1$) before creating the map.

*Step 5 (Decryption Using ART):* The original image is restored by scrambling $I_1$ with Arnold's transform using the initial values, $a$, $b$, and $c$.

## V. EXPERIMENTAL EVALUATION

The model is implemented in MATLAB R2021b platform on 64-bit machine with Intel®Core™i7-4500U CPU @ 1.80 GHz processor and 8 GB RAM on Windows 10 Operating System. In the following sub-sections, the dataset, the performance measures, the experimental evaluation outcomes, and the interpretation of the results are presented in detail.

### A. DATASET

The dataset used to test the proposed model contains nine popular grayscale images: Lena, Cameraman, Baboon, Barbara, Peppers, Couple, QR code, Black, and white. These images are of size 256∗256 pixels.

### B. EXPERIMENTAL RESULTS

To comprehensively evaluate the proposed model, twelve popular performance measures are employed that reflect the following aspects: key analysis, time complexity analysis, statistical attacks analysis, and differential attacks analysis. In the following descriptions of each measure, the value NA implies that this image is not used by this model (Table 6–13). Details about the measures' calculations and the results of the proposed model are compared to those of the benchmark approaches [1]–[8], [12] and presented in the following sub-sections.

#### 1) KEY ANALYSIS

In encryption, the key plays a crucial role. It should be password-protected, and it must be of high level of security. The proposed model is so sensitive to its key, to the original image, and to the encrypted image that any small change in any of them causes a significant error in the resulting image. For instance, if any minor noise is applied to the encrypted image, the resulting image is random looking. The main reason is applying chaotic systems, i.e. piece-wise chaotic map, that they are very sensitive to minor changes [17], [18]. Two tests are used for this: key space and key sensitivity, both of which are described in detail.

##### a: ANALYSIS OF THE KEY SPACES

To resist brute-force attacks, the key space should be larger than $2^{100}$ [1], [2]. The number of variables created, and their

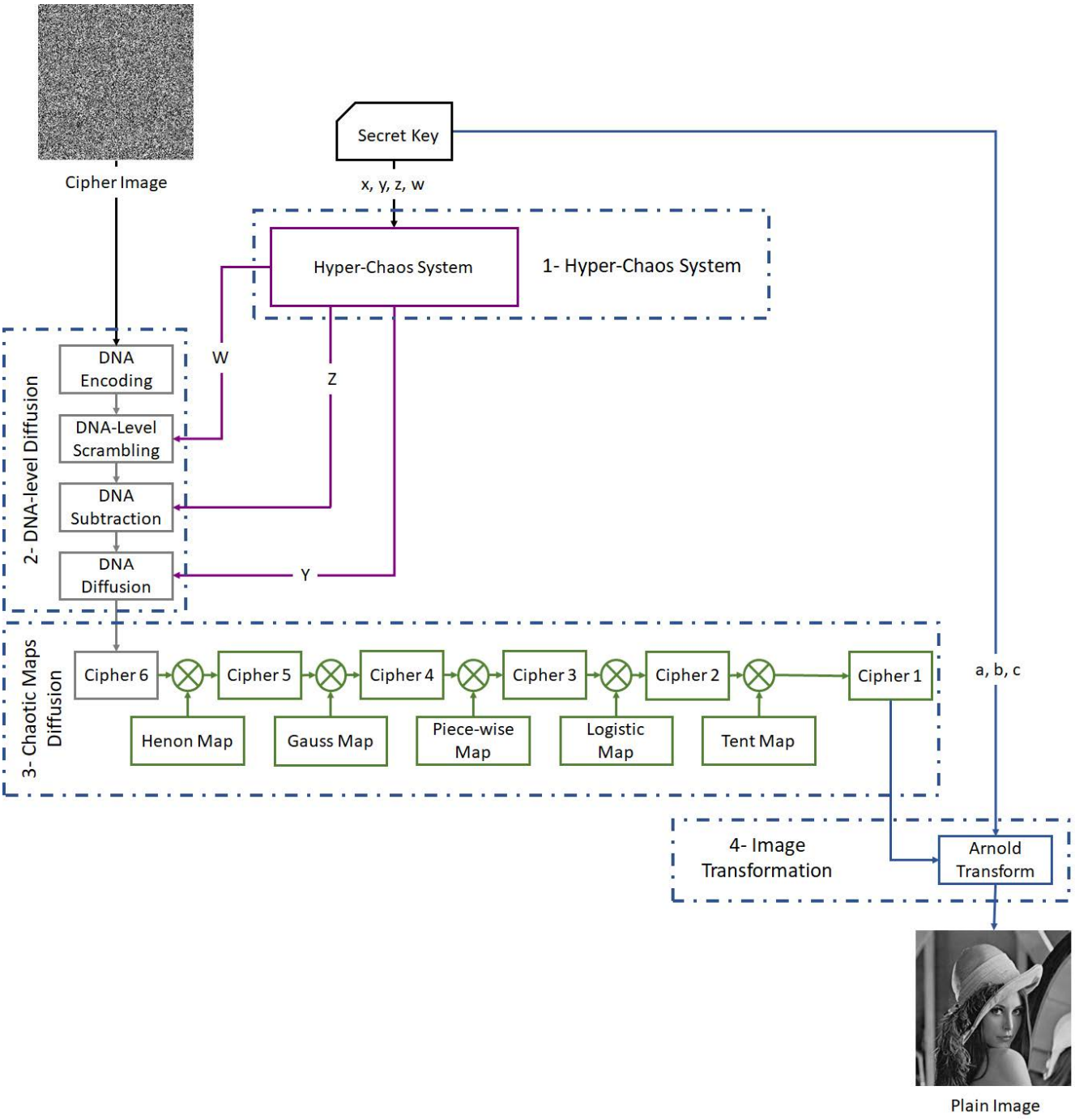


**FIGURE 2.** **Proposed decryption model.**

probabilities are used to calculate the key space [19]. In the model, the key consists of two 128-bit hexadecimal numbers and one 256-bit hexadecimal number; control parameters of hyper-chaos system and $g_1$, $g_2$, $g_3$, $g_4$ generated by original image; the selected DNA-encoding rule (8 kinds); the selected DNA-decoding rule (8 kinds). So, the key space will be:

$K = 2^{256} * 2^{128*2} * 10^{14*4} * 8 * 8 = 2^{704}$, which is extremely higher than $2^{100}$.

*b: ANALYSIS OF KEY SENSITIVITY*

The sensitivity of the key has an impact on the model's attack. It is determined by altering a small portion of the key and analyzing the outcome. To test key sensitivity, some variables: $x_0$, $y_0$, $z_0$, $e$, and $d$, are changed from the key by only the $10^{\text{th}}$ digit before its decimal point by adding $10^{-10}$ instead of correct variables. The results are shown in Fig. 3–5 on Lena image of variables $x_0$, $y_0$, and $z_0$. When changing the values of $e$ and $d$ by adding $10^{-10}$ to them, the system

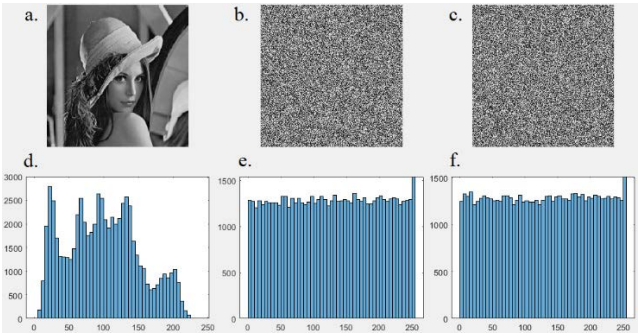


**FIGURE 3.** Histogram of Lena image: a. original image, b. encrypted image, c. decrypted image after $x_0$ change, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image after $x_0$ change.

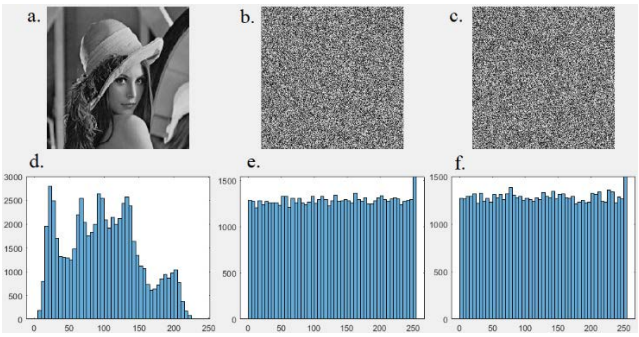


**FIGURE 4.** Histogram of Lena image: a. original image, b. encrypted image, c. decrypted image after $y_0$ change, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image after $y_0$ change.

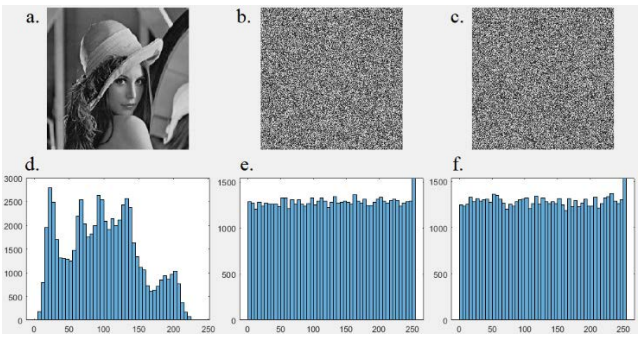


**FIGURE 5.** Histogram of Lena image: a. original image, b. encrypted image, c. decrypted image after $z_0$ change, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image after $z_0$ change.

couldn't run since they determine the index of rules. These variables only run when using integer numbers which lead to undefined image. The results prove that the proposed model is very sensitive to minor changes in the key.

### 2) COMPUTATIONAL AND TIME COMPLEXITY ANALYSIS

The time used to apply encryption should be minimized. Computational and time complexity are analyzed using computational and time analysis results in this section.

#### *a: COMPUTATIONAL ANALYSIS*

It depends on how many times any statement executes. The system consists of five phases: key generation using hashing functions, chaotic sequence generation, encryption using ART, encryption using chaotic maps, and encryption in DNA level. Applied on an image of size $M*N$, the key generation computational complexity is $O(M*N)$. The computational complexity of second phase, generation of chaotic sequences, is $O(M*N)$. The computational complexity of encryption using ART is $O(c*M*N)$, where $c$ is secret key parameter. The computational complexity of encryption using chaotic maps is $O(5*N1 + 5*M*N)$. The computational complexity of the final phase, encryption in DNA level, is $O(4*M*N)$. Therefore, the computational complexity of the proposed model is $O(5*N1 + (11 + c)*M*N) \approx O(M*N)$ which is linear and increases by increasing size of image.

#### *b: TIME ANALYSIS*

It depends on the implementation and the computing environment. Table 6 contains the results of the encryption and decryption times in seconds. The results show long execution time taken by the DNA encoding and DNA decoding steps.

### 3) STATISTICAL ATTACKS ANALYSIS

Statistical attacks on the model should be avoided as much as possible. The degree of security is determined using the histogram, chi-square test, information entropy, irregular deviation, and correlation coefficient analysis. Experimental results of these five measures are presented in detail in this section.

#### *a: HISTOGRAM*

It represents the distribution of the values of the pixel intensity. The histogram should be distributed eventually in the encrypted image. The results of the nine images are shown in Fig. 6–14. Each figure is divided into six parts. By analyzing the histograms, the values are distributed which helps in completely hiding the original image.

#### *b: CHI-SQUARE TEST*

It is a statistical test to justify the distribution of the values of the pixel intensity. It represents the distribution of the values of the pixel intensity. It is calculated using Equation (9) [20], [21]:

$$x^2 = \sum_{i=0}^{255} \frac{(x_i - \bar{x})^2}{\bar{x}} \tag{9}$$

where $x_i$ is the frequency of pixel value $i$, $\bar{x}$ is the average of frequency of pixel values from 0 to 255.

Table 7 shows the results of the Chi-square calculation. The results of the original images are high (between 20856 and 16711680). On the other hand, the proposed model predicts results of encrypted images (from 249 to 284), which proves the great distribution of the values after applying the model.

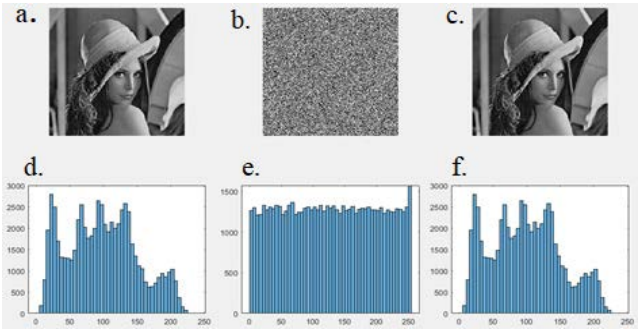


**FIGURE 6.** Histogram of Lena image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

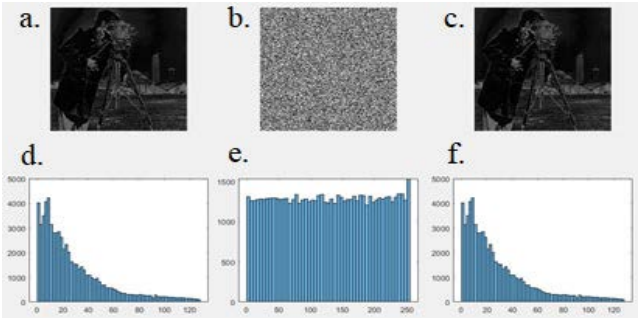


**FIGURE 7.** Histogram of Cameraman image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

### *c: INFORMATION ENTROPY*

It is a statistical measure of randomness. It is calculated using Equation (10) [22]:

$$H = -\sum_{i=1}^{M}\sum_{j=1}^{N}\left(p\left(C\left(i,j\right)\right) * \log_{10} p\left(C\left(i,j\right)\right)\right) \qquad (10)$$

where $C\left(i,j\right)$ is the pixel value in encrypted image, $p(C\left(i,j\right))$ is the probability of the occurrence of $C\left(i,j\right)$. The entropy of encrypted image should be near 8 [2], [6], [7].

In Table 8, the proposed model is compared to the models in [2]–[5], [7], [8] on Lena, Cameraman, Baboon, Peppers, Black, and White images in terms of entropy. The proposed model outperforms [4] in the Cameraman image with a value of 7.9973, as indicated in the table. In Lena, Baboon, Peppers, and White images, it is as excellent as [2]–[5], [7], [8] models with values of 7.9971, 7.9970, 7.9971, and 7.9970, respectively. As seen in the table, the model predicts results that are 0.0001 to 0.0002 better than those anticipated by others.

### *d: IRREGULAR DEVIATION*

It is employed to calculate the deviation of pixel value before and after encryption. It is calculated using equation (11) [4]:

$$\mathrm{ID} = \sum_{i=0}^{255} |\mathrm{HD}_i - \mathrm{MH}| \qquad (11)$$

where $\mathrm{HD}_i$ is the histogram of the absolute values of the difference between original and cipher images, MH is the

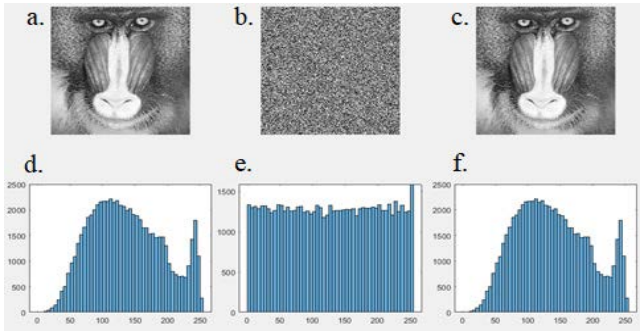


**FIGURE 8.** Histogram of Baboon image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

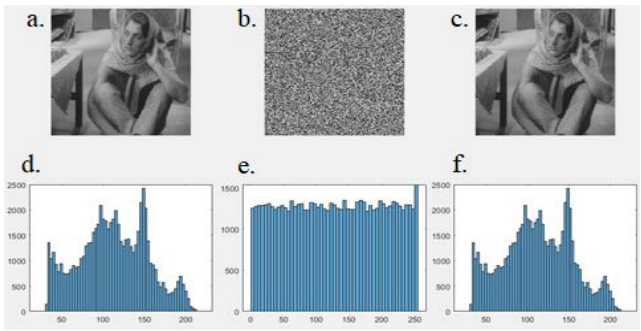


**FIGURE 9.** Histogram of Barbara image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

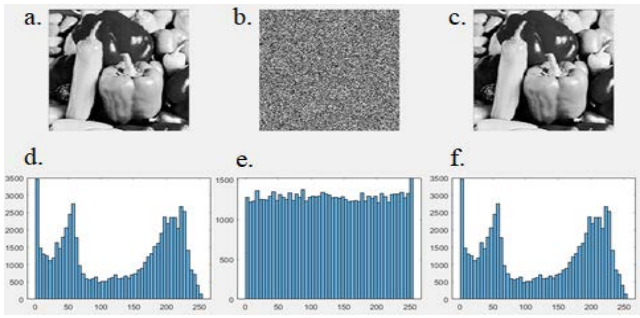


**FIGURE 10.** Histogram of Peppers image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

average of HD. The expected value of ID should be of minimum value which indicates that the histogram is close to uniformity [23]. Table 6 contains the results of applying ID test on the dataset.

### *e: CORRELATION COEFFICIENT ADJACENT (HORIZONTAL, VERTICAL, DIAGONAL)*

It is employed to calculate the similarity between two adjacent pixels. In original images, they always have a high correlation coefficient which is vulnerable to statistical attack. Meanwhile in encrypted images, the correlation coefficient should be near 0% [2], [6], [7]. It is calculated

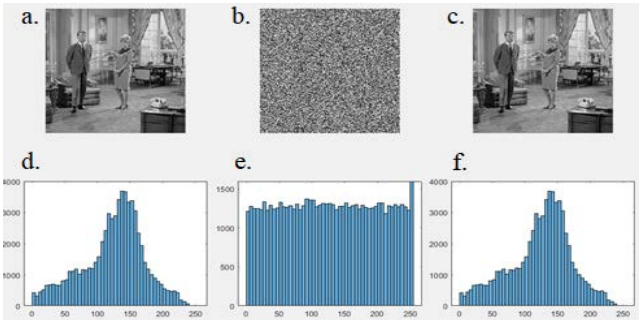


**FIGURE 11.** Histogram of Couple image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

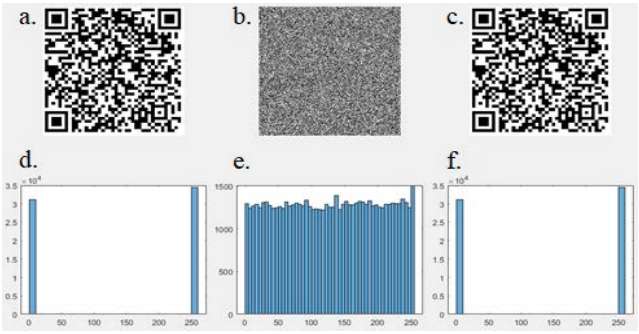


**FIGURE 12.** Histogram of QR Code image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

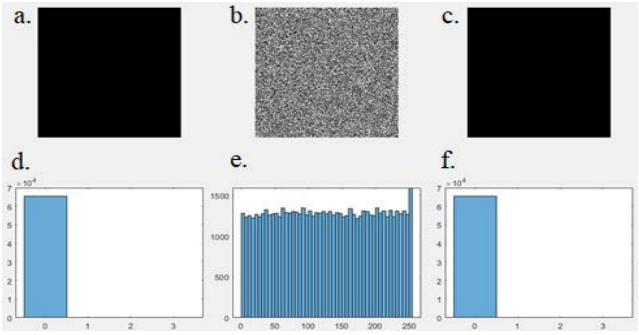


**FIGURE 13.** Histogram of Black image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

using equation (12), as shown at the bottom of the page, [24], where $C_{A1}(i,j)$ and $C_{A2}(i,j)$ are gray-scale value of adjacent pixels, $M$ and $N$ are dimensions of the image, $\bar{C_{A1}}(i,j) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} C_{A1}(i,j)$ and $\bar{C_{A2}}(i,j) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} C_{A2}(i,j)$. The correlation coefficient is tested on horizontal, vertical, and diagonal directions.

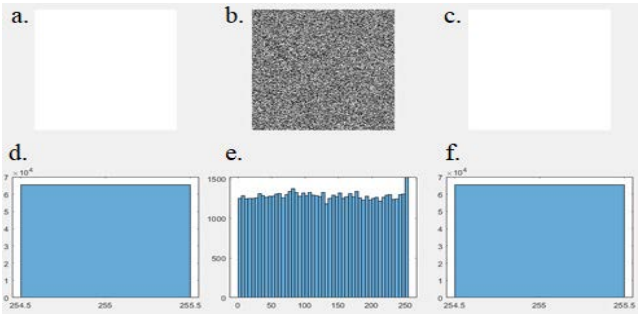


**FIGURE 14.** Histogram of White image: a. original image, b. encrypted image, c. decrypted image, d. histogram of original image, e. histogram of encrypted image, f. histogram of decrypted image.

In terms of CCA, Table 9 compares our proposed model to [2]–[5], [7] models using Lena, Cameraman, Peppers, Black, and White images. As seen in the table, the proposed model is as good as the previous techniques and tends to 0.

The actual improvement in CCA can be detected by computing the average. Equation (13) [1] is used to determine the average:

$$\text{Correlation Coefficient} = (|\text{HC}| + |\text{VC}| + |\text{DC}|)/3 \quad (13)$$

where HC, VC, and DC are correlation coefficient horizontally, vertically, and diagonal, respectively.

Table 10 shows the results of the CCA average calculation. The proposed model predicts results that are 0.0001% to 0.0232% stronger than those predicted by others.

#### 4) DIFFERENTIAL ATTACKS ANALYSIS

Differential attacks analysis tests detect the relationship between the original and encrypted images. The differential attacks analysis descriptions and results are described in this section.

##### *a: NUMBER OF PIXELS CHANGE RATE (NPCR)*

It is the rate of the number of pixels changed of the encrypted image from the original image. It is calculated using Equations (14), (15):

$$\text{NPCR} = \sum_{i,j} \frac{D(i,j)}{MN} * 100\% \quad (14)$$

$$D(i,j) = \begin{cases} 0 & if\ C_1(i,j) = C_2(i,j) \\ 1 & if\ C_1(i,j) \neq C_2(i,j) \end{cases} \quad (15)$$

where $C_1(i,j)$ and $C_2(i,j)$ are respectively encrypted images before and after changing one pixel of original image. The expected value of NPCR should be near 100% [2], [6], [7].

On Lena, Cameraman, Baboon, Peppers, Black, and White images, Table 11 compares our proposed model to [3]–[5],

$$cc = \frac{\left|\sum_{i=1}^{M}\sum_{j=1}^{N}\left[C_{A1}(i,j) - \bar{C_{A1}}(i,j)\right]\left[C_{A2}(i,j) - \bar{C_{A2}}(i,j)\right]\right|}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[C_{A1}(i,j) - \bar{C_{A1}}(i,j)\right]^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[C_{A2}(i,j) - \bar{C_{A2}}(i,j)\right]^2}} \quad (12)$$

**TABLE 6.** Proposed model results.

| Criteria | LENA | Cameraman | Baboon | Barbara | Peppers | Couple | QR Code | Black | White |
|---|---|---|---|---|---|---|---|---|---|
| Encryption Time (s) | 7.9 | 8 | 6.3 | 8.7 | 8.6 | 8.4 | 5.8 | 5.8 | 5.6 |
| Decryption Time (s) | 4.5 | 5.1 | 5.4 | 5 | 5.2 | 4.9 | 5.2 | 5.3 | 5.1 |
| NPCR | 99.68% | 99.61% | 99.61% | 99.63% | 99.66% | 99.62% | 99.60% | 99.61% | 99.64% |
| UACI | 33.57% | 33.56% | 33.29% | 33.44% | 33.56% | 33.49% | 33.29% | 33.47% | 33.38% |
| MSE | 8972 | 16178 | 21725 | 7480 | 11967 | 7669 | 21725 | 21801 | 21760 |
| Entropy | 7.9971 | 7.9973 | 7.9971 | 7.9970 | 7.9973 | 7.9970 | 7.9972 | 7.9971 | 7.997 |
| ID | 9904 | 4569 | 10249 | 11656 | 7071 | 11705 | 1234 | 1 | 933 |
| CC (H) | 0.0042 | 0.0058 | -0.0091 | -0.0014 | -0.0081 | 0.0092 | -0.0091 | -0.0019 | 0.0014 |
| CC (V) | 0.000049 | -0.0111 | -0.0005 | 0.0052 | 0.0031 | 0.0088 | -0.0005 | -0.0024 | 0.0033 |
| CC (D) | 0.0033 | -0.0039 | 0.0042 | -0.0018 | -0.0021 | 0.0054 | 0.0042 | -0.0003 | 0.0014 |

**TABLE 7.** $X^2$ results.

| Image | LENA | Cameraman | Baboon | Barbara | Peppers | Couple | QR Code | Black | White |
|---|---|---|---|---|---|---|---|---|---|
| Plain Image | 30666 | 299789 | 20856 | 53078 | 28838 | 45732 | 8343778 | 16641637 | 16711680 |
| Encrypted Image | 256 | 257 | 249 | 262 | 262 | 284 | 253 | 260 | 264 |

**TABLE 8.** Information entropy comparison with other models.

| References \ Dataset | LENA | Cameraman | Baboon | Peppers | Black | White |
|---|---|---|---|---|---|---|
| Zhang et al. [2] | 7.9975 | NA | NA | NA | 7.9971 | 7.9970 |
| Wang et al. [3] | 7.997 | NA | NA | 7.9971 | 7.9973 | 7.9971 |
| ElKhamy et al. [4] | 7.9973 | 7.9971 | 7.9972 | 7.9977 | NA | 7.997 |
| Wang et al. [5] | 7.9975 | NA | NA | 7.9975 | 7.9974 | 7.9973 |
| Xu et al. [7] | 7.9974 | 7.9973 | NA | NA | 7.997 | 7.997 |
| Tian et al. [8] | 7.9977 | NA | 7.9973 | 7.9974 | NA | NA |
| **Proposed Model** | **7.9971** | **7.9973** | **7.9971** | **7.9970** | **7.9971** | **7.9970** |

[7], [8] models with respect to NPCR. In the Lena, Peppers, and White images, the proposed model outperforms models [3]–[5], [7], [8] with values of 99.68%, 99.66%, and 99.64%, respectively. Moreover, with a value of 99.61%, it is comparable to [3], [5], [7], and superior to [4] in Black image. Regarding the Cameraman and Baboon images, with values of 99.61% and 99.61%, respectively, it is comparable to others. After examining these numbers, the proposed model surpasses others by 0.02% to 0.09% in respect of NPCR.

### *b: UNIFIED AVERAGE CHANGING INTENSITY (UACI)*

It detects the distance between the average intensity of the original image and encrypted image. The expected value is near 33% [2], [6], [7]. It is calculated using Equation (16):

$$\text{UACI} = \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255 * MN} * 100\% \tag{16}$$

where $C_1(i,j)$ and $C_2(i,j)$ are respectively encrypted images before and after change of one pixel of original image.

Table 12 compares our proposed model to [3]–[5], [7], [8] models with respect of UACI on Lena, Cameraman, Baboon, Peppers, Black, and White images. In Baboon, Black, and White images, the proposed model outperforms [3]–[5], [7], [8] with values of 33.29%, 33.47%, and 33.38%, respectively. Moreover, with values of 33.57%, 33.56%, and 33.56%, it is comparable to [3], [5], [7], [8] in the Lena, Cameraman, and Peppers images. After examining these numbers,

**TABLE 9.** CCA comparison with other models.

| References \ Dataset | | Lena | Cameraman | Peppers | Black | White |
|---|---|---|---|---|---|---|
| Zhang et al. [2] | H | 0.0018 | NA | NA | -0.0091 | -0.0194 |
| | V | 0.0003 | NA | NA | -0.0210 | 0.0134 |
| | D | -0.0014 | NA | NA | -0.0075 | 0.0428 |
| Wang et al. [3] | H | 0.0011 | NA | 0.0040 | NA | NA |
| | V | 0.0013 | NA | 0.0015 | NA | NA |
| | D | 0.0053 | NA | 0.0018 | NA | NA |
| ElKhamy et al. [4] | H | 0.0005 | 0.0002 | 0.0003 | 0.0014 | 0.0011 |
| | V | 0.0009 | 0.0003 | 0.0011 | 0.0012 | 0.0022 |
| | D | 0.0002 | 0.0012 | 0.0023 | 0.0025 | 0.0045 |
| Wang et al. [5] | H | 0.0003 | NA | 0.0002 | 0.0038 | -0.0015 |
| | V | - 0.00003 | NA | 0.00007 | -0.0008 | 0.0019 |
| | D | - 0.00003 | NA | - 0.00009 | -0.0035 | -0.0003 |
| Xu et al. [7] | H | 0.9725 | 0.9565 | NA | 0.0080 | 0.0013 |
| | V | 0.0010 | 0.0006 | NA | 0.0044 | −0.0065 |
| | D | −0.0029 | 0.0014 | NA | −0.0024 | 0.0022 |
| **Proposed Model** | **H** | **0.0042** | **0.0058** | **-0.0081** | **-0.0019** | **0.0014** |
| | **V** | **0.000049** | **-0.0111** | **0.0031** | **-0.0024** | **0.0033** |
| | **D** | **0.0033** | **-0.0039** | **-0.0021** | **-0.0003** | **0.0014** |

the model results are better than others by 0.01% to 0.23% in respect of UACI.

### c: MEAN SQUARE ERROR (MSE)

It represents the diffusion characteristics of an image. It is applied on encrypted image. The theoretical value is >10,000 [2], [6], [7]. It is calculated using Equation (17):

$$\mathrm{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(I\left(i,j\right) - C\left(i,j\right)\right)^2 \qquad (17)$$

where $I\left(i,j\right)$ and $C\left(i,j\right)$ are the pixel values of original image and encrypted image, respectively.

In Table 13, our proposed model is compared to the [3], [4] models, on images of Lena, Cameraman, Baboon, and Peppers. With values of 8972, 16178, 21725, and 11967, the model outperforms [3], [4] on the Lena, Cameraman, Baboon, and Peppers images, respectively. As seen in the table, the proposed model predicts results that are greater than those predicted by others by a range of 1170 to 21680.

## C. RESULTS INTERPRETATION

Recent studies have employed various models to encrypt images, as indicated in the related work section. In respect of their dataset, there is a discrepancy in the number of images used to verify their models. Only one approach applied his work [4] on six different images, whereas four of them [3], [5], [7], [8] used four images, and only [2] used three images. On the other hand, the proposed model results are robust resulting from a comprehensive assessment over nine images.

**TABLE 10.** CCA average comparison with other models.

| Dataset / Reference | Lena | Cameraman | Peppers | Black | White |
|---|---|---|---|---|---|
| Zhang et al. [2] | 0.0021 | NA | NA | 0.0125 | 0.0252 |
| Wang et al. [3] | 0.0026 | NA | 0.0024 | NA | NA |
| ElKhamy et al. [4] | 0.0005 | 0.0006 | 0.0012 | 0.0017 | 0.0026 |
| Wang et al. [5] | 0.0001 | NA | 0.0001 | 0.0027 | 0.0012 |
| Xu et al. [7] | 0.3255 | 0.3195 | NA | 0.0049 | 0.0033 |
| **Proposed Model** | **0.0025** | **0.0069** | **0.0044** | **0.0015** | **0.0020** |

As shown in Table 9, the diversity of the images – in terms of the color distribution - had a significant impact on the results. If the histogram of the original image is distributed uniformly, the values of CCA results decrease significantly.

**TABLE 11.** NPCR comparison with other models.

| Reference \ Dataset | LENA | Cameraman | Baboon | Peppers | Black | White |
|---|---|---|---|---|---|---|
| Wang et al. [3] | 99.59% | NA | NA | 99.61% | 99.62% | 99.58% |
| El Khamy et al. [4] | 99.61% | 99.63% | 99.61% | 99.63% | 99.56% | NA |
| Wang et al. [5] | 99.62% | NA | NA | 99.61% | 99.62% | 99.62% |
| Xu et al. [7] | 99.61% | 99.61% | NA | NA | 99.62% | 99.6% |
| Tian et al. [8] | 99.59% | 99.62% | 99.61% | 99.61% | NA | NA |
| **Proposed Model** | **99.68%** | **99.61%** | **99.61%** | **99.66%** | **99.61%** | **99.64%** |

**TABLE 12.** UACI comparison with other models.

| Reference \ Datasets | LENA | Cameraman | Baboon | Peppers | Black | White |
|---|---|---|---|---|---|---|
| Wang et al. [3] | 33.51% | NA | NA | 33.44% | 33.5% | 33.45% |
| El Khamy et al. [4] | 33.46% | 33.44% | 33.44% | 33.46% | 33.39% | NA |
| Wang et al. [5] | 33.41% | NA | NA | 33.45% | 33.49% | 33.58% |
| Xu et al. [7] | 33.48% | 33.5% | NA | NA | 33.47% | 33.39% |
| Tian et al. [8] | 33.45% | 33.42% | 33.58% | 33.52% | NA | NA |
| **Proposed Model** | **33.57%** | **33.56%** | **33.29%** | **33.56%** | **33.47%** | **33.38%** |

For example, the histogram of the Cameraman image in Fig. 7 shows that the number of pixels with values around zero exceeds 4050. The results of $X^2$ test in Table 7 prove the uniform distribution of the histogram. However, this number is dramatically decreased until the pixel values reach 255. When the CCA average result of this image is compared to the CCA average result of the other images, it is discovered that most of them, including ours, were affected by this feature, since the CCA average results of the Cameraman image specifically vastly exceeded the other images.

From another perspective, because the proposed model contains more key generation layers, even minor changes to the original image magnifies the difference between the encrypted image before and after the modification (compared to recent studies). The reason behind that is the key generated using hash functions (SHA-256 and MD5 algorithms), which are extremely sensitive to small changes in the input. NPCR and UACI results (in Table 11–XII) demonstrate this effect, with NPCR results ranging from 99.61% to 99.68%, whereas the UACI results range from 33.29% to 33.57%. This effect is also supported by the key sensitivity analysis.

Also, the execution time of encryption and decryption processes vary in the proposed model. The key generation steps depend on the size of original image. The key is generated using hashing functions which are applied on the image and its metadata, leading to more execution time. Therefore, encryption process takes more time than decryption process.

**TABLE 13.** MSE comparison with other models.

| Reference \ Dataset | LENA | Cameraman | Baboon | Peppers |
|---|---|---|---|---|
| Wang et al. [3] | 7802 | NA | NA | 9215 |
| ElKhamy et al. [4] | 40.84 | 40.82 | 40.79 | 40.94 |
| **Proposed model** | **8972** | **16178** | **21725** | **11967** |

Additionally, the increase in encryption layers increases the difference between the original image and the encrypted image while keeping the statistical randomness of a cipher image. This effect appears in the MSE and ID results as shown in Table 13 and 6, respectively. The MSE results ranged dramatically between 8972 and 21725, while results in [4] ranged between 40.82 and 40.94 and in [3] ranged between 7802 and 9215. In addition, ID results ranged between 1 and 11705. This can be attributed to variations of the characteristics in each layer. Some layers (ART) depend on image scrambling without any change in its pixels' values. Others are responsible for changing the pixels values within the image. It is applied with the five chaotic maps and the three matrices (generated using HCS) in performing diffusion to the image. DNA encoding, and DNA operations increased this effect. These two categories of encryption techniques make the proposed model more secure.

To summarize, the structure of the proposed model, the key generation methodology and the dataset image features are the main reasons why the proposed model outperforms other benchmark approaches in most of the performance measures. The CCA and $X^2$ results were affected by differences in image characteristics in all results, with slight variations in the ratio. The proposed model results were affected by the key generation levels that was superior to the majority of the other works; moreover, the execution time is affected with the key generation levels. In respect of the MSE and ID results, the difference between the proposed model and other works is magnified by the variation in encryption layer categories.

## VI. CONCLUSION

DNA encryption has become increasingly relevant in recent years. It has a low power consumption, a high information density, and a large number of parallelisms. This research focuses on improving the image encryption key generation as well as the system security. In this paper, we introduce a novel model that consists of five phases to achieve these objectives. To enhance the key analysis, computational complexity analysis, statistical attacks analysis, and differential attacks analysis, the proposed model was applied to nine popular images and assessed using twelve performance measures.

The results show that the model is resistant to statistical and differential attacks. The model is very sensitive to minor changes in the original image and the key. In the future, we plan to add a security layer by integrating the encrypted image into a DNA sequence that will be difficult to extract in addition to applying dynamic DNA coding to the model and improving the execution time of the model.

## REFERENCES

[1] I. Aouissaoui, T. Bakir, and A. Sakly, "Robustly correlated key-medical image for DNA-chaos based encryption," *IET Image Process.*, vol. 15, no. 12, pp. 2770–2786, Oct. 2021, doi: 10.1049/ipr2.12261.
[2] Y. Zhang, L. Zhang, Z. Zhong, L. Yu, M. Shan, and Y. Zhao, "Hyper-chaotic image encryption using phase-truncated fractional Fourier transform and DNA-level operation," *Opt. Lasers Eng.*, vol. 143, Aug. 2021, Art. no. 106626, doi: 10.1016/j.optlaseng.2021.106626.
[3] X. Wang, W. Xue, and J. An, "Image encryption algorithm based on LDCML and DNA coding sequence," *Multimedia Tools Appl.*, vol. 80, no. 1, pp. 591–614, Jan. 2021, doi: 10.1007/s11042-020-09688-7.
[4] S. E. El-Khamy, N. O. Korany, and A. G. Mohamed, "A new fuzzy-DNA image encryption and steganography technique," *IEEE Access*, vol. 8, pp. 148935–148951, 2020, doi: 10.1109/ACCESS.2020.3015687.
[5] X. Wang and Y. Li, "Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and DNA sequence," *Opt. Lasers Eng.*, vol. 137, Feb. 2021, Art. no. 106393, doi: 10.1016/j.optlaseng.2020.106393.
[6] M. M. Elamir, W. I. Al-Atabany, and M. S. Mabrouk, "Hybrid image encryption scheme for secure E-health systems," *Netw. Model. Anal. Health Informat. Bioinf.*, vol. 10, no. 1, Dec. 2021, doi: 10.1007/s13721-021-00306-6.
[7] J. Xu, J. Mou, L. Xiong, P. Li, and J. Hao, "A flexible image encryption algorithm based on 3D CTBCS and DNA computing," *Multimedia Tools Appl.*, vol. 80, no. 17, pp. 25711–25740, Apr. 2021, doi: 10.1007/s11042-021-10764-9.
[8] J. Tian, Y. Lu, X. Zuo, Y. Liu, B. Qiao, M. Fan, Q. Ge, and S. Fan, "A novel image encryption algorithm using PWLCM map-based CML chaotic system and dynamic DNA encryption," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 32841–32861, Sep. 2021, doi: 10.1007/s11042-021-11218-y.
[9] S. E. El-Khamy and A. G. Mohamed, "An efficient DNA-inspired image encryption algorithm based on hyper-chaotic maps and wavelet fusion," *Multimedia Tools Appl.*, vol. 80, no. 15, pp. 23319–23335, Jun. 2021, doi: 10.1007/s11042-021-10527-6.
[10] A. Y. Niyat and M. H. Moattar, "Color image encryption based on hybrid chaotic system and DNA sequences," *Multimedia Tools Appl.*, vol. 79, no. 1, pp. 1497–1518, Jan. 2020, doi: 10.1007/s11042-019-08247-z.
[11] V. R. F. Signing, T. F. Fonzin, M. Kountchou, J. Kengne, and Z. T. Njitacke, "Chaotic jerk system with hump structure for text and image encryption using DNA coding," *Circuits, Syst., Signal Process.*, vol. 40, no. 9, pp. 4370–4406, Sep. 2021, doi: 10.1007/s00034-021-01665-1.
[12] M. Liu and G. Ye, "A new DNA coding and hyperchaotic system based asymmetric image encryption algorithm," *Math. Biosci. Eng.*, vol. 18, no. 4, pp. 3887–3906, 2021, doi: 10.3934/mbe.2021194.
[13] J.-J. Chen, D.-W. Yan, S.-K. Duan, and L.-D. Wang, "Memristor-based hyper-chaotic circuit for image encryption," *Chin. Phys. B*, vol. 29, no. 11, Nov. 2020, Art. no. 110504, doi: 10.1088/1674-1056/abbbfe.
[14] W.-W. Hu, R.-G. Zhou, S. Jiang, X. Liu, and J. Luo, "Quantum image encryption algorithm based on generalized Arnold transform and logistic map," *CCF Trans. High Perform. Comput.*, vol. 2, no. 3, pp. 228–253, Sep. 2020, doi: 10.1007/s42514-020-00043-8.
[15] W.-W. Hu, R.-G. Zhou, J. Luo, S.-X. Jiang, and G.-F. Luo, "Quantum image encryption algorithm based on Arnold scrambling and wavelet transforms," *Quantum Inf. Process.*, vol. 19, no. 3, pp. 1–29, Mar. 2020, doi: 10.1007/s11128-020-2579-9.
[16] X. Zhang and Y. Hu, "Multiple-image encryption algorithm based on the 3D scrambling model and dynamic DNA coding," *Opt. Laser Technol.*, vol. 141, Sep. 2021, Art. no. 107073, doi: 10.1016/j.optlastec.2021.107073.
[17] M. Ghebleh, A. Kanso, and D. Stevanović, "A novel image encryption algorithm based on piecewise linear chaotic maps and least squares approximation," *Multimedia Tools Appl.*, vol. 77, no. 6, pp. 7305–7326, Mar. 2018, doi: 10.1007/s11042-017-4634-9.
[18] U. Zia, M. McCartney, B. Scotney, J. Martinez, M. AbuTair, J. Memon, and A. Sajjad, "Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains," *Int. J. Inf. Secur.*, Apr. 2022, doi: 10.1007/s10207-022-00588-5.
[19] Y. Wang, X.-W. Li, and Q.-H. Wang, "Integral imaging based optical image encryption using CA-DNA algorithm," *IEEE Photon. J.*, vol. 13, no. 2, pp. 1–12, Apr. 2021, doi: 10.1109/JPHOT.2021.3068161.
[20] H. Ghazanfaripour and A. Broumandnia, "Designing a digital image encryption scheme using chaotic maps with prime modular," *Opt. Laser Technol.*, vol. 131, Nov. 2020, Art. no. 106339, doi: 10.1016/j.optlastec.2020.106339.
[21] A. Broumandnia, "Image encryption algorithm based on the finite fields in chaotic maps," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102553, doi: 10.1016/j.jisa.2020.102553.
[22] S. Yoosefian Dezfuli Nezhad, N. Safdarian, and S. A. Hoseini Zadeh, "New method for fingerprint images encryption using DNA sequence and chaotic tent map," *Optik*, vol. 224, Dec. 2020, Art. no. 165661, doi: 10.1016/j.ijleo.2020.165661.
[23] N. Ahmed, H. M. S. Asif, and G. Saleem, "A benchmark for performance evaluation and security assessment of image encryption schemes," *Int. J. Comput. Netw. Inf. Secur.*, vol. 8, no. 12, pp. 28–29, Dec. 2016, doi: 10.5815/ijcnis.2016.12.03.
[24] X. Wang, X. Zhu, and Y. Zhang, "An image encryption algorithm based on Josephus traversing and mixed chaotic map," *IEEE Access*, vol. 6, pp. 23733–23746, 2018, doi: 10.1109/ACCESS.2018.2805847.

**NADA H. SHARKAWY** received the B.Sc. degree from the Department of Bioinformatics, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt, where she is currently pursuing the M.Sc. degree.

She is a Teaching Assistant at the Management Information Systems Department, Higher Institute of Computer and Information Technology, El-Shrouk Academy, Cairo.

**YASMINE M. AFIFY** received the Ph.D. degree from the Faculty of Computer and Information Sciences, Ain Shams University, Egypt.

She is currently a Lecturer at the Information Systems Department, Faculty of Computer and Information Sciences, Ain Shams University. She has published more than 20 research papers in international journals and conferences. Her main research interests include information retrieval, bioinformatics, social networks, and knowledge management. She is a referee for several international journals and conferences.

**WALAA GAD** received the B.Sc. and M.Sc. degrees in computer and information sciences from Ain Shams University, Cairo, Egypt, in 2000 and 2005, respectively, and the Ph.D. degree from the Pattern and Machine Intelligence (PAMI) Group, Faculty of Electrical and Computer Engineering, University of Waterloo, Canada.

Her master's degree was about designing and planning a network model in the presence of obstacles using clustering around medoids techniques. She is currently a Professor with the Faculty of Computer and Information Sciences, Ain Shams University. She is the author of several publications. Her current research interests include data science, semantic web, machine learning, bioinformatics, and big data analytics.

**NAGWA BADR** received the B.Sc. degree in computer science, in 1996, and the Ph.D. degree in software engineering and distributed systems from Liverpool John Moores University, U.K., in 2003.

She had done her postdoctoral studies at Glasgow University, U.K. She is currently a Professor and the Dean at the Faculty of Computer and Information Sciences, Ain Shams University. She is also the Head of the committee that contributed to research projects funded by national and international grants of information systems, bioinformatics, business analytic, and health informatics (http://www.heal-plus.eu/). Her current research interests include software engineering, cloud computing, big data analytics, social networking, Arabic search engines, and bioinformatics.