entropy

MDPI

Article

Image Encryption Using a Spectrally Efficient Halton Logistics Tent (HaLT) Map and DNA Encoding for Secured Image Communication

Sakshi Patel and Thanikaiselvan Veeramalai *

School of Electronics Engineering, Vellore Institute of Technology (VIT), Vellore 632014, India; sakshi.patel@vit.ac.in
* Correspondence: thanikaiselvan@vit.ac.in

Abstract: With the advancement of technology worldwide, security is essential for online information and data. This research work proposes a novel image encryption method based on combined chaotic maps, Halton sequence, five-dimension (5D) Hyper-Chaotic System and Deoxyribonucleic Acid (DNA) encoding. Halton sequence is a known low-discrepancy sequence having uniform distribution in space for application in numerical methods. In the proposed work, we derived a new chaotic map (HaLT map) by combining chaotic maps and Halton sequence to scramble images for cryptography applications. First level scrambling was done by using the HaLT map along with a modified quantization unit. In addition, the scrambled image underwent inter- and intra-bit scrambling for enhanced security. Hash values of the original and scrambled image were used for initial conditions to generate a 5D hyper-chaotic map. Since a 5D chaotic map has complex dynamic behavior, it could be used to generate random sequences for image diffusion. Further, DNA level permutation and pixel diffusion was applied. Seven DNA operators, i.e., ADD, SUB, MUL, XOR, XNOR, Right-Shift and Left-Shift, were used for pixel diffusion. The simulation results showed that the proposed image encryption method was fast and provided better encryption compared to 'state of the art' techniques. Furthermore, it resisted various attacks.

Keywords: Halton sequence; chaotic maps; 5D hyper-chaotic system; DNA computation; Lyapunov exponent spectrum; spectral entropy; image encryption

Citation: Patel, S.; Veeramalai, T. Image Encryption Using a Spectrally Efficient Halton Logistics Tent (HaLT) Map and DNA Encoding for Secured Image Communication. *Entropy* **2022**, *24*, 803. https://doi.org/10.3390/e24060803

Academic Editor: Amelia Carolina Sparavigna

Received: 23 April 2022
Accepted: 3 June 2022
Published: 8 June 2022

1. Introduction

In the past few decades, use of technology has grown rapidly. Transmission and reception of text, voice, image or video is possible due to growth in the field of communication. Large amounts of data are transferred from one point to another every second, which gives rise to security concerns regarding private information. Digital images are relevant and contain large amounts of information. Therefore, lots of research has been done in order to prevent unauthorized access to malicious nodes. Many techniques, like image encryption, steganography and watermarking, are used to protect images from attackers [1]. Image encryption is a method whereby the input image is converted to an unreadable format, which cannot be decrypted without a key. There are two main methods through which a meaningful image can be converted into an unreadable format: confusion and diffusion. Confusion is a process whereby the position of the pixels is changed in such a manner that, visually, it becomes impossible to predict the original image. On the other hand, diffusion is a process whereby the pixel values are changed to encrypt the image. AES (Advanced Encryption Standard) and DES (Data Encryption Standard) entail some of the very first algorithms designed for cryptography. Due to new advancements in communication systems these methods are now found to be dated, have many defects and are more suitable for text encryption [2]. In order to overcome these problems, modern research has focused more on chaos theory, encoding techniques and artificial intelligence. Chaotic maps are extremely

sensitive to initial conditions, and are nonlinear, highly random and unpredictable in nature. These characteristics are used in cryptography applications in order to provide better security in the channel [3,4]. The main motivation of the proposed research work was to prevent multimedia threats in the communication channel. Traditional encryption schemes use chaotic maps to confuse and diffuse the pixels. The initial and control parameters used in these algorithms are fixed throughout the encryption process for all input images; which makes the image vulnerable to attack. Higher dimensional maps show more randomness and complex dynamic behavior, compared to traditional one-dimensional chaotic systems.

In general, image encryption methods contain two phases: image scrambling, using chaotic maps, and pixel diffusion, using XOR operation. Various encryption algorithms have been developed by researchers. In [5], a chaos-based cryptosystem was proposed, composed of several rounds of diffusion and substitution. Chaotic maps were used to shuffle the pixels of the image and values were sequentially altered to achieve diffusion. The pixel confusion algorithms included new sorting methods literature, like the quantization unit [6], matrix semi-tensor product theory [7,8], zigzag permutation, graph-theory and fractal sorting matrix [9], to correctly decrypt the image at the receiver's end. Authors in [10] proposed a color image encryption scheme, based on one-time keys and chaotic maps. This algorithm generated keys by means of the MD5 algorithm of the mouse position, which makes the algorithm robust to chosen-plaintext attack. Color image encryption algorithm, using multi dimension systems with bit-level permutation, was proposed in [11]. For the scrambling and diffusion process, a piece-wise linear chaotic map and Chen multi-dimensional system were used, respectively. In [12], the initial keys for the chaotic maps were generated by MD5 algorithm with input image pixels. This method ensured a different key for every input image. The research discussed above were based on the traditional one-dimensional chaotic map, which is also well known to hackers. Therefore, there is a need to design a new robust chaotic map and an efficient encryption algorithm for secured communication. Multidimensional chaotic maps are a new subject of research as they provide better hyper-chaotic behavior than the one-dimensional chaos system. In 1963, the first chaotic attractor was found which led to development of chaos theory in many fields [13]. The hyper-chaotic system enhances randomness and indefiniteness; therefore, it is more popular in engineering applications. In [14], a 5D hyper-chaotic map was used to generate pseudorandom sequences. The obtained sequences were recombined for confusion and diffusion of image pixels. The two popular hyper-chaotic systems are Rössler [15] and Chua's circuit [16]. Chen's hyper-chaotic system [17] is also a popular technique to generate pseudorandom sequences and it is widely used by researchers for image encryption [18]. Authors in [19] used a 5Dhyper-chaotic system for secure communication, based on a microcontroller unit. New multidimensional chaotic systems were obtained in order to achieve better randomness and complex dynamical behavior [20–23]. High-dimensional systems, combined with neural networks, also provide high security and resistance against various attacks [24]. Recent research has been based on compressive sensing, where the input image is compressed to a smaller size which is, then, further embedded with secret data and encrypted to achieve highly secured image communication [25]. This method has an extra advantage of reserving less bandwidth while the data is transmitted to the receiver. Mathematical characteristics of chaotic attractors are rigorously studied. Sequences like the Halton sequence [26], which shows uniform behavior in space, cannot be used in cryptography. These sequences need to be scrambled to obtain non-uniform behavior for security applications. Research in [27], evaluated some pre-existing scrambling techniques of the Halton sequence and also proposed simple techniques, like increasing the number of points between the bases, in order to generate a random Halton sequence. In [28], a survey of randomized quasi-Monte Carlo methods was conducted to study the transformation of uniformly distributed sequences after applying scrambling methods. For better security, bit-level operators were applied on the shuffled image [29,30]. DNA-based pixel diffusion algorithms mainly focus on changing the gray pixel value to DNA base streams. Further, various DNA operations, like compliment or XOR, were applied among the base values

to diffuse the pixels. Finally, the pixels were converted back to gray level values. DNA-fused chaos theory is a very popular and successful method for image pixel diffusion. These approaches are very sensitive to initial conditions and are resistant to various brute attacks. DNA-based encryption technique is blended with chaotic maps, hash functions and other methods to provide better security [31–34]. Authors in [35] generated random sequence, using coupled map lattice and chaotic map, which are then combined with the DNA method for image encryption. DNA encoding was also used for medical image encryption combined with chaotic maps in the frequency domain to achieve robustness in the proposed algorithm [36–41].

The proposed work focused on building a new chaotic map (HaLT map) by combining Logistic Tent map and the Halton sequence. This map can be used in many engineering applications where a system with high chaotic nature is required. In the proposed algorithm, a new HaLT map was used in multimedia security applications to provide better privacy from unauthorized users. In this work, two levels of scrambling process were applied to achieve high randomness among image pixels. The use of a HaLT map and inter–intra bit-level permutation was done to confuse the pixels in the first and second levels of the scrambling process, respectively. Further, a 5D hyper-chaotic map and DNA computations with seven operations were used in the diffusion stage. The initial and control parameters for the 5D chaotic map were generated using MD5 and SHA256 hash functions. The initial seed to the hash functions were pixels of the input image. This method generated a new key for every input image, making the algorithm resistant to cryptography attacks.

The main contributions of the proposed scheme are as follows,

- A novel random sequence (HaLT map) generator is proposed, which combines a CLT map (Combined Logistic Tent map) and the Halton sequence.
- A modified quantization unit is developed to sort the generated HaLT sequence for first level scrambling.
- For second level scrambling, bit-level operations are performed for enhanced security.
- MD5 and SHA256 hash functions are obtained from an original and scrambled image, respectively, and they are used to calculate the initial seeds for a 5D hyper-chaotic map.
- A five-dimension chaotic map is used for DNA computing in order to provide great confidentiality and high security.
- Pixel permutation is performed by double sorting in the quantization unit to efficiently change the pixel position of the matrix.
- Seven DNA operations, namely ADD, SUB, MUL, XOR, XNOR, Right-Shift and Left-Shift, are performed to efficiently diffuse the pixels of the permutated image.
- The selection of DNA rules and seven operations are carried out using the five chaotic sequences obtained from the 5D hyper-chaotic map.

Further organization of this research paper is as follows; preliminaries of the proposed methodology is described in Section 2; step by step explanation of the proposed methodology is given in Section 3; Section 4 discusses the results obtained and the conclusion is given in Section 5.

2. Preliminaries

2.1. Halton Sequence

In statistics, Halton sequence is a standard low-discrepancy sequence which seems random but is deterministic in nature. This sequence generates points in space which cover the domain uniformly [26]. It is very popular among researchers because of its ease of implementation, due to its definition via the radical inverse function. Halton sequence is highly correlated between the inverse function and the base used for different dimensions, which results in poor randomization of the points in space. This sequence is generated

using co-prime numbers as its base. If q is an integer, then it can be expressed in terms of base b as shown in Equation (1),

$$q = \sum_{i=0}^{k} d_i b^i \tag{1}$$

where, d_i is the sequence of digits $d_i \ldots d_2 d_1$, k is the number of points. The q^{th} number in the Halton sequence is given in radical inverse of Equation (2), where all the digits d_i are in the interval 0 to 1.

$$H(q, b) = \sum_{i=0}^{k} d_i b^{i-1} \tag{2}$$

Consider, base $b = 2$ the interval 0 to 1 is divided into half, then into a fourth, then eighth, and so on. Notice that the Halton sequence is basically filling the gaps between the intervals. This behavior is similar to uniform distribution. Therefore, to overcome this phenomenon, Halton sequences are randomized. Instead of using a traditional chaotic method for image scrambling, we proposed an algorithm to scramble Halton sequences using combined chaotic map to obtain a nondeterministic sequence for better pixel scrambling for image encryption applications.

2.2. Cryptographic Hashing

A cryptographic hash function is an algorithm that converts input data (message) into a fixed length of bit array (hash, digest, hash value). This function has many information security applications as it is able to withstand all known cryptanalytic attacks [1]. Using a dedicated algorithm [10,12,25,41], any input data can be converted to a secret hash value. The two hash functions used in the proposed work are:

2.2.1. MD5

Message Digest 5 is a hash function which gives 128 bits of hash value. It has a one-way function which converts any input data to fixed length, hexadecimal, output bits, in order to authenticate the original message.

2.2.2. SHA256

Secure Hash Algorithm-256 is a modified version of the MD5 algorithm, providing more security and authenticity to the original data. The hash value obtained by SHA can take years or decades to break, thus making it unbreakable. Another advantage of this function is its uniqueness, i.e., there are likely to be few, or no, collisions between the two hash values.

The detailed explanation of how these hash functions were used in the proposed work is discussed in Section 3.1.3.

2.3. Chaotic Maps

Chaotic maps are mathematical functions that exhibit chaotic behavior. These maps are used for generating random sequences for various engineering applications [6]. The two chaotic maps used in the proposed algorithm are:

2.3.1. Logistic Map

A logistic map is a member of the chaos family, represented in Equation (3),

$$p_{n+1} = \mu * p_n * (1 - p_n) \tag{3}$$

where, μ is a positive number with range (3.5, 4). It is also known as biotic potential, which has a maximum capacity to generate chaotic values. p_{n+1} and p_n are the outputs in the range (0, 1) for $(n+1)^{th}$ and n^{th} iteration, respectively. A bifurcation diagram of the logistic map is shown in Figure 1.

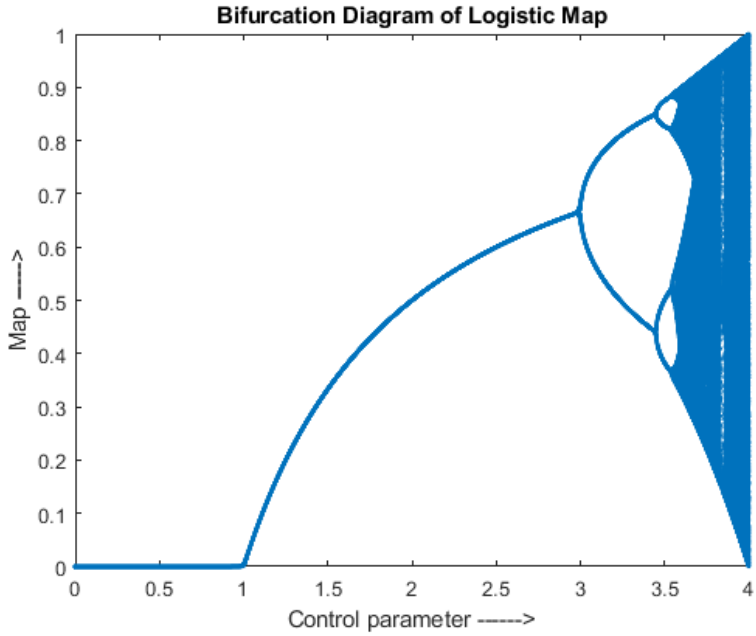


Figure 1. Logistic Map-Bifurcation Diagram.

2.3.2. Tent Map

A tent map is a member of the chaos family, represented in Equation (4),

$$p_{n+1} = \begin{cases} \frac{\alpha * p_n}{2}; for\ p_n < 0.5 \\ \frac{\alpha * (1-p_n)}{2}; for\ p_n \geq 0.5 \end{cases} \quad (4)$$

where, α is a positive number with range (2, 4), p_{n+1} and p_n are the outputs in the range (0, 1) for $(n+1)^{th}$ and n^{th} iteration, respectively. A bifurcation diagram of a tent map is given in Figure 2.

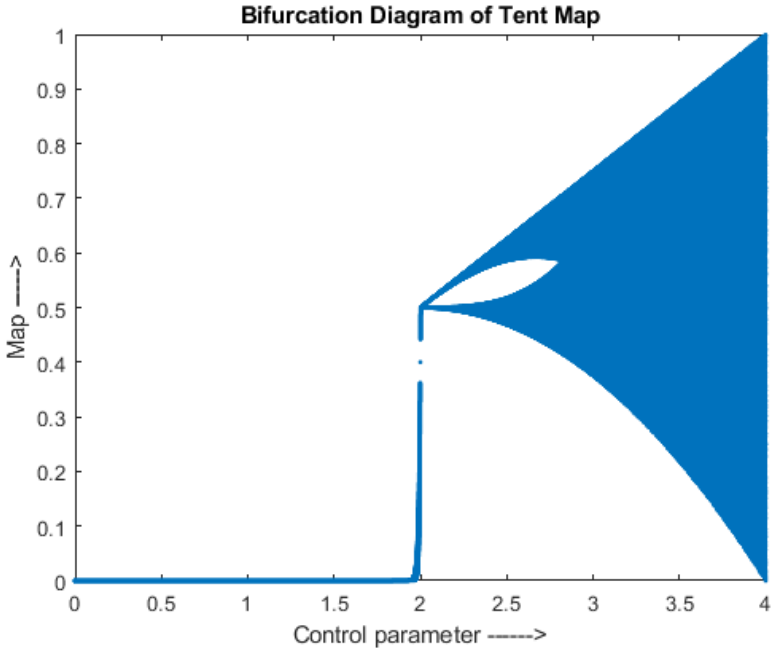


Figure 2. Tent Map-Bifurcation Diagram.

2.3.3. Combined Chaotic Map: Combined Logistic-Tent (CLT) Map

The combined maps are obtained using Equation (5),

$$p_{n+1} = mod(X(x_1, y_1) + Y(x_2, y_2), 1) \tag{5}$$

where, X and Y are the two maps to be combined, x is the initial value and y is the control parameter of the maps. p_{n+1} gives outputs in the range (0, 1) for $(n+1)^{th}$ iteration. A combined Logistic-Tent map was generated as per Equation (5) and it is represented in Equation (6), where X is the logistic map, Y is the tent map, (x_1, x_2) = (p_n,p_n) are the initial values of the logistic and tent maps, respectively, and (y_1, y_2) = (μ,α) are the control parameters of the logistic and tent maps.

$$p_{n+1} = \begin{cases} mod(\mu * p_n * (1 - p_n) + \frac{\alpha * p_n}{2}, 1); for\ p_n < 0.5 \\ mod(\mu * p_n * (1 - p_n) + \frac{\alpha * (1 - p_n)}{2}, 1); for\ p_n \geq 0.5 \end{cases} \tag{6}$$

μ and α are in the range 3.5 to 4 and 2 to 4, respectively. The outputs p_{n+1} and p_n are in the range (0, 1) for $(n+1)^{th}$ and n^{th} iteration, respectively. The bifurcation diagram is shown in Figure 3 which verifies that the map is chaotic in the range (0, 4).

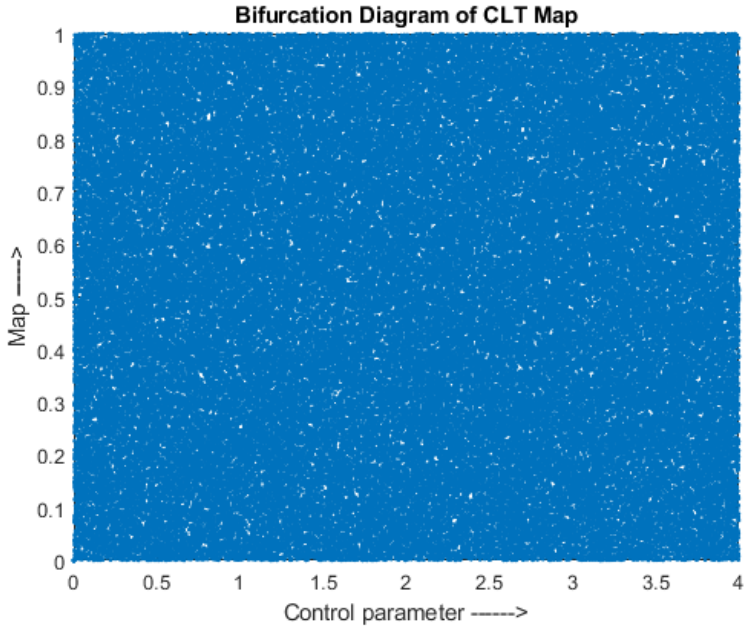


Figure 3. Combined Logistic Tent map—Bifurcation diagram.

2.3.4. Hyper-Chaos System: 5D Hyper-Chaotic Map

Chaos theory is used in many fields, such as secured communication and mathematics. The logistic map discussed above is a one-dimensional chaotic map, used to generate random sequences for various applications. When hyper-chaotic systems are compared with lower dimension chaotic systems, they provide strong confidentiality, high randomness, more complex dynamic behavior, large key space and unpredictability with at least two positive Lyapunov exponents [22]. For any n-dimensional Hyper-Chaotic map consisting of $n-2$ Lyapunov exponents, the following two conditions should be satisfied: firstly, the phase space, where the Hyper-Chaotic map exists, should be at least n, which means the number of coupled first-order differential equations required are n. Secondly, at least two terms should be present in the differential equations which provide dynamic instability, and at least one should be nonlinear in nature. In the proposed research, the following 5D Hyper-Chaotic attractor was used, which is shown in Equation (7). This system was

derived from a 3D modified Lorenz system, by adding a coupling and a nonlinear feedback controller, which gave rise to two quadratic nonlinearities.

$$\begin{cases} \dot{a_1} = b_1(a_2 - a_1) \\ \dot{a_2} = b_3 a_1 + b_4 a_2 - a_1 a_3 + a_3 \\ \dot{a_3} = -b_2 a_3 + {a_1}^2 \\ \dot{a_4} = b_5 a_2 + b_6 a_4 \\ \dot{a_5} = -b_7 a_1 - b_8 a_5 \end{cases} \tag{7}$$

where, b_1, $b_2 > 0$, $b_3 > -b_4$, $b_5 b_7 \neq 0$, b_1, b_2, b_3, b_4 and b_6 are constant parameters, b_5 is the coupling coefficient and b_7, b_8 are control parameters. The 3D and 2D projection of the 5D hyper-chaotic system is shown in Figure 4.

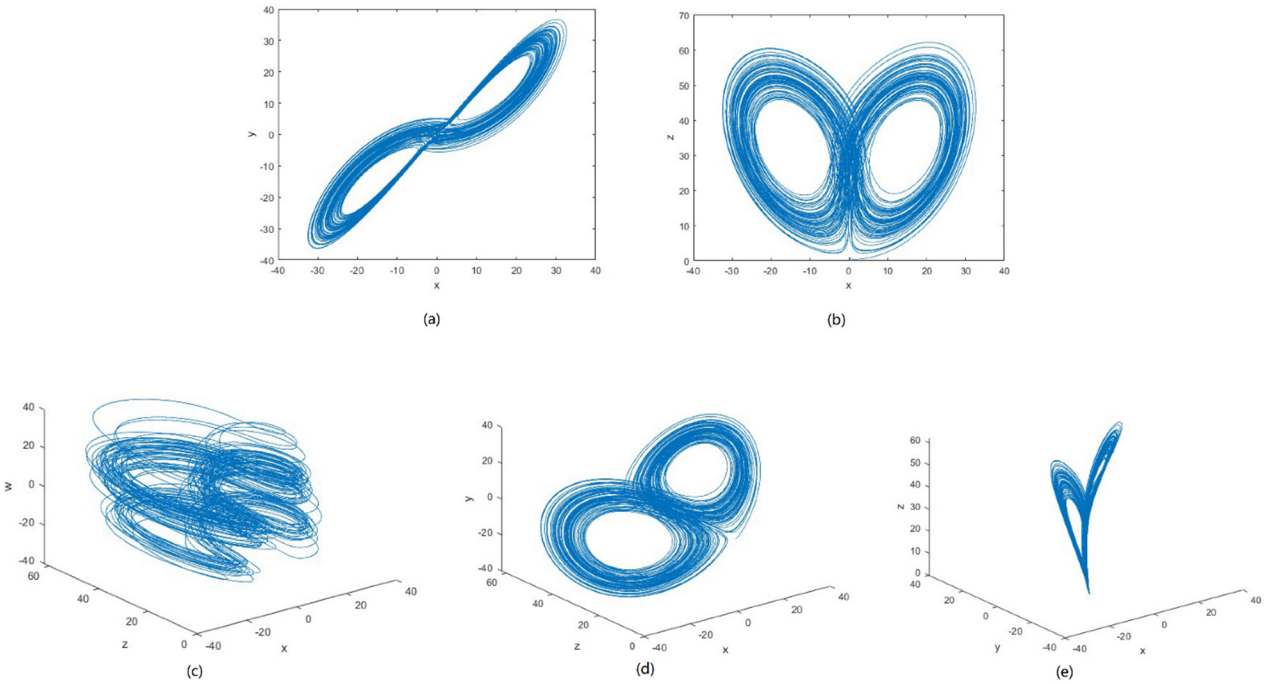


Figure 4. Phase diagram of 5D Hyper−chaotic system with parameters (b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8) = (35, 7, 35, −5, 10.6, 1, 5, 0.05). (**a**,**b**) 2D projection $a_1 - a_2$, $a_1 - a_3$ respectively. (**c**–**e**) 3D projection $a_1 - a_3 - a_5$, $a_1 - a_3 - a_2$, $a_1 - a_3 - a_2$ respectively.

2.4. DNA Computing for Cryptography

Genetic code in a living cell is a term used for the set of instructions that translates DNA information within genetic material into amino acid sequences of proteins. This encoding technique is a vast area of research, not only in the field of biology, but also in other branches, like science and engineering. A DNA sequence contains four types of proteins: namely, Adenine (A), Thymine (T), Cytosine (C), Guanine (G). The bases A and T are complementary, and G and C are complementary to each other. Similarly, in the binary system, 1 and 0 are complementary; therefore, 00 and 11 are also complementary, as are 10 and 01complementary to each other. Using DNA to encode a binary system 00, 11, 10 and 01, there are 4! = 24 combinations possible. Among these 24 combinations, only 8 kinds of bases satisfy the complementary rule, shown in Table 1 [38,39].

Table 1. DNA Rules.

Binary	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	C	G	C	G	A	T	A	T
10	G	C	G	C	T	A	T	A
11	T	T	A	A	G	G	C	C

Randomly, a rule is chosen among eight rules in order to DNA encode image pixels. For example, consider a pixel value 156 in decimal is converted to binary value '10011100'. This binary sequence can be encoded to 8 kinds of amino acid strands, 'GCTA', 'CGTA', 'GCAT', 'CGAT', 'TAGC', 'ATGC', 'TACG', 'ATCG', using the rules in Table 1. Among these 8 DNA sequences, any one rule is chosen for a particular pixel value. The seven possible DNA operations used in this work are: ADD, SUB, MUL, XOR, XNOR, Right Circular Shift and Left Circular Shift. These operations were derived based on fusing mathematics and biological operators, which are shown in Figure 5. The seven DNA operations were performed according to the binary ADD, SUB, MUL, XOR, XNOR operations. In right and left circular shift, the binary value of the image pixels was circularly shifted and the DNA encoded according to the rules given in Figure 5f,g. These operations were used to fuse the plain image pixels with key image pixels. In order to provide good security to the input images, a different operation was used for every pixel value.

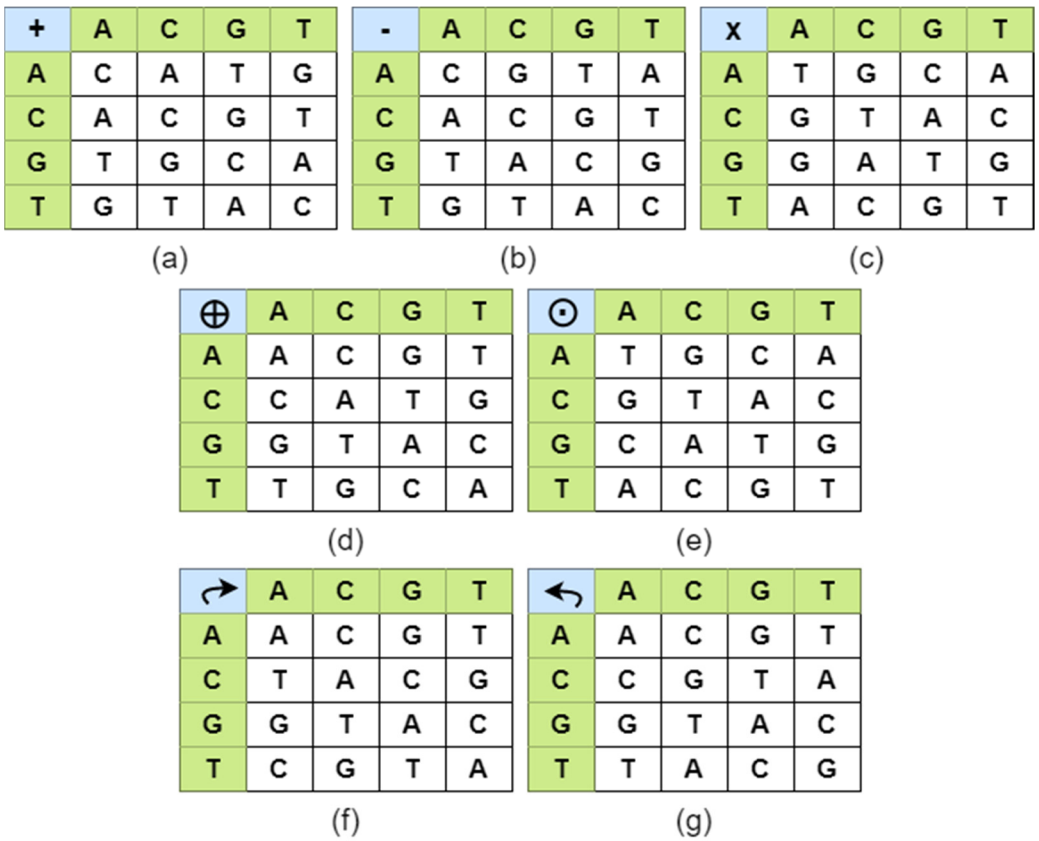


Figure 5. DNA algebraic operations (**a**) ADD, (**b**) SUB, (**c**) MUL, (**d**) XOR, (**e**) XNOR, (**f**) Right circular shift, (**g**) Left circular shift.

3. Proposed Image Encryption Methodology

The proposed image encryption methodology consisted of two phases which are explained in the following subsections. Figure 6 gives the proposed block diagram.

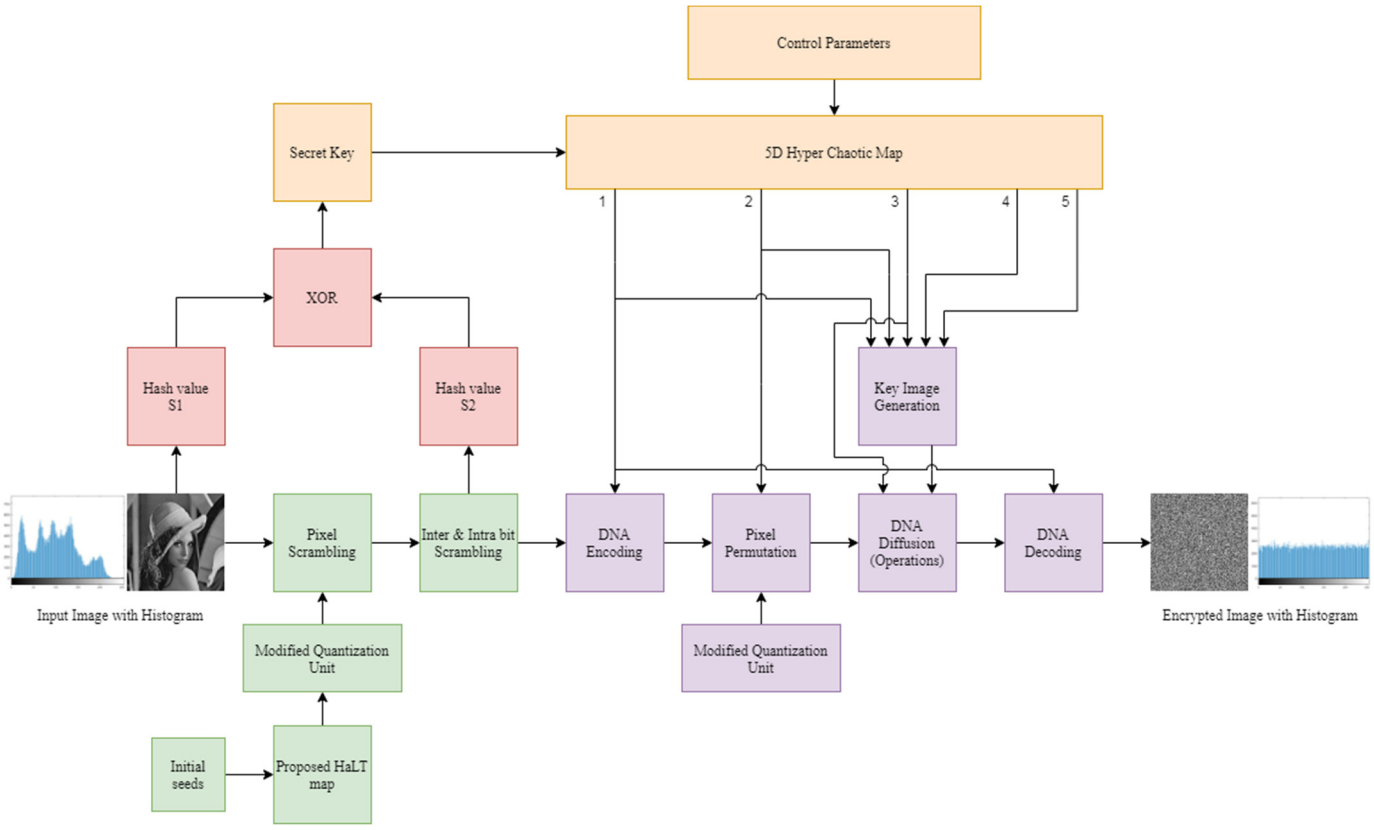


Figure 6. Block Diagram of Proposed Image Encryption Method; 1–5 are the five sequences of 5D Hyper−chaotic map.

3.1. Phase 1

The encryption steps in this phase contain first level scrambling, with the proposed HaLT map, second level scrambling with bit-level operations, and seed generation, using hash functions for the diffusion process.

3.1.1. Proposed HaLT Map

As discussed earlier, Halton sequences are deterministic in nature and show uniform distribution in space. Therefore, there is a need to scramble the Halton sequence to use it in information security applications. In this research, the CLT chaotic map was used to efficiently scramble the Halton sequence.

The idea behind combining any two chaotic maps is given in Equation (5) and it is used in this method. Consider a Halton sequence obtained using Equation (2) with base $b = 4$, the 1000 points generated are given in Figure 7a. To generate a scrambled sequence, the combined chaotic-CLT map was used. As discussed earlier, the CLT map was obtained by combining logistic and tent chaotic maps, which showed high randomness in the range 0 to 1. Similarly, using Equation (5), the Halton sequence and CLT map were combined to get a new HaLT map for input image scrambling. The 1000 random points of the HaLT map in the interval 0 to 1 are shown in Figure 7b. Figure 7c shows 65,536 random points of the generated map for the application in [256 × 256] size image encryption.

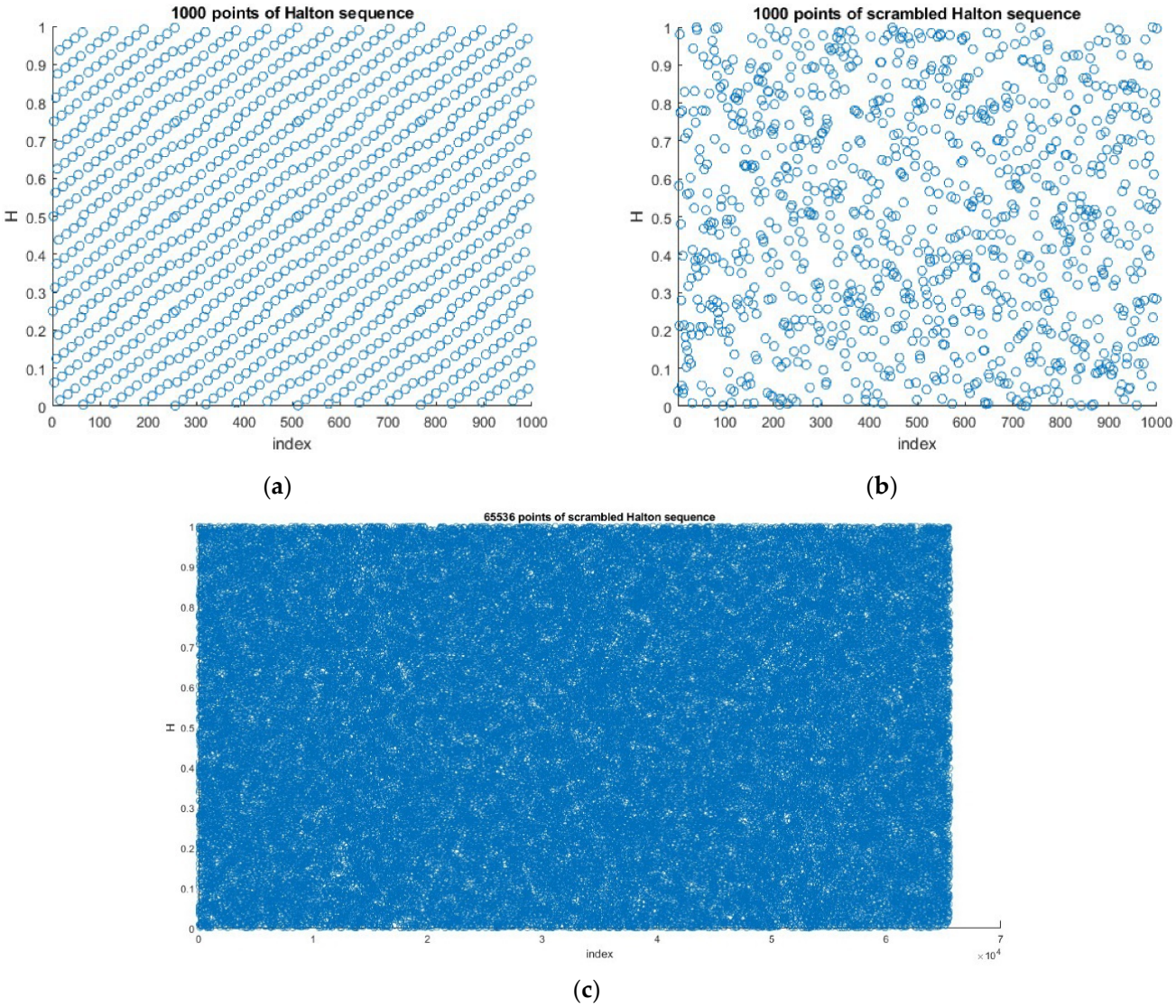


Figure 7. (**a**) 1000 Halton sequence points for base = 4, (**b**) 1000 HaLT map points, (**c**) 65,536 HaLT map points.

In [6], the quantization unit was used for pixel scrambling, where the input image was mapped with the sorted chaotic map. Here, the HaLT map was double sorted, which was then mapped with the input image. Consider the generated HaLT sequence S shown in Figure 8a. This sequence S was sorted and the sorting order s_1 recorded as shown in Figure 8b. In the modified quantization unit, s_1 was arranged in ascending order to obtain s_2, as shown in Figure 8c. This sorted sequence was used to map the input image pixels I given in Figure 8d. Finally, the first level of scrambled image I_m was obtained, as shown in Figure 8e.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0.040	0.581	0.212	0.981	0.774	0.480	0.279	0.054	0.779	0.397	0.034	0.044	0.213	0.829	0.707	0.183

(a) HaLT sequence 'S'

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
11	1	12	8	16	3	13	7	10	6	2	15	5	9	14	4

(b) Sorting order 's_1' of 'S'

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	11	6	16	13	10	8	4	14	9	1	3	7	15	12	5

(c) Sorting order 's_2' of 's_1'

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
128	255	255	0	32	64	16	32	32	64	8	32	128	255	255	128

(d) Input 'I'

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
8	128	32	32	128	255	128	16	64	64	255	255	32	32	255	0

(e) Mapped Image 'I_m' using 's_2'

Figure 8. Modified Quantization Unit; 1–16 are the number of pixels.

3.1.2. Bit-Level Operations

The next step in the first phase of the proposed methodology is inter- and intra-bit scrambling. The proposed bit scrambling algorithm scrambles the bits of the image pixels and the planes of the image. After the first level of image scrambling, bit plane slicing was done to extract the eight planes of the matrix. Now the odd and even number of planes were swapped among each other, as shown in Figure 9. In inter-bit scrambling, the even bit plane pixels were flipped up to down, using Equation (8), where arr_{new} and arr_{old} were the new and old bit plane arrays, respectively, $flipud$ is the function in MATLAB which flips the array up to down, and i is the index for bit plane varying from 1 to 8. The image arrays were then converted from binary to decimal values and reshaped to the size of the original image to obtain a final scrambled image I_s.

$$arr_{new(i)} = \begin{cases} flipud\left(arr_{old(i)}\right) & if \quad arr_{old(i)} = even \\ arr_{old(i)} & if \quad arr_{old(i)} = odd \end{cases} \tag{8}$$

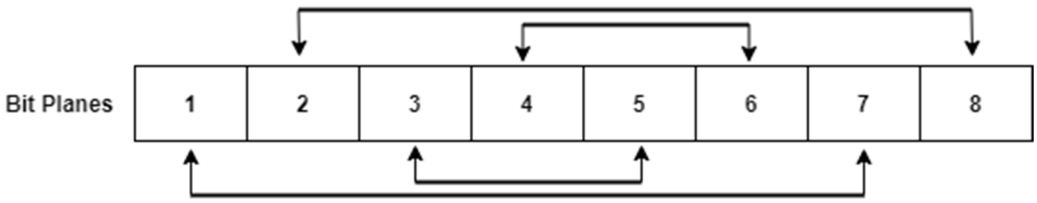


Figure 9. Intra-bit scrambling; 1–8 are the eight planes of the image.

3.1.3. Seed Generation Using Hash Functions

The last step of the first phase was to generate initial seeds for the 5D hyper-chaotic map. In this method, SHA-256 and MD5 hash functions were used. Two hashes were obtained, one from the original image and another from the scrambled image, which were then XOR to create the seed for the chaotic map. Hash functions play a vital role in cryptography, because they are irreversible in nature and they can resist many attacks. Since SHA-256 and MD5 can generate 256 and 128 bits, respectively, they were used in the proposed algorithm to provide better security and prevent various attacks. The number of rows and columns of the original image were considered as $[row \times col]$.

A pixel value at a particular location can be represented as $I(i,j)$, where I is the original image and (i,j) is the i^{th} row and j^{th} column. Three vectors V_1 of size row, V_2 of size col and V_3 of size $[row+col-1]$ were generated from the original image. Here $V_1(i)$ was the sum of all the pixels in the i^{th} row, $V_2(j)$ was the sum of all the pixels in the j^{th} column and $V_3(i)$ was the sum of all the pixels across i^{th} diagonal of image I. The MD5 algorithm was applied on all the three vectors using Equation (9), and then SHA-256 function was applied on the resultant hash value obtained from the previous MD5 algorithm to get a 32-bit hash value, using Equation (10). The above hash value generation algorithm was also applied on the scrambled image I_s to get another 32-bit value. The two hash values obtained were further XOR, as given in Equation (11); the resultant initial seed was used as initial seed for the Hyper-Chaotic map in the next phase, where, M_i^1 and M_i^2 are the MD5 hash values obtained from the original image and the scrambled image, respectively. Then, $i = 1, 2, 3$. S_1 and S_2 were the SHA-256 hash values obtained from M_i^1 and M_i^2 respectively. S was the final 32-bit key used as initial seed in the further steps of proposed algorithm.

$$\begin{cases} M_1^1 = MD5(V_1),\ M_2^1 = MD5(V_2),\ M_3^1 = MD5(V_3) & ,\quad for\ Original\ Image \\ M_1^2 = MD5(V_1),\ M_2^2 = MD5(V_2),\ M_3^2 = MD5(V_3) & ,\quad for\ Scrambled\ Image \end{cases} \tag{9}$$

$$\begin{cases} S_1 = SHA256\left(M_1^1, M_2^1, M_3^1\right) \\ S_2 = SHA256\left(M_1^2, M_2^2, M_3^2\right) \end{cases} \tag{10}$$

$$S = S_1 \oplus S_2 \tag{11}$$

3.2. Phase 2

The encryption steps in this phase contained the 5D hyper-chaotic map, key image generation and, finally, DNA computing.

3.2.1. D Hyper-Chaotic Map

The generated 32-bit key S in the first phase was given as an input for the initial seed of the 5D hyper-chaotic map defined in Equation (7). The control parameters used were (b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8) = (35, 7, 35, −5, 10.6, 1, 5, 0.05). Map values $a_1(i)$, $a_2(i)$, $a_3(i)$, $a_4(i)$ and $a_5(i)$ were initialized using key S, as shown in Equation (12), where $a_1(0)$ was obtained by XOR the first six values of key S. Similarly, the remaining values $a_2(0)$, $a_3(0)$, $a_4(0)$ and $a_5(0)$ were calculated by applying XOR operation on the consecutive six values in the key. After initializing the values, the 5D system was pre-iterated for $[S_{31} + S_{32}]$ times to remove any transition effect. Then, the map was iterated for $[4 \times row \times col]$ times to obtain five chaotic sequences. These sequences were normalized from 0 to 1 using Equation (13), in order to be utilized for key image generation and DNA computing. Where, seq was the five sequences (a_1, a_2, a_3, a_4, a_5), the $large\ value = 10^4$, $\lfloor\ \rfloor$ was floor value.

$$\begin{cases} a_1(0) = XOR(S_{1-6})/256 \\ a_2(0) = XOR(S_{7-12})/256 \\ a_3(0) = XOR(S_{13-18})/256 \\ a_4(0) = XOR(S_{19-24})/256 \\ a_5(0) = XOR(S_{25-30})/256 \end{cases} \tag{12}$$

$$seq = (seq \times large\ value) - \lfloor seq \times large\ value \rfloor \tag{13}$$

3.2.2. Key Image Generation

The map obtained after $[4 \times row \times col]$ times iteration was normalized from 0 to 255, using Equation (14). The key image of size $[4 \times row \times col]$ was obtained from Equation (15), where '*key*' was the key image, a_1, a_2, a_3, a_4 and a_5 were the normalized sequences and i was the index position.

$$seq = mod\left(round\left(seq \times 10^4\right), 256\right) \tag{14}$$

$$\begin{cases} key(i) = a_1(i) \\ key(i+1) = a_2(i) \\ key(i+2) = a_3(i) \\ key(i+3) = a_4(i) \\ key(i+4) = a_5(i) \end{cases} \quad (15)$$

3.2.3. DNA Computing

The steps used in DNA computing were DNA first level encoding, DNA diffusion (operations) and DNA second level encoding.

- DNA Encoding

The scrambled image and the key image were first-level encoded with the help of the 8 DNA rules, shown in Table 1. To randomly choose any one rule among eight DNA rules, sequence $a_1(i)$ was normalized in the range 1 to 8, using Equation (16), where x_1 was a vector containing values from 1 to 8.

$$x\_1(i) = \lfloor 8 * a_1(i) \rfloor + 1 \quad (16)$$

The scrambled image and key image were DNA encoded to get I_{DNA} and K_{DNA}, where each pixel value used a different rule which was randomly chosen by the vector x_1. Modified quantization unit was applied on the sequence y and mapped with the encoded image I_{DNA} to obtain a permutated image I_p.

- DNA Diffusion

As mentioned in earlier sections, seven DNA operations were used in order to diffuse the DNA encoded scrambled image. For every pixel a different operation was used from the seven operations, namely ADD, SUB, MUL, XOR, XNOR, Right Circular Shift and Left Circular Shift, as shown in Figure 5. To randomly choose one operation among the seven, chaotic sequence $a_3(i)$ was normalized in the range 1 to 7, using Equation (17), where z_1 was a vector containing values from 1 to 7, i was the index value.

$$z\_1(i) = \lfloor 7 * a_3(i) \rfloor + 1 \quad (17)$$

DNA operations were selected as follows:
$z_1 = 1; ADDoperation$
$z_1 = 2; SUBoperation$
$z_1 = 3; MULoperation$
$z_1 = 4; XORoperation$
$z_1 = 5; XNORoperation$
$z_1 = 6; Rightcircularshiftoperation$
$z_1 = 7; Leftcircularshiftoperation$

The above operations were applied on the permutated image and encoded key image to get a diffused image I_d. For every pixel value a different operation was used to ensure better security in the algorithm. The vector z_1 randomly chose an operation for a particular pixel to be diffused.

- DNA Encoding

In order to obtain the final encrypted image, I_{en}, the diffused matrix obtained from the above step was second level DNA encoded. The x_1 vector was used to select a rule from the eight DNA rules to encode every pixel.

3.3. Proposed Algorithm Steps

This section gives the step-by-step explanation of the encryption Algorithm 1.

Algorithm 1. Proposed Image Encryption Algorithm

Input—Gray Image I of size $(row,\ col)$, control parameters for hyper-chaotic map, (b_1,b_2, b_3, b_4, b_5, b_6, b_7, b_8) = (35, 7, 35, −5, 10.6, 1, 5, 0.05), parameters for logistic map $(p,\ \mu) = (0.1, 4)$, parameters for tent map $(p,\ \alpha) = (0.1, 4)$, parameters for Halton sequence $b = 4$.
Output—Encrypted Gray Image I_e of size $(row,\ col)$.
Step 1:-Input gray scale image I of size $(row,\ col)$.
Step 2:-Initialize CLT map shown in Equation (6) with control parameters $(\mu,\ \alpha) = (4, 4)$ and initial condition, $p_1 = 0.1$. Iterate it for finite number of times to remove transient effect. Continue iterating the map for $row * col$ times. This process created a chaotic sequence of size $[1,\ row * col]$.
Step 3:-Generate Halton sequence using Equations (1) and (2) of size $[1,\ row * col]$.
Step 4:-Combine the CLT map and Halton sequence, using Equation (5), to form a new chaotic sequence (HaLT map) of size $[1,\ row * col]$. Where X is the CLT map and Y is the Halton sequence.
Step 5:-As shown in Figure 9 (Modified Quantization Unit), double sort the HaLT sequence and map it with the input image I to get scrambled image I_m.
Step 6:-Using bit plane slicing technique extract the eight planes of the image I_m.
Step 7:-Intra-bit scrambling—Swap odd and even number of planes, as shown in Figure 9.
Step 8:-Inter-bit scrambling—Flip all the even plane pixels up to down, using Equation (8).
Step 9:-Convert the values from binary to decimal and reshape the matrix to size $[row \times col]$ to obtain a final scrambled image I_s.
Step 10:-Generate three vectors V_1(sum of all rows), V_2(sum of all columns) and V_3(sum of all pixels across diagonal) of both original image I and scrambled image I_s.
Step 11:-Apply MD5 hash function on V_1, V_2 and V_3, using Equation (9). Further apply SHA256 hash function, as given in Equation (10), to generate two 32-bit hash values S_1 and S_2. Using Equation (11), XOR S_1 and S_2 to obtain S.
Step 12:-Using S for initial seed calculation, as given in Equation (12), and control parameters as (b_1,b_2, b_3, b_4, b_5, b_6, b_7, b_8) = (35, 7, 35, −5, 10.6, 1, 5, 0.05) generate 5D Hyper-Chaotic map given in Equation (7). Iterate the map for finite number of times to remove transient effect. Continue iterating the map for $4 * row * col$ times. This process created a chaotic sequence of size $[1,\ 4 * row * col]$.
Step 13:-Key Image Generation—Normalize all five chaotic sequences in the range 0 to 255, using Equation (14). Key Image '*key*' of size [$4 \times row \times col$] was formed, using Equation (15).
Step 14:-The sequences obtained from Step 12 were normalized in the range 0 to 1, using Equation (13).
Step 15:-DNA Encoding (level 1)—To randomly choose the eight DNA rules for encoding the scrambled image I_s and key image *key*, the sequence a_1 was normalized into values from 1 to 8, using Equation (16). The normalized sequence x_1 randomly chose a rule and encoded I_s and *key* to obtain I_{DNA} and K_{DNA}.
Step 16:-Pixel permutation—Modified quantization unit was applied on the sequence y and mapped with the encoded image I_{DNA} to obtain a permutated image I_p.
Step 17:-DNA Diffusion—To randomly choose the seven DNA operations to apply between I_p and K_{DNA}, sequence a_3 was normalized in the range 1 to 7, using Equation (17). The normalized sequence z_1 randomly chose an operation to be performed between I_p and K_{DNA} to obtain a diffused image I_d.
Step 18:-DNA Encoding (level 2)—Vector x_1 was used to encode the diffused image I_d to get the final encrypted image I_{en}.

The decryption process was the reverse of the proposed encryption algorithm.

4. Results and Discussion

The results and detailed discussions on the generated HaLT map and proposed image encryption technique are demonstrated in this section. Various tests and security analyses were performed to estimate the efficiency and robustness of the proposed work. Experiments were performed by MATLAB2020a on Windows 7 OS, Intel(R) Core(TM) i5-4570U CPU 3.20GHz, 4 GB RAM.

4.1. Simulation Results of Proposed HaLT Map

NIST test, correlation analysis, Lyapunov exponent spectrum and spectral entropy complexity analysis were performed on the HaLT map, described in the subsections below.

4.1.1. NIST Test

The National Institute of Standards and Technology (NIST) is an important document used to analyze the strength of randomness in a sequence. A total of 15 tests were performed on the generated sequence and on the final encrypted image. Table 2 tabulates the results obtained to check whether the generated values are suitable for cryptographic applications. The generated HaLT map was normalized in the range 0 to 255, using Equation (14) to get H_n, then converted to binary values, using Equation (18) for the NIST test. The encrypted image pixels were also converted to binary values, using Equation (18), for the NIST test. Figure 10 shows the 1000 generated HaLT random sequence. The test results show that the proposed HaLT map passed all 15 NIST randomness tests.

$$H_b(i) = \begin{cases} 0 & , \quad if\ H_n(i) \leq 127 \\ 1 & , \quad if\ H_n(i) \geq 128 \end{cases} \tag{18}$$

Table 2. NIST Test Result of Generated Random Sequence of Size 256 × 256 = 65,536 and on seven encrypted images.

NIST Test	Generated HaLT Map		Average of 7 Encrypted Images	
	p-Value	Result	*p*-Value	Result
Frequency	0.45325	Pass	0.46518	Pass
Block Frequency	0.28364	Pass	0.67522	Pass
Run	0.90276	Pass	0.62306	Pass
Longest Run	0.56939	Pass	0.64648	Pass
Rank	0.91141	Pass	0.53462	Pass
DFT	0.84651	Pass	0.35814	Pass
Overlapping Template	0.71166	Pass	0.40358	Pass
Non-Overlapping Template	0.84651	Pass	0.52204	Pass
Linear Complexity	0.94350	Pass	0.35856	Pass
Serial	0.78696	Pass	0.46366	Pass
Approximate Entropy	0.96258	Pass	0.35566	Pass
Cumulative Sums (Forward)	0.61853	Pass	0.52716	Pass
Cumulative Sums (Reverse)	0.39183	Pass	0.55648	Pass
Random Excursions: Chi-Squared	0.86500	Pass	0.89098	Pass
Random Excursions Variant: Counts	0.42371	Pass	0.93064	Pass

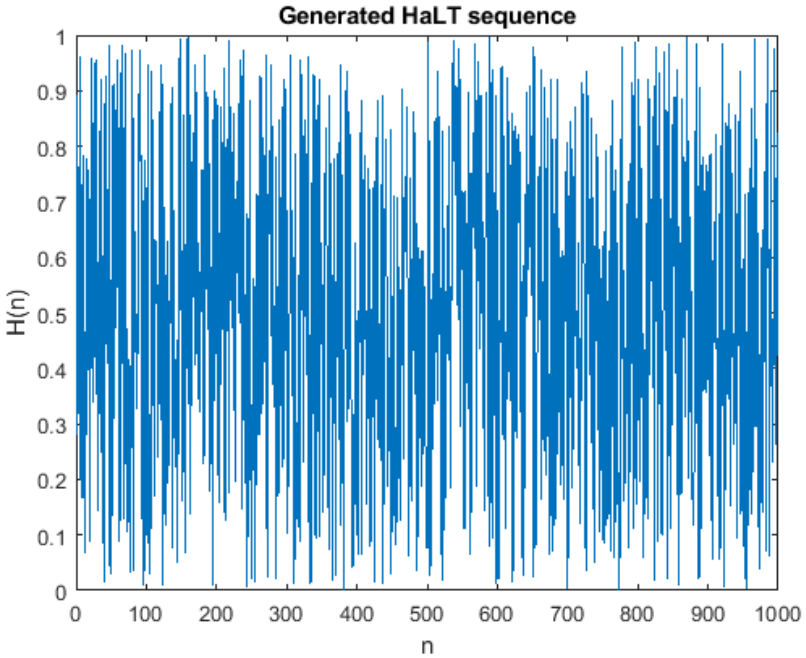


Figure 10. Generated HaLT Sequence.

4.1.2. Correlation

Figure 11 shows the correlation analysis of the generated sequence. The auto-correlation of the sequence is given in Figure 11a. Figure 11b shows the cross-correlation of the two sequences when one of the bits was changed in the initial condition of the CLT map. The correlation plot showed the sensitivity towards initial conditions of the generated HaLT map.

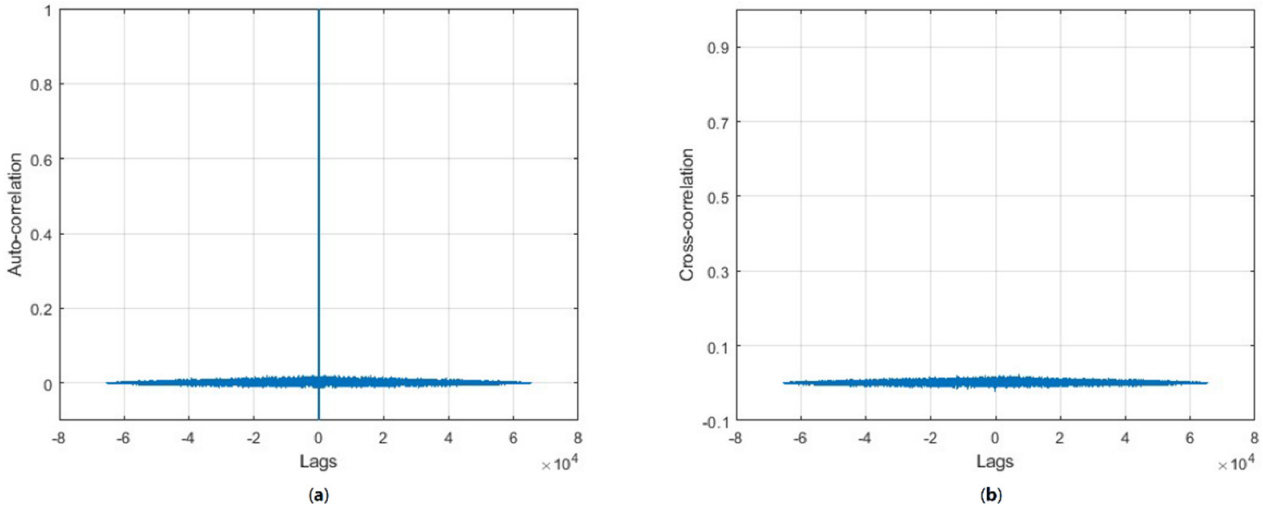


Figure 11. Correlation analysis of the generated sequence (**a**) auto correlation plot, (**b**) cross correlation plot.

4.1.3. Lyapunov Exponent Spectrum

In mathematics, Lyapunov exponents are the measure of a dynamic system's sensitivity to change in its initial conditions. It is an essential method to determine the rate of exponential separation of close trajectories. The separation rate can be different for different initial conditions. Therefore, to determine convergence or divergence of the system in phase space, the Lyapunov exponent spectrum was plotted [42]. The positive maximal Lyapunov exponent determined that the dynamic system was chaotic in nature.

If the Lyapunov exponent was negative, the behavior of the system would be non-chaotic. The evolution equation of a dynamic system, defined by Equation (19), and the spectrum of Lyapunov exponents were given by Equation (20),

$$x_{i+1} = F(x_i) \tag{19}$$

$$\lambda_{F(x)} = \lim_{n\to\infty} \left\{ \frac{1}{n} \sum_{i=0}^{n-1} ln\left|F'(x_i)\right| \right\} \tag{20}$$

Figure 12 is the plot of the Lyapunov exponent spectrum for the logistic map, tent map, CLT map and proposed HaLT map. The spectrum plot shows that the HaLT map provided positive maximal Lyapunov exponents for a larger range, as compared to other standard chaotic maps. Therefore, the derived HaLT map had chaotic behavior and it was more suitable for cryptographic applications.

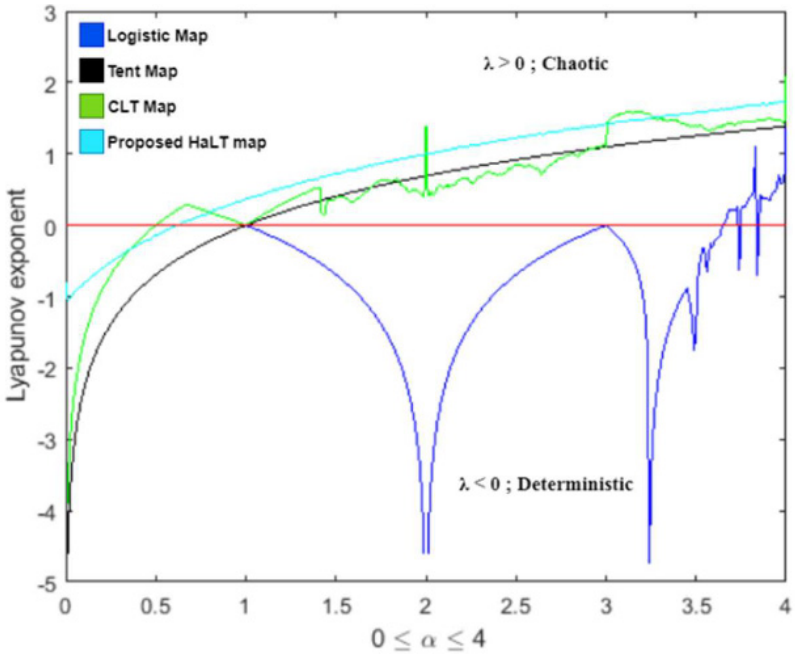


Figure 12. Lyapunov exponent spectrum of logistic map, tent map, CLT map and proposed HaLT map.

4.1.4. Information Spectral Entropy Analysis

The generated HaLT sequence dynamic complexity was determined by Spectral Entropy (SE) analysis. In this paper, the spectral entropy complexity algorithm [43] was used to calculate the behavior complexity of the generated sequence. The SE value was normalized in the range 0 to 1. A greater SE value determined the stronger chaotic nature of the system. Using such random systems for communication purposes gives high information security. The smaller SE value corresponded to lesser complexity in the system, otherwise complexity was high. Figure 13a plots the bifurcation diagram of the generated chaotic sequence, where the map was chaotic in the entire parameter range. Change in system control parameters influenced SE, with r $\in$ (0, 4), as shown in Figure 13b. This plot showed that the SE value was large, with negligible fluctuation, depicting the chaotic nature of the system throughout the parameter range. Figure 13c denotes the chaotic characteristics distribution of the generated sequence versus the system control parameter, r $\in$ (0, 4), and the Halton sequence base value b $\in$ (2, 10). To observe the SE distribution more clearly, the color range of the contour plot was taken from 0.5 (white) to 1 (black). The main color on the contour plot was brown, depicting a larger SE value in the range 0.8971 to 0.9459, showing the chaotic nature of the generated system in the entire parameter range. Therefore, the proposed HaLT map is spectrally efficient and it is more suitable for image encryption applications.

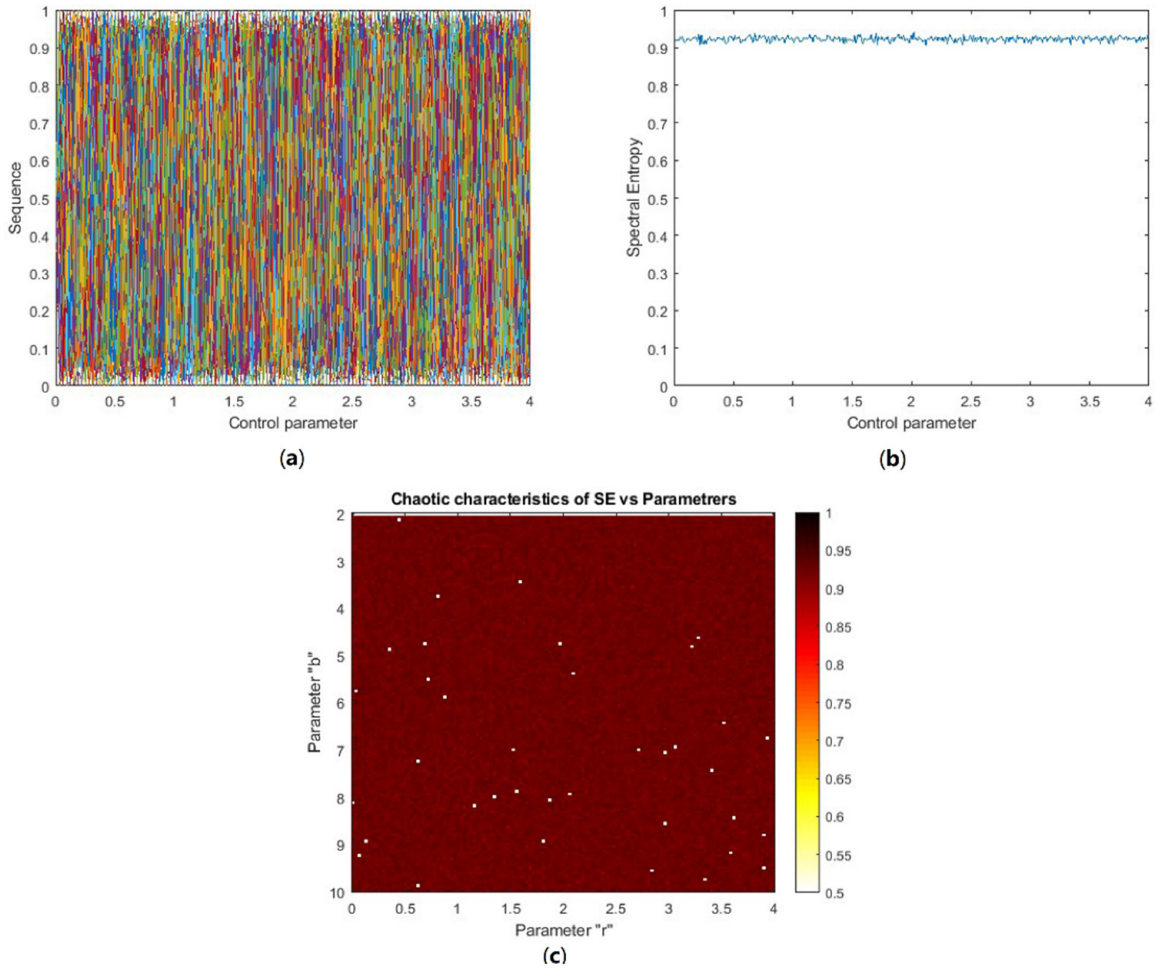


Figure 13. (**a**) Bifurcation diagram of HaLT map; different colors show the results for different initial values and control parameters, (**b**) SE vs. control parameter ($0 \leq r \geq 4$) plot, (**c**) Chaotic characteristics of SE verses parameters of the system, with $0 \leq r \geq 4$ and $2 \leq b \geq 10$.

4.2. Simulation Results of Proposed Image Encryption Algorithm

In this section, various simulations and tests were performed on five test images [44], which are displayed in Figure 14: namely, Lena, Baboon, Goldhill, Cameraman and Bridge of size [256 × 256]. The simulation results were compared with the 'state of the art' techniques [18,35,36,38,41,45–49], on the basis of correlation, information entropy, NPCR, UACI, cropping attack and noise attack. The subsections below show the various tests performed on the proposed method.

Figure 14. Test Images (**a**) Lena, (**b**) Baboon, (**c**) Goldhill, (**d**) Cameraman, (**e**) Bridge.

4.2.1. Statistical Attacks

The statistical analyses performed on the proposed algorithm were histogram analysis, correlation analysis, chi-square test, information entropy and deviation from ideality.

- Histogram Analysis

A histogram is basically a graphical plot of gray intensity levels and pixel distribution in an image. For a secure and robust encryption algorithm, the cipher image was validated on the basis of a uniformly distributed, flat histogram plot for all gray level values. For all five input images, histogram analysis is given in Figure 15, where the cipher images exhibited a uniformly distributed flat histogram plot.

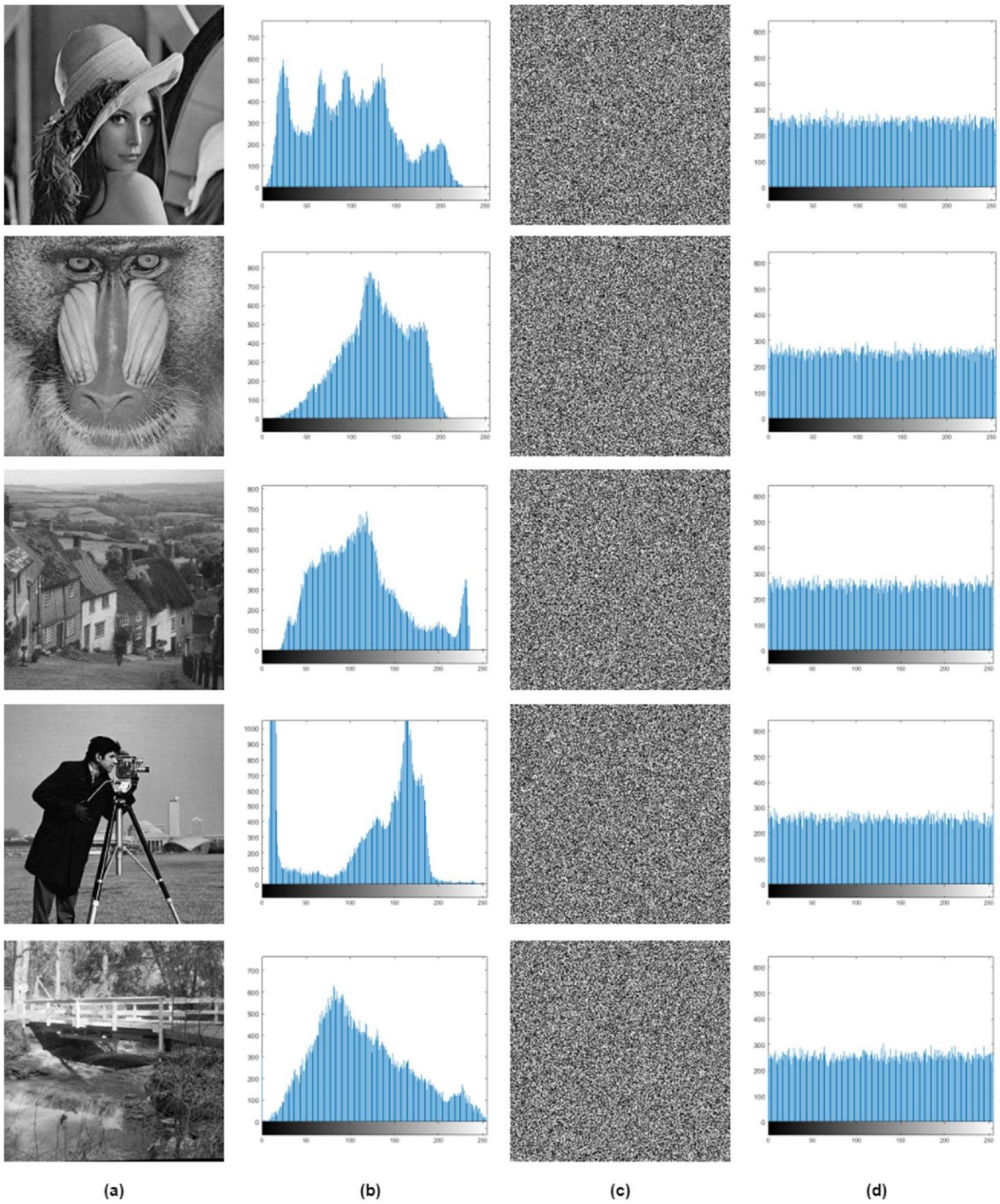


Figure 15. Histogram analysis, *x*-axis in histogram plot represents gray level values and *y*-axis represents number of pixels (**a**) Test Images, (**b**) Test images histogram plot, (**c**) Cipher Images, (**d**) Cipher images histogram plot.

For histogram quantity analysis, a variance of five cipher images is tabulated in Table 3. Most importantly, there were five secret keys (b, μ, α, p(0) and S) used in the proposed encryption algorithm. We also calculated the variance of the same cipher image when any of the five secret keys were changed. Table 1 also shows the % change in the variance value compared with the encrypted image when one of the keys was changed. It was observed that the algorithm was very sensitive to the initial conditions. A lower value of variance indicated higher uniformity in the cipher image histogram [50]. The variance of the Lena plain image was 632100. As shown in the table, the proposed method was very sensitive to the hash key 'S'. Even one bit change in the hash value gave around 10% difference in variance. The % variance difference in the cipher image showed that the algorithm was extremely sensitive to initial conditions and also depended on the input plain image. Therefore, the proposed work was robust to any statistical attacks. The histogram variance was calculated, using Equation (21), where n was the number of gray level values, h_x and h_y were the number of pixels at x and y graylevel and the vector of histogram values $H = (h_1, h_2..h_{256})$.

$$var(H) = \frac{1}{n^2} \sum_{x=1}^{n} \sum_{y=1}^{n} \frac{1}{2} (h_x - h_y)^2 \tag{21}$$

Table 3. Histogram variance analysis of encrypted images.

		Lena	Baboon	Goldhill	Cameraman	Bridge
Encrypted		5460.9	5471.01	5480.8	5455.01	5469.6
b	variance	5256.2	5360.3	5431.5	5361.5	5270.5
	% change	3.7	2.02	0.89	1.71	3.64
μ	variance	5350.8	5272.6	5398.3	5500.4	5342.1
	% change	2.01	3.6	1.50	0.83	2.33
α	variance	5300.4	5479.9	5429.5	5276.4	5398.1
	% change	2.93	0.16	0.93	3.27	1.29
p(0)	variance	5411.3	5358.4	5269.1	5385.7	5210.6
	% change	0.908	2.05	3.86	1.27	4.73
S	variance	5201.3	4975.2	4895.7	5015.01	4830.4
	% change	4.75	9.06	10.67	8.06	11.68

- Correlation Coefficient

The correlation analysis is a test to check the connection between the pixels. Usually, pixels in plain images have high correlation among themselves in all directions. However, for cipher images, this correlation coefficient should be near to zero. Table 4 tabulates the correlation analysis done on all the test images, and the results obtained were compared with the existing techniques [18,35,36,38,41,45]. The correlation values for the test images were very close to the ideal value 0, and also better than many 'state of the art' techniques. The comparison showed that the algorithm broke the correlation between the adjacent pixels and could resist statistical attacks. The left and right column of Figure 16 show the pixel distribution of two adjacent horizontally, vertically and diagonally arranged pixels in the original image and encrypted image, respectively. It was observed that the correlation of input image was very close to 1. Whereas, after applying the proposed algorithm on the plain image, the strong correlation among pixels broke and it was scattered throughout the plane. The equations to calculate correlation analysis are shown in Equations (22)–(24), where a and b represent gray level intensity of the two adjacent pixels, n is the number of pixel pairs, $d(a)$ is variance, $cov(a, b)$ is covariance and $r_{a,b}$ is the correlation coefficient.

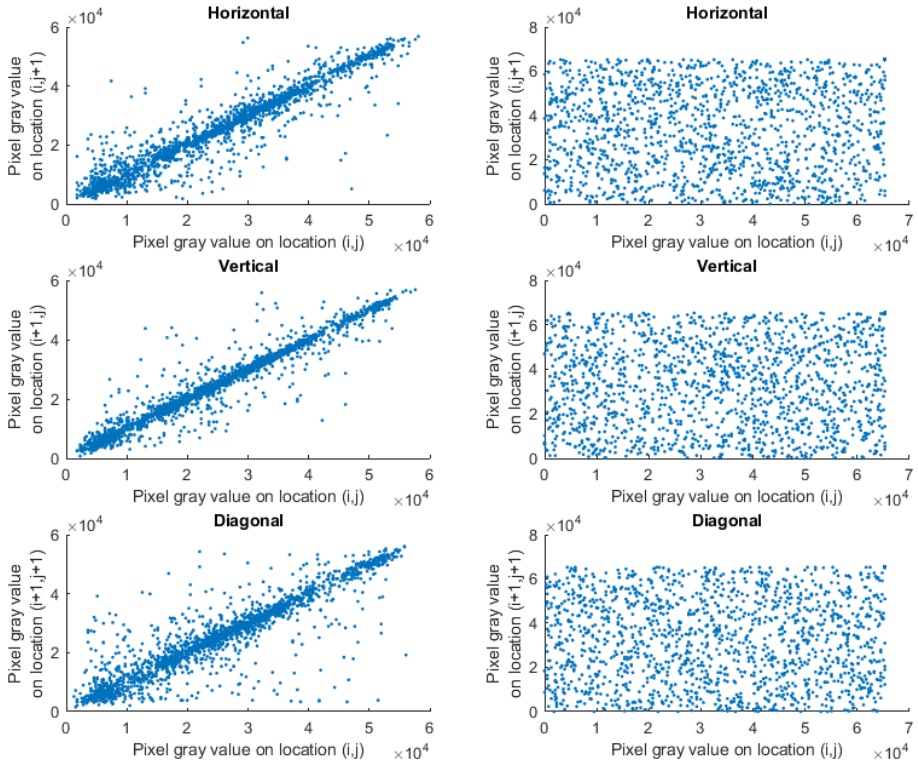
$$r_{a,b} = \frac{cov(a, b)}{\sqrt{d_a}\sqrt{d_b}} \tag{22}$$

$$d(a) = \frac{1}{n} \sum_{j=1}^{n} \left(a_j - \frac{1}{n} \sum_{j=1}^{n} a \right)^2 \tag{23}$$

$$cov(a,b) = \frac{1}{n}\sum_{j=1}^{n}\left(a_j - \frac{1}{n}\sum_{j=1}^{n}a_j\right)\left(b_j - \frac{1}{n}\sum_{j=1}^{n}b_j\right) \tag{24}$$

Table 4. Correlation Analysis.

Metric	Images	Correlation		
		Horizontal	Vertical	Diagonal
Proposed	Lena	0.0034	−0.0052	0.0066
Ref. [45]		0.0197	−0.0196	0.0088
Ref. [41]		−0.0004	0.0037	−0.0378
Ref. [18]		0.0100	0.0083	−0.0143
Ref. [38]		0.0011	−0.0001	−0.0002
Ref. [35]		0.0016	−0.0028	−0.0001
Proposed	Baboon	0.0328	−0.0324	0.0065
Ref. [45]		0.0119	0.0014	−0.0055
Ref. [41]		0.0124	−0.0118	−0.0215
Ref. [18]		−0.0151	0.0006	0.0033
Ref. [38]		−0.0027	−0.0040	0.0047
Proposed	Cameraman	0.0026	0.0108	−0.0111
Ref. [45]		0.0119	0.0175	−0.0179
Ref. [41]		−0.0061	0.0058	0.0166
Proposed	Black	0.0125	0.0105	−0.0019
Ref. [36]		0.0041	0.0063	0.0009
Proposed	White	0.0217	0.0046	−0.0106
Ref. [36]		−0.0028	−0.0005	0.0032
Proposed	Goldhill	0.0014	0.0101	−0.0166
	Bridge	0.0325	0.0061	0.0050



(**a**) (**b**)

Figure 16. Correlation plot in all three directions (**a**) Input image, (**b**) Cipher image.

- Chi-square Analysis

Unlike histogram analysis, which gives a visual representation of pixel distribution, the chi-square test delivers the statistical representation of the pixel uniformity across the gray scale intensities. The test performed on cipher images is given in Table 5, which tabulates the chi-square value and p-value. The values obtained were compared with the 5% and 1% probabilities for $d = 255$ degrees of freedom and passed all the tests, showing uniform distribution of pixel values. Equation (25) gives the chi-square test formula, where L is the gray scale values expected to be 256 for 256×256 image size, I is the observed value and E is the expected value.

$$\chi_d^2 = \sum_{L=0}^{255} \frac{(I-E)^2}{E} \tag{25}$$

Table 5. Chi-Square Analysis.

Images	Chi-Square	p-Value	5% = 293.2478	1% = 310.457
Lena	242.1328	0.7088	Pass	Pass
Baboon	253.9688	0.5066	Pass	Pass
Goldhill	246.0313	0.6451	Pass	Pass
Cameraman	240.9297	0.7276	Pass	Pass
Bridge	245.7344	0.6502	Pass	Pass
White	260.7422	0.3890	Pass	Pass
Black	281.1328	0.1252	Pass	Pass

- Information Entropy

Entropy measures the uncertainty of a message in an image. This value for cipher images having highest uncertainty was expected to be approximately equal to 8. The entropy for the original image and resultant cipher image of the proposed algorithm are tabulated in Table 6; the results obtained were compared with the existing techniques [18,35,36,38,41,45,51]. The values of encrypted images were almost equal to 8 showing high randomness of information in the image and were also better than many 'state of the art' techniques. The formula of entropy is given in Equation (26), where M is the length of pixel value in bits, $p(x_i)$ is the probability of symbol x_i in message x.

$$H(x) = \sum_{i=0}^{2^M-1} p(x_i) log_2 \frac{1}{p(x_i)} \tag{26}$$

Table 6. Information entropy analysis.

Metric	Images	Entropy	
		Original	Encrypted
Proposed	Lena	7.568285	7.997523
Ref. [45]		7.568285	7.9975
Ref. [41]		7.568285	7.9993
Ref. [18]		7.568285	7.9981
Ref. [38]		7.568285	7.9923
Ref. [35]		7.568285	7.9977
Ref. [51]		7.568285	7.9970
Proposed	Baboon	7.228317	7.997656
Ref. [45]		7.228317	7.9975
Ref. [41]		7.228317	7.9993
Ref. [18]		7.228317	7.9983
Ref. [38]		7.228317	7.9925
Ref. [35]		7.228317	7.9973
Ref. [51]		7.228317	7.9969

Table 6. *Cont.*

Metric	Images	Entropy	
		Original	**Encrypted**
Proposed	Cameraman	7.009716	7.997338
Ref. [45]		7.009716	7.9972
Ref. [41]		7.009716	7.9992
Ref. [51]		7.009716	7.9972
Proposed	Black	0	7.996997
Ref. [36]		0	7.9974
Proposed	White	0	7.997839
Ref. [36]		0	7.9969
Proposed	Goldhill	7.471596	7.997290
	Bridge	7.668557	7.997304

- Deviation from Ideality

It is a property which measures the deviation of a resultant encrypted image from an ideal cipher image. The formula to obtain an ideal cipher image is given in Equation (27) and the deviation formula is given in Equation (28), where d is the deviation, $H(I_c)$ and $H(I_I)$ represents the histogram of the encrypted image and of the original image, respectively. To generate a cipher image of size 256 × 256, the number of pixels for all the gray level should be equal to 256, L is the intensity level from 0 to 255. The results obtained for all test images are tabulated in Table 7 and showed a lot less deviation from the resultant encrypted images than from the ideal cipher images.

$$H(I_c) = \begin{cases} \frac{row \times col}{256} & , \quad for\ 0 \leq L \leq 255 \\ 0 & , \quad otherwise \end{cases} \tag{27}$$

$$d = \frac{\sum_{L=0}^{255} |H(I_c) - H(I_I)|}{row \times col} \tag{28}$$

Table 7. Deviation from Ideality.

Images	Deviation
Lena	0.0461
Baboon	0.0443
Goldhill	0.0485
Cameraman	0.0485
Bridge	0.0487
White	0.0498
Black	0.0523

4.2.2. Differential Attacks

The differential analyses of Number of Pixel Change Rate (NPCR) and Unified Average Change in Intensity (UACI) were performed on the proposed algorithm which showed the performance of proposed work when a random pixel is changed in the original image. Table 8 tabulates NPCR and UACI values of the obtained encrypted images when a single bit is randomly changed in the input image. The expected values were: NPCR = 99.6094% and UACI = 33.4635% [52]. The results obtained were near to the expected values and compared with the 'state of art' techniques [18,35,38,41,45,53]. The values of encrypted images were all in the critical range, showing high sensitivity towards input image and were also better than many 'state of the art' techniques. The analysis performed on the proposed method showed that it could resist various differential attacks as it passed all the critical values. Differential analysis was performed on 100 test images of size 256 × 256.

The obtained results are in Figure 17, which shows that all the test images were near to the expected value of NPCR and UACI.

Table 8. Differential Attack Analysis.

Metric	Images	NPCR (%)	UACI (%)	Critical Values (NPCR) 5% = 99.5693 1% = 99.5527 0.1% = 99.5341	Critical Values (UACI) 5% = + = 33.2824 - = 33.6447 1% = + = 33.2255 - = 33.7016 0.1% = + = 33.1594 - = 33.7677
Proposed	Lena	99.6307	33.4740	Pass	Pass
Ref. [45]		99.5392	33.2406	Pass	Fail
Ref. [41]		99.6000	33.4500	Pass	Pass
Ref. [18]		99.6141	33.4473	Pass	Pass
Ref. [38]		99.6387	33.6129	Pass	Pass
Ref. [35]		99.5911	33.4488	Pass	Pass
Ref. [53]		99.5864	33.4808	Pass	Pass
Proposed	Baboon	99.6117	33.5345	Pass	Pass
Ref. [45]		99.5496	33.2543	Pass	Fail
Ref. [41]		99.6000	33.4400	Pass	Pass
Ref. [18]		99.6100	33.4621	Pass	Pass
Ref. [38]		99.6727	33.2071	Pass	Fail
Ref. [35]		99.6078	33.5766	Pass	Pass
Proposed	Cameraman	99.6143	33.4792	Pass	Pass
Ref. [45]		99.5453	33.2742	Pass	Fail
Ref. [41]		99.5900	33.4200	Pass	Pass
Ref. [35]		99.6153	33.4216	Pass	Pass
Proposed	Goldhill	99.6356	33.4663	Pass	Pass
	Bridge	99.6278	33.4641	Pass	Pass
	White	99.6102	33.4644	Pass	Pass
	Black	99.6140	33.4670	Pass	Pass

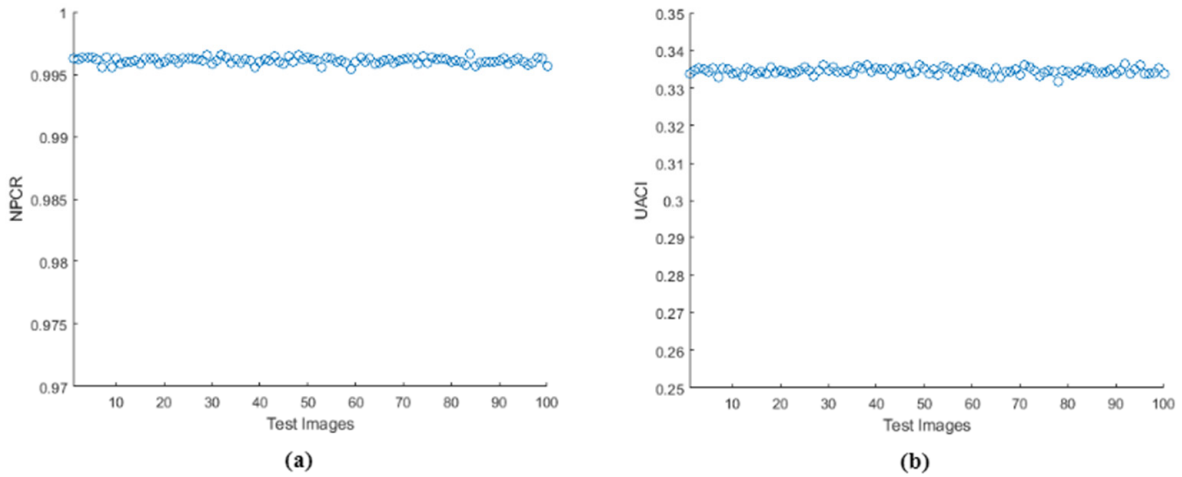


Figure 17. Differential analysis for 100 test images (**a**) NPCR, (**b**) UACI.

- NPCR

This metric calculates the percentage change which occurred in the cipher image when one bit of the input image was changed. The NPCR formula is stated in Equation (29),

where I_{c1} and I_{c2} are the encrypted images before and after randomly changing one bit at (k,l) index position of the input image.

$$NPCR = \frac{\sum_{k,l} M_{(k,l)}}{row \times col} \times 100\% \tag{29}$$

$$M(k,l) = \begin{cases} 1 & if \quad I_{c1}(k,l) \neq I_{c2}(k,l) \\ 0 & if \quad I_{c1}(k,l) = I_{c2}(k,l) \end{cases} \tag{30}$$

- UACI

This metric calculates the average intensity change in pixels when one bit is randomly replaced in the original image. UACI formula is stated in Equation (31),

$$UACI = \frac{1}{row \times col} \sum_{k,l} \frac{|I_{c1}(k,l) - I_{c2}(k,l)|}{255} \times 100\% \tag{31}$$

4.2.3. Key Space and Key Sensitivity Analysis

The most important aspect in an image encryption algorithm is key security analysis. Key space is the maximum possible keys used to encrypt the image. Larger key space will provide more security to an image [54]. In the proposed technique, for generating the HaLT map, two keys for initial condition with 10^{-2} precision, two keys for control parameter with 10^{-1} precision and one key for base parameter were used. For generation of the 5D hyper-chaotic map two 256-bit hash keys and seven control parameters with 10^{-2} precision were required. Hence, an approximately 608-bit key (each key had 8 bits), i.e., 76 Bytes of key, is required to decrypt the image. The key space required was more than 2^{608}, which is large enough to maintain a high security level to resist against brute force attack.

The decryption algorithm, which is extremely sensitive to the initial secret keys used in the encryption algorithm, is considered to be a good method. Even a single bit modification in the encryption key should result in the decryption algorithm failing. The key sensitivity test was performed on test image Lena, illustrated in Figure 18, where Figure 18a is the input image; Figure 18b had single bit change in the initial secret key of the HaLT map generator and Figure 18c had a single bit change in the hash value; showing approximately 99.3% change from the correctly decrypted image, shown in Figure 18d.

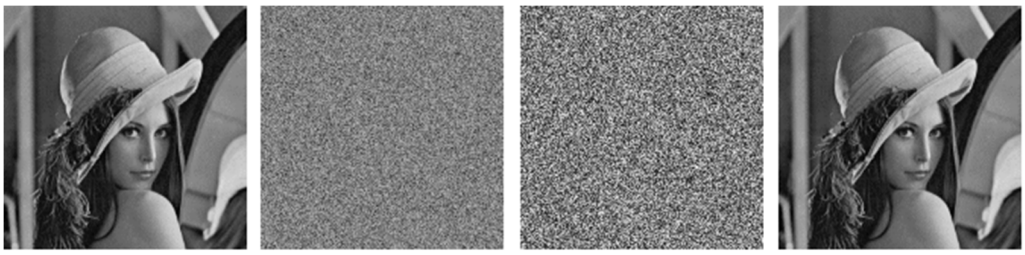


Figure 18. Key sensitivity analysis; (**a**) Input image, (**b**) Decrypted image when single bit changed in HaLT map key, (**c**) Decrypted image when single bit changed in hash key, (**d**) Correctly decrypted image.

4.2.4. Robustness Analysis

This is the measure to test the strength of the proposed image encryption algorithm to withstand adverse conditions. Cropping attack and noise attack is performed in order to test the algorithm. The Peak Signal to Noise Ratio (PSNR) was calculated for all the input images and their corresponding decrypted images to tabulate the quality of the decryption algorithm, given in Table 9. The tabulation also compared the proposed work with 'state of the art; techniques [41,45–47], and shows that the values obtained were approximately equal to the existing methods. Figure 19 shows the 6.25%, 25% and 50% cropping attack

analysis on the Lena test image. The results show that even after 50% cropping of the encrypted image, the decryption algorithm was able to decode the original image, which makes the algorithm robust against cropping attack. For noise attack analysis, salt and pepper noise, with 0.005, 0.05 and 0.1 densities, were used, as demonstrated in Figure 20. The decrypted image showed that the proposed work was robust against noise attack.

Table 9. Robustness analysis.

Metric	Input Images	Crop			Noise		
		6.25%	**25%**	**50%**	**0.005**	**0.05**	**0.1**
Proposed	Lena	20.2853	14.5627	11.9464	31.5440	21.8211	18.8376
Ref. [45]		20.3668	14.3939	11.3895	31.2751	21.2502	18.3147
Ref. [41]		20.3469	14.3600	11.3754	31.4956	21.3079	18.2648
Ref. [46]		20.3745	14.4533	11.4365	30.2494	20.3405	17.4711
Ref. [47]		16.7418	-	-	24.4812	-	-
Proposed	Baboon	20.8067	14.3976	11.6392	31.6535	21.8956	18.9134
Ref. [45]		21.3753	15.3844	12.3608	32.4933	22.2653	19.2684
Ref. [41]		21.2741	15.2579	12.2990	32.2013	22.2380	19.2031
Ref. [46]		21.2852	15.3401	12.3520	31.3731	21.2675	18.3223
Ref. [47]		18.5936	-	-	30.5936	-	-
Proposed	Cameraman	20.5147	14.8108	11.7425	31.1133	21.5995	18.5968
Ref. [45]		20.7255	14.5872	11.5183	31.6209	21.5922	18.6581
Ref. [41]		20.6414	14.6259	11.5914	31.0920	21.2046	18.2498
Ref. [46]		20.3855	14.3947	11.4288	30.8824	20.7612	17.6897
Proposed	Goldhill	20.8903	15.0199	12.3222	31.6763	21.9825	18.9869
	Bridge	20.2805	14.4503	11.5082	30.9841	20.9767	17.8936
	White	19.9051	14.2204	11.0333	30.8069	20.5284	17.2869
	Black	19.9847	13.9664	11.0583	30.9237	20.5690	17.3458

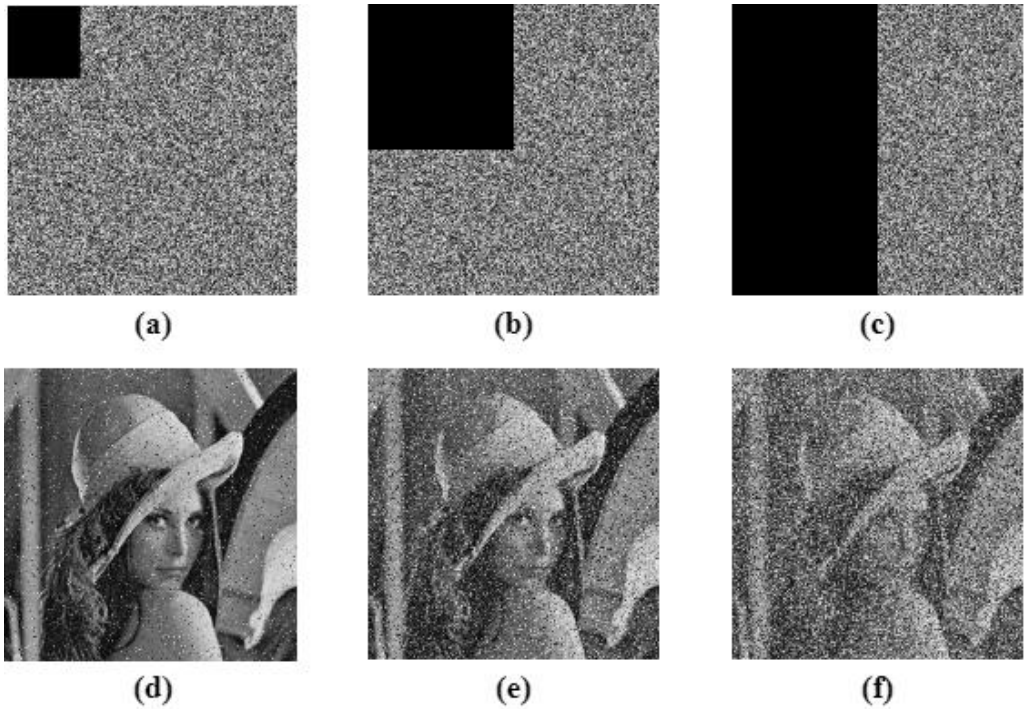


Figure 19. Cropping attack analysis, (**a**–**c**) are 6.25%, 25% and 50% cropping of cipher images respectively, (**d**–**f**) are the corresponding decrypted image.

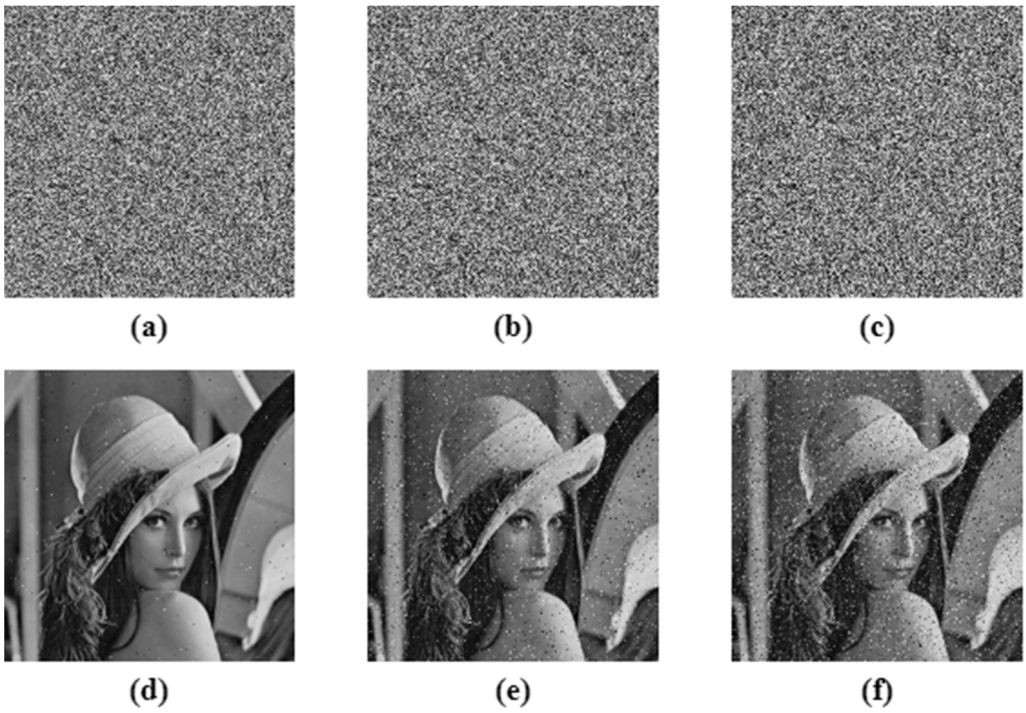


Figure 20. Noise attack analysis, (**a**–**c**) are 0.005, 0.05 and 0.1 noise density cipher images respectively, (**d**–**f**) are the corresponding decrypted image.

4.2.5. Cryptanalytic Attacks

Cryptanalysis is a study in cryptology which helps to find flaws or imperfections in encryption algorithms. In order to find fault in the method we have to attack and then analyze the system with the following [55]:

- Ciphertext Only Analysis (COA): In this type of attack, the attacker knows some ciphertext and tries to find the encryption key and plain text.
- Chosen Plaintext Analysis (CPA): In this, the attacker knows the encryption algorithm, chooses a random plaintext and generates a cipher text to find the encryption key.
- Known Plaintext Analysis (KPA): In this, the attacker maps the known plain text and cipher text to figure out the encryption key.
- Chosen Ciphertext Analysis (CCA): Here the attacker knows the decryption algorithm and tries to find the plain text by using a random cipher text.

Among the cryptanalytic attacks discussed above, CPA is the most common and a powerful attack. This attack analysis is done for encryption algorithms where, in particular, XOR operation is performed for diffusing image pixels. The equation used is given in Equation (32), where I_1 and I_2 are the two test images, Lena and Baboon, and E_1 and E_2 are the two corresponding to cipher images. Figure 21 shows that the proposed algorithm did not justify Equation (32); therefore, it could resist chosen plain text attack.

$$I_1(k,l) \otimes I_2(k,l) = E_1(k,l) \otimes E_2(k,l) \tag{32}$$

If the proposed system can pass the CPA test, then it is resistible to other types of attacks. As discussed in Section 4.2.3, the encryption algorithm is extremely sensitive to initial conditions. The encryption key depends on the plain image itself, which makes it difficult for the attacker to predict the key because it keeps changing with the input plain image.

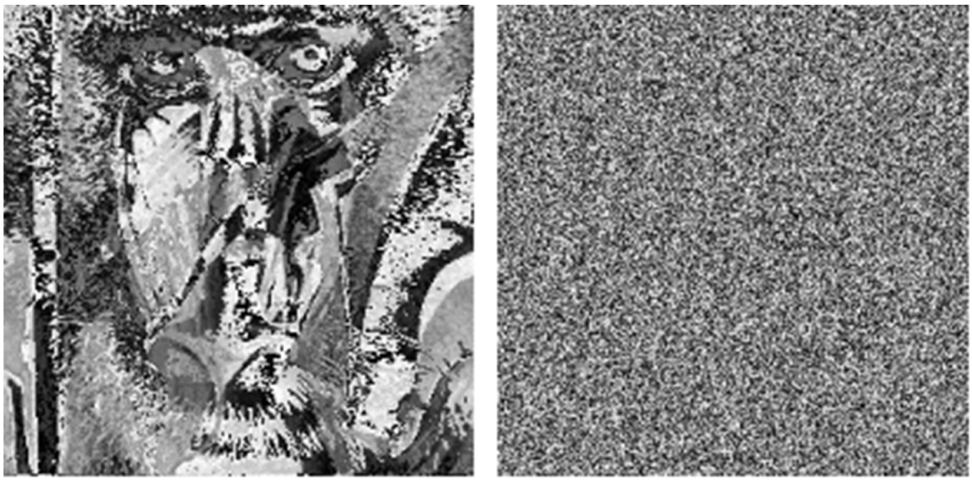


Figure 21. Chosen plain text analysis (**a**) $I_1 \otimes I_2$, (**b**) $E_1 \otimes E_2$.

4.2.6. Execution Time Analysis

In today's world of internet and big data, time analysis plays an important role for encryption algorithms. The execution time for encryption and decryption algorithms on test images are tabulated in Table 10, and compared with exciting techniques [45,46,48,49,56]. All the references used 256 × 256 image size, 4 GB or more RAM size and i5+ core processor. The results obtained showed that the proposed method ran faster than other algorithms and also provided better security.

Table 10. Execution Time Analysis.

Metric	Input Images	Encryption (s)	Decryption (s)
Proposed	Lena	0.329276	0.217033
Ref. [48]		0.40585	-
Ref. [49]		1.7351	3.4689
Ref. [56]		0.3440	-
Ref. [46]		10.8232	10.6952
Ref. [45]		14.8401	14.9266
Proposed	Baboon	0.319188	0.207845
Ref. [46]		10.7477	10.7146
Ref. [45]		14.9134	14.9678
Proposed	Cameraman	0.327988	0.220698
Ref. [49]		1.7223	2.9887
Ref. [46]		10.8053	10.7977
Ref. [45]		15.0087	15.2032
Proposed	Goldhill	0.333641	0.223227
	Bridge	0.339475	0.222850
	White	0.331611	0.214559
	Black	0.313327	0.203875

5. Conclusions

In this research work, we proposed a two-phase image encryption method for secure communication. In the first phase, three methods were used: HaLT map generator, bit-level operation and double sorting quantization unit for pixel scrambling. A method was proposed to generate a random sequence by combining CLT map and Halton sequence to derive a HaLT map for cryptographic applications. Hash values were obtained individually from the original image and the scrambled image using a combination of MD5 and SHA-256 hash function algorithms. The obtained two hash values were then XOR which was

fed to the next phase. The new hash value obtained after XOR was used as seed for a 5D hyper-chaotic map in the second phase. The five pseudorandom sequences generated by the 5D map were used for DNA first level encoding, key image generation, DNA operations and DNA second level encoding. Firstly, the scrambled image obtained from the first phase of the proposed algorithm was DNA encoded. Then, pixel permutation was done by applying the quantization unit on the encoded image. Random sequences obtained from the hyper-chaotic map were used to generate the key image. The permutated image and the key image were diffused using seven DNA operations, namely ADD, SUB, MUL, XOR, XNOR, Right-Shift and Left-Shift. DNA second phase encoding was done on the diffused image to get the final cipher image. The simulation results showed that the proposed HaLT map generator and image encryption algorithm provide high security as they resist various cryptography attacks and are fast for practical applications.

Author Contributions: S.P. and T.V. conceptualization, S.P. and T.V. formal analysis, T.V. funding acquisition, S.P. and T.V. investigation, S.P. and T.V. methodology, T.V. supervision, S.P. and T.V. validation, S.P. writing-original draft, T.V. writing-review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by SENSE School, Vellore Institute of Technology, Vellore, India.

Institutional Review Board Statement: The study was not involving humans or animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: http://sipi.usc.edu/database/database.php?volume=misc (accessed on 20 February 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 5th ed.; Prentice Hall Press: Upper Saddle River, NJ, USA, 2010.
2. Wang, X.; Feng, L.; Zhao, H. Fast image encryption algorithm based on parallel computing system. *Inf. Sci.* **2019**, *486*, 340–358. [CrossRef]
3. Wang, X.; Yang, J. A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient. *Inf. Sci.* **2021**, *569*, 217–240. [CrossRef]
4. Wang, X.; Liu, P. A New Full Chaos Coupled Mapping Lattice and Its Application in Privacy Image Encryption. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 1291–1301. [CrossRef]
5. Wong, K.W. Image Encryption Using Chaotic Maps. In *Intelligent Computing Based on Chaos*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 184, pp. 333–354.
6. Patel, S.; Thanikaiselvan, V.; Pelusi, D.; Nagaraj, B.; Arunkumar, R.; Amirtharajan, R. Colour image encryption based on customized neural network and DNA encoding. *Neural Comput. Appl.* **2021**, *33*, 14533–14550. [CrossRef]
7. Wang, X.; Gao, S. Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. *Inf. Sci.* **2020**, *539*, 195–214. [CrossRef]
8. Wang, X.; Gao, S. Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. *Inf. Sci.* **2020**, *507*, 16–36. [CrossRef]
9. Xian, Y.; Wang, X. Fractal sorting matrix and its application on chaotic image encryption. *Inf. Sci.* **2021**, *547*, 1154–1169. [CrossRef]
10. Liu, H.; Wang, X. Color image encryption based on one-time keys and robust chaotic maps. *Comput. Math. Appl.* **2010**, *59*, 3320–3327. [CrossRef]
11. Liu, H.; Wang, X. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Opt. Commun.* **2011**, *284*, 3895–3903. [CrossRef]
12. Liu, H.; Wang, X.; Kadir, A. Image encryption using DNA complementary rule and chaotic maps. *Appl. Soft Comput.* **2012**, *12*, 1457–1466. [CrossRef]
13. Lorenz, E.N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [CrossRef]
14. Fang, D.; Sun, S. A new secure image encryption algorithm based on a 5D hyperchaotic map. *PLoS ONE* **2020**, *15*, e0242110. [CrossRef]
15. Rossler, O. An equation for hyperchaos. *Phys. Lett. A* **1979**, *71*, 155–157. [CrossRef]
16. Kapitaniak, T.; Chua, L.O. Hyperchaotic Attractors of Unidirectionally-Coupled Chua's Circuits. *Int. J. Bifurc. Chaos* **1994**, *4*, 477–482. [CrossRef]

17. Tong, X.; Chen, P. A joint image lossless compression and encryption method based on chaotic map. *Multimed. Tools Appl.* **2017**, *76*, 13995–14020. [CrossRef]
18. Zhang, Y.; Li, X. A fast image encryption scheme based on integer wavelet and hyper-chaotic system. In Proceedings of the IEEE International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 26–28 May 2018; pp. 139–143.
19. Peng, Z.; Yu, W.; Wang, J.; Zhou, Z.; Chen, J.; Zhong, G. Secure Communication Based on Microcontroller Unit with a Novel Five-Dimensional Hyperchaotic System. *Arab. J. Sci. Eng.* **2022**, *47*, 813–828. [CrossRef]
20. Yang, Q.; Bai, M. A new 5D hyperchaotic system based on modified generalized Lorenz system. *Nonlinear Dyn.* **2017**, *88*, 189–221. [CrossRef]
21. Wolf, A.; Swift, J.B.; Swinney, H.L.; Vastano, J.A. Determining Lyapunov exponents from a time series. *Phys. D Nonlinear Phenom.* **1985**, *16*, 285–317. [CrossRef]
22. Yang, Q.; Chen, C. A 5D Hyperchaotic System with Three Positive Lyapunov Exponents Coined. *Int. J. Bifurc. Chaos* **2013**, *23*, 1350109. [CrossRef]
23. Hu, G. Generating Hyperchaotic Attractors with Three Positive Lyapunov Exponents via State Feedback control. *Int. J. Bifurc. Chaos* **2009**, *19*, 651–660. [CrossRef]
24. Wang, X.-Y.; Yang, L.; Liu, R.; Kadir, A. A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn.* **2010**, *62*, 615–621. [CrossRef]
25. Wang, X.; Liu, C.; Jiang, D. A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT. *Inf. Sci.* **2021**, *574*, 505–527. [CrossRef]
26. Halton, J.H. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.* **1960**, *2*, 84–90. [CrossRef]
27. Ostadkalayeh, A.M.; Vajargah, B.F. The scrambles of Halton sequence and thier weaknesses. *J. Hyperstruct.* **2021**, *9*, 40–53.
28. Ökten, G.; Liu, Y. Randomized quasi-Monte Carlo methods in global sensitivity analysis. *Reliab. Eng. Syst. Saf.* **2021**, *210*, 107520. [CrossRef]
29. Es-Sabry, M.; El Akkad, N.; Merras, M.; Saaidi, A.; Satori, K. A new image encryption algorithm using random numbers generation of two matrices and bit-shift operators. *Soft Comput.* **2020**, *24*, 3829–3848. [CrossRef]
30. Zhang, L.; Zhang, X. Multiple-image encryption algorithm based on bit planes and chaos. *Multimed. Tools Appl.* **2020**, *79*, 20753–20771. [CrossRef]
31. Barik, R.C.; Changder, S. A novel and efficient amino acid codon based medical image encryption scheme colligating multiple chaotic maps. *Multimed. Tools Appl.* **2021**, *80*, 10723–10760. [CrossRef]
32. Zhou, S.; Wang, B.; Zheng, X.; Zhou, C. An Image Encryption Scheme Based on DNA Computing and Cellular Automata. *Discret. Dyn. Nat. Soc.* **2016**, *2016*, 5408529. [CrossRef]
33. Nandy, N.; Banerjee, D.; Pradhan, C. Color image encryption using DNA based cryptography. *Int. J. Inf. Technol.* **2021**, *13*, 533–540. [CrossRef]
34. Chai, X.; Fu, X.; Gan, Z.; Lu, Y.; Chen, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Processing* **2019**, *155*, 44–62. [CrossRef]
35. Tian, J.; Lu, Y.; Zuo, X.; Liu, Y.; Qiao, B.; Fan, M.; Ge, Q.; Fan, S. A novel image encryption algorithm using PWLCM map-based CML chaotic system and dynamic DNA encryption. *Multimed. Tools Appl.* **2021**, *80*, 32841–32861. [CrossRef]
36. Banu, S.A.; Amirtharajan, R. A robust medical image encryption in dual domain: Chaos-DNA-IWT combined approach. *Med. Biol. Eng. Comput.* **2020**, *58*, 1445–1458. [CrossRef] [PubMed]
37. Ravichandran, D.; Banu, S.A.; Murthy, B.K.; Balasubramanian, V.; Fathima, S.; Amirtharajan, R. An efficient medical image encryption using hybrid DNA computing and chaos in transform domain. *Med. Biol. Eng. Comput.* **2021**, *59*, 589–605. [CrossRef]
38. Guan, M.; Yang, X.; Hu, W. Chaotic image encryption algorithm using frequency-domain DNA encoding. *IET Image Processing* **2019**, *13*, 1535–1539. [CrossRef]
39. Chai, X.; Gan, Z.; Lu, Y.; Chen, Y.; Han, D. A novel image encryption algorithm based on the chaotic system and DNA computing. *Int. J. Mod. Phys. C* **2017**, *28*, 1750069. [CrossRef]
40. Li, T.; Yang, M.; Wu, J.; Jing, X. A Novel Image Encryption Algorithm Based on a Fractional-Order Hyperchaotic System and DNA Computing. *Complexity* **2017**, *2017*, 9010251. [CrossRef]
41. Zefreh, E.Z. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. *Multimed. Tools Appl.* **2020**, *79*, 24993–25022. [CrossRef]
42. Vulpiani, A.; Cecconi, F.; Cencini, M. *Chaos: From Simple Models to Complex Systems*; Series on Advances in Statistical Mechanics; World Scientific Publishing Co. Pte. Ltd.: Singapore, 2009; Volume 17.
43. Liu, L.; Du, C.; Zhang, X.; Li, J.; Shi, S. Dynamics and Entropy Analysis for a New 4-D Hyperchaotic System with Coexisting Hidden Attractors. *Entropy* **2019**, *21*, 287. [CrossRef]
44. SIPI Image Database—Misc. Available online: http://sipi.usc.edu/database/database.php?volume=misc (accessed on 20 February 2022).
45. Hu, T.; Liu, Y.; Gong, L.-H.; Ouyang, C.-J. An image encryption scheme combining chaos with cycle operation for DNA sequences. *Nonlinear Dyn.* **2016**, *87*, 51–66. [CrossRef]
46. Chai, X.; Gan, Z.; Yuan, K.; Chen, Y.; Liu, X. A novel image encryption scheme based on DNA sequence operations and chaotic systems. *Neural Comput. Appl.* **2019**, *31*, 219–237. [CrossRef]

47. Hosny, K.; Kamal, S.; Darwish, M.; Papakostas, G. New Image Encryption Algorithm Using Hyperchaotic System and Fibonacci Q-Matrix. *Electronics* **2021**, *10*, 1066. [CrossRef]
48. Girdhar, A.; Kapur, H.; Kumar, V. A novel grayscale image encryption approach based on chaotic maps and image blocks. *Appl. Phys. A* **2021**, *127*, 39. [CrossRef]
49. Yan, X.; Wang, X.; Xian, Y. Chaotic image encryption algorithm based on arithmetic sequence scrambling model and DNA encoding operation. *Multimed. Tools Appl.* **2021**, *80*, 10949–10983. [CrossRef]
50. Zhang, Y.-Q.; Wang, X.-Y. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. *Inf. Sci.* **2014**, *273*, 329–351. [CrossRef]
51. Wang, X.-Y.; Zhang, Y.-Q.; Bao, X.-M. A novel chaotic image encryption scheme using DNA sequence operations. *Opt. Lasers Eng.* **2015**, *73*, 53–61. [CrossRef]
52. Ping, P.; Fan, J.; Mao, Y.; Xu, F.; Gao, J. A Chaos Based Image Encryption Scheme Using Digit-Level Permutation and Block Diffusion. *IEEE Access* **2018**, *6*, 67581–67593. [CrossRef]
53. Wang, X.; Liu, L.; Zhang, Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. *Opt. Lasers Eng.* **2015**, *66*, 10–18. [CrossRef]
54. Zhang, Y.-Q.; Wang, X.-Y. A new image encryption algorithm based on non-adjacent coupled map lattices. *Appl. Soft Comput.* **2015**, *26*, 10–20. [CrossRef]
55. Wang, X.; Teng, L.; Qin, X. A novel colour image encryption algorithm based on chaos. *Signal Processing* **2012**, *92*, 1101–1108. [CrossRef]
56. Ping, P.; Xu, F.; Mao, Y.; Wang, Z. Designing permutation–substitution image encryption networks with Henon map. *Neurocomputing* **2018**, *283*, 53–63. [CrossRef]