Chaotic Image Encryption

Cryptography Project Report

MASTER OF TECHNOLOGY

IN

CYBER SECURITY SYSTEMS AND NETWORKS

Submitted By

ARAVIND AJI

AM.EN.P2CSN19003

AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
AMRITAPURI CAMPUS
KOLLAM - 690525

TABLE OF CONTENTS

INTRODUCTION

Encryption algorithms like Data Encryption Standard(DES) and Advanced Encryption Standard(AES) which are effective in most cases might not be useful for the encryption of an image. This is largely due to the high redundancy among the pixels and the data size. For developing a better algorithm for image encryption, the researches have been using different types of techniques such as DNA coding, optical transform, chaos theory, etc,. [1].

Among these different technologies, chaotic image encryption is proved to be the most efficient one [2]. This is due to the excellent properties such as ergodicity, periodicity, unpredictability and high sensitivity to initial conditions. The chaotic image encryption schemes can be mainly divided into two: One-dimensional and Multi-dimensional. One dimensional encryption schemes are easy to implement but come with certain limitations such as the range of the chaotic behaviour becoming discontinuous and vulnerabilities [2].

Generally a chaotic image encryption scheme has two parts: A chaotic system and the image encryption algorithm. The first paper that we had chosen implements a bit-level image encryption scheme using piecewise linear chaotic maps(PWLCM). Bit-level encryption schemes have many advantages over pixel-level encryption schemes as the former can change the value and the position of a pixel simultaneously [1].

The second paper introduced a simple and efficient one-dimensional chaotic image encryption using two of the existing one-dimensional chaotic maps [2]. Also, the encryption algorithm is able to generate completely different encrypted images each time, when applied with the same key to the same plain image. Both of the algorithms were tested against a number of security analyses and attacks such as key sensitivity, differential attacks, contrast analysis, etc.

CHAOS THEORY

Systems that appear random and disoriented from the outside, but in reality guided by deterministic equations and laws that are highly dependent on the initial conditions are known as Chaotic Systems [3]. Study of these systems is known as Chaos Theory. A chaotic system has many excellent intrinsic properties such as ergodicity and high sensitivity to initial conditions that makes it very useful in areas like encryption of digital images, where the existing encryption algorithms fail to perform well [4].

A chaotic system can be easily observed in the real world in the motion of a double pendulum. If the displacement is small, the motion of a double pendulum is simple harmonic. But as the displacement is made larger and larger, the motion becomes completely chaotic and random. We can see that this chaotic motion largely depends upon the initial point from where the pendulum starts its motion and the displacement. Thus we can say that, this simple chaotic system too, depends highly on the initial conditions.

A chaotic map is a discrete map which is generated from functions or equations that evolve over time. The most prominent and oldest one is the Logistic Map which showed that chaos can be formed from simple formulas. The map, which was developed for analysing the population rate(equilibrium population) was highly dependent on the initial growth rate(r). The diagram obtained when r was plotted equilibrium population is known as the Bifurcation diagram.

Some of the other chaotic maps include Sine map, Tent map, Piecewise linear map, etc. While sine map shows a similar chaotic behaviour as the logistic map, the tent map shows a tent-like structure in the bifurcation diagram and piecewise linear chaotic map composes of multiple linear segments [1][2].

Paper 1

In the first paper, the authors have used a Piecewise Linear Chaotic Map(PWLCM). This map is composed of multiple linear segments, and is given by the equation:

$$x_i = F(x_{i-1}, \eta) = \begin{cases} x_{i-1}/\eta, & 0 < x_{i-1} < \eta \\ (x_{i-1} - \eta)/(0.5 - \eta), & \eta \leq x_{i-1} < 0.5, \\ F(1 - x_{i-1}, \eta), & 0.5 \leq x_{i-1} < 1. \end{cases}$$

where $x_i \in (0,1)$ is the initial condition and $\eta \in (0,0.5)$ is the positive control parameter. The map contains no window in the bifurcation diagram and is chaotic in the range of η. The PWLCM map has better balance than the Logistic map.

Paper 2

In the second paper, the authors have used a nonlinear combination of two one-dimensional chaotic maps(seed maps) such that the system follows the equation:

$$X_{n+1} = \mathcal{A}_{FG} = (F(a, X_n) + G(b, X_n)) mod\ 1$$

where $F(a,X_n)$ and $G(b,X_n)$ are two seed maps with a and b as parameters and n is the iteration number. As two chaotic maps are combined, it shows mixed chaotic property. To ensure that the output is in the range of [0,1], mod is used.

The seed maps can be used interchangeably. Some of the examples are Logistic-Tent, Logistic-Sine and Tent-Sine map. In our implementation, we used Logistic-Tent map which is defined by the equation:

$$\begin{aligned} X_{n+1} &= \mathcal{A}_{\mathcal{LT}}(r, X_n) = (\mathcal{L}(r, X_n) + \mathcal{T}((4-r), X_n)) mod\ 1 \\ &= \begin{cases} (rX_n(1 - X_n) + (4-r)X_n/2) mod\ 1 & X_i < 0.5 \\ (rX_n(1 - X_n) + (4-r)(1-X_n)/2) mod\ 1 & X_i \geq 0.5 \end{cases} \end{aligned}$$

CONTRIBUTIONS

Main Contribution

The main contribution from my part to this project was the implementation of the Decryption algorithm and optimisation of the code to reduce the end-to-end time.

Paper 1

- In the encryption, the image is decomposed to bit-planes in which a grayscale image is divided into 8 binary bit-planes.
  Then the bit-planes are transformed into row vectors A1 and A2 and then applied to a Diffusion Algorithm. Then A1 is cyclically shifted to the right by sum of the elements in A2. Then each element is encrypted using the previous element in the vector(first element is encrypted using the last), the corresponding element in the second vector A2 and the key. This process is repeated for the second vector A2, until all the elements are encrypted.
  Then the encrypted row vectors B1 and B2 are passed to the Confusion Algorithm. In the confusion algorithm, the elements in B1 and B2 are swapped according to the chaotic sequences Y and Z formed from the secret keys.
- In the paper, the decryption procedure is just given as the reverse process of the encryption. This is true for the confusion phase. When the swapping was done in a reverse order properly, the confusion phase was decrypted.
- But in the diffusion phase, as each element is encrypted using the previous element in a cyclic manner(first element is encrypted by the last), just doing the reverse procedure will not result in the decrypted output.

- So, after careful observation, it was found out that the equations for each element in the encrypted row vectors formed a system of linear equations that need to be solved in-order to obtain the correct decrypted output.
- So, Gaussian elimination was applied to the equations, and with the correct key it successfully gave the correct elements for the row vectors of the plain image.
- After that, the bit-planes are combined to form the decrypted image.
- End-to-end optimisation: The problem was that for a larger image, say 1024x1024 image, the total equations that need to be solved was 41,94,304 which took a long time. But correct optimisation of creating the matrices for the gaussian elimination and solving the equations, the time was significantly reduced.

Paper 2

- The authors of this paper implemented a four-round encryption system, which includes five steps:
  The first step is to insert one pixel with a random value in the beginning of each row of the image. Then, each row is separated into individual arrays(1D matrices). After the separation, each element is encrypted using a substitution process by XORing with previous element and the random sequence keys derived using the proposed Logistic-Tent system. After the substitution, the rows are combined together and the image is rotated 90° counterclockwise. This process is repeated four times to obtain the encrypted image.
- The decryption process is exactly the opposite of the encryption process.
- But, the round keys which are used for XORing in the substitution phase of every round has to be calculated from the initial value and the Logistic-Tent system beforehand as the first substitution is with the fourth round key and next round is with the third round key and so on(The keys are reversed).

- Therefore, the round keys are calculated first. Then the rows are separated, and the substitution is done in the reverse order.
- After the first round of substitution in the decryption process(effectively the fourth round in the encryption), the rows are combined together.
- Then the image is rotated clockwise and the process is repeated four times.

Other Contributions

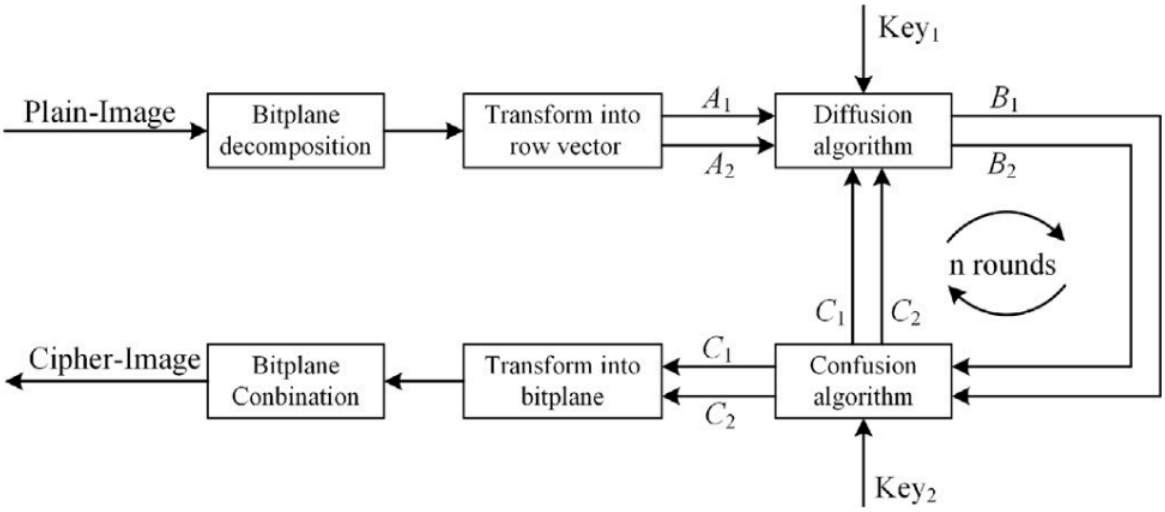
- Error correction and optimisation of the Matlab codes for various parts of the project.
- Obtaining, comparing and forming the results of various images of size and format.

System Specifications

A. Operating System: macOS Catalina 10.15.6
B. Processor: Intel Core i7
C. RAM: 16GB
D. Programmed in: Matlab R2019b

METHODOLOGY

Paper 1



Block Diagram of the Proposed Crypto-System

I. Encryption

1. The image is divided into 8 binary bit-planes and is transformed into two row vectors A1 and A2.
2. Then the two row vectors are passed onto the Diffusion Phase.
3. First the row vector A1 is cyclically shifted to the right by sum of all the elements in vector A2(sum1) to get A11.
4. Then each element in A11 is encrypted using the previous element, the corresponding element in the vector A2 and the key stream b1 to obtain encrypted vector B1 by the equation:

$$B_1(1) = A_{11}(1) \oplus A_{11}(L) \oplus A_2(1) \oplus b_1(1).$$ ——>For the first element

$$B_1(i) = A_{11}(i) \oplus A_{11}(i-1) \oplus A_2(i) \oplus b_1(i)$$ ———>For the rest of the elements

5. Similarly, A2 is shifted right according to the sum of all the elements in B1(sum2) and encrypted in the same fashion with key stream b2 to obtain B2 by the equation:

$$B_2(1) = A_{22}(1) \oplus A_{22}(L) \oplus B_1(1) \oplus b_2(1). \quad \text{——>For the first element}$$

$$B_2(i) = A_{22}(i) \oplus A_{22}(i-1) \oplus B_1(i) \oplus b_2(i) \quad \text{———>For the rest of the elements}$$

The chaotic sequences for this phase, having M x N(size of the plain image) values formed from the initial secret key, $key_1(x_0,\mu_1)$ is given by the equation:

$$X_1 = \mathrm{mod}(floor(X \times 10^{14}), 256).$$

This chaotic sequence X1 is divided into two groups equally and is transformed into binary sequences b1 and b2(the encryption keys).

6. Next the two row vectors B1 and B2 are given as input to the Confusion Phase.
7. The integer key sequences Y and Z for the confusion phase is given by the equation:

$$Y = \mathrm{mod}(floor(S_1 \times 10^{14}), L) + 1.$$

$$Z = \mathrm{mod}(floor(S_2 \times 10^{14}), L) + 1.$$

where S1 and S2 are chaotic sequences produced from the second secret key $key_2(y_0,\mu_2)$.

8. Then the binary elements in B1 and B2 are swapped according to the integer key sequences Y and Z such that B1(i) is swapped with B2(Y(i)) and B2(i) is swapped with B1(Z(i)).
9. The resultant encrypted row vectors are transformed to an M x N encrypted image.

II. Decryption

1. Decryption process is almost the reverse process of encryption.
2. First the encrypted images is converted to 8 binary bit-planes and is converted to row vectors D1 and D2.
3. Then the two vectors are passed onto the Reverse Confusion Phase where the elements in D1 and D2 are swapped back such that D1(Z(i)) is swapped with D2(i) and D2(Y(i)) is swapped with D1(i).
4. After that, the row vectors D1 and D2 are passed onto the Reversed Diffusion Phase.
5. As each element is encrypted using the previous element(the first element is encrypted using the last), each equation has two unknowns. Therefore, reversing the equation will not yield the decrypted values.
6. Therefore, Gaussian elimination is done to solve the linear system of equations.
7. First the second row vector D2 is solved. The equations are written in AX=B format and is solved for X to get the shifted version of the decrypted row vector, A22.
8. Then, the row vector is shifted to the left according to the sum of all elements in D1 to get the decrypted vector A2.
9. Similarly, row vector D1 is solved. Gauss elimination is done to get the shifted vector A11 and is shifted back to get the decrypted vector A1.
10. These row vectors are then combined and transformed back together to get the MxN plain image back.

Paper 2

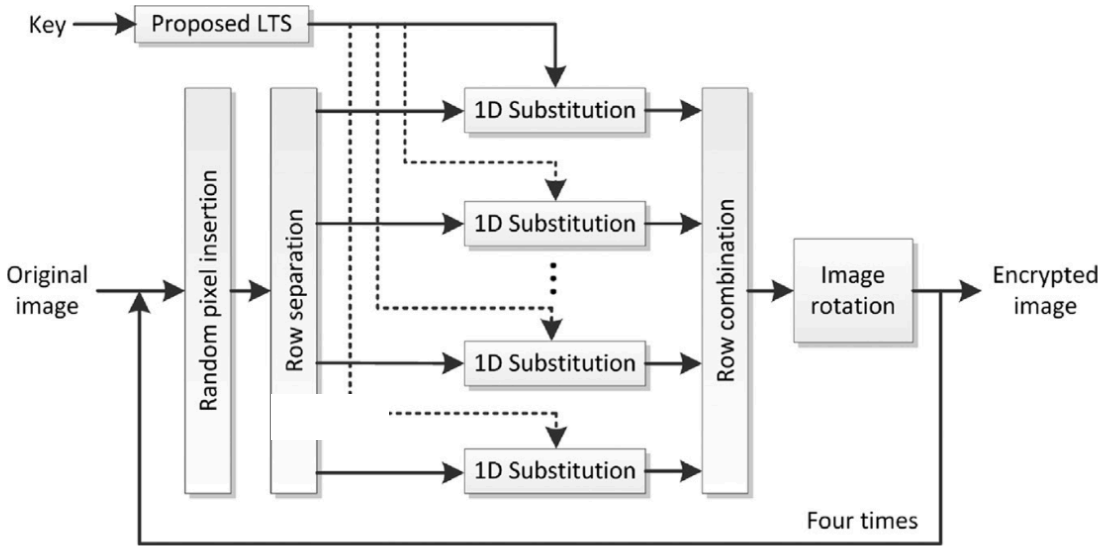
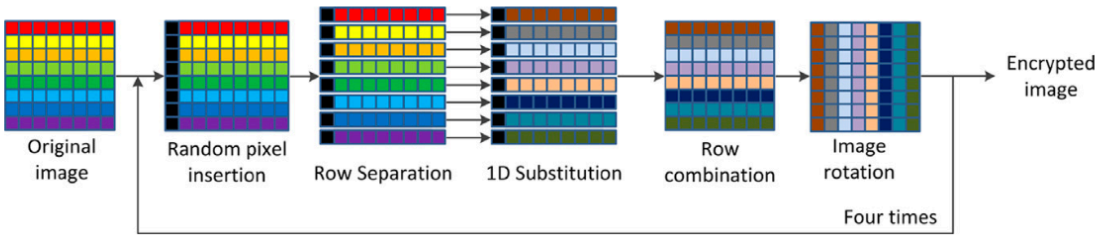


Fig. 4. The new image encryption algorithm.



Illustrative Example of the Proposed Algorithm

I. Encryption

1. The total encryption process consists of four rounds and each round contains five steps.
2. The first step of encryption is random pixel insertion. In this step, a random value is added as the first element of each row in the image. After this, the size of the image increases from M x N to M x (N+1).
3. Then each row is separated into arrays or one-dimensional matrices.
4. The next process is 1D substitution in which each values in every array is substituted by the equation:

$$B_i(j) = \begin{cases} R_i(j) & \text{if } j = 1 \\ B_i(j-1) \oplus R_i(j) \oplus (\lfloor S_k(i,j) \times 10^{10} \rfloor mod\ 256) & \text{otherwise} \end{cases}.$$

   where $S_k(i,j)$ is the random chaotic sequence for the kth round(four rounds) of encryption. This sequence is generated using the proposed Logistic-Tent System, defined by the equation:

$$S_k(i,j) = \begin{cases} S_1(0,0) & \text{for } i=0, j=0, k=1 \\ S_2(M,0) & \text{for } i=0, j=0, k=3 \\ S_{k-1}(N,0) & \text{for } i=0, j=0, k=2,4 \\ \mathcal{A}_{\mathcal{LT}}(r_0, S_k(i-1,0)) & \text{for } i>1, j=0 \\ \mathcal{A}_{\mathcal{LT}}(r_k, S_k(i,j-1)) & \text{for } i>1, j>0 \end{cases}$$

   where s_k is the initial value and r_k is the parameter in the kth round. User defined values include $S_1(0,0)$, r_0 and r_k.
5. Then, the rows are combined together and is rotated 90° counterclockwise.
6. This process is continued for four rounds to obtain the encrypted image.

II. Decryption

1. Decryption is mostly the reverse of the encryption process.
2. All the round keys are generated first, as the rounds are reversed(fourth round comes first) according to the equation:

$$S_k(i,j) = \begin{cases} S_1(0,0) & \text{for } i=0, j=0, k=1 \\ S_2(M,0) & \text{for } i=0, j=0, k=3 \\ S_{k-1}(N,0) & \text{for } i=0, j=0, k=2,4 \\ \mathcal{A}_{\mathcal{LT}}(r_0, S_k(i-1,0)) & \text{for } i>1, j=0 \\ \mathcal{A}_{\mathcal{LT}}(r_k, S_k(i,j-1)) & \text{for } i>1, j>0 \end{cases}$$

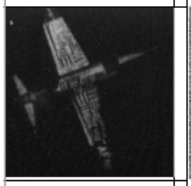
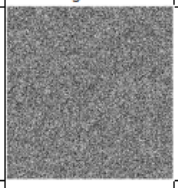
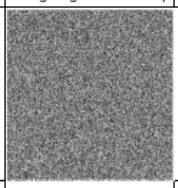
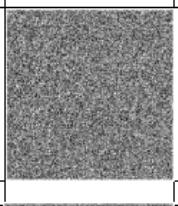
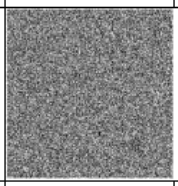
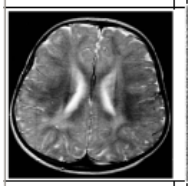
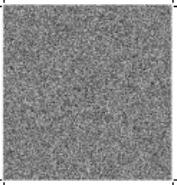
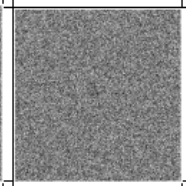
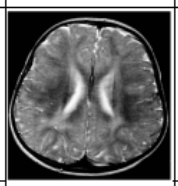
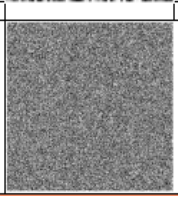
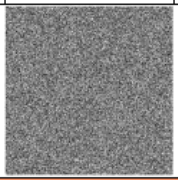
3. After generating the keys, the image is rotated 90° clockwise.
4. Then, each row of the encrypted image is separated into arrays or one-dimensional matrices.
5. After the row separation, the rows are passed onto the Inverse 1D Substitution process, given by the equation:

$$R_i(j) = B_i(j-1) \oplus B_i(j) \oplus (\lfloor S_k(i,j) \times 10^{10} \rfloor mod\ 256)$$

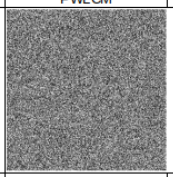
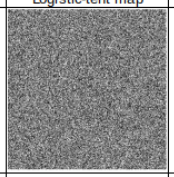
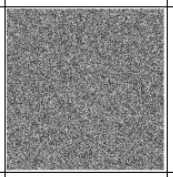
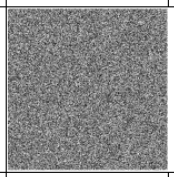
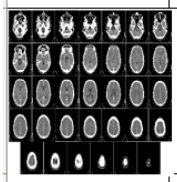
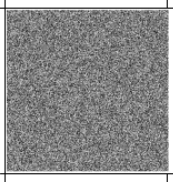
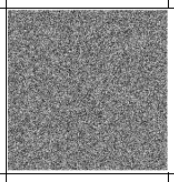
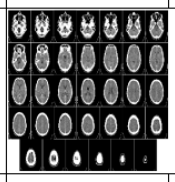
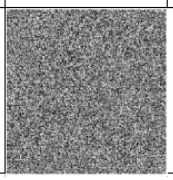
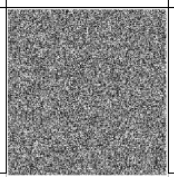
6. After the inverse substitution process, the rows are combined together to form the image matrix.
7. Then the inserted random pixel is removed from the matrix.
8. The resultant matrix is again rotated 90° clockwise and the process is repeated for four rounds.
9. After the fourth decryption round, the resultant matrix is transformed to get the decrypted image.

RESULTS

I. End-to-End Time

Original image	Encrypted Image Using PWLCM	Encrypted Image Using Logistic-tent map	Decrypted Image	Total Time taken Using PWLCM (sec)	Total Time taken Using Logistic-tent map (sec)	size
				54.181	74.87	1024 x 1024
				3.877	3.638	256 x 256
				40.255	23.411	675 x 901
				0.67	0.667	64 x 64
				30.758	23.538	512 x 512

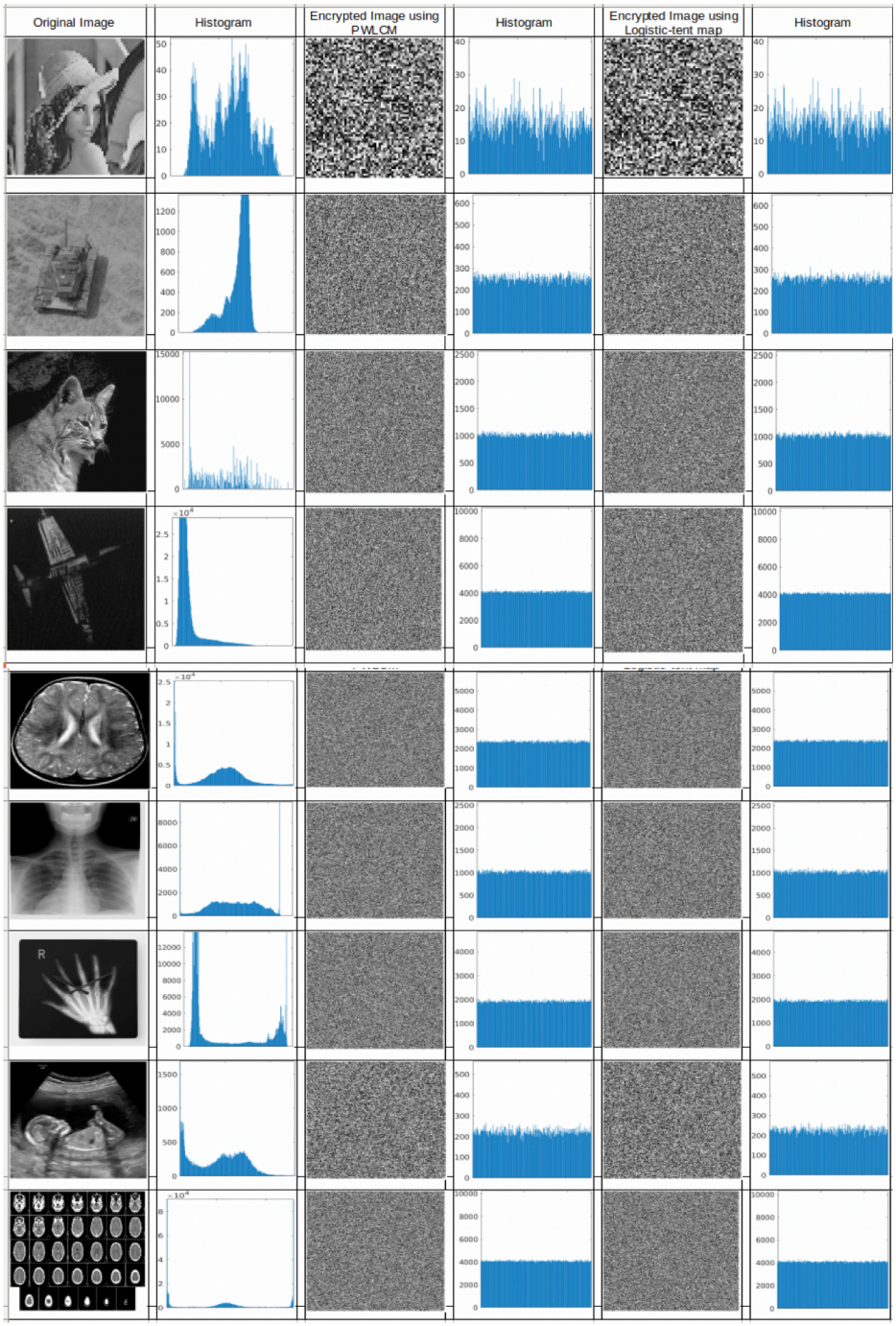
Encrypted and Decrypted Images Along With Total Time Taken

Original Image	Encrypted Image using PWLCM	Encrypted Image using Logistic-tent map	Decrypted Image	Time Taken for PWLCM (sec)	Time Taken for Logistic-tent map (sec)	Size
				22.884	9.868	512 x 512
				63.272	23.023	704 x 704
				118.268	57.591	1024 x 1024
				5.24	2.606	234 x 246

Encrypted and Decrypted Images Along With Total Time Taken

From the above images, it is clear that the implemented system encrypts and decrypts images of different sizes and format. Also, as the code implemented is optimised thoroughly, the end-to-end time is very less, even for larger images. The end-to-end time depends upon the image resolution and the image size and varies according to that.

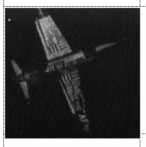
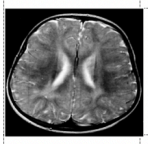
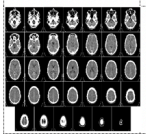
II. Histogram Analysis



Histogram Comparison of the Original and Encrypted Images

Histogram is the distribution of pixel intensities in an image. As we can see from the Histogram analysis, the encrypted images have a uniform histogram when compared to the plain images. This way, the implemented system can resist statistical attacks.

III. Entropy and Contrast Analysis

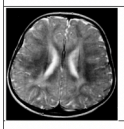
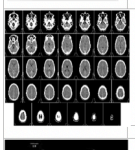
Original Image	Entropy of cipher image Using PWLCM	Entropy of cipher image Using Logistic-tent map	Contrast Value of Original image	Contrast in cipher Image using PWLCM	Constrast in cipher image Using logistic-tent map
	7.91102	7.91107	1.35860	10.42560	10.73260
	7.94777	7.94775	0.13110	10.42870	10.43470
	7.05045	7.94963	0.20780	10.49700	10.51900
	7.94823	7.94822	0.14210	10.49930	10.51470
	7.95489	7.95432	0.12520	10.52050	10.49570
	7.94994	7.94928	0.06070	10.51410	10.51130
	7.94883	7.94853	0.09930	10.49330	10.55600
	7.94906	7.94899	0.18600	10.58960	10.51790
	7.94943	7.94833	1.40430	10.48050	10.49980

Entropy of the Encrypted Images and Contrast Comparison

When all of the pixels of an image is equally distributed, the entropy of an 8-bit grayscale image will be 8. This makes the information random and thus resist attacks. As we can see from the entropy analysis, all the values of the encrypted image is close to 8. Contrast analysis can be used to analyse the distribution of pixels and its local variation of an image. In the above analysis we can see that the contrast of the cipher images is around 10, which is the expected value for an 8-bit grayscale image.

SECURITY ANALYSIS

I. Differential Attack and Key Sensitivity Analysis

Original image	size	Key Sensitivity (Encryption) For PWLCM (% Matched)	Key Sensitivity (Decryption) For PWLCM (% Matched)	Differential Attack For PWLCM (NPCR & UACI)	Key Sensitivity (Encryption) For Logistic - tent map (% Matched)	Key Sensitivity (Decryption) For Logistic - tent map (% Matched)	Differential Attack For Logistic - tent map (NPCR & UACI)
	1024 x 1024	0.3892	0.3912	NPCR : 99.6012 UACI : 33.4935	0.3986	0.3871	NPCR : 99.6045 UACI :33.4585
	256 x 256	0.3769	0.4044	NPCR : 99.6399 UACI : 33.585	0.4101	0.3738	NPCR : 99.6199 UACI :33.5581
	675 x 901	0.3861	0.3782	NPCR : 99.6152 UACI : 33.4254	0.4057	0.4017	NPCR : 99.6051 UACI :33.4719
	64 x 64	0.41	0.3418	NPCR : 99.4466 UACI : 33.3499	0.3593	0.4639	NPCR : 99.6786 UACI :34.3064
	512 x 512	0.3914	0.3868	NPCR : 99.6147 UACI : 33.4063	0.377	0.3882	NPCR : 99.6185 UACI : 33.4361
	512 x 512	0.3685	0.3838	NPCR : 99.6075 UACI : 33.4333	0.3861	0.4082	NPCR : 99.6071 UACI : 33.5447
	704 x 704	0.3873	0.3817	NPCR : 99.6063 UACI : 33.4919	0.3965	0.3959	NPCR : 99.6087 UACI : 33.4508
	1024 x 1024	0.3855	0.3931	NPCR : 99.6122 UACI : 33.4629	0.3873	0.3871	NPCR : 99.6169 UACI : 33.4716
	234 x 246	0.3804	0.4082	NPCR : 99.6514 UACI : 33.4944	0.3906	0.4091	NPCR : 99.6088 UACI :33.4392

Key Sensitivity Analysis and Differential Attack

Key Sensitivity

- The key sensitivity test is performed in both encryption side and decryption side.
- In the encryption side, it is performed to check whether a slight change in the encryption key would result in a completely different encrypted image.
- In the decryption side, it is performed to check whether making a slight change to the decryption key would produce a completely different output image(not able to decrypt properly without the correct key).
- As we can see from the analysis above, when the encrypted and decrypted images are compared(before and after changing the key), the percentage matched is less than 0.4% of the entire image.
- Also when changing the decryption key and trying to decrypt, the resultant image is completely different from the plain image and is not recoverable.
- This shows that the implemented system has perfect key sensitivity.

Differential Attack

- Any slight change to the plain image should result in a completely different cipher image, when encrypted with the same key. This property of an image encryption is very important to resist differential attacks.
- For the performance analysis of the implemented system against differential attacks, two parameters are defined and calculated. They are Number of Pixel Change Rate(NPCR) and Unified Average Changing Intensity(UACI).
- As we can see from the analysis above, the NPCR of the implemented system is in the range of 99% and UACI is in the range of 33%. This is comparable to the result in the paper referred.
- Thus, the implemented system can resist differential attack effectively.

CONCLUSION

Image encryption is one of the most important areas in cryptography that proper attention should be given to. It is because, the conventional encryption algorithms like Data Encryption Standard(DES) and Advanced Encryption Standard(AES) have been proven not effective in image encryption because of the size of the data and the pixel redundancy. Therefore, many types of image encryption systems have been proposed by researchers throughout the years. The more prominent among those are the Chaotic Encryption Schemes that proved to be excellent in encrypting images because of its various intrinsic properties like ergodicity and sensitivity to initial conditions.

In this project, we implemented two of the chaotic image encryption schemes which uses different variations of chaotic maps and/or combinations of them. We also did various security and statistical analyses to assess the performance of our implementation against various attacks such as differential attacks. Also, we performed end-to-end time analysis and histogram analysis to compare the results of our implementation to the results in the reference papers.

REFERENCES

[1] Xu, L., Li, Z., Li, J., & Hua, W. (2016). A novel bit-level image encryption algorithm based on chaotic maps. Optics and Lasers in Engineering, 78, 17-25.

[2] Zhou, Y., Bao, L., & Chen, C. P. (2014). A new 1D chaotic system for image encryption. *Signal processing*, *97*, 172-182.

[3] "The Definitive Glossary of Higher Mathematical Jargon — Chaos". Math Vault. 2019-08-01.

[4] Hua, Z., Jin, F., Xu, B., & Huang, H. (2018). 2D Logistic-Sine-coupling map for image encryption. Signal Processing, 149, 148-161.

[5] C. Li, D. Lin, B. Feng, J. Lü and F. Hao, "Cryptanalysis of a Chaotic Image Encryption Algorithm Based on Information Entropy," in *IEEE Access*, vol. 6, pp. 75834-75842, 2018.