Encryption Algorithm using Rubik's Cube Principle for Secure Transmission of Multimedia Files

[1]Parita Oza, [2]Vishakha Kathrecha, [3]Pooja Malvi

Computer Science and Engineering Department, Nirma University, Ahmedabad, Gujarat

***Abstract*- Multimedia has the equal threats against security as texts or raw data. In the field of information security, image encryption plays an important role. There are many image encryption algorithms available but most of them have performance and security issues. In this paper, we have analyzed six different most used image encryption algorithms and proposed our new image encryption algorithm.**

***Keywords*- Image encryption, chaotic system, transformation table, performance analysis, Rubik's cube principle**

I. INTRODUCTION

Images are different from text data in many properties, such as high redundancy and correlation, local structure and characteristics of amplitude frequency. As a result, the methods of conventional encryption used for text data perhaps cannot be applicable to images. Each pixel of the image is represented as value between 0 to 255. And thus, there can be three ways to encrypt an image using values of pixel or number of pixels in the image. Pixel Transformation, Value Transformation and Chaotic Transformation.

The paper consists of various analysis parameters, overview of six most used image encryption algorithms and their analysis and at last, our proposed algorithm.

Analysis Parameters

1) *Number of pixel change rate (NPCR)*

It is used to check the effect of one pixel change on the entire image. NPCR indicates the percentage of different pixels between two images. Let (i,j) be the pixel value of the image

$$NPCR = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} D(i,j) \times 100\%$$

then,

Where,

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

Histogram Analysis

This analysis reveals the distribution of pixel values within an image. Histogram reflects image statistical distribution, and usually is used for statistics analysis attack.

2) *Unified Average Changing Intensity (UACI)*

This measure is helpful to identify the average intensity of difference in pixels between the two images because a small change in plaintext image must cause some significant change in cipher image. The formula of the same is given by,

$$UACI = \frac{1}{m \times n \times 255} \sum_{i=1}^{m} \sum_{j=1}^{n} \left| C_1(i,j) - \right| C_2(i,j) \times 100\%$$

3) *Entropy*

Entropy gives an idea about self information of image data. It is calculated by the following equation,

$$He = -\sum_{K=0}^{G-1} P(K) \log 2(\mathbf{P(K)}).$$

Where:

He: entropy.
G: gray value of input image (0... 255).
P(k): is the probability of the occurrence of symbol *k*.

The other parameters includes correlation coefficient and covariance.

II. OVERVIEW

Secure Image Encryption Algorithm based on Rubik's Cube Principle [1].

Chaos based calculation are promptly utilized as a part of scrambling, substitution, transposition and stage of the picture which changes the pixel values. This system uses Rubik's shape standard and another disarray based quick picture encryption calculation as far as the parameters like NPCR, UACI, Entropy and Correlation coefficient [1]. A new –chaos based fast image encryption algorithm was proposed in this paper, a 128-bits key is used for the algorithm [1].

i. Image is partitioned into blocks of 8x8 pixels.
ii. Pseudorandom numbers are generated from NCML
iii. Pixel values in each block are changed and permutation is performed simultaneously
iv. The operations are repeated till the number of rounds are reached.

Performance Analysis: The test image is of size 256×256. The outcomes in the wake of scrambling the picture utilizing two cycles are given. The diffusion characteristics including NPCR, UACI, correlation coefficients and entropy are used for the evaluation. A key space investigation is additionally included toward the end of the evaluation.

Table 1 [1]

Algorithm	NPCR		UACI	
	Lena	Baboon	Lena	Baboon
Scheme in [8]	99.641	99.609	28.620	27.409
Scheme in [9]	99.607	99.606	33.463	33.470

Connection coefficient of the first picture is typically high (near one). Weaker the connection coefficient of the encoded picture better the calculation.

Table 2 [1]

Correlation coefficient Correlation	Horizontal	Vertical	Diagonal
Lena (original)	0.9864	0.9886	0.9776
Scheme in [8]			
Lena(encrypted)	0.0068	0.0091	0.0063
Scheme in [9]			
Lena(encrypted)	0.0007	0.0021	0.0148

Key analysis: One of the parameters used to quantify the security of a picture encryption calculation is its key space. As the key space builds the security of encryption progresses. The strategy in 1 depicts an encryption system that uses a 8-bit grey scale picture of size 256 × 256 pixels and emphasis count = 1, in which the key space is dictated by the blend of picture size and the quantity of cycles. This key space is sufficiently vast to oppose thorough assaults.

In this paper we examined two novel chaos based picture encryption calculations. The system utilizing Rubik's cube Principle has an expansive key space and its execution is entirely basic. The new quick chaos based picture encryption calculation joins the stage and dissemination for quick handling and uses NCML which lessens the issue of periodicity in the era of pseudorandom groupings.

An Inter-Component Pixels Permutation Based Color Image Encryption Using Hyper-haos [5]

This strategy uses high-dimensional1 chaotic map

1) 3D-Arnold transformation[5]
2) 2D-Hyper Chaotic Map[5]

The whole encryption process consists of following steps of operations.

Step 1. Evaluate the sum of all the gray-values of color plain-image P of size M×N×3, let it be gvSum.[5]
Step 2. Extract the control parameters of Arnold transform from gvSum as[5]:

a = 13 + (gvSum)mod(97)
b = 23 + (gvSum)mod(59)
c = 17 + (gvSum)mod(79)
d = 37 + (gvSum)mod(43)
n = 07 + (gvSum)mod(31)

Step 3. Shuffle the plain-image P using 3D Arnold transform with parameters a, b, c, d and n.
Step 4. Let the shuffled colour image obtained is S. Reshape S into 1D sequence
Step 5. Iterate the hyper-chaotic system given in Equation (2), with parameters x0, y0, a1, a2, a3 and a4, for M×N×3 times to get X and Y sequence. For each iteration i, we get values xi and yi
Step 6. Pre-process the two sequences using Equation (3)-(4) to improve their statistical properties.
Step 7. The pre-processed xi and yi are multiplied by 1014, extract their integer parts, then apply modulo-256 operation to get values in the range of 0 – 255 corresponding to each xi and yi. Let the converted sequence be K1 and K2 having size of M×N×3. Choose C(0) in the range of 0 – 255.
Step 8. Applying XOR operation in following way using CBC mode to encrypt all pixels

.

C(i) = S(i) Å K1(i) Å K2(i) Å C(i-1)

Where i = 1, 2, . . ., M×N×3, Reshape 1D sequence C(i) to a color encrypted image. The decryption progresses similar to the encryption one, described above, but in the reverse order.

Analysis

- Control parameters (keys) extracted from plain text for Arnold transform (Establishing dependencies).
- Confusion is done by CBC, which makes encrypted text dependent.
- Chaotic system are sensitive for initial conditions and immune to crypto attack.

It uses pseudorandom property and non-periodicity

A New Image Encryption Scheme Based on DES Algorithm and Chua's Circuit [4]

Chaotic encryption is embedded into DES algorithm's iteration [4], the chaotic sequence generated by Chua's circuit is used as beginning key. Through the XOR operation between chaotic sequences sub-key and plaintext, the new calculation can change the consistency of plaintext move in 16-round structure successfully, therefore it amplifies the key space and upgrades hostile to assault character for beast power assault and measurable assault.
Firstly, a digital image is chosen as encrypt object. The detail encryption process is as follows:

1) *Data Block:* Group the obtain image data into 64 bit plaintext blocks. Through initial permutation, a block

of the 64 bits permuted data is divided into a 32 bit left sub-block (L 0) and a 32 bit right sub-block (R 0).

2) *Key Generation:* By setting Chua's circuit's initial value and generating chaotic sequences, take 64 bit data as initial key and another 48 bit data denote as CHUA. Initial key is permutated to 56 bit data, then the 56 bit permutated data is divided into 28 bit on the left (C) and 28 bit on the right (D), C and D are put into 16-round left shift, at the same time, the result after every shift is transposed under choice-2 permutation, then the 48 bit sub-key K *i* are generated.
3) *16-round Structure*: In the 16-round structure, L *i* and R *i* are the input of the 16-round iteration. First, 32 bit R *i* is extended to 48 bit data when it is substituted into F-function, then 48 bit R *i* 、 K*i* and data CHUA are XORed together, after s-box and P permutation, the output of F-function is obtained. At last, according to the follow iterative formula, the next input will be calculated.
4) *Left and Right Exchange*.: When 16-round iteration is over, exchange L15 and R15
5) *Output Result:* The output of the left and right exchange is transposed under inverse initial permutation, then the last result is ciphered image.

It uses less key space for DES. Initial key generated is done by Chua circuit and thus is immune to brute force attack.

Image Encryption Algorithm Based on a new combined chaotic system: A new combined chaotic system is introduced in this paper. The algorithm is a nonlinear combination of several traditional chaotic maps like Logistic map and sine map. The relations between the Chaotic systems are controlled by the encryption keys. The procedure for encryption is as follows:

Step1. Set the values of the encryption key K, which has four components, the iteration number N, parameter k_0 and a and the format coefficient F . Then set i = 1.

Step2. Applying Equations

$$Sum = \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} + K_0$$

,

$$X_{i0} = 0.001 \times i \times \mathrm{mod}(Sum, 99)$$

to generate the Chaotic sequence

$$X_i = \{X_{i1}, X_{i2}, \cdots X_{ij}\}, j = m \times n - 1 .$$

where $X_{t+1}=r_t * X_t (1-X_t)$

$r_t=a*\sin(pi*r_{t-1})$

a belongs to R and t=1,2,3...

Step3. *Step3.* Using equations $S_i' = \psi(S_i)$, $P = \psi(P)$ (confusion property) to perform the permutation process or Equations $\tilde{S} = \mathrm{mod}(S \cdot 10^{32}, F)$ (substitution process), $C = \mathrm{mod}(\tilde{S} + P, F)$ (diffusion process) to perform the substitution process.

Where the format coefficient *F* is determined by the format of given plaintext image. For gray images, *F* = 256,
P are the pixel values before substitution. *C* is the encrypted pixel value.

Step4. If i < N, i = i +1 and go back to step 2. Or go to step 5.

Step5. End.

The control parameter a has a much larger chaotic interval compared with the Logistic map and Sine map so the new combined chaotic map has a high level of security.

The decryption process is a simple reverse process using the same security key.

Experimental Analysis: Histogram Analysis-
The experimental results of the histogram analysis for the image encryption using proposed algorithm almost reach to a balanced point as the pixel values distribute uniformly.

Differential Attacks: NPCR and UACI tests: The tested image is Elaine image, and get the *NPCR* = 99.63%, *UACI* = 33.50% are close to the theoretical values from equations. Algorithm has shown its better random-like property and high sensitivity to its initial values and parameters. Moreover, the control parameter has a larger chaotic interval compare to traditional chaotic maps. This ensures the proposed chaotic system more suitable for various cryptography applications.

Image Encryption Algorithm by using Block Based Symmetric Transformation Algorithm (Hyper Image Encryption Algorithm)[2]

This is simple and direct mapping algorithm using Feistal structure, some logical operation and transformation table (new permutation technique) for encrypting the image.

Steps for Encryption Algorithm:

1. Select an Image which is having at least 256 bits in Size to be encryption.
2. Calculate Binary Value of Image.
3. Select First 256 bits form Binary Value and create 16 sub blocks of 16 bits. This process will repeat till end of file.
4. Select Key Value of 256 bits. And create 16 sub blocks of 16 bits.
5. Select 64 bits from transformation table. And create 4 blocks of 16 bits.
6. Apply logical operation XOR between first 8 block of selected image and second 8 block of selected key. Result will stored in image blocks

7. Apply logical operation XOR between last 4 blocks of selected images and 4 blocks of transformation table. Result will store in image blocks.
8. Apply circular shift operation on last 4 block of selected key and second last 4 block of selected image.
9. Apply logical XOR operation between selected image and key which is output of step 8. Result will store in image block.
10. Apply circular shift operation on 4 blocks of transformation table and second last 4 block of selected key.
11. Apply logical XOR operation between transformation table and selected key, which is output of step 10. Result will store in key block.
12. Combine output of step 6, 7, 9, and 11 in such that it should be produced 256 bits total.
13. output of step 12 will become input for next round.
14. Repeat step-1 to step-13, 10 times.
15. After 10th round, cipher text will produce of selected image.
16. Exit.

Algorithm for Creating Transformation Table

1. Select Image to be encryption from data store
2. Insert key of 256 bits
3. Calculate Image Pixels Value Horizontal Value of Pixel = pixelwidth/10, Vertical Value of Pixel = pixelheight/10
4. Select a Random Function to Calculate Final value for Horizontal and Vertical Pixels. Horizontalpixel Select Random Value between Horizontal Value of Pixel and pixelwidth, verticalpixel Select Random Value between Vertical Value of Pixel and pixelheight
5. Select a Variable No-Of-Pixel to store Multiple Value of horizontalpixel and verticalpixel
   No-Of-Pixel = horizontalpixel X verticalpixel
6. Using Hash Function (Here I am using SHA-1) to generate a Seed Value. This SHA-1 will apply on 256 bits
   Selected Key
   Seed = SHA-1(Above Selected KEY)
7. Divide Seed into two Part equally Seed-1 and Seed-2
   Seed-1 First Half of Seed
   Seed-2 Second Half of Seed
8. If Seed-1 is Greater than Seed-2 Then we will Select another Variable seedvalue and assign any numeric value between 0 to 4 (Randomly Chooseable) Otherwise Value of seedvalue Variable vary between 5 to 9 (Randomly chooseable).
9. If Variable seedvalue is Equal Between 0 to 4 then calculate new seed value (Here we are working on ASCII
   Value of seed).
   Seed = Seed + (Seed-1 Mod 2) + 1
   Otherwise
   Seed = Seed + (Seed-2 Mod 2) + 1
10. Repeat Process 8 to 9 till No-Of-Pixel/2
11. Final Output of Step 10 will represent Create transformation Table
12. Exit

Experimental Analysis: Encryption and Decryption Execution time, Entropy and CPU-Memory utilization- The experiment done on a same image and in same platform and the proposed algorithm is compared with Image Encryption Using Block-Based Transformation Algorithm and Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption and the proposed algorithm had the lowest execution time, highest entropy (70% better).

Improving for Chaotic Image Encryption Algorithm Based on Logistic Map

This algorithm gives a new position permutation method of image pixels. Firstly, a secret-key is used to generate nine real chaos sequences by Logistic map. Among them, six are used to scramble image pixel positions and three are used to confuse and diffuse image pixels.

Experimental Analysis:

Secret key Sensitivity-The decryption result is significantly different when secret-key has any tiny change.

Histogram Analysis- The histogram of the encrypted image is distributed evenly, it hides the statistical distribution of the original image and increases the difficulty of decryption.

Correlation- In order to reject the statistical attack, the correlation between two adjacent pixels must be decreased. It can be seen from the result that adjacent pixels' correlation of R-layer of the original image is very strong; and adjacent pixels of encrypted image are completely distributed to the entire image.

III. PROPOSED ALGORITHM

We introduced new algorithm based on the analysis and the parameters used in those algorithm. The below mentioned algorithm uses Rubik's cube principle to form a random matrix which is used to encrypt the image based on its pixel value. The purpose of encrypting the image is for the secure transmission of multimedia files across various end points. Here, taking an image an encrypting that image with a reference image satisfies the criteria of secure passing of multimedia files in the network.

The parameters used in this algorithm are described in detail in the following section with emphasising the properties of pixel change. The three methods described in Section 1 gives an overview of how the image can be encrypted or decrypted using its various attributes. We use pixel transformation using basic Bitwise functions and permutation technique. One more

important parameter to form a matrix of random values using sample image is using Rubik's cube Principle.

The Rubik's cube principle [1] is mentioned below:-

1) Scramble the pixels of image using Rubik's cube principle.
2) Generate two random keys.
3) Bitwise XOR is applied into odd rows and columns
4) Bitwise XOR is applied to even rows and columns using flipped secret keys.

We are taking an image of pixel size equal to MxN with each pixel of size 8 bytes and a sample image of same size to overlap it. The key size is taken same as the pixel size of the image which may change according to image resolution. Thus, it would reduce the chances of crypto attack.

The steps followed to encrypt the image are as follows:

Step 1: Pixels of image are partitioned into fixed length blocks of size M*N (image resolution) bytes. And will be represented in the form of matrix Mp.
Step 2: Find M_p^T
Step 3: For Key matrix- In the matrix Mrand, values are chosen from Rubik's cube principle.
Step 4: Key matrix Ke=mod 2 (Mrand)
Step 5: $C_{pk}=Ke+Mp^T$
Step 6: Obtain Matrix Chr by shifting row elements circularly, one element in first row, two elements in second row, three elements in third row and no elements in the last row.
Step 7: Obtain matrix Cvr from Chr by shifting column elements circularly, one element in first column, two elements in second column, three elements in third column and no elements in the last column.
Step 8: Find $Cvr^T=C_e$.
Step 9: Replace 256 to 1 by subtracting 255. (Assumption taken here.)

Brute force Analysis: As we have used Rubik's principle for generating the random matrix of values of the pixels, it is immune to brute force attack. $1920x1080x256x256=1.3x10^{12}$ total matrixes. (Supposing image resolution is 1920x1080)

IV. CONCLUSION AND FUTURE WORK

Every image encryption algorithms have their own issues. We have mentioned each algorithm's overview and their analysis. As they have not been measured in the same platform, we could not have their comparison but could find how they behave and what are their advantages and disadvantages. The proposed algorithm is yet to be implemented on the simulator platform but we have tried to make it more secure against Brute force attack and complex to crack with more throughput. We will be having implementation and then exact analysis and comparison of its behaviour.

REFERENCES

[1]. Lini Abraham, Neenu Daniel. Secure Image Encryption Algorithms: A Review. International Journal Of Scientific & Technology Research Volume 2, Issue 4, April 2013.
[2]. Hiral Rathod, Mahendra Singh Sisodia, Sanjay Kumar Sharma. Design and Implementation of Image Encryption Algorithm by using Block Based Symmetric Transformation Algorithm (Hyper Image Encryption Algorithm): International Journal of Computer Technology and Electronics Engineering (IJCTEE), Volume 1, Issue 3
[3]. Guiliang Zhu, Weiping Wang, Xiaoqiang Zhang 2, Mengmeng Wang. Digital Image Encryption Algorithm Based on Pixels. Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference
[4]. Qian Gong-bin, Jiang Qing-feng , Qiu Shui-sheng. A New Image Encryption Scheme Based on DES Algorithm and Chua's Circuit, IST 2009 - International Workshop on Imaging Systems and Techniques, Shenzhen, China, May 11-12, 2009.
[5]. Musheer Ahmad and Hamed D Al-Sharari, An Inter-Component Pixels Permutation Based Color Image Encryption Using Hyper-chaos. European Journal of Scientific Research, Vol. 116, No. 1, pp. 115-121, 2013
[6]. Sunita Bhati, Anita Bhati, S. K. Sharma, A New Approach towards Encryption Schemes: Byte – Rotation Encryption Algorithm, Proceedings of the World Congress on Engineering and Computer Science 2012 Vol II WCECS 2012, October 24-26, 2012, San Francisco, USA.
[7]. Lini Abraham, Neenu Daniel, Secure Image Encryption Algorithms: A review, International Journal of Science and Technology Research volume 2 , Issue 4, April 2013