Binary Key Based Permutation For Medical Image Encryption

Sachin Kailas Bhopi
Department of Information Technology
Ramrao Adik Institute of Technology
Nerul, Navi Mumbai, India
sach.703@gmail.com

Nilima M. Dongre
Department of Information Technology
Ramrao Adik Institute of Technology
Nerul, Navi Mumbai, India
nilimarj@gmail.com

Reshma R. Gulwani
Department of Information Technology
Ramrao Adik Institute of Technology
Nerul, Navi Mumbai, India
reshmagulwani@gmail.com

***Abstract*—The patient information such as x-rays, CTs and MRIs needs to protect from unauthorized access as per the HIPPA act. To provide better security for medical image novel encryption algorithm is proposed. In first phase, row wise, column wise and diagonal wise rotation is performed based on binary key provided. In second phase, four chaotic logistic maps are used to generate pseudorandom numbers which can work as a key to perform pixel value permutation to encrypt the image. Experimental analysis shows proposed technique does not provide any statistical information through histogram analysis. The correlation coefficient close to zero shows encrypted image is uncorrelated and NPCR and UACI values satisfy security requirement.**

Keywords— chaotic map; image security; pixel permutation.

I. INTRODUCTION

Nowadays, People often take second opinion from other doctors who suggest them same kind of tests which include x-rays, CTs, and MRI and so on which generate redundant data. This will waste lots of resources as well as time and money of the patient. To overcome this problem, many experts suggest using storage infrastructure where every doctor can store and retrieve the data of his own patient and the other patients as well. To provide security to such important data is necessary to protect it from unauthorized access.

Image encryption can be seen as a way of hiding information intelligently [1]. Text data is significantly different than that of image data as image data has some special properties: first, image file size is much larger than text file. Second, adjacent pixels of the image mostly contain redundant data and they are highly correlated therefore there is a special requirements on any encryption technique [2].

The text encryption algorithms like AES, DES etc. are insufficient for image encryption [3]. Chaotic maps are more suitable for image encryption because of its properties like systems depend completely on initial condition; it has lower mathematical complexity and provides better security [4].

The image encryption using chaotic maps can have applications in the areas like Tele-medicine, military, government documents etc. Towards this direction, we design an efficient chaos based symmetric cryptography system for medical image encryption.

The remainder of this paper is organized as follows. Section II presents Literature survey. Section III reports methodology in which position permutation and value permutation is used to encrypt the image. Section IV reports Experimental Result to check security level of technique and Finally, Section V concludes the topic.

II. LITERATURE SURVEY

Literature involves techniques for providing image security through block permutation, pixel permutation and use of chaos function along with their merits and demerits.

A. Block-based Transformation Algorithm

This algorithm consists of image conversion as well as Blowfish algorithm. The image is decomposed into the sub blocks. The sub blocks can be reposition to get possible scrambling then the image is again encrypted using blowfish algorithm to provide resultant encrypted image. The correlation between adjacent pixels is reduced. Drawback of this technique is too much memory utilization for execution of image encryption and its correlation and entropy value can be reduce further.[5]

B. Permutation Technique followed by Encryption

In this technique Rijndael algorithm as well as image permutation is used. In this original image is converted into sub blocks of size 4*4 pixels. Each block of size 4*4 is then reposition to random location and then using Advance Encryption Standard algorithm it is further encrypted. The correlation between image elements is significantly reduced and higher entropy is achieved but permutation process is very complicated and also time taking process [6].

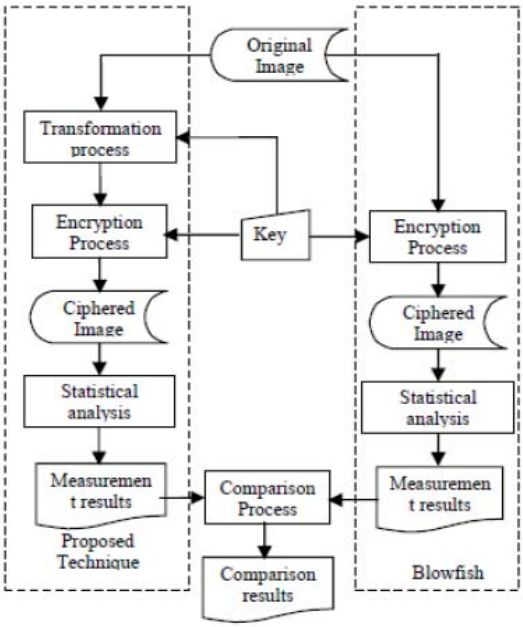

Fig. 1. Permutation followed by encryption[7]

C. Chaos and DES

This technique composed of chaotic algorithm and Data Encryption Standard algorithms for image encryption. In this, logistic map generates random numbers which applied to the color channel of the image to generate cipher image and then again encrypt using Data Encryption Standard. Their result show high starting value sensitivity, and high security but because of the characteristics of image information, DES algorithm is not the ideal choice for digital image encryption.[7]

D. Chaotic maps and DNA addition operation

This technique compares chaotic maps and identifies effects of noise on image. Chaotic maps include Henon, Logistic, Ikeda and Cross maps. In this original image is encrypted using chaotic map and then apply noise on encrypted image which means try to decrypt using random key which will give desired encrypted image. It is sensitive to the secret keys, it has larger key space but the quality of image degrades due to the effect of noise but not to an extend that image cannot be recognized.[8]

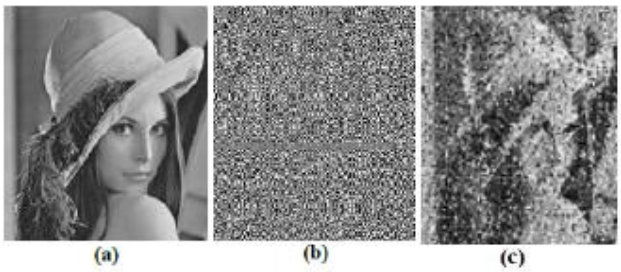

Fig. 2. Chaotic Maps and DNA addition: (a) Original Image, (b) Encryption Image, (c) Decrypted Image

III. METHODOLOGY

In proposed method, there are two phase. In first phase Fig. 3, Position permutation is perform on input image by dividing image into blocks and applying row, column and cross wise rotation on every block. In second phase Fig. 4, Value permutation is applied on resultant image of the position permutation process. In this, Image is read as a 1D array and each byte is extract to apply value permutation algorithm. The resultant output image is final encrypted image.

A. Position Permutation Algorithm

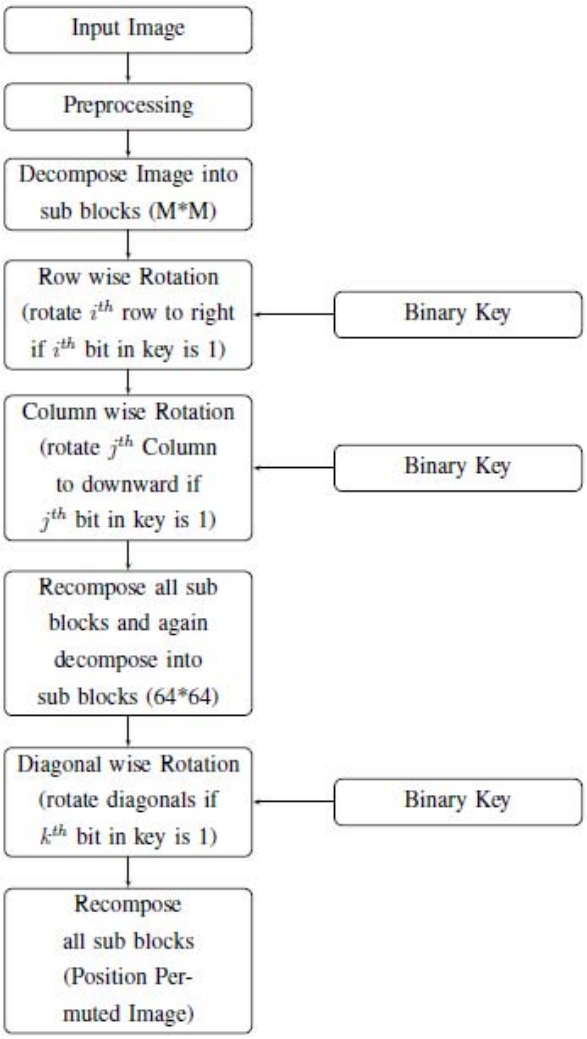

Fig. 3. Position Permutation

Step 1: Input image is decomposed into *M*M* pixels sub blocks (*M* can be 8, 16, 32, 64) to perform further operation on each block separately.

Step 2: Each sub block is rotated using predefined binary key. Rotations are row wise and column wise.

– Horizontal rotation: If i^{th} bit in binary key is 1, i^{th} row is rotated horizontally else first column rotated vertically.

– Vertical rotation: If j^{th} bit in binary key is 1, j^{th} column is rotated vertically else first row rotated horizontally.

Step 3: Recompose all the sub blocks which is again decompose into little bigger blocks N*N pixels sub blocks (*N* can be 64) to perform diagonal wise rotation.

Step 4: Cross wise rotation: If k^{th} bit in binary key is 1, sub-blocks are rotated corner to corner.

Step 5: Recompose all the sub blocks to get resultant image with position permutation algorithm.

B. Value Permutation Algorithm

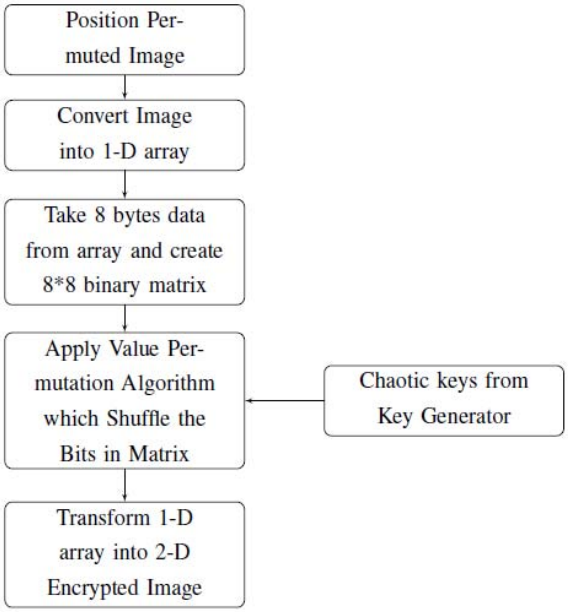

Fig. 4. Value Permutation

Step 1: Take position permuted image and convert it into 1-D array.

Step 2: Take eight 8-bit values from an array and create binary matrix of size 8*8.

Step 3: Let M denotes binary matrix of size 8*8 and *M0* denotes encryption result of *M*.

– $RotateX_p^{r\,i}$ *(M): M→M'* is define as rotate all bits in the i^{th} row of *M*, $0 \le i \le 7$, *r* bits in the left direction if p=1 or *r* bits in the right direction if p=0.

– $RotateY_q^{s\,j}$ *(M)*: *M→M'* is define as rotate all bits in the j^{th} column of *M*, $0 \le j \le 7$, *s* bits in the up direction if q=1 or *s* bits in the down direction if q=0.

Step 4: Perform encryption using keys generated from chaos key generator.

Step 5: Convert encrypted array into image.

Step 6: Decrypt image using same and exactly opposite procedure as that of encryption using same keys.

C. Chaotic Logistic Map

The logistic map is very simple non-linear dynamical equation used to generate complex and chaotic behavior. Chaotic logistic map is written as follows:

$$x_{n+1} = rx_n(1-x_n) \qquad (1)$$

Where, x_n is a number between 0 and 1 that represents the ratio of existing population to the maximum possible population. The values of interest for the parameter r are those in the interval [0, 4].

D. Key Generator

Choose *m* maps M_0, M_1, M_{m-1} from the map bank and set the order of the chosen maps to hop. Consider a key of random numbers (numbers only from 0 to 15). Out of the total key size (i.e. 58 numbers) no. of maps used is denoted by preceding two numbers of the key.

Here 1 1 is hex will be $(11)_h = (17)_d$. We calculate 18 (mod 7) = 3. So the #maps = 3(i.e. 0-3 = 4 maps). Each map needs a 56 bit sub key, so we need another 56 x 4 = 224 bits. Consider those random numbers are as follows:

1 1 11 13 1 4 4 11 3 10 8 14 6 9 7 7 1 14 10 14 6 2 14 15 9 7 1 7 11 0 8 10 7 1 6 11 8 4 11 9 14 5 3 4 3 7 1 10 14 7 5 9 5 6 5 15 8 11

Represented in hexadecimal format, the key will be: "11 371AE759565F8B 8A716B84B9E534 1EAE62EF9717B0 BD144B3A8E6977"

Here are the four chaotic maps:

0^{th} Logistic map: $x_{n+1} = 3{:}901x_n(1-x_n);\ x_n \in (0, 1)$

1^{th} Logistic map: $x_{n+1} = 3{:}931x_n(1-x_n);\ x_n \in (0, 1)$

2^{th} Logistic map: $x_{n+1} = 3{:}961x_n(1-x_n);\ x_n \in (0, 1)$

3^{th} Logistic map: $x_{n+1} = 4x_n(1-x_n);\ x_n \in (0, 1)$

According to the key, there are four maps involved. The details are in table 1.

Table 1: Chaotic Maps

	Map 0	**Map 1**	**Map 2**	**Map 3**
Seed	0.003611 367	0.00907 3003	0.00201 0722	0.00123 91499
offset	0.000022 87	0.00003 3977	0.00006 1335	0.00001 499
Settles	125	259	53	135
Orbits	12	7	15	11
Sample	15	8	4	11

The above map will give numbers (x_n) from 0 to 1. Required random number is extracted using following procedure:

Step 1: Remove the decimal point from x_n so we will get integer number.

E.g. 0.1346798520123→1346798520123, 0.58791→58791

Step 2: Transform number into 8 digits. If the number is larger than 8 digits, remove extra digits from LSB. If the number fails to fill 8 digits, append 0.

E.g. 1346798520123 → 98520123, 58791 → 58791000

Step 3: Perform Mod operation on 8 digit number by 256. Result will be pseudo random number.

E.g., 98520123 (mod 256) = 59, 58791000 (mod 256) = 88 the generated random numbers are in between 0 and 255 which can give 8-bit binary key in value permutation algorithm.

E. Experimental Analysis

Several tests are performed to check the security of the proposed system. Statistical tests include histogram analysis and calculation of correlation coefficient of adjacent pixels. Security tests against differential attack include calculation of the NPCR and UACI. Experimental analysis for the proposed encryption scheme is performed on "lena", "pepper", "BrainMRI" and "AbdomenMRI" images. Their original, encrypted and recovered images are shown in figure 5.

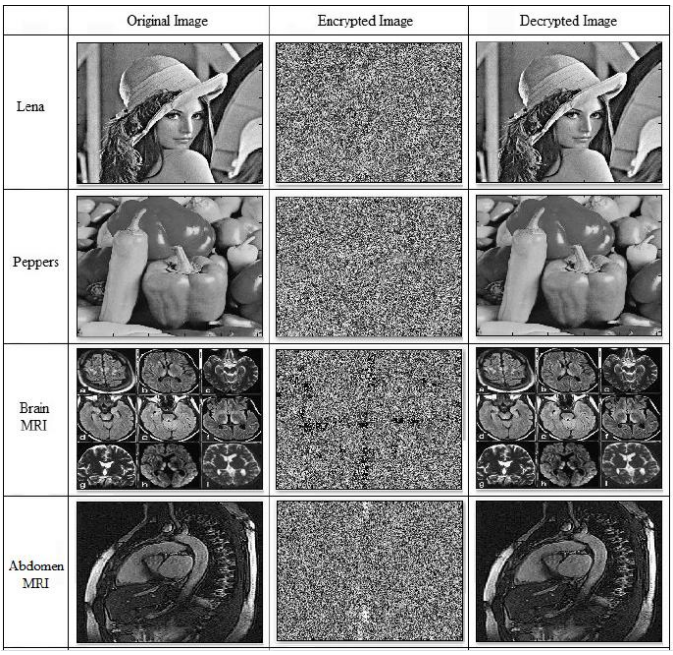

Fig. 5. The Plain, Encrypted and Decrypted Image

1. Histogram analysis

Histogram of plain image and encrypted image should be different from each other and encrypted image should not reveal any statistical details related to the plain image.

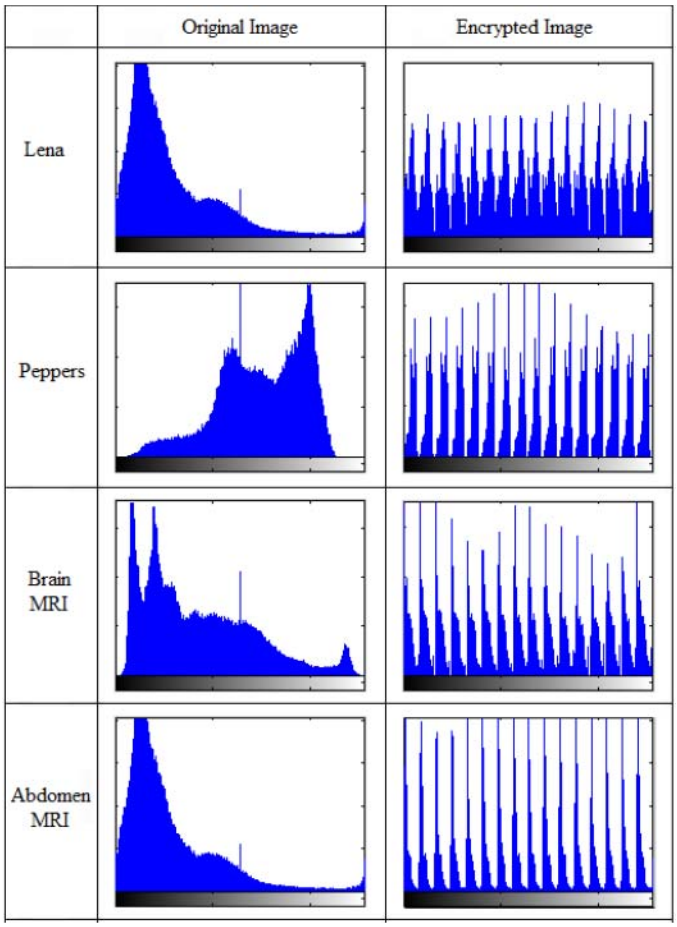

Fig. 6. Histogram of Original Image and Cipher Image

2. Correlation Coefficient analysis

Correlation between adjacent pixels helps to predict 2nd pixels in the pair if knowing the 1st pixel. Hence it should be near to 0 (i.e. uncorrelated). Correlations between nearby pixels can be computed using following formula:

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (2)$$

Where x and y are the values of two adjacent pixels in the image. $cov(x,y)$ is Co-variance, $D(x)$ is Variance.

Table 2: Correlation Coefficients of Original and Encrypted Image

Images		Horizontal	Vertical	Diagonal
Lena	Plain	0.9498	0.9586	0.9408
	Ciphered	0.0162	0.0134	0.0213
Pepper	Plain	0.9654	0.9623	0.9506
	Ciphered	-0.0060	0.0005	0.0286
Brain MRI	Plain	0.8907	0.9315	0.8490
	Ciphered	0.0675	0.0727	0.0361
Abdomen MRI	Plain	0.9494	0.9313	0.9043
	Ciphered	0.0648	0.0508	0.0313

Table 2 indicates the performance analysis of the proposed method using "lena", "pepper", "BrainMRI" and "AbdomenMRI" images. correlation coefficients, ranging from '1' highly correlated to '0' uncorrelated, of pairs of adjacent pixels in different directions. These coefficients ensure the two considered images are statistically independent.

3. Differential Attack Analysis

To check sensitive of a single bit change in the plain image two criteria are used. NPCR (Number of Pixel Change Rate) is the percentage of different pixels among encrypted images whose corresponding original image has one pixel difference. UACI (Unified average Changing Intensity) is average of intensity difference between two encrypted images whose corresponding original image has only one pixel difference.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W * H} * 100\% \quad (3)$$

$$UACI = \frac{1}{W * H}\left[\sum \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] * 100\% \quad (4)$$

Where, *W* and *H* are the width and height of the encrypted images. Two encrypted images *C1* and *C2* are chosen whose corresponding original images have only one-pixel difference.

Table 3: NPCR and UACI of Ciphered Images

Images	NPCR%	UACI%
Lena	99.4251	30.1199
Pepper	99.4308	28.3747
BrainMRI	99.0631	31.3750
AbdomenMRI	99.6669	33.6548

In order to resist differential attack, the NPCR and UACI values should be large enough for an ideal cipher system. We have performed tests on four gray scale images of size 512*512 to measure the influence of differential attack. NPCR and UACI values of our proposed scheme are presented in Table 3. According to the values of NPCR and UACI, proposed algorithm can satisfy security requirements.

4. Video Encryption and Decryption

Video encryption experiment is performed on "viplanedeparture.avi" video file with 40 frames. Table 7 shows frame1, frame11, frame30 of original, encrypted and decrypted video. Times takes to perform video encryption is 90 sec and will increase as number of frames increase.

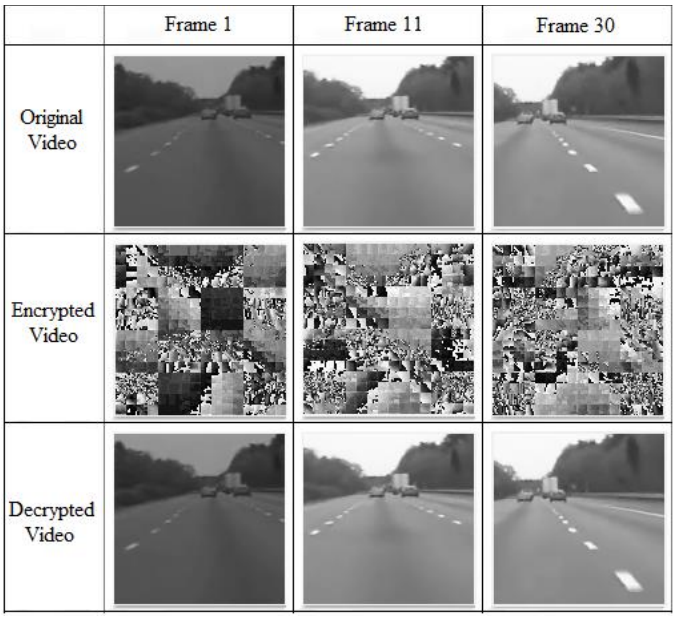

Fig. 7. The Plain, Encrypted and Decrypted Video

IV. CONCLUSION

The proposed scheme is an efficient encryption scheme for confidential storage and transmission of medical images to produce a high security. Out of two phases, first phase performs position permutation on decomposed sub blocks of images and value permutation in second phase gives additional security to resist attack on confidentiality. Experimental analysis shows proposed technique provides required security using correlation coefficient and differential attack analysis. This technique also works well for video encryption. In future, it can be used for color image encryption.

References

[1] Bruno Apolloni, R.J. Howlett, "Image Information Hiding Encryption Using Chaotic Sequence", 11th International Conference, KES 2007, Italy, September 12-14, 2007.

[2] N.D. Parmar, Neha Pandya, "Analysis Of Encryption And Watermarking Techniques For Secure Bluetooth Transmition Of Image Files", International Journal of Engineering Research and Technology, 2013.

[3] Tapas B., B N Chatterji, "Secure Image encryption through key hashing and wavelet transform techniques", International Journal of Emerging Technology and Advanced Engineering, February 2012.

[4] Shoaib Ansari, "Cryptography Technique using Chaotic Map", International Journal of Innovative Research in Engineering and Multidisciplinary Physical Sciences (IJIRMPS) December 2013.

[5] Bani Younes and Aman Jantan, "Image Encryption Using Block Based Transformation Algorithm", IAENG International Journal of Computer Science, 2008.

[6] Mohammad Ali, and A. Jantan, "An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption", IJCSNS, 2008.

[7] LIU Wei and CAO Shui-ping, "Digital Image Encryption Algorithm Based on Chaos and Improved DES", Proceedings of the 2009 IEEE International Conference on Systems, TX, USA - October 2009.

[8] K. Singh and K. Kaur, "Image Encryption using Chaotic Maps and DNA Addition Operation and Noise Effects on it", International Journal of Computer Applications 2011.