

Image encryption based on permutation-substitution using chaotic map and Latin Square Image Cipher

H.T. Panduranga^{1,a}, S.K. Naveen Kumar^{1,b}, and Kiran^{2,c}

¹ DoS in Electronics University of Mysore, Hemangothri PG Center, Hassan, Karnataka, India

² Dept. of E and C Engg., Malnad College of Engineering, Hassan, Karnataka, India

Received 21 January 2014 / Received in final form 10 February 2014
Published online 19 March 2014

Abstract. In this paper we presented a image encryption based on permutation-substitution using chaotic map and Latin square image cipher. The proposed method consists of permutation and substitution process. In permutation process, plain image is permuted according to chaotic sequence generated using chaotic map. In substitution process, based on secrete key of 256 bit generate a Latin Square Image Cipher (LSIC) and this LSIC is used as key image and perform XOR operation between permuted image and key image. The proposed method can applied to any plain image with unequal width and height as well and also resist statistical attack, differential attack. Experiments carried out for different images of different sizes. The proposed method possesses large key space to resist brute force attack.

1 Introduction

In today world of developing technology, it is becoming more difficult to keep information safe. One method to protect information from unauthorized eavesdropping is to use an encryption algorithm. Image security attracts extensive concerns from the public and the government in recent years. Unexpected exposure of private photos and divulged military and governmental classified images emphasizes the importance of image security again and again. With the fast development of digital storage devices, computers, and the world-wide network, a digital image can be easily copied to mobile storage or transferred to the other side of the world within a second. However, such convenience could also be used by malicious/ unauthorized users to rapidly spread the image information such that it may cause uncountable losses for the owner(s) of images.

^a e-mail: ht_pandu@yahoo.co.in

^b e-mail: nave12@gmail.com

^c e-mail: kiran.mtech12@gmail.com

Two basic techniques to obscure high redundancies and strong correlations are confusion (substitution) and diffusion (permutation). Confusion increases the complexity between the key and the cipher text and bans all attempts to study the cipher text for redundancies and statistical patterns. Diffusion, on the other hand, spreads the redundancy of the plaintext over the entire cipher text, thus decreasing redundancy. While either of these techniques alone is highly susceptible to being cracked, they generally make an excellent security solution when combined. Images, however, have various intrinsic features, such as bulk data capacity and high correlation among pixels, that render traditional encryption algorithms such as DES, IDEA and RSA unsuitable [1,2]. Xindyuan Wang et al. presented a novel chaotic image encryption algorithm based on water wave motion and water drop diffusion models. Secret keys will be processed by key generator before they can really be used in the encryption scheme, and in this stage this paper associates plain image with secret keys; Secondly, by imitating the trajectory of water wave movement, encryption algorithm will do scrambling operations to the image. Finally combines water drop motion and dynamic look up table to realize diffusion operations. For an 8 bits pixel, this algorithm will just dispose the higher 4 bits, which is because the higher 4 bits contain the vast majority of information of the image [3]. Ruisong Ye presented a novel chaos based image encryption scheme with an efficient permutation diffusion mechanism. Generally permutation diffusion mechanism permuting the positions of image pixels in order to reduce the high correlation between adjacent pixels of plain image and change the pixel value in diffusion stage. In the permutation process, a generalized Arnold map is utilized to generate one chaotic orbit used to get two index order sequences for the permutation of image pixel positions; in the diffusion process, a generalized Arnold map and a generalized Bernoulli shift map are employed to yield two pseudo-random gray value sequences for a two-way diffusion of gray values. Encryption scheme is easy to manipulate and can be applied to any image with unequal width and height as well [4]. Ahmed A. Abd El-Latif et al. have proposed a hybrid chaotic system and cyclic elliptic curve for image encryption and provides a external secret key of 256 bit and one generalized chaotic logistic map. using the cyclic elliptic curve to derive generated key stream are mixed with key sequences [5]. Xingyuan Wang have proposed A novel color image encryption algorithm based on chaos. They uses chaotic system to encrypt the R, G, B components of a color image at the same time and makes these three components affect each other. So it can reduces correlation between R, G, B components and security is increased [6]. Vinod patidar et al. presented a secure chaotic based permutation – substitution scheme of image encryption. This is loss-less symmetric block cipher. They are used secret key of length 161 bit and this key can be used as initial condition and system parameter of chaotic map. Number of rounds depends on secret key. To increase the speed of encryption they convert 3D image matrix into 2 D image matrix. Permutations are done by row by row and column by column using pseudo random number sequence generated from chaotic sequence. In substitution process uses chaotic sequence and initial vector depends on secret key and mixed with plain image [7]. Chong Fu et al. presented a novel chaos based bit level permutation scheme for digital image encryption and provides a fast and high security. To overcome the drawbacks of conventional algorithms they propose significant diffusion effect in permutation procedure through a two stage bit level shuffling algorithm. Arnold cat map and chaotic sequence are used for shuffle all bit planes. This method decreases computational complexity and real time image communication applications [8]. Yue Wu et al. have presented image encryption using the Sudoku matrix. Sudoku matrix define as no two digits in the same block can be aligned in the same row, column or box. encryption of the image consists of three stages. in first stage, a reference Sudoku matrix is generated and it is used for scrambling process. The image pixels intensities are then changed by

using the reference Sudoku matrix values, and then the pixels positions are shuffled using the Sudoku matrix as a mapping process. so using this matrix we can encrypt any digital images such as binary images, gray and RGB images. Logistic map used for control the size of Sudoku matrix [9]. Yue Wu et al. have proposed A novel Latin square image cipher. provides a 256 bit key length for generating s Latin square and generates 256×256 square image and it looks like Sudoku matrix that is no two digit in the same block can be aligned in the same row, column or box. LSIC achieve many desired properties of a secure cipher including a large key space, high key sensitivities, uniformly distributed cipher text, excellent confusion and diffusion properties, semantically secure, and robustness against channel noise [10]. Yue Wu et al. have presented Sudoku associated two dimensional bijections for image Scrambling. Sudoku configuration provides us a new alternative way of matrix element representation by using block-grid pair besides the conventional row-column pair. And also discovers six more representations by using row digit pair, digit row pair, column digit pair, digit column pair, digit block pair, block digit pair associated with a Sudoku matrix. Sudoku Associated Image Scrambler only using Sudoku associated two dimensional bijections for image scrambling without bandwidth expansion [11]. Have proposed a Sudoku based wet paper hiding scheme in which a secret key has been used to randomly select a subset of pixels from a cover image as dry pixels. Then a total automorphism is applied to the cover image to maximize the number of dry pixel pairs and each secret digit in the base-9 numeral system is embedded into one dry pixel pair. Azzaz, MS and Tanougast [12] presented FPGA implementation of new real-time image encryption based switching chaotic systems allows a low cost image encryption for embedded systems and provide a good trade – off between performance and hardware resource. Chin-Chen Chang et al. [13] have presented a Sudoku based secret image sharing scheme to lossless reveal of secret image. And also their approach derives the secret shadows and generates the meaningful shadow images by adopting the Sudoku. Azzaz, MS and Tanougast [14] presented Real-time image encryption based chaotic synchronized embedded cryptosystems that deals with the chaotic synchronization for embedded hardware cryptosystems and its FPGA implementation for designing a real time image secure symmetric encryption scheme.

The proposed method consists of permutation and substitution process. In permutation process, plain image is permuted according to chaotic sequence generated using chaotic map. In substitution process, based on secrete key of 256 bit generate a Latin Square Image Cipher (LSIC) and this LSIC is used as key image and perform XOR operation between permuted image and key image. In the following sections, chaotic functions and overview of Latin square image cipher are introduced, the proposed method is explained and the experimental results and conclusions are presented.

2 Chaotic function and Latin square image cipher

In this section, the chaotic function and Latin square image cipher will be described briefly.

2.1 Chaotic function

Chaotic functions are similar to noise signals, but they are completely certain; that is, if we have the primary quantities and the drawn function, the exact signal can be reproduced. The advantage of these signals are as follows [15]: (1) Sensitivity to primary conditions By this advantage, we mean that a minor change in the primary amount will cause a significant difference in subsequent measures. If we have a small

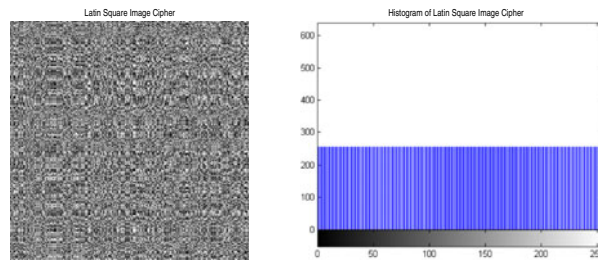


Fig. 1. Latin Square Image Cipher and Histogram.

change in the signal amount, the final signal will be completely different (2). Apparently accidental feature In comparison to producing accidental natural numbers in which the range of the numbers cannot be reproduced, the technique used for producing accidental numbers is based on a chaotic function such that if we have the primary quantities and the drawn function, then we can reproduce the numbers (3). Deterministic work As the chaotic functions have an accidental manifest, they are completely exact. Because we have the drawn function and the primary quantities, we can produce and reproduce sets of numbers that seemingly have no system or order. Equation (1) shows one of the most famous signals that exhibits chaotic features; this equation is known as the logistic map signal:

$$X_{n+1} = rX_n(1 - X_n). \quad (1)$$

The logistic map chaotic signal used has primary values of $X_0 \in [0, 1]$ and $r \in [3.57, 4]$.

2.2 Overview of Latin Square Image Cipher

A Latin square of order n is an $n * n$ array in which each of the n^2 cells contains a symbol from an alphabet of size n , such that each symbol in the alphabet occurs just once in each row and once in each column. The name “Latin square” was inspired by mathematical papers by Leonhard Euler, who used Latin characters as symbols. we derived the theory of Latin Square and generation of Latin Square Image Cipher (LSIC) from the work presented by Yue Wu [10]. To generate LSIC we have used a key of 64 hexadecimal characters length. Each character has 4 bits length, i.e. length of key is 256 bits. The used key is F5A172A6E8B163D987C23A78B12F73A6519D76C53B12A64CC67B8981267ABFD. The Figure 1 shows Latin Square Image Cipher generated from the above mentioned 256 bits length key and also its histogram.

2.3 Size independent

The proposed method can applied to any plain image with unequal width and height as well. If the plain image of size is not divisible by 16, then following method is used to select a sub image from the plain image as explained below.

Consider a plain image of width M and height N respectively. Which is not divisible by 16, then selection of sub image from the plain image as shown in Figure 2.

steps for selecting the sub image from plain image as below.

step 1: plain image I of size M and N ,

step 2: if($\text{mod}(M, 16) = 0$) and ($\text{mod}(N, 16) = 0$).

$$temp_m = 16 * \text{floor}((M/16));$$

$$temp_n = 16 * \text{floor}((N/16)); \text{intermediate_row} = m - temp_m;$$

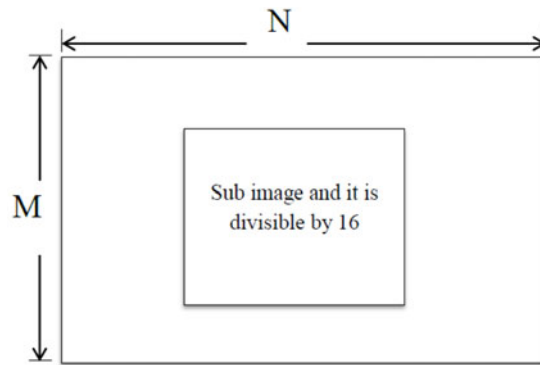


Fig. 2. Selection of sub image from the plain image.

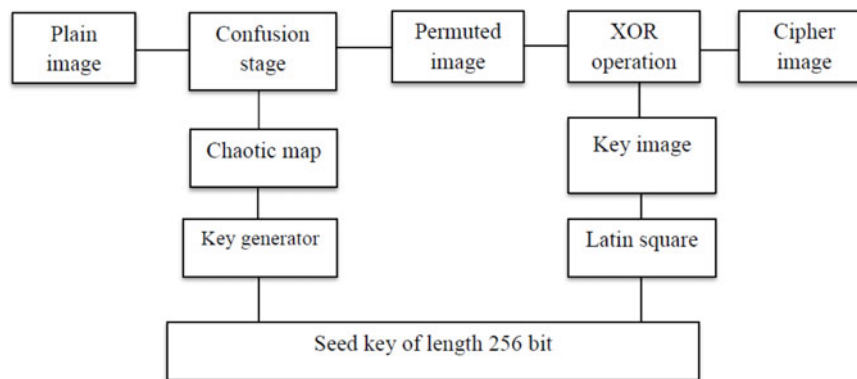


Fig. 3. Block diagram of proposed method.

$$intermediate_col = n - temp_n;$$

$$st_row = \frac{\text{floor}(intermediate_row)}{2};$$

$$end_row = intermediate_row - st_row;$$

$$st_col = \frac{\text{floor}(intermediate_col)}{2};$$

$$end_col = intermediate_col - st_col;$$

$$sub_image = I((st_row + 1) : (m - end_row), (st_col + 1) : (n - end_col))$$

$$\text{else } sub_image = I$$

step 3:end.

So proposed method is applied to sub image and remaining pixels from the plain image is stored into 1D array and shuffled this 1D array according to chaotic sequence generated from chaotic map. Again placed this encrypted pixels in the same position and also encrypted version of sub image into same position and finally get encrypted image.

3 The proposed method

The architecture of the overall image encryption cryptosystem using the proposed algorithm is shown in Figure 3. The proposed method consists of two stages.

Table 1. Compare the key space of the proposed algorithm with the existing algorithms.

Algorithms	key space
Ref. [3]	10^{72}
Ref. [19]	$3.40 * 10^{38}$
Proposed algorithm	$2^{256} = 1.15 * 10^{77}$

The first stage that is permutation stage, plain image is permuted using the chaotic sequences. According to this initial parameters chaotic sequences is generated using 1D chaotic map. After which the index values that are stored in a row matrix, are generated according to ascending order of the chaotic sequence. Index value represents the position of chaos sequence. According to this index values plain image are permuted and get a permuted image. In diffusion stage external secret key of 256 bit is applied to Latin square. Basically Latin Square looks like a Sudoku and generate key image of size 256×256 and each row and column of key image each pixel value ranging from 0 to 255 is repeated exactly one time and using this key image perform XOR operation between permuted image and key image and get a encrypted image.

The proposed image encryption process utilizes an external secret key of 256-bit long. Further, the secret key is divided into blocks of 8-bit each, referred as session keys.

$K = k_1, k_2, k_3, \dots, k_{64}$

here, k_i s are the alphanumeric characters (09 and AF) and each group of two alphanumeric characters represents a session key. Alternatively, the secret key can be represented in ASCII mode as

$K = k_1, k_2, k_3, \dots, k_{32}$ (in ASCII)

binary form of K is

$K = b_0, b_1, b_2, b_3, \dots, b_{255}$

X_0 is initial condition for chaotic map and calculated as

$X_0 = 0;$

$$X_0 = X_0 + \sum_{i=0}^{255} \frac{b(i)}{2^i} X_0 = \text{mod}(X_0, 1). \quad (2)$$

4 Experimental results and performance analysis

4.1 Key space analysis

Key space size is the total number of different keys which can be used in the encryption process. For a good encryption algorithm, the key space should be large enough to make brute-force attack impossible. In the proposed algorithm, the initial secret keys set $K = 256$ bit. Therefore, the key space is 2^{256} . Such a big key space can provide a sufficient security against brute-force attacks [16]. Comparison of key space analysis of various algorithms are tabulated in Table 1. From the Table 1 we can say that proposed method having larger key space compared to existing algorithms. So this key space is also large enough to resist brute attacks.

4.2 Statistical analysis

In the proposed encryption algorithm, a random number key stream was generated and this key applied to Latin square to get a Latin image cipher (LSIC), With the LSIC we encrypt the pixel values sequentially. Therefore, the diffused image is randomly distributed. This is shown by a test on the histograms of the cipher-images in Section 4.2.1, the information entropy of the cipher-image in Section 4.2.2, the correlations of adjacent pixels in the plain-image and cipher image in Section 4.2.3, Analysis of differential attack in Section 4.2.4. Analysis of key sensitivity in Section 4.2.5.

4.2.1 Histogram analysis

An image histogram illustrates that how pixels in an image are distributed by plotting the number of pixels at each gray scale level. The distribution of cipher-text is of much importance. More specifically, it should hide the redundancy of plain-text and should not leak any information about the plain-text or the relationship between plain-text and cipher-text. Table 2 shows the histograms of plain-images and its ciphered images generated by the proposed scheme respectively. It's clear from that the histograms of the cipher-images are fairly uniform and significantly different from that of the plain image and hence do not provide any clue to employ statistical attack.

4.2.2 Information entropy analysis

In information theory, entropy is the most significant feature of disorder, or more precisely unpredictability. To calculate the entropy $H(X)$ of a source x , we have:

$$H(X) = \sum_{i=1}^n Pr(x_i) \log_2 \frac{1}{Pr(x_i)} \quad (3)$$

where X denotes the test image, x_i denotes the i^{th} possible value in X , and $Pr(x_i)$ is the probability of $X = x_i$, that is, the probability of pulling a random pixel in X and its value is x_i . For a truly random source emitting $2N$ symbols, the entropy is $H(X) = N$. therefore, for a ciphered image with 256 gray levels, the entropy should ideally be $H(X) = 8$. If the output of a cipher emits symbols with entropy less than 8, there exists certain degree of predictability, which threatens its security. The entropies for plain image and ciphered images using various images are calculated in Table 3. Apparently, the proposed algorithm is much closer to the ideal situation. This means that information leakage in the encryption process is negligible and the cryptosystem is secure against entropy attack.

Table 4 shows the comparison of proposed algorithm with existing algorithm. It is clear that proposed algorithm is better than existing algorithms. In our proposed method we are used Latin Square (entropy of Latin Square image = 8 = entropy of random image) at diffusion stage – which leads to an improved entropy in encrypted image. As compared to entropy of encrypted images in reference [3, 18, 19] resultant entropy of our proposed method is more nearer to entropy of random image.

4.2.3 Analysis of correlation of adjacent pixels

For an ordinary image having definite visual content, each pixel is highly correlated with its adjacent pixels either in horizontal, vertical or diagonal direction.

Table 2. Resultant Encrypted Images and its histogram of proposed method.

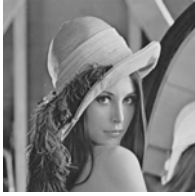
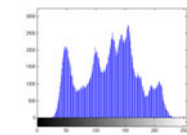
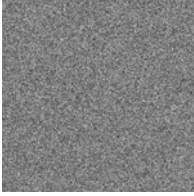
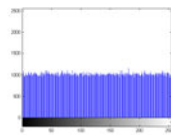

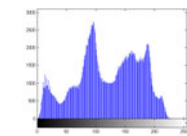
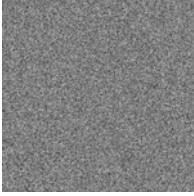
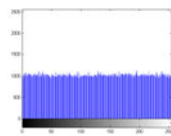

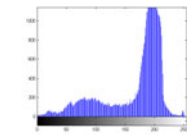
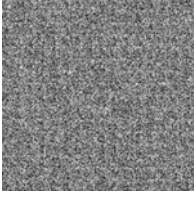
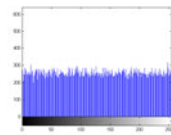
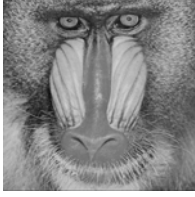
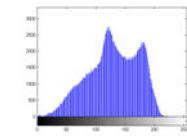
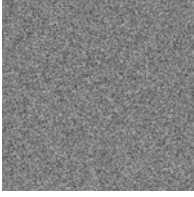
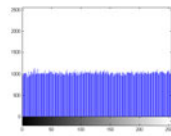

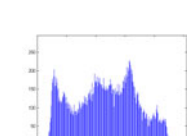
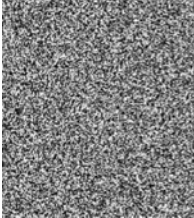
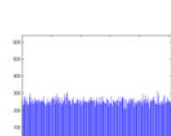

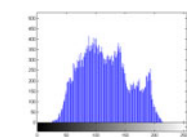
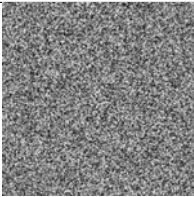
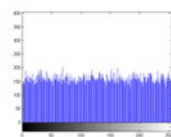

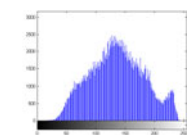
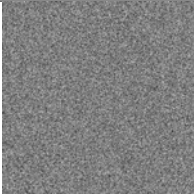
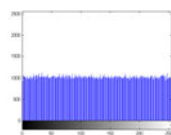
Input Image	Histogram	Encrypted Image	Histogram
			
			
			
			
			
			
			

Table 3. Entropy.

Input Images	size	Plain Image Entropy	Encrypted Image Entropy
Lena	512 × 512	7.4451	7.9993
Peppers	512 × 512	7.5937	7.9993
Plane	256 × 256	6.9550	7.9989
Babon	512 × 512	7.3583	7.9993
Barb	150 × 172	7.6219	7.9931
Butterfly	203 × 203	7.3060	7.9952
Elaine	512 × 512	7.5060	7.9993

Table 4. Encrypted image entropy of proposed algorithm with existing algorithms.

Image	Size	Proposed	Ref. [3]	Ref. [17]	Ref. [18]	Ref. [19]
Lena	512 × 512	7.9993	7.99904	7.9923	7.99828	7.9952
babon	512 × 512	7.9993	7.98996	7.9926	7.99891	7.9972

Table 5. Correlation coefficients of two adjacent pixels in plain-image and ciphered-images of proposed method.

Image	Plain image			Encrypted Image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	0.9691	0.9841	0.9842	0.0008	0.0023	0.0045
Peppers	0.9733	0.9764	0.9809	0.0003	-0.0032	-0.0015
Plane	0.8908	0.8850	0.8861	-0.0051	-0.0151	-0.0116
Babon	0.8652	0.7524	0.7567	0.0019	0.0011	0.0025
Barb	0.8948	0.8824	0.8786	0.0042	0.0001	0.0059
Butterfly	0.9261	0.9116	0.9119	0.0007	-0.0034	0.0012
Elaine	0.9738	0.9695	0.9697	-0.0032	0.0007	0.0025

However, an efficient image cryptosystem should produce the cipher image with sufficiently low correlation in the adjacent pixels. The visual testing of the correlation of adjacent pixels can be done by plotting the distribution of the adjacent pixels in the plain image and its corresponding cipher image. The correlation distribution of two horizontally adjacent pixels, two vertically adjacent pixels and two diagonally adjacent pixels of the plain image and the cipher image produced by the proposed scheme is shown in Table 5, respectively. It is clear that the strong correlation between adjacent pixels in plain image is greatly reduced in the cipher image produced by the proposed scheme. To quantify and compare the correlations of adjacent pixels in the plain and cipher image, the following procedure is carried out. First, randomly select 1000 pairs of adjacent pixels in each direction from the plain image and its ciphered image. Then, calculate the correlation coefficient $r_{x,y}$ of each pair by using the following four formulas:

$$\text{cov}(x, y) = E(x - E(x))(y - E(y)) \tag{4}$$

$$E(x) = \frac{1}{N} \sum_{i=1}^N X_i \tag{5}$$

$$D(x) = \frac{1}{N} \sum_{i=1}^N (X_i - E(X_i))^2 \tag{6}$$

Table 6. Comparison of the correlation coefficients of Lena.

Method	Plain image		
	Horizontal	Vertical	Diagonal
Plain Lena image	0.9841	0.9691	0.9842
Ref. [3]	0.0174	0.02046	0.0231
Ref. [17]	0.0342	0.0192	0.0084
Ref. [18]	0.0271	0.0321	0.0383
Ref. [19]	0.0067	0.0031	0.0383
Proposed algorithm	0.0004	-0.0042	-0.0040

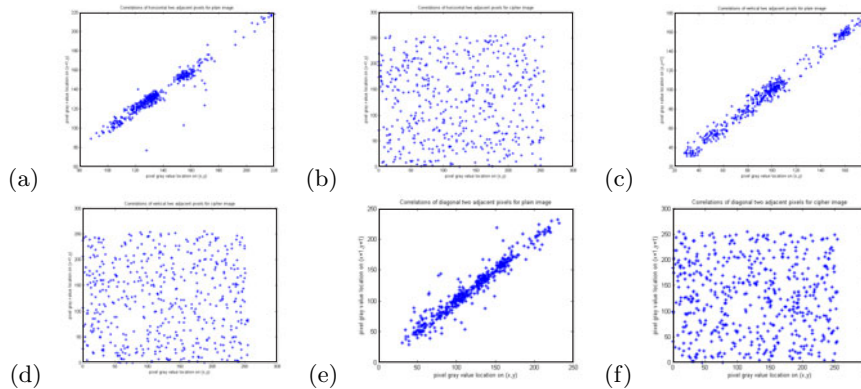


Fig. 4. Correlations of two adjacent pixels for lena image of size $512 * 512$ (a) horizontal direction of the plain image, (b) horizontal direction of the cipher image, (c) vertical direction of the plain image, (d) vertical direction of the cipher image, (e) diagonal direction of the plain image, and (f) diagonal direction of the cipher image.

where x and y are grayscale values of two adjacent pixels in the image and N denotes the total number of samples, $cov(x, y)$ is covariance, $D(x)$ is variance, $E(x)$ is mean.

Table 6 shows that proposed algorithm lower than the existing ones, so the encryption effect is rather good.

4.2.4 Analysis of differential attack

A well-designed encryption algorithm should be highly sensitive to plain-image and keys, so a slight change in plain-image or keys will make the cipher-image quite different. If an encryption scheme contains no confusion or diffusion stage, it would easily be destroyed by differential attacks. In order to confirm whether the proposed encryption algorithm is sensitive to plain image and keys, this paper brings out two tests: Number of pixels change rate (NPCR) and Unified average changing intensity (UACI) [21]. The equation to calculate UACI is Eq. (7)

$$UACI = \frac{1}{M * N} \sum_{i,j} \frac{|C1(i, j) - C2(i, j)|}{255} \times 100\%. \tag{7}$$

Where, M stands for images width, N stands for images height, $c1(i, j)$ means the gray-scale value of cipher-image in position (i, j) , and $c2(i, j)$ means the gray-scale

Table 7. NPCR and UACI of proposed method.

Image	NPCR(%)	UACI(%)
Lena	99.6892	33.4047
Peppers	100	33.5141
Plane	100	33.5128
Babon	100	33.5175
Barb	100	35.6165
Butterfly	100	39.22
Elaine	100	33.5268

Table 8. Comparison of NPCR and UACI with respect to Lena image.

Algorithms	NPCR(%)	UACI(%)
Ref. [3]	99.611	33.485
Ref. [18]	99.617	33.4933
Ref. [19]	96	31.79
Proposed method	99.6892	33.4047

value of the new cipher-image which is the encryption result of modified plain image that has just one different pixel to the original plain-image. NPCR can be calculated by Eq. (8)

$$NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%. \quad (8)$$

Where, M stands for images width, N stands for images height and where $D(i,j)$ defined as follows

$$D(i,j) = \begin{cases} 1 & \text{if } C1(i,j) \neq C2(i,j); \\ 0 & \text{if } C1(i,j) = C2(i,j). \end{cases}$$

When one bit of a pixels gray-scale value in the plain image is changed, then a new plain image is generated from the original one. Encrypt the two images with the same secret keys, then take cipher images into Eqs. (7) and (8) and results are shown in Table 7.

From the Table 8 results we can find that our algorithm is very sensitive to tiny changes in the plain image, even if there is only one bit difference between two plain images, the decrypted images will be completely different.

4.2.5 Key sensitivity analysis

An ideal image encryption procedure should be sensitive to the secret key. It means that a change in a single bit of the secret key should produce a completely different encrypted image. Key sensitivity analysis has been performed for the proposed image encryption algorithm and the results are summarized as follows:

- 1) The encrypted image of Lena is decrypted by making a slight modification in the original key "99AE698B2081E75FE916CBAC09A54FD9C887D8A2CED335A8C980FD7BCF40A626" and the resultant encrypted image is referred as C1 and shown in Fig. 5a.

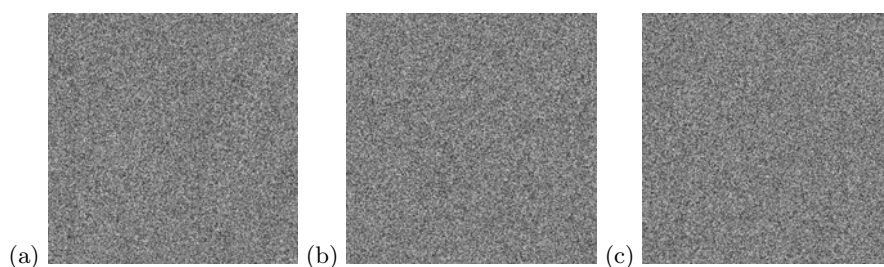


Fig. 5. Frame (a–c) show the decrypted images from the encrypted image of Lena using slightly different keys than the key used for the encryption.

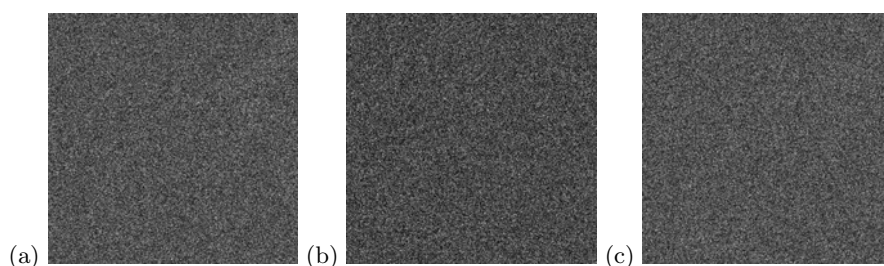


Fig. 6. (a) Difference ($C1-C2$). (b) Difference ($C2-C3$). (c) Difference ($C3-C1$).

- 2) The encrypted image Lena is decrypted by making a slight modification in the original key “49DE698B2081E75FE916CBAC09A54FD9C887D8A2CED335A8C980FD7BCF40A626” and the resultant encrypted image is referred as C2 and shown in Fig. 5b.
- 3) The encrypted image Lena is decrypted by making a slight modification in the original key “49AE698B2081E75FE916CBAC01A54FD9C887D8A2CED335A8C980FD7BCF40A626” and the resultant encrypted image is referred as C3 and shown in Fig. 5c.

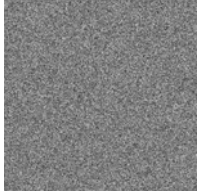

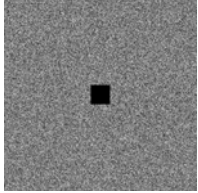

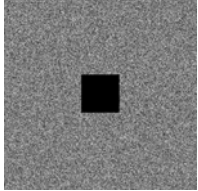

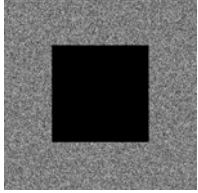

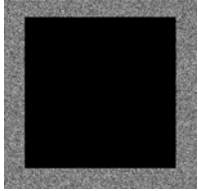

With a slight change in the key, one is unable to find any clue about the original image from the decrypted image. Having the right pair of secret key is an important part while decrypting the image, as a slight change in the secret key will not retrieve the exact original image. Above example shows that the decryption of the encrypted image with the wrong secret key will not reveal any information about the original image hence shows the effectiveness of the proposed technique.

Encrypted images $C1$, $C2$ and $C3$ have significant differences. These can be verified by the fact that their differences i.e. ($C1-C2$), ($C2-C3$) and ($C3-C1$) are random-like images as shown in Fig. 6.

4.2.6 Noise robustness analysis

A good cipher should also be able to tolerate a certain amount of noise, e.g. noise in a channel or decoding errors. As discussed previously, the proposed Latin square image cipher adopts an asymmetric structure for encryption and decryption, and one noisy pixel in cipher image will only propagate in a factor of two in each round. Table 9 shows the results of the decryption robustness of the Latin square image cipher against various noise ratio in cipher images. After decryption, noise concentrating in the center

Table 9. Sample results of noise robustness in decryption.

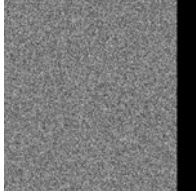

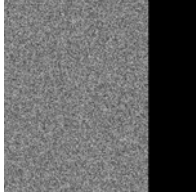

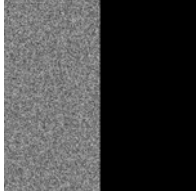

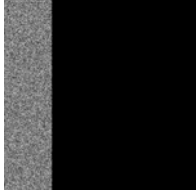

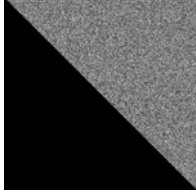

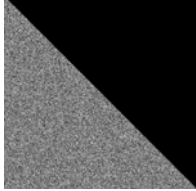

Encrypted Image	Decrypted image	PSNR
		infy
		47.5360
		41.5787
		33.5654
		29.7034

square of a cipher image now distributed almost evenly over the deciphered image. Due to the psychovisual redundancy within an image, human vision system is still able to recognize the deciphered image contents as long as it is not fully unintelligible. We check the tolerance against the loss of encrypted image. By replace the 10, 25, 50 and 75 percent of the encrypted image and upper Diagonal and lower Diagonal pixels with intensity value 0 (pure black.) The occluded encrypted images and the corresponding decrypted images with correct keys are shown in Table 10.

5 Conclusion

Image encryption based on permutation-substitution using chaotic map and Latin square image cipher. Plain image is permuted according to chaos sequence. In diffusion stage all the pixel values are changed by performing XOR operation between

Table 10. Sample results of Random noise robustness in decryption.

Encrypted Image	Decrypted image	PSNR
		37.4247
		33.5638
		30.5738
		28.8163
		30.6079
		30.6067

permuted image and key image. Security analysis including statistical attack analysis and differential attack analysis are performed numerically and visually. All the experimental results show that the proposed encryption scheme is secure thanks to its large key space, its high sensitivity to the cipher keys and plain-images. The results show that the proposed scheme leads to a higher security level in terms of NPCR, UACI and entropy of the cipher-images. The proposed encryption scheme is easy to manipulate and can be applied to any image with unequal width and height as well.

References

1. Salleh, Mazleena, Ibrahim, J. Teknol. **39**, 1 (2012)
2. Usama, Muhammad, Khan, Biom. Sec. Technol., 1 (2008)
3. Wang, Xingyuan, Yang, Lei, Opt. Comm. **285**, 4033 (2013)
4. Ye, Ruisong, J. Teknol. **284**, 5290 (2011)
5. Abd El-Latif, Ahmed A., Niu, AEU-International J. Electron. Comm. (2012)
6. Wang, Xingyuan, Teng, Lin, Qin, Xue, Signal Proc. **92**, 1101 (2012)
7. Patidar, Vinod, Pareek, Opt. Comm. **284**, 4331 (2011)
8. Fu, Chong, Lin, Bin-bin, Miao, Opt. Comm. **284**, 5415 (2011)
9. Wu, Yue, Zhou, Yicong, Noonan, Joseph P., Panetta, SPIE Defense, Sec. Sensing, 77080 (2010)
10. Wu, Yue, Zhou, arXiv preprint [arXiv:1204.2310] (2012)
11. Wu, Yue, Agaian, arXiv preprint [arXiv:1207.5856] (2012)
12. Azzaz, M.S., Tanougast, Signals Syst. Conf. 1 (2009)
13. Chang, Chin-Chen, Lin, J. Comm. **5**, 5 (2010)
14. Azzaz, M.S., Tanougast, NEWCAS Conf. **92**, 61 (2010)
15. Wang, Xingyuan, Teng, Lin, Qin, Xue, Adv. Recent Technol. Comm. Comput., 623 (2009)
16. Tong, Xiaojun, Cui, Image Vision Comput. **26**, 843 (2008)
17. F. Sun, Z.L.S. Liu, Opt. Comm. **283**, 2066 (2010)
18. Abdullah, Abdul Hanan, AEU-Int. J. Electron. Comm. **66**, 806 (2012)
19. Wang, Xingyuan, Jin, J. Comm. **5**, 5 (2010)
20. Pareek, Narendra K., Patidar, Digital Signal Proc. (2012)
21. Wu, Yue, Noonan, Joseph P., Agaian, arXiv preprint [arXiv:1103.5520] (2011)
22. Wu, Yue, Noonan, Joseph P., Agaian, Cyber J.: Multidiscipl. J. Sci. Technol., 31 (2011)