Image Scrambling through Two Level Arnold Transform

Satish A*, Erapu Vara Prasad*, Tejasvi R*, Swapna P*, Vijayarajan R*

* Department of ECE,
*RGM College of Engineering and Technology, Nandyal, AP
anchulasathish@gmail.com, varaprasad.erapu@gmail.com, reddytejasvi1998@gmail.com
swapna1cube@gmail.com, viraj2k@gmail.com

Abstract- In today's world of multimedia applications, digital image security is of utmost importance. This paper proposes a two level image scrambling algorithm using Arnold transform with different number of iterations. Pixels are scrambled for each iteration in the conventional Arnold scrambling. Due to its periodic property, the original image can be retrieved by experimenting the inverse Arnold transform on the scrambled image. The main objective of this paper is to improve the complexity of malicious decryption by considering Arnold scrambling for pixels and blocks. In the first level, Arnold pixel scrambling is performed on the original image of size N × N, where the pixels are scrambled based on number of iterations. In the second level the scrambled image is divided in to 8×8 blocks for which Arnold scrambling is conducted block-wise. In the decryption process, inverse Arnold block scrambling is applied first for the scrambled image and Arnold pixel scrambling is conducted second. Decryption process needs to be carried out with proper order of scrambling and iterations. This order of scrambling and number of iterations increase the complexity of malicious decryption. Correlation coefficient, number of pixel change rate and unified average changing intensity are computed between the scrambled and original images for performance analysis. The proposed method is also analysed in each level for its robustness.

Index Terms-Arnold scrambling, encryption, decryption, structural similarity index

I. INTRODUCTION

Due to recent advancement in imaging technologies and multimedia applications, a huge amount of multimedia content in the form of text, audio, image, video and computer generated contents are exchanged in various networks. These multimedia contents often exposed to malicious access, tampering, copyright and ownership issues. All these issues are dealt with the solutions in the form of watermarking [1, 2, 3], authentication and cryptography algorithms [4, 5]. High security should be provided for an image that contains confidential information. Subsequently, the confidential information will be hidden in the image in the form of secret authentication codes by using different algorithms [6]. In order to differ from conventional methods to hide data in the image, an algorithm can be developed using scrambling. Digital image scrambling transforms an image in such a way that an intruder can't recognise the image as it is a disguised one. Image scrambling technology provides an algorithm which doesn't involves any password or any authentication codes [7] but provides high security for the image that is being transmitted. The main purpose is to transmit the image through the public networks. After scrambling an image, the scrambled image will be disorganised, so an intruder can't decipher it [7].

Image scrambling can be performed by using Arnold scrambling algorithm [8]. The special feature of Arnold scrambling is that it uses periodicity concept [7]. According to Arnold scrambling, the original image can be recovered after a certain number of iterations based on the size of the image. But the number of iterations will be different for different size of the images and the number of cycles does not follow any order. Currently, Arnold scrambling is applied to pixels only but it can be extended to blocks of the image also. If the scrambling is performed on both pixels and blocks the robustness and security of the image can be improved. Arnold scrambling for pixels can be applied to any image of any size. But to apply Arnold scrambling to an image which is divided into blocks, the image size should be of order M×M. If the size of the image is not M×M, it can be made M×M by adding zeros to the image which is called as padding. Arnold transform is widely used in image stenography, authentication, tamper detection, self-recovery and image cryptography algorithms. In all these cases, Arnold transform is used as a scrambling step in which the number of iterations is used as a key. Arnold transform for pixel scrambling is used in most of the applications and hence provides one key for the security. This paper proposes Arnold transform for pixel scrambling and block scrambling. The coordinates of the pixels are scrambled first which is followed by the coordinates of the blocks and thus providing two levels of security for scrambling. If the first level descrambling is successful, then only the second level descrambling can be carried out. This increases the complexity of malicious and unauthorized descrambling of images.

This paper proposes a two level image scrambling to increase the robustness of Arnold transform. First, the plain image is divided into blocks and each block is assigned a coordinate. The block coordinates can be transformed through Arnold scrambling. Hence, each block of the image will get a new coordinate and gets scrambled. Once the blocks are scrambled and arranged as an image, pixel scrambling can be carried out to scramble all the pixels in the image. This two level can also be implemented by doing pixel scrambling first which is followed by block scrambling. This paper analyses the two methods and the performance is analysed by Correlation coefficient (CC), number of pixel change rate (NPCR) and unified average changing intensity (UACI).

This paper is organized as follows. Section 2 deals with conventional and the proposed two level Arnold pixel scrambling which is followed by experimental results and analysis in section 3. Conclusion is given in section 4.

II. ARNOLD TRANSFORM

Arnold scrambling transforms the position of a pixel from (x, y) to a new position (s, t) [7]. The position of pixels changes from one point to another [7,9,10 & 11] based on the equation (1)

$$\begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\text{mod } 1) \tag{1}$$

In order to apply the transformation to a digital image the term mod 1 can be replaced by mod N where N is the size of the digital image [12].

$$\begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (\text{mod N}) \tag{2}$$

The transformation matrix of Arnold scrambling is a mod2 transformation matrix which is given by A. (x, y) T is the input and (s, t) T is the output. Considering the feedback and iterative process, the equation (2) can be written as follows

$$P_{xy}^{n+1} = A\, P_{xy}^{n} \ (\text{mod N}) \tag{3}$$
$$P_{xy}^{n} = (x, y)^T$$

Where n specifies the number of iterations i.e., n=0, 1, 2, … After applying scrambling to an image, the scrambled image will be different from the original image. Apart from being simple and easy, Arnold scrambling has the advantage of periodicity.

Inverse Arnold transform [10] can be given by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} (mod\ N) \tag{4}$$

III. TWO LEVEL ARNOLD SCRAMBLING

This paper proposes a two level image scrambling to increase the robustness of Arnold transform. First, the plain image is divided into blocks and each block is assigned a coordinate. The block coordinates can be transformed through Arnold scrambling. Hence, each block of the image will get a new coordinate and gets scrambled. Once the blocks are scrambled and arranged as an image, pixel scrambling can be carried out to scramble all the pixels in the image. This two level algorithm can also be implemented by doing pixel scrambling first which is followed by block scrambling.

A. Block Scrambling

In block scrambling, image is divided into M×M blocks and each block is assigned a coordinate {m,n} according to their spatial orientation. For an image of size 512×512, image can be divided into 64×64 blocks. Hence, there are spatial coordinates in the set of {(1,1), (1,2),…,(1,8),…,(8,1),(8,2),…,(8,8)}. Arnold scrambling is applied to the coordinates of blocks and hence each coordinate is assigned a new coordinate as given by equation (2). This paper proposes a block scrambling method for the blocks with same spatial resolution. This is possible only when the input image I(x,y) is of spatial resolution $2^n \times 2^n$; where n=1,2,3,…,N. Scrambling of blocks through Arnold transform is given as follows.

$$\begin{bmatrix} \{B(x_i)\} \\ \{B(y_i)\} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} \{B(x_{i-1})\} \\ \{B(y_{i-1})\} \end{bmatrix} \pmod{M} \quad (5)$$

Where $B(x_i,y_i)$ is the coordinate of the block of an image. M is number of rows or number of columns of all the blocks.

B. Illustration of Arnold block scrambling

Fig. 1 Illustration of Arnold block scrambling

First, the image is divided into blocks. For example, a 512 x 512 image is converted into blocks of size 128 x 128. Then the image will be divided into 16 blocks as shown in Fig. When Arnold scrambling is applied to blocks then the positions of blocks will get shifted. Let us consider block6 whose original position is at (2, 2) but after scrambling the position of block6 is shifted to new position i.e., (4, 3) for one iteration. For further iterations the input will be the output of previous iteration.

C. Scrambling Process

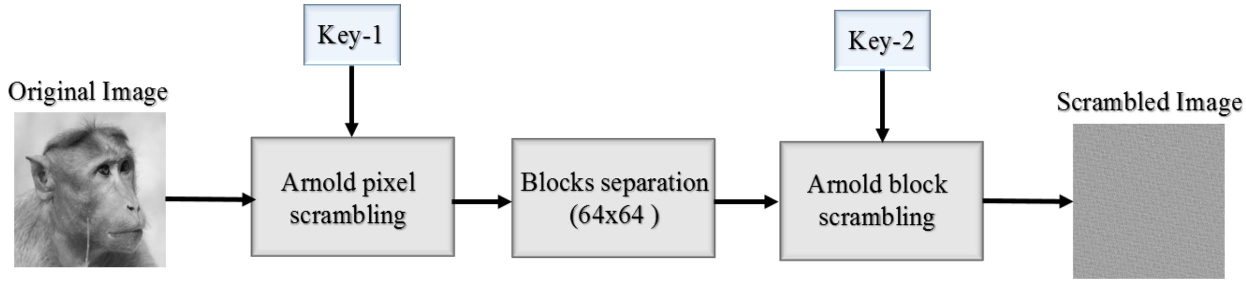

Fig. 2 Scrambling process

In the scrambling process, the original image is scrambled in two levels as shown in Fig. 2. In the first level the image is subjected to pixel Arnold scrambling with a specific number of iterations. The information related to number of iterations of level1 will be in key1 (K1). The scrambled image is divided into 8 x 8 blocks. In the second level the divided image is subjected to block Arnold scrambling with other specified number of iterations. The information of number of iterations of second level scrambling will be in key2 (K2). The image obtained after the second level scrambling is the image that will be transmitted.

D. Descrambling Process

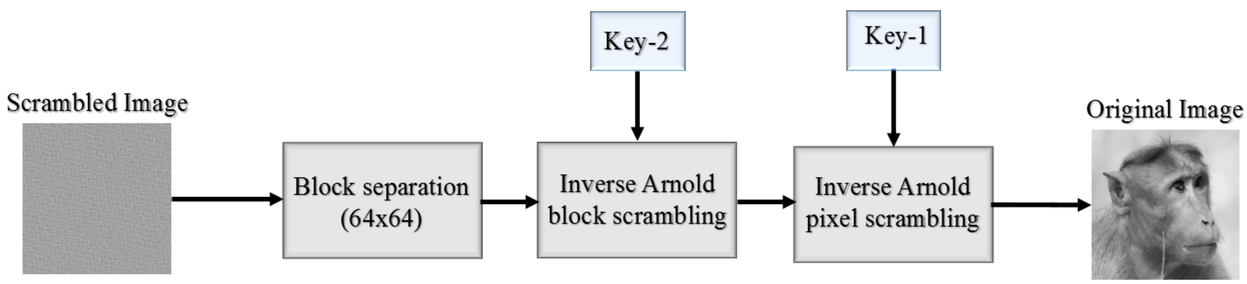

Fig. 3 Descrambling process

Same as scrambling, the original image will be extracted in two levels as shown in Fig 3. The scrambled image is divided into blocks and inverse block Arnold scrambling is applied based on key2. In the second level, the obtained image after the inverse block Arnold scrambling will be subjected to pixel level inverse Arnold scrambling based on key1. The obtained image after the second level is the original image.

E. Algorithm

The scrambling and descrambling steps for pixel & block two level Arnold scrambling are stated below.

Scrambling steps

1. Arnold pixel scrambling with certain number of iterations (key1) as given in equation (1) is applied to the input image I(x,y) and denoted as Ip(x,y).
2. Ip(x,y) is divided into blocks of size 64×64.
3. Each block is arranged in a matrix format and assigned a spatial coordinate (m, n).
4. Arnold scrambling with certain number of iterations (key2) is applied to the coordinates of the blocks and the blocks are rearranged with the new coordinate as an image.
5. The final two level scrambled image I'(x,y) is obtained.

Descrambling steps

1. I'(x,y) is divided into blocks as done in the scrambling
2. Blocks with the coordinates are descrambled and all the blocks are arranged in an image format.
3. Arnold descrambling is applied to the output image of block descrambling image to get the original image I(x,y).

IV. Experimental Results and Analysis

The proposed two level pixel scrambling is carried out in two ways. In the first method, block level scrambling applied first which is followed by pixel level scrambling. In the second method, pixel scrambling is followed by block scrambling. Images with spatial resolution of 512×512 or 2n×2n; where n=4,5,6,7,8,… are considered for experiments and block separation is also carried out with the block size of 2m×2m; where m=4,5,6,7,8,… and m < n. Images with other spatial resolution can also be considered with proper padding to carry out block processing.

A. Performance metrics

Performance of the two level Arnold scrambling is analysed by the metrics such as CC, NPCR and UACI.

Correlation Coefficient (CC):

The correlation coefficient analyzes the relationship between two normally distributed random variables. It is evaluated through covariance and variance of the plain-scrambled images [13, 14]. The covariance between a plain image (Ip), and scrambled image (Is) is given by cov ($\mathrm{Ip}, \mathrm{I_s}$) and variance is represented as $\sigma_{\mathrm{I_p}}$ and $\sigma_{\mathrm{I_s}}$. The correlation coefficient is given by

$$\mathrm{r_{xy}} = \frac{|\mathrm{cov}(\mathrm{I_p}, \mathrm{I_e})|}{\sqrt{\sigma_{\mathrm{I_p}}}\sqrt{\sigma_{\mathrm{I_e}}}} \tag{5}$$

NPCR

NPCR [15] measures the different pixel numbers between the scrambled image and the plain image.

$$\mathrm{NPCR}\ \% = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{P(i,j)}{MN} \times 100 \tag{6}$$

Where $P(i,j)=0$ if $I_p(i,j)=I_s(i,j)$ and $P(i,j)=1$ if $I_p(i,j) \neq I_s(i,j)$

For this experiment, original image is scrambled by the proposed two level Arnold scrambling algorithm. The scrambled images are compared with the original image and NPCR value is evaluated. If NPCR is high that denotes good performance of the scrambling algorithm.

UACI

UACI [15] evaluates the average intensity difference between the plain and scrambled images. High UACI represents good performance of the scrambling algorithm.

$$\mathrm{UACI}\ \% = \frac{1}{MN}\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{I_P(i,j) - I_S(i,j)}{255}\right] \times 100 \tag{7}$$

B. Selection of Keys for two level scrambling

Scrambling and descrambling of blocks and pixels of images are effectively carried out with key1 and key2. The keys are number of iterations for Arnold scrambling that is based on spatial resolution of the image and coordinates assigned to the blocks. For different spatial resolution, number of iterations are listed out in Table 1. Keys for two level scrambling can be selected according to the number of iterations. For descrambling, number of iterations is given by

$$\begin{aligned} &\mathit{Number\ of\ iterations\ for\ descrambling} \\ &\quad = \{\mathit{maximum\ number\ of\ iterations\ for\ the\ spatial\ resolution}\ (2^n \times 2^n)\} \\ &\quad - \mathit{Key\ selected\ for\ scrambling} \end{aligned}$$

Table 1. Number of iterations for different sizes of an image

Size of Image	Iterations
4×4	3
8×8	6
16×16	12
32×32	24
64×64	48
128×128	96
256×256	192
512×512	384

C. Block-pixel scrambling

This method is accomplished by block scrambling followed by pixel scrambling. The reverse is done for descrambling. Block separation is a trivial task in which the spatial resolution of the blocks can be $2^n \times 2^n$. The spatial resolution of the blocks can be decided by the user and hence helps to improve the robustness of the method. Once the block separation is carried out with certain number of iterations (key1), all the blocks are arranged to get the block scrambled image. Arnold pixel scrambling is applied to this scrambled image with certain number of iteration (key2) to get the final scrambled image.

Table 2. Performance analysis of Block-pixel scrambling

	Boat	Barbara	Baboon	Airplane	Zelda
CC					
Block scrambling	0.957443	0.944454	0.743324	0.948021	0.976404
Pixel Scrambling	0.21381	0.14124	0.09179	0.07031	0.11767
NPCR					
Block scrambling	97.5399	97.826	97.81647	96.96465	97.74857
Pixel Scrambling	99.02649	99.4606	99.28589	98.54507	99.29657
UACI					
Block scrambling	19.6261	23.395	19.03114	18.91619	17.14758
Pixel Scrambling	19.52677	24.6594	18.94954	18.31416	18.26919

Descrambling is done with the reverse of the above mentioned steps and keys. Pixel descrambling with the certain number of iterations is applied to the scrambled image that delivers another scrambled image. From this scrambled image, one malicious user can guess some details about block scrambling that obstructs the robustness objective of this method. This is illustrated in Fig. 4. From Table 2, it is observed that block scrambling does not scramble the image completely. With the pixel scrambling step, robustness is improved.

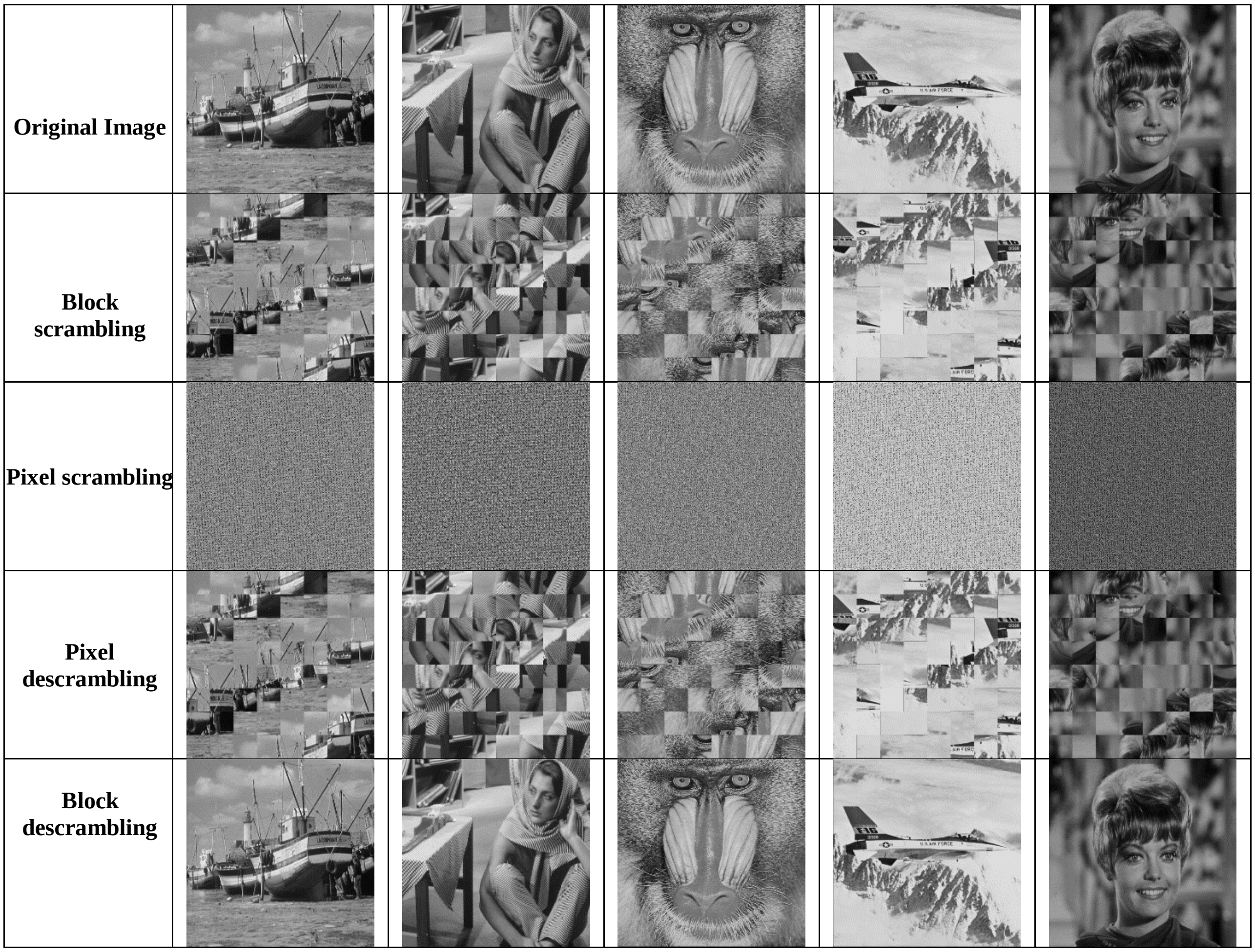

Fig. 4 Block-pixel scrambling plain and scrambled images

D. Pixel-block scrambling

This method deals with pixel scrambling followed by block scrambling with key1 and key2. Descrambling is done vice versa. In this method, block level descrambling should be carried out first with two factors, size of the blocks and number of iterations (key2). If both factors are correctly estimated then only the malicious user can do successful block level descrambling. Even after successful descrambling, the malicious user will get another scrambled image. Pixel level descrambling with correct key (key1) should be carried out to get back the original image. Compared to method1, the first descrambling step in this method does not reveal any information about the second descrambling step. This increases the robustness of this two level scrambling method. This is illustrated in Fig. 5. By analysing the descrambled images in Fig. 4 and Fig. 5, it can be observed that pixel scrambling followed by block scrambling provides more security to the plain images. From Table 3. It is revealed that the first pixel scrambling step itself, the plain image is scrambled. The scrambled image is again scrambled block wise. For malicious users, estimating the size of the blocks for descrambling is complex task. If this step is completed correctly, then only the pixel descrambling can be carried out to get the plain image.

Table 3. Performance analysis of pixel - Block scrambling

	Boat	Barbara	Baboon	Airplane	Zelda
CC					
Pixel Scrambling	0.13751	0.11479	0.047531	0.13927	0.04106
Block scrambling	0.21431	0.1397	0.091503	0.07029	0.11699

NPCR					
Pixel Scrambling	99.05396	99.46938	99.3309	98.57788	99.31679
Block scrambling	99.01886	99.43352	99.30725	98.53745	99.27635
UACI					
Pixel Scrambling	19.56608	24.66429	18.9602	18.35283	18.30821
Block scrambling	19.53834	24.67745	18.95002	18.32674	18.27697

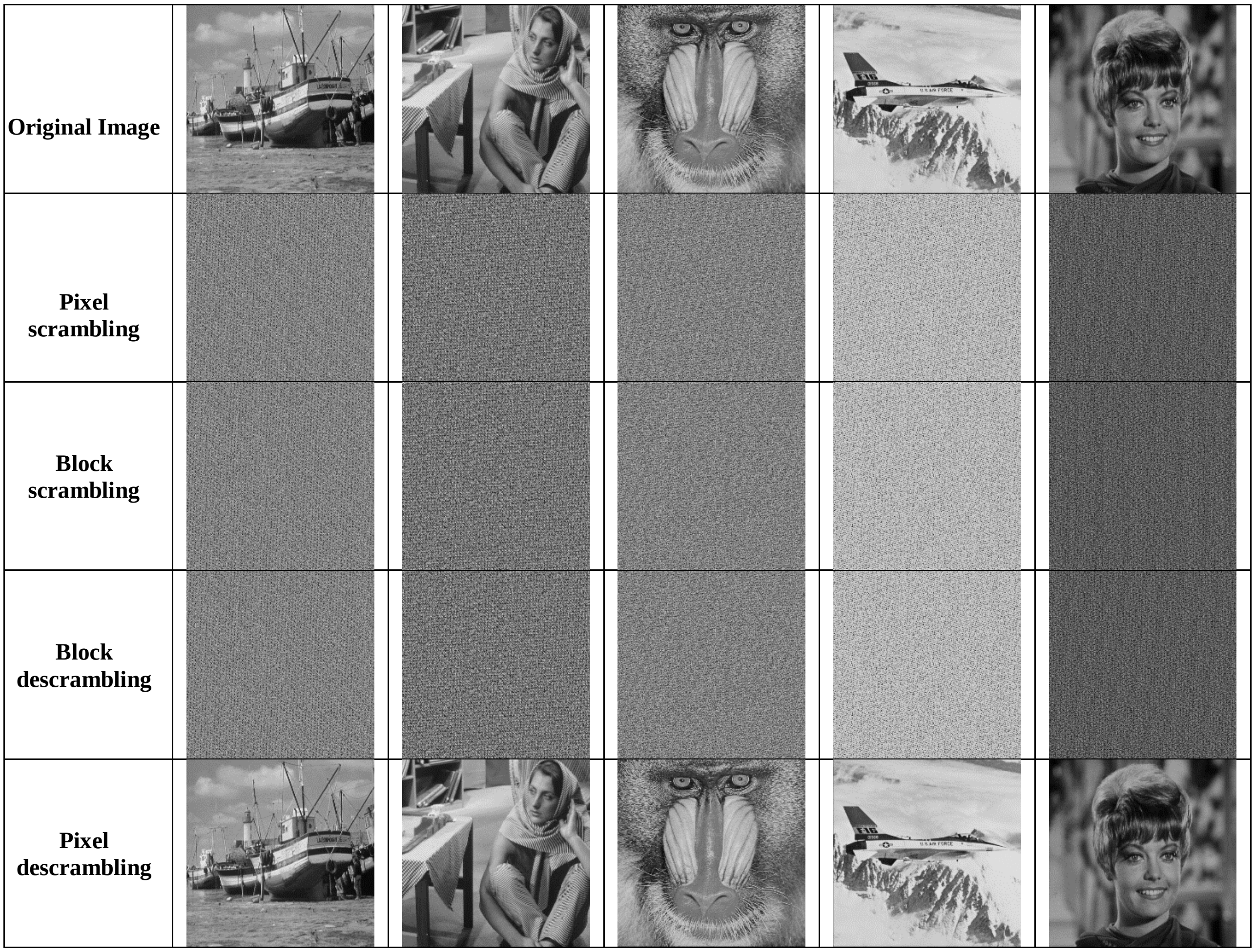

Fig. 5 Pixel-block scrambling plain and scrambled images

V. CONCLUSION

This work proposes a two level Arnold scrambling method which increases the security of scrambling technique. Conventional Arnold scrambling has one level of scrambling in which the original image can be obtained by applying descrambling iteratively. This paper introduces Arnold scrambling for pixels and blocks of the plain image. This pixel and block scrambling can be applied alternatively, but pixel scrambling followed by block scrambling delivers more robustness compared to block scrambling followed by pixel scrambling. This is substantiated by both subjective and objective analysis. Performance of this two level Arnold scrambling is also analysed by Correlation coefficient, number of pixel change rate and unified average changing intensity. This two level Arnold scrambling can be used an image scrambling step in image watermarking, image tamper detection & recovery and image authentication algorithms.

REFERENCES

[1]. Matsuo T and Kurosawa K (2004) On parallel hash functions based on block-ciphers, IEICE Trans. Fundam. Electron. Commun. Comput. Sci.,vol. E87-A, no. 1, pp. 67_74.

[2]. Li N, Du W, and Boneh A D (2005) Oblivious signature-based envelope, Distrib. Comput., vol. 17, no. 4, pp. 293_302, 2005.

[3]. Yeung M M and Mintzer F (1997) An invisible watermarking technique for image verification, in Proc. Int. Conf. Image Process., vol. 2. Oct. 1997,pp. 680_683

[4]. Jan Z et al (2017) A review on automated diagnosis of malaria parasite in microscopic blood smears images, Multimedia Tools Appl., pp. 1_26. Available: https://doi.org/10.1007/s11042-017-4495-2

[5]. Muhammad K, Sajjad M, Mehmood I, Rho S and Baik S W (2018) Image steganography using uncorrelated color space and its application for security of visual contents in online social networks, Future Generation Computer Systems, Volume 86. Pages 951-960

[6]. Abdulaziz shehab et al. (2018) Secure and robust watermarking scheme for medical images, IEEE Access, Vol. 6, pp. 10269 – 10278.

[7]. Min Li et al. (2013) Arnold Transform Based Image Scrambling Method, 3rd International Conference on Multimedia Technology（ICMT 2013), pp. 1309-1316.

[8]. Wu L, Zhang J, Deng W and He D (2009) Arnold transformation algorithm and anti-Arnold transformation algorithm. in Proc. 1st IEEE Int. Conf. Inf. Sci. Eng., Dec. 2009, pp. 1164_1167.

[9]. Vaiopoulos, A. D Developing Matlab scripts for image analysis and quality assessment. Proceedings of SPIE - The International Society for Optical Engineering 8181:6, DOI: 10.1117/12.897806

[10]. Siva Shankar S (2016) Data Hiding In Encrypted Images Using Arnold Transform. ICTACT Journal On Image And Video Processing, Volume: 07, Issue: 01.

[11]. Sukhwinder Kaur (2017) Arnold transform based Security Enhancement using Digital Image Watermarking with Complex Wavelet Transform. International Journal of Electronics Engineering Research. ISSN 0975-6450 Volume 9, Number 5 pp. 677-693.

[12]. Sujatha S S (2010) Feature Based Watermarking Algorithm by Adopting Arnold Transform. V. V Das, R. Vijaykumar et al. (Eds.): ICT 2010, CCIS 101, pp. 78–82, 2010.

[13]. Hung I. Hsiao and Junghsi Lee (2015) Fingerprint image cryptography based on multiple chaotic systems. Signal Processing, 113, pp. 169–181.

[14]. Tieyu Zhao, Qiwen Ran and Yingying Chi (2015) Image encryption based on nonlinear encryption system and public-key cryptography. Optics Communications, 338, pp. 64–72.

[15]. Saeed Bahrami and Majid Naderi (2012) Image encryption using a lightweight stream encryption algorithm. Advances in Multimedia, Article ID: 767364, pp. 1-8.