International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.6, No.5 (2013), pp.275-290
http://dx.doi.org/10.14257/ijsip.2013.6.5.25

A New Fast and Simple Image Encryption Algorithm Using Scan Patterns and XOR

Reza Moradi Rad, Abdolrahman Attar, and Reza Ebrahimi Atani
Department of Computer Engineering, University of Guilan,Rasht, Iran
reza.moradirad@gmail.com, a.attar.q@gmail.com, rebrahimi@guilan.ac.ir

Abstract

In this paper, a new simple and fast algorithm for image encryption is proposed. It exploits the scan patterns and function XOR in three standalone steps. The decryption procedure is similar to that of the encryption but in the reversed order. We implement and test the proposed algorithm using different sample image inputs and our experimental result and security analysis indicates the robustness and advantages of the new proposed algorithm. Using this algorithm it is possible to reproduce the original image with no loss of information for the encryption and decryption process. Algorithm is fast and simple enough to be comparable to other recent approaches, and it has passed all the security requirements and it is fast and secure to be used in very broad range of industrial applications.

Keywords: *Visual cryptography, Image Encryption, Security Analysis. Scan Patterns, Histogram, Entropy, Correlation*

1. Introduction

With the noteworthy advance in visual sensing and image analysis techniques, image data are used in many sensitive applications. This digital data is vulnerable to illegal copying and distribution, thus requiring confidentiality at different stages of data archival, transmission or distribution [1], thus many image data security solutions have revealed up to now. Image Encryption is one of the most important and efficient common tools.

Image Encryption is the process of transforming information (referred to as plain-image) using an algorithm (called cipher) to make it unintelligible to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted image (referred to as cipher-image). Decryption is the process of converting cipher-image back into its original form, so it can be perceived.

Image Encryption can be used to protect images at rest, such as images on computers and storage devices in a situation which personal records being exposed through loss or theft of laptops or backup drives. Image encrypting at rest helps protect them from being uncovered and shared. Encryption is also used to protect data in transit, for example data being transferred via networks (*e.g.*, the Internet, e-commerce), mobile telephones, wireless systems, Bluetooth devices and so on.

In order to facilitate secret communication, image encryption has found a significant place in both public and private services such as military surveillance, satellite information systems, health-care, meteorology, confidential video conferencing, online personal photograph album, internet banking transactions, multimedia systems, telemedicine, and medical imaging systems [2, 6]. These applications rely on wired/wireless communication channels that are bandwidth-limited and open in nature [7].

ISSN: 2005-4254 IJSIP

Most of the algorithms specifically designed to encrypt digital images are proposed in the mid-1990s [8]. Each image encryption algorithms have different characteristics, and there is no single encryption algorithm satisfies all of system requirements. Each of them has its strength and weakness in terms of security level, speed, and similarity between original image and decrypted image.

Some of algorithms use the traditional text based cryptosystems to encrypt images directly. Due to existing difference between image and text, it is not a good idea. One problem is that the image size is almost always much greater than that of text. Therefore, the traditional cryptosystems need much time to directly encrypt the image data. The other problem is that the decrypted text must be equal to the original text. However, this requirement is not necessary for image data. Due to the characteristic of human perception, a decrypted image containing small distortion is usually acceptable [5, 8, 9]. Additionally Application of text encryption techniques on images may not completely hide all the image features and hence proper encryption of images cannot be achieved [2, 10].

Throng of algorithm have carried out on the image encryption which they have they own pros and cons. Here are some of them. Paper [11] has proposed a chaotic key based algorithm (CKBA) to change the pixel values of the plain-image. Paper [12] has proposed a random combinational image encryption approach with bit, pixel and block permutations. Paper [13] has presented a new image encryption scheme, in which shuffling the positions and changing the grey values of image pixels are combined to confuse the relationship between the cipher image and the plain image. Paper [14] proposed an image encryption by using Fractional Fourier Transform (FRFT) and JigSaw Transform (JST) in image bit planes. Paper [15] proposed image and video encryption using SCAN patterns. The image encryption is performed by SCAN-based permutation of pixels and a substitution rule which together form an iterated product cipher. Paper [4] proposed new schemes which add compression capability to the mirror-like image encryption MIE and Visual Cryptography VC algorithms to improve these algorithms. Paper [16] proposed a technique to encrypt an image for secure transmission using the digital signature of the image. Digital signatures enable the recipient of a message to authenticate the sender of a message and verify that the message is intact. Paper [17] has proposed two methods for the encryption of an image; selective encryption and multiple selective encryptions. Paper [18] has presented a new methodology, which performs both lossless compression and encryption of binary and gray-scale images.

In this paper, we propose a new fast algorithm to be used as an encryption algorithm which is based on predefined scan patterns for shuffling the pixels of image and simple basic encryption function XOR.

First, the original image is divided into a specific number of blocks which are shuffled by scan patterns to build a newly transformed image. Second, for reaching more entropy, pixels of each block are repositioned by scan patterns again. Third, the new generated transformed image is then exploits simple function XOR for encryption. It is symmetric key algorithms which use a single, shared secret key. The same 128-bit key is used for both encrypting and decrypting side. We believe that by proposing this simple and fast encryption and decryption algorithm, it will help to reduce the relationship among image elements by increasing the entropy value of the encrypted images as well as lowering the correlation in regard to reducing time.

The rest of this paper is organized as follows. The Section 2 describes Scan language. In Section 3, the proposed algorithm is explained. Simulation results and Security analysis are provided in Section 4. Finally, a conclusion is drawn in Section 5.

2. Overview of Scan Language

There are strong correlations between values of neighboring pixels in almost all natural images, so that the value of any given pixel can be reasonably predicted from the values of its neighbors [8, 19-21, 24, 25]. In order to dissipate the high correlation among pixels and increase the entropy value, we exploit scan patterns that generated by the SCAN methodology to shuffle the position of blocks and then pixels of blocks and at the last step function XOR is used. For more clarification, we commence the explanation by brief definition of Scan language and its concept.

SCAN language is an Image preprocessing language, devoted to generate a family of 2D scanning patterns (or fractals). The scanning path of the image is a random code form, and by specifying the pixels sequence along the scanning path. Note that scanning path of an image is simply an order in which each pixel of the image is accessed exactly once. Such the encryption also involves the specification of set secret scanning paths. Therefore, the encryption needs a methodology to specify and generate a larger number of wide varieties of scanning paths effectively.

A scanning of a two dimensional array $P_{m\times n} = \{p(i, j) : 1 \le i \le m, 1 \le j \le n\}$ is a bijective function from $P_{m\times n}$ to the set $\{1, 2, ..., mn-1, mn\}$. In other word, a scanning of a two dimensional array is an order in which each element of the array is accessed exactly once. In this paper the terms scanning, scanning paths, scan pattern, and scan words are used interchangeably. Note that an $n \times n$ array has $(n \times n)!$ scanning.

The SCAN is a formal language based two-dimensional spatial accessing methodology which can represent and generate a large number of wide varieties of scanning paths. The SCAN is a family of formal languages such as Simple SCAN, Extended SCAN, and Generalized SCAN, each of which can represent and generate a specific set of scanning paths. Each SCAN language is defined by a grammar and each language has a set of basic scan patterns, a set of transformation of scan patterns and a set of rules to recursively compose simple scan patterns to obtain complex scan patterns. Note that this set of basic scan patterns can be extended or reduced as needed by a specific application [3, 19, 20, 22, and 23].

We consider 8 following scan patterns for a block of size 8x8 (it is obvious they can be used in a block of size $n \times n$) as shown in Figure 1. They are indexed from 0 to 7 respectively.

3. Proposed Method

The algorithm is mainly based on 8 scan patterns depicted in previous section. Following 3 steps formed the new image encryption. The 128-bit key is gradually explained and formed during each step. Please note that we limit and implement our system for 512×512 gray scale images.

1. Rearrangement of Blocks using Scan Patterns

In first step, the original image divided into 8x8 blocks (64 blocks in each column and row). Blocks are repositioned by the needed number of predefined scanning algorithm (Figure 1) to shuffle the image structure. Obviously, exploiting desired number of different 8 scanning algorithms make the result image being more confusion. For example consider a situation that three scanning algorithm including 0, 5, and 7 is run subsequently and each of them for 0 to 7 times (*i.e.*, it can be not scanned and can be scanned for several times but at most in 7 times). Of course, the result image is more shuffled than a situation that just scan

pattern 5 is used for 1 time. For more illustration, Figure 2 shows implementation of 3 scan patterns on an arbitrary image.

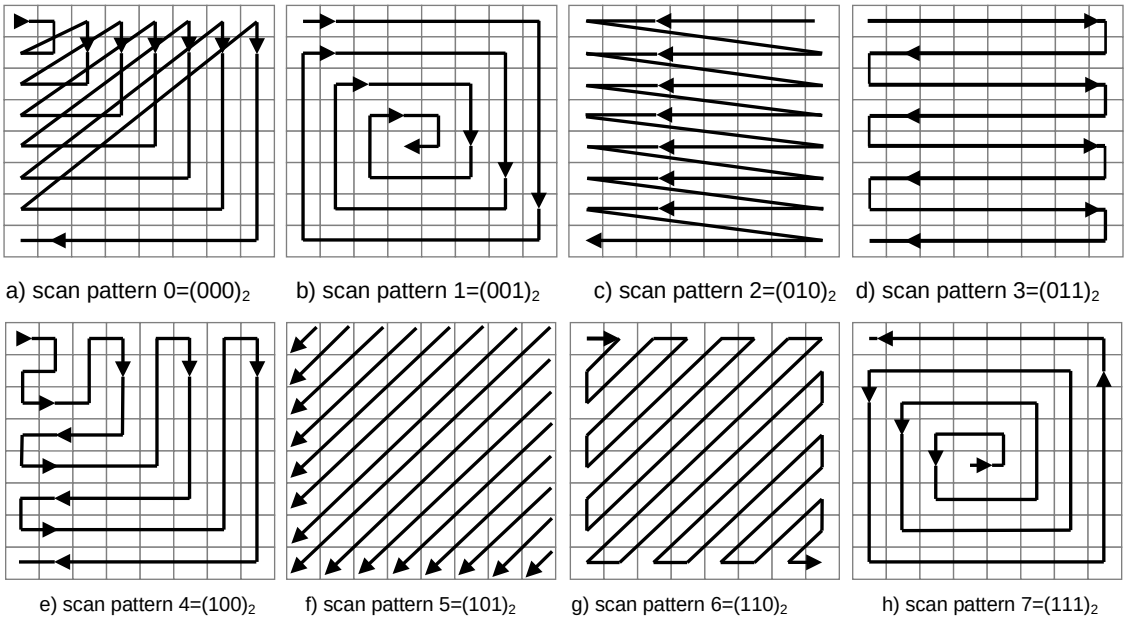


Figure 1. Different Scan Patterns Indexed from 0=$(000)_2$ to 7=$(111)_2$

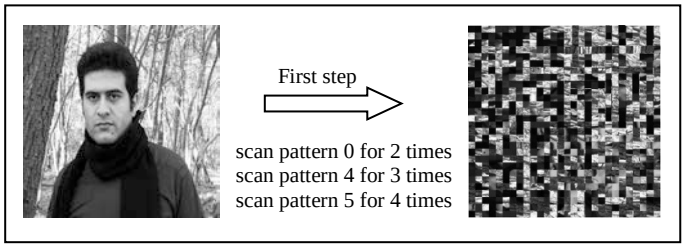
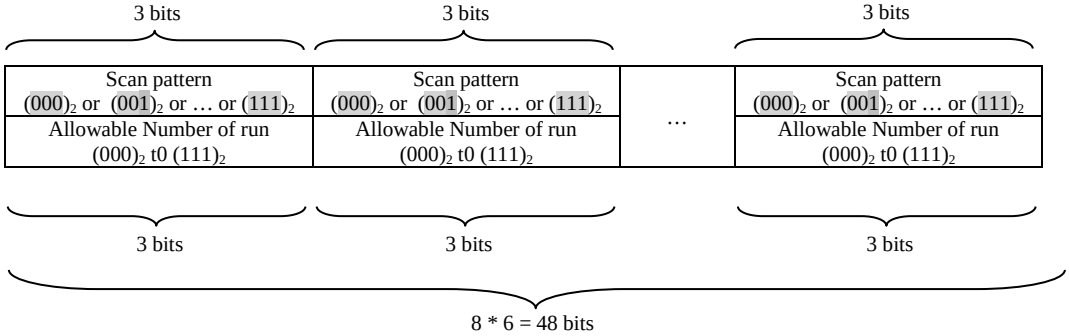


Figure 2. Whole 8*8 Blocks of Original Image Shuffled by 3 Scan Patterns for Desired Times

Formation of key (48 bits): 8 different Scan patterns need 3 bits to be indexed. Each scan pattern can be run from 0 to 7 times, hence it needs 3 bits. Each scan pattern conjunction with number of its running needs 3+3=6 bits. Therefore 8(number of scan patterns) * 6 = 48 bits is enough for addressing desired repositioning.



For example two following partial key indicate that image repositioned by scan patterns as following:

1) Partial key include all scan patterns.

$02152637415364 71 = (000010 \quad 001101 \quad 010110 \quad 011000 \quad 100001 \quad 101011 \quad 110100 \quad 111001)_2$

Scan pattern $(000)_2$ for 2 times.

Scan pattern $(001)_2$ for 5 times.

Scan pattern $(010)_2$ for 6 times.

Scan pattern $(011)_2$ for 7 times.

Scan pattern $(100)_2$ for 1 time.

Scan pattern $(101)_2$ for 3 times.

Scan pattern $(110)_2$ for 4 times.

Scan pattern $(111)_2$ for 1 time.

2) Partial key include scan patterns 0, 4, and 6.

$03644601426364 01 = (000011 \quad 110100 \quad 100110 \quad 000001 \quad 100010 \quad 110011 \quad 110100 \quad 000001)_2$

Scan pattern $(000)_2$ for 3 times.

Scan pattern $(110)_2$ for 4 times.

Scan pattern $(100)_2$ for 6 times.

Scan pattern $(000)_2$ for 1 time.

Scan pattern $(100)_2$ for 2 time.

Scan pattern $(110)_2$ for 3 times.

Scan pattern $(110)_2$ for 4 times.

Scan pattern $(000)_2$ for 1 time.

2. Rearrangement of Pixels of each Block using Scan Patterns

By repositioning the blocks of image in previous step, tangible amount of correlation between different regions of original image has removed but pixels of each block are correlated yet. It is obvious that more shuffling cause more independency of pixels and make the concept of image impossible.

In second step, for more shuffling the transformed image obtaining in previous step, the rearrangement is run on the each block of image individually. To reaching that aim the scanning algorithms (Figure 1) are exploited again.

Each block is treated as an image so that it can be scanned for 0 to 7 times by different 8 scan patterns. So all pixels of blocks can rearrange several times, so a new order of scanning algorithm can be generated again, and shuffle the arrangement of pixels within each block more than before.

In these two steps, by considering the input image obtained in first step and then in second step, propitious scan patterns and number of run should be chosen intellectually to come over the image correlation and shuffle the image elaborately.

Formation of key (48 bits): The story about rest of key formation which is related to this step is same as previous step. Hence 48-bits are needed for addressing the desired order of rearrangement of pixels within blocks.

3. XOR all Blocks with 2 Arbitrary Blocks from the Rearrangement Image

In third step, when we make the image as shuffled and unperceivable as we could, the function XOR is exploited finally. We divide the obtained image in previous step into 4 parts (as shown in Figure 3), and index them from 0 to 3. Then XOR each two part with one of two arbitrary 8*8 blocks. For example, part 0 and 2 with one 8*8 block and part 1 and 3 with another 8*8 block.

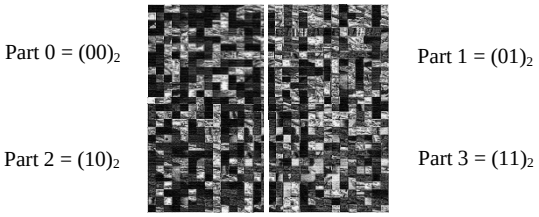
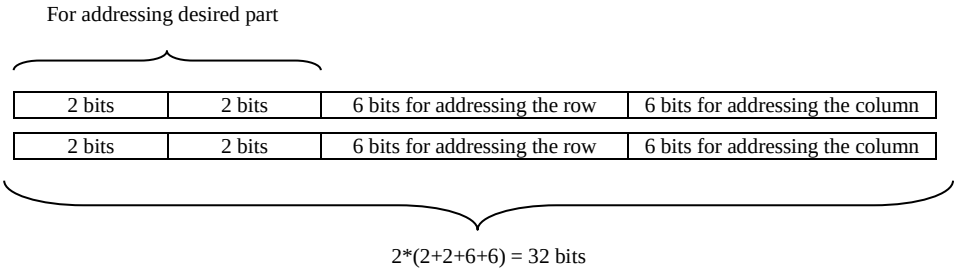


Figure 3. Obtained Image in Previous Step Divided into 4 Parts and Indexed from 0 to 3

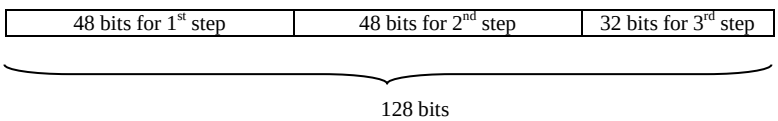
Formation of key (32 bits): For choosing and addressing the row (within 64 rows) and column (within 64 columns) of two arbitrary blocks it is needed to 24 bits. In order to determining which part should be XORed with one of following block 4*2-bits=8 bits is needed. Consequently 24+8=32 bits is used in this step as the rest of key.



For example, below partial key indicates that part 0 and 2 XORed with block at position of 23*45, and part 1 and 3 XORed with block at position of 12*30.

$$022345 = (00\quad 10\quad 010111\quad 101101)_2$$
$$131230 = (01\quad 11\quad 001100\quad 111110)_2$$

Finally the general key will be 128 bits:



4. Performance and Security Analysis

We have done several tests to check the security of the proposed cryptosystem. Statistical tests include histogram analysis and calculation of correlation coefficients of adjacent pixels. Security tests against differential attack include calculation of the NPCR and UACI, and information entropy evaluation. We have done our experimental analysis for the proposed encryption scheme on four gray level images "Lena", "Camera man", "Barbara", and "Pepper" with 512×512 pixels as the sample plain images. Their plain, cipher and decrypted images are shown in Figure 4.

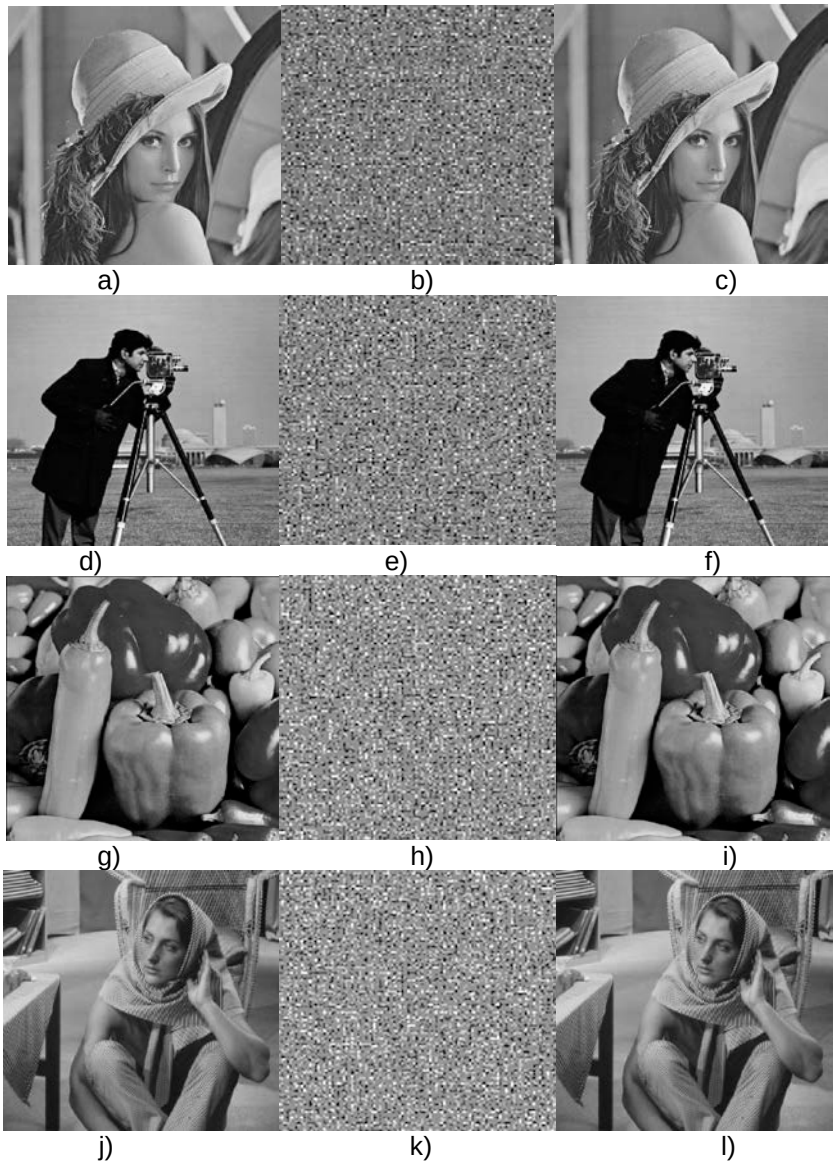


Figure 4. The Plain, Cipher, and Decrypted Images. (a) Plain-image Lena, (b) Cipher Image, (c) the Decrypted Image, (d) Plain-image Camera Man, (e) Cipher Image, (f) the Decrypted Image, (g) Plain-image Pepper, (h) Cipher Image, (i) the Decrypted Image, (j) Plain-image Barbara, (k) Cipher Image, (l) The Decrypted Image

4.1 Statistical Analysis

4.1.1 Histogram: Image histogram is a very important feature in image analysis. From Figure 6 it is obvious that the histograms of the encrypted image are nearly uniform and meaningfully different from the histograms of the original image. Hence it does not provide any clue to employ any statistical analysis attack on the encryption image. Histograms of the plain and the cipher images are depicted in Figure 5.

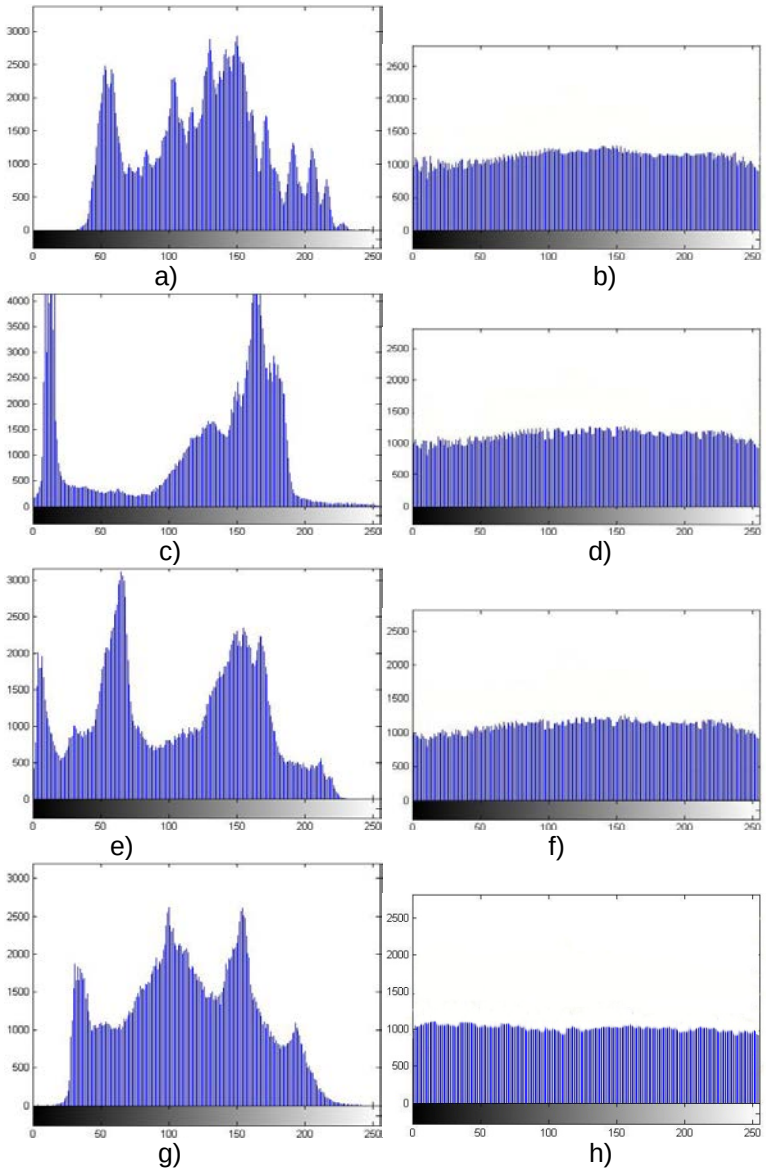


Figure 5. Histogram of Original Image and Cipher Image. (a) Histogram of the Plain-image Lena, (b) Histogram of the Cipher-image Lena, (c) Histogram of the Plain-image Camera Man (d) Histogram of the Cipher-image Camera Man, (e) Histogram of the Cipher-image Pepper (f) Histogram of the Cipher-image Pepper, (g) Histogram of the Cipher-image Barbara (h) Histogram of the Cipher-image Barbara

4.1.2. Correlation Analysis of Two Adjacent Pixels: We have analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, two diagonally adjacent pixels, and two opposite diagonally adjacent pixels in an image. 1000 pairs of two adjacent (in vertical, horizontal, diagonal, and opposite diagonal direction) pixels from plain-image and ciphered image were randomly selected and the correlation coefficients were calculated by using the following equations:

$$r_{xy} = \frac{|Cov(x,y)|}{\sqrt{D(x)}\ \sqrt{D(y)}}, \tag{1}$$

$$E = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{2}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{3}$$

$$Cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(x)) \tag{4}$$

The x and y represent gray level values of two adjacent pixels. Figure 6 is the horizontal, vertical, diagonal, and opposite diagonal relevance of adjacent elements in image before and after encryption. Table 1 shows the results of correlation analysis. Table 1 and Figure 6 show significant reduction in relevance of adjacent elements.

Table 1. Correlation Coefficients of Plain Image and Ciphered Image

Image		Horizontal	Vertical	Diagonal	Opposite-Diagonal
Lena	Plain	0.9632	0.9451	0.9342	0.9601
	Ciphered	-0.0009	-0.0101	-0.0023	-0.0158
Camera man	Plain	0.9877	0.9783	0.9654	0.9812
	Ciphered	0.0001	-0.0015	-0.0001	-0.0011
Pepper	Plain	0.9511	0.9659	0.9734	0.9649
	Ciphered	-0.0023	-0.1420	0.0020	-0.0883
Barbara	Plain	0.9776	0.9713	0.9801	0.9699
	Ciphered	-0.0886	-0.0349	0.0021	0.0006

Above table indicates the performance analysis of the proposed method using Lena, Camera man, Pepper, and Barbara images correlation coefficients, ranging from '1' highly correlated to '-1' highly uncorrelated, of pairs of adjacent pixels in different directions. These coefficients ensure the two considered images are statistically independent.

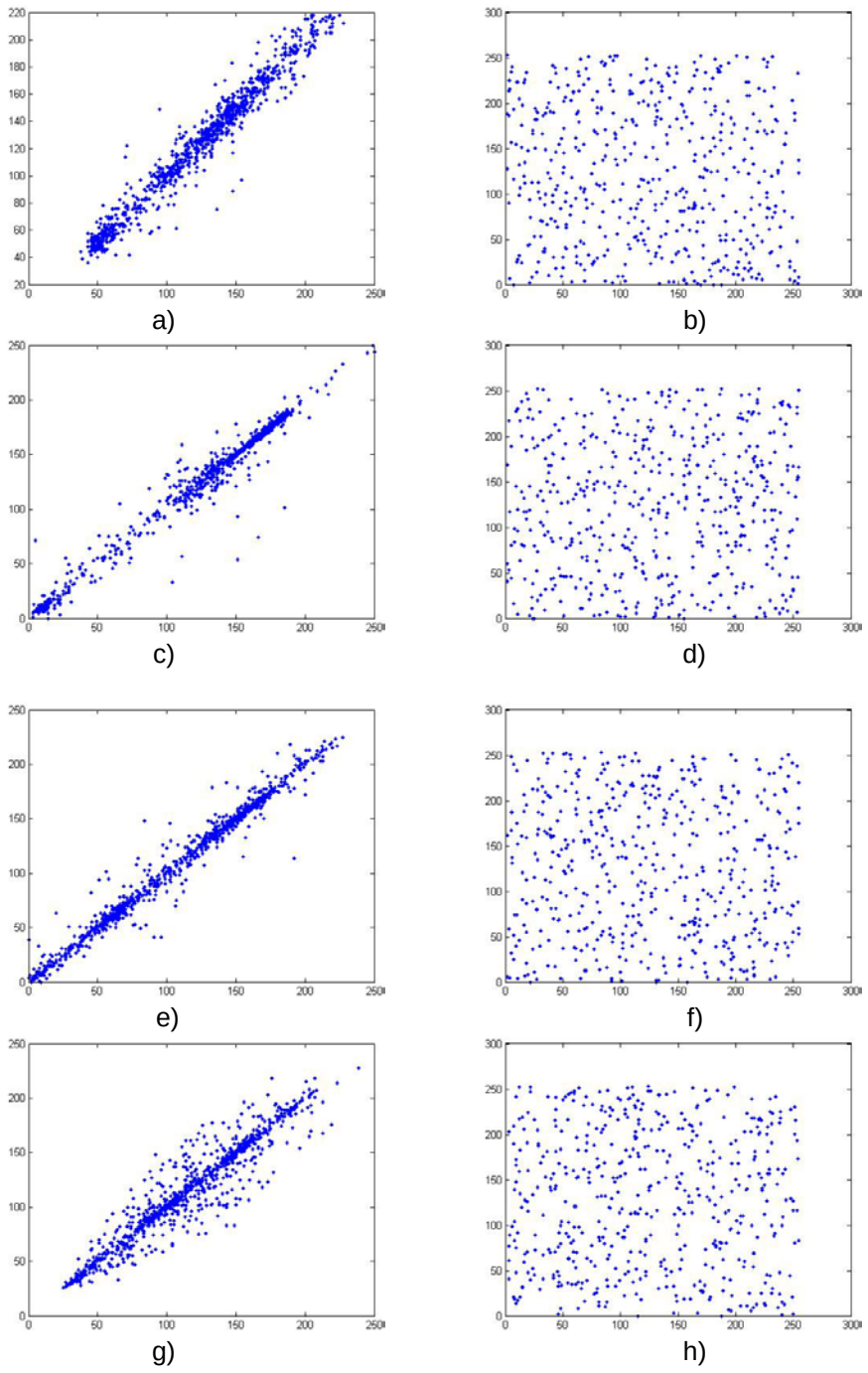


Figure 6. Correlation Analysis of Two Adjacent Pixels in Original Images and Images Obtained using the Proposed Scheme. (a) Plain-image Lena, (b) Cipher-image Lena, (c) Plain-image Camera Man, (d) Cipher-image Camera Man, (e) Plain-image Pepper, (f) Cipher-image Pepper, (g) Plain-image Barbara, (h) Cipher-image Barbara

4.2. Sensibility Analysis

4.2.1. Differential Attack: As a general requirement for all the image encryption schemes, the encrypted image should be greatly different from its original form. In order to test the influence of changing a single pixel in the original image on the encrypted image, we have measured the number of pixels change rate by calculating the number of pixel change rate (NPCR) defined in Eq. (5) and the unified average changing intensity (UACI) defined in Eq. (6) for the two encrypted images:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{5}$$

and

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\% \tag{6}$$

Where W and H are the width and height of the encrypted image. We use two encrypted images $C1$ and $C2$, whose corresponding original images have only one-pixel difference. We also define a two-dimensional array D, which has the same size as $C1$ and $C2$. If $C_1(i,j) = C_2(i,j)$, then $D(i,j) = 0$ otherwise $D(i,j) = 1$. It is clear that 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 grayscale images of size 512×512 to measure the influence of one pixel change on the original image. In the test procedure the plain image is encrypted. After that a pixel of the image is chosen randomly and then changed. The changed image is encrypted using the same key and a new cipher image will be obtained. This procedure is performed 50 times for different images obtained from different key fed to the proposed algorithm. The resulting maximum, minimum, and average NPCR and UACI values of our proposed scheme are presented in Table 2. According to the values of NPCR and UACI, proposed algorithm can satisfy security requirements.

Table 2. The NPCR and UACI of Ciphered Images by Changing their Original Images One Pixel

		Lena	Camera man	Pepper	Barbara
NPCR (%)	Max	99.7725	99.6904	99.8979	99.6798
	Min	99.5412	99.5035	99.2601	98.9795
	Average	99.6135	99.6529	99.5987	99.6669
UACI (%)	Max	34.2785	34.1250	34.6712	34.6759
	Min	33.5899	33.6095	33.5934	33.6390
	Average	33.6548	33.6015	33.6001	33.6548

4.2.2. Information Entropy Analysis: It is known that the entropy H(S) of a message source S can be calculated by Eq. (7) Where $P(s_i)$ represents the probability of symbol s_i. The entropy is expressed in bits. If the source emits 2^8 symbols with equal probability, *i.e.*, $S = \{s_1, s_2, s_3, ..., s_{2^8}\}$, then the result of entropy is $H(s) = 8$, which corresponds to a true random source. Test results of proposed algorithm for the cipher images of four gray level images Lena, Camera man, Peppers, and Barbara are listed in Table 3. Results shows that

cipher images are close to a random source and the proposed algorithm is secure against the entropy attack.

$$H(S) = \sum_{S} p(s_i) \log_2 \frac{1}{p(s_i)} bits, \quad (7)$$

Table 3. Results of Information Entropy

	Lena	Camera man	Pepper	Barbara
Entropy	7.9971	7.9968	7.9981	7.9974

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

1) Original 512x512 gray scale Lena image is encrypted by using the following test key. The details of test key are clarified in Table 4.

"42356324011133125132617342340322032502121411"

Table 4. Details of Test Key

42	35	63	24	01	11	33	12	51	32	61	73	42	34	03	22	032502	121411
Run scan pattern 4 for 2 times	Run scan pattern 3 for 5 times	Run scan pattern 6 for 3 times	Run scan pattern 2 for 4 times	Run scan pattern 0 for 1 time	Run scan pattern 1 for 1 times	Run scan pattern 3 for 3 times	Run scan pattern 1 for 2 times	Run scan pattern 5 for 1 time	Run scan pattern 3 for 2 times	Run scan pattern 6 for 1 time	Run scan pattern 7 for 3 times	Run scan pattern 4 for 2 times	Run scan pattern 3 for 4 times	Run scan pattern 0 for 3 times	Run scan pattern 2 for 2 times	XOR block at position of 25x2 with part 0 and 3	XOR block at position of 14x11 with part 1 and 2
First Step								Second Step								Third Step	

2) Then, the original key after change becomes, say "22356324011133125132617342340322032502121411" in this example, which is used to encrypt the same image.

3) Finally, the above two ciphered images encrypted by the two slightly different keys, are compared. This test shows that although the two keys are different in only one bit, there is a difference of up to 99% in terms of pixel grey-scale values between the two images encrypted by following keys. Figure 7 shows the test results.

"42356324011133125132617342340322032502121411"

"22356324011133125132617342340322032502121411".

Also when a key is used to encrypt an image and another altered key is used to decrypt the ciphered image, the decryption will completely fail. Figure 8 clearly shows that the image encrypted by the key "42356324011133125132617342340322032502121411" is not

correctly decrypted by using the key "223563240111331251326173423403220325021214 11" which is different from the first key in only one bit.

**Figure 7. Key Sensitivity Result1:
(a) Original Image,
(b) Encrypted Image: Key =
"423563240111331251326173423403220325021214 11",
(c) Encrypted Image: Key =
"223563240111331251326173423403220325021214 11"**

a) b)

**Figure 8. Key Sensitivity Result2:
(a) Decrypted Image by using Key =
"423563240111331251326173423403220325021214 11" ,
(c) Decrypted Image by using Key =
"223563240111331251326173423403220325021214 11"**

4.3. Time Analysis

A part from the security consideration, running speed of the algorithm is an important aspect for a good encryption algorithm. The proposed algorithm is implemented using MATLAB 7.6 under Microsoft windows 7 Professional Version 64-bit, Intel(R) Core(TM) 2 Due CPU 2.53GHz, 4GB of RAM. We have implemented the proposed method on four

images Lena, Camera man, Pepper, and Barbara with different keys. Time of encryption in maximum complexity mode (include all scan patterns in first two steps and maximum number of running, 7 times) are listed in Table 5.

Table 5. Results of Elapsed Time Measured during Algorithm Running

	Lena	Camera man	Pepper	Barbara
Encryption Time (s)	0.1022	0.1120	0.1051	0.1010

As shown in table, our algorithm has fast performance on the speed. It is obvious when the algorithm run on less complexity mode the performance become faster. Due to decryption procedure is similar to that of the encryption but in the reversed order so encryption and decryption speeds are almost equal.

5. Conclusion

In this paper, we have presented a novel algorithm for image encryption based on scan patterns that first shuffle the image completely in two steps and then exploit function XOR. The algorithm trades off between speed and security, so that more complex key which shuffle the image completely results more security but it consume more time. Yet in most complexity mode the algorithm act fast. The proposed algorithm implement and test on four known gray-level images “Lena”, “Camera man”, “Pepper”, and “Barbara”. The simulation results for gray-level images show that the proposed algorithm has great performance in terms of sensitivity, speed, and security so that even by a simple key the NPCR, UACI, and Entropy can satisfy security and performance requirements.

References

[1] N. Taneja, B. Raman and I. Gupta, “Combinational domain encryption for still visual data”, Journal of Multimedia Tools and Applications, DOI 10.1007/s11042-011-0775-4, **(2011)**.
[2] K. C. Ravishankar and M. G. Venkateshmurthy, “Region Based Selective Image Encryption”, International Conference on Computing & Informatics, ICOCI 06, IEEE, Kuala Lumpur, **(2006)**.
[3] C. S. Chen and R. J. Chen, “Image Encryption and Decryption Using SCAN Methodology”, Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT, IEEE, Taipei, Taiwan, **(2006)**.
[4] I. Ozturk and I. Sogukpınar, “Analysis and Comparison of Image Encryption Algorithms”, Journal of transactions on engineering, computing and technology, vol. 3, **(2004)**, pp. 38.
[5] A. Gautam, M. Panwar and P. R. Gupta, “A New Image Encryption Approach Using Block Based Transformation Algorithm”, International Journal Of Advanced Engineering Sciences and Technologies, IJAEST, **(2011)**.
[6] I. S. Sam, P. Devaraj and R. S. Bhuvaneswaran, “A novel image cipher based on mixed transformed logistic maps”, Journal of Multimedia Tools and Applications, DOI 10.1007/s11042-010-0652-6, **(2010)**.
[7] N. Taneja, B. Raman and I. Gupta, “Chaos based cryptosystem for still visual data”, Journal of Multimedia Tools and Applications, DOI 10.1007/s11042-011-0837-7, **(2011)**.
[8] M. A. B. Younes and A. Jantan, “Image Encryption Using Block-Based Transformation Algorithm”, IAENG International Journal of Computer Science, vol. 35, no. 1, pp. 15-23, **(2008)**.
[9] M. A. E. Wahed, S. Mesbah and A. Shoukry, “Efficiency and Security of Some Image Encryption Algorithms”, Proceedings of the World Congress on Engineering, WCE, UK, vol. 1, **(2008)**.
[10] H. Shuihua and Y. Shuangyuan, “An Asymmetric Image Encryption Based on Matrix Transformation”, ECTI Transactions on Computer and Information Technology, vol. 1, no. 2, **(2005)**.
[11] J. C. Yen and J. I. Guo, “A new chaotic key-based design for image encryption and decryption”, IEEE International Conference Circuits and Systems, Switzerland, vol. 4, **(2000)**, pp. 49-52.
[12] A. Mitra, Y. V. Subba Rao and S. R. M. Prasnna, “A new image encryption approach using combinational permutation techniques”, Journal of computer Science, vol. 1, no. 1, **(2006)**, pp. 127.

[13] G. Zhi-Hong, H. Fangjun and G. Wenjie, "Chaos-based image encryption algorithm", Department of Electrical and computer Engineering, University of Waterloo, Canada. Published by: Elsevier, pp. 153-157, **(2005)**.
[14] A. Sinha and K. Singh, "Image encrypt ion by using fractional Fourier transform and Jigsaw transform in image bit planes", Source: optical engineering, spie-int society optical engineering, vol. 44, no. 5, **(2005)**, pp. 15-18.
[15] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns", Journal of Pattern Recognition Society, vol. 37, no. 4, **(2004)**, pp. 725-737.
[16] A. Sinha and K. Singh, "A technique for image encryption using digital signature", Source: Optics Communications, vol. 218, no. 4, **(2003)**, pp. 229-234.
[17] M. V. Droogenbroech and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images", Proceedings of Advanced Concepts for Intelligent Vision Systems, ACIVS'02, Ghent, Belgium, **(2002)**.
[18] S. S. Maniccam, G. Nikolaos and Bourbakis, "Lossless image compression and encryption using SCAN", Journal of: Pattern Recognition, vol. 34, no. 6, **(2001)**, pp. 1229-1245.
[19] S. P. Nanavati and P. K. Panigrahi, "Wavelets: applications to image compression- I", joined of the scientific and engineering computing, vol. 9, no. 3, **(2004)**, pp. 4-10.
[20] A. Vitali, A. Borneo, M. Fumagalli and R. Rinaldo, "Video over IP using standard-compatible multiple de script ion coding", Journal of Zhejiang University- Science A, vol. 7, no. 5, **(2006)**, pp. 668-676.
[21] A. Attar, R. M. Rad and A. Shahbahrami, "An Accurate Gradient-Based Predictive Algorithm for Image Compression", The 8th ACM International Conference on Advances in Mobile Computing and Multimedia, MoMM, Paris, France, **(2010)**.
[22] S. S. Maniccam and N. G. Bourbakis, "SCAN Based Lossless Image Compression and Encryption", The IEEE International Conference on Information Intelligence and Systems, ICIIS, Washington, DC, **(1999)**.
[23] N. G. Bourbakis, "Image Data Compression-Encryption Using G-Scan Patterns", IEEE International Conference on Computational Cybernetics and Simulation Systems, Man, and Cybernetics, USA, **(1997)**.
[24] A. Attar, R. M. Rad and A. Shahbahrami, "A Comprehensive Layer Based Encryption Method for Visual Data", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 6, no. 1, **(2013)**, pp. 37-48.
[25] A. Attar, R. Moradirad and R. Ebrahimi Atani, "A survey of Image spamming and filtering techniques", Artificial Intelligence Review, vol. 40, no. 1, **(2013)**, pp. 71-105.

Authors

Reza Moradi Rad, he was born in Rasht, Iran. He began his journey through life on September 12, 1989. He graduated from Kharazmi high school of Rasht and awarded the diploma in Mathematics and Physics in 2006. In 2007, he started to study computer engineering in University of Guilan, the largest and most prestige university in north of Iran. Currently he is in his last semester of B.Sc. studies and throughout his final research project; he is working on image processing, machine vision, and pattern recognition. He has published more than 12 journal and conference papers so far.

Abdolrahman Attar Qazaani, he was born in Tehran, Iran. He began his journey through life on September 16, 1989. He graduated from Kharazmi high school of Rasht and awarded the diploma in Mathematics and Physics in 2006. In 2007, he started to study computer engineering in University of Guilan, the largest and most prestige university in north of Iran. Currently he is in his last semester of B.Sc. studies and throughout his final research project; he is working on image processing, machine vision, and pattern recognition. He has published more than 12 journal and conference papers so far.

Reza Ebrahimi Atani, he was born in Arak, Iran. He studied Electronics Engineering at the University of Guilan in Rasht, Iran. He got his B.Sc degree in 2002 and at the same year he was accepted to follow his masters study at Iran University of Science & Technology (IUST) in Tehran, Iran. He joined the Electronics Research Center of IUST and received the M.Sc degree in Electronics-VLSI design in 2004. He received the Ph.D. degree in 2010 where he worked on Design and implementation of a stream ciphers for mobile communications. He has an assistant professor position in Department of Computer Engineering at the University of Guilan. His research interests include Cryptology, Cryptographic Hardware and Embedded Systems (CHES), Computer and Network security, Information Hiding, Semi-Ring Theory and Semi-Module Theory **(Corresponding Author)**.