Research Journal of Applied Sciences, Engineering and Technology 4(18): 3440-3447, 2012
ISSN: 2040-7467

Submitted: April 17, 2012 Accepted: May 06, 2012 Published: September 15, 2012

Analysis and Improvement of Encryption Algorithm Based on Blocked and Chaotic Image Scrambling

[1]Yunpeng Zhang, [2]Peng Sun, [1]Liang Yi, [1]Yongqiang Ma and [1]Ziyi Guo
[1]College of Software and Microelectronics, Northwestern Polytechnical University, 710072 Xian, China
[2]College of Computer, Northwestern Polytechnical University 710072 Xian, China

Abstract: Based on blocked image scrambling encryption, this study presents a new image encryption algorithm by introducing chaos theory. This algorithm firstly makes spatial scrambling based on image blocking in order to interrupt pixel position, then furthering this interruption through Arnold Mapping in the chaos and transforms pixel RGB color space through optimized Arnold Mapping. After this process, we get the final encrypted image through a series of iteration. This algorithm has a lot of advantages such as a large key space, high effectiveness and resisting common attack successfully, but it needs improvements on some aspects, such as the key sensitivity

Keywords: Blocking, chaos, cryptography, image

INTRODUCTION

With the rapid development of computer science, multimedia technology has become increasingly mature, so image encryption is a more and more important subject. The chaotic mapping owns sensitivity about parameters and initial conditions, which corresponds to the needs of encryption system. In this study, we absorb the spatial scrambling idea and introduce a spatial scrambling method on the base of image blocking in order to enhance anti-attack ability of the image. We make further efforts to scramble the image with optimized chaotic system and get the encrypted image in the end.

ALGORITHM

Arnold mapping: In this section, we talk about Arnold Mapping. Document (Daniel and Robert, 1991) presents a digital image scrambling scheme based on Arnold Mapping. Document (Shujun and Xuan, 2002) spread Arnold Mapping to high dimensional, the corresponding transformation matrix is:

$$A_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & 3 & \cdots & N \end{bmatrix} \quad (1)$$

For a pixel, it is its RGB color space that influences the color. Here, we can view the RGB color space as a cube in a three-dimensional space, whose one vertex lies in the origin of coordinates. As we often use integers to represent the RGB color components in the computer, actually discrete grid points are used:

$$V_{RGB} = \{(x, y, z), x, y, z = 0, 1, \ldots, 255\} \quad (2)$$

For the RGB color shown above, we use the spread Arnold transformation to scramble in this three-dimensional grid. In this way, the minimum three-dimensional transformation matrix A_N is used:

$$A_N = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad (3)$$

then:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \bmod 265, \ (x, y, z) \in V_{RGB} \quad (4)$$

For the RGB color space, if the transformation above is used iteratively, we can computer to confirm that its cycle is 448. However, when it comes to images, different images have different permutations and combinations of color, so we can only think that the cycle is just a supremum. One pixel point can be iterated several times

Corresponding Author: Yunpeng Zhang, College of Software and Microelectronics, Northwestern Polytechnical University 710072 Xian, China

for the encryption and the number of iterations can be used as the cipher key.

Image preprocessing: Effective preprocessing is able to enhance the security of encryption algorithm. Scrambling on the spatial domain is a common preprocessing and could weaken the correlation of the neighbor pixels on the initial image.

The preprocessing in this study rests on the blocked image transformation, making the spatial transformation before chaotic mapping, which will definitely improve the security of algorithm.

Divide the initial image into four parts: left upper, right upper, left lower and right lower and mark these sub-images with f1 (1) (x, y), f2 (1) (x, y), f3 (1) (x, y), f4 (1) (x, y).

Uniformly Disperse the pixels in the sub-images of f2 (1) (x, y) and f4 (1) (x, y) into f1 (1) (x, y) and f3 (1) (x, y) respectively; then block the image we have got as step (1) and mark the sub-images with f1 (2) (x, y), f2 (2) (x, y), f3 (2) (x, y), f4 (2) (x, y). The sequence is not unique, we can also disperse the pixels of f2 (1) (x, y) and f1 (1) (x, y) into f3 (1) (x, y) and f4 (1) (x, y) or some other combinations. The sequence can be used as one part in the cipher key after promise.

Uniformly Disperse the pixels in the sub-images of f1 (2) (x, y) and f2 (2) (x, y) into f3 (2) (x, y) and f4 (2) (x, y), respectively, mark the result with f^0 (x), or we can also disperse the pixels in f1 (2) (x, y) and f2 (2) (x, y) into f4 (2) (x, y) and f3 (2) (x, y), respectively. The sequence can also be used as another part in the cipher key after promise. Then we get image F.

Repeat step 2-3, scramble the pixel positions in the image adequately.

In this part, pixels are dispersed uniformly in step 2: we disperse one sub-image P = (plk) m*m into anothe r sub-image Q = (qlk) m*m. as shown in the following expression (5):

$$\pi_{i,j} = \begin{cases} p_{i,j/2} & if\ j\ is\ an\ even \\ q_{i,(j+1)2} & f\ j\ is\ an\ odd \end{cases} \quad (5)$$

$\prod = (\pi_{(i,\ j)m/2m})$ is the image after scrambling.

Similarly, pixels are dispersed uniformly in step 3: we disperse one sub-image $P = (p_{lk})_{m*m}$ into another sub-image $Q = (q_{lk})_{m*m.}$ as shown in the following expression (6):

$$\pi_{i,j} = \begin{cases} p_{i/2,j} & if\ j\ is\ an\ even \\ q_{(i+1)/2,j} & if\ j\ is\ an\ odd \end{cases} \quad (6)$$

$\prod = (\pi_{(I,\ j)m/2m})$ is the image after vertical scrambling.

This is the finish of the image preprocessing.

Image encryption scheme: Input: Initial image I with size, encryption key (X_1, X_2, n, a, b, X_{Arnodd}, T_{Arnold}).

X_1 is the dispersion sequence in step 1, X_2 is the dispersion sequence in step 2, n is the total number of dispersion iterations, a and b are parameters in Arnold transformation, X_{Arnold} is the iteration times of image pixel position while T_{Arnold} is the iteration times of image pixel value in Arnold transformation.
Output: encrypted image I*.

Step 1: According to four steps in image preprocessing, firstly, we divide the initial image into four equal parts, introducing four arrays pointer1[], pointer2[], pointer3[], pointer4[] respectively.In the preprocessing, step2 and step3 use the cipher

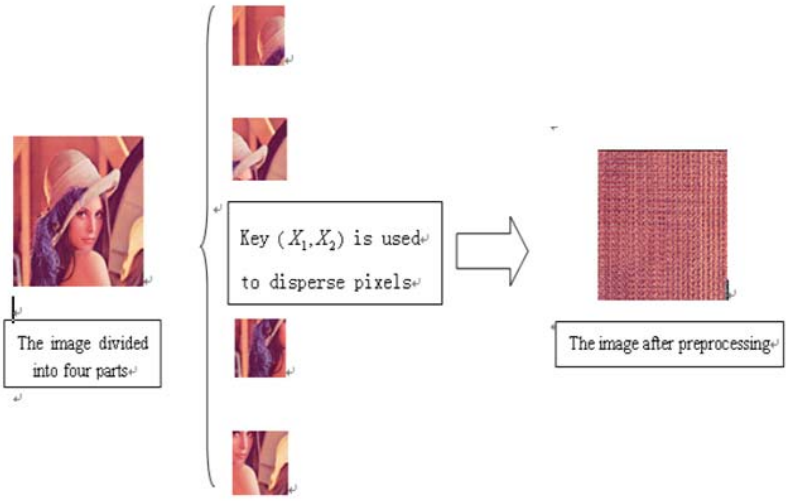


Fig. 1: The image after preprocessing

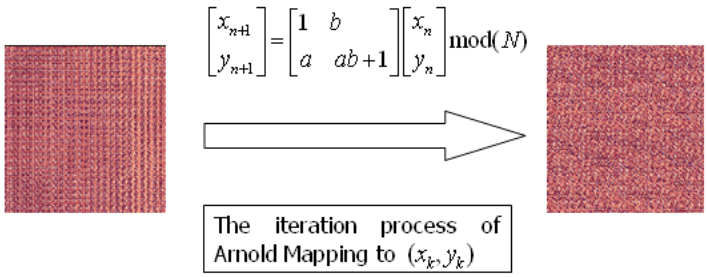


Fig. 2: The arnold mapping to image pixel positions

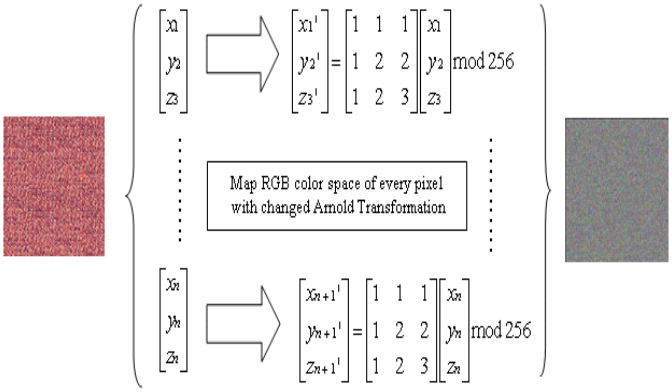


Fig. 3: Map RGB color space with changed arnold transformation

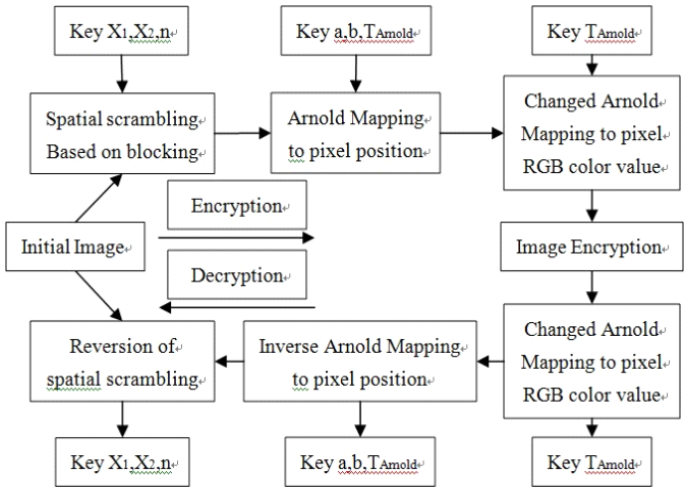


Fig. 4: Algorithm flow chart

key (X_1, X_2) to disperse sub-images. In step4 the dispersions are iterated for n times. This process is shown in Fig. 1.

Step 2: We make Arnold Transformation to the initial image I with the formula $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} (\text{mod } 1)$ and the cipher key (a, b, X_{Arnold}) and get image I'.

Let (x_1, y_1) be horizontal and vertical coordinate of the first pixel point before transformation and (x_1', y_1') be horizontal and vertical coordinate of the first pixel point after transformation. Similarly,(x_k, y_k) and (x_k', y_k') are horizontal and vertical coordinates of the k-th pixel point before and after transformation respectively. The transformation process is shown in Fig. 2.

Step 3: We use formula (1) to map the RGB color space value and change its pixel grey value. Next we iterate the mapping with T_{Arnold} cipher key and get the final encrypted image I*.

We know that the RGB space value influences the color of one pixel. Let (x_1,y_1,z_1) be the RGB color space value of the first pixel and (x_1', y_1', z_1') be the value after transformation. Similarly, we can finish the mapping to every pixel and encrypt the image by using T_{Arnold} iteratively. The transformation is shown in Fig. 3.

The overall algorithm process is shown in Fig. 4.

ANALYSIS AND COMPARISON

Theoretical analysis of security:

- **Diffusibility analysis:** Algorithm's diffusibility is a significant factor that cipher text could conceal information of plain text. Apart from some characteristics of chaotic system, the algorithm in this study takes diffusibility into full consideration. In the encryption process, firstly, the algorithm blocks the initial image and scrambles its useful information and then, Arnold Mapping is introduced to transform pixel positions and to further the scrambling. In the end, we use changed Arnold Mapping to transform RGB color space. The process is successive and the former encryption result can be viewed as an input in the next step.

  In addition, the difference of iteration times in every step will have an effect on the next. So, the algorithm in this study has good diffusibility.
- **Complexity resisting exhaustive-key-research attack:** In the algorithm, we block and scramble the image with key (X_1, X_2) and both X_1 and X_2 have 2 C^2_4 kinds. The scrambling iteration is 3, so there are 432 possibilities in the preprocessing. In the Arnold Mapping to pixel position, the key is (a, b), a $\in$ n, b $\in$ N and the estimated cycle of Arnold Mapping is $T_n \leq N2/2$. For image 256*256, its cycle is $T_n \leq 2562/2$. The cycle of RGB color space based on Arnold transformation is 448 and its exhaustive key space cycle is $6.3*10^9*N^2$ (N could be a very large number). In this way, its key space is safe enough to resist exhaustive-key-research attack.
- **Phase space reconstruction attack research:** In this study, we use Arnold Mapping on pixel position transformation and changed Arnold Mapping on pixel RGB value, which is proved to improve the ability to resist phase space reconstruction attack.

The image after chaotic encryption is the computing result of plain text image pixel and its corresponding chaotic sequence. Because the range of plain text image pixel value is comparatively small and the chaotic sequence's is comparatively large, the numeric value of encrypted image must remain some characteristics of chaos equation. If we reconstruct the phase space of encrypted image and compare the result with the initial phase space image of chaotic kinetic equation, we will the one with the most similar kinetic characteristics and determine the kinetics equation. In this section, we reconstruct phase space of encrypted image with delay time and embedded dimension. The result is shown in Fig. 5, 6 and 7.

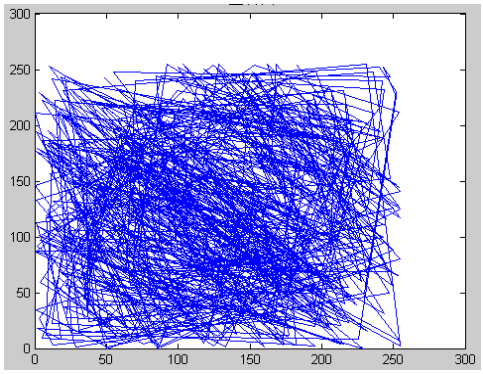


Fig. 5: Phase space reconstruction on B

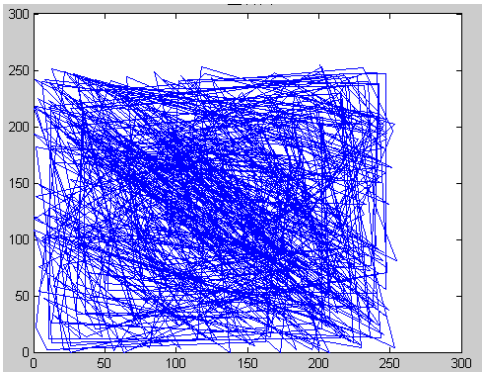


Fig. 6: Phase space reconstruction on G

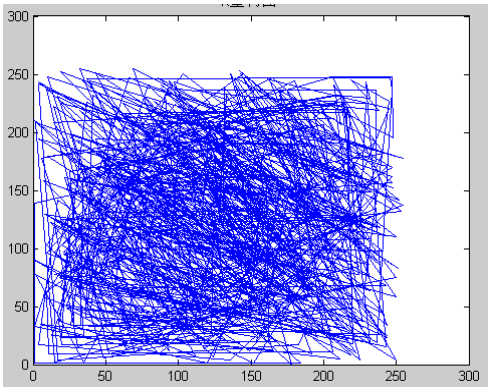


Fig. 7: Phase space reconstruction on R

The reconstruction phase space has no meaning and we cannot obtain useful information about chaos system. The couple-ch In the preprocessing, step2 and aotic and variable-parameter system in this study owns high immunity to phase space reconstruction attack and is practical in encryption.

RESULTS AND ANALYSIS

Furthermore, to verify validity and security of the encryption algorithm, we test:

- Sensibility of the cipher key
- Characteristic of histogram statistics
- Sequence
- Information entropy on MFC visualized image encryption platform under Visual C++ 6.0 environment.

The result shows that the encryption algorithm is useful and efficient.

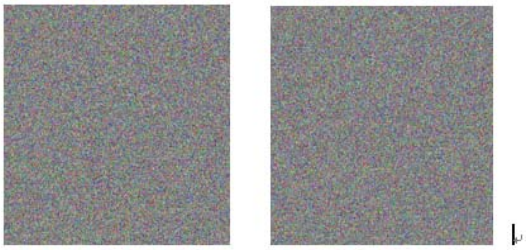
Analysis to sensibility of cipher key: As shown in Fig. 8 the left one is a standard 512*512 chromo photograph. If we input the key (X_1, X_2, 3, 1, 2, 10, 50), we will get the encrypted image on the right. Obviously, the image after encryption is in uniformly carpeted disorder and the initial information is unrecognized.

(a) Initial image (b) Encrypted image

Fig. 8: Image encryption

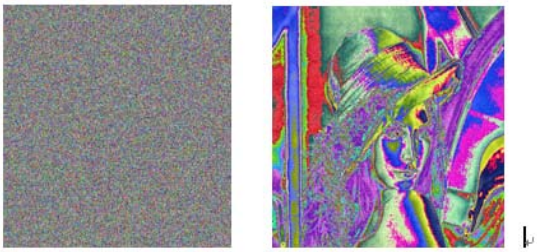
(a) Encrypted image (b) Decrypted image

Fig. 9: Image decryption



(a) Encrypted image (b) Decrypted image with wrong key

Fig. 10: Image decryption with wrong key



(a) Encrypted image (b) Decrypted image with wrong key

Fig. 11: Image decryption with wrong key

If we input the correct decryption key, the decrypted result is as the same as the initial image shown in Fig. 9. If the key input is ($_{X1}$, X_2, 3, 1, 2, 10, 48), the result is not correct shown in Fig. 10.

If the key input is (X_1, X_2, 3, 1, 2, 10, 49), the result is not correct shown in Fig. 11. However, the basic outline of image is shown, which indicates that the sensibility of encryption algorithm needs to be improved.

- **Statistics analysis:** Histogram is the objective reflection of statistical regularity. A good image encryption algorithm should not provide any useful information from histogram statistics and an ideal algorithm will change the initial image with non-uniform pixel value distribution to a uniform one, which means that every pixel value has the same probability in the whole space. As a result, statistical characteristics of plain text will be totally disturbed, largely reducing the relevance between plain text and cipher text.

Figure 12 is the histogram on B, G and R of the initial image. Figure 13 is the histogram result of the encrypted image. By comparison, we know that this encryption algorithm effectively conceal statistical characteristics and we cannot get any useful information from encrypted image grey value. Adjacent pixel relevance analysis

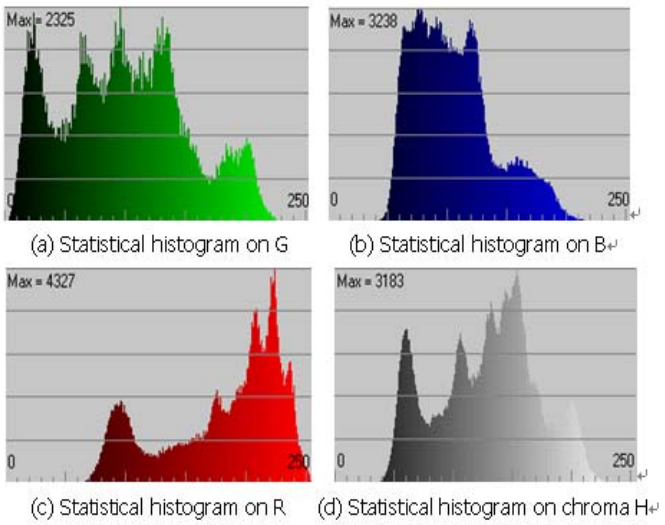


Fig. 12: Statistical histogram of the initial image

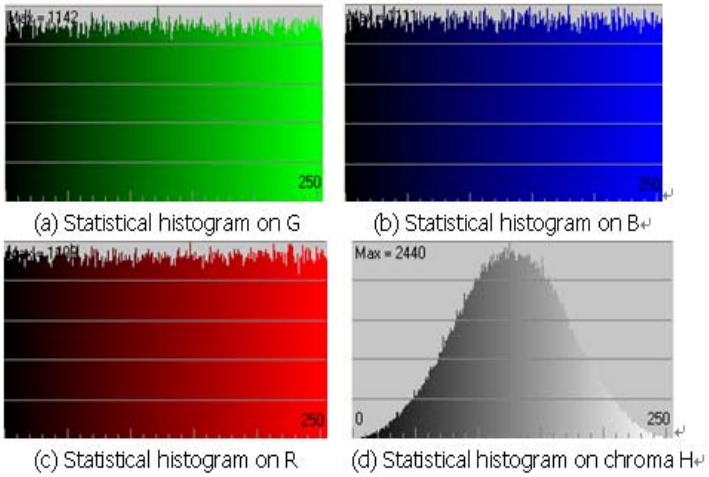


Fig. 13: Statistical histogram of the encrypted image

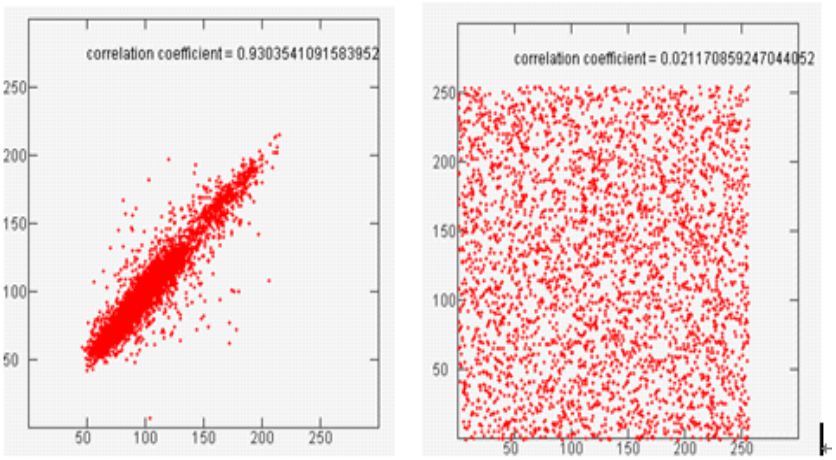
Usually, we extract several adjacent pixel pairs (horizontal, vertical or diagonal) from the image randomly to test the adjacent pixel relevance of the plain test image and the cipher text image, using the following formulas:

$$
\begin{aligned}
E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i \\
D(x) &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)^2 \\
Con(x,y) &= \frac{1}{N}\sum_{i=1}^{N} \left(x_i - E(x)\right)\left(y_i - E(y)\right) \\
\gamma_{xy} &= \frac{Con(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}
\end{aligned}
\tag{7}
$$

In the formulas, x and y are grey values of two adjacent pixels and γ_{xy} is the correlation coefficient. Generally, γ_{xy} of the initial image is close to 1 while γ_{xy} of the encrypted image is close to 0 if the encryption algorithm is good enough, indicating that there is almost no relevance between adjacent pixels and statistical characteristics of plain test are spread into random cipher text.

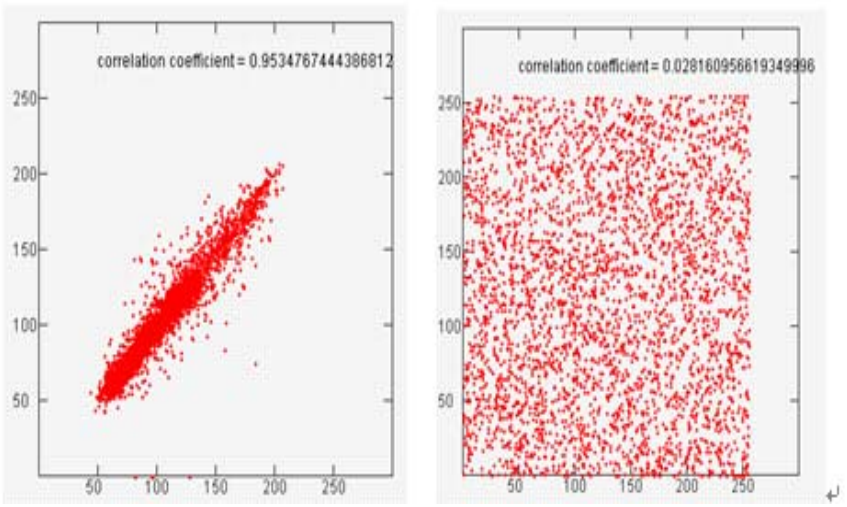
In this section, we select 3000 pair's adjacent pixel (horizontal or vertical) and use formulas above to test the relevance.

From Fig. 14 and 15, the adjacent pixels of initial image are of high relevance with correlation coefficient being close to 1. However, after encryption, the adjacent pixels are almost of no relevance with correlation coefficient being close to 0.



(a) Plain text images (b) Cipher text image

Fig. 14: Horizontal adjacent pixel relevance analysis



(a) Plain text images (b) Cipher text image

Fig. 15: Vertical adjacent pixel relevance analysis

- **Information entropy analysis:** Information entropy (Shannon, 1948), the average information quantity, is defined as follows:

$$H(x) = -\sum_{i}^{n} p(x_i) \log_2 p(x_i) \quad (8)$$

$p(x_i)$ expresses the probability of the symbol x_i occurring in the message, which satisfies:

$$\sum_{i=1}^{n} p(x_i) = 1 \quad 0 \le p(x_i) \ge 1 (i = 1, 2, ..., n) \quad (9)$$

When every symbol occurs with equal probability, that is $p(x_i) = 1/n$, the information entropy takes a maximum value, which can be expressed as $H_{max} = \log_2 n$. According to the concepts above, we know that information entropy represents statistical characteristics of the general information source and the general average uncertainty. Before encryption, information entropy of the initial image is 7.7502 and yet this number is 7.9978 after encryption. The image has become more confused since information entropy is higher.

Now, we compare the algorithm in this study with some typical image scrambling encryption algorithms, including the controllability of safety grade, the efficiency and the size of cipher key space. We get the experiment

Table 1: Comparison between different algorithms

	Controllability of safety grade	Efficiency	Key space size
Algorithm in this study	yes	0.719 s	10^{13}
Algorithm 1(Bian, 2007)	no	6.516 s	10^{18}
Algorithm 2 (Wang, 2004)	no	0.75 s	10^{23}

data about algorithm efficiency under AMD 3200+ 2.1GHz CPU 1.00GB Main Memory. The results are shown in Table 1.

CONCLUSION

In this study, we put forward a new image encryption scheme based on image blocking and chaos. First, we block the initial image, disperse pixels of the blocked images and add space scrambling. Second, we use Arnold Mapping to transform pixel positions. Third, we use changed Arnold Mapping to transform pixel RGB color space value. Through analysis and experiment, this algorithm has good security, large key space and high efficiency. However, its key sensitivity needs to be improved.

ACKNOWLEDGMENT

This study is supported by Science and Technology Development Project of Shaanxi Province Project (2010K06-22 g), Industrial application technology research and development project of Xi'an (CXY1118 (1)), Basic research fund of Northwestern Polytechnical University (GAKY100101) and R Fund of College of Software and Microelectronics of Northwestern Polytechnical University (2010R001).

REFERENCES

Bian, L., 2007. Research of image encryption based on Chaos [D]. M.Sc. Thesis Submitted to Guangdong University of Technology.

Daniel, D.W. and A.J.M. Robert, 1991. Supercomputer investigations of a chaotic encryption algorithm[J]. Cryptologia, 15(2): 140-151.

Shujun, L. and Z. Xuan, 2002. On the security of an image encryption method[C]. Proceedings of IEEE 2002 International Conference on Image Processing (ICIP'2002), 2: 925-928.

Shannon, C.E., 1948. A mathematical theory of communication [J]. Bell Syst. Techn. J., 27: 379-423, 623-656.

Wang, J., 2004. Application Status of Chaos in Digital Image Encryption [D]. University of Science and Technology of China.