International Journal of Security and Its Applications
Vol.8, No.1 (2014), pp.343-354
http://dx.doi.org/10.14257/ijsia.2014.8.1.32

Efficient Image Scrambling based on any Chaotic Map

Cao Guanghui[1,2*], Hu Kai[1] and Zhou Jun[2]
[1]*School of Computer Science and Technology, Beihang University, Beijing, 100191, China;*
[2]*Electronic & Information Engineering College, Liaoning University of Technology, Jinzhou 121001, China;*
**caoguanghuineu@163.com*

Abstract

This paper presents a new image permutation algorithm to improve permutation efficiency. Based on uniform trend theorem and ideal bias theorem, we propose two classes of extractors, which solve the critical problem of any chaotic sequence uniformity. Then, the uniform chaotic sequence is transformed into integer space and the random integers are used to perform one-dimensional data permutation algorithm, which is based on data position interchange. Based on the data vector permutation, a secure and efficient image permutation is presented. Comparing with previous approaches, the proposed scheme is secure and its efficiency is enhanced.

Keywords: *Image scrambling; position interchange; uniform distribution; sorting permutation*

1. Introduction

With the rapid development and popularity of information technology, people worry about its effects on the environment. Gartner, the world's leading information technology research and advisory company, counts carbon emissions of information technology and data shows that its proportion in the whole energy consumption is equal to that of the airline industry. Mc Kinsey, a global management consulting firm, reports that by 2020, all kinds of IT devices, such as laptop, smart phone and Tablet computer, will become one main source of the emitted greenhouse gas. Facing such serious environment problem, Electronic manufactures adopt active measures to reduce the environmental pollution [1]. Due to use the same hardware and provide the same function, different software methods bring different energy consumption and resource requirement. Based on these, in our paper, we propose an efficient image permutation for improving energy efficiency in image protection field.

Image permutation was firstly proposed by Bourbakis in 1992 [2]. The idea originates from the specification and development of scanning language, devotes to image encryption. With the network and multimedia development, image permutation has made great progress in many areas, *e.g.*, image encryption [3, 4], information hiding [5, 6], digital watermarking [7, 8] and so on. The widely used image permutation algorithms are based on: Arnold transformation or gray code or generalized gray codes; fractal geometry; Hilbert curve, FASS curve and so on. Recently, along with further study of chaos theory, two kinds of image permutation algorithms are proposed. One is based on chaos sorting transformation; the other is chaotic position interchange. The former has many examples, such as image permutation based on bit pixel [9], image permutation based on combinatorial matrix [10], image permutation based on multi-chaotic system [11], and a total image permutation algorithm [12]

ISSN: 1738-9976 IJSIA

and so on. And the latter has image scrambling based on Logistic uniform distribution [13]. The core technology of the former is sorting chaotic sequence and permuting data vector, which takes time $O(n\log n)$. The latter one is permuting data vector by position interchange, which takes time $O(n)$. In view of efficiency, our paper focuses on the latter. Given that the latter needs chaos uniform sequence, but the current generation method of such sequence is conditional, which can be implemented only when probability density function (pdf) of chaotic map is known. However, the fact is, for the majority of chaos equations, their pdfs are unknown. How to use such chaos in this field is the problem we are concerned here. In this paper, we solve this problem by designing two classes of random extractors, which can transform any chaos sequence from non-uniform to uniform distribution. And then, based such uniform sequence, efficient image permutation is implemented.

2. Efficient Image Scrambling

2.1. Construction of Chaotic Uniform Sequence

At present, there are many methods that transform chaotic sequence from non-uniform to uniform distribution. For example, chaotic sequence generated by TD-ERCS system is transformed into uniform pseudorandom number by arccosine and arctan function [14].Chaotic sequence generated by z-Logistic map is transformed into uniform pseudorandom number [15]. However, both schemes aim at special chaos maps and therefore not representative. A general uniformity method, which based on bit operation of computer representation of floating-point numbers, is proposed and suitable for hardware implementation [16]. But this procedure is too complicated to use in efficient image permutation field. Although a simple and efficient uniform algorithm is provided based on chaos with pdf known [13], unfortunately, most of chaotic maps are not with pdf. A shift-and-extract method, shifting the decimal point k digits to the right and cutting off the integer, is proposed in [17], to realize chaotic sequence uniformity. But since it isn't based on theory but based on a lot of experiments, it only discusses chaotic sequence uniformity based on Henon map. Aiming at these shortcomings, we first give theory foundation for shift-and-extract method, which proves that this method can apply to any chaos map, and then design two classes of random extractors to simply and effectively obtain uniform pseudorandom sequence from any chaotic sequence.

2.1.1. Basic Theory: From literature [16], we not only find Theorem 1 but also induce Theorem 2, which lay theory foundation for shift-and-extract method.

Theorem 1 (Uniform trend theorem)

Let $\xi \in [0,1]$ be a real variable with continuous (or piecewise continuous) probability density $p_\xi(x)$, ξ is represented in binary form $\xi=0.\xi_1\xi_2...\xi_i...$, in which ξ_i is i_{th} bit of ξ. Then, $\xi_i \in \{0,1\}$, (i=1,2,…) is a binary random variable sequence, and when $i \to \infty$, ξ_i is close to uniform distribution, that is, $\lim_{n\to\infty} P(\xi_n = 0) = \lim_{n\to\infty} P(\xi_n = 1)$.

This theorem shows that, random bit ξ_i in ξ has a naturally uniform tendency, which has no relationship with ξ distribution.

Theorem 2 (Ideal bias theorem)

Random variable ξ_n, the nth bit of ξ, has uniform distribution with a bias $(1/2^n) \times \left| p'_\xi(x_k) \right|$ on the interval length of $1/2^n$, where $p'_\xi(x_k)$ denotes the derivative of $p_\xi(x)$ at x_k.

This theorem shows that $p_{\xi}(x)$ affects approximation degree of ξ_n to uniform distribution, however it does not change the uniform distribution trend. Random variable ξ_n obeys uniform distribution with a bias less than $O(2^{-n})$. And the bias will become smaller when number *n* increases.

2.1.2. Uniformity Algorithm: Based on Theorem 1 and 2, a generalized algorithm of chaotic sequence uniformity is proposed as follows.

Suppose chaotic dynamics system is: $\boldsymbol{x}(n+1) = f_{\boldsymbol{k}}(\boldsymbol{x}(n))$ Where $\boldsymbol{x} \in \mathbb{R}^m$, $f_{\boldsymbol{k}} : S \in \mathbb{R}^m \to \mathbb{R}^m$ denotes a nonlinear map, $\boldsymbol{k} \in \mathbb{R}^v$ denotes the parameters of map *f*, *S* denotes the domain of map *f*.

Procedure: First, iterate selected chaos equation, and then use suitable random extractor *E* to extract bit substring from the current chaos variable, repeat the above process until finish. As a result, pseudorandom uniform sequence is generated with a finite bias on the interval (0, 1).

Suppose *E* is random extractor, the formulas for pseudorandom generator are as follows:

$$\boldsymbol{x}(n) = f_{\boldsymbol{k}}(\boldsymbol{x}(n\text{-}1)) \ , \quad r_n = E(\boldsymbol{x}(n))$$

Where, pseudorandom extractor *E* has two kinds of implementation ways.

1) The first class of extractor

The first class (of extractor) is $r_i = \boldsymbol{x}(i) \times 2^n - fix(\boldsymbol{x}(i) \times 2^n)$, which generates random number with bias less than $O(2^{-n})$ on the interval [0,1]. Where, $x(i)$ denotes the *i*th component of iteration sequence *x*, *fix* denotes round towards zero. Since different computers have different finite-precision data representation, which may affect calculation result, thus random extractor with fixed precision is devised as follows.

2) The second class of extractor

The second class (of extractor) is $r_i = fix(\boldsymbol{x}(i) \times 2^n - fix(\boldsymbol{x}(i) \times 2^n) \times 2^m) / 2^m$, Which generates random number in the interval [0,1] with bias less than $O(2^{-n})$ and with *m* significant digits. Here, *m* and *n* are affected by application environments and security strength.

2.1.3. Extractors Parameters Studying: Let variables be double-precision floating-point numbers, taking Logistic map $x = \beta \times x \times (1 - x)$ as chaotic map, iteration times number=50,000; initial value *x*=0.832; control parameter β=3.8898;

1) Parameter for the first class

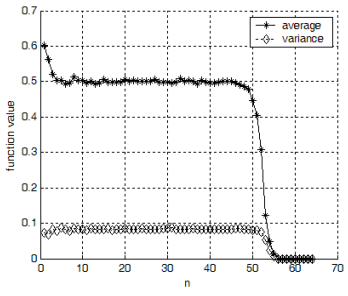


Figure 1. Mean Value And Variance Curve for the First Class of Extractor

The mean value curve (denoted by the sign star) and variance curve (denoted by diamond) of pseudorandom sequence generated by the first class of extractor are shown in Figure 1. From the results, we know that when extractor parameter $n \in [4,50]$, pseudorandom sequence has good pseudorandom property.

2) Parameters for the second class

Under the conditions of $m \in [8,16]$ and $n \in [4,50-m]$, the mean value curve (in Figure 2) and the variance curve (in Figure 3) are shown. From these experimental results, we know that, in the parameter range given, the mean value and the variance are stable in the around 0.5 and 0.08, respectively.

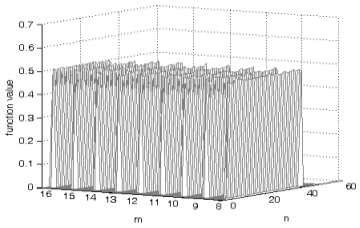


Figure 2. Mean Value

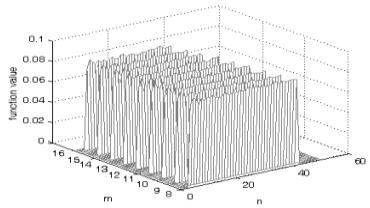


Figure 3. Variance Curve

2.1.4. Uniform Distribution Test: In this section, we conduct a detailed evaluation about the above algorithm through simulation and statistical tests.

1) Simulation Results

Here we still choose Logistic map as chaotic dynamics, and the corresponding initial value x=0.01, parameter β=3.98, iterate 250, 000 times, the total number of bins is 100. Parameters m=10 and n=10 for extractors. Figure 4 shows the histogram for original Logistic sequence. Figure 5 shows the histogram for the first class. Figure 6 shows the histogram for the second class. From experimental results, we know that both extractors can transform non-uniform chaotic sequence into uniform pseudorandom number.

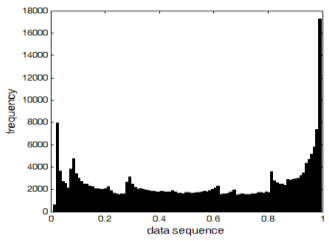
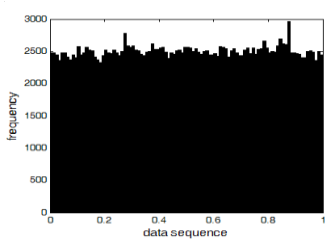


Figure 4. Original Histogram Figure



5. Histogram for the First Class

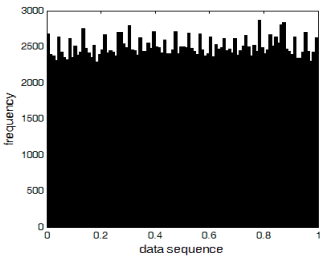


Figure 6. Histogram for the Second Class

2) Discussion about two extractors

As can be seen from Figure 5 and Figure 6, the uniformity effect for the first class is superior to that of the second class.

This phenomenon is illustrated as follows.

Suppose $\xi_i=0.\xi_{i1}\xi_{i2}...\xi_{in}...\xi_{im}...\xi_{ik}$, where ξ_{in} is the *n*th bit of ξ_i. The elements generated by the first class consist of such bits as $\xi_{in},...\xi_{ik}$, the corresponding bias set is $\{O(2^{-n}),...,O(2^{-k})\}$. Those generated by the second class consist of such bits as $\{\xi_{in},...\xi_{im}\}$, the corresponding bias set is $\{O(2^{-n}),...,O(2^{-m})\}$. Thus, Each output of the first class outnumbers the output of the second class by a set of random bits with a bias $\{O(2^{-m-1}),...,O(2^{-k})\}$. Thus, the phenomenon is produced.

3) Statistical test

Here, we use statistical test theory to verify uniformity. By comparison with traditional methods, method [18] provides a quantitative relationship between sample size and confidential interval, which yields more rigorous test result. The basic principle is as follows:

Theorem 3 [18]

Let μ, δ^2 represent expectation and square deviation respectively, α is significance level, ε is the difference between sample and population mean, ζ is the probability corresponding with ε, then sample size needed is *N*. When sample mean *x* deviates from the lower bound of interval of Equation 1, generation algorithm can't pass the test and hence is non-uniform distribution.

$$[\mu-(\delta/\sqrt{N})\times Z_{1-\alpha/2},\mu+(\delta/\sqrt{N})\times Z_{1-\alpha/2}] \quad (1)$$

Where $N=\delta^2/(\zeta\varepsilon^2)$.

Here, chi-square test is used as test statistic, $\{r_i\}$ as sample, to decide whether the population obeys uniform distribution.

In practical test, let $\zeta=0.01$, $\varepsilon=0.01$, and $\mu\geq 0.99$, under these conditions, based on Theorem 3, sample size required is *N*=10, 000. First, generate sample with the length of 10, 000 by tested extractor, and then calculate Chi-square statistics from the sample, finally, make inference from the result. Repeat *N* times of this process. Experimental results show that all samples pass the chi-square test. Thus, one can infer that the two extractors meet desired uniform distribution.

2.1.5. Sensitivity to Initial Value: Sequence sensitivity is very important to image permutation security. Chaos original sequence has very good sensitivity and hence been widely used in security field. Here, we examine the sensitivity of the new sequence generated by two classes of extractors.

As shown from Figure 7 and Figure 8, new sequence still retains good sensitivity characteristic. Thus, transformation implemented by extractor doesn't degrade permutation security.

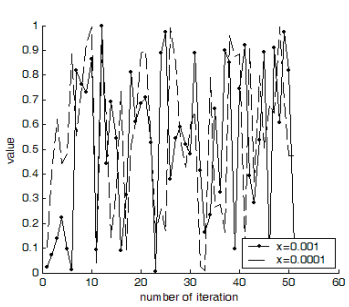


Figure 7. Sensitivity for the First Class

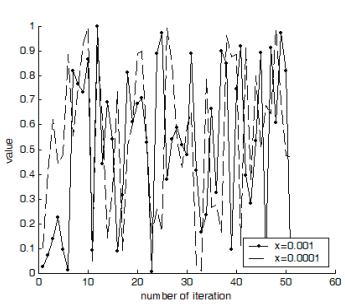


Figure 8. Sensitivity for the Second Class

Since image permutation based on position interchange needs uniform integer random number, uniformity extension Theorem is given below, serving the purpose of mapping uniform (0, 1) sequence to any desired integer space.

2.1.6. Construction of Uniform Integer Space: Theorem 4 (Uniformity extension Theorem) [13]

Let U denote the uniform (0, 1) distribution, then

$$X = \mathrm{Int}(nU) + 1, n \in \square \tag{2}$$

will be equally likely to take on any of the values 1, 2,…n.

2.2. Image Permutation base on Chaotic Uniform Sequence

In Section 2.1, we have generated uniform random integer variable. Here, we first present data vector permutation algorithm based on the random integers, and then apply it into image field to implement efficient image permutation.

2.2.1. Data Vector Permutation: Suppose initial order is $P_1, P_2, ..., P_v$, where $P_i \in [1, v]$. We pick one of the positions 1, …, v at random and then interchange the number in that position with the one in position v. Then, we randomly choose one of the positions 1,…, v-1 and interchange the number in this position with the one in position v-1, and so on. Such as this, data vector permutation is achieved by position exchange [19].

Recalling that equation (2) will be equally likely to take on any of the values 1, 2, …, v, especially, when random variable U is generated by the above extractors, an ordinary random permutation by position exchange is modified into chaos random permutation. We call this procedure CRP. Pseudocode is as follows.

$\boldsymbol{A}[1{:}v] \leftarrow 1...v$
for $j \leftarrow v$ to 2
  $x \leftarrow f_k(x)$
  $y \leftarrow E(x)$
  $val \leftarrow fix(y \times j)+1$
  $A(j) \leftrightarrow \boldsymbol{A}(val)$
end

2.2.2. Efficient Image Permutation Algorithm: For fair comparison, the proposed image permutation algorithm uses exactly the same permutation mode as the one based on chaos sorting [9]. Of course, the proposed algorithm can use any other chaotic map.

For an arbitrary-sized image I, use gray image as an example, let image size be $M \times N$. The proposed algorithm can be written as follows.

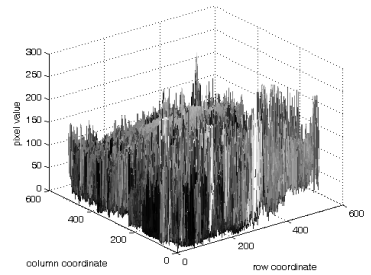
1) Set initial keys of random extractor;
2) Generate control vector $V_1 = [t_1, t_2, \ldots, t_M]$ by CRP;
3) Regard each row of the image as an entry, rearrange all entries by V_1 and obtain the intermediary image $I' = [R'_1, R'_2, ..., R'_M]^T$, where $R'_i = R_{t_i}$ and R_i denotes the ith row, T denotes transposition;
4) For each row of intermediary image, first transform each pixel into 8 bits and then concatenate them to form a long string with length $L = N \times 8$;
5) Generate control vector V_2 with length L by CRP;
6) Rearrange the long string by V_2, which similar to row permutation, and then transform contrarily each 8 bits into a pixel. Repeat 4)-6) until the last row.

The above process efficiently implements image permutation. The decryption algorithm is similar to the above process except step 4) and step 6) with the reverse operation.

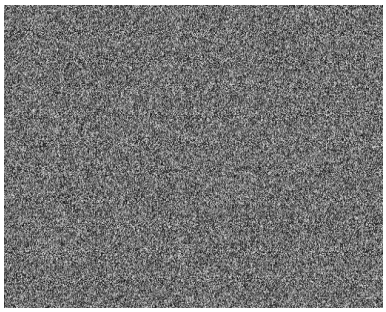
3. Experiment Results

We have scrambled a lot of images using the proposed algorithm. Here, take Bridge image as an example, in which Logistic map is taken as chaotic map, initial value x=0.01, parameter β=3.98 , the second class of extractor is chosen, its parameter m=10, n=10. Experiment results are shown in Figure 9. Seen from the permuted image and corresponding space histogram, the proposed algorithm has good visual security.

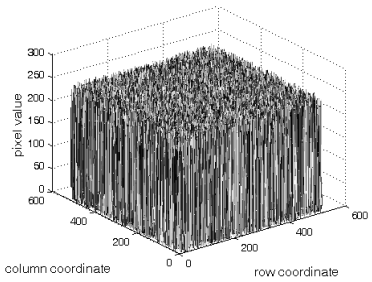
(a) Bridge image



(b) Space histogram of (a)



(c) Permutation image



(d) Space histogram of (c)

Figure 9. Experiment Results

4. Comparison with Chaos Sorting Permutation

Among image permutation algorithms based on chaos sorting, Ye's [9] is a representative algorithm. Thus, it is chosen to compare with the proposed algorithm.

4.1. Key Uniformity

By Shannon theory, for any cryptosystem, there exits

$$I(M,C) \geq H(M) - H(K) \tag{3}$$

Where $I(M,C)$ denotes mutual information of plain M and cipher C, that is, obtainable plain information from cipher. $H(M)$, $H(K)$ denote information entropy of plaintext and cipher key respectively.

According to Equation 3, as $H(K)$ gradually increase, $I(M,C)$ correspondingly decrease, which means decoded information by attacker decrease. Especially, when $H(K) \geq H(M)$, cryptosystem is perfect. When the entire key in key space is equiprobable, $H(K)$ is maximized, $I(M,C)$ is minimized. Therefore, the breakable probability would decrease. It follows from this that the key uniformity is an important factor for cryptosystem designer.

Ye algorithm uses original chaos sequence as key stream, but which doesn't obey uniform distribution. Thus, security of this algorithm needs to be improved. In this paper, we design random extractors, which produce chaos cipher key with maximized uniform distribution. Thus, the security of the proposed system is better than Ye's.

4.2. Key Space

On condition that the cipher key in key space obeys uniform distribution, the value of key entropy depends on the quantity of key. The more the quantity of key is; the larger the value of key entropy is; the more difficult for an attacker to break the cryptosystem. Thus, the quantity of cipher key is another important factor for security.

In Ye algorithm, key space includes an initial value and one parameter of Logistic map. In the proposed algorithm, any chaos can be chosen, even in the case of using Logistic map, key space includes extra one or two parameters than Ye algorithm due to the introduction of key extractors. Thus, the proposed algorithm is more secure.

4.3. Correlation of Adjacent Pixels

To test the correlation between two adjacent pixels, the following procedures are carried out. First, randomly select 10,000 pairs of horizontally (vertically, diagonally) adjacent pixels from an image and then use the following equations to calculate correlation coefficient r_{xy}.

$$cov(x,y) = E(x - E(x))(y - E(y)) \quad r_{xy} = cov(x,y) / \sqrt{D(x)}\sqrt{D(y)}$$

Where x, y are gray value of two adjacent pixels. In numerical calculation, the following discrete formulas are used.

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \quad cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

In order to guarantee the accuracy, repeat 1000 times for the above process and then take the mean value of r_{xy} as the final result. The corresponding formula is as follows.

$$r = mean(r_{xy}) \tag{4}$$

For a reference, a random image, called Rand image, is generated and its correlation coefficient is calculated. Table 1 lists correlation coefficients of image Bridge before and after permuted. From these calculation results, we can see that permuted image has the same order as rand image in correlation, which all approaches to zero. Thus, the proposed algorithm possesses high security against statistical attacks.

Table 1. Correlation Coefficients

	Correlation		
	Horizontal	Vertical	Dialog
Bridge image	0.7746	0.8981	0.8029
Rand image	0.0122	0.0161	-0.0141
Proposed scheme	0.0231	-0.0446	-0.0558
Ye algorithm [9]	0.0153	-0.0238	-0.0837

4.4. Running Time Analysis

In order to verify the efficiency of the proposed scheme, we analyze data permutation time and image permutation time. We use Matlab 6.5.1 as the simulation tool, the hardware environment is one computer. Its configuration is Intel, core2, Quad CPU, 2.66 GHz, 3.37GB. In addition, in Matlab function library, sort() function has been optimized. For fair comparison, we write our own code to finish its function.

4.4.1. Data Permutation Time: As can be seen from pseudocode, Theory estimation times of data permutation based on sorting and position interchange are $O(n\log n)$ and $O(n)$ respectively. Thus, the time ratio of the former to the latter is $O(\log n)$.

The running time ratios (include both theory estimation and empirical measure) of data permutation based on sorting and position interchange are shown in Figure 10. The experiment result shows that the ratio of theory estimation is slightly different from that of empirical measure. This difference is caused mainly by the valuation formula, besides, by computer hardware, software, current computer memory allocation and running processes, *etc*. But on the whole, the two kinds of ratios have the same development trend.

4.4.2. Image Permutation Time: Take widely used image sizes as examples, the running time ratio of Ye and the proposed algorithm is shown in Figure 11. The result shows that the ratio increases steadily with the increasing image size.

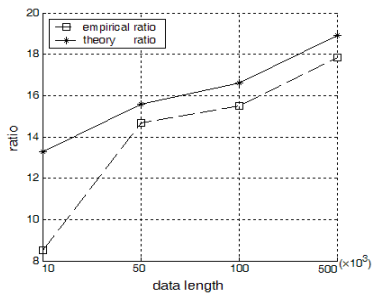


Figure 10. Time Ratio of Data Vector

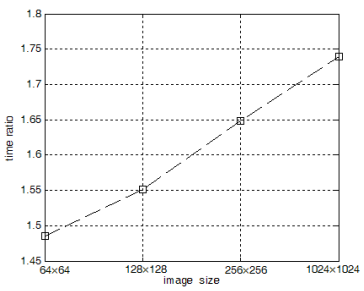


Figure 11. Time Ratio of Image Permutation

5. Conclusion

In order to improve permutation algorithm efficiency, we present fast image permutation algorithm. In this scheme, based on sound theory, we design two classes of extractors, which solve the critical problem of chaotic sequence uniformity. Comparing with Cao algorithm [13], the proposed algorithm is not restricted by the known probability density condition. Comparing with Ye [9], the proposed algorithm has uniform key, larger key space. And most important is that running efficiency is improved by more than 0.5 percent, and it has much more improvement as the image size grows larger.

Acknowledgements

The research is supported by the National Natural Science Foundation of China (Grant No 61073013) and Aeronautical Science Foundation of China (Grant No 2010ZA04001), as well as the Doctoral Scientific Research Foundation of Liaoning Province (Grant No 20121045) and the Scientific Research Foundation of Liaoning University of Technology (Grant No X201316).

References

[1] L. Chuang, T. Yuan and Y. Min, Chinese Journal of Computers, vol. 34, **(2011)**, pp. 593.
[2] N. Bourbakis and C. Alexopoulos, Pattern Recogn., vol. 25, **(1992)**, pp. 567.
[3] T. X. Jun, Commun. Nonlinear Sci. Numer. Simulat., vol. 18, **(2013)**, pp. 1725.
[4] F. Chong, M. W. Hong and Z. Y. Feng, Comput. Biol. Med., vol. 43, **(2013)**, pp. 1000.
[5] K. T. Lin, Applied Optics, vol. 49, **(2012)**, pp. 220.
[6] S. M. Rahman, M. A. Hossain, H. Moufta, A. E. Saddik and E. Okamoto, Multimedia Syst., vol. 18, **(2012)**, pp. 145.
[7] H. Sadreazami and M. Amini, AEU Int. J. Electron. Commun., vol. 66, **(2012)**, pp. 364.
[8] A.Sleit, S. Abusharkh and R. Etoom, Imag. Sci. J., vol. 60, **(2012)**, pp. 29.
[9] Y. G. Dong, Pattern Recogn. Lett., vol. 31, **(2010)**, pp. 347.
[10] J. W. Yoon and K. Hyoungsgick, Comm. Nonlinear Sci. Numer. Simul., vol. 15, **(2010)**, pp. 3998.
[11] C. K. Huang and H. H. Nien, Opt Commun, vol. 282, **(2009)**, pp. 2123.
[12] W. X. Yuan, T. Lin and Q. Xue, Signal Process, vol. 92, **(2012)**, pp. 1101.
[13] C. G. Hui, H. Kai and T. Wei, Acta Phys. Sin., vol. 60, **(2011)**, pp. 110508.
[14] S. L.Yuan, C. L. Ling and S. K. Hui, Acta Phys. Sin., vol. 54, **(2005)**, pp. 4031.
[15] W. Lei, W. F. Ping and W. Z. Ji, Acta Phys. Sin., vol. 55, **(2006)**, pp. 3964.
[16] S. L. Yuan, X. Y. Yu and S. Zhe, Acta Phys. Sin., vol. 57, **(2008)**, pp. 4007.
[17] F. Zhang, X. J. Tian, J. Y. Song and X.Y. Li, J. China Univ. Post Telecom., vol. 15,**(2008)**, pp. 64.
[18] S. H. Song, Z. C. Sheng and Y. Y. Sheng, J. Tsinghua Univ (Sci and Tech), vol. 51, **(2011)**, pp. 1269.
[19] D. Knuth, Editor, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, Third Edition, Reading, Massachusetts: Addison-Wesley, **(1997)**.

Authors

Cao Guanghui, he received the M.S. degree from Northeastern University, China, in 2004, and he's now working towards his Ph.D in Beihang University, China. His research interests include algorithm design, security protocol and chaotic image encryption.

Hu Kai, is currently an associate Professor, Beihang University, China. He received Ph.D. degree in 2001 from Beihang University, He was a Research Fellow from 2001 to 2003, School of Computer Engineering, Nanyang Technological University, Singapore. Now his research interests include high performance parallel computing, network technology, and image encryption.

Zhou Jun, is currently a Professor, Electronic & Information Engineering College, Liaoning university of Technology. She received Ph.D. degree in 2007 from Northeastern University. Now her research interests include artificial intelligent, and data security.