A New Chaos-Based Image Encryption Algorithm

Ming Xu
Department of Mathematics and Physics, Shijiazhuang Tiedao University, China

Abstract: *In this paper, we propose a new image encryption algorithm based on the Compound chaotic image encryption algorithm. The original one can't resist chosen-plaintext attack and has weak statistical security, but our new algorithm can resist the chosen-plaintext attack using a simple improvement solution. The improvement solution is novel and transplantable, it can also be used to enhance the ability of resisting differential attack of other image encryption algorithms. The experimental results show that the new algorithm has higher security but its encryption speed is very nearly the same as the original one.*

Keywords: *Chaotic; image encryption; chosen-plaintext attack; transplantable.*

Received July 7, 2015; accepted January 13, 2016

1. Introduction

An image encryption scheme based on a compound chaotic sequence was proposed in [7]. The security of this scheme is studied and two main problems are found in [2]:

1. The compound chaotic sequence does not work as a good random number resource.
2. A differential chosen-plaintext attack can break the scheme with only three chosen plain-images.

Then it has been declared in [2] that the scheme in [7] is not secure enough, so it is not recommended for applications requiring a high level of security. In fact, the scheme in [7] has some merits, for example, it uses chebyshev map for confusion and diffusion, and the permutation is constructed of horizontal shifts and vertical shifts dynamical, all these are worthy of our reference.

In fact, many image encryption algorithms [4, 8, 9] have weak ability of resisting differential attack [3, 5, 6, 10]. The convential improvement solution is increasing the rounds of encryption, such as [1], but this solution will increase the running time and computation complexity enormously. Our improvement solution can also solve the problem of differential attack by simply increasing one key and (M+N) XOR operations. So our solution is more efficient. Moreover, our solution is also transplantable, it can be used to modify other encryption algorithms which easily be attacked by differential attack.

The outline of the Paper is as follows: In the next section we describe the original encryption scheme and corresponding attack scheme briefly, meanwhile analyze the reason why the original scheme easily be attacked. In section 3, based on the original one, we propose a new algorithm which can resist the chosen-paintext attack and has other good cryptographic properties. In Section 4, we give security analysis of the two schemes by simulation experiments. Finally, we give concluding remarks.

2. The Related Work

2.1. The Original Encryption Algorithm

Denoting the plain-image by I=$\{I(i,j)\}_{\substack{1\le i\le M\\ 1\le j\le N}}$ and the corresponding cipher-image by I′=$\{I'(i,j)\}_{\substack{1\le i\le M\\ 1\le j\le N}}$, the image encryption scheme proposed in [7] can be described as follows.

1. The secret keys include two floating-point numbers of precision 10^{-14}, $x_0, y_0 \in [-1,1]$, which are the initial states of the following two Chebyshev maps $f_0(x)=8x^4-8x^2+1$ and $f_1(y)=4y^3-3y$.
2. The initialization procedure includes generation of three pseudo-random integer sequences: Pseudo-random sequence $\{S_1(k)\}_{k=1}^{MN}$ for XOR substitution of pixel values; Pseudo-random sequence $\{S_2(i)\}_{i=1}^{M}$ for circular shift operations of rows; Pseudo-random sequence $\{S_3(j)\}_{j=1}^{N}$ for circular shift operations of columns (the specific process is obtainable in [7]).
3. The encryption procedure includes an XOR substitution part and two permutation parts.
   a) XOR substitution part
   Taking I as input, an intermediate image I^*={$I^*(i,j)$} is obtained as

$$I^*(i,j) = I(i,j) \oplus S_1((i-1)\bullet N + j) \qquad (1)$$

   Where $\oplus$ denotes the bitwise XOR operation.
   b) Permutation part-horizontal circular shift operations

Taking I^* as input, a new intermediate image $I^{**}=\{I^{**}(i,j)\}_{1\le i\le M,1\le j\le N}$ is obtained by performing the following horizontal circular shift operations:

$$I^{**}(i,j)=I^*(i,(j-S_2(i))\bmod N) \tag{2}$$

c) Permutation part–vertical circular shift operations.
Taking I^{**} as input, the cipher-image I' is obtained finally by performing the following vertical circular shift operations:

$$I'(i,j)=I^{**}((i-S_3(j))\bmod M,j) \tag{3}$$

Combining the three operations above, the encryption procedure can be represented in the following compact form:

$$I'(i,j)=I(i^*,j^*)\oplus S_1((i^*-1)\bullet N+j^*) \tag{4}$$

Where
$j^*=(j-S_2(i))\bmod N,\ i^*=(i-S_3(j^*))\bmod M$.

4. The decryption procedure is the reversion of above.

2.2. Differential Chosen-Plaintext Attack to the Original Algorithm

It is found in [2] that the algorithm above can be broken with only three chosen plain-images. The proposed attack is based on the following fact: given two plain-images I_1, I_2 and the corresponding cipher-images I_1', I_2', one can easily verify that

$$I_1'(i,j)\oplus I_2'(i,j)=I_1(i^*,j^*)\oplus I_2(i^*,j^*) \tag{5}$$

Where $j^*=(j-S_2(i))$ mod N, $i*=(i-S_3(j^*))$ mod M.
Then the attacker choose I_1 and I_2 as

$$I_1(i,:)\oplus I_2(i,:)=\begin{cases}0, & i=1\\ 255, & 2\le i\le M\end{cases} \tag{6}$$

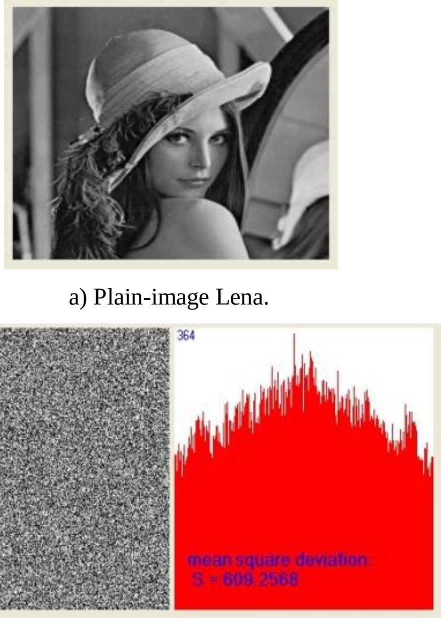
He can immediately derive all values of $\{S_3(j)\}_{j=1}^N$ using Equation (6), then the vertical shift operations are broken.

After that the attacker continue to choose I_1 and I_3 as:

$$I_1(:,j)\oplus I_3(:,j)=\begin{cases}0, & j=1\\ 255, & 2\le j\le N\end{cases} \tag{7}$$

He can immediately derive all values of $\{S_2(i)\}_{i=1}^M$.

The simulation results of the attack are shown in [2].Furthermore, in our Paper, we also study the histogram of the encrypted image in the original algorithm, which is shown in Figure 1, the plain-image is Lena. We find the variance is 609.2568, It is too big, so the algorithm may easily be attacked by statistical analysis. The result of histogram is totally different from that in [7]. But it has been declared in [2] that the randomness of the compound chaotic sequence in [7] is insufficient. So our result is fit in with [2], we can conclude that the corresponding conclusion in [7] is not accurate.



a) Plain-image Lena.

b) Encrypted image and its histogram.

Figure 1. The histogram of encryted image in the original algorithm.

From the description above, we can see the reason why the original scheme easily be attacked:

1. The substitution and permutation of pixels are independent of the plain-image.
2. The compound chaotic sequence used for substitution doesn't work as a good random number resource. Based on this consideration, we put forward our new algorithm.

3. The New Image Cryptosystem

In our new algorithm, we make the encryption process dependent on the plain-image, besides, we also exploit Chen's chaotic system to produce the substitution sequence instead of the compound chaotic sequence.

3.1. The Secret Keys

The secret keys of the new algorithm include: $x_0, y_0\in[-1,1]$ which are the initial values of the following two Chebyshev maps:
$f_0(x)=8x^4-8x^2+1$ and $f_1(y)=4y^3-3y$ respectively, x_1, y_1, z_1 which are the initial values of Chen's chaotic system and an initial key u_0.

3.2. The Initialization Procedure

The initialization procedure includes generation of three pseudo-random integer sequences.

1. Pseudo-random sequence $\{S_1(k)\}_{k=1}^{MN}$ for XOR

substitution of pixel values.

Using Runge-Kutta step size 0.001, iterate Chen's chaotic system N_0=(M×N)/3 times and obtain the real values x_i, y_i, z_i, $1 \le i \le N_0$. Then obtain the XOR sequence $\{S_1(k)\}_{k=1}^{MN}$ as

$$\begin{aligned} S_1(3(i-1)+1) &= \left| x_i - \lfloor x_i \rfloor \right| \times 10^{14} \bmod 256, \\ S_1(3(i-1)+2) &= \left| y_i - \lfloor y_i \rfloor \right| \times 10^{14} \bmod 256, \\ S_1(3(i-1)+3) &= \left| z_i - \lfloor z_i \rfloor \right| \times 10^{14} \bmod 256, \end{aligned} \tag{8}$$

Where $\lfloor x \rfloor$ denotes the largest integer not larger than x. Here, we assume that the encryption setup represents the real numbers with 14 decimal digits after the point.

2. Pseudo-random sequence $\{S_2(i)\}_{i=1}^{M}$ for circular shift operations of rows

Iterate f_0 from x_0 for M more times to obtain a chaotic sequence $\{x_i\}_{i=1}^{M}$, and then transform it into $\{S_2(i)\}_{i=1}^{M}$ by

$$S_2(i) = \begin{cases} \left[\dfrac{1+x_i}{2} \bullet N \right], & x_i \in [- \ 1, \\ N-1 & x_i = 1 \end{cases} \tag{9}$$

3. Pseudo-random sequence $\{S_3(j)\}_{j=1}^{N}$ for circular shift operations of columns.

Iterate f_1 from y_0 for N more times to obtain a chaotic sequence $\{y_j\}_{j=1}^{N}$, and then transform it into $\{S_3(j)\}_{j=1}^{N}$ by

$$S_3(j) = \begin{cases} \left[\dfrac{1+y_j}{2} \bullet M \right], & y_j \in [- \ 1, \\ M-1 & y_j = 1 \end{cases} \tag{10}$$

3.3. The Encryption Algorithm

The encryption procedure also includes an XOR substitution part and two permutation parts as in the original algorithm, but the sequence in the three parts is different from the original one, the new sequence is more conducive to resisting differential attack.

1. Permutation part-horizontal circular shift operations. Take plain-image I={m_{ij}|$1 \le i \le M$,$1 \le j \le N$} as input, where m_{ij} denotes the pixel of I located at the i^{th} row and j^{th} colunmn. At start, assign 1 to i, and convert the initial key u_0 to unsigned integer u in the range of 0 to 255 using mod operation, subsequently assign S_2(1) to variation d (d denotes the step length of horizontal-shift), then for each row of the image I (from i=1 to i=M), repeat the following operations:

   a) Construct the sequence $R_1 = \{R_1(j)\}_{j=1}^{N+1}$, where $R_1(1)=u$, $R_1(j+1)=m_{ij}$ $1 \le j \le N$, then do horizontal shift at R_1 by d pixels , we can obtain a new sequence $R_2 = \{R_2(j)\}_{j=1}^{N+1}$ as $R_2(j)=R_1((j-d)mod(N+1))$.

   b) Get the i^{th} row of intermediate image I^* $= \{m_{ij}^* \mid 1 \le i \le M, 1 \le j \le N\}$ as $m_{ij}^* = R_2(j) \quad j = 1,\dots,N$, Meanwhile assign the last pixel $R_2(N+1)$ to u, assign $m_{iN}^* \oplus S_2(i+1)$ to d;

   c) i=i+1, if $i \le M$, go back to 1; if $i > M$, jump out of the loop.

2. XOR substitution part
   Taking I^* as input, another intermediate image $I^{**} = \{m_{ij}^{**}\}_{1 \le i \le M, 1 \le j \le N}$ is obtained as:
   $m_{ij}^{**} = m_{ij}^* \oplus S_1((i-1) \bullet N + j) \quad 1 \le i \le M, 1 \le j \le N$.

3. Permutation part - vertical circular shift operations
   Taking I^{**} as input, at start, assign 1 to j, and assign $m_{MN}^{**} \oplus S_3(1)$ to variation d (d denotes the step length of vertical-shift now), then for each column of the image I^{**} (from j=1 to j=N), repeat the following operations:

   a) Construct the sequence $C_1 = \{C_1(i)\}_{i=1}^{M+1}$, where $C_1(1)=u$ (u is the variation which has appeared in (1), at present, the value of u is that in the last step of (1)), $C_1(i+1) = m_{ij}^{**} \quad 1 \le i \le M$, then do vertical shift at C_1 by d pixels , we can obtain a new sequence $C_2 = \{C_2(i)\}_{i=1}^{M+1}$ as $C_2(i)=C_1((i-d)$ mod $(M+1))$

   b) Get the jth column of the final cipher image I' $= \{c_{ij} \mid 1 \le i \le M, 1 \le j \le N\}$ as $C_{ij}=C_2(i)$ i=1,…,M, Meanwhile assign the last pixel $C_2(M+1)$ to u, assign $c_{iM} \oplus S_3(j+1)$ to d;

   c) j=j+1. if $j \le N$, go back to 1; if $j > N$, jump out of the loop.

At the end of the encryption process, we can obtain a cipher image I' and a ciphertext u.

3.4. The Decryption Algorithm

The decryption procedure is the reversion of above (after finishing the same initialization process) and can be described as:

1. Permutation part-vertical circular shift operations Taking cipher image I' $= \{c_{ij} \mid 1 \le i \le M, 1 \le j \le N\}$ and ciphertext u as input, at start, assign N to j, then for each column of the cipher image I' (from j=N to j=2), repeat the following operations:

a) Construct the sequence $C_2 = \{C_2(i)\}_{i=1}^{M+1}$, where $C_2(M+1)=u$, $C_2(i) = c_{ij} \; 1 \le i \le M$, compute $d = c_{M(j-1)} \oplus S_3(j)$, then do vertical shift at C_2 by

d pixels in the direction opposite to the encryption's, we can obtain a new sequence $C_1=\{C_1(i)\}_{i=1}^{M+1}$ as $C_1(i)$=$C_2((i+d)\mathrm{mod}(M+1))$;

b) Get the jth column of the intermediate image $I^{**}=\{m_{ij}^{**}\,|1\le i\le M,1\le j\le N\}$ as $m_{ij}^{**}=C_1(i+1)\quad i=1,...,M$, Meanwhile assign the first pixel $C_1(1)$ to u;

c) j=j-1. if j>1, go back to 1, if j=1, jump out of the loop and go to 4;

d) Construct the sequence $C_2=\{C_2(i)\}_{i=1}^{M+1}$, where $C_2(M+1)=u$, $C_2(i)=c_{i1}\ 1\le i\le M$, compute $d=m_{MN}^{**}\oplus S_3(1)$, then do vertical shift at C_2 by d pixels in the direction opposite to the encryption's, we can obtain a new sequence $C_1=\{C_1(i)\}_{i=1}^{M+1}$ as $C_1(i)$=$C_2((i+d)\mathrm{mod}(M+1))$;

Then we can get the first column of the intermediate image I^{**} as $m_{i1}^{**}=C_1(i+1)\quad i=1,...,M$, Meanwhile assign the first pixel $C_1(1)$ to u.

2. XOR substitution part

Taking I^{**} as input, another intermediate image $I^{*}=\{m_{ij}^{*}\}_{1\le i\le M,1\le j\le N}$ is obtained as:

$$m_{ij}^{*}=m_{ij}^{**}\oplus S_1((i-1)\bullet N+j)\quad 1\le i\le M,1\le j\le N$$

3. Permutation part - horizontal circular shif operations

Take image $I^{*}=\{m_{ij}^{*}\,|1\le i\le M,1\le j\le N\}$ and variation u as input (u is the variation which has appeared in (1), at present, the value of u is that in the last step of (1)). At start, assign M to i, then for each row of the image I^{*} (from i=M to i=2), repeat the following operations:

a) Construct the sequence $R_2=\{R_2(j)\}_{j=1}^{N+1}$, where $R_2(N+1)=u$, $R_2(j)=m_{ij}^{*}\ 1\le j\le N$, compute $d=m_{(i-1)N}^{*}\oplus S_2(i)$. Then do horizontal shift at R_2 by d pixels in the direction opposite to the encryption's, we can obtain a new sequence $R_1=\{R_1(j)\}_{j=1}^{N+1}$ as $R_1(j)$=$R_2((j+d)\bmod(N+1))$

b) Get the i^{th} row of plain-image I $=\{m_{ij}\,|1\le i\le M,1\le j\le N\}$ as m_{ij}=$R_1(j+1)$ j=1,…..,N, Meanwhile assign the first pixel $R_1(1)$ to u;

c) i=i-1. if i>1, go back to 1, if i=1, jump out of the loop and go to 4;

d) Construct the sequence $R_2=\{R_2(j)\}_{j=1}^{N+1}$, where $R_2(N+1)=u$, $R_2(j)=m_{1j}^{*}\ 1\le j\le N$. Then do horizontal shift at R_2 by $S_2(1)$ pixels in the direction opposite to the encryption's, we can obtain a new sequence $R_1=\{R_1(j)\}_{j=1}^{N+1}$ as $R_1(j)$=$R_2((j+d)\bmod(N+1))$, Then we can get the first row of plain-image I as m_{1j}=$R_1(j+1)$ j=1,…..,N.

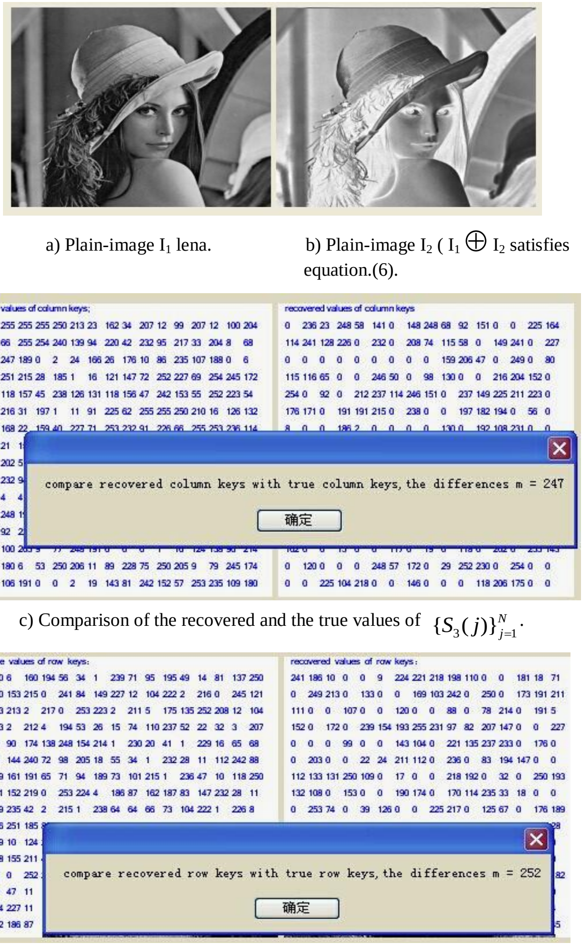
At the end of (3), we can recover the plain-image I.

4. Security Analysis of the Two Algorithms

This section provides the simulation results of the two algorithms. In those simulation experiments, we choose an 256×256 image Lena as object. we set the secret keys of the two algorithms as: the initial values of the two Chebyshev maps $f_0(x)$ and $f_1(y)$ are x_0=0.321456, y_0=0.481244; the initial values of Chen's chaotic system are x_1=－10.058, y_1=0.368, z_1=37.368, the parameter c is 28; besides, the initial key u_0 of the new algorithm is 112.

4.1. The Ability of Resisting Chosen-Plaintext Attack

From section 2, we know that the original algorithm can't resist chosen-plaintext attack, the main reason is (4), where the permutation procedure is independent of the image. However, in our new algorithm, the permutation procedure is dependent of the image, so (4) is no longer tenable, the chosen- plaintext attack in [2] is failed to our new algorithm.



a) Plain-image I_1 lena.

b) Plain-image I_2 ($I_1 \oplus I_2$ satisfies equation.(6).

```
true values of column keys;
63  255 255 255 250 213 23  162 34  207 12  99  207 12  100 204
7   66  255 254 240 139 94  220 42  232 95  217 33  204 8   68
255 247 189 0   2   24  166 26  176 10  86  235 107 188 0   6
54  251 215 28  185 1   16  121 147 72  252 227 69  254 245 172
15  118 157 45  238 126 131 118 156 47  242 153 55  252 223 54
251 216 31  197 1   11  91  225 62  255 255 250 210 16  126 132
114 168 22  159 40  227 71  253 232 91  226 66  255 253 236 114
168 21  1
32  202 5
42  232 9
181 4   4
51  248 1
232 92  2
12  100
27  180 6   53  250 206 11  89  228 75  250 205 9   79  245 174
13  106 191 0   0   2   19  143 81  242 152 57  253 235 109 180
```

```
recovered values of column keys
0   236 23  248 58  141 0   148 248 68  92  151 0   0   225 164
114 241 128 226 0   232 0   208 74  115 58  0   149 241 0   227
0   0   0   0   0   0   0   0   0   159 206 47  0   249 0   80
115 116 65  0   0   246 50  0   98  130 0   0   216 204 152 0
254 0   92  0   212 237 114 246 151 0   237 149 225 211 223 0
176 171 0   191 191 215 0   238 0   0   197 182 194 0   56  0
8   0   0   186 2   0   0   0   0   130 0   192 108 231 0   0
0   120 0   0   0   248 57  172 0   29  252 230 0   254 0   0
0   0   225 104 218 0   0   146 0   0   0   118 206 175 0   0
```

```
compare recovered column keys with true column keys, the differences m = 247
```

c) Comparison of the recovered and the true values of $\{S_3(j)\}_{j=1}^{N}$.

```
true values of row keys:
210 6   160 194 56  34  1   239 71  95  195 49  14  81  137 250
170 153 215 0   241 84  149 227 12  104 222 2   216 0   245 121
253 213 2   217 0   253 223 2   211 5   175 135 252 208 12  104
223 2   212 4   194 53  26  15  74  110 237 52  22  32  3   207
13  90  174 138 248 154 214 1   230 20  41  1   229 16  65  68
82  144 240 72  98  205 18  55  34  1   232 28  11  112 242 88
169 161 191 65  71  94  189 73  101 215 1   236 47  10  118 250
171 152 219 0   253 224 4   186 87  162 187 83  147 232 28  11
109 235 42  2   215 1   238 64  64  66  73  104 222 1   226 8
136 251 185
209 10  124
248 155 211
38  0   252
59  47  11
254 227 11
162 186 87
```

```
recovered values of row keys:
241 186 10  0   0   9   224 221 218 198 110 0   0   181 18  71
0   249 213 0   133 0   0   169 103 242 0   250 0   173 191 211
111 0   0   107 0   0   120 0   0   88  0   78  214 0   191 5
152 0   172 0   239 154 193 255 231 97  82  207 147 0   0   227
0   0   0   99  0   0   143 104 0   221 135 237 233 0   176 0
0   203 0   0   22  24  211 112 0   236 0   83  194 147 0   0
112 133 131 250 109 0   17  0   0   218 192 0   32  0   250 193
132 108 0   153 0   0   190 174 0   170 114 235 33  18  0   0
0   253 74  0   39  126 0   0   225 217 0   125 67  0   176 189
```

```
compare recovered row keys with true row keys, the differences m = 252
```

d) Comparison of the recovered and the true values of $\{S_2(i)\}_{i=1}^{M}$.

Figure 2. The ability of resisting chosen-plaintext attack.

The simulation results which show the ability of the new algorithm to resist chosen-plaintext attack are shown in Figure 2. I_1 is the plain-image Lena, I_2 is another image satisfying Equation (6). Using the differential attack algorithm in [2] to attack our new algorithm, we find that we can't obtain the correct keys (see (c) and (d)).

4.2. Histogram Analysis

An image-histogram illustrates how pixels in an image are distributed by graphing the number of pixels at each color intensity level. We have calculated and analyzed the histogram of the encrypted image in original algorithm (see section 2), we can see the variance is too big, so the algorithm may easily be attacked by statistical analysis. However, the histogram in our new algorithm is fairly uniform and significantly different from the histogram of plain-image Lena (see Figure 3), so it does not provide any clue to employ any statistical attack on the new encryption procedure.

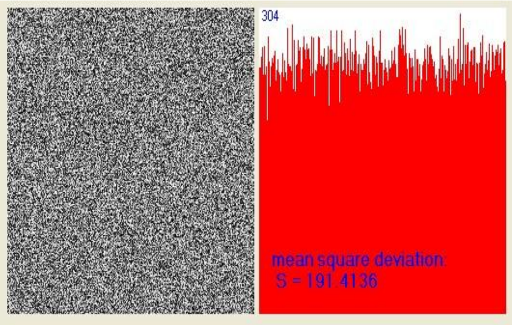


Figure 3. The histogram of encrypted image (plain-image is lena) in new algorithm.

4.3. Information Entropy Analysis

For testing the robustness of the encryption algorithm, the concept of entropy is also used. Higher the value of entropy of encrypted image, better the security. The information entropy for the encrypted image Lena in original algorithm and our new algorithm are 7.948901 and 7.997121 respectively, the latter is very near to the max of entropy value 8. It indicates that in the encrypted image all the pixels occur with almost equal probability, thus proves the ability of our new algorithm against the entropy attack.

4.4. Differential Analysis

The aim of this analysis is to determine the sensitivity of the encryption algorithm to slightest changes. Two criteria Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are used to test the sensitiveness. NPCR is defined as the percentage of different pixel numbers between two encrypted images, whose plain images have only one pixel difference. UACI is defined as the average intensity of differences between 2 cipher images, corresponding to plain images that have only one pixel difference. The high values of these two parameters indicate that small change in plain image creates significant change in the cipher image. In the simulation experiment we choose the image Lena as the plain-image, the values of NPCR and UACI in the original algorithm are 0.0015% and 2.99E - 05% respectively, so it is easily attacked by differential attack. But the two values in our new algorithm are 99.64% and 33.46% respectively, hence the new algorithm is highly resistive against differential attack.

4.5. Image Encryption Speed Test

We have compared the original algorithm and our new algorithm in encryption speed. An 256×256 image Lena is encrypted, the average running time of 10 encryption procedures in the original algorithm is 87.355 milliseconds, the average running time of 10 encryption procedures in the new algorithm is 97.114 milliseconds. The results show that the encryption speed in the new algorithm is very close to the original's, although our new algorithm has higher security.

4.6. The Comparison of Our Improvement Solution with the Conventional One

One can also utilize the conventional improvement solution to raise the values of NPCR and UACI, i.e. increasing the rounds of encryption. The simulation results show that if one wants to obtain the NPCR as 99.6231%, he needs 3 rounds of encryption, approximately 208.7782 milliseconds. So it is easily seen that our improvement solution is more efficient.

5. Conclusions

In this paper we propose a new image encryption scheme based on the Compound chaotic image encryption scheme in [7]. The original one can't resist chosen-plaintext attack, but our new algorithm can by using a simple and novel improvement solution. The conclusion has been verified by simulation experiments. The experimental results also show that the cipher has good randomness, and it can resist statistical analysis. In addition, compared to the original scheme, the running time of the new algorithm is not obviously increased, and compared to the conventional improvement solution, our improvement method is more efficient. Especially, our modified method is transplantable, it can be also used to modify other image encryption schemes that are easily exposed to differential attack.

References

[1] Arthanari S., Mastan M., and BaganK B., "Chaotic Image Encryption using Modular Addition and Combinatorial Techniques," *The*

International Arab Journal of Information Technology, vol. 12, no. 2, pp.110-117, 2015.

[2] Li C., Li S., Chen G., and Halang W., "Cryptanalysis of an Image Encrypion Scheme Based on A Compound Chaotic Sequence," *Image and Vision Computing*, vol. 27, no. 8, pp.1035-1039, 2009.

[3] Lian S., Sun J., and Wang Z., "Security Analysis of A Chaos-Based Image Encryption Algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 351, no. 2-4, pp. 645-661, 2005.

[4] Liu L., Zhang Q., and Wei X., "A RGB Image Encyption Algorithm Based on DNA Encoding and Chaos Maps," *Computers and Electrical Engineering*, vol. 38, no. 5, pp. 1240-1248, 2012.

[5] Ozkaynak F., Ozer A., and Yavuz S., "Security Analysis of an Image Encryption Algorithm Based on Chaos and DNA Encoding," *in Proceedings of Signal Processing and Communications Applications Conference*, Haspolat, pp. 1-4, 2013.

[6] Sun S., *Image Encryption Algorithms and Practices with Implementations in C#*, Science Press, 2013.

[7] Tong X. and Cui M., "Image Encryption with Compound Chaotic Sequence Cipher Shifting Dynamically," *Image and Vision Computing*, vol. 26, no. 6, pp. 843-850, 2008.

[8] Zhen W., Xia H., Yu-Xia L., and Xiao-Na S., "A New Image Encryption Algorithm Based on the Fractional-Order Hyperchaotic Lorenz System," *Chinese Physics B*, vol. 22, no. 1, pp. 28-35, 2013.

[9] Zhang Q., Guo L., and Wei X., "A novel Image Fusion Encryption Algorithm Based on DNA Sequence Operation and Hyper-Chaotic System," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 18, pp. 3596-3600, 2013.

[10] Zhang Y., Wen W., Su M., and Li M., "Cryptanalyzing A Novel Image Fusion Encryption Algorithm Based on DNA Sequence Operation and Hyper-Chaotic System," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 4, pp. 1562-1564, 2014.

Ming Xu, Female, Born in china, 1981; Master in Applied Mathematics, Northeastern University, ShenYang, China, the degree was earned in 2006; The main research fields: cryptography.