Cryptography based on Spatiotemporal Chaos System and Multiple Maps

Jun He
Department of Computer Science
East China Normal University
Shanghai, China
Email: hallyhejun@163.com

Zhi-bin Li and Hai-feng Qian
Department of Computer Science
East China Normal University
Shanghai, China
Email: {lizb, hfqian}@cs.ecnu.edu.cn

Abstract—In this paper, a stream image cryptography has been proposed based on spatiotemporal chaos system. It is specifically designed for the colored images, which are 3D arrays of data streams. One-way coupled map lattices (OCML) are used to generate pseudorandom sequences and then encrypt image pixels one by one. By iterating randomly chosen chaotic maps from a set of chaotic maps for certain times, the generated pseudorandom sequences obtain high initial-value sensitivity and good randomness. The initial conditions of chaotic maps and parameters of the algorithm are generated by a 128-bit external key. The results of several experimental, statistical analyses and key sensitivity tests prove the security robustness of the proposed cryptosystem.

Index Terms—Color image cipher; Chaos; Multiple chaotic maps; OCML

I. INTRODUCTION

With the rapid growth of multimedia production systems, electronic publishing and widespread dissemination of digital multimedia date over the Internet, protection of digital information against illegal copying and distribution has become extremely important. However, the traditional ciphers such as IDEA, AES, DES and RSA are not suitable for real time encryption especially for image and video because these ciphers require a large computational time and high computing power. Chaos-based encryption appeared in the early 1990s as an original application of nonlinear dynamics in the chaotic regime, which is deterministic but simple. Therefore, it can provide a fast and secure means for data protection, which is crucial for multimedia data transmission through fast communication channels. Chaotic system is distinguished by its ergodicity and sensitivity to initial conditions and system parameters, and which can be considered analogous to some ideal cryptographic properties such as confusion, diffusion, balance, avalanche properties, etc. Those attributes allow the chaotic dynamics to be a promising alternative for the conventional cryptographic algorithms.

The use of chaos for image encryption has been proposed in [1-4]. In most of the digital encryption techniques developed for the document/text encryption, one-dimensional chaotic maps have been used [5-8] as the digital documents can be considered as the one-dimensional data stream, which can be encrypted block-by-block (in block ciphers) or bit-by-bit (in stream ciphers) with an appropriate choice of one-dimensional chaotic system. Baptista [5] designed an algorithm to encrypt a text message using the ergodic property of a simple low-dimensional and chaotic logistic map. The basic idea is to encrypt each character of the message as the integer number of iterations performed in the logistic equation, in order to transfer the trajectory from an initial condition towards an \( \varepsilon \) –interval inside the logistic chaotic attractor. Pareek [10] proposed an approach for image encryption based on a chaotic logistic map. In the encrypted scheme, an external secret key of 80-bit and two chaotic logistic maps are employed. The initial conditions for the both logistic maps are derived using the external secret key. Further, eight different types of operations are used to encrypt the pixels of an images and which one of them will be used for a particular pixel is decided by the outcome of the logistic map.

To manage the trade offs between the security and speed, various image encryption techniques based on multiple one-dimensional chaotic maps, two-dimensional or higher-dimensional chaotic systems, spatiotemporal chaotic systems, couple map lattices (CML), etc. which have been proposed in [2,4,10-19]. Fridrich [1] demonstrated the construction of a symmetric block encryption technique based on the two-dimensional standard baker map. In the algorithm, the substitution is achieved by permuting all the pixels as a whole using a two-dimensional chaotic map. The new pixel...
moved to the current position is taken as the substitution of the original pixel. In the diffusion process, the pixel values are altered sequentially and the change made to a particular pixel depends on the accumulated effect of all the previous pixel values. This substitution-diffusion architecture formed the basic structure for many of the chaos based image encryption techniques. Pareek [9] has developed a symmetric key block cipher algorithm using multiple one-dimensional chaotic maps. In the cryptosystem, plaintext is divided into groups of variable length and these are encrypted sequentially by using randomly chosen chaotic map from a set of chaotic maps. Form block-by-block encryption of variable length group, number of iterations and initial condition for the chaotic maps depend on the randomly chosen session key and encryption of previous block of plaintext, respectively. It is a variation of another cryptosystem by the same authors [8]. Spatiotemporal chaos system [20] is regarded as of better properties suitable for data protection than one-dimensional chaos system, such as larger parameter space, better randomness and more chaotic sequences, etc. For example, the spatiotemporal chaos is used in a color image authentication scheme [21]. In the scheme, the spatiotemporal chaos is employed to randomly select the embedding positions for each watermark bit as well as for watermark encryption. The spatiotemporal chaos is also used in stream cipher designing [22, 23]. The one-way coupled map lattice system (OCML) [20, 24-27] is a spatiotemporal chaos system.

In this paper, a stream color cipher based on spatiotemporal chaos system has been constructed, and multiple one-dimensional chaotic maps are used to obtain high key sensitivity. The spatiotemporal chaotic system, which is modeled by one-way coupled map lattices (OCML), enhances the key sensitivity. The three components (Red, Green and Blue) of plain image are encrypted in a coupling fashion in such a way to strengthen the cryptosystem security. Furthermore, inclusion of more than one chaotic map increases the confusion in the encryption process and it results in a more secure cryptosystem. The initial conditions and parameters are generated by an external secret key of maximum 128-bits length. The decryption process is symmetric to the encryption process.

In Section 2, we will discuss procedure of image encryption and in Section 3, we will prove the security of the proposed image encryption scheme by performing statistical analysis, key and plaintext sensitivity analysis etc. Finally, in Section 4, we conclude the paper.

II. THE PROPOSED IMAGE ENCRYPTION SCHEME

In this section, we discuss the proposed color image encryption as well as decryption process with several one-dimensional chaotic maps and the OCML system. The color image of size $M \times N$ is converted into its RGB components. And then, each color matrix (R, G or B) is stored in a vector of integers within $\{0, 1, \ldots, 255\}$. Each vector has a length $L = M \times N$. The three vectors obtained $P(1), P(2), P(3)$ represent the RGB components of plaintext which will be encrypted.

### A. One-way coupled map lattice system (OCML)

Spatiotemporal chaos has its evident advantages in cryptography as it possesses a large number of positive Lyapunov exponents. Moreover, the problem of short period encountered by discretized chaotic maps is overcome since multiple chaotic systems behave like a random interference to each other. In the scheme, spatiotemporal chaos is based on one-way coupled map lattice model (OCML). The OCML used for the plaintext is given by

$$x_{n+1} = \lambda x_n (1 - x_n)$$

where $x_n$ is the variable state, $\lambda$ is the system parameter of $\lambda = 3.99$.

### B. Chaotic maps

Four one-dimensional chaotic maps are employed to make equation $f()$: logistic map, tent map, sine map and cubic map, and each chaotic map is identified by an integer index (map number $N$). The name, equation and system parameter value of each map used in our algorithm are given in Table 1.

<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
<th>Equation</th>
<th>System parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Logistic</td>
<td>$x_{n+1} = \lambda x_n (1 - x_n)$</td>
<td>$\lambda = 3.99$</td>
</tr>
<tr>
<td>1</td>
<td>Tent</td>
<td>$x_{n+1} = \begin{cases} \lambda x_n &amp; x_n &gt; 0.5 \ \lambda (1 - x_n) &amp; x_n \leq 0.5 \end{cases}$</td>
<td>$\lambda = 1.97$</td>
</tr>
<tr>
<td>2</td>
<td>Sine</td>
<td>$x_{n+1} = \lambda \sin(\pi x_n)$</td>
<td>$\lambda = 0.99$</td>
</tr>
<tr>
<td>3</td>
<td>Cubic</td>
<td>$x_{n+1} = \lambda x_n (1 - x_n^2)$</td>
<td>$\lambda = 2.59$</td>
</tr>
</tbody>
</table>

### C. Generation of the initial conditions and parameters

Based on the three components (Red, Green and Blue), there are four initial conditions ($X_0, X_1, X_2, X_3$) in equation (1) with one definitive chaotic map function. At the beginning, the four chaotic maps have the same initial conditions. With the encryption/decryption of each pixel of
TABLE II. SHOWING MAP NUMBER (N) AND NUMBER OF ITERATIONS FOR THE MAP(IT) OF EACH IMAGE PIXEL USING SECRET KEY 'ABCDEF0123456789ABCDEF0123456789'

<table>
<thead>
<tr>
<th>Pixel sequence</th>
<th>Map number (N)</th>
<th>Iterations number for map N (IT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>71</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

plaintext/ciphertext, the initial conditions for these maps are updated and further the updating depends on the outcome of the previous pixel of image.

Firstly, the proposed image encryption process utilizes a 128-bit external secret key (K). This key is divided into 8-bit blocks (k_i) referred to session keys. The 128-bit external secret-key (K) is given by

\[ K = k_1, k_2, \ldots, k_{16} \]  

The initial conditions for each chaotic map are the same and generated from the session keys as

\[ x_i(0) = (((k_1 \oplus k_2 \oplus k_3 \oplus k_4) + \sum_{j=1}^{16} (k_{j}^i)) / 2^8) \mod 1 \]

\[ x_i(1) = (((k_5 \oplus k_6 \oplus k_7 \oplus k_8) + \sum_{j=1}^{16} (k_{j+1}) / 2^8) \mod 1 \]

\[ x_i(2) = (((k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12}) + \sum_{j=1}^{16} (k_{j+2}) / 2^8) \mod 1 \]

\[ x_i(3) = (((k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16}) + \sum_{j=1}^{16} (k_{j+3}) / 2^8) \mod 1 \]  

(3)

When the secret key 'ABCDEF0123456789ABCDEF0123456789' is used, the parameters of chaotic maps for each pixel are given in Table 2. Each row shows encryption /decryption with one image pixel and three columns: the first column for the sequence number of the image pixel, the second for the map number (N) and the last for the number of chaotic map iterations (IT). Value of N and IT are calculated with the help of a congruential random number generator (LCG), which generates a sequence of random numbers with the following recurrence relation

\[ Y_{n+1} = (aY_n + c) \mod m \]  

(4)

where \( a, c \) and m are multiplier, increment and modulus, respectively. We set \( a = 16, b = 7 \) and \( m = 81 \) here. The value of the seed (Y_0) for the LCG, the map number (N) and iteration number (IT) described above are generated as follows

\[ Y_n = \sum_{i=1}^{16} (k_i)_{10} \mod 256 \]

\[ N_n = Y_n \mod 4 \]

\[ IT_n = (k_i)_{10} \mod 100, \quad (i = (Y_n \mod 16) + 1) \]  

(5)

D. Design of the encryption and decryption algorithm

1) Encryption

First, the three color vectors (P(1), P(2), P(3)) are read from the image file. Then, the values of the nth image pixel are represented as X_n(k).

Next, the encryption for the pixel \( P_n(k) \) is given by

\[ C_n(k) = (P_n(k) + \text{int}(x_{n}(k) \times L) + C_{n-1}(k)) \mod 256, \quad n = 0, 1, \ldots, L; k = 1, 2, 3. \]  

(6)

The rows of P being the R, G and B components, respectively, equation (6) can be rewritten as follows

\[ C_1(1) = (R_n + \text{int}(x_{1}(1) \times L) + C_{0}(1)) \mod 256 \]

\[ C_2(2) = (G_n + \text{int}(x_{2}(2) \times L) + C_{1}(2)) \mod 256 \]

\[ C_3(3) = (B_n + \text{int}(x_{3}(3) \times L) + C_{2}(3)) \mod 256 \]

\( x_n(k) \) is generated by the OCML system like equation (1). The initial conditions are given by equation (3) and we set the parameter \( \varepsilon_j = 0.5 \), for \( j = 1, 2, 3 \). Function \( f() \) is one of the four chaotic maps in Table 1. For the nth pixel, a chaotic map equation is chosen from Table 1 by the value of map number \( (N_n) \) and iterate \( (IT_n) \) times which are generated by equation (5). The two operations compose function \( f() \).

The key sequence of the OCML \( x_n(0) \) is given by

\[ x_n(0) = C_n(3) / 2^3 \]  

(7)

The stream cipher of color components depends on each other due to the coupling structure of the OCML. These characteristics strengthen the security of the encryption.

2) Decryption

In the receiver's side, the dynamics of the OCML is regenerated as equation (1) and the value of parameters are the same as the encryption.

The decryption transformation is

\[ P_n(k) = (C_n(k) - \text{int}(x_{n}(k) \times L) - C_{n-1}(k)) \mod 256, \]

\[ x_n(0) = C_n(3) / 2^3 \]

\( n = 1, \ldots, L; k = 1, 2, 3. \)  

(8)

Equation (8) can be rewritten to give pixels’ value in the RGB components as follows

\[ R_n = (C_n(1) - \text{int}(x_{1}(1) \times L) - C_{0}(1)) \mod 256 \]

\[ G_n = (C_n(2) - \text{int}(x_{2}(2) \times L) - C_{1}(2)) \mod 256 \]

\[ B_n = (C_n(3) - \text{int}(x_{3}(3) \times L) - C_{2}(3)) \mod 256 \]

Note that the receiver must have the same keystream which can decrypt the color image, so it has the same secret key \( K = k_1, k_2, \ldots, k_{16} \). Therefore, it is possible to generate
the same initial conditions and parameters \( x_j(j) \) and \( \varepsilon_j \), for \( j = 1, 2, 3 \).

III. PERFORMANCE ANALYSIS

A good encryption procedure should be robust against all kinds of cryptanalytic, statistical and brute-force attacks. In this section, the security includes statistical analysis, sensitivity analysis and differential attacks.

A. Statistical analysis

In order to resist statistical attacks, the encryption images should possess certain random properties. A detail study has been explored and the results are summarized. However, due to the page limit, only results for the Lena (Fig.1 a) are used for illustration.

1) Color histogram

In Fig.2, Frame (a), (b) and (c) show the histogram of RGB colors for original (Fig.1 a), Frame (e), (f) and (g) show the histogram of the encrypted images (Fig.1 b). The figure clearly presents the random-like appearance of the histogram of the encrypted images.

2) Correlation of adjacent pixels

For an ordinary image, each pixel is always highly correlated with its adjacent pixels either in horizontal or vertical directions. These high-correlation properties can be quantified as their correlation coefficients for comparison. The correlation between two vertically and horizontally adjacent pixels is analyzed in the several images and their encrypted images. For this calculation, the following two formulas are used

\[
\text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))
\]

\[
r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)} \sqrt{D(y)}}
\]

where \( x \) and \( y \) are the grey-scale value of two adjacent pixels in the image and \( N \) is the total number of pixels selected from the image for the calculation. In numerical computation, the following discrete formulas are used

\[
E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

\[
D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2
\]

In Table 3, the correlation coefficients of Lena image and those of its encrypted image with the secret key 'ABCDEF0123456789ABCDEF0123456789' are given. It is clear that there is negligible correlation between these two adjacent pixels in the encrypted image. However, the two adjacent pixels in the original image are highly correlated. This result is illustrated in Fig.3, which presents the distribution of two adjacent pixels in the original and encrypted images of Lena for horizontal (a, b) as well as vertical (c, d) directions.

Extensive study of the correlation between image and its corresponding encrypted image by using the proposed encryption algorithm also have been done. Some pictures of USC-SIPI image database (freely available at http://sipi.usc.edu/database/) have been chosen to measure the correlation coefficient. The encryption has been done using the secret key 'ABCDEF0123456789ABCDEF0123456789'. Results are shown in Table 4. The average
correlation coefficient is very small which implies that no correlation exist between original and its corresponding cipher images.

**B. Sensitivity analysis**

An ideal image encryption procedure should be sensitive with the secret key. It means that the change of a single bit in the secret key should produce a completely different encrypted image. (Fig.4 a) is the original image. In Fig.4, Frame (b), (c) and (d) illustrate the corresponding encrypted images using different secret keys 'ABCDEF0123456789AB', 'CDEF0123456789ABCDEF0123456789' and 'ABCDEF0123456789ABCDEF0123456780'. The correlation coefficients of the three encrypted images are calculated in Table 5.

Moreover, in Fig.5, the results of some attempts are shown to decrypt an encrypted image with slightly different secret keys from the one used for the encryption of the original image. In Table 6, the correlation coefficients between the decryption image with a slightly different key and its original image are shown. It can be observed that the decryption with a slightly different key fails completely. Therefore, the proposed image encryption procedure is highly key sensitive.

<table>
<thead>
<tr>
<th>File name</th>
<th>File description</th>
<th>Size</th>
<th>Correlation coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.01 Girl</td>
<td>256*256</td>
<td></td>
<td>0.01272</td>
</tr>
<tr>
<td>4.1.02 Couple</td>
<td>256*256</td>
<td></td>
<td>0.02716</td>
</tr>
<tr>
<td>4.1.03 Girl</td>
<td>256*256</td>
<td></td>
<td>-0.02593</td>
</tr>
<tr>
<td>4.1.04 Girl</td>
<td>256*256</td>
<td></td>
<td>-0.00803</td>
</tr>
<tr>
<td>4.1.05 House</td>
<td>256*256</td>
<td></td>
<td>-0.01453</td>
</tr>
<tr>
<td>4.1.06 Tree</td>
<td>256*256</td>
<td></td>
<td>0.00219</td>
</tr>
<tr>
<td>4.1.07 Jelly beans</td>
<td>256*256</td>
<td></td>
<td>-0.00869</td>
</tr>
<tr>
<td>4.1.08 Jelly beans</td>
<td>256*256</td>
<td></td>
<td>-0.01368</td>
</tr>
<tr>
<td>4.2.01 Splash</td>
<td>512*512</td>
<td></td>
<td>0.00819</td>
</tr>
<tr>
<td>4.2.02 Girl (Tiffany)</td>
<td>512*512</td>
<td></td>
<td>0.02036</td>
</tr>
<tr>
<td>4.2.03 Baboon</td>
<td>512*512</td>
<td></td>
<td>-0.02370</td>
</tr>
<tr>
<td>4.2.04 Girl (Lena)</td>
<td>512*512</td>
<td></td>
<td>-0.01323</td>
</tr>
<tr>
<td>4.2.05 Airplane (F-16)</td>
<td>512*512</td>
<td></td>
<td>-0.00369</td>
</tr>
<tr>
<td>4.2.06 Sailboat on lake</td>
<td>512*512</td>
<td></td>
<td>0.00101</td>
</tr>
</tbody>
</table>

**TABLE III. CORRELATION COEFFICIENTS FOR THE TWO ADJACENT PIXELS IN THE ORIGINAL AND ENCRYPTED IMAGE SHOWN IN FIG.1**

<table>
<thead>
<tr>
<th></th>
<th>Original image (Fig.1 a)</th>
<th>Encrypted image (Fig.1 b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.9720</td>
<td>-0.0260</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9851</td>
<td>-0.0005</td>
</tr>
</tbody>
</table>

**TABLE IV. CORRELATION COEFFICIENTS BETWEEN THE IMAGE AND CORRESPONDING CIPHER IMAGE FOR SOME IMAGES OF THE USC-SIPI IMAGE DATABASE. THE ENCRYPTION HAS BEEN DONE USING SECRET KEY 'ABCDEF0123456789AB'.**

**Figure 4.** Key sensitivity test I: Frame (a) show the Lena image, Frame (b), (c) and (d) show the encryption image of the Lena image using tiny different secret keys.

**TABLE V. CORRELATION COEFFICIENTS BETWEEN THE CORRESPONDING PIXELS OF THE THREE DIFFERENT ENCRYPTED IMAGES OBTAINED BY USING SLIGHT DIFFERENT SECRET KEY OF AN IMAGE SHOWN IN FIG.4.**

<table>
<thead>
<tr>
<th>Image1</th>
<th>Image2</th>
<th>Correlation coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted image (Fig.4 b)</td>
<td>Encrypted image (Fig.4 c)</td>
<td>-0.0029</td>
</tr>
<tr>
<td>Encrypted image (Fig.4 c)</td>
<td>Encrypted image (Fig.4 d)</td>
<td>-0.0003</td>
</tr>
<tr>
<td>Encrypted image (Fig.4 d)</td>
<td>Encrypted image (Fig.4 b)</td>
<td>-0.0010</td>
</tr>
</tbody>
</table>
Figure 5. Key sensitivity test II: Frame (a) and (b) show the Baboon image and its encrypted image by using the secret key 'ABCDEF0123456789ABCDEF0123456789'. Frame (c) and (d) show the images after the decryption of the encrypted image in Frame (b) using the secret key 'BBCDEF0123456789ABCDEF0123456789' and 'ABCDEF0123456789ABCDEF0123456789'.

C. Differential attack

NPCR [3] means the change rate of the number of pixels of ciphered image while the one pixel of plain-image is changed. UACI measures the average intensity of differences between the plain-image and cipher-image.

Denote two cipher-images, whose corresponding plain-images have only one-pixel difference, by $C_i$ and $C_j$, respectively. Define a two-dimensional array $D$, with the same size as image $C_i$ and $C_j$. Then, $D(i,j)$ is determined by $C_i(i,j)$ and $C_j(i,j)$, namely, if $C_i(i,j) = C_j(i,j)$ then $D(i,j) = 0$; otherwise, $D(i,j) = 1$.

NPCR and UACI are defined by the following formulas

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{WH} \times 100\%$$

$$\text{UACI} = \frac{1}{WH} \left[ \frac{\sum_{i,j} |C_i(i,j) - C_j(i,j)|}{255} \right] \times 100\% \tag{11}$$

Where $W$ and $H$ are the width and height of $C_i$ or $C_j$. Tests have been performed on the proposed algorithm to assess the influence of changing a single pixel in the original image on the encrypted image. The results of some pictures of USC-SIPI image database (freely available at http://sipi.usc.edu/database/) are shown in Table 7.

### Table VI. Correlation Coefficients between the Corresponding Pixels of the Original Image (Fig.5 a) and Decrypted Image (Fig.5 c and Fig.5 d).

<table>
<thead>
<tr>
<th>Image1</th>
<th>Image2</th>
<th>Correlation coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original image (Fig.5 a)</td>
<td>Decrypted image (Fig.5 c)</td>
<td>0.0298</td>
</tr>
<tr>
<td>Original image (Fig.5 a)</td>
<td>Decrypted image (Fig.5 d)</td>
<td>0.0185</td>
</tr>
</tbody>
</table>

### Table VII. Pixel Difference between Two Encrypted Images for Some Images of the USC-SIPI Image Database, Which From Two Original Images with One Bit Difference in the Corresponding Color Component Using Secret Key 'ABCDEF0123456789ABCDEF0123456789'.

<table>
<thead>
<tr>
<th>File name</th>
<th>File description</th>
<th>Size</th>
<th>NPCR(%)</th>
<th>NACI(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.02</td>
<td>Couple</td>
<td>256*256</td>
<td>R 99.6231 G 99.6170 B 99.5895</td>
<td>33.5460 R 33.5041 G 33.4899</td>
</tr>
<tr>
<td>4.1.03</td>
<td>Girl</td>
<td>256*256</td>
<td>R 99.6460 G 99.6124 B 99.6155</td>
<td>33.3853 R 33.3822 G 33.3026</td>
</tr>
<tr>
<td>4.1.04</td>
<td>Girl</td>
<td>256*256</td>
<td>R 99.6140 G 99.5636 B 99.5926</td>
<td>33.5092 R 33.4297 G 33.6353</td>
</tr>
<tr>
<td>4.1.05</td>
<td>House</td>
<td>256*256</td>
<td>R 99.6201 G 99.5926 B 99.6277</td>
<td>33.3606 R 33.5757 G 33.6396</td>
</tr>
<tr>
<td>4.1.06</td>
<td>Tree</td>
<td>256*256</td>
<td>R 99.6155 G 99.6124 B 99.5529</td>
<td>33.5199 R 33.5497 G 33.5070</td>
</tr>
<tr>
<td>4.1.07</td>
<td>Jelly beans</td>
<td>256*256</td>
<td>R 99.6384 G 99.6384 B 99.6693</td>
<td>33.4234 R 33.5336 G 33.5417</td>
</tr>
<tr>
<td>4.1.08</td>
<td>Jelly beans</td>
<td>256*256</td>
<td>R 99.6048 G 99.5850 B 99.5819</td>
<td>33.3579 R 33.5172 G 33.5509</td>
</tr>
<tr>
<td>4.2.01</td>
<td>Splash</td>
<td>512*512</td>
<td>R 99.6006 G 99.6315 B 99.6216</td>
<td>33.4799 R 33.5272 G 33.5377</td>
</tr>
<tr>
<td>4.2.02</td>
<td>Girl (Tiffany)</td>
<td>512*512</td>
<td>R 99.5979 G 99.6201 B 99.6246</td>
<td>33.5118 R 33.4584 G 33.5120</td>
</tr>
<tr>
<td>4.2.03</td>
<td>Baboon</td>
<td>512*512</td>
<td>R 99.6185 G 99.6097 B 99.6097</td>
<td>33.5581 R 33.3739 G 33.5061</td>
</tr>
<tr>
<td>4.2.04</td>
<td>Girl (Lena)</td>
<td>512*512</td>
<td>R 99.6304 G 99.6044 B 99.6330</td>
<td>33.4630 R 33.4434 G 33.4716</td>
</tr>
<tr>
<td>4.2.05</td>
<td>Airplane (F-16)</td>
<td>512*512</td>
<td>R 99.6014 G 99.6285 B 99.5899</td>
<td>33.5355 R 33.5483 G 33.4890</td>
</tr>
<tr>
<td>4.2.06</td>
<td>Sailboat on lake</td>
<td>512*512</td>
<td>R 99.6250 G 99.6002 B 99.6414</td>
<td>33.4523 R 33.5527 G 33.3859</td>
</tr>
</tbody>
</table>
D. Information entropy analysis

Information theory is a mathematical theory of data communication and storage founded by Shannon. It is well known that the entropy \( H(m) \) of a message source \( m \) can be calculated as

\[
H(m) = \sum_{m_i} p(m_i) \log \left( \frac{1}{p(m_i)} \right)
\]

where \( N \) is the number of bits to represent a symbol \( m_i \in m \). \( p(m_i) \) represents the probability of symbol \( m_i \), and \( \log() \) represents the base 2 logarithm so that the entropy is expressed in bits. For a truly random source emitting \( 2^8 \) symbols, the entropy is \( H(m) = N \).

If the output of a cipher emits symbols with entropy less than \( N \), there is a certain degree of predictability, which threatens its security. Let us consider the ciphertext of Lena's image of size \( 512 \times 512 \) encrypted using the proposed scheme. The number of occurrence of each ciphertext pixel \( m_i \) is recorded and the probability of occurrence is computed for the three image colour components (R, G and B). The entropy for the three image colour components is

\[
\begin{align*}
H_R(m) &= \sum_{i=0}^{2^8-1} p(R_i) \log \left( \frac{1}{p(R_i)} \right) = 7.9993 \approx 8, \\
H_G(m) &= \sum_{i=0}^{2^8-1} p(G_i) \log \left( \frac{1}{p(G_i)} \right) = 8.0008 \approx 8, \\
H_B(m) &= \sum_{i=0}^{2^8-1} p(B_i) \log \left( \frac{1}{p(B_i)} \right) = 8.0007 \approx 8
\end{align*}
\]

(13)

With \( R_i, G_i \) and \( B_i \) are the colour component of the pixel \( m_i \). The values obtained are very close to the theoretical value \( N = 8 \) for the three image colour entropy. This means that information leakage in the encryption process is negligible and the encryption system is secure against the entropy attack.

E. Speed performance

Apart from the security consideration, running speed of the algorithm is also an important aspect for a good encryption algorithm. The simulator for the proposed scheme is implemented using Matlab 7.0. Performance was measured on a 2.0 GHz Pentium Dual-Core with 1 GB RAM running Windows XP. Simulation results show that the average running speed is 22.02 MB/s for encryption and 23.04 MB/s for decryption.

IV. CONCLUSION

In this paper, a color image encryption scheme is proposed, which is based on OCML dynamic system with multiple one-dimensional chaotic maps. The transformation modeled by the coupling structure of the OCML can contribute to the diffusion of the encrypted image. While the use of multiple chaotic maps further increase the confusion of ciphered image. All of the initial conditions and parameters are generated by a 128-bit-long external key as well. Indeed, all performance analysis prove the security robustness of the proposed algorithm.

ACKNOWLEDGEMENT

This work was supported by the National Key Basic Research Project of China (Grant No 2004CB318000), the National Natural Science Foundation of China (Grant No 10771072) and the National Natural Science Foundation of China (Grant No 60703004).

REFERENCES


Jun He
Address: Room 507, Information Building, 500, Dongchuan Road, Shanghai, P.R, China
Date of birth: Nov, 1980
She received the MS degree in Computer Science from East China Normal University, China in 2004. Since September 2007 she has been working towards her PhD degree at the Department of Computer Science and Technology, School of Information Science and Technology, East China Normal University. Her current research interests include image encryption, chaos encryption, information security and cryptology.

Zhi-bin Li
He is professor of Department of Computer Science, School of Information Science and Technology, East China Normal University from 2000 to present. He received the PhD degree in Mathematics, Lanzhou University, 1988. His research interests include Computer symbolic manipulation, Constructive mathematics, Nonlinear mathematical physics.

Hai-feng Qian
He is associate professor of Department of Computer Science, School of Information Science and Technology, East China Normal University from 2007 to present. He received the PhD degree in Computer software and theory, Shanghai Jiaotong University, 2007. His research interest include computer security, information security and cryptography, network security.