# Chaos-based encryption for fractal image coding*

Yuen Ching-Hung(袁正雄)† and Wong Kwok-Wo(黄国和)

*Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China*

(Received 25 July 2011; revised manuscript received 18 August 2011)

A chaos-based cryptosystem for fractal image coding is proposed. The Rényi chaotic map is employed to determine the order of processing the range blocks and to generate the keystream for masking the encoded sequence. Compared with the standard approach of fractal image coding followed by the Advanced Encryption Standard, our scheme offers a higher sensitivity to both plaintext and ciphertext at a comparable operating efficiency. The keystream generated by the Rényi chaotic map passes the randomness tests set by the United States National Institute of Standards and Technology, and so the proposed scheme is sensitive to the key.

**Keywords:** chaos, cryptography, joint image compression and encryption, fractal image coding

**PACS:** 05.45.–a, 05.45.Gg, 05.45.Df **DOI:** 10.1088/1674-1056/21/1/010502

## 1. Introduction

In recent years, the rapid growth in the demand for transmitting multimedia files via public networks has attracted a lot of interest in image compression and encryption. The goals are to minimize the amount of transmission data and to protect the confidentiality of the multimedia content. As chaos demonstrates good properties in ergodicity and randomness, several chaos-based cryptosystems operating in the spatial domain have been suggested.[1−5] In those cryptosystems, the diffusion and permutation based on chaos are directly performed on the image pixels. Therefore, the cipher image cannot be further compressed because of the flattened histogram. Apart from those schemes, Chang *et al.*[6] proposed a lossless image compression and encryption approach using a linear quadtree. Its security relies on the scanning order of the four quadrants. However, the compression capability is not satisfactory due to the lossless reconstruction.

Many encryption schemes operating in the frequency domain have been proposed.[7−10] The discrete cosine transform (DCT) is usually employed to transform an image from the spatial domain to the frequency domain. Lian[7] suggested a selective encryption scheme, which only encrypted the important portion of the multimedia content, such as the values of low frequency coefficients and the signs of high frequency coefficients. The scheme is efficient as only certain parts of the information are encrypted. However, Wu and Kuo[8] analysed the performance of the selective encryption scheme on the DCT. They found that it is possible to recover a recognizable image by assigning all the encrypted direct current (DC) coefficients to 128 and all the alternating current (AC) coefficients to positive values. To provide a full and efficient encryption, a block-based permutation was introduced.[9,10]

In addition to the compression algorithms operating in the frequency and the spatial domains, fractal image coding is one of the most-studied image compression techniques. An image is divided into a number of blocks and each block is encoded as its similarity to another block. Every block is reconstructed by applying iterated contract transformations on another similar block in the same image. The transformation is governed by a set of fractal parameters. A number of approaches have been suggested to obtain a better fractal transformation.[11−13] A hybrid image compression scheme based on the fractal and the DCT was proposed by Curtis *et al.*[14] to achieve a superior performance on the image quality and the compression ratio.

Lian *et al.*[15] investigated the sensitivity and the distribution of the fractal parameters, and proposed a

*Project supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant No. CityU 123009).

†Corresponding author. E-mail: chyuen@student.cityu.edu.hk

© 2012 Chinese Physical Society and IOP Publishing Ltd

selective encryption scheme for the fractal image coding. In the scheme, the depth of the quadtree, the contrast scaling factor $\alpha_i$ and the brightness offset $\Delta g_i$ are selected and packed as $X_i$ for the encryption. Figure 1 shows the chained encryption mode adopted in Lian *et al.*'s scheme, where $V_0$ is an initialization vector and $K_i$ is the secret key. When the depth of the quadtree is also encrypted, decoding with a wrong key may result in a failure because the number of the encoded blocks cannot be determined.

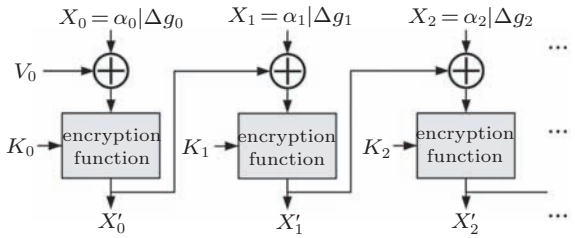



**Fig. 1.** Chained encryption mode adopted in Lian *et al.*'s scheme.[15]

It can be observed that Lian *et al.*'s scheme is efficient only if a stream cipher such as RC4 is employed.[15] The major drawback of the stream cipher is that the key cannot be reused. If a block cipher, such as the Advanced Encryption Standard (AES), is used, each ciphertext pair $X'$ will be expanded to fit the block size and hence the compression ratio will be degraded. Moreover, the length of $X$ is around 15 bits. It is inefficient to encrypt it using a block cipher that usually processes a block of 128 bits.

The above analyses lead to the consideration of whether selective encryption is suitable for fractal image coding. The main advantage of selective encryption is the reduced computational load. However, the computation complexity of the fractal image coding is far higher than that required by any practical encryption algorithm. Moreover, selective encryption usually allows a decoder without the correct key to recover the image at a degraded quality. However, Lian *et al.*'s scheme does not have such a feature.[15] The cipher image can only be decoded after the whole encoded sequence has been received. Therefore, it is not suitable for real-time applications. Hence, the advantages of selective encryption cannot be utilized. There is a need to develop an encryption scheme that possesses the following properties. (i) The encryption part causes no or only a negligible effect on the compression capability and the quality of the recovered image. (ii) The operating efficiency must not be substantially downgraded after incorporating encryption into the fractal image coding. (iii) The key space should be large enough to resist brute force attack. (iv) The cipher image should be sensitive to the key, the plaintext and the ciphertext, so it can resist common attacks, such as the differential attack.

Here, a full chaos-based encryption scheme for fractal image coding is proposed. As the digitized Rényi chaotic map can generate a pseudo-random binary sequence at a low computation complexity,[16] it is employed in our scheme. A 111-bit combined Rényi chaotic map is chosen to partially control the encoding order and to generate a keystream for the masking operation. The randomness of the generated keystream will be verified using the United States National Institute of Standards and Technology (NIST) SP800-22 test suite for pseudo-random number generators (PRNGs).[17] In addition, a feedback mechanism and backward masking will be adopted to guarantee the sensitivities on the key, the plaintext and the ciphertext. The simulation results justify that our scheme offers higher sensitivity to both plaintext and ciphertext at a comparable operating efficiency when compared with the traditional approach of fractal image coding followed by the AES.

The rest of this paper is organized as follows. The background of fractal image coding and the Rényi chaotic map is given in the next section. It is followed by the details of the proposed cryptosystem in Section 3. Then the experimental results are presented in Section 4 and the security analysis of our scheme is given in Section 5. Finally, conclusions are drawn in Section 6.

## 2. Background

### 2.1. Fractal image coding

The fractal image coding was proposed by Jacquin[18−20] in 1989. As natural images usually have high redundancy, they can be exploited through the self-transformation of one block to another. Thus, the theory of iterated contract transformation forms the root of the fractal image coding. The encoded sequence contains a set of fractal parameters, which specifies the transformation details rather than the pixel values. Suppose that there is a matrix space $M$ for digital images and $d$ is a metric for distortion measurement. For a given image $\mu_{\text{org}}$, it is able

to construct a contractive image transformation $\tau$ by solving the inverse problem of the iterated contract transformation.[21] By iterating the contractive image transformation $\tau$, it will eventually bring the image from space $(M,\ d)$ to an approximate fixed point, which is close to $\mu_{\rm org}$. Once $\tau$ satisfies the requirements specified as[20]

$$\exists s < 1 \quad \text{such that}$$
$$\forall \mu, \nu \in M, d(\tau(\mu), \tau(\nu)) \leq s d(\mu, \nu), \qquad (1)$$
$$d(\mu_{\rm org}, \tau(\mu_{\rm org}))$$
$$\text{is as close to zero as possible,} \qquad (2)$$

where $s$ is the contract factor of $\tau$. The storage space of $\tau$ will be smaller than that of $\mu_{\rm org}$ and hence compression is achieved.

As a smaller block size will increase the probability of self-similarity, each image is divided into many $W \times H$ range blocks and $2W \times 2H$ domain blocks, where $W$ and $H$ are the width and the height of the range block, respectively. For every range block, the domain block that is the most matched will be found. As the domain block is contracted during the transformation, its size must be at least twice that of the range block. In fractal image coding, quadtree partitioning is the most common method adopted, in which the image is divided into non-overlapping square blocks. As the blocks are non-overlapped with fixed sizes, the temporary storage space required to keep the domain blocks is optimal. In addition, the encoding time is directly proportional to number of domain blocks. Too many partitions will cause an exhaustive encoding time. Thus, the minimum and the maximum numbers of the quadtree partition are specified.

The contractive image transformation $\tau$ is given by

$$\tau \begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} = \begin{bmatrix} l_{11} & l_{12} & 0 \\ l_{21} & l_{22} & 0 \\ 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ u(x,y) \end{bmatrix} + \begin{bmatrix} D_x \\ D_y \\ \Delta g \end{bmatrix}, \quad (3)$$

which consists of four fractal parameters. The description and the bit length of each parameter are listed in Table 1. The bit length of $D_{xy}$ depends on the number of domain blocks at the same quadtree level. The isometric transformation $l_i$ is a 2×2 matrix, which specifies the flip and the rotation in the block. Every time a domain block is compared with a range block, it will be downsampled to the size of the range block. The downsampled domain block is denoted by $u$ in Eq. (3). By iterating $\tau$ defined in Eq. (3) until a fixed point is reached, the block can be reconstructed.

**Table 1.** Fractal parameters and corresponding bit lengths.

| Fractal parameter | Description | Number of bits |
|---|---|---|
| $D_{xy}$ | position of matched domain block | variable |
| $\alpha_i$ | contrast scaling factor | 6 |
| $\Delta g_i$ | brightness offset | 9 |
| $l_i$ | isometric transformation | 3 |

## 2.2. Rényi chaotic map

Chaotic systems possess good pseudo-randomness and ergodicity. They are highly sensitive to the initial conditions and are involved in various research topics, such as the equalization and demodulation of the chaotic direct sequence spread spectrum and the statistical analysis of the complexity of chaotic systems.[22,23] In cryptography, chaotic systems have also attracted substantial research focus in the past few years.[1−5] Wong and Yuen[24] embedded compression into the chaotic encryption by incorporating entropy coding into the chaos-based cryptosystem. In addition, Mi *et al.*[25] incorporated chaotic encryption into compression by proposing a chaotic encryption scheme for arithmetic coding. Apart from the cryptography, there are plenty of researches on designing PRNGs using chaos.[16,26−28] In particular, Addabbo *et al.*[16] investigated the randomness and the periodicity of the digitized Rényi chaotic map given by

$$f(k) = \left( q 2^{n-i} k + \left\lfloor \frac{k}{2^j} \right\rfloor \right) \bmod 2^n, \qquad (4)$$

where $k$ is the pseudo-random number generated, $i$ and $j$ are positive integers, $q$ is an odd positive integer, and $n$ is the bit length of $k$. The hardware complexity

of this PRNG is low, because it only involves multiplication, addition and shift operations. More importantly, Addabbo *et al.*[16] stated that Eq. (4) must be invertible in order to reach the maximum period. It occurs if and only if $i = j$. The cardinality of the set of nonlinear maps in Eq. (4) to be analysed is reduced to $2^n - 1$. The PRNG in Eq. (4) with the maximum period length was simulated and reported.[16]

In this paper, four Rényi chaotic maps are combined with the logical exclusive OR (XOR) operation to generate a pseudo-random stream using the secret key as the seed. To obtain the maximum period for the combined generator, the period of each map must be a relatively prime. Therefore, we select $n$ to be 25, 26, 29 and 31, respectively. Hence, the length of the seed, which acts as the secret key, is 111 bits. Every time the combined Rényi chaotic map is iterated, the output will be packed as a 32-bit binary sequence $M_i$ for encryption and masking operations.

## 3. Proposed cryptosystem

A block diagram of the proposed scheme is shown in Fig. 2.

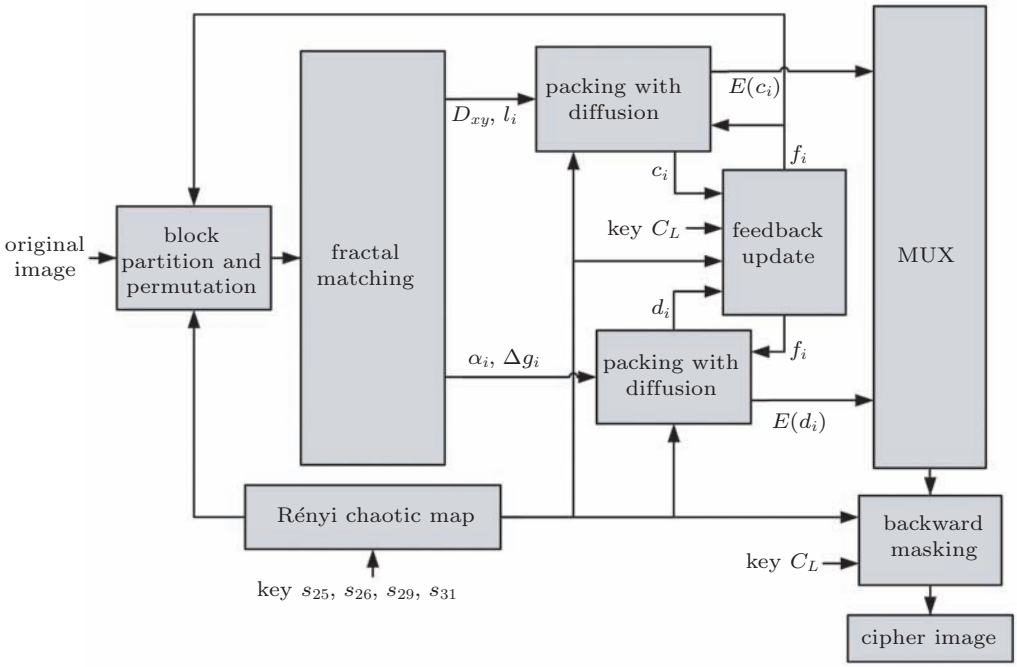



**Fig. 2.** Block diagram of the proposed scheme of fractal image coding with encryption.

The encoding order of the range blocks is determined by the Rényi chaotic map and the feedback $f_i$, which is actually an accumulated sum with cyclic shift from the previous plaintext and keystream. Let $T_i = M_i + f'_i$, the update of $f_i$ is given by

$$f'_i = \begin{cases} \mathrm{CYC}(f_i, \mathrm{left}, \mathrm{LSB}_3(f_i)), & \text{quadtree partition is needed,} \\ \mathrm{CYC}(f_i, \mathrm{right}, \mathrm{LSB}_3(f_i)), & \text{quadtree partition is not needed,} \end{cases} \tag{5}$$

$$\begin{aligned} f_{i+1} &= T_i + c_i + d_i, \\ c_i &= \begin{cases} D_{xy}|l_i, & \text{if } D_{xy}|l_i \text{ is chosen,} \\ l_i|D_{xy}, & \text{if } l_i|D_{xy} \text{ is chosen,} \end{cases} \\ d_i &= \begin{cases} \alpha_i|\Delta g_i, & \text{if } \alpha_i|\Delta g_i \text{ is chosen,} \\ \Delta g_i|\alpha_i, & \text{if } \Delta g_i|\alpha_i \text{ is chosen,} \end{cases} \end{aligned} \tag{6}$$

where $\mathrm{CYC}(a, b, e)$ represents the operation of cyclic shift $a$ towards $b$ by $e$ bits and $\mathrm{LSB}_y(\cdot)$ returns the value of the least $y$ significant bits of the argument. The $f_0$ is initialized by secret key $C_L$. The detailed procedures for encrypting a fractal-coded image are as follows.

i) An image is partitioned into a number of range and domain blocks. Since quadtree partitioning is used in the scheme, there are several levels of partition. In our scheme, the permutation on the sequence of range blocks is done at the lowest depth only. The order of each sub-level will not be affected. Suppose that there are $N$ range blocks at the lowest level, the searching position of the range block will be initialized as $\mathrm{Pack}_y(T_i) \mathrm{mod} N$, where $\mathrm{Pack}_y(T_i)$ packs $T_i$ into $y$ bits by right-shifting and XOR-ing $T_i$ for $\lceil (\text{length of } T_i)/y \rceil$ times, and $y$ is the bit length of $N$. If the block has already been encoded, the next nearest range block that is not encoded yet will be processed.

ii) For each located range block $R_i$, the domain block that is closest to $R_i$ after the transformation will be searched. This is done by solving the inverse of Eq. (3) for every domain block $D_{xy}$ in the same class.[21] The four fractal parameters listed in Table 1 will be computed. By iterating Eq. (3) with these parameters, a block $B_i$, which is close to $R_i$, is reconstructed. The root-mean-square (RMS) difference between $R_i$ and $B_i$ will then be computed. The domain block and the corresponding fractal parameters with the minimum RMS value will be selected. If the minimum RMS is still larger than a predefined tolerance, the range block will be further partitioned by the quadtree unless the maximum quadtree depth is encountered. The partitioned ranged blocks will undergo the fractal transformation again. The $f_i$ will also be cyclic shifted according to Eq. (5).

iii) In practice, it is not necessary to store $D_{xy}$ and $l_i$ in the output file when $\alpha_i$ is equal to zero. This is because the pixel values of the reconstructed block are determined by $\Delta g_i$ in Eq. (3). Therefore, $D_{xy}$ and $l_i$ will be packed into one codeword $c_i$ whereas $\alpha_i$ and $\Delta g_i$ will be packed into another codeword $d_i$ to undergo encryption with diffusion. There are two possible orders each to pack codewords $c_i$ and $d_i$. The order is determined by $\mathrm{LSB}_2(T_i)$, and the encryption with diffusion is governed by

$$E(c_i) = \begin{cases} D_{xy}|l_i + \mathrm{Pack}_p(T_i), & \text{if } \mathrm{LSB}_2(T_i) = 0 \text{ or } 1, \\ l_i|D_{xy} + \mathrm{Pack}_p(T_i), & \text{if } \mathrm{LSB}_2(T_i) = 2 \text{ or } 3, \end{cases}$$
$$E(d_i) = \begin{cases} \alpha_i|\Delta g_i + \mathrm{Pack}_q(T_i), & \text{if } \mathrm{LSB}_2(T_i) = 0 \text{ or } 2, \\ \Delta g_i|\alpha_i + \mathrm{Pack}_q(T_i), & \text{if } \mathrm{LSB}_2(T_i) = 1 \text{ or } 3, \end{cases} \tag{7}$$

where $p$ is the bit length of $D_{xy}|l_i$, and $q$ is the bit length of $\alpha_i|\Delta g_i$.

iv) The $E(c_i)$ and $E(d_i)$ will then be multiplexed into a 32-bit sequence $P$. The feedback $f_i$ will also be updated according to Eq. (6). After all range blocks have been encrypted, a backward masking operation will be performed on sequence $P$ according to

$$C_j = P_j + P_{j+1} + M_{(\sum_{k=j+1}^{L} C_k) \bmod L}, \tag{8}$$

where $C_j$ is the $j$-th 32-bit cipher-image block, and $L$ is the length of $P$ in 32-bit. The value of $P_L$ is the same as the secret key $C_L$. By this operation, any tiny bit change in the plaintext or the key will spread to the front of the cipher image. Finally, the encrypted sequence $C$ will be written into the output file.

A block diagram of the decoding and decryption process is illustrated in Fig. 3. Before the decryption, a set of headers including the minimum ($Q_{\min}$) and the maximum ($Q_{\max}$) numbers of the quadtree partitions, the size of the original image, the number of the encoded range blocks $S$ will be transmitted to the receiver. Then the Rényi map will be iterated $k$ times to generate a keystream $M_i$, where $k = \max(L, S)$.

The unmasking in a backward direction will then be performed on the cipher image to obtain the sequence $P$. Afterwards, the fractal parameters $D_{xy}$, $l_i$, $\alpha_i$ and $\Delta g_i$ will be decrypted. Since the decryption direction is the same as that of the encryption, $f_i$ can be synchronized with that in the encryption directly. The matched domain block will then be transformed into $B_i$ by using Eq. (3). The location of the reconstructed block is recovered by applying the inverse permutation on the encoding sequence using $T_i$. The contractive transformation $\tau$ will be performed several times so that the reconstructed image becomes close to the original one.

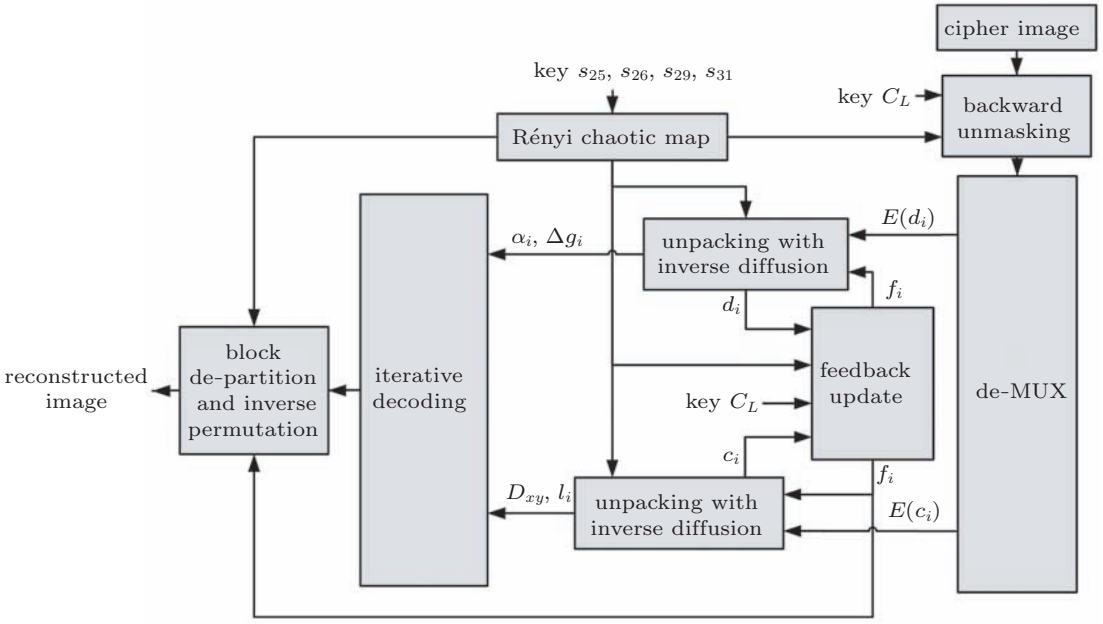



**Fig. 3.** Block diagram of decoding and decryption process.

## 4. Experimental results

The proposed approach of fractal image coding and the encryption is implemented in a C++ program, which is executed on a personal computer equipped with an Intel Core i5 750 CPU at 2.67 GHz and 2 GB memory. Seeds $s_{31}$, $s_{29}$, $s_{26}$ and $s_{25}$ of the four Rényi maps are chosen as the secret key. They are randomly selected to be 142548935, 173468835, 28655065 and 8269143, respectively. Another key is the initialization vector $C_L$ in Eq. (8), which is arbitrarily taken to be 4126907150. The quadtree level is bounded by [5,7]. Eight standard 512×512 images, namely, Aerial, Baboon, Barb, Boat, Frog, Goldhill, Lena and Sailboat, are encrypted and decrypted using the proposed scheme. To evaluate the performance, our scheme is compared with the fractal image coding without any encryption and the fractal image coding encrypted using the AES (128-bit key length running in the cipher feedback (CFB) mode). The comparison results are presented in the following sections.

### 4.1. Compression ratio and image quality

As the proposed scheme does not affect the inverse of $\tau$ given by Eq. (3), both the RMS error and the compression ratio (CR) remain the same. However, the decryption process is smoothened by iterating sequence $M_i$ in advance based on the number of the encoded range blocks S, which has a length of $2Q_{\max}$ bits. There is an extra header apart from the header of the original fractal coded image. Nevertheless, the length of $S$ is shorter compared with that of the zeros padded in the last block of the AES.

For the eight standard images, the corresponding compression ratios and peak signal-to-noise ratios (PSNRs) obtained using the original fractal image coding, the proposed scheme and the fractal image coding encrypted with the AES are listed in Table 2. The differences in compression ratio between the three schemes are within ±0.005, which justifies that our approach induces negligible effect on the compression ratio. In addition, the PSNR of our scheme is almost the same as that of the original fractal image coding. The slight difference in the PSNR arises because the transformation order of the encoded range block is changed. These results show that the encryption causes a negligible effect on both the compression ratio and the quality of the reconstructed image.

**Table 2.** Compression ratios (CRs) and peak signal-to-noise ratios (PSNRs) of eight standard images using the original fractal image coding, the proposed scheme and the fractal image coding encrypted with the AES.

| Test image | Original fractal image coding | | Proposed scheme | | Fractal image coding with AES | |
|---|---|---|---|---|---|---|
| | CR | PSNR/dB | CR | PSNR/dB | CR | PSNR/dB |
| Frog | 4.253 | 27.085 | 4.253 | 27.085 | 4.253 | 27.085 |
| Baboon | 5.112 | 25.216 | 5.111 | 25.217 | 5.112 | 25.216 |
| Aerial | 5.565 | 29.164 | 5.564 | 29.164 | 5.565 | 29.164 |
| Sailboat | 7.670 | 30.863 | 7.670 | 30.863 | 7.669 | 30.863 |
| Barb | 7.835 | 27.307 | 7.835 | 27.307 | 7.834 | 27.307 |
| Goldhill | 8.149 | 31.710 | 8.148 | 31.711 | 8.146 | 31.710 |
| Boat | 8.761 | 32.597 | 8.761 | 32.597 | 8.760 | 32.597 |
| Lena | 13.109 | 33.156 | 13.108 | 33.156 | 13.104 | 33.156 |
| Average | 7.557 | 29.637 | 7.556 | 29.637 | 7.555 | 29.637 |

## 4.2. Encryption/encoding and decryption/decoding speed

Apart from the compression ratio and the image quality, the speeds of the encryption with encoding and the decryption with decoding of the proposed scheme are also compared with those of the original fractal image coding, and the fractal image coding with the AES. To minimize the fluctuation caused by other background applications running on the computer during the test, each image is tested 4000 times. The average values are listed in Table 3.

**Table 3.** Encryption and decryption time of eight standard images obtained using the original fractal image coding, the proposed scheme and the fractal image coding in conjunction with the AES. The time is in the unit of ms.

| Test image | Original fractal image coding | | Proposed scheme | | Fractal image coding with AES | |
|---|---|---|---|---|---|---|
| | encryption | decryption | encryption | decryption | encryption | decryption |
| Lena | 143.745 | 39.637 | 144.171 | 39.879 | 143.856 | 39.783 |
| Barb | 219.245 | 42.166 | 219.716 | 42.446 | 219.267 | 42.607 |
| Goldhill | 231.197 | 42.228 | 231.596 | 42.377 | 231.090 | 42.551 |
| Boat | 234.643 | 41.491 | 234.936 | 41.623 | 234.477 | 41.977 |
| Sailboat | 272.110 | 42.410 | 272.266 | 42.587 | 272.174 | 42.971 |
| Baboon | 303.498 | 45.945 | 303.936 | 46.058 | 303.666 | 46.333 |
| Aerial | 317.743 | 45.205 | 317.943 | 45.336 | 317.449 | 45.593 |
| Frog | 360.738 | 48.087 | 361.146 | 47.973 | 360.879 | 48.954 |
| Average | 260.365 | 43.396 | 260.714 | 43.535 | 260.357 | 43.846 |

The encryption with encoding speed of the proposed scheme varies from 0.692 MB/s to 1.734 MB/s. Its average speed is about 0.002 MB/s lower than those of the original scheme and the fractal image coding with the AES. The encryption with encoding speed of the fractal image coding with the AES is almost the same as that of the original scheme without any encryption. These justify that the time spent on the encryption is negligible when compared with that for the compression. Indeed, it is well known that fractal image coding requires extensive computation for the encoding. It is observed that the time spent on encrypting some images, such as Aerial, using the AES is even shorter than that using the original fractal image coding without any encryption. This is because the time needed for the fractal image coding exceeds that required for the encryption.

In the decryption process, the average decryption with decoding speed of the proposed scheme is 5.761 MB/s, which is 0.020 MB/s lower than that of the original fractal image coding without any encryption. However, it is 0.038 MB/s faster than that of the fractal image coding with the AES on average. Among the eight test images, seven of them were decrypted in

a shorter time. Nevertheless, the differences in speed are not apparent. This is because the time spent on the decompression still dominates the overall running time.

# 5. Security analysis

To analyse the security of the proposed scheme, the key space is evaluated. Moreover, the key, the plaintext and the ciphertext sensitivities are tested. As the keystream generated by the Rényi chaotic map is used to mask the encrypted fractal parameters, the security concerns of the stream cipher is also considered. Thus, the randomness of keystream $M_i$ is examined.

## 5.1. Key space

The secret key is composed of five components, i.e., four seeds $s_{31}$, $s_{29}$, $s_{26}$, $s_{25}$ of the combined Rényi chaotic maps and the 32-bit initialisation vector $C_L$ used in Eq. (8). The bit lengths of $s_{31}$, $s_{29}$, $s_{26}$ and $s_{25}$ are 31, 29, 26, and 25, respectively. Therefore, the total key length is equal to 31+29+26+25+32=143 bits. This value is considered to be secure enough for symmetric encryption schemes.

Figures 4(a) and 4(b) show the plain images of Frog and Lena, respectively. The corresponding reconstructed images using the correct key are shown in Fig. 5. The cipher images decrypted using an incorrect key are depicted in Fig. 6. It can be observed that the images reconstructed with an incorrect key are totally unintelligent. The corresponding PSNRs drop to less than 9 dB.




**Fig. 4.** Plain images of (a) Frog and (b) Lena.

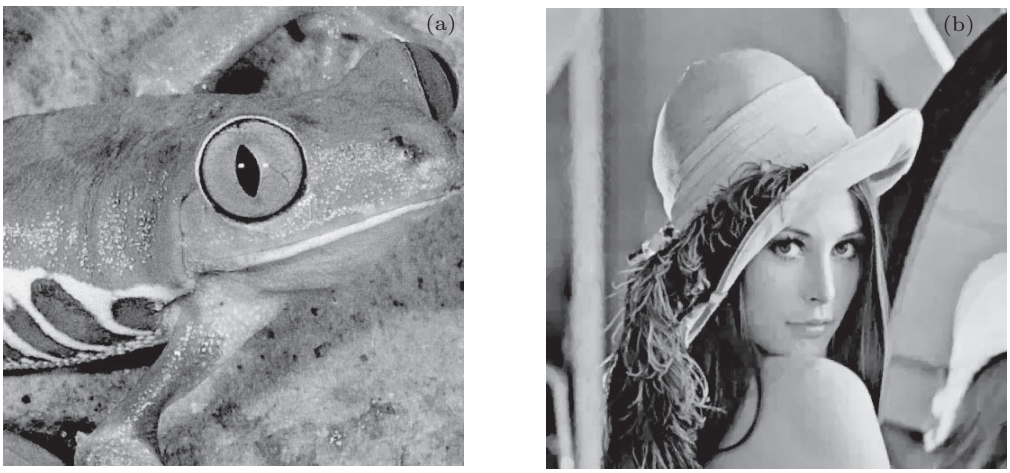
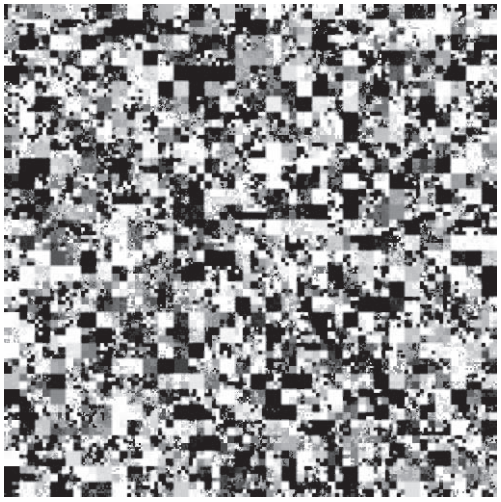
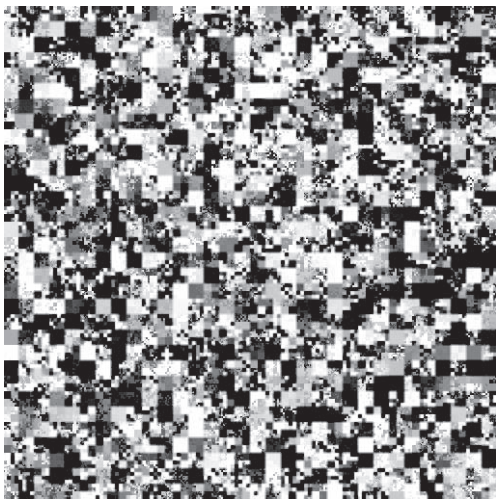



**Fig. 5.** Decrypted images with correct key: (a) Frog, PSNR=27.085 dB; (b) Lena, PSNR=33.156 dB.



(a)



(b)

**Fig. 6.** Decrypted images with incorrect key: (a) Frog, PSNR=8.803 dB; (b) Lena, PSNR=8.296 dB.

## 5.2. Key sensitivity

For a good cryptosystem, half of the bits in the ciphertext should be changed when the key is changed. Therefore, the key sensitivity is evaluated by encrypting the plain image with a chosen key. Then the secret key is changed slightly, and the same plain image is encrypted again. The resulting cipher image is compared with the original cipher image bitwise, and the number of different bits is counted.

In our evaluation, the LSB or the MSB of one of the secret parameters $s_{31}$, $s_{29}$, $s_{26}$, $s_{25}$, and $C_L$ is toggled at each time, while other parameters remain unchanged. The results show that the percentages of unequal bits in the cipher images range from 49.75% to 50.11%, which are close to 50%. They justify that the cipher image is very sensitive to the key.

## 5.3. Plaintext sensitivity

To resist differential attacks, the cipher image should be sensitive to the plain image in such a way that any tiny change in the plain image produces a cipher image with half of the bits changed. This can be verified by evaluating the plaintext sensitivity. First of all, the plain image is encrypted as usual. Then a bit change is introduced into the plain image, which is encrypted again using the same secret key. Every bit in the resulting cipher image is compared with the corresponding bit in the original cipher image, and the number of unequal bits is counted. Since the length of the cipher image may be affected by the bit change, the bit comparison stops at the end of the shorter sequence.

In our test, a single bit is changed from LSB to MSB at the first, the middle and the last plain-image pixel, respectively. The length of the bit comparison and the percentages of the different bits are measured. This test is performed on both the proposed scheme and the fractal image coding encrypted with the AES. The results are listed in Table 4. Among the 24 cipher images evaluated, the lengths of the cipher images are extended by 0.24% at most in both schemes, which justifies that the compression ratio is insensitive to a tiny bit change in the plain image. In addition, the 0-bit difference is recorded when bit 0 or bit 1 of the middle pixel is changed. This means that the resulting cipher image is exactly the same as the original one. This is because the bit change does not affect the similarities between the range and the domain blocks. Thus, the same set of fractal parameters is extracted.

Apart from the two cases mentioned above, the bit differences of other cipher images using the proposed scheme fall within 50±0.26%, which indicates that the cipher image is very sensitive to the plain image due to the feedback mechanism and the backward masking. The feedback mechanism spreads the encrypted fractal parameters in the forward direction. It also affects the encoding order behind. Although the permutation on the encoding sequence is limited by the number of the range blocks, its effect is magnified by both the feedback mechanism and the backward masking, which effectively spread any tiny change to the whole cipher image.

**Table 4.** Plaintext sensitivities and differences in cipher-image length of the proposed scheme and the fractal image encrypted with the AES.

| Location of bit change | Proposed scheme | | | | | | Fractal image coding encrypted with AES | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | plaintext sensitivity/% | | | difference in cipher-image length/% | | | plaintext sensitivity/% | | | difference in cipher-image length/% | | |
| | first pixel | middle pixel | last pixel | first pixel | middle pixel | last pixel | first pixel | middle pixel | last pixel | first pixel | middle pixel | last pixel |
| Bit 0 | 49.77 | 0.00 | 49.86 | 0.00 | 0.00 | 0.00 | 50.09 | 0.00 | 1.15 | 0.00 | 0.00 | 0.00 |
| Bit 1 | 49.77 | 0.00 | 49.93 | 0.00 | 0.00 | 0.00 | 50.09 | 0.00 | 1.15 | 0.00 | 0.00 | 0.00 |
| Bit 2 | 49.88 | 49.98 | 50.26 | 0.00 | 0.00 | 0.00 | 3.11 | 10.20 | 1.18 | 0.00 | 0.00 | 0.00 |
| Bit 3 | 49.80 | 49.98 | 50.26 | 0.00 | 0.00 | 0.00 | 20.46 | 10.20 | 1.18 | 0.00 | 0.00 | 0.00 |
| Bit 4 | 49.94 | 49.98 | 49.83 | 0.00 | 0.00 | 0.04 | 40.46 | 10.20 | 23.22 | 0.00 | 0.00 | 0.00 |
| Bit 5 | 49.90 | 50.12 | 49.98 | 0.00 | 0.00 | 0.00 | 49.92 | 49.56 | 32.28 | 0.00 | 0.00 | 0.00 |
| Bit 6 | 49.90 | 49.97 | 50.11 | 0.00 | 0.00 | 0.04 | 50.01 | 49.75 | 32.21 | 0.00 | 0.00 | 0.00 |
| Bit 7 | 50.03 | 50.10 | 50.14 | 0.22 | 0.10 | 0.04 | 49.86 | 49.64 | 46.48 | 0.24 | 0.08 | 0.00 |

However, more than half of the cases in the fractal image coding encrypted with the AES fail to produce half of the bit changes in the cipher images. In the worst case, there is only 1.15% bit change in the cipher image. This is because the standard AES in the CFB mode only performs the encryption once in the forward direction. Thus, the change occurring at the last cipher block can not diffuse backward to the previous blocks.

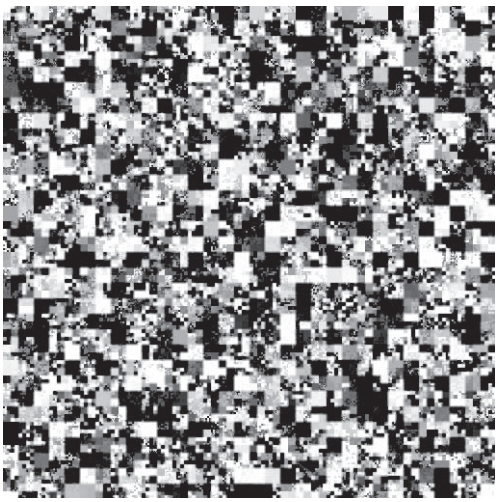
## 5.4. Ciphertext sensitivity

One of the malicious attacks is to degrade the quality of the decrypted image by replacing the values of certain ciphertext blocks. The decrypted image may be visually the same, but its image quality is degraded. This attack defects the efficiency of the scheme because a higher compression ratio could be obtained when the image quality is lowered. In practice, a change in the ciphertext block will result in an incorrect decoding of the quadtree depth and thus a wrong estimation of the ciphertext length. If the decoder reads the required data within the end of the ciphertext, the malicious attack may not be detected. Otherwise, errors will occur at decoding.

To ensure that the attack is also visually detectable, any change in the ciphertext should cause a dramatic degradation in the image quality. Here, we toggle the LSB or the MSB at the first, the middle or the last ciphertext block, respectively. The PSNRs of the recovered images are computed and listed in Table 5. All the PSNRs of the decrypted images using the proposed scheme are lower than 9 dB. When the bit is toggled at the last AES ciphertext block, the corresponding PSNR only drops by a quantity smaller than 0.5 dB. It implies that the importance of each bit in the ciphertext is unequal in the fractal image encrypted with the AES. Figures 7(a) and 7(b) show the decrypted Lena images when the MSB at the middle block is toggled using the proposed scheme and the fractal image coding encrypted with the AES, respectively.

**Table 5.** PSNRs of the decrypted images with ciphertext bit changed using the proposed scheme and the fractal image coding encrypted with the AES in the CFB mode (given within the parentheses).

| Test image | PSNR of the decrypted images/dB | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LSB of the first block | | MSB of the first block | | LSB of the middle block | | MSB of the middle block | | LSB of the last block | | MSB of the last block | |
| Aerial | 7.536 | (11.083) | 7.519 | (28.879) | 7.519 | (10.465) | 7.475 | (10.324) | 7.510 | (29.150) | 7.444 | (29.152) |
| Baboon | 8.627 | (10.160) | 8.610 | (10.162) | 8.615 | (12.419) | 8.462 | (12.319) | 8.555 | (25.214) | 8.539 | (25.216) |
| Barb | 8.303 | (27.082) | 8.353 | (10.092) | 8.289 | (13.944) | 8.181 | (13.948) | 8.344 | (27.017) | 8.135 | (27.307) |
| Boat | 8.208 | (9.333) | 8.261 | (8.553) | 8.004 | (12.746) | 8.162 | (12.747) | 8.298 | (32.595) | 8.020 | (32.597) |
| Frog | 8.803 | (26.939) | 8.780 | (27.015) | 8.732 | (27.029) | 8.757 | (11.465) | 8.615 | (27.084) | 8.764 | (27.085) |
| Goldhill | 8.037 | (9.931) | 8.078 | (9.918) | 8.220 | (31.566) | 8.143 | (31.574) | 8.077 | (31.710) | 8.149 | (31.710) |
| Lena | 8.296 | (10.758) | 8.268 | (9.769) | 8.367 | (12.117) | 8.273 | (32.481) | 8.326 | (33.139) | 8.160 | (33.156) |
| Sailboat | 7.389 | (30.360) | 7.518 | (9.065) | 7.464 | (28.570) | 7.557 | (12.019) | 7.515 | (30.863) | 7.592 | (30.863) |
| Average | 8.150 | (16.956) | 8.173 | (14.182) | 8.151 | (18.607) | 8.126 | (17.110) | 8.155 | (29.597) | 8.100 | (29.636) |



(a)



(b)

**Fig. 7.** Decrypted Lena with the MSB at the middle ciphertext block toggled using (a) proposed scheme (PSNR=8.273 dB) and (b) fractal image coding encrypted with AES (PSNR=32.481 dB).

### 5.5. Randomness of the keystream

One of the security concerns is the randomness of the binary mask sequence $M_i$ generated by using the combined Rényi chaotic map. Thus, 300 sequences, each with a length of 1,000,000 bits, are extracted from $M_i$. The randomness of these sequences is evaluated by using the statistical test suite recommended by the NIST.[17] The test suite is composed of 15 tests. Each of them is employed to analyse a statistical property of the sequence.

For each test, a $P$ value is calculated to indicate the probability for a perfect random number generator to produce a sequence that is less random than the test sequence. When the $P$ value is greater than a pre-defined threshold $\beta$, which is commonly taken as 0.01 in the cryptographic research, the test sequence is considered to be random. After evaluating 300 sequences in each test, the $P$ values are collected to compute a $U$ value, which indicates the uniformity of the $P$ values. If the $U$ value is greater than 0.0001, the test sequences can be considered to have a uniform distribution.

In addition, there is a minimum pass rate for each statistical test, which indicates the minimum number of test sequences required to pass the test. In our evaluation, the minimum pass rate is 0.97. In Table 6 we list the tests and the corresponding results for 300 sequences. All $U$ values are greater than 0.0001, and the pass rates are all greater than 0.97. These results justify that keystream $M_i$ generated by the combined Rényi chaotic map possesses good randomness.

**Table 6.** Results of the statistical test suite recommended by NIST for 300 sequences extracted from the combined Rényi chaotic map.

| Statistical test | Pass rate | $U$ value |
|---|---|---|
| Frequency (monobit) test | 0.9967 | 0.7856 |
| Frequency test within a block | 0.9900 | 0.7598 |
| Cumulative sums (cusum) test [a)] | 0.9967 | 0.2803 |
| Runs test | 0.9833 | 0.1025 |
| Test for the longest run of ones in a block | 0.9967 | 0.5208 |
| Binary matrix rank test | 0.9967 | 0.9429 |
| Discrete fourier transform (spectral) test | 0.9867 | 0.1005 |
| Non-overlapping template matching test[a)] | 0.9733 | 0.0028 |
| Overlapping template matching test | 0.9800 | 0.9281 |
| Maurer's universal statistical test | 0.9967 | 0.6163 |
| Approximate entropy test | 0.9833 | 0.7981 |
| Random excursions test[a)] | 0.9784 | 0.0110 |
| Random excursions variant test[a)] | 0.9730 | 0.0613 |
| Serial test[a)] | 0.9933 | 0.6579 |
| Linear complexity test | 0.9800 | 0.5681 |

[a)]The statistical test consists of several sub tests. The worst case is reported.

## 6. Conclusion

A chaos-based full encryption scheme for the fractal image coding is proposed. Four Rényi chaotic maps are combined to generate a keystream, which is used to change the encoding order of the range blocks and to mask the encrypted parameters. The proposed encryption scheme has only negligible effects on both the compression capability and the reconstructed image quality. The experimental results show that the feedback mechanism and the backward masking operation effectively enhance the sensitivities on the key,

the plaintext and the ciphertext. Although the encryption speed of the proposed scheme is slightly lower than that of the fractal image coding followed by the AES, its decryption speed is faster in general. Moreover, our scheme is shown to have higher sensitivities to both the plaintext and the ciphertext.

# References


[1] Wang Y, Wong K W, Liao X F and Chen G R 2011 *Appl. Soft. Comput.* **11** 514

[2] Chen G R, Mao Y B and Chui C K 2004 *Chaos Soliton. Fract.* **21** 749

[3] Wong K W, Kwok B S H and Law W S 2008 *Phys. Lett.* A **372** 2645

[4] Wang Y, Wong K W, Liao X F, Xiang T and Chen G R 2009 *Chaos Soliton. Fract.* **41** 1773

[5] Sun F Y and Lu Z W 2011 *Chin. Phys.* B **20** 040506

[6] Chang H K C and Liu J L 1997 *Signal Processing: Image Communication* **10** 279

[7] Lian S G 2009 *Chaos Soliton. Fract.* **40** 2509

[8] Wu C P and Kuo C C J 2005 *IEEE T. Multimedia* **7** 828

[9] Krikor L, Baba S, Arif T and Shaaban Z 2009 *European Journal of Scientific Research* **32** 47

[10] Ge X, Liu F L, Lu B, Wang W and Chen J 2010 *Proceedings of the 2nd IEEE International Conference on Information Management and Engineering*, April 16–18, 2010 Chengdu China p. 267

[11] Bouboulis P and Dalla L 2005 *Fractals* **13** 227

[12] Drakopoulos V, Bouboulis P and Theodoridis S 2006 *Fractals* **14** 259

[13] He C, Xu X and Yang J 2006 *Chaos Soliton. Fract.* **27** 1178

[14] Curtis K M, Neil G and Fotopoulos V 2002 *Proceedings of the 14th International Conference on Digital Signal Processing* **2** 1337

[15] Lian S G, Chen X and Ye D P 2009 *Fractals* **17** 149

[16] Addabbo T, Alioto M, Fort A, Pasini A and Rocchi S 2007 *IEEE T. Circuits*-I **54** 816

[17] Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert A, Dray J and Vo S 2010 *A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications*

[18] Jacquin A E 1989 *PhD Thesis* Georgia Institute of Technology 2225

[19] Jacquin A E 1992 *IEEE T. Image Process.* **1** 18

[20] Jacquin A E 1993 *Proc. IEEE* **81** 1451

[21] Fisher Y 1995 *Fractal Image Compression Theory and Application* (New York: Springer-Verlag)

[22] Sun K H, He S B and Sheng L Y 2011 *Acta Phys. Sin.* **60** 020505 (in Chinese)

[23] Xu X Z and Guo J B 2011 *Acta Phys. Sin.* **60** 020510 (in Chinese)

[24] Wong K W and Yuen C H 2008 *IEEE T. Circuits*-II **55** 1193

[25] Mi B, Liao X F and Chen Y 2008 *Chaos Soliton. Fract.* **38** 1523

[26] Wang F L 2010 *Chin. Phys.* B **19** 090505

[27] Liu S B, Sun J, Xu Z Q and Liu J S 2009 *Chin. Phys.* B **18** 5219

[28] Addabbo T, Alioto M, Fort A, Rocchi S and Vignoli V 2006 *IEEE T. Circuits*-II **53** 329