NEW IMAGE ENCRYPTION/DECRYPTION ALGORITHMS BASED ON CHAOS

Hung-Chun Lin, Shuenn-Shyang Wang and Li-Ling Chen

Department of Electrical Engineering
Tatung University
40 Chungshan N.Road, 3rd Sec.
Taipei, Taiwan, R.O.C.
Email: sswang@ttu.edu.tw

Abstract- *In recent years, owing to the great development of communication system and multimedia technology, the digital images can be altered and destroyed by the illegal user easily. Hence, image data security has become a critical issue. This paper is intended to propose some image encryption/decryption algorithms by taking advantage of the features of the chaotic sequences. The computational complexity and security level of proposed image encryption/decryption algorithms are analyzed. Finally, the simulation results and their fractal dimensions are given to demonstrate the effectiveness of the proposed schemes.*

Keywords: Image Encryption/Decryption, Chaotic sequence

1. Introduction

In the digital world nowadays, illegal data accessing has become easier and more prevalent in communication networks because of the great development in communication equipment. Hence, multimedia data security has become a critical issue. To solve the security problem, many encryption techniques have been proposed to protect valuable image data from undesirable reader. The encryption/decryption algorithms of digital images can be classified into three major types: position permutation [1-5], value transformation [6-9] and the combining form [10]. Moreover, chaotic sequences have several good properties including the ease of their generation, their sensitive dependence on initial condition and noise like [11].Thus, applying the chaos to cryptography was a promising way to improve the security of data since the excellent properties of chaotic sequences.

In this paper, we proposed some image encryption/decryption algorithms that take advantages of the chaotic sequences to scramble either the positions of pixels of the images or the positions of the bitplane of the images individually to form the encrypted images. The algorithms we proposed have some beneficial features including low complexity and high security. The rest of this paper is organized as follows: in section 2, we will review the existed image encryption methods. In section 3, the new digital image encryption/decryption algorithms based on chaos will be presented. Finally the experimental results and performance comparison of these algorithms are presented in section 4 and conclude it in section 5.

2. Image Encryption/Decryption Algorithms Based On Chaos

In this section, some new image encryption /decryption algorithms based on chaos are proposed. One algorithm is to encrypt the images by scrambling the pixel positions, and the other one is to encrypt the images by scrambling each bitplane of the images individually. There are three schemes (*scheme A1*, *scheme A2* and s*cheme A3*) belonging to *Category A* that encrypts the images by scrambling the pixel positions as described in the following:

Scheme A1:

Step 1: Consider an image f of size $M \times N$ pixels, and let $f(x, y)$, $0 \le x \le M-1$, $0 \le y \le N-1$, be the gray level of this image f at position (x,y). Determine two different 1-D chaotic systems with the initial values $x(0)$ and $y(0)$.

Step 2: Generate two chaotic sequences $x(0),x(1),x(2),\ldots$and $y(0),y(1),y(2),\ldots$from two different chaotic systems. Then create $b_x(0)$, $b_x(1)$, $b_x(2),\ldots$, $b_x(MN\text{-}1)$ from $x(0)$, $x(1)$, $x(2),\ldots,x(MN/8\text{-}1)$ and $b_y(0)$, $b_y(1)$, $b_y(2),\ldots,b_y(MN\text{-}1)$from $y(0)$, $y(1)$, $y(2),\ldots$, $y(MN/8\text{-}1)$ by the scheme that $0.b(8n+0)b(8n+1)b(8n+2)b(8n+3)b(8n+4)b(8n+5)$ $b(8n+6)b(8n+7)$ is the binary representation of $x(n)$ or $y(n)$ for n= 0, 1, 2, …,$MN/8$-1.

Step 3: Apply ***Pixel Swapping Function*** defined below to the image $f(x,y)$:

Pixel Swapping Function ($f(i,j)$; $b_x(k)$, $b_y(k)$; k, $0 \le i \le M-1$, $0 \le j \le N-1$):
Initial setting: k=0
Step(i):
for(j=0;j<$N/2$;j++)
for (i=0;i<$M/2$;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i+M/2,j)$)

Swapping $_{bx(k)=0,\ by(k)=1}$ ($f(i,j)$, $f(i,j+N/2)$
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f(i,j)$, $f(i+M/2,j+N/2)$)
$k=k+1$;
End
End
Step(ii):
for($j=0;j<N/2;j++$)
for ($i=M/4;i<M/2;i++$) and ($i=3M/4;i<M;i++$)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i-M/4,j)$)
Swapping $_{bx(k)=0 \text{ and } by(k)=1}$ ($f(i,j)$, $f(i,j+N/2)$)
Swapping $_{bx(k)=1 \text{ and } by(k)=1}$ ($f(i,j)$, $f(i-M/4,j+N/2)$)
$k=k+1$;
End
End
Step(iii):
for($j=N/4;j<N/2;j++$) and ($j=3N/4;j<N;j++$)
for ($i=M/2;i<M;i++$)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i-M/2,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f(i,j)$, $f(i,j-N/4)$)
Swapping $_{bx(k)=1 \text{ and } by(k)=1}$ ($f(i,j)$, $f(i-M/2,j-N/4)$)
$k=k+1$;
End
End
Step(iv):
for($j=N/2;j<N;j++$)
for ($i=0;i<M/2;i++$)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i+M/2,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f(i,j)$, $f(i,j-N/2)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f(i,j)$, $f(i+M/2,j-N/2)$)
k=k+1;
End
End

Step 4: Stop the algorithm.

From Fig.2, we can realize the swapping action of the ***Pixel Swapping Function*** of this scheme. In the decryption procedure, we should reverse ***Pixel Swapping Function*** with the same chaotic systems and their initial states. That is, the step order is *Step (iv), Step (iii),Step (ii), and Step (i).* To analyze its computational complexity, the numbers of different kinds of operations required to perform the ***Pixel Swapping Function*** are listed in Table 1. In the analysis, we made the assumption that $Prob(b(k)=1)= Prob(b(k)=0)=1/2$. It is seen that compared this scheme with CMLIE algorithm this scheme has a low computational complexity.

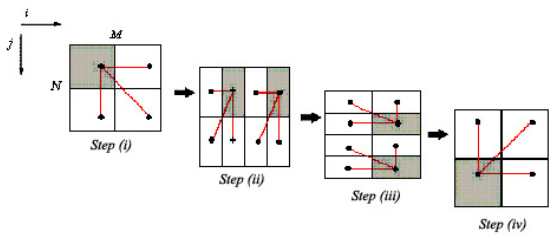


Fig.2 *Pixel Swapping Function* of *scheme A1*.

Table 1.Numbers of different kinds of operations for *Scheme A1*

	Increase i,j,k	if	+	-	Memory swap
Step (i)	3MN/4	3MN/4	MN/4	0	3MN/16
Step (ii)	3MN/4	3MN/4	MN/8	MN/8	3MN/16
Step (iii)	3MN/4	3MN/4	0	MN/4	3MN/16
Step (iv)	3MN/4	3MN/4	MN/8	MN/8	3MN/16
Total	3MN	3MN	MN/2	MN/2	3MN/4

Scheme A2:

Step 1: Consider an image f of size $M \times N$ pixels, and let $f(x, y)$, $0 \le x \le M-1$, $0 \le y \le N-1$, be the gray level of this image f at position (x,y). Then determine three different 1-D chaotic systems with the initial value $x(0)$, $y(0)$, and $z(0)$.

Step 2: Generate three chaotic sequences $x(0),x(1),x(2),\ldots;y(0),y(1),y(2),\ldots$and $z(0),z(1),z(2),\ldots$ from three different chaotic systems. Then create $b_x(0)$, $b_x(1)$, $b_x(2),\ldots$, $b_x(MN\text{-}1)$ from $x(0)$, $x(1)$, $x(2),\ldots,x(MN/8\text{-}1)$ and $b_y(0)$, $b_y(1)$, $b_y(2),\ldots,b_y(MN\text{-}1)$ from $y(0)$, $y(1)$, $y(2),\ldots$, $y(MN/8\text{-}1)$ by the scheme that $0.b(8n+0)b(8n+1)b(8n+2)b(8n+3)b(8n+4)b(8n+5)$ $b(8n+6)b(8n+7)$ is the binary representation of $x(n)$ or $y(n)$ for n= 0, 1, 2, ….

Step 3: Apply ***Pixel Swapping Function*** followed with ***Word XOR Function*** defined below to the image $f(x,y)$:

Word XOR Function ($f(i, j)$; $z(t)$; t, $0 \le i \le M-1$, $0 \le j \le N-1$***):***
Initial setting: $t=0$
for($j=0;j<N;j++$)
for ($i=0;i<M;i++$)
$f(i, j) = f(i, j) \oplus z(t)$;
$t=t+1$;
End
End

Step 4: Stop the algorithm

The operation " $\oplus$ "in ***Word XOR Function*** represents the "XOR" operation by the word. In view of Fig.3, we can realize the encryption action of *Scheme A2*.The decryption procedure of this scheme involves the ***Word XOR Function*** followed by the reverse of the ***Pixel Swapping Function***. The computational complexity of this scheme is higher than the *scheme A1*, since it includes the ***Word XOR Function***. In other words, we increase a little computational complexity in order to make the encrypted images look more disorderly.

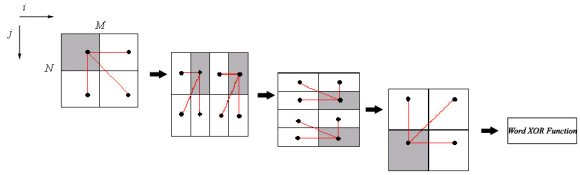


Fig.3 The encryption action of *Scheme A2*.

***Scheme A3*:**

Step 1: Consider an image f of size $M \times N$ pixels, and let $f(x, y)$, $0 \le x \le M-1$, $0 \le y \le N-1$, be the gray level of this image f at position (x,y). Then determine three different 1-D chaotic systems with the initial value $x(0)$ $y(0)$, and $z(0)$, and set $k=0$, $t=0$.

Step 2: Generate three chaotic sequences $x(0),x(1),x(2),\ldots;y(0),y(1),y(2),\ldots$and $z(0),z(1),z(2),\ldots$ from three different chaotic systems. Then create $b_x(0)$, $b_x(1)$, $b_x(2),\ldots$, $b_x(MN\text{-}1)$ from $x(0)$, $x(1)$, $x(2),\ldots$, $x(MN/8\text{-}1)$ and $b_y(0)$, $b_y(1)$, $b_y(2),\ldots,b_y(MN\text{-}1)$ from $y(0)$, $y(1)$, $y(2),\ldots$, $y(MN/8\text{-}1)$ by the scheme that $0.b(8n+0)b(8n+1)b(8n+2)b(8n+3)b(8n+4)b(8n+5)$ $b(8n+6)b(8n+7)$ is the binary representation of $x(n)$ or $y(n)$ for $n=0, 1, 2, \ldots$.

Step 3:

for(j=0;j<N/2;j++)
 for (i=0;i<M/2;i++)
  Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i+M/2,j)$)
  Swapping $_{bx(k)=0,\ by(k)=1}$ ($f(i,j)$, $f(i,j+N/2)$)
  Swapping $_{bx(k)=1,\ by(k)=1}$ ($f(i,j)$, $f(i+M/2,j+N/2)$)
  k=k+1;
 End
End
for(j=0;j<N;j++)
 for (i=0;i<M;i++)
  $f(i, j) = f(i, j) \oplus z(t)$;
  t=t+1;
 End
End

Step 4:

for(j=0;j<N/2;j++)
 for (i=M/4;i<M/2;i++) and (i=3M/4;i<M;i++)
  Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i\text{-}M/4,j)$)
  Swapping $_{bx(k)=0 \text{ and } by(k)=1}$ ($f(i,j)$, $f(i,j+N/2)$)
  Swapping $_{bx(k)=1 \text{ and } by(k)=1}$ ($f(i,j)$, $f(i\text{-}M/4,j+N/2)$)
  k=k+1;
 End
End
for(j=0;j<N;j++)
 for (i=0;i<M;i++)
  $f(i, j) = f(i, j) \oplus z(t)$;
  t=t+1;
 End
End

Step 5:

for(j=N/4;j<N/2;j++) and (j=3N/4;j<N;j++)
 for (i=M/2;i<M;i++)
  Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i\text{-}M/2,j)$)
  Swapping $_{bx(k)=0,\ by(k)=1}$ ($f(i,j)$, $f(i,j\text{-}N/4)$)
  Swapping $_{bx(k)=1,\ by(k)=1}$ ($f(i,j)$, $f(i\text{-}M/2,j\text{-}N/4)$)
  k=k+1;
 End
End
for(j=0;j<N;j++)
 for (i=0;i<M;i++)
  $f(i, j) = f(i, j) \oplus z(t)$;
  t=t+1;
 End
End

Step 6:

for(j=N/2;j<N;j++)
 for (i=0;i<M/2;i++)
  Swapping $_{bx(k)=1,\ by(k)=0}$ ($f(i,j)$, $f(i+M/2,j)$)
  Swapping $_{bx(k)=0,\ by(k)=1}$ ($f(i,j)$, $f(i,j\text{-}N/2)$)
  Swapping $_{bx(k)=1,\ by(k)=1}$ ($f(i,j)$, $f(i+M/2,j\text{-}N/2)$)
  k=k+1;
 End
End
for(j=0;j<N;j++)
 for (i=0;i<M;i++)
  $f(i, j) = f(i, j) \oplus z(t)$;
  t=t+1;
 End
End

Step 7: Stop the algorithm.

In *scheme A3*, the *Word XOR Function* was be added to the *Step 3-6*. The pixel permutations are followed by the *Word XOR Function* for transforming the pixel values in each step. We can observe the encryption action of this scheme as shown in Fig.4. The decryption procedure of this scheme reverses the *Step 3-6* stated above. That is, the step order is *Step 6*, *Step 5*, *Step 4*, and *Step 3*.

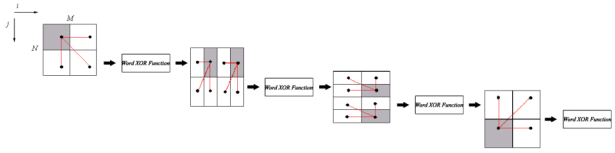


Fig.4 The encryption action of *Scheme A3*

Moreover, we propose three schemes(*scheme B1*, *scheme B2* and *scheme B3*) belonging to *Category B* that encrypt the images by scrambling each bitplane individually, which are described as follows:

***Scheme B1*:**

Step 1: Consider an image f of size $M \times N$ pixels, and let $f(x, y)$, $0 \le x \le M-1$, $0 \le y \le N-1$, be the gray level of this image f at position (x,y). Divide the image f into 8 bit-plane $f_b(x, y)$, b=0, 1, 2,…, 7, according to its gray level value $f(x, y)$, where $f(x, y)$ = {0, 1, 2,…, 255}. Then determine two different 1-D chaotic systems with the initial value $x(0)$ and $y(0)$.

Step 2: Generate two chaotic sequences $x(0),x(1),x(2),\ldots,x(MN\text{-}1)$and $y(0),y(1),y(2),\ldots, y(MN\text{-}1)$ from two different

chaotic systems. Then create $b_x(0)$, $b_x(1)$, $b_x(2)$,…, $b_x(8MN\text{-}1)$ from $x(0)$, $x(1)$, $x(2)$,…,$x(MN\text{-}1)$ and $b_y(0)$, $b_y(1)$, $b_y(2)$,…, $b_y(8MN\text{-}1)$ from $y(0)$,$y(1)$, $y(2)$,…,$y(MN\text{-}1)$ by the scheme that $0.b(8n+0)b(8n+1)b(8n+2)b(8n+3)b(8n+4)b(8n+5)$ $b(8n+6)b(8n+7)$ is the binary representation of $x(n)$ or $y(n)$ for n= 0, 1, 2, …, MN-1.

Step 3: Apply ***Bit Swapping Function*** individually to 8 bit-plane $f_b(x, y)$, b=0, 1, 2,…,7. The ***Bit Swapping Function*** is defined as follows:

Bit Swapping Function($f_b(i,j)$; $b_x(k)$, $b_y(k)$; k, b, $0 \le i \le M-1, 0 \le j \le N-1$***):***

Initial setting: k=MNb;
Step (i):
for(j=0;j<N/2;j++)
for (i=0;i<M/2;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f_b(i,j)$, $f_b(i+M/2,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i,j+N/2)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i+M/2,j+N/2)$)
k=k+1;
End
End
Step (ii):
for(j=0;j<N/2;j++)
for (i=M/4;i<M/2;i++) and ((i=3M/4;i<M;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f_b(i,j)$, $f_b(i-M/4,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i,j+N/2)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i-M/4,j+N/2)$)
k=k+1;
End
End
Step (iii):
for(j=N/4;j<N/2;j++) and (j=3N/4;j<N;j++)
for (i=M/2;i<M;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f_b(i,j)$, $f_b(i-M/2,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i,j-N/4)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i-M/2,j-N/4)$)
k=k+1;
End
End
Step (iv):
for(j=N/2;j<N;j++)
for (i=0;i<M/2;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f_b(i,j)$, $f_b(i+M/2,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i,j-N/2)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i+M/2,j-N/2)$)
k=k+1;
End
End

Step 4: Stop the algorithm.

The encryption action of this scheme is shown in Fig.5. In the decryption procedure, we should perform the reverse of the ***Bit Swapping Function***. The computational complexity of this category is higher than *Category A* since it requires eight times permutation for an image.

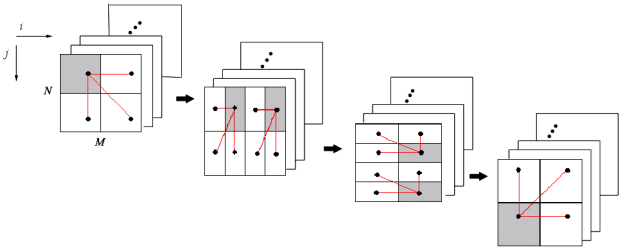


Fig.5 The encryption action of *scheme B1*.

***Scheme B2*:**

In the *Scheme B2*, it is different form *Scheme B1* that the ***Bit Swapping Function*** is followed by ***Bit XOR Function*** that is defined as follows:

Bit XOR Function($f_b(i,j)$; $b_z(t)$;t, $0 \le i \le M-1, 0 \le j \le N-1$***):***

Initial setting: t=MNb;
for(j=0;j<N;j++)
for (i=0;i<M;i++)
$f_b(i,j) = f_b(i,j) \oplus b_z(t)$;
t=t+1;
End
End

***Scheme B3*:**

In the *Scheme B3*, it is different form *Scheme B1* that we apply 8 bit-plane $f_b(x, y)$,b=0, 1, 2,…, 7, to ***Bit Swapping-XOR Function*** individually. The ***Bit Swapping-XOR Function*** is defined as follows:

Bit Swapping-XOR Function($f_b(i,j)$; $b_x(k)$, $b_y(k)$, $b_z(t)$; k, b,t, $0 \le i \le M-1, 0 \le j \le N-1$***):***

Initial setting: k=MNb, t= MNb;
Step (i):
for(j=0;j<N/2;j++)
for (i=0;i<M/2;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f_b(i,j)$, $f_b(i+M/2,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i,j+N/2)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i+M/2,j+N/2)$)
k=k+1;
End
End
for(j=0;j<N;j++)
for (i=0;i<M;i++)
$f_b(i,j) = f_b(i,j) \oplus b_z(t)$;
t=t+1;
End
End
Step(ii):
for(j=0;j<N/2;j++)
for (i=M/4;i<M/2;i++) and ((i=3M/4;i<M;i++)
Swapping $_{bx(k)=1,\ by(k)=0}$ ($f_b(i,j)$, $f_b(i-M/4,j)$)
Swapping $_{bx(k)=0,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i,j+N/2)$)
Swapping $_{bx(k)=1,\ by(k)=1}$ ($f_b(i,j)$, $f_b(i-M/4,j+N/2)$)
k=k+1;
End
End

```
    for(j=0;j<N;j++)
      for (i=0;i<M;i++)
       f_b(i, j) = f_b(i, j) ⊕ b_z(t);
       t=t+1;
      End
     End
Step (iii):
   for(j=N/4;j<N/2;j++) and (j=3N/4;j<N;j++)
     for (i=M/2;i<M;i++)
       Swapping bx(k) =1, by(k) =0 ( f_b(i,j) , f_b(i-M/2,j) )
       Swapping bx(k) =0, by(k) =1 ( f_b(i,j) , f_b(i,j-N/4) )
       Swapping bx(k) =1, by(k) =1 ( f_b(i,j) , f_b(i-M/2,j-N/4) )
       k=k+1;
      End
    End
   for(j=0;j<N;j++)
     for (i=0;i<M;i++)
       f_b(i, j) = f_b(i, j) ⊕ b_z(t);
       t=t+1;
      End
     End
Step (iv):
   for(j=N/2;j<N;j++)
    for (i=0;i<M/2;i++)
      Swapping bx(k) =1, by(k) =0 ( f_b(i,j) , f_b(i+M/2,j) )
      Swapping bx(k) =0, by(k) =1 ( f_b(i,j) , f_b(i,j-N/2) )
      Swapping bx(k) =1, by(k) =1 ( f_b(i,j) , f_b(i+M/2,j-N/2) )
      k=k+1;
     End
   End
for(j=0;j<N;j++)
      for (i=0;i<M;i++)
       f_b(i, j) = f_b(i, j) ⊕ b_z(t);
       t=t+1;
      End
     End
```

From Fig.6 and Fig.7, one can see the difference between *scheme B2* and *scheme B3*. In *scheme B2*, ***Bit XOR Function*** is performed once after swapping, while in *scheme B3* four ***Bit XOR Function*** are performed.

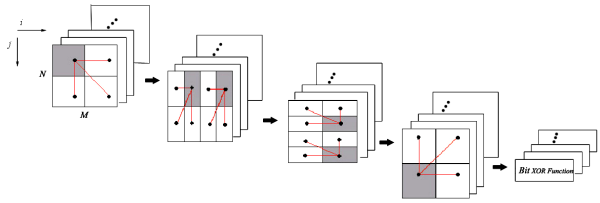


Fig.6 The encryption action of *Scheme B2*.

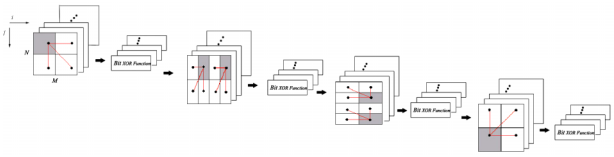


Fig.7 The encryption action of *scheme B3*.

3. Experimental Results

In this section, the simulation results of the proposed image encryption/decryption algorithms are demonstrated. Here, we use the following 1-D chaotic systems [13]:

The logistic map:

$$F_\mu(x) = \mu x(1-x) \tag{2}$$

where μ is the parameter between 0 and 4.

The skew tent map:

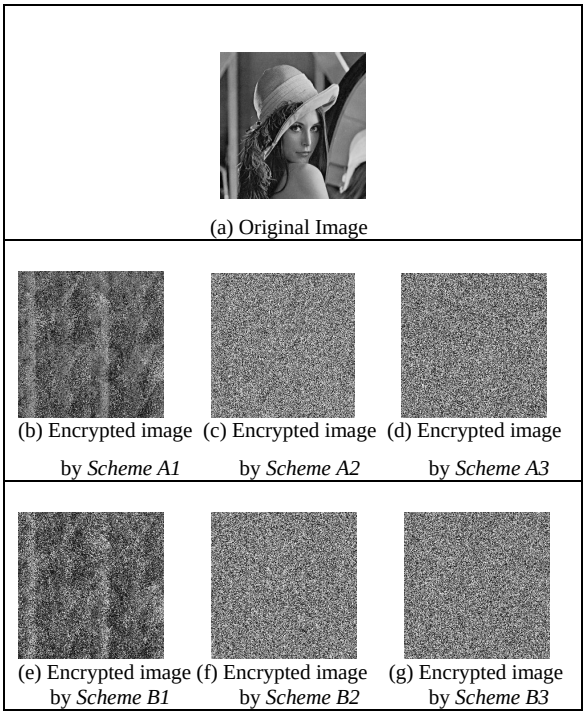
$$f_a = \begin{cases} \dfrac{x}{a} & , 0 < x \le a \\ \dfrac{x-1}{a-1} & , a < x \le 1 \end{cases} \tag{3}$$

The tent map:

$$\Delta_\mu(x) = \mu(1-2\left|x-\frac{1}{2}\right|) = 2\mu\begin{cases} x, & \text{if } 0 \le x \le 1 \\ 1-x, & \text{if } \frac{1}{2} \le x \le 1 \end{cases} \tag{4}$$

The parameters and the initial states of three chaotic maps can be treated as the secret keys in this encryption/decryption algorithm. By setting μ= 3.96, $x(0)$=0.75 in logistic map , a=0.42, $y(0)$=0.65 in skew tent map and μ= 0.72, $z(0)$=0.80 in tent map, the gray image "Lena" of size 256×256 is simulated as shown in Fig.8. In the simulation, an image f is regarded as a surface with $z = f(x,y)$ in R^3. Moreover, to demonstrate how rough the encrypted image surface is, its fractal dimension D is calculated according to the method proposed by Chen et al. [12]. In calculating the fractal dimension, the maximal distance between two pixels is set at 10. The fractal dimensions of the original and encrypted image are calculated as listed in Table 3. By visual perception, the encrypted images are in chaos and completely undistinguishable. Since the maximal fractal dimension for a surface is 3, it is evident that all the encrypted images were in a state of chaos after our encryption/decryption algorithms. The simulated results demonstrated that our image encryption algorithms achieve high security level as other existed methods.

Moreover, it is observed that the encrypted images of all the proposed schemes with different computational complexity are all undistinguishable by visual perception. On the other hand, it is clear that the computational complexity of these schemes is *Scheme B3> Scheme B2> Scheme B1> Scheme A3> Scheme A2> Scheme A1*, and the security level is also *Scheme B3> Scheme B2> Scheme B1> Scheme A3> Scheme A2> Scheme A1*. Any one of schemes with different security levels and computational complexity may be chosen to suit for different applications according to the requirements by users.



(a) Original Image

(b) Encrypted image by *Scheme A1* (c) Encrypted image by *Scheme A2* (d) Encrypted image by *Scheme A3*

(e) Encrypted image by *Scheme B1* (f) Encrypted image by *Scheme B2* (g) Encrypted image by *Scheme B3*

Fig. 8 The simulation results of image encryption algorithms

Table 3 The fractal dimensions of the original and encrypted image

	FD
Original Image	2.48509
Scheme A1	2.96896
Scheme A2	3
Scheme A3	3
Scheme B1	2.98826
Scheme B2	2.99786
Scheme B3	3
BRIE	2.98439
CMLIE	2.98502
HCIE	2.77220

4. Conclusion

In this paper, we proposed some encryption/decryption algorithms based on chaos. We take advantages of the superior properties of the chaos to enhance the security level. The algorithms presented have some good features: (i) reconstruction with no distortion (ii) low complexity (iii) high security. It is evident that the performance of schemes with high computational complexity are better than that the schemes with low computational complexity. Any one of schemes may be chosen to suit for different applications according to the requirements by users.

References

[1] Jui-Cheng Yen and J. I. Guo, "A New Chaotic Mirror-Like Image Encryption Algorithm and its VLSI Architecture", *Pattern Recognition and Image Analysis*, vol.10, no.2, pp.236-247, 2000.

[2] M. Salleh and S. Ibrahim and I.F. Isnin, "Enhanced chaotic image encryption algorithm based on Baker's map," *IEEE, Proceedings of the 2003 International Symposium on Circuits and Systems*, Vol.2, pp.508-511, May 2003.

[3] Y J.-C. Yeo and J.-I Guo, "Efficient hierarchical chaotic image encryption algorithm and its VLSI realization," *Vision, Image and Signal Processing, IEE Proceedings*, Vol.147, pp.167-175, April 2000.

[4] Jiri Fridrich, "Image encryption based on chaotic maps," *IEEE* , International Conference on *Systems, Man, and Cybernetics, 1'Computational Cybernetics and Simulation'*, Vol. 2, pp.1105 – 1110, Oct.1997.

[5] Jiri Fridrich, "Symmetric ciphers based on two dimensional chaotic maps," *International Journal of Bifurcation and Chaos (IJBC) in Applied Sciences and Engineering*, Vol. 8 , No. 6, pp.1259—1284, 1998.

[6] J Jui-Cheng Yen and Jiun-In Guo, "A new image encryption algorithm and its VLSI architecture," Workshop on *IEEE Signal Processing Systems*, pp.430 – 437, 20-22 Oct. 1999.

[7] S. Su, A. Lin and Jui-Cheng Yen, "Design and realization of a new chaotic neural encryption/decryption network," *The 2000 IEEE Asia-Pacific Conference on Circuits and System*, pp.335 – 338, *4-6* Dec. 2000.

[8] Jui-Cheng Yen and Jiun-In Guo, "A new chaotic key-based design for image encryption and decryption," *The 2000 IEEE International Symposium on Circuits and Systems*, Vol. 4, pp.49 – 52, May 2000.

[9] Shujun Li and Xuan Zheng, "Cryptanalysis of a chaotic image encryption method," *ISCAS 2002, IEEE International Symposium on Circuits and Systems*, Vol. 2, pp.708-711, May 2002.

[10] C.J. Kuo and M.S Chen, "A new signal encryption technique and its attack study," *IEEE International Carnahan Conference on Security Technology*, pp.149 – 153, Oct. 1991.

[11] T.S. Parker and L.O. Chua, "Chaos—A Tutorial for Engineers," *Proc. IEEE*, Vol. 75(8), pp. 982–1008, 1987.

[12] C.-C. Chen, J.S. DaPonte and M.D. Fox," Fractal feature analysis and classification in medical imaging," *IEEE Transactions on Medical Imaging*, Vol. 8, pp.133-142, June 1989.

[13] S.Neil Rasband, *Chaotic Dynamics of Nonlinear Systems*, New York, 1992.