CHAPTER 5

INTERTWINING CHAOTIC MAPS BASED IMAGE ENCRYPTION

5.1 INTRODUCTION

In any communication system, including the satellite and the Internet, it is almost impossible to prevent unauthorized people from eavesdropping. When information is broadcast from a satellite or transmitted through the Internet, there is a risk of information interception. Security of image and video data has become increasingly important for many applications, including video conferencing, secure facsimile, medical and military applications.

Most chaotic image encryptions or encryption systems use the permutation substitution architecture. These two processes are repeated for several rounds, to obtain the final encrypted image. Fridrich (1998) suggested a chaotic image encryption method composed of permutation and substitution. All the pixels are moved using a 2D chaotic map. The new pixels moved to the current position are taken as a permutation of the original pixels. In the substitution process, the pixel values are altered sequentially. Chen et al (2004) employed a three dimensional 3D Arnold cat map and a 3D Baker map in the permutation stage. Lian et al (2005) used a chaotic standard map in the permutation stage and a quantized logistic map in the substitution stage. The cryptanalysis techniques on chaotic maps have also been attempted in recent years (Alvarez et al 2009, Li et al 2009, 2009). Most of the schemes in the
literature are linear in nature, weak in key and fail to withstand brute-force, known/chosen plaintext attacks.

This chapter addresses a sequence of equivalent keys, weak keys and the weakness of the Patidar et al (2010) scheme. The design of an intertwining chaotic maps based scheme is done to overcome the aforementioned weaknesses. The maps are intertwining together well to increase the pseudorandom values. Moreover, the chaotic sequences are uniformly distributed and the key size has increased considerably. The algorithm uses significant features, such as sensitivity to the initial condition, permutation of keys, intertwining chaotic maps, byte substitution, nonlinear diffusion and sub diagonal diffusion. The scheme is robust and secure, and can be used for secure image applications.

Patidar et al (2009) proposed that logistic and standard maps could be used to generate a pseudorandom number sequence (PRNS), controlling two kinds of encryption operations. This scheme was later cryptanalyzed by Rhouma et al (2010), and they found that it was not secure in the sense that an equivalent key can be obtained from only one known/chosen plain-image and the corresponding cipher-image. In order to resist Rhouma et al's attack, a modified version of the original scheme was proposed by Patidar et al (2010). It is claimed that the modified image cipher preserves all the good properties of the original cipher, and is also capable of withstanding the chosen-plaintext and known-plaintext attacks. But the scheme is found to be insecure by Li et al (2011). The major steps are as follows:

1) Modified Horizontal Diffusion (mHD):

\[ mHD(X) = HD(X) \oplus mHD(\bigodot) \]

where X is the input matrix and \( \bigodot \) is a zero matrix of the same size as X.
2) Modified Vertical diffusion (mVD):

\[ mVD(X) = VD(X) \oplus mVD(\overline{X}) \]

3) The encryption procedure of the equivalent is as follows

\[ I' = VD(HD(I)) \oplus I_{key} \]

where

\[ I_{key} = VD(HD(I_{skey})) \oplus VD(mHD(R)) \oplus mVD(\overline{R}) \oplus I_{CKS} \]

4) From the properties of step 1 and 2, \( I' \) can also be written as

\[ I' = mVD(mHD(I \oplus I_{skey})) \oplus I_{CKS} \]

\[ = mVD(HD(I \oplus I_{skey}) \oplus mHD(\overline{R})) \oplus I_{CKS} \]

\[ = VD(HD(I \oplus I_{skey})) \oplus VD(mHD(\overline{R})) \oplus mVD(\overline{R}) \oplus I_{CKS} \]

\[ = VD(HD(I)) \oplus VD(HD(I_{skey})) \oplus VD(mHD(\overline{R})) \oplus mVD(\overline{R}) \oplus I_{CKS} \]

\[ = VD(HD(I)) \oplus \tilde{I}_{key} \]

Since the above operations are linear in nature, the scheme is highly vulnerable to linear attacks. Note that the equivalent key \( \tilde{I}_{key} \) can be obtained as

\[ \tilde{I}_{key} = VD(HD(I)) \oplus I' \]

In order to address the problem, this scheme introduced nonlinear operations.
5.2 INTERTWINING LOGISTIC MAP

The proposed intertwining chaotic maps are defined as follows:

\[ x_{n+1} = [\mu \times y_n \times (1 - x_n) \times k_1 + z_n] \mod 1 \]
\[ y_{n+1} = [\mu \times y_n + \sin(z_n) \times k_2 \times x_{n+1}] \mod 1 \]
\[ z_{n+1} = [\mu \times (z_n + y_{n+1}) \times k_3 \times (1 - x_{n+1})] \mod 1 \]

where \( 0 \leq \mu \leq 3.999, \ |k_1| > 33.5, \ |k_2| > 37.9 \) and \( |k_3| > 35.7 \) are used to increase the chaotic keys. This is discussed in section 5.3.1.

5.3 PROPOSED SCHEME

The architecture of the proposed intertwining chaotic maps based image cryptosystem is shown in Figure 5.1. The scheme consists of four major phases, permutation, byte substitution, nonlinear diffusion and sub diagonal diffusion. The permutations on pixel position, the change of pixel value and byte substitution are carried out to enable the confusion process. Two rounds of operations are carried out. The pixel value mixing effect of the whole cryptosystem is altered greatly.

![Figure 5.1 Architecture of the proposed image cryptosystem](image-url)
The plain image is stored in a two dimensional array of \([R_{i,j}, G_{i,j}, B_{i,j}]\) pixels. Here, \(1 \leq i \leq H\) and \(1 \leq j \leq W\), where \(H\) and \(W\) represent the height and width of the plain-image.

### 5.3.1 Key Generation

Three initial values are used for the key generation. They are randomly chosen float numbers stored in \(x_{1,1}, y_{1,1}\) and \(z_{1,1}\). They are used as the secret keys for the scheme. The corresponding integer values are \(X_{1,1} = \lfloor x_{1,1} \times 256 \rfloor\), \(Y_{1,1} = \lfloor y_{1,1} \times 256 \rfloor\) and \(Z_{1,1} = \lfloor z_{1,1} \times 256 \rfloor\) respectively. The key stream is generated with the help of the intertwining chaotic maps as follows:

\[
\begin{align*}
\text{for } i &= 1 \text{ to } 256 \\
\text{for } j &= 1 \text{ to } 256 \\
\text{for } i = 1 \text{ to } 256 \\
\text{for } j &= 1 \text{ to } 256 \\
\end{align*}
\]

\[
\begin{align*}
x_{i,j+1} &= (3.72842 \times y_{i,j}(1 - x_{i,j}) \times k_1 + z_{i,j}) \mod 1 \\
y_{i,j+1} &= (3.69339 \times y_{i,j} + \sin(z_{i,j}) \times k_2 \times x_{i,j+1}) \mod 1 \\
z_{i,j+1} &= (3.82386 \times (z_{i,j} + y_{i,j+1}) \times k_3 \times (1 - x_{i,j+1})) \mod 1 \\
X_{i,j} &= \lfloor x_{i,j+1} \times 256 \rfloor \\
Y_{i,j} &= \lfloor y_{i,j+1} \times 256 \rfloor \\
Z_{i,j} &= \lfloor z_{i,j+1} \times 256 \rfloor \\
\end{align*}
\]

end

\[
\begin{align*}
x_{i+1,1} &= x_{i,j+1} \\
y_{i+1,1} &= y_{i,j+1} \\
z_{i+1,1} &= z_{i,j+1} \\
\end{align*}
\]

end

where \(X_{i,j}\), \(Y_{i,j}\), \(Z_{i,j}\) are the set of chaotic keys. The values lie between 3.72842, 3.69339 and 3.82386. They must be in between 3.567 to 3.999 to achieve chaotic behavior. The size is the same as the plain-image. Moreover,
three float numbers are used as multipliers $k_1$, $k_2$, $k_3$ in the proposed maps to increase the randomness and uniform distribution of the key values.

### 5.3.2 Permutation with XORing Operation

Permutation transformations are basic operations in many scrambling and encryption systems. There are two iterative stages in the chaos based image cryptosystem. Generally, the confusion effect is obtained in the permutation stage, while the diffusion effect is obtained in the pixel value diffusion stage. Confusion makes the relationship between the key and the ciphertext as complex as possible.

The confusion stage permutes the pixels in the image, without changing its value. The pixel values are modified sequentially in the diffusion stage, so that a small change in one pixel in the image causes an enormous difference in the whole image. In order to decorrelate the relationship between adjacent pixels, the permutation of pixels is introduced in the confusion stage. The confusion stage of the proposed scheme is composed of position permutation, byte substitution and simple pixel value modification. Here, the process uses six odd random secret key values for scrambling the plain-image, and then XORing it with the first chaotic key for pixel value modification simultaneously. The pixels are permuted using the following operations:

for $i = 1$ to $H$
    for $j = 1$ to $W$
        $\text{CR}_{i,j} = R[1 + (p_1 \times i + 3) \mod 256, 1 + (p_2 \times j + 3) \mod 256] \oplus X_{i,j}$
        $\text{CG}_{i,j} = G[1 + (p_3 \times i + 3) \mod 256, 1 + (p_4 \times j + 3) \mod 256] \oplus X_{i,j}$
        $\text{CB}_{i,j} = B[1 + (p_5 \times i + 3) \mod 256, 1 + (p_6 \times j + 3) \mod 256] \oplus X_{i,j}$
    end
end
where $X_{i,j}$ is the first chaotic key, $R[i, j]$, $G[i, j]$ and $B[i, j]$ represent the red, green, blue channels in the plain-image, and $CR_{i, j}$, $CG_{i, j}$, $CB_{i, j}$ denote the $(i,j)^{th}$ pixel of the permuted image. The method uses $p_1$, $p_2$, $p_3$, $p_4$, $p_5$ and $p_6$, which are odd constant values to improve the pixel scrambling of the image. In order to get a reversible permutation $p_1$, $p_3$, $p_5$ must be chosen to be relatively prime to $H$ and $p_2$, $p_4$, $p_6$ must be relatively prime to $W$. The typical images have height and width as even numbers, and in such cases the $p_1$, $p_2$, $p_3$, $p_4$, $p_5$, $p_6$ must be odd. The above permutation operations are good, and hence, enhance the security against known-plaintext attacks.

5.3.3 Byte Substitution

Each individual RGB pixel byte of the state is replaced with a new byte by using the S-box of the Advanced Encryption Standard (AES) algorithm. The size of the S-box is 16×16. Here, the RGB component of each pixel is divided into two parts; the left half consists of the left most four bits, and the right half consists of the right most 4-bits. They are denoted by $LR$, $RR$, $LG$, $RG$, $LB$, $RB$. For the red channel, the byte substitution is performed treating $LR$ as the row number and $RR$ as the column number of the S-box. Similarly, the other channel substitutions are done. The resultant values are XORed with the second chaotic key. It improves the security against the known/chosen plaintext attacks.

5.3.4 Nonlinear Diffusion

Diffusion refers to the property that redundancy in the statistics of the plaintext is dissipated in the statistics of the cipher text. The RGB diffusion is obtained by the 5 Least Significant Bits (LSB) circular shift method. The resultant values are again XORed with the first chaotic key to all the red, green and blue channels. The procedure for the nonlinear diffusion is as follows:
for $i = 1$ to $H$

for $j = 1$ to $W$

$$\text{CR}_{i,j} = (R_{i,j} \gg 5) \mod 256$$

$$\text{CG}_{i,j} = (G_{i,j} \gg 5) \mod 256$$

$$\text{CB}_{i,j} = (B_{i,j} \gg 5) \mod 256$$

$$\text{CSR}_{i,j} = \text{CR}_{i,j} \oplus X_{i,j}$$

$$\text{CSG}_{i,j} = \text{CG}_{i,j} \oplus X_{i,j}$$

$$\text{CSB}_{i,j} = \text{CB}_{i,j} \oplus X_{i,j}$$

end

end

where $X_{i,j}$ is the first chaotic key. $\text{CR}_{i,j}$, $\text{CG}_{i,j}$ and $\text{CB}_{i,j}$ denote the resultant values of the circular shift operation, and $\text{CSR}_{i,j}$, $\text{CSG}_{i,j}$ and $\text{CSB}_{i,j}$ denote the nonlinear diffusion image of the XORed operation. The combination of the 5 bit circular shift and XORing makes the encryption operation nonlinear, and hence, the system becomes strong against known/chosen-plaintext attacks.

### 5.3.5 Sub Diagonal Diffusion

Diffusion is obtained with the help of sub diagonal XORing, XORing with the third chaotic key and so on. The sub diagonal operation is shown in Figure 5.2.
Figure 5.2 Sub diagonal pixel value reading

The procedure for the red channel is as follows: Choose three random integers $R_0$, $G_0$ and $B_0$ in the range of 1 to 256.

$IvR = R_0$
$IvG = G_0$
$IvB = B_0$

for $i = 1 : \text{max}$
    $outR(i, \text{max} - i + 1) = in(i, \text{max} - i + 1) \oplus IvR$
    $IvR = outR(i, \text{max} - i + 1) \oplus Z_{i,j}$
end

for $j = 1 : \text{max} - 1$
    for $i = 1 : \text{max} - j$
        $outR(i, \text{max} - (j - 1) - i) = in(i, \text{max} - (j - 1) - i) \oplus IvR$
        $IvR = outR(i, \text{max} - j - 1) \oplus Z_{i,j}$
    end
    for $i = 1 : \text{max} - j$
        $outR(i + j, (\text{max} - i + 1)) = in(i + j, (\text{max} - i + 1))$
        $IvR = outR(i + j, \text{max} - i + 1) \oplus Z_{i,j}$
    end
Here, max is the maximum size of the image, in and outR are the input and output of the image, and IvR, IvG, IvB are the initial vectors of each channel which may also be treated as an 8-bit secret key. Z_{i,j} is the third chaotic key. The pixel is modified by XORing the first pixel and second pixels with the chaotic key, the third pixel is modified by XORing the modified second and third pixels with the key, and the process continues till the end of the image. A similar procedure is applied for the other channels. These operations enhance the diffusion property and hence improve the security levels.

5.3.6 Decryption

The decryption algorithm is just the reverse of the encryption one. In order to get the original image, the encrypted image pixel values are XORed with the same set of secret keys which were used in the encryption process.

5.3.6.1 Inverse sub diagonal diffusion

The inverse procedure for the red channel is as follows:

\[ \text{IvR} = R_0 \]
\[ \text{IvG} = G_0 \]
\[ \text{IvB} = B_0 \]

for \( i = 1 : \text{max} \)

\[ \text{IvR}_1 = \text{in}(i, \text{vmax} - i + 1) \]
\[ \text{outR}(i, \text{max} - i + 1) = \text{in}(i, \text{max} - i + 1) \oplus \text{IvR} \]
\[ \text{IvR} = \text{IvR}_1 \oplus Z_{i,j} \]
end

for j = 1 : max - 1
    for i = 1 : max - j
        IvR1 = in(i, max - (j - 1) - i)
        outR(i, max - (j - 1) - i) = in(i, max - (j - 1) - i) ⊕ IvR
        IvR = IvR1 ⊕ Z_{i,j}
    end
for i = 1 : max - j
    IvR1 = in(i + j, (max - i + 1))
    outR(i + j, (max - i + 1)) = in(i + j, (max - i + 1))
    IvR = IvR1 ⊕ Z_{i,j}
end
end

5.3.6.2 Inverse nonlinear diffusion

The inverse nonlinear diffusion is obtained by the 5 bit left circular shift method. The resultant values are XORed with the same RGB channels. The procedure for the inverse nonlinear diffusion is as follows:

for i = 1 to H
    for j = 1 to W
        CR_{i,j} = (R_{i,j} \ll 5) \mod 256
        CG_{i,j} = (G_{i,j} \ll 5) \mod 256
        CB_{i,j} = (B_{i,j} \ll 5) \mod 256
        CSR_{i,j} = CR_{i,j} \oplus X_{i,j}
        CSG_{i,j} = CG_{i,j} \oplus X_{i,j}
        CSB_{i,j} = CB_{i,j} \oplus X_{i,j}
    end
end
5.3.6.3 **Inverse byte substitution**

The inverse S-box is used for byte substitution. Each individual RGB pixel byte of the state is replaced with a new byte by using the inverse S-box of the AES algorithm. The same technique is followed as for encryption, to replace the substituted values. Finally, the inverse resultant values are XORed with the second chaotic key.

5.3.6.4 **Inverse permutation**

The permutation is replaced by the inverse value permutation. The inverse method is described by

for i = 1 to H
    for j = 1 to W
        $R_{ij} = X_{ij} \oplus R[((i-3) \times p_1^{-1}) \bmod 256] + 1, ((j - 3) \times p_2^{-1}) \bmod 256] + 1]$ 
        $G_{ij} = X_{ij} \oplus G[((i-3) \times p_3^{-1}) \bmod 256] + 1, ((j - 3) \times p_4^{-1}) \bmod 256] + 1]$ 
        $B_{ij} = X_{ij} \oplus B[((i-3) \times p_5^{-1}) \bmod 256] + 1, ((j - 3) \times p_6^{-1}) \bmod 256] + 1]$ 
    end
end

where $p_1^{-1}, p_2^{-1}, p_3^{-1}, p_4^{-1}, p_5^{-1}$ and $p_6^{-1}$ are the inverse odd random values. The original image can be recovered, once the above decryption process is completed.

5.4 **SECURITY ANALYSIS**

A good encryption scheme should resist all kinds of known attacks, such as known-plaintext, ciphertext, statistical, differential, and various brute
force attacks. Several security analyses have been performed on the proposed image encryption scheme, including the most important ones like the key sensitivity, statistical, and differential analyses, which have demonstrated the robust security of the new scheme, as shown in the following.

5.4.1 Statistical Analysis

Statistical analysis has been performed on the proposed image encryption scheme, demonstrating its superior confusion and diffusion properties, which strongly resist statistical attacks. This is shown by the test on the histograms of the enciphered images, and on the correlations of the adjacent pixels in the ciphered image.

5.4.2 Histogram Analysis

The histogram analysis is used to illustrate the confusion and diffusion properties in the encrypted data. The USC-SIPI image database is used (freely available at http://sipi.usc.edu/database/) for testing purposes. In the permutation process, the odd values are taken as $p_1=7$, $p_2=31$, $p_3=23$, $p_4=9$, $p_5=15$, $p_6=91$. In the inverse permutation process the values are used as $p_1^{-1}=183$, $p_2^{-1}=223$, $p_3^{-1}=167$, $p_4^{-1}=57$, $p_5^{-1}=239$, $p_6^{-1}=211$. The histogram of the plain image 'Lena' and the histogram of the encrypted image are shown in Figure 5.3. Comparing the two, it may be observed that the histogram of the encrypted image is uniform and is significantly different from that of the original image, and that the encrypted images transmitted, do not provide any suspicion to the attacker and can resist statistical attacks strongly.
Figure 5.3 Histogram of plain image Lena and its encrypted image
5.4.3 Correlation of Two Adjacent Pixels

The effect of image scrambling is related to the correlation of adjacent pixels. A better scrambling effect indicated a lower correlation value. In order to test the correlation between the plain-image and the cipher-image, pairs of plain-image channels and cipher-image channels are analyzed. The following formulae are used to calculate the correlation coefficients in the horizontal, vertical and diagonal directions. The calculated results are listed in Table 5.1.

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

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

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

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

where \( x \) and \( y \) denote two adjacent pixels and \( N \) is the total number of duplets \((x, y)\) obtained from the image.

The proposed cipher image channel values are close to zero, and hence, show a better scrambling effect.
Table 5.1 Correlation coefficients of two adjacent pixels in a plain-image and a cipher-image

<table>
<thead>
<tr>
<th>Directions</th>
<th>Plain-image</th>
<th>Cipher-image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>Horizontal</td>
<td>0.9518</td>
<td>0.9587</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9753</td>
<td>0.9747</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9317</td>
<td>0.9378</td>
</tr>
</tbody>
</table>

5.4.4 Key Space Analysis

An ideal encryption scheme should have larger secret keys, and the key space should be large enough to make brute-force attacks infeasible. In the proposed scheme, the initial conditions and parameters of three maps are used as keys. The multipliers $k_1$, $k_2$ and $k_3$ are treated as keys in the intertwining maps, the key space is approximately $2^{192}$. As IvR, IvG, IvB are also used as part of the key, the key space is increased up to $2^{216}$. The combination of the key space is large enough in the proposed scheme, to resist any attacks.

Table 5.2 Key space of the proposed scheme and different encryption scheme

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Key size</td>
<td>$2^{216}$</td>
<td>$2^{157}$</td>
<td>$2^{178}$</td>
</tr>
</tbody>
</table>

As shown in Table 5.2, the proposed scheme has the largest key size among the schemes.
5.4.5 Sensitivity Analysis

Key sensitivity means that the change of a single bit in the secret key produces a completely different encrypted image. The two cipher-images are compared pixel-by-pixel. Different images and different key values were tested. A test image 'pepper' is encrypted with the following set of secret keys: $x_0=0.41324738544344$, $y_0=0.52638928350638$, $z_0=0.98644737157579$, $k_1=33.1$, $k_2=37.3$, $k_3=35.7$ and IvR=35, IvG=25, IvB=65. The output cipher is shown in Figure 5.4(b).

A single key bit changed in any one of the key. Here, the single key value changed as: $x_0=0.41324738544345$, $y_0=0.52638928350638$, $z_0=0.98644737157579$, $k_1=33.1$, $k_2=37.3$, $k_3=35.7$ and IvR=35, IvG=25, IvB=65. The output cipher is shown in Figure 5.4(c). The result shows 99.82% difference between the two cipher-images. It shows that this algorithm has a great sensitivity to the key and plaintext.

![Figure 5.4](image)

Figure 5.4 Key sensitivity analysis of the proposed scheme on the test image ‘pepper’ (a) Plain-image, (b) Cipher-image using single bit change of plaintext and (c) Cipher-image using single bit change of secret key
5.4.6 Differential Analysis

In order to assess the effect of changing a single bit of key or any pixel value in the plain-image on the cipher-image, the Number of Pixel Change Rate (NPCR) and the Unified Averaged Changing Intensity (UACI) are computed in the proposed scheme. The NPCR is used to measure the change rate of the number of pixels of the cipher-image when only one bit of key or pixel is modified. The UACI measures the average intensity of the one bit changes of cipher-images. Let us assume that two ciphered images $C^1$ and $C^2$ whose corresponding plain images have only one-pixel difference. The color RGB-level values of the ciphered images $C^1$ and $C^2$ at row i, column j are labeled as $C^1(i,j)$ and $C^2(i,j)$, respectively. The NPCR is defined as:

$$NPCR = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} D(i,j)}{W \times H} \times 100\%$$

where $W$ and $H$ are the width and height of two random images and $D(i, j)$ is defined as

$$D(i, j) = \begin{cases} 0 & C^1(i, j) = C^2(i, j) \\ 1 & C^1(i, j) \neq C^2(i, j) \end{cases}$$

Further, the UACI is used to measure the average intensity difference in a color component between the two cipher images $C^1(i, j)$ and $C^2(i, j)$. It is defined as

$$UACI = \frac{1}{W \times H} \left[ \sum_{i,j}^{H,W} \left| \frac{C^1(i,j) - C^2(i,j)}{2^L - 1} \right| \right] \times 100\%$$

where $L$ is the number of bits used to represent the color component of red, green and blue respectively. The results of NPCR and UACI are presented in Table 5.3 for the different images.
Table 5.3 NPCR and UACI criteria of the proposed scheme

<table>
<thead>
<tr>
<th>Cipher images</th>
<th>NPCR%</th>
<th>UACI%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>Lena</td>
<td>99.5621</td>
<td>99.5773</td>
</tr>
<tr>
<td>Baboon</td>
<td>99.6214</td>
<td>99.6105</td>
</tr>
<tr>
<td>House</td>
<td>99.6328</td>
<td>99.6278</td>
</tr>
<tr>
<td>Tree</td>
<td>99.6292</td>
<td>99.6154</td>
</tr>
</tbody>
</table>

Wu et al (2011) tested and claimed that many existing image encryption methods are actually not as good as they are purported, although some methods do pass these randomness tests. The results show that a small change in the original image will result in a significant difference in the cipher-image; proposed scheme can be found that the NPCR > 99.63% and the UACI > 33.43%. So the scheme has good ability to withstand a differential attack.

The performance of each stage of the difference between the cipher and plain-images is measured by the Mean Absolute Error (MAE) which is defined as:

\[ MAE = \frac{1}{W \times H} \sum_{i=1}^{H} \sum_{j=1}^{W} |P_{i,j} - C_{i,j}| \]

where the parameters \( P_{i,j} \) and \( C_{i,j} \) are pixel values of the plain and cipher images. The high MAE value indicates the better security. The results for the MAE values are shown in Table 5.4.
Table 5.4 MAE in different stages

<table>
<thead>
<tr>
<th>Proposed stages</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permutation</td>
<td>80.9610</td>
</tr>
<tr>
<td>Byte Substitution</td>
<td>81.5108</td>
</tr>
<tr>
<td>Nonlinear diffusion</td>
<td>82.1309</td>
</tr>
<tr>
<td>Sub diagonal diffusion</td>
<td>84.2461</td>
</tr>
</tbody>
</table>

It is found that the value in the proposed scheme is high. Thus, the cryptosystem is better in security.

5.4.7 Performance Analysis

Apart from the security consideration, the running speed of the algorithm is also an important aspect for a good encryption algorithm. The simulator for the proposed scheme was implemented using MATLAB 7.4. The performance was measured on a 3.0 GHz Pentium Core 2 Duo with the 4 GB RAM running Windows Vista Business Edition. Simulation shows that the average running speed is 0.6372 (s) for encryption and 0.6516 (s) for decryption.

5.4.8 Avalanche Criterion

A small change in either the key or the plaintext should cause a drastic change in the cipher-text, ideally 50% difference in the bits of the cipher. The analysis exhibited that the changing rate of bits is 49.97%. So, the proposed scheme is nearly ideal.

5.4.9 Information Entropy Analysis

Information entropy is one of the criteria to measure the strength of a symmetric cryptosystem. The entropy $H(m)$ of a message $m$ can be calculated as
\[ H(m) = \sum_{i=0}^{L-1} P(m_i) \log_2 \frac{1}{P(m_i)} \text{(bits)} \]

where \( P(m_i) \) represents the probability of the occurrence of symbol \( m_i \) and \( \log \) denotes the logarithm to the base 2. \( L \) is the total number of symbols \( m \). If there are 256 possible outcomes of the message \( m \) with equal probability, it is considered as random. In this case, \( H(m) = 8 \), is an ideal value.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>7.9994</td>
<td>7.9884</td>
<td>7.9891</td>
</tr>
<tr>
<td>Lion</td>
<td>7.9993</td>
<td>7.9879</td>
<td>7.9895</td>
</tr>
</tbody>
</table>

As shown in Table 5.5, we notice that the values obtained in the proposed scheme are closer to the theoretical value of 8, than the other schemes. This means that information leakage in the encryption process is negligible, and the encryption system is secure against entropy attack.

5.5  CONCLUSION

In this chapter, an intertwining chaotic maps based image encryption scheme is discussed. The proposed cipher provides good confusion and diffusion properties that ensure high security. Confusion and diffusion are achieved using permutation, byte substitution, nonlinear diffusion and sub diagonal diffusion. This scheme is immune to various types of cryptographic attacks, like known/chosen plaintext attacks and brute force attacks. Statistical analysis, key sensitivity analysis, key space analysis differential analysis and entropy analysis are carried out to demonstrate the security of the new image encryption procedure. Since the entropy is found to be close to the theoretical
value, it may be observed that the information leakage is negligible, and hence, the scheme is highly secure. The experimental results also show that the performance of the proposed scheme is fast, and hence, more suitable for real time image encryption for transmission applications.