CHAPTER V
SIMULATION ANALYSIS AND EXPERIMENTAL RESULTS

A good Biometric Encryption procedure should be robust against all kinds of
cryptanalytic, statistical and brute-force attacks. In this thesis, the security analysis of
the proposed nonlinear biometric encryption scheme such as statistical analysis,
sensitivity analysis with respect to the key and plaintext, key space analysis etc. to
prove that the proposed cryptosystem is secure against the most common attacks.

5.1 Statistical Analysis

It is well known that many ciphers have been successfully analyzed with the
help of statistical analysis [36] and several statistical attacks have been devised on
them. Therefore, an ideal cipher should be robust against any statistical attack. To
prove the robustness of the proposed biometric image encryption procedure, this thesis
performed statistical analysis by calculating histograms, the correlations of two
adjacent pixels in the biometric encrypted images and the correlation coefficient for
several images and its corresponding biometric encrypted images of a biometrics
database.

5.2 Histogram

An Image Histogram illustrates how pixels in an image are distributed by
graphing the number of pixel at each color intensity level. The Gray value histogram
of explicit image distributed unevenly with the information itself. The Biometric
Encrypted image histogram is comes under the Section Statistical Testing Procedure,
while the gray value distributed evenly. The gray’s statistical properties of explicit
image is not fully shown in the ciphertext image, making the way that the attacker try to make use of explicit images or select the attracting means ineffectively. It shows that the scheme has good characters in disturbing and that the scheme can effectively resist the attack based on the statistics of the gray value.

5.3 Correlation of Adjacent Pixels

For an ordinary image, each pixel is usually highly correlated with its adjacent pixels either in horizontal, vertical or diagonal directions. These high-correlation properties can be quantified as the correlation coefficient for comparison. Taking the horizontal correlation as an example, for each pixel of the image, a duplet \((p_i, q_i)\), can be found, where \(q_i\) the horizontal adjacent pixel of is \(p_i\). Obviously, there may be more than one duplet for each pixel, and the horizontal correlation coefficient is computed as

\[
\rho = \frac{\text{cov}(p, q)}{\sqrt{D(p)D(q)}} \tag{5-1}
\]

Where, \(D(p) = \frac{1}{M} \sum_{i=1}^{M} (p_i - \overline{p})^2\), \(\text{cov}(p, q) = \frac{1}{M} \sum_{i=1}^{M} (p_i - \overline{p})(q_i - \overline{q})\), \(M\) is the total number if duplets \((p_i, q_i)\) obtained from the image, \(\overline{p}\) and \(\overline{q}\) are the mean values of \(p_i\) and \(q_i\), respectively. For an Image with \(m\times n\) pixels \((m\) and \(n\) indicate the number of pixels in horizontal and vertical directions, respectively), \(M = 2m(n-1)\).

By similar approach, the vertical and diagonal correlation coefficients of an image can also be found. In this section, correlation coefficient procedure shows all the three correlation coefficient of NIST Database Fingerprint Image and those of its encrypted image. It can be observed that the encrypted image obtained from the proposed
scheme retains small correlation coefficients in all directions. The results are also compared with a random image and an encrypted image generated by PRNN, BSTS, BEPM, BEBFCAS, BEOCML and BEBM methods. Based on the Experimental results, the correlation coefficients of the encrypted images are very small, indicating that the attacker cannot obtain any valuable information by exploiting a statistic attack.

\[
\bar{p} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]  

(5-2)

Correlation is a measure of the relation between two or more variables. Correlation coefficients can range from -1.00 to +1.00. The value of -1.00 represents a perfect negative correlation. The value of +1.00 represents a perfect positive correlation. A value of 0.00 represents a lack of correlation.

### 5.3.1 Correlation Coefficient Implementation

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>X-E(X)</th>
<th>(X-E(X))^2</th>
<th>Y</th>
<th>Y-E(Y)</th>
<th>(Y-E(Y))^2</th>
<th>(X-E(X))^*</th>
<th>(Y-E(Y))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\[\sum X \quad \sum (X-E(X))^2 \quad \sum Y \quad \sum (Y-E(Y))^2 \quad \sum (X-E(X))^* \quad (Y-E(Y))\]
5.4 Differential Analysis

In this section, the performance of the proposed biometric image encryption scheme is analyzed in detail.

As a general requirement for all the biometric image encryption schemes, the encrypted image should be greatly different from its original form. To quantify this requirement, two measures, including the number of pixels change rate (NPCR) and unified average changing intensity (UACI) can be adopted [2].

The $NPCR_{R,G,B}$ is used to measure the number of pixels in difference of a color component in two images. Let $C(i, j)$ and $C'(i, j)$ be the $i^{th}$ row and $j^{th}$ column pixel of two images $C$ and $C'$, respectively, the $NPCR_{R,G,B}$ can be defined as

$$NPCR_{R,G,B} = \frac{\sum_{i,j} D_{R,G,B}(i, j)}{N} \times 100\%$$  \hspace{1cm} (5-3)

where $N$ is the total number of pixels in the image and $D_{R,G,B}(i, j)$ is defined as

$$D_{R,G,B}(i, j) = \begin{cases} 0 & C_{R,G,B}(i, j) = C'_{R,G,B}(i, j) \\ 1 & C_{R,G,B}(i, j) \neq C'_{R,G,B}(i, j) \end{cases}$$  \hspace{1cm} (5-4)

where $C_{R,G,B}(i, j)$ and $C'_{R,G,B}(i, j)$ are the values of the corresponding color component red(R), green (G) or blue (B) in the two images, respectively.

Considering two random images, the expected value of $NPCR_{R,G,B}$ is found to be

$$\xi[NPCR_{R,G,B}] = (1 - 2^{-L_{R,G,B}}) \times 100\%$$  \hspace{1cm} (5-5)
Where \( L_{R,G,B} \) is the number of bits used to represent the color component of red, green or blue. For example, for two random images with \( 512 \times 512 \) pixels and 24-bit true color (8 bit for each RGB color component, and hence \( L_r = L_g = L_b = 8 \)), \( \xi[NPCR_r] = \xi[NPCR_g] = \xi[NPCR_b] = 100\% \). Another quantity, \( UACI_{R,G,B} \), is to measure the average intensity differences in a color component and can be defined as

\[
UACI_{R,G,B} = \frac{1}{N} \left( \sum_{i,j} \left| C_{R,G,B}(i,j) - C'_{R,G,B}(i,j) \right| \right) \times 100\%
\]  

(5-6)

In the case of two random images, the expected value of \( UACI_{R,G,B} \) can be computed as

\[
\xi[UACI_{R,G,B}] = \frac{1}{2^{L_{R,G,B}}} \left( \sum_{i=1}^{2^{L_{R,G,B}-1}} i(i+1) \right) \times 100\%
\]  

(5-7)

The \( NPCR_{R,G,B} \) and \( UACI_{R,G,B} \) of two cipher images, from two images with one bit difference in the corresponding color component using the same password, are obtained.

5.5 Speed Performance

The speed of the proposed biometric encryption algorithm is much faster than those existing encryption algorithms. Its average data encryption speed is about 40MB/s with a Pentium IV 2.8 GHz Personal Computer.
5.6 Experimental Design

The Biometrics Encryption is done by Non Linear Algorithms. The BE techniques are implemented to analyze their performance using image processing software IDL version 7.0 and Ajax under Win32 and Linux environment. To evaluate the performance of this proposed novel class of non linear biometric encryption techniques, simulation is carried out with twenty four 24 bit digital color images from NIST Database [37].

Figure 5.1 Testing Images (NIST Database)
Biometric Encryption Framework

Figure 5.2 Biometric Encryption and Authentication Framework
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Logistic Map

![Image](image.png)

Figure 5.3 Logistic Pseudo Random Sequence (Logistic Map-IDL Implementation)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Henon Map

Figure 5.4 Graph View of Henon Map (Henon Map-IDL Implementation)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Henon Map

Figure 5.5 Henon Map Pseudo Random Sequence (Henon Map-IDL Implementation)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Tent Map

Figure 5.6 Key Sequence of Tent Map
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Tent

Figure 5.7 Tent Map Pseudo Random on Sequence (Tent Map-IDL Implementation)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Ikeda Map

Figure 5.8 Key View of Ikeda Map (IDL Implementation)

Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)
Figure 5.9 Ikeda Map Pseudo Random Numbers (IDL Implementation)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

Nonlinear Map: Modified Logistic Map

Figure 5.10 Modified Logistic Map Pseudo Random Numbers (IDL Implementation)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

**Biometric Encryption and Authentication Web Access Framework Using Fingerprint, Iris, Palmprint**

*Security Domain Registration*

![Image](image_url)

Figure 5.11 PHP –AJAX Implementation of Biometric Encryption and Authentication using Nonlinear Dynamics
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

![Biometric Encryption & Authentication - Enrollment](image)

Figure 5.12 Biometric Encryption – Enrollment Module
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

![Biometric Authentication - Verification Form](image)

**Figure 5.13 Biometric Authentication Verification Form**
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

Figure 5.14 (a) Output of the Authentication Module
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

Figure 5.14 (b) Output of the Authentication Module
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

Figure 5.15 Authentication Database (Mysql Database- WAMP Server)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

Figure 5.16 Enrollment Database View (User Information and Fingerprint)
Algorithm: Pseudo Random Numbers using Nonlinear Dynamics (PRNN)

(PHP-AJAX and MySql Module)

Biometric Encryption and Authentication using Pseudo Random Numbers

Figure 5.17 Nonlinear Random Number Storage
Algorithm: Biometric Encryption and Bio-Fusion Authentication using Combined Arnold Shuffling (BEBFCAS)

Procedure: Arnold 3D Shuffling and Grey Level Transformed Biometrics (Step 1)

Figure 5.18 (a) Input Biometric Image; (b) Grey Level Transformed Biometric Image (c) Combined Arnold Shuffling Biometric Image, (d) Reverse Arnold Shuffling Biometric Image, (e) Reverse Combined Arnold Shuffling Biometric Image
Algorithm: Biometric Encryption and Bio-Fusion Authentication using Combined Arnold Shuffling (BEBFCAS)

Procedure: Arnold Shuffling Iteration Level

Figure 5.19 Iteration Levels of Arnold Shuffling: Input Biometric Image (a) Arnold Iteration Level 1 (b) Arnold Iteration Level 2 (c) Arnold Iteration Level 3 (d) Arnold Iteration level 4 (e) Arnold Iteration Level 5 (f) Arnold Iteration Level 6 (g) Arnold Iteration Level 7 (h) Arnold Iteration Level 8
Procedure: Iteration Levels of Combined Arnold Shuffling

Figure 5.20. Input Biometric Image (i) Arnold Iteration Level 9 (j) Arnold Iteration Level 10 (k) Arnold Iteration Level 11 (l) Arnold Iteration Level 12 (m) Arnold Iteration Level 13 (n) Arnold Iteration Level 14 (o) Arnold Iteration Level 15 (p) Arnold Iteration Level 16 (q) Arnold Iteration Level 300
Algorithm: Biometric Encryption and Bio-Fusion Authentication using Combined Arnold Shuffling (BEBFCAS)

Procedure: Biometric Encryption using Combined Arnold Shuffling

Figure 5.21 Biometric Encryption using CAS (a) Fingerprint Input (b) Biometric Encryption using Combined Arnold Shuffling (c) Biometric Decryption and Arnold DeShuffling (d) Biometric Decryption
Algorithm: Biometric Encryption using Permutation Matrix (BEPM)
Procedure: Color Fingerprint Input Image Encryption using Nonlinear Map and Permutation Matrix

Figure 5.22 (a) Input Biometric Image; (b) Red Channel Encryption Biometric Image (c) Green Channel Encryption Biometric Image, (d) Blue Channel Encryption Biometric Image, (e) Encrypted Biometric Image (f) Decryption of Red Channel Biometric Image (g) Decryption of Green Channel Biometric Image (h) Decryption of Blue Channel Biometric Image (i) Decrypted Biometric Image
**Algorithm:** Biometric Encryption using Permutation Matrix

**Procedure:** Biometric Encryption using Hyperchaotic and Permutation Matrix

Figure 5.23 (a) Input Biometric Image (b) Column Encryption using Hyperchaotic Algorithm (c) Decryption of Biometrics.
**Algorithm:** Biometric Encryption using Permutation Matrix

**Procedure:** Biometric Encryption using Hyperchoatic (Row and Column Encryption)

![Figure 5.24](image)

Figure 5.24: a) Input Biometric Image b) HC Key1 Values c) HC Key2 Values
d) Row and Column Encryption using HC e) Decryption of HC f) Histogram of Input Biometrics
g) Histogram of Encrypted Biometrics
**Algorithm:** Biometric Encryption Using Permutation Matrix

**Procedure:** Biometric Encryption using Permutation Matrix and Logistic Map

Figure 5.25 a) Input Biometric Image b) Biometric Encryption Using Permutation Matrix c) Biometric Decryption
**Algorithm:** Biometric Encryption using Permutation Matrix and Logistic Map

**Statistical Analysis:** Biometric Histogram Analysis

---

Figure 5.26 (a) Original Image Histogram (b) Biometric Encrypted Image Histogram
**Algorithm:** Biometric Encryption using Permutation Matrix

**Performance Analysis:** Encrypted Biometric Image Correlation Coefficient (IDL – PSYM Value=3)

![Biometric Input Correlation pixels](image)

Figure: 5.27 Input Biometric Image Correlation Coefficient
Algorithm: Biometric Encryption using Permutation Matrix

Performance Analysis: Encrypted Biometric Image Correlation Coefficient (IDL – PSYM Value=3)

Figure 5.28 Biometric Encryption Correlation Coefficient
**Algorithm:** Biometric Encryption using Permutation Matrix

**Statistical Analysis:** Correlation Coefficients of two adjacent pixels in Encrypted and Decrypted Images

---

**Table 5.2 Correlation Coefficient of Biometric Original and Biometric Encrypted Image**

<table>
<thead>
<tr>
<th>Input Size</th>
<th>Correlation Coefficient of Input Biometrics</th>
<th>Correlation Coefficient of Encrypted Image Biometrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Horizontal</td>
<td>Vertical</td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.08967</td>
<td>-0.32560</td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.39332</td>
<td>-0.39332</td>
</tr>
<tr>
<td>128×128</td>
<td>0.79586</td>
<td>0.61031</td>
</tr>
<tr>
<td>256×256</td>
<td>0.94501</td>
<td>0.87568</td>
</tr>
</tbody>
</table>
Algorithm: Biometric Encryption Using Permutation Matrix

Performance Analysis: BEPM- Differential Attack Analysis

Table 5.3 BEPM- Differential Attack Analysis

<table>
<thead>
<tr>
<th>Input Biometric Size</th>
<th>Number of Pixel Change Rate(NPCR)</th>
<th>Unified Average Changing Intensity(UACI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>32 × 32</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>34.1673</td>
<td>32.0073</td>
</tr>
<tr>
<td>64 × 64</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>34.2435</td>
<td>39.189</td>
</tr>
<tr>
<td>128×128</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>41.2562</td>
<td>43.9838</td>
</tr>
</tbody>
</table>
**Algorithm:** Biometric Encryption using OCML Encryption

**Procedure:** Improved Colour Biometric Encryption based on chaotic map and OCML Model

---

Figure 5.29 (a) Input Biometric Image (b) Biometric Encryption using OCML Method (c) Biometric Decryption
Algorithm: Biometric Encryption using one-way coupled-map lattices Encryption

Performance Analysis: Input Biometric Image Correlation Coefficient

Figure 5.30 Correlation Pixels of Input Biometric Image (OCML Method)
**Algorithm:** Biometric Encryption using one-way coupled-map lattices Encryption

**Performance Analysis:** Encrypted Biometric Image Correlation Coefficient

![Graph](image)

Figure 5.31 Encrypted Image Correlation Pixel Biometric Encryption Image (OCML Method)
**Algorithm:** Biometric Encryption using one-way coupled-map lattices Encryption

**Statistical Analysis:** Input Biometric Image histogram

Figure 5.32 Histogram of Input Biometric (OCML Method)
Algorithm: Biometric Encryption using one-way coupled-map lattices Encryption

Statistical Analysis: Biometric Image histogram

Figure 5.33 Histogram of Biometric Encryption Image (OCML Method)
Algorithm: Biometric Encryption using One-Way Coupled-Map Lattices Encryption

Security Analysis: BEOCML-Differential Attack Analysis

OCML Security Analysis

Table 5.4 Biometric Encryption using OCML method

<table>
<thead>
<tr>
<th>Input Biometric Size</th>
<th>Number of Pixel Change Rate(NPCR)</th>
<th>Unified Average Changing Intensity(UACI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
</tr>
<tr>
<td>32 × 32</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>64 × 64</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>128×128</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
**Algorithm:** Biometric Encryption using One-Way Coupled-Map Lattices Encryption

**Statistical Analysis:** Correlation Coefficients of Two Adjacent Pixels in Encrypted and Decrypted Images

Table 5.5 OCML Security Analysis

<table>
<thead>
<tr>
<th>Input</th>
<th>Correlation Coefficient of Input</th>
<th>Correlation Coefficient of Encrypted Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biometric Size</td>
<td>Biometrics</td>
<td></td>
</tr>
<tr>
<td>32 $\times$ 32</td>
<td>0.999901 0.997966 0.997964</td>
<td>-0.001402 0.003011 0.000711</td>
</tr>
<tr>
<td>64 $\times$ 64</td>
<td>0.999803 0.993890 0.923891</td>
<td>-0.000702 0.002099 0.013401</td>
</tr>
<tr>
<td>128$\times$128</td>
<td>0.901012 0.989102 0.929201</td>
<td>-0.006011 0.090111 0.000121</td>
</tr>
</tbody>
</table>
Algorithm: Biometric Encryption using Bakers Map

Performance Analysis: BEBM- Differential Attack Analysis

Table 5.6 Bakers Map Security Analysis

<table>
<thead>
<tr>
<th>Input Biometric Size</th>
<th>Number of Pixel Change Rate (NPCR)</th>
<th>Unified Intensity (UACI)</th>
<th>Average Changing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
<td>Blue</td>
</tr>
<tr>
<td>32 × 32</td>
<td>100 %</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>64 × 64</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>128 × 128</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
**Algorithm:** Biometric Encryption using Bakers Map

**Procedure:** Bakers Map Encryption and Decryption

![Biometric Input](image1)

![Biometric Encryption](image2)

![Biometric Decryption](image3)

Figure 5.34 (a) Biometric Input Image (b) Bakers Map Biometric Encryption (c) Biometric Decryption
**Algorithm:** Biometric Encryption using Bakers Map

**Statistical Analysis:** Input Biometric Image Histogram

Figure 5.35 Histogram of Biometric Input Image (Bakers Algorithm)
Algorithm: Biometric Encryption using Bakers Map

Statistical Analysis- Encrypted Biometric Image Histogram

![Histogram of Biometric Encrypted Image](image)

Figure 5.36 Histogram of Biometric Encrypted Image (Bakers Map)
Algorithm: Biometric Encryption using Bakers Map

Performance Analysis- Input Biometric Image Correlation Coefficient

Figure 5.37 Correlation Pixels of Biometric Image (Bakers Map)
Algorithm: Biometric Encryption using Bakers Map

Performance Analysis- Encrypted Biometric Image Correlation Coefficient

Figure 5.38 Correlation Pixels of Biometric Encryption Image (Bakers Map)
**Algorithm:** Biometric Encryption using Bakers Map

**Statistical Analysis:** Correlation Coefficients of Two Adjacent Pixels in Encrypted and Decrypted Images

Table 5.7 Bakers Map Security Analysis

<table>
<thead>
<tr>
<th>Input Biometric Size</th>
<th>Correlation Coefficient of Input Biometrics</th>
<th>Correlation Coefficient of Encrypted Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Algorithm: BEBM</td>
<td>Algorithm: BEBM</td>
</tr>
<tr>
<td></td>
<td>Horizontal                    Vertical          Diagonal          Horizontal          Vertical          Diagonal</td>
<td></td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.547083                      0.616432          0.565639          -0.0378787          0.00999202          -0.034978</td>
<td></td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.620350                      0.566605          0.652186          -0.0117266          -0.0076480          -0.016552</td>
<td></td>
</tr>
<tr>
<td>128 × 128</td>
<td>0.814783                      0.627028          0.635846          0.00989430          -0.0363194          0.0178498</td>
<td></td>
</tr>
</tbody>
</table>
**Algorithm:** Biometric Encryption Using Baker Map

**Performance Analysis:** BEBM- Timing Attack

Table 5.8 BEBM- Timing Attack Analysis

<table>
<thead>
<tr>
<th>Input Biometrics</th>
<th>Time Taken for Biometric Encryption (Seconds)</th>
<th>Time Taken for Biometric Decryption (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>BSBM</td>
<td>BSBM</td>
</tr>
<tr>
<td>32×32</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>64×64</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>128×128</td>
<td>0.00001</td>
<td>0.00002</td>
</tr>
<tr>
<td>256×256</td>
<td>0.0002</td>
<td>0.00025</td>
</tr>
<tr>
<td>512×512</td>
<td>0.12</td>
<td>0.20023</td>
</tr>
<tr>
<td>1024×1024</td>
<td>1.5341</td>
<td>1.849002</td>
</tr>
</tbody>
</table>
**Algorithm:** Biometric Encryption using Total Shuffling

**Procedure:** Biometric Encryption based on a New Total Shuffling, Lorenz chaotic System and Chen’s Chaotic System

Figure 5.39 (a) Input Biometric Image (b) Biometric Encryption using Total Shuffling (c) Lorenz and Chens Chaotic Biometric Encryption (d) Decryption of Biometrics (e) Deshuffled Biometrics
Algorithm: Biometric Encryption using Total Shuffling

Performance Analysis: Input Biometric Image Correlation Coefficient (IDL- PSYM value=4)

Figure 5.40 Correlation Horizontally Adjacent Pixels in Input Biometric Images (Total Shuffling Method)
**Algorithm**: Biometric Encryption using Total Shuffling

**Performance Analysis**: Encrypted Biometric Image Correlation Coefficient (IDL-PSYM Value=4)

![Image of correlation horizontally adjacent pixels in ciphered images](image)

Figure 5.41 Correlation Horizontally Adjacent Pixels in Ciphered Images (Total Shuffling Method)
Algorithm: Biometric Encryption using Total Shuffling

Performance Analysis: Input Biometric Image Correlation Coefficient (IDL- PSYM value=3)

Figure 5.42 Correlation Horizontally Adjacent Pixels in Input Biometric Images (Total Shuffling Method)
Algorithm: Biometric Encryption using Total Shuffling

Performance Analysis: Encrypted Biometric Image Correlation Coefficient (IDL-PSYM Value=3)

Figure 5.43 Correlation Horizontally Adjacent Pixels in Ciphered Images (Total Shuffling Method)
Algorithm: Biometric Encryption using Total Shuffling

Statistical Analysis: Correlation Coefficients of Two Adjacent Pixels in Encrypted and Decrypted Images

Table 5.9 Biometric Total Shuffling Method Security Analysis

<table>
<thead>
<tr>
<th>Input Biometric Size</th>
<th>Correlation Coefficient of Input Biometrics</th>
<th>Correlation Coefficient of Encrypted Image</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Algorithm: BSTS</td>
<td>Algorithm: BSTS</td>
</tr>
<tr>
<td></td>
<td>Horizontal  Vertical  Diagonal</td>
<td>Horizontal  Vertical  Diagonal</td>
</tr>
<tr>
<td>32 × 32</td>
<td>0.0896772  0.325609  0.217073</td>
<td>0.106876  0.0416935  0.00535391</td>
</tr>
<tr>
<td>64 × 64</td>
<td>0.393326  0.393326  0.393326</td>
<td>0.102733  0.0171105  0.015843</td>
</tr>
<tr>
<td>128 × 128</td>
<td>0.795865  0.610317  0.393973</td>
<td>0.0744475  0.00217384 -0.0128510</td>
</tr>
<tr>
<td>256 × 256</td>
<td>0.945015  0.875680  0.807739</td>
<td>0.127536 -0.0302647 -0.0623550</td>
</tr>
</tbody>
</table>
**Algorithm:** Biometric Encryption using Total Shuffling

**Performance Analysis:** Timing Attack Analysis

Table 5.10 Total Shuffling Timing Attack Analysis

<table>
<thead>
<tr>
<th>Input Biometrics</th>
<th>Time Taken for Biometric Encryption (Seconds)</th>
<th>Time Taken for Biometric Decryption (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>BSTS</td>
<td>BSTS</td>
</tr>
<tr>
<td>32×32</td>
<td>0.015000105</td>
<td>0.016000032</td>
</tr>
<tr>
<td>64×64</td>
<td>0.030999899</td>
<td>0.047000170</td>
</tr>
<tr>
<td>128×128</td>
<td>0.12500000</td>
<td>0.15600014</td>
</tr>
<tr>
<td>256×256</td>
<td>0.45300007</td>
<td>0.60899997</td>
</tr>
<tr>
<td>512×512</td>
<td>1.8280001</td>
<td>2.9370000</td>
</tr>
<tr>
<td>1024×1024</td>
<td>7.3440001</td>
<td>16.172000</td>
</tr>
</tbody>
</table>

5.7 **Security Analysis of Biometric Encryption using Nonlinear Algorithms**

With the properties of sensitivity to initial conditions and control parameters, pseudo-randomness and ergodicity, chaotic maps have been widely used in biometric encryption and Signal Encryption recently. Compared with traditional cryptosystems, the ones based on chaos are easier to be realized, which makes it more suitable for large-scale data encryption such as biometric encryption, images, videos or audio data.
A chaos-based Biometric Encryption scheme was recently proposed in and is widely referenced afterwards, but its security has not been analyzed efficiently. In General, encryption scheme is composed of two steps: Chaotic Confusion and pixel diffusion, where the former process permutes a image with 2-D chaotic map, and the later process changes the value of each pixel one by one. In the confusion process, the parameters of the chaotic map serve as the confusion key; in the diffusion process, such parameters as the initial value or control parameter of the diffusion function serve as the diffusion key. As shown in the following Figure 5.44, the confusion and diffusion process are both repeated for several times to enhance the security of this cryptosystem.

![Figure 5.44 Confusion and Diffusion System Architecture](image)

In the confusion process, many different 2D chaotic maps can be used, such as bakers map and cat map, which must be discretized over the biometrics lattice to realize the confusion pixels. For a $N \times N$ biometric lattice, cat map and baker map are as follows:
\[
\begin{bmatrix}
    x_{j+1} \\
    y_{j+1}
\end{bmatrix} = \begin{bmatrix}
    1 & u \\
    v & uv + 1
\end{bmatrix} \begin{bmatrix}
    x_j \\
    y_j
\end{bmatrix} \pmod{N}
\] (5-8)

\[
\begin{cases}
    x_{j+1} = \frac{N}{k_i} (x_j - N_i) + y_j \mod \frac{N}{k_i}, \\
    y_{j+1} = \frac{k_i}{N} (y_j - y_j \mod \frac{N}{k_i}) + N_i
\end{cases}
\text{With}
\begin{align*}
    k_1 + k_2 + \ldots + k_i &= N, \\
    N_i &= k_1 + \ldots + k_{i-1}, \\
    N_i \leq x_j < N_i + k_i, \\
    0 \leq y_j < N
\end{align*}
\] (5-9)

The key is composed of parameters \(u\) and \(v\) in Cat Map. In Baker Map, the key is \(K = [k_1, K_2, \ldots, K_i]\) that satisfies the condition.

2D Chaotic map is used to realize confusion process, and diffusion function is used to realize diffusion process. This, the cryptosystem can be regarded as a block cipher similar to traditional block ciphers based on confusion and diffusion operations. For convenience of analysis, the cryptosystem is defined as

\[
Y = [D(C(X, K_1), K_2)]^n,
\] (5-10)

Where \(X\) is Biometric Image, \(Y\) is Cipher Image, \(K_1\) is diffusion key, and \(n\) is iteration time. \(C()\) and \(D()\) mean confusion process and diffusion process respectively. As can be seen, the cryptosystems security is determined by the used chaotic map \(C()\), diffusion function \(D()\) and iteration time \(n\).
5.8 Key Space Security Analysis

The key space of the cryptosystem is the multiplication between the ones of the two processes. Supposing the one of confusion process is S1, and the one of diffusion process is S2, then the one of the cryptosystem is

\[ S = S_1 \cdot S_2 \] (5-11)

Here, the same key is used in different iterations. In practice, different keys can be used in different iterations. If n is the iteration time, and different keys are used in different iterations, then the key space is \( S = (S_1 \cdot S_2)^n \). According to Equation (5-8) and (5-9), the confusion space is determined by the parameter space of chaotic map, and the diffusion space is determined by the initial-value space of diffusion function. As can be seen, the cryptosystem’s key space S increases with the rise of parameter space S1, initial-value space S2, or iteration time n. Among them, initial-value space is determined by the gray level of image pixels, parameter space can be adjusted by selecting a suitable chaotic map, and the iteration time can be chosen according to security and complexity requirements. Taking N× N-sized image for example, if the gray-level is L, then the key spaces of the cryptosystems based on the above three chaotic maps are shown in Table 5.11. Seen from the table, for certain chaotic map, the key space is larger if different key is used in different iteration. For different chaotic maps, it is easy to verify that cat map has the smallest one, and the one of Baker map is in the middle. From this view, Baker map is preferred than cat map. Additionally, if the chaotic map is certain, then bigger iteration time is preferred.
### 5.9 Security and Complexity

The cryptosystem’s security is in relation with its computing complexity. And its computing complexity depends on iteration time \( n \), the computing complexity of 2D chaotic map and the one of diffusion function. Among them, the high cost caused by chaotic map or diffusion function can be decreased through selecting suitable one according to its computing complexity. Here, the computing complexity of the above two chaotic maps and two diffusion functions are given, which is shown in Table 5.12.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Baker Map</th>
<th>Cat Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition/Subtraction</td>
<td>( O(N^2) )</td>
<td>( O(N^2) )</td>
</tr>
<tr>
<td>Multiplication/Division</td>
<td>( O(N^2) )</td>
<td>( O(N^2) )</td>
</tr>
</tbody>
</table>
5.9.1 Security Improvement

According to the above analysis and experiments, this thesis employs the following advice to choose suitable chaotic map, diffusion function and iteration time n.

(a) Different chaotic map has different confusion property that is in relation with iteration time. Enough iteration time should be satisfied in order to keep high security against statistic attack or differential attack. For Example the iteration time for Cat map is no smaller than 6, and the one for Baker map should be no smaller than 12 or above.

(b) Baker map has the lowest computing complexity and middle parameter space, and Cat map has the smallest parameter space and the middle computing complexity. Thus, Baker map is preferred as a tradeoff between security and computing complexity.

(c) Based on Select-plaintext Attack, the pixel in corner (0, 0) cannot be permuted to other positions, which is a threat to the whole cryptosystem. In order to avoid it, a method can be adopted to change the position of the corner-pixel. That is, to change the scan order after each chaotic map, this changes the position of the first pixel. Thus, it is difficult to get the cipher text $Q_0^x$, which increases the difficulty of breaking the diffusion key.

(d) Based on complexity, the cryptosystem’s security contradicts with its computing complexity. In order to solve this contradiction, a method can be used to get a suitable tradeoff. That is, to divide n times into
groups and use different key in different group. In each group, the chaotic map is iterated with the same key for \( n_0 \) times. Thus, the computing complexity and key space are computed respectively as

\[
\begin{align*}
R(N) &= \frac{n}{n_0} R_c(N) + n \cdot R_d(N) \\
K(N) &= [K_c(N), K_d(N)]^{n'/n_0}
\end{align*}
\]  

(5-12)