Chaotic-Based Encryption Algorithm using Henon and Logistic Maps for Fingerprint Template Protection

Apri Siswanto[1,2], Norliza Katuk[2] and Ku Ruhana Ku-Mahamud[2]

[1]Informatics Department, Faculty of Engineering, Universitas Islam Riau, Indonesia
[2]School of Computing, College of Arts and Sciences, Universiti Utara Malaysia

Abstract: Fingerprint is a reliable user authentication method as it is unique to individual users that makes it efficient for authenticating users. In a fingerprint authentication system, user fingerprint information is stored in databases in an image format known as a fingerprint template. Although fingerprint is reliable, the templates stored in the database are exposed to security threats either during the data transmission process over the network or in storage. Therefore, there is a need to protect the fingerprint template, especially in unsecured networks to maintain data privacy and confidentiality. Many past studies proposed fingerprint template protection (FTP) using chaotic-based encryption algorithms that are more suitable to secure images than conventional encryption such as DES, AES, and RSA. The chaotic-based encryption algorithms have been improved a lot in terms of their robustness. However, the robustness of the algorithm caused a trade-off to encryption speed where it remains an issue in FTP. Hence, this study aims to improve the limitations found in the existing chaotic-based encryption algorithms for FTP by improving its encryption speed using Henon and Logistic map. A series of simulations were conducted using MATLAB to evaluate the performance of the proposed chaotic-based encryption algorithm for FTP through different analyses covering key sensitivity, histogram, correlations, differential, information entropy, and encryption/decryption speed. The performance proposed encryption algorithm was promising which could be a starting point for detailed analysis and implementation in real application domains.

Keywords: FTP, fingerprint, encryption, chaotic, cryptography, Henon map, logistic map.

1. Introduction

To date, fingerprints have been widely used as a method for user authentication [1] in various digital systems such as electronic government systems, access to office buildings, and smart home systems. Fingerprints are popular because they address issues like lost or stolen smart cards and forgotten or stolen users' passcode. Fingerprint authentication requires users to register the fingerprint data for the first time before the authentication process takes place. The fingerprint data are stored in a file or database in an image format known as the fingerprint template. Like other types of data such as text, sounds, or video [2], the fingerprint template is susceptible to security attacks, including forgery and server spoofing [3]. As most digital systems are connected through networks, data protection becomes one of the most critical aspects of network communication. Fingerprint template protection (FTP) is required to protect the fingerprint template in an unsecured communication network that prevents information leakage, a severe threat to users' privacy [4]. The basic idea of FTP is to conceal the fingerprint template by applying an encryption algorithm on the image so that it is protected from being copied or modified by attackers during data transmission of an authentication process.

FTP is achieved through encryption schemes that provide an effective protection covering the algorithm, key management, and perhaps the message authentication techniques. There are two critical issues to consider when designing an FTP scheme, i.e., encryption speed and robustness. In general, a complex encryption algorithm scrambles fingerprint data to produce high security. As a trade-off, the performance of the encryption algorithm is low when considering the encryption speed. On the other hand, a simple encryption algorithm will have faster encryption speed; however, it generally provides low-security protection. Most previous studies in the FTP scheme focused on the robustness of the encryption algorithms. However, recent trends have shown that studies are mostly directed towards achieving an ideal FTP scheme that meets low encryption speed but high-security protection [5].

A recent study by Murillo Escobar et al. [6] proposed an FTP scheme that encrypts fingerprint templates in grayscale ".bmp" format using chaotic map encryption. The FTP scheme reduced the complexity of the encryption algorithm in encrypting fingerprint templates and achieve 0.234 seconds of encryption speed, while the decryption process takes 0.210 seconds. The good thing is that the scheme is still able to maintain the robustness of the finger data protection. Based on the performance of the Murillo Escobar et al.'s [6, 7] scheme, there is an opportunity to reduce the encryption speed by enhancing the encryption algorithm using Henon maps combined with a logistic map. We hypothesize that the use of the Henon map could reduce the encryption speed whereas keeping the security of the FTP scheme at the highest level. The remaining sections are divided into four. Section 2 elaborates on the related studies, then it is followed by the proposed of the chaotic-based encryption algorithm in Section 3. Later, the results of the performance analysis of the proposed encryption algorithm are eloborated, and finally, the the obejective of the study is revisited in Section 5.

2. Related Studies

Analysis of the literature suggested a considerable number of studies have been conducted on FTP schemes whereby they proposed encryption algorithms suitable for encrypting fingerprint images. The plethora of studies is mainly conducted to address the limitations in the conventional encryption algorithms like AES and RSA, that do not work efficiently on images as they have enormous capacity, high-level of redundancy, and close relationships of the nearby pixels [8, 9]. Chaotic-based encryption algorithms have been one of the possible solutions for protecting images and addressing the limitations of traditional encryption techniques [10]-[11]. Al-Romema, et al. [12] illustrated the typical process in encrypting images using chaotic-based encryption, as shown in Figure 1. The plain image will undergo n and m rounds of confusion and diffusion processes, respectively, using a secret key generated by a key generator to create a ciphered image.

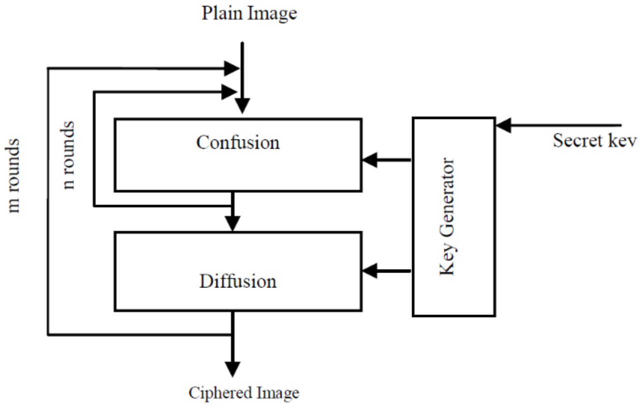


Figure 1. A typical process of chaotic-based encryption [12]

Li, et al. [13] demonstrated evident in which the original chaotic-based encryption algorithm applied to text data were exposed to four types of attack, namely chosen-plaintext, chosen-ciphertext, ciphertext-only, and known-plaintext attacks. Then, they proposed an improvement version of the encryption method by strengthening security protection against the four attacks. However, the proposed solution retained the problem in encryption speed unsolved as the original. Ever since, researchers attempted to improve the limitations in the original version of the chaotic-based encryption in achieving a robust algorithm with lower encryption speed, an ideal encryption algorithm for FTP.

Murillo-Escobar et al. [7] presented a chaotic-based encryption algorithm where it is secure from statistical attacks, although the permutation stage is removed. Abundiz-Pérez, et al. [6] used hyperchaotic Rosslermap to produce a secure encryption algorithm and obtain an acceptable encryption speed. Cui [14] proposed a chaotic-based encryption algorithm using a fractional Fourier transform that is robust again cryptanalysis. Another study by Liu [15] proposed an FTP scheme using chaotic maps that is resilient against brute-force attacks. Bhatnagar and Wu [16] also proposed an FTP scheme with a chaotic map, Heisenberg decomposition, and fractional wavelet packet transform, and the schemes were proven robust using edge similarity, information entropy, histogram, correlation, and keyspace analyses.

Hsiao and Lee [17] investigated FTP schemes with multiple chaotic encryption algorithms. This scheme consists of two one (1)-dimension and two three (3)-dimension chaotic systems. The results of their study suggested that this scheme could protect fingerprint databases from unauthorized intrusive attacks. Further, Mirzaei, et al. [18] proposed an FTP scheme using a parallel encryption algorithm with a chaotic system that has large keyspace, robust, and low processing complexity. Similarly, Azzaz, et al. [19] proposed an FTP scheme with chaotic encryption technique implemented with the System on Programmable Chip (SoPC) that is secure against frequent security attacks. All the FTP schemes mentioned above aimed at increasing the robustness of the chaotic-based encryption; unfortunately, the encryption speed of the scheme remains slow, which has been the trade-off to the increased security. Therefore, the study presented in this paper intends to address the limitations of the existing chaotic-based encryption algorithms by utilizing a more suitable method to reduce the encryption speed while able to offer a similar degree of security protection**.**

3. The Proposed Encryption Algorithm for FTP

The content of this section covers the proposed chaotic-based encryption algorithm that embeds Henon and logistic maps. Henon map is a two (2)-dimensional mapping for representing a chaotic-behavior [20, 21], a more straightforward form of Lorenz' system [22]. The Henon map can be written as in equation (1):

$$\begin{aligned} \boldsymbol{x_{i+1}} &= \boldsymbol{1 - a x_i^2 + y_i} \\ \boldsymbol{y_{1+1}} &= \boldsymbol{b x_n} \end{aligned} \tag{1}$$

Where x_i and y_i is original coordinates, x_{i+1} and y_{i+1} new coordinates after transformation, a and b is Henon map parameter values. Henon map could produce random numbers for an encryption key [23]. In the case of image encryption, Henon map can be used to randomize pixel positions. Henon map has been used to randomize pixel position [8, 24, 25]. This study will use 1-dimensional Henon map that has been modified from the 2-dimensional map. It yields a new equation as rendered in equation (2).

$$\boldsymbol{x_{i+1} = 1 - a x_{i+1}^2 + b x_i} \tag{2}$$

The canonical Henon equation has a value of a =1.4 and b=0.3. After the pixel position is randomized with the Henon map, the pixel value is encrypted with the confusion technique whose value is obtained through the logistic map function. For the initial logistical map value X_0 = 0.1, and r = 3.99 was used in this study to produce the key, K. The simple structured of a 1-D logistic map is stated in equation (3).

$$\boldsymbol{x_{i+1} = a x_i (1 - x_i)} \tag{3}$$

Table 1 lists the symbols and notations that represent the process in the proposed algorithm for chaotic-based encryption described in this study.

In the proposed chaotic-based encryption algorithm, a plain fingerprint image (P) will be converted into a ciphered fingerprint image (P') through a transformation function that is characterized by a set of user-specific parameters, which come from random external keys, K. The external keys are randomly generated by the Henon map function. After the confusion process, the fingerprint image (I') is obtained. Then (I') will undergo diffusion process using the K parameter generated from the logistic map, so that the fingerprint template becomes a protected image (P'). T is calculated by calculating the sum of fingerprint plain template elements using Henon and logistic maps. The Henon and logistic maps are iterated 1000 and 3000 times respectively. Henon and logistic maps present the non-uniform distribution because they generate many values that close to 0 and 1 that can change the original form of the images. In countermeasure, we amplify X_1^H *and* X_1^L 1000 times as in equations 5 and 8. The iterations will make the encryption algorithm more robust from chosen-plaintext and known-plaintext attacks.

Table 1. The notations of the proposed encryption algorithm

Symbol	Description
CO_n	Confusion
RE_m	Round encryption M pixels
x_{10}	Initial condition logistic map
x_{20}	Initial condition Henon map
x^H	Henon map chaotic sequence
x^L	Logistic map chaotic sequence
α_1	Logistic map control parameter
α_2	Henon map control parameter
A	A control parameter for Part 1
B	A control parameter for Part 2
C	A control parameter for Part 3
D	A control parameter for Part 4
F	Constant' initialized in zero'
I'	Image P that has been undergone confusion process
K	External key
M	M Pixels (256)
N	N Pixels (256)
P	Plain fingerprint image
P'	Ciphered fingerprint image
R	Iteration
S	Constant' initiated in zero'
Su	Sum
T	v2 / 255
V1	mod((S*1000) + F,1
V2	1 + round(v1*253)
$ArrI$	Array element plain image (M)
$ArrJ$	Array element plain image (N)
E	Encryption
Mg	Optimized diffusion process

Figure 2 illustrates the proposed chaotic-based encryption using Henon and logistic maps.

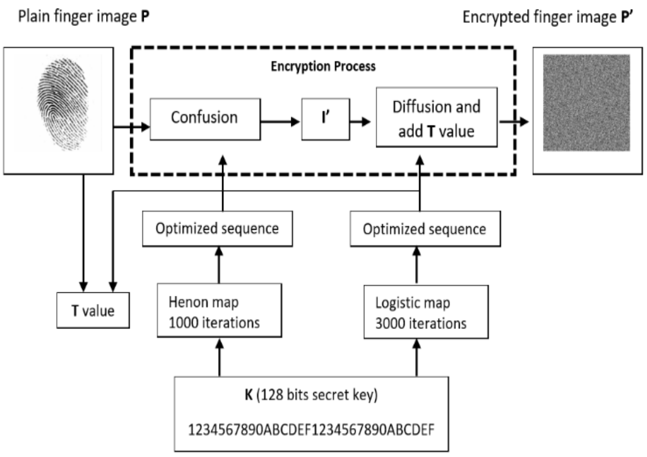

Plain finger image P
Encrypted finger image P'
Encryption Process
Confusion
I'
Diffusion and add T value
Optimized sequence
Optimized sequence
T value
Henon map 1000 iterations
Logistic map 3000 iterations
K (128 bits secret key)
1234567890ABCDEF1234567890ABCDEF

Figure 2. The proposed chaotic-based encryption process using Henon and logistic maps

3.1 Secret Key Forming Process

The key size K is generated based on Murillo-Escobar et al.'s [7] algorithm that is 128 bits secret key in 32 hexadecimal digits (0-9; A-F). Table 2 describes the four parts of the secret key, K that are used to produce the initial conditions and control parameters of the two chaotic maps. The key will produce a chaotic sequence to avoid the chaotic range of stops, small space keys, and periodicity in the chaotic range for the algorithm.

Table 2. Handling of the secret key

Secret Key	Control Parameter	Initial Condition
32 HEX digits	$H_1, H_2, \dots, H_{32}$ $where\ H\ \in \|0-9, A-F\|$	
Calculation	$A = \frac{(H_1, H_2, \dots, H_8)_{10}}{2^{32}+1}$ $C = \frac{(H_{17}, H_{18}, \dots, H_{24})_{10}}{2^{32}+1}$	$\mathrm{B} = \frac{(H_9, H_{10}, \dots, H_{16})_{10}}{2^{32}+1}$ $D = \frac{(H_{25}, H_{26}, \dots, H_{32})_{10}}{2^{32}+1}$
Logistic Map	$\alpha_1 = 3.999 + \left[\left((A+B+T)mod1\right) * 0.001\right]$	$x_{10} = (C+D+Z)\ mod\ 1$
Henon Map	$\alpha_2 = 1.4 + \left[\left((A+B)mod1\right) * 0.3\right]$	$x_{20} = (C+D)\ mod\ 1$

3.2 Calculation of T Value

The T value has specialized roles in which it is used to increase (1) sensitivity at the plaintext level and secret key, and (2) security from chosen and differential attacks. T is calculated by summing plain template with chaotic data from the Henon map. The Henon process is iterated R = 1000 speeds using α_2 and x_{20}. Then a chaotic sequence $x^H = x_1^H, x_2^H, x_3^H, \dots, x_i^H\ with\ x^H\ \in (0,1)\ and\ 10^{-15}$ are generated. Subsequently, an optimized chaotic sequence x^H by expression $x_1^H = \{[x_1^H] * 1000\} mod\ 1$, where I = 1,2, 3,1000. Plain fingerprint image is changed from P ∈ [0,255] to P € (0,1) with 10^{-15} decimal precision and the following operation is performed in equation (4):

$$\boldsymbol{Sum = S + P(i,j)} \tag{4}$$

where $\mathrm{i} = 1,2,3, \dots, \mathrm{M}; \mathrm{j} = 1,2,3, \dots; \mathrm{N};$ and S is a constant initiated in zero. To increase the sensitivity of plain fingerprint images, S is amplified at 1,000 speeds. Next, it will sum the last 96 chaotic data form $\mathrm{x_h}$ as stated in equation (5)

$$\boldsymbol{F = F + X^H_{R-t}} \tag{5}$$

t=0,1,2,3...95, F is constant initialized in zero. The next operation is determined by v1 = mod((S*1000) + F,1); for range values 0-1, and v2 = 1 + round(v1*253); to range values 1-254. Finally, T is derived by t = v2 / 255.

3.3 Encryption Process

At the encryption stage, logistic map performs iteration R of confusion and diffusion processes. The chaotic sequence $x^L = \{x_1^L, x_2^L, x_3^L, \dots, x_I^L\}$ is calculated using R, R=3000 iterations, the control parameter α_1, and initial condition x_{10} from Table 1. It has a decimal precision of 10^{-15} and $x^L \in (0,1)$. For the confusion process, a sub-sequence x^L is calculated according to M row of plain fingerprint image P as stated in equation (6).

$$\boldsymbol{RE_m = round\left[X^L_{(I-M+m)} \cdot (M-1)\right] + 1} \tag{6}$$

where m=1,2,3,…,M, RE ∈ [1,M] is a length vector M; a multiplication of each chaotic value $\mathrm{X_L}$ per $(\mathrm{M}-1)$. Then the other sequence is calculated from $\mathrm{X_L}$ according to N. It is calculated using equation (7).

$$\boldsymbol{CO_n = round\left[(X^L_{I-N+m}).\,(N-1)\right] + 1} \qquad (7)$$

where n=1,2,...,N, CO ∈ [1,N] is length vector N, round is round operation and '.' represents multiplication' of each chaotic value X_L per (N-1). The third sub-sequence of 3,000 values is calculated for the optimized diffusion process as stated in equation (8).

$$\boldsymbol{Mg = \left\{\left[x^L_g * 1000\right] + T\right\}(mod)\ for\ g = 1, 2, 3, \dots, 3000} \qquad (8)$$

where M ∈ (0,1) with 10^{-15} decimal precision. Lastly, the plain fingerprint' image' is transformed from P ∈ [0,255] to P ∈ (0,1) with 10^{-15} decimal precision. The confusion-diffusion process is calculated with the formula stated in equation (9).

$$\boldsymbol{E(i,j) = \left[\left(RE_i, CO_j\right) + (M_g)\right] \quad (mod\ 1)} \qquad (9)$$

where i=1,2,3,...M, j=1,2,3,...,N, g=1,2,3,..., MxN (mod 3.000) , E is ciphered fingerprint and P is plain fingerprint image. The ciphered fingerprint is transformed from E ∈(0,1) to E ∈ [0,255] with MxN size. Next, T value is added to the cryptogram, T hides within pseudo-random pixels using equation (10).

$$\mathbf{ArrI = round\left\{\left[X^L_{(R-10)} x(M-1)\right] + 1\right\},} \qquad (10)$$
$$\mathbf{ArrJ = round\left\{\left[X^L_{(R-100)} x(N-1)\right] + 1\right\}}$$

where I ∈ [1, M], J ∈ [1, N]]. From here, T is included in the encrypted image as E(i,j,k)=V2. The complete process of the proposed encryption algorithm is shown in Algorithm 1.

Algorithm 1: Chaotic-based Encryption Algorithm using Henon and Logistic Maps

```
Inputs
 P=(P1,..,Pn): Plain fingerprint image
 Key : '1234567890ABCDEF1234567890ABCDEF'
  Manage key separated by four section A = Key(1:8);  B =
 Key(9:16); C = Key(17:24); D = Key(25:32);
 Determine the value to be added in the A, B, C, and D
 Output P' = (P'1,….,P'm): Encrypted fingerprint image
 Iteration with henon map
for n=1:I  hen(n+1)=(1-(n)) + hen(n);
 Sum of plain image pixels with 96chaotic data henon map
 for n=1:96 F = F + hen (1001-n)
        T Calculation
        Iteration with logistic map
for n=1: T log(n+1)=a1*log(n)*(1-log1(n));
   Sub-sequence for row confusion: RE
   Sub-sequence for column diffusion: CO
   Sub-sequence for diffusion: M
Encryption process: a round of confusion-diffusion: E
Construct M x N array
Compute E = RE, CO + Matrix
```

3.4 Decryption Process

The decrypting the ciphered fingerprint image, the algorithm performs the encryption process in a reverse order, with the same secret key, the value of T. To recover a ciphered fingerprint image, equation (11) was used.

$$\boldsymbol{D\left(RE_i, CO_j, k\right) = \left[E(i,j) - (M_g)\right] \quad (mod\ 1)} \qquad (11)$$

where i=1,2,3,...,M; j=1,2,3,...,N; g=1,2,3,..., MxN (mod 3000), D ∈ (0,1) is recover image fingerprint and E ∈ (0,1) is encrypted fingerprint. The decrypted fingerprint images are transformed from D ∈ (0,1) to D ∈ [0,255].

4. Evaluation and Results

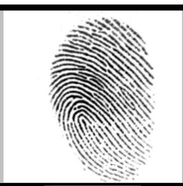
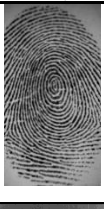
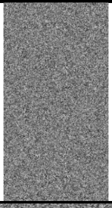
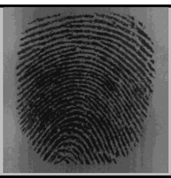
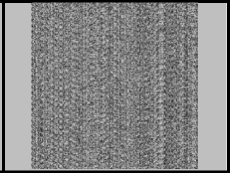
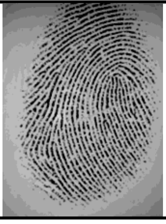
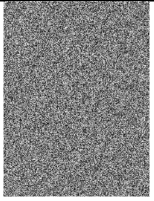
4.1 Tools and Data Set

The proposed chaotic-based encryption algorithm was programmed in MATLAB. The software was installed on a computer laptop runs Intel (R) Core (TM) i7, with 2GHz RAM, and Windows 10 operating system. Analyses on the key sensitivity, histogram, correlation, differential, and information entropy were conducted in evaluating the effects of the proposed chaotic-based encryption algorithm. This analysis, similarly almost the same as the analysis perform in Alsaedi paper [26]. FVC2002 [27] was used as the source of the fingerprint image dataset. It consists of 4 databases, namely DB1, DB2, DB3, and DB4, with 80 fingerprint images in each database. The image size in DB1 is 388x374 pixels, and 500 dpi resolution while the image size of DB2 is 296x560 pixels, with 569 dpi image resolution. On the other hand, the DB3 has the image size of 300x300 pixels with 500 dpi image resolution, while the DB4 has the image size of 288x384 pixels with 500 dpi image resolution.

4.2 The Form of Ciphered Fingerprint Images

The proposed chaotic-based encryption with Henon and logistic maps managed to encrypt the fingerprint images properly. Table 3 shows the plain fingerprint images in the second column and the ciphered fingerprint in the third columns, respectively, for one fingerprint images taken from each FVC2020 database. The ciphered (encrypted) fingerprint images are not recognized as it only shows the grayscale dotted images.

Table 3. The fingerprint ciphered images

FVC2002 Databases	**Plain fingerprint images**	**Ciphered fingerprint images**
DB1(101_1.tif)		
DB2(101_1.tif)		
DB3(101_1.tif)		
DB4(101_1.tif)		

4.3 Key Sensitivity Analysis

Every cryptographic system is vulnerable to brute-force attack including the key used by the encryption algorithms. A small keyspace that is less than 2^{56} is considered vulnerable to a brute-force attack. Munir [28] suggested that adequate security can be achieved by using a keyspace that is greater than 2^{100}. The keyspace of the secret key in the proposed chaotic-based encryption algorithm consists of 32 hexadecimal digits (128 bits), which leads to 2^{128} possible combinations of a secret key, which is considered robust against brute-force attack.

An ideal encryption algorithm must be sensitive to small changes in the secret key during the encryption and decryption process. In the decryption process, if a plain fingerprint image is encrypted twice with two similar keys, the ciphered fingerprint images must be completely different. A plain fingerprint image (i.e., "103_5.tif") from DB1 was encrypted using three different keys to verify the sensitivity of the secret keys. Then, a correlation analysis was conducted on the cryptograms of the three secret keys. Table 4 shows the results of the key sensitivity analysis in which the correlation that is close to 0 indicates that the images are entirely different from others.

Table 4. Key sensitivity analysis

Secret Key	Diagonal Correlation	Vertical Correlation	Horizontal Correlation
1234567890ABCDEF 1234567890ABCDEF	-0.00517	-0.00247	0.00701
12345678**8**0ABCDEF 1234567890ABCDEF	0.00679	-0.01202	-0.00410
1234567890ABCDEF 12345678**8**0ABCDEF	0.00783	-0.00645	-0.00868

4.4 Histogram Analysis

In image analysis, a histogram shows the distribution of the intensity of pixels image. In carrying out attacks with statistical analysis techniques, attackers usually analyze a histogram to get the occurrence of pixel intensity that could be used to deduce keys or pixels in a plain image. In other words, a histogram of ciphered images should not have statistical similarities with its plain image histogram [29]. Therefore, the pixels in the ciphered image should have a distribution that is (relatively) uniform in which it is indicated by a flat histogram. A series of histogram analyses were conducted to evaluate the performance of the proposed chaotic-based encryption algorithm. A plain fingerprint image (i.e., "101_1.tif") from all databases were selected to demonstrate the histogram analyses. The results of the histogram analyses are presented in Figure 3. The left column of Figure 3 left column shows the histogram for the plain fingerprint images, while the right column represents the histogram for the corresponding ciphered fingerprint image. In the plain fingerprint image of DB1, the histogram shows much stacking on the right side because the image contains intense values close to 255 (white). The peak point in the histogram shows intense pixels, while lower point shows the less intense pixels. On the other hand, the ciphered fingerprint image for DB1 looks uniform and flatter after the encryption process. For DB2, DB3, and DB4, the intense pixels of the images are closer to 0, and they represented normal fingerprint images. The ciphered images of the three databases also look flat and uniform, which indicates that plain images and encrypted are different after the encryption process. Besides, the cipher image type is png due to providing a stable balance between lossy compression, small file size, and suitable for file image, which is transmitted on the network.

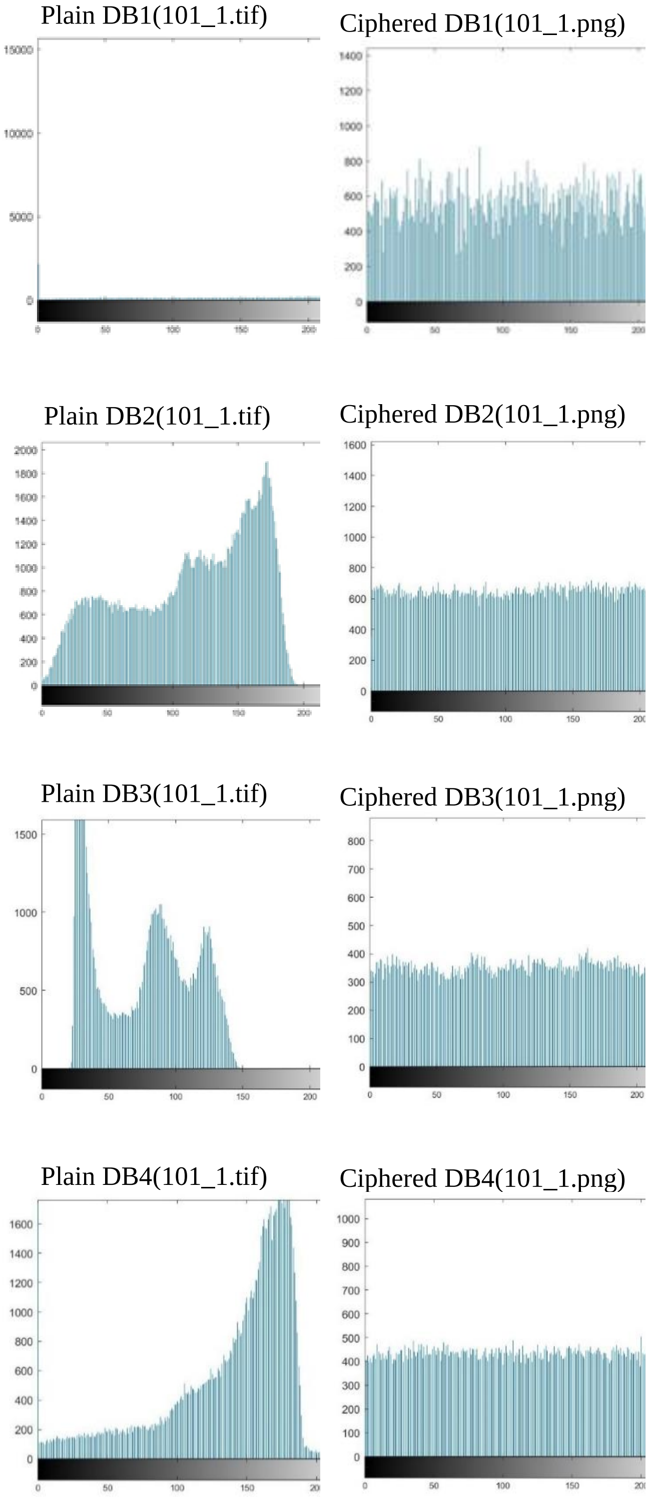

Plain DB1(101_1.tif)
Ciphered DB1(101_1.png)
Plain DB2(101_1.tif)
Ciphered DB2(101_1.png)
Plain DB3(101_1.tif)
Ciphered DB3(101_1.png)
Plain DB4(101_1.tif)
Ciphered DB4(101_1.png)

Figure 3. Histograms for the plain and ciphered fingerprint images

The generated histograms for the ciphered fingerprint images for DB1, DB2, DB3, and DB4 are significantly different from the plain fingerprint images. The flat peaks in the histogram of the ciphered fingerprint images make it difficult for the malicious parties to perform a frequency analysis of the appearance of pixel values because they do not have relationships of the pixels. Therefore, the proposed chaotic-based encryption algorithm can protect the fingerprint images from histogram attacks.

4.5 Correlation Analysis

High correlation coefficient (close to +1 or -1) is usually found in adjacent pixels in a plain fingerprint image. The ciphered fingerprint images should have a low correlation or the correlation coefficient close to zero [28]. An analysis was conducted to find out the correlation of pixels in the plain and ciphered fingerprint images. A plain fingerprint image (i.e., "101_1.tif") from all databases were selected to demonstrate the histogram analyses. The correlation coefficient between two adjacent pixels vertically, horizontally, and diagonally is calculated in this analysis. The results of the analysis are summarized in Table 5. The correlation analysis is calculated using equations (12-15).

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \, , E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad (12)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, D(y) = \frac{1}{N}\sum_{i=1}^{N} (y_i - E(y))^2 \qquad (13)$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \qquad (14)$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\ X\ \sqrt{D(y)}} \qquad (15)$$

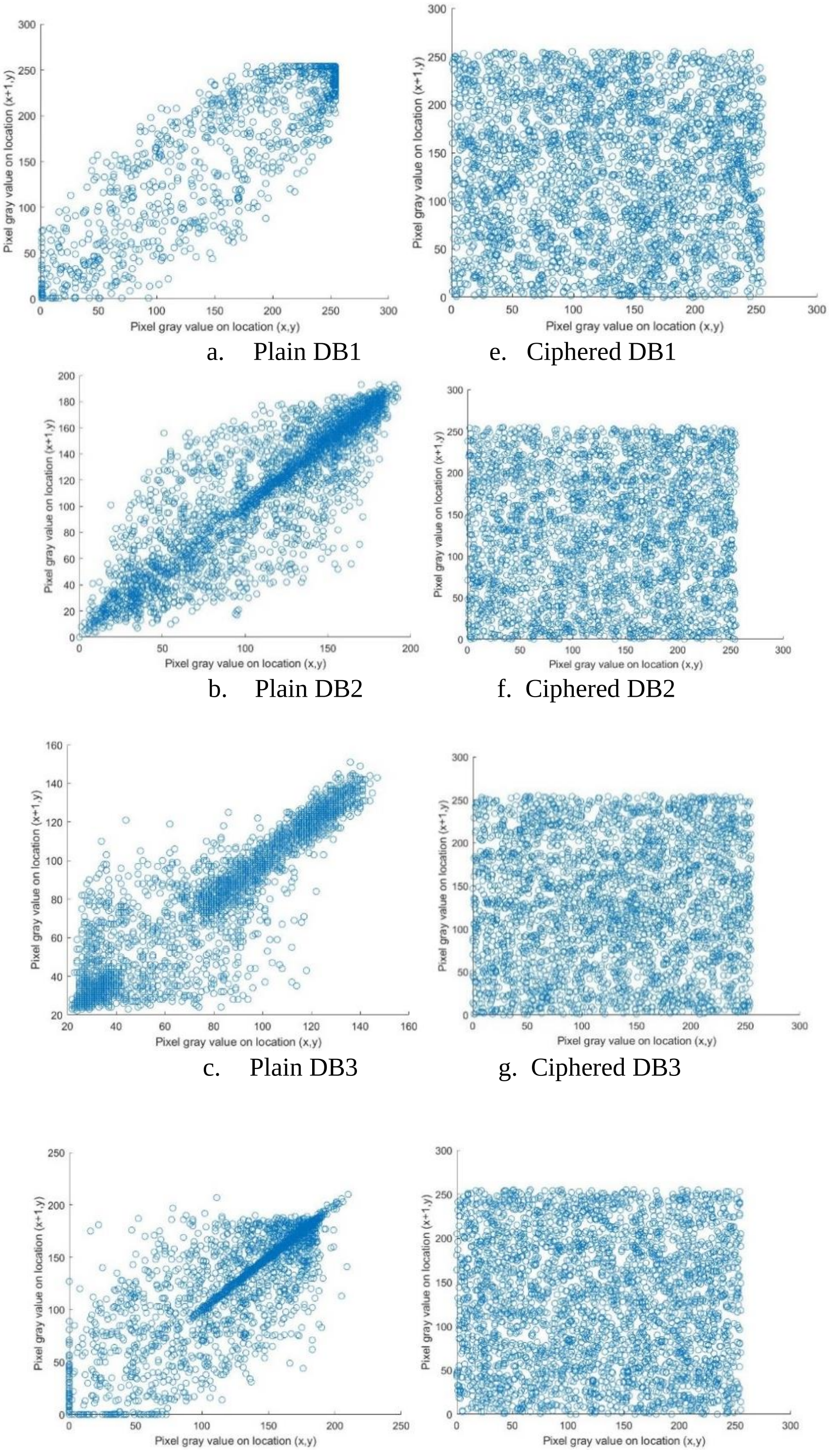
where x and y are the grey values of two adjacent pixels in the image, cov(x, y) is the covariance, D is the variance, E is mean, and r is the correlation coefficient.

Table 5. Correlation coefficient analysis

Correlation coefficient	**Diagonal**	**Vertical**	**Horizontal**
Plain DB1(101_1.tif)	0.9223	0.9102	0.9367
Ciphered DB1	-0.0017	0.0093	0.0023
Plain DB2(101_1.tif)	0.8032	0.9262	0.8703
Ciphered DB2	0.0011	-0.0069	0.0017
Plain DB3(101_1.tif)	0.8716	0.9034	0.9235
Ciphered DB3	-0.0030	0.0020	0.0072
Plain DB4(101_1.tif)	0.8237	0.9023	0.8802
Ciphered DB4	0.0056	-0.0046	-0.0021

Table 5 shows the correlation distribution of selected DB1(101_1.tif), DB2(101_1.tif), DB3(101_1.tif), and DB4(101_1.tif) images, where each one-pixel value and adjacent diagonal, vertical and horizontal pixel values are the same (high correlation). On the other hand, the ciphered fingerprint images have the values between pixels, and adjacent ones are very different, suggesting a low correlation. The positive and negative value provides information about the orientation of the relationship of the two variables.

Positive values in two variables suggested that they have a same orientation of relationship in which an increase in one variable will cause an increase in the other variable and the other way around. On the other hand, negative values suggested that the opposite correlation between the two variables, where an increase in the value of one variable will be accompanied by a decrease in the other variable. Figure 4 illustrates the correlation distribution of the plain fingerprint images (left side) and ciphered fingerprint images (right side). The plain and ciphered fingerprint images are distinct. The images on the left side show that the linear pattern is uphill (bubbles) suggested a strong (high) correlation, whereas the pixels of ciphered images are spread over a wider band, which indicates that the encryption process makes the pixels of ciphered images almost independent of each other (low correlation). This evaluation proves that the proposed encryption algorithm conceals the attributes of the plain fingerprint images, while the cipher fingerprint images are entirely random and highly uncorrelated [30].



a. Plain DB1 e. Ciphered DB1

b. Plain DB2 f. Ciphered DB2

c. Plain DB3 g. Ciphered DB3

d. Plain DB4 h. Ciphered DB4

Figure 4. (a)-(d) Correlation distribution of fingerprint image for each dataset; and (e)-(h) Correlation distribution of cipher fingerprint image

4.6 Differential Analysis

This analysis was conducted to test the effects of changing each pixel on a ciphered fingerprint image. There are two indicators commonly used in this analysis, namely the number of pixels change rate (NPCR) and unified average changing intensity (UACI). NPCR calculates the percentage of the number of pixels that changes of the ciphered images as compared to the plain fingerprint images. While the UACI is the percentage of the average intensity of differences between plain image and encrypted image [31]. The NPCR and UACI calculation are defined in equations (16-18), respectively. The ideal values of NPCR and UACI are 99.6094% and 33.4635%, respectively.

$$c(i,j) = \begin{cases} 0 if T_1(i,j) \neq T_2(i,j) \\ 1 if T_1(i,j) \neq T_2(i,j) \end{cases} \tag{16}$$

$$NPCR = \frac{\sum_i^M \ \sum_j^N \ C(i,j)}{MXN} X\ 100\% \tag{17}$$

$$UACI = \frac{\sum_i^M \sum_j^N |T_1(i,j) - T_2\ (i,j)|}{255\ x\ M\ x\ N} x\ 100\% \tag{18}$$

Where M and N are the height and width of the image and $T_1(i,j)$ and $T_2(i,j)$ are the pixel value in location (I,j) of encrypted images. A comparison was made between the proposed encryption algorithm with two other algorithms reported in Murillo-Escobar, et al. [7] and Zhang et al.[32]. The results of the defferential analysis suggested that the proposed algorithm is highly sensitive on plain fingerprint images, further, it is also secure against differential attacks. Considering the results summarized in Table 6, the proposed encryption algorithm outperformed Murillo-Escobar, et al. [7] and Zhang et al. [32] for NPCR; however, it is almost at par for UACI. Although these three encryption algorithms are robust from differential attacks and are very sensitive to plain images.

Table 6. NPCR and UACI results of the proposed algorithm and two similar algorithms.

Test	Proposed FTP Scheme	Murillo-Escobar, et al. [7]	Zhang, et al. [32]
NPCR	99.63%	99.50%	99.60%
UACI	33.41%	33.36%	33.40%

4.7 Information Entropy Analysis

Message entropy (T) is calculated by the equation (19) [25]:

$$\boldsymbol{H(T) = \sum_{i=0}^{2^N-1} p(Ti) log2 \left(\frac{1}{p(Ti)}\right)} \tag{19}$$

Where H*(T)* is information entropy, $\boldsymbol{Ti}$ is the probability of $\boldsymbol{Ti}$; N is the number of bits that is required to represent the symbol $\boldsymbol{Ti}$, and log 2 represent the base 2 logarithm.

A randomize image data will have the entropy value close to 8, whereas in less randomize image data will have the value less than eight. When the entropy value is less than 8; the image data is predictable which increase the risk of attacks [25, 33]. In the case of fingerprint image encryption, the results of the ciphered fingerprint image are a random image, so the entropy should be ideal close to 8. Because there are 256 degrees of greyness in the image (T0 = 0, T1 = 1, ..., T255 = 255) and each grey degree is recorded (calculated from the histogram). Table 7 shows the entropy values of the ciphered images of (i.e., "101_1.tif") taken from DB1, DB2, DB3, and DB4, respectively.

Table 7. Entropy value of the ciphered images

Ciphered Image	Proposed FTP scheme	Bhatnagar[16]	Hsiao[17]
DB1 (101_1CI)	7.9729	7.9947	7.905972
DB2 (101_1CI)	7.9970	7.9954	-
DB3 (101_1CI)	7.9951	7.9955	-
DB4 (101_1CI)	7.9962	7.9947	-

Using the proposed algorithm, the entropy values for the images are approaching 8 which suggested that it is secured from entropy attacks. The performance of the proposed algorithm is comparable with the other two studies by Bhatnagar [16] and Hsiao [17] .

4.8 Analysis of encryption and decryption speeds

An analysis was also made on time taken by the proposed encryption algorithm to perform the encryption and decryption process. The analysis was made on a fingerprint image (101_1.tif) of DB1, DB2, DB3, and DB4, respectively. The size of DB1 101_1.tif is 388 × 374 pixels (145,624 bytes), DB2 101_1.tif is 296 x 560 pixels (166,272 bytes), DB3 101_1.tif is 300 x 300 pixels (90,512 bytes) and DB4 101_1.tif is 288 x 384 pixels (111,104 bytes). Whereas the size of cipher image DB1 101.png, DB2 101.png, DB3 101.png, and DB4 101.png is 142,079 bytes, 166,697 bytes bytes, 90,544 bytes and 111,249 bytes, respectively. The proposed chaotic-based encryption algorithm managed to encrypt the fingerprint images in 0.144148 seconds of decrypting it back to the plain image in 0.120648 seconds for the image taken from DB1. Table 8 shows the encryption and decryption speeds of images from DB2, DB3, and DB4.

Table 8. Encryption and decryption speed

Datasheet	Encryption speed (seconds)	Decryption speed (seconds)
DB1 101_1.tif	0.108987	0.085487
DB2 101_1.tif	0.117915	0.094415
DB3 101_1.tif	0.092196	0.068696
DB4 101_1.tif	0.112331	0.088831

5. Conclusions

This a study proposed a chaotic-based encryption algorithm by applying Henon and logistic maps to improve encryption and decryption speeds. Besides, the chaotic-based encryption process is designed according to the characteristics of plain fingerprint images and the 128-bit key to withstanding active attacks of a plain known attack. Security analysis through a series of simulations run in MATLAB suggested the effectiveness of the proposed encryption algorithm. However, the proposed encryption algorithm requires further improvements and evaluations as the analyses conducted in this study were limited to selected dataset. Nevertheless, the results could be used as a starting point to a further and advanced study that seeks to develop an encryption algorithm and scheme for fingerprint authentication in real application domains such as the smart home.

6. Acknowledgement

The research is funded by the Ministry of Education, Malaysia under the Trans-Disciplinary Research Grant Scheme (Ref: TRGS/2/2014/ UUM/01/3/4, UUM S/O Code: 13164).

References

[1] J. B. Kho, J. Kim, I.-J. Kim, and A. B. Teoh, "Cancelable fingerprint template design with randomized non-negative least squares," Pattern Recognition, vol. 91, pp. 245-260, 2019.

[2] U. Sivarajah, M. M. Kamal, Z. Irani, and V. Weerakkody, "Critical analysis of Big Data challenges and analytical methods," Journal of Business Research, vol. 70, pp. 263-286, 2017.

[3] M. K. Khan and J. Zhang, "Improving the security of 'a flexible biometrics remote user authentication scheme'," Computer Standards & Interfaces, vol. 29, pp. 82-85, 2007.

[4] X. Lu, Q. Li, Z. Qu, and P. Hui, "Privacy information security classification study in internet of things," in 2014 International Conference on Identification, Information and Knowledge in the Internet of Things, 2014, pp. 162-165.

[5] S. Askar, A. Karawia, and A. Alshamrani, "Image encryption algorithm based on chaotic economic model," Mathematical Problems in Engineering, vol. 2015, 2015.

[6] F. Abundiz-Pérez, C. Cruz-Hernández, M. Murillo-Escobar, R. López-Gutiérrez, and A. Arellano-Delgado, "A Fingerprint Image Encryption Scheme Based on Hyperchaotic Rössler Map," Mathematical Problems in Engineering, vol. 2016, p. 15, 2016.

[7] M. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, and R. López-Gutiérrez, "A robust embedded biometric authentication system based on fingerprint and chaotic encryption," Expert Systems with Applications, vol. 42, pp. 8198-8211, 2015.

[8] K. Zhou, M. Xu, J. Luo, H. Fan, and M. Li, "Cryptanalyzing an image encryption based on a modified Henon map using hybrid chaotic shift transform," Digital Signal Processing, vol. 93, pp. 115-127, 2019.

[9] H. Gao, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption," Chaos, Solitons & Fractals, vol. 29, pp. 393-399, 2006.

[10] S. Subashanthini, A. K. Cherukuri, and M. Pounambal, "Image Encryption Using New Chaotic Map Algorithm," in Advances in Intelligent Systems and Computing vol. 941, ed, 2020, pp. 458-466.

[11] C. Liang, Q. Zhang, J. Ma, and K. Li, "Research on neural network chaotic encryption algorithm in wireless network security communication," EURASIP Journal on Wireless Communications and Networking, vol. 2019, p. 151, 2019.

[12] N. A. Al-Romema, A. S. Mashat, and I. AlBidewi, "New chaos-based image encryption scheme for RGB components of color image," Computer Science and Engineering, vol. 2, pp. 77-85, 2012.

[13] S. Li, X. Mou, and Y. Cai, "Improving security of a chaotic encryption approach," Physics Letters A, vol. 290, pp. 127-133, 2001.

[14] D. Cui, "A novel fingerprint encryption algorithm based on chaotic system and fractional Fourier transform," in Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on, 2010, pp. 168-171.

[15] R. Liu, "Chaos-based fingerprint images encryption using symmetric cryptography," in Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, 2012, pp. 2153-2156.

[16] G. Bhatnagar and Q. J. Wu, "Enhancing the transmission security of biometric images using chaotic encryption," Multimedia systems, vol. 20, pp. 203-214, 2014.

[17] H.-I. Hsiao and J. Lee, "Fingerprint image cryptography based on multiple chaotic systems," Signal Processing, vol. 113, pp. 169-181, 2015.

[18] O. Mirzaei, M. Yaghoobi, and H. Irani, "A new image encryption method: parallel sub-image encryption with hyper chaos," Nonlinear Dynamics, vol. 67, pp. 557-566, 2012.

[19] M. S. Azzaz, N. Aissaoui, and C. Tanougast, "A Novel Fingerprint Protection Approach based on SoPC Chaotic Encryption," in 2018 International Symposium on Networks, Computers and Communications (ISNCC), 2018, pp. 1-6.

[20] M. Hénon, "A two-dimensional mapping with a strange attractor," in The Theory of Chaotic Attractors, ed: Springer, 1976, pp. 94-102.

[21] H. Wen, "A review of the Hénon map and its physical interpretations," School of Physics Georgia Institute of Technology, Atlanta, GA, pp. 30332-0430, 2014.

[22] P. Ping, Y. Mao, X. Lv, F. Xu, and G. Xu, "An image scrambling algorithm using discrete Henon map," in 2015 IEEE International Conference on Information and Automation, 2015, pp. 429-432.

[23] B. F. Vajargah and R. Asghari, "A pseudo random number generator based on chaotic henon map (CHCG)," International Journal of Mechatronics, Electrical and Computer Technology (IJMEC), vol. 5, pp. 2026-37, 2015.

[24] S. Sheela, K. Suresh, and D. Tandur, "Image encryption based on modified Henon map using hybrid chaotic shift transform," Multimedia Tools and Applications, vol. 77, pp. 25223-25251, 2018.

[25] A. Jolfaei and A. Mirghadri, "An image encryption approach using chaos and stream cipher," Journal of Theoretical and Applied Information Technology, vol. 19, pp. 117-125, 2010.

[26] M. Alsaedi, "Image Encryption and Decryption Using Chua's Circuit," International Journal of Communication Networks and Information Security, vol. 11, pp. 112-118, 2019.

[27] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," in Object recognition supported by user interaction for service robots, 2002, pp. 811-814.

[28] R. Munir, "Security analysis of selective image encryption algorithm based on chaos and CBC-like mode," in Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on, 2012, pp. 142-146.

[29] W. Wieclawek, "Information granules in image histogram analysis," Computerized Medical Imaging and Graphics, vol. 65, pp. 129-141, 2018.

[30] I. F. Elashry, O. S. F. Allah, A. M. Abbas, S. El-Rabaie, and F. E. A. El-Samie, "Homomorphic image encryption," Journal of Electronic Imaging, vol. 18, p. 033002, 2009.

[31] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications, vol. 1, pp. 31-38, 2011.

[32] W. Zhang, K.-w. Wong, H. Yu, and Z.-l. Zhu, "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," Communications in Nonlinear Science and Numerical Simulation, vol. 18, pp. 2066-2080, 2013.

[33] G. Ye, C. Pan, X. Huang, Z. Zhao, and J. He, "A chaotic image encryption algorithm based on information entropy," International Journal of Bifurcation and Chaos, vol. 28, p. 1850010, 2018.