Protecting Large Size Medical Images with Logistic Map Using Dynamic Parameters and Key Image

M. Y. Mohamed Parvees[1], J. Abdul Samath[2], and B. Parameswaran Bose[3]

(Corresponding author: M.Y. Mohamed Parvees)

Research and Development Centre, Bharathiar University[1]
Maruthamalai Rd., Coimbatore-641046, India
(Email: yparvees@gmail.com)
Department of Computer Science, Government Arts College, Elayamuthur Rd., Udumalpet-642126, India[2]
No. 35, 1st Main, 3rd Cross, Indiragandhi St., Udaynagar, Bangalore-560016, Karnataka, India[3]

(Received July 22, 2016; revised and accepted Nov. 15 & Dec. 25, 2016)

Abstract

This paper presents a faster and efficient algorithm to encrypt large size 24-bit color medical images. The algorithm generates permutation sequences which shuffles the image pixels and color bytes. The generated masking sequences alters the value of the pixels and color bytes. The dynamic chaotic parameters initialisation is one of the key factors in the proposed cryptosystem. Further, the key image enhances efficiency of the encryption algorithm and the whole cryptosystem becomes complex by executing XOR operation with the shuffled pixels and bytes. The proposed algorithm yields final cipher images of the test medical images that have good confusion - diffusion properties and validated through series of tests to ensure the high security level.

Keywords: Chaotic Map; Immunohistochemical Images; Logistic Map; Medical Image Encryption

1 Introduction

Telemedicine and E-health are becoming more popular at times as there is a necessity for quick and secure diagnosis of diseases by transmitting and storing information among the experts. Since the internet is open to everyone, data protection is most important thing to avoid unauthorized access. The rapid growth of internet and E-techniques allows transmitting large files such as image, video and audio files. The medical experts, educationists and particularly insurance companies need methods to protect the patient's medical information [9, 10, 20]. The patient's information should be accessed only by the authorized personnel during the transmission and storage.

The microscopic colour images such as immunohistochemical (IHC) images have been transmitted among the experts for confirming the status of patient's disease. Further, the experts or doctors always need a second opinion to confirm the absolute status of the disease. Similarly, the health insurance companies wanted to know the real information of the patients by transmitting the images to the experts and getting opinion from them [2, 24, 32, 33]. The main idea is to protect the large medical images (colour) during the transmission and storage using chaotic maps. Since the telepathology and high-content whole slide imaging (WSI) techniques (virtual slides) are growing rapidly [5], the large size IHC images are considered for the analysis of proposed algorithm.

Chaos theory has elucidated considerable interest in computer science for several years because of its applications in the field of cryptography where the traditional algorithms DES, 3DES, RSA are lacking to encrypt bulky data efficiently [7, 18]. Many cryptographic protocols are designed in the literature and chaos theory was proposed in 1989 [17]. Chaos has lot of important qualities *i.e.* high reactive to initial conditions, aperiodicity and topological transitivity [30]. Further, the chaos theory can be use in symmetric key encryption due to their better computational performance over public key encryption [14]. Chaos theory involves in encryption based on the two processes, one is 'confusion' and another one is 'diffusion'. The chaotic maps and matrix manipulations are useful in generating confusion and diffusion processes [11]. In the chaotic image encryption, some of the algorithms with one round of shuffling and diffusion are also reported [6]. But for a cipher image with good confusion and diffusion property, the processing round should be more than three rounds [1]. So far, several researchers have been proposed different chaotic models to encrypt images using single choatic map[22], combinations of one or two maps[21], higher order maps [25] and even the chaotic

maps are combined with other techniques [19] to provide higher level of security. In this study, the chaotic image protection scheme is intended to encrypt immensely colossal size medical images using a Logistic map with dynamic parameters. The large size immunohistochemical images have been analyzed towards the proposed encryption algorithm.

2 The Mathematical Background of Logistic Map

The Logistic map is a simple one dimensional map and efficient in studying nonlinear dynamic systems which can able to reveal chaos behaviour. It is defined as follows:

$$x_{n+1} = r \times x_n \times (1 - x_n), x_n \in (0, 1), r \in (0, 4) \qquad (1)$$

where, x_n is an independent variable; r is the control parameter of logistic map; $n = 1, 2, 3 \ldots$. When the value of the control parameter lies between 3.5699456 and 4 that is, $(3.5699456 < r \leq 4)$ and $x_n \in (0, 1)$, this logistic map is chaotic which can able to produce n length sequences.

A bifurcation diagram represents the possible period orbits of a chaotic map based on the bifurcation parameter values. For a diminutive change in the bifurcation parameter values the possible periodic orbits of the logistic map represented by Equation (1) will be more complicated. From the bifurcation diagram the X-axis represents the bifurcation parameter and Y-axis represents the possible population values x of the logistic map function. As the control parameter value of the Logistic map increases, the bifurcation occurs. From the Figure 1 it is possible to say that many bifurcations occurred for the value of r lies between 3.6 to 4.0. The positive Lyapunov number of the Logistic map influences the non periodic orbits significantly which is shown in Figure 2.

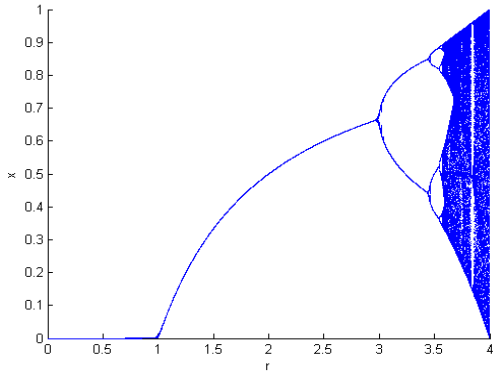

Figure 1: Bifurcation diagram of logistic map

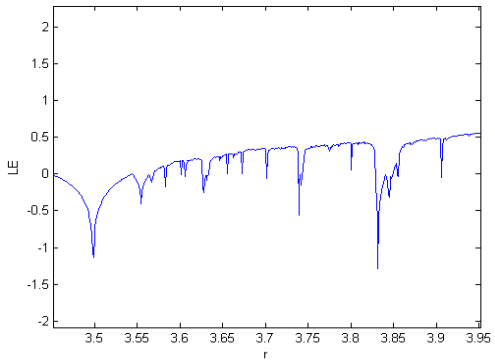

Figure 2: Lyapunov exponent diagram of logistic map

3 The Proposed Cryptosystem

3.1 Dynamic Control and Initial Parameters

In the proposed image encryption algorithm, the eight different dynamic control and initial parameters helps to generate chaotic sequences, thereby scrambles the image pixels efficiently. The eight dynamic control and initial parameters are generated from the two master dynamic control and initial parameters.

The master parameter $r_1 = 3.612345678901234$ and $x_1 = 0.112345678901234$ are used to generate a chaotic sequence $O = \{o_1, o_2, o_3, .o_{10000}\}$ to get dynamic control parameters by using Equation (1). Similarly, the master parameter $r_2 = 3.712345678901234$ and $x_2 = 0.212345678901234$ are used to generate a chaotic sequence $I = \{i_1, i_2, i_3, \ldots, i_{10000}\}$ to get dynamic initial parameters by using Equation (1).

The sequence element (o_{n*s}) is found and checked for sequence element $(o_{n*s}) \geq 0.6$, if true, $dr_n = 3.0 + o_{n*s}$ else moved to o_{n*s+1} until the sequence element value is ≥ 0.6 and added with 3.0 to get dr_n. The sequence element i_{n*s} is found and assigned as $dx_n = i_{n*s}$. Since the number of parameters are eight, the technique produces 8 different control $(dr_1, dr_2, dr_3, dr_4, dr_5, dr_6, dr_7, dr_8)$ and initial $(dx_1, dx_2, dx_3, dx_4, dx_5, dx_6, dx_7, dx_8)$ parameters. The 8 dynamic chaotic parameters are used to produce the permutation and masking sequences which are mainly used for image encryption. By varying the values of $S(diff), r_1, x_1, r_2, x_2$, it is possible to generate entirely different control and initial parameters set. The pseudocode to generate the dynamic control and initial parameters are shown in Algorithm 1.

3.2 Chaotic Sequence

A chaotic map produces non-converging and non-periodic sequences while assigning values for the initial and control parameters. These sequences are termed as chaotic sequences. By using the real valued chaotic sequences, it is able to produce an integer valued sequence. This se-

Algorithm 1 Generation of dynamic chaotic parameters

BEGIN
Intitialise the chaos parameters
SET $no\_params \leftarrow 8$
SET $diff \leftarrow 888$
SET $seq\_no \leftarrow 100000$
SET $c\_miu \leftarrow 3.612345678901234$
SET $c\_pre \leftarrow 0.112345678901234$
SET $i\_miu \leftarrow 3.712345678901234$
SET $i\_pre \leftarrow 0.212345678901234$
Generate control sequences using Equation (1)
Method: genSeqArray(seq_no, c_pre, c_miu)
INPUT: chaos parameters seq_no, c_pre, c_miu.
OUTPUT: Array of control sequences(c_seq).
$c\_seq[0] \leftarrow c\_pre$
for $i \leftarrow 1$ to seq_no **do**
  $c\_seq[i] = c\_miu \times c\_seq[i-1] \times (1 - c\_seq[i-1])$
  RETURN c_seq
end for
Generate initial sequences using Equation (1)
inisequences $\leftarrow$ genSeqArray(seq_no, i_pre, i_miu)
Method: generateDynamicParams()
for $i \leftarrow 1$ to no_params **do**
  $control \leftarrow c\_seq[(i+1) \times diff]$
  $c\_count \leftarrow 1$
  while $control < 0.6$ **do**
    $control \leftarrow c\_seq[((i+1) \times diff) + c\_count]$
    $c\_count + 1$
  end while
  $cparams.dynamiccontrol \leftarrow control + 3.0$
  $controlparameterlist \leftarrow dynamiccontrol$
  $dynamicinitial \leftarrow i\_seq[(i+1) \times diff]$
  $cparams.setpreviouselement \leftarrow dynamicinitial$
  $cparamslist \leftarrow cparams$
end for
RETURN cparamslist.
END

quence is called permutation sequence and it can be used for image encryption and decryption. The sample chaotic sequences and the sorted chaotic sequences for the given values are clearly illustrated in Figures 3(a) - (d).

3.3 Generation of Permutation Sequence by Linear Search

The chaotic sequences are real valued sequences, where as the permutation sequences are integer valued sequences. Generating permutation sequence is the vital part of the proposed algorithm. This integer valued permutation sequence is used to shuffle the pixels and color bytes. A chaotic sequence is generated using Equation (1). The first 1000 chaotic elements are discarded to avoid the transient effect. Then, the chaotic sequence $C = \{c_1, c_2, c_3, \ldots, c_n\}$ is chosen from the 1001st chaotic element and sorted in ascending order to get $S = \{s_1, s_2, s_3, \ldots s_n\}$.

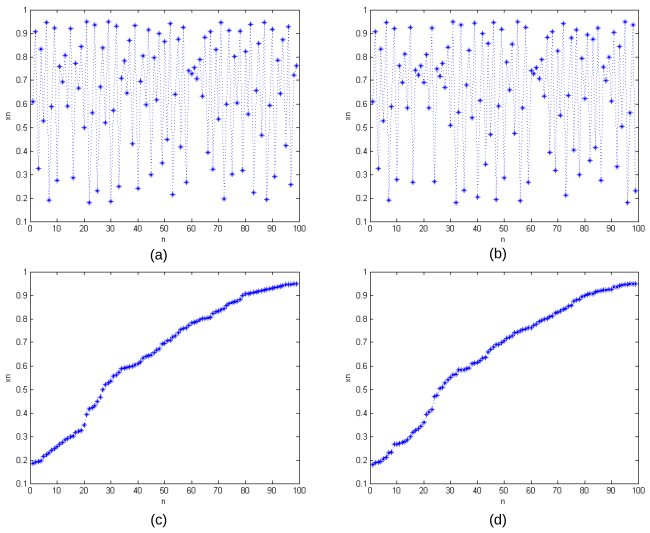

Figure 3: The sample chaotic sequences (a) x_n for $r = 3.80001$, $x_0 = 0.80001$, (b)x_n for $r = 3.80002, x_0 = 0.80002$ and the sorted chaotic sequences (c) x_n for $r = 3.80001, x_0 = 0.80001$, (d) x_n for $r = 3.80002, x_0 = 0.80002$

Further, iterated through S to find the index position of each element in C to get $P = p_1, p_2, p_3, \ldots, p_n$. The indexing an element in an array can be done through linear search. But, the linear search is time consuming. The pseudocode to create permutation sequence using linear search is shown in Algorithm 2 and the sample permutation sequences for the given values are shown in Figures 4(a)-(b).

Algorithm 2 Permutation sequence generation by linear search

BEGIN
$C_{\{c_1,c_2,\ldots,c_n\}} \leftarrow genSeqArray(seq\_no, pre, miu)$
$S_{\{s_1,s_2,\ldots,s_n\}} \leftarrow sortSeqArray(C_{\{c_1,c_2,\ldots,c_n\}})$
$P_{\{p_1,p_2,\ldots,p_n\}} \leftarrow linearSearch(S_{\{s_1,s_2,\ldots,s_n\}}, c(i))$
END

3.4 Generation of Permutation Sequence by Binary Search

The indexing of an array element is done using the binary search which is very faster while comparing to linear search. The S has been iterated to find the index position of each element in C to get $P = p_1, p_2, p_3, \ldots, p_n$ using binary search. The pseudocode for generating permutation sequence using binary search is given in Algorithm 3. The sample permutation sequences for the given values are shown in Figures 4(c)-(d).

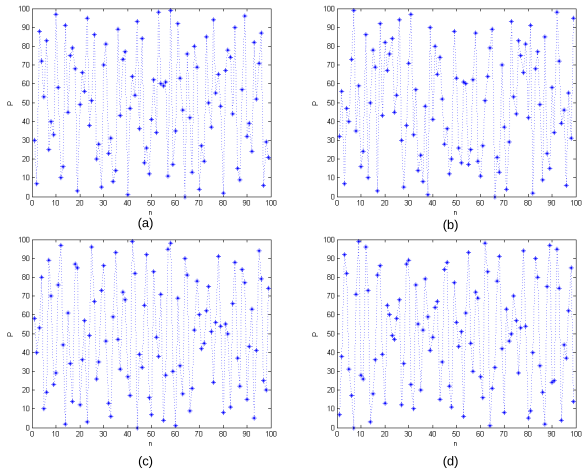

Figure 4: The permutation sequences for the given values (a) P for $r = 3.80001, x_0 = 0.80001$, (b) P for $r = 3.80002, x_0 = 0.80002$ and the permutation sequence using binary search for the given values (c) P for $r = 3.80001, x_0 = 0.80001$, (d) P for $r = 3.80002, x_0 = 0.80002$

Algorithm 3 Permutation sequence generation by binary search

BEGIN
$C_{\{c_1,c_2,...,c_n\}} \leftarrow genSeqArray(seq\_no, pre, miu)$
$S_{\{s_1,s_2,...,s_n\}} \leftarrow sortSeqArray(C_{\{c_1,c_2,...,c_n\}})$
$A_{\{a_1,...,a_n\}} \leftarrow C_{\{c_1,c_n,c_2,c_{n-1},...\}} \leftarrow cSeAr(C_{\{c_1,...,c_n\}})$
$P_{\{p_1,p_2,...,p_n\}} \leftarrow binarySearch(S_{\{s_1,s_2,...,s_n\}}, a(i))$
END

Algorithm 4 Masking sequence generation for pixel shuffling

BEGIN
$C_{\{c_1,c_2,...,c_n\}} \leftarrow genSeqArray(seq\_no, pre, miu)$
$S_{\{s_1,s_2,...,s_n\}} \leftarrow sortSeqArray(C_{\{c_1,c_2,...,c_n\}})$
$n_{max} \leftarrow S_{\{s_1,s_2,...,s_n\}}[S_{\{s_1,s_2,...,s_n\}}.length - 1]$
for $i \leftarrow 0$ to $C_{\{c_1,c_2,...,c_n\}}.length$ **do**
  $curr\_value \leftarrow Math.abs(C[i]/n_{max}) \times (2^{24} - 1)$
  $M[i] \leftarrow curr\_value$
end for
RETURN $M_{\{m_1,m_2,m_3...,m_n\}}$
END

3.5 Comparison of Time Complexity for Permutation Sequence Generation

In this section, the time complexity to generate the permutation sequence by linear search as well as by binary search method with various pixel lengths is investigated. The time taken to generate the permutation sequence are tabulated in the Table 1. The investigation clearly shows that the time taken to generate permutation sequence by binary search method is very little when compared to linear search method.

Algorithm 5 Masking sequence generation for colour byte shuffling

BEGIN
$C_{\{c_1,c_2,...,c_n\}} \leftarrow genSeqArray(seq\_no, pre, miu)$
$S_{\{s_1,s_2,...,s_n\}} \leftarrow sortSeqArray(C_{\{c_1,c_2,...,c_n\}})$
$n_{max} \leftarrow S_{\{s_1,s_2,...,s_n\}}[S_{\{s_1,s_2,...,s_n\}}.length - 1]$
for $i \leftarrow 0$ to $C_{\{c_1,c_2,...,c_n\}}.length$ **do**
  $curr\_value \leftarrow Math.abs(C[i]/n_{max}) \times 255$
  $M[i] \leftarrow curr\_value$
end for
RETURN $M_{\{m_1,m_2,m_3...,m_n\}}$
END

3.6 Proposed Algorithm for Image Encryption

The whole cryptosystem comprises of dynamic key generation, generation of permutation and diffusion sequences, confusion-diffusion operations and XOR operation of source image pixels with key image pixels. All these operation together makes the proposed system more secure. The dynamic control and initial parameters are generated which make the algorithm more secured. Thereby, the key become more secure and difficult to guess. The permutation sequences are generated using dynamically controlled logistic map. During the permutation sequence generation, the sequence element is indexed using binary search which makes the algorithm to behave faster. The colour byte scrambling and pixel scrambling are done iteratively with respect to different permutation sequences. Similarly, the masking sequences are generated to alters the values of pixels as well as colour bytes. The confusion and diffusion operations are done iteratively to makes the algorithm more complex.

The test image pixels are read and divided into small chunks of length n and the chunks are stored into a container map CM with index. The total number of pixels of test medial image are 9000000. But, the permutation and diffusion sequences are generated for 100000. It is not necessary to generate 9000000 sequences which takes more time. Further, the chunk size is 100000. So, the time taken for sequence generation is very less. The permutation sequences P_1, P_3 generated where length $length = n$ and P_2 & P_4 where $length = n \times 3$ to scramble image pixels' and colour bytes' positions respectively. Similarly the masking sequences M_1 & M_3 are generated to alter the 24-bit pixel values and colour bytes where $m_i = Int\left[(|(m_i/max\,(m_i))|) \times (2^{24} - 1)\right]$ and M_2 & M_4 where $m_i = Int\left[(|(m_i/max\,(m_i))|) \times 255\right]$. The pseudocode for generating masking sequence is given in Algorithms 4 and 5.

Then, the one dimensional array of pixels P is retrieved from container map. The pixels P is shuffled using permutation sequence P_1 and a bitwise XOR operation is done between P, M_1 and K $(P \oplus M_1 \oplus K)$. The K is the pixels of key image. The size of the key image 1000×1000 which consists random pixel and byte values. Then, the

Table 1: Time taken to generate the permutation sequence for various pixel lengths

$Width \times Height$	No. of pixels	Linear Search (sec)	Binary Search (sec)
256×256	65536	7.4	0.132
512×512	262144	117.29	0.596
1024×1024	1048576	1894.009	3.246
1920×1080	2073600	1921.561	5.815
3000×3000	9000000	38542.067	38.205

color bytes C, Y is separated from pixels P, K respectively. Now C is shuffled using permutation sequence P_2.

Then, the bitwise XOR operation is done between C, M_2 and Y. Similarly, the sequences P_3, P_4, M_3 and M_4 are employed for pixel confusion, colour byte confusion, pixel diffusion and colour byte diffusion respectively. The bitwise XOR operation between shuffled image with key image enhance the randomness of the cipher image. The alogorithm becomes efficient due to the number of rounds of scrambling done. The decryption is the reverse process of encryption. The pseudocode for the proposed algorithm is given in Algorithm 6.

The block diagram of the proposed scheme is given in Figure 5.

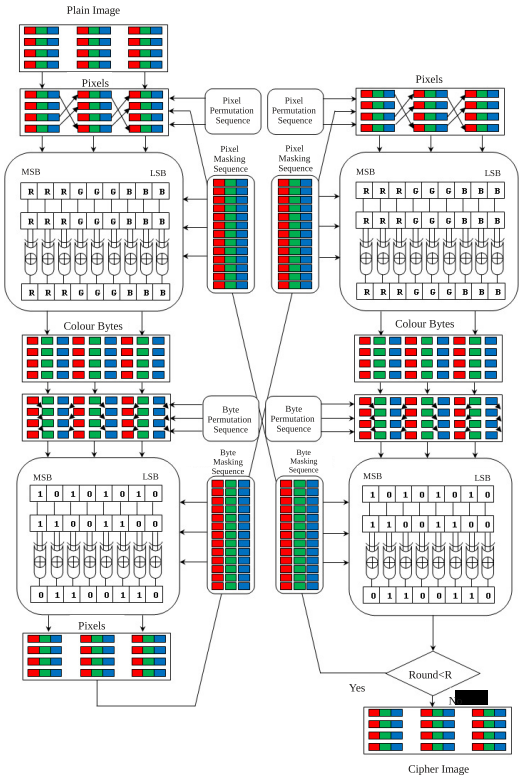

Figure 5: The block diagram of proposed algorithm

4 Result and Discussion

The proposed algorithm is experimented using a system with the Intel Core i5-3230M processor with a speed of 2.60 GHz, and the RAM capacity is 4GB. The algorithm has been coded using Java JDK 1.7.0 (64 bit). To analyze the proposed algorithm, seven large size and 24-bit colour images with size 3000 3000 are considered and they are downloaded from The Human Protein Atlas [23, 27].

The experimental results are given to illustrate the efficiency of the proposed cryptosystem with high resolution images. The selected initial and control parameters are $r_1 = 3.612345678901234, x_1 = 0.112345678901234, r_2 = 3.712345678901234, x_2 = 0.212345678901234$. According to the proposed algorithm, the values of $l = 3000 \times 3000$, $n = 100000$, and $R = 4$. The experimental plain, cipher and key images are given in Figure 6.

4.1 Security Analysis

The systematic security analyses have been done on proposed encryption scheme to verify that the system is much protective against the most frequent attacks. The key space and sensitivity analysis, statistical analysis, histogram analysis, mean variance color byte, information entropy analysis have been done to prove that the system withstands on different attacks.

4.1.1 Key Space Analysis

The sensitivity of the proposed cryptosystem is around 10^{-16} based on the control parameter and the sensitivity is around 10^{-15} based on the initial condition. The cryptosystem is designed in such a way that the encryption of colour components rely on each other. The keys of colour component are independent from each other. The key space is calculated by the control and the initial parameters, $K = (10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16}) \times (10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15}) = 10^{248}$. The key space is much higher while comparing with other literatures [3, 35, 26] as given in Table 2. This proves that the system defy against different brute-force attacks.

4.1.2 Differential Attack Analysis

The original image can be modified slightly (e.g., change a single pixel value) and encrypted to study the sensitiv-

Algorithm 6 Proposed algorithm for image encryption

BEGIN
READ image
SET $W \leftarrow$ image width
SET $H \leftarrow$ image height
CALCULATE $l \leftarrow W \times H$
SET I as one dimensional array of image pixels
READ key image
SET K as one dimensional array of key image pixels
GENERATE indexed Container Map(CM)contains small chunks of length n from pixels I of length l
GENERATE Permutation sequences P_1 and P_3 where $length = n$ and P_2 and P_4 where $length = n \times 3$
GENERATE Masking sequences $M_1\&M_3$ where masking sequence array $(m_i) \leftarrow [max(m_i) \times (2^{24} - 1)]$ and $M_2\&M_4$ where masking sequence array $m_i = [max(m_i) \times 255]\{m_i$-sequence Array[i], $max(m_i)$−maximum Sorted Sequence Array Value$\}$
READ one dimensional array P from Container map
for $i \leftarrow 0$ to 4 **do**
  for $i \leftarrow 0$ to length of the permutation sequence array index P_1 **do**
    Encrypted pixels[i] ← Pixel array[sequence Array Index [i]]
  end for
  COMPUTE Encrypted pixels P $\leftarrow P \oplus M_1 \oplus K$
  GENERATE colour bytes $C\&Y$ from pixels $P\&K$
  for $i \leftarrow 0$ to length of the permutation sequence array index P_2 **do**
    Encrypted bytes[i] $(C) \leftarrow$ byte array[sequence Array Index [i]]
  end for
  COMPUTE Encrypted Colour bytes $\leftarrow C \oplus M_2 \oplus Y$
  GENERATE pixels $P\&K$ from the color bytes $C\&Y$
  for $i \leftarrow 0$ to length of the permutation sequence array index P_3 **do**
    Encrypted pixels[i] ← Pixel array[sequence Array Index [i]]
  end for
  COMPUTE Encrypted pixels$\leftarrow P \oplus M_3 \oplus K$
  GENERATE colour bytes $C\&Y$ from pixels $P\&K$
  for $i \leftarrow 0$ to length of the permutation sequence array index P_4 **do**
    Encrypted bytes[i] $(C) \leftarrow$ byte array[sequence Array Index [i]]
  end for
  COMPUTE Encrypted Colour bytes $(C) \leftarrow C \oplus M_4 \oplus Y$
  GENERATE pixels $(P) \leftarrow$ colour bytes (C)
end for
COMPUTE resultant pixels I (Cipher image) ← chunks of pixels assembled in container map (CM) of length l.
END

Table 2: The key space value comparisons

Encryption Algorithm	Key Space
Proposed cryptosystem	$2^{720} \approx 10^{248}$
Ref. [26]	$2^{298} \approx 10^{90}$
Ref. [3]	2^{292}
Ref. [35]	2^{400}

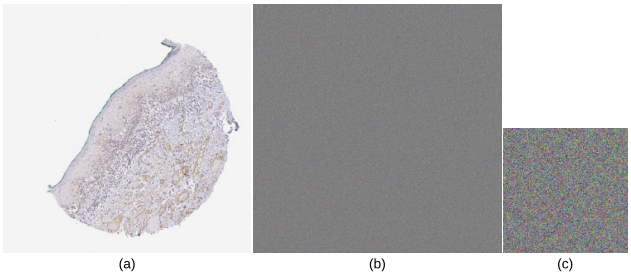

Figure 6: (a) The original plain image (3000 × 3000), (b) cipher image (3000×3000) and (c) key image (1000×1000)

ity of the encryption algorithm. If a slight change in the original image makes a significant change in the cipher image, then the algorithm is efficient and can withstands on any differential attacks. The differential analysis has been carried out by calculating the NPCR (Net Pixel Change Rate) and UACI (Unified Average Changing Intensity) using Equations (2) and (3).

$$NPCR = \frac{1}{W \times H} \sum_{p=1}^{W} \sum_{q=1}^{H} D(p,q) \times 100\% \quad (2)$$

$$UACI = \frac{1}{W \times H} \sum_{p=1}^{W} \sum_{q=1}^{H} \frac{|E_1(p,q) - E_2(p,q)|}{255} \times 100\% \quad (3)$$

where $D(p,\ q)$ represent the difference between $E_1(p,q)$ and $E_2(p,q)$. If $E_1(p,q) = E_2(p,q)$ then $D(p,\ q) = 0$, else $D(p,\ q) = 1$.

For an 8-bit grey image, the expected approximate are $NPCR_E = 99.6094\%$ and $UACI_E = 33.4635\%$. The calculated values of proposed algorithm are given in Table 3. The calculated NPCR and UACI values of immunohistochemical images are higher than the given in [4, 19]. Table 3 evidences that the proposed cryptosystem withstands strongly against differential attacks.

4.1.3 Correlation Coefficient Analysis

The correlation in between the two neighbouring pixels in the original and cipher image is tested by considering the whole pairs of neighbouring pixels in different directions (in horizontal, vertical and diagonal) from the original and cipher images and the correlation coefficients are es-

Table 3: The NPCR and UACI values of the experimental images

File name	NPCR%			UACI%		
	Red	Green	Blue	Red	Green	Blue
109225_A_8_1	99.6022	99.6119	99.6095	41.8200	41.3991	41.8393
109225_A_9_1	99.6066	99.6119	99.6077	42.0469	41.9835	43.2646
7436_A_7_1	99.6090	99.6117	99.6115	36.2793	36.3419	38.2842
7436_A_8_1	99.6070	99.6130	99.6159	39.6193	39.6151	40.9691
7436_A_9_1	99.6050	99.6085	99.6120	36.7789	36.7797	38.5883
54482_A_8_1	99.6096	99.6014	99.6094	41.2274	41.2965	42.0912
54482_A_9_1	99.6098	99.6035	99.6125	41.8828	41.9248	42.5739
Lena	99.6219	99.6044	99.6067	33.0119	30.6249	27.63317
Ref. [12] Image 1	99.6253	99.6332	99.6259	33.4312	33.3385	33.3522
Ref. [19] Lena	98.8373	99.6277	99.6445	33.0753	30.7349	28.0029
Ref. [4] Lena	99.6013	99.6131	99.6226	33.4210	33.4485	33.4815

timated using the equations given below.

$$
\begin{aligned}
E(x) &= \frac{1}{W \times H} \sum_{p=1}^{W} \sum_{q=1}^{H} x(p,q) \\
D(x) &= \frac{1}{W \times H} \sum_{p=1}^{W} \sum_{q=1}^{H} [x(p,q) - E(x)]^2 , \\
Conv(x,y) &= \frac{1}{W \times H} \sum_{p=1}^{W} \sum_{q=1}^{H} [x\,(p,q) - E(x)] \\
&\qquad [y\,(p,q) - E(y)] \\
\gamma_{xy} &= \frac{Conv(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \qquad (4)
\end{aligned}
$$

where x and y are the Red, Green, Blue values of two-adjacent pixels in the image and γ_{xy} is the correlation coefficient of two adjacent pixels. The correlation coefficients in horizontal (HC), vertical (VC) and diagonal (DC) directions of cipher images are shown in Table 4. From Table 4, the cipher images' correlation analysis values are nearing '0' and they are farther apart from the correlation analysis values of plain images. The correlation coefficient value of plain images are nearing 1. For the image 109225_A_8_1, the horizontal correlation coefficient values of plain and cipher images are 0.9 and -0.00034 respectively which proves the superiority of confusion and diffusion characteristics. Further, the values are identical with other literatures [12, 19, 34].

The values of the two neighbouring pixels for different directions are plotted as shown in the following Figures 7(a)-(f).

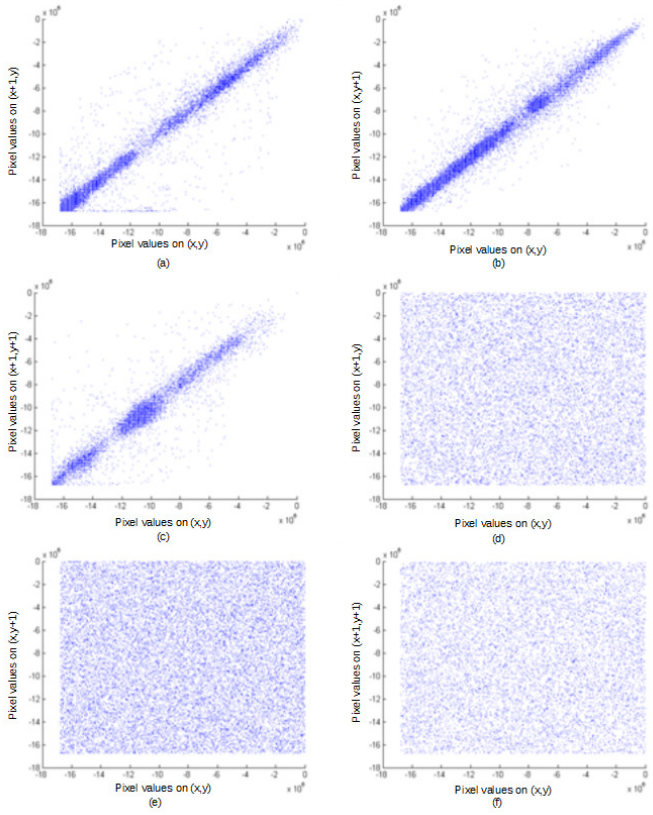

Figure 7: (a), (b), and (c) represents the horizontal, vertical, and diagonal correlation coefficients of original plain image; and (d), (e), and (f) represents horizontal, vertical, and diagonal correlation coefficients of cipher image.

4.1.4 Key Sensitivity Analysis

The control and the initial parameters used for medical image encryption in the experimental part are recalled and attempted to decrypt the ciphered images with different key. The different key is prearranged with the value of $r_1 = 3.612345678901235$ which have the tiny difference from the encryption key and then the consequent decrypted images are shown in Figures 8(a)-(b). It is evident that the use of slightly different key for decryption results in the entirely different decrypted image from the original which shows that the proposed system is highly key sensitive.

4.1.5 Histogram Analysis

A histogram is a graph which is used to show the pixel value distribution of an image. An opponent can guess

Table 4: The correlation coefficients (CC) in horizontal (HC), vertical (VC) and diagonal (DC) directions of cipher images

Component	CC	109225_A_8_1	109225_A_9_1	7436_A_7_1	7436_A_8_1	7436_A_9_1	54482_A_8_1	54482_A_9_1	Lena	Ref.[19]	Ref.[34]
Red	HC	-0.00034	-0.00027	0.00125	0.00238	0.00143	0.00280	0.00296	0.00298	-0.00110	-0.00201
	VC	-0.00043	-0.00028	-0.00270	-0.00157	-0.00290	-0.00160	-0.00108	0.00242	-0.01110	0.00293
	DC	0.00052	0.00015	-0.00012	-0.00080	0.00001	-0.00109	-0.00074	0.00312	0.00120	-0.00164
Green	HC	-0.00005	-0.00002	-0.00213	-0.00153	-0.00202	-0.00114	-0.00102	0.00105	-0.00008	-0.00149
	VC	-0.00095	-0.00088	-0.00447	-0.00491	-0.00507	-0.00451	-0.00462	0.00377	0.00130	-0.00221
	DC	0.00003	0.00003	0.00215	0.00192	0.00242	0.00141	0.00140	0.00148	0.00370	0.00349
Blue	HC	0.00012	0.00011	-0.00061	-0.00165	-0.00071	-0.00200	-0.00213	0.00247	0.00120	0.00055
	VC	-0.00147	-0.00151	0.00248	0.00235	0.00271	0.00200	0.00189	0.00103	0.00540	0.00037
	DC	0.00007	-0.00015	0.00332	0.00512	0.00294	0.00565	0.00602	0.00019	0.00110	-0.00092

Figure 8: (a)Decrypted cipher 109225_A_8_1 with wrong key (b) Decrypted cipher 109225_A_9_1 with wrong key

certain amount of data if the histogram is not flat enough. This is known as cipher only attack by analyzing the statistical property of the ciphered image. So the flat distribution is obtained during the medical image encryption. The distribution diagram of the color bytes Red, Green and Blue of the plain image and the cipher images are shown in the Figures 9(a)-(f). From the figures, it is proved that the cipher images are capable to resist the cipher only attack.

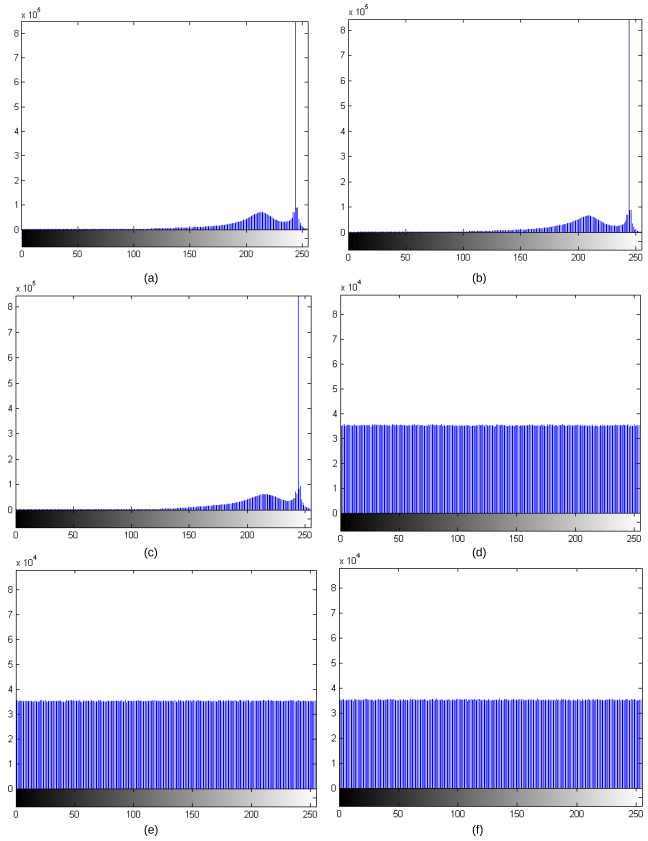

Figure 9: (a), (b), and (c) represents the R, G and B distribution of original plain image; and (d), (e), and (f) represents the R, G and B distribution of cipher image.

4.1.6 Mean-variance Color Byte Analysis

The definition of mean-variance color byte value is,

$$C = \frac{1}{W \times H} \sum_{p=1}^{W} \sum_{q=1}^{H} \left| B\left(p,q\right) - \overline{B} \right| \tag{5}$$

where $\bar{B}$ denotes the average of all the color values of image pixels, and $W \times H$ is the size of the plain image. The mean-variance colour byte value of cipher image is higher than the plain image and it proves that the scramble performance of the proposed algorithm is better. The red component's mean-variance color byte values of the plain and encrypted image are 20.16568 and 64.03818 respectively as shown in Table 5.

4.1.7 Information Entropy Analysis

To measure the randomness of encryption, information entropy is the metric which is given by Shannon is used. It is a mathematical theory which is used for data communication and storage. The entropy value of the source one is smaller than the ideal one. Let m be the information source, and the formula for calculating information

Table 5: Mean-variance color byte value of images

File name	Size	Plain Image			Cipher Image		
		Red	Green	Blue	Red	Green	Blue
109225_A_8_1	3000 × 3000	20.16568	22.89676	19.77083	64.03818	64.00136	63.98380
109225_A_9_1	3000 × 3000	18.75448	19.47938	12.89327	64.02837	64.00637	63.98659
7436_A_7_1	3000 × 3000	38.53359	37.90420	26.13839	63.97741	64.03066	63.93428
7436_A_8_1	3000 × 3000	26.88287	26.93553	18.96092	63.99242	64.06941	63.89905
7436_A_9_1	3000 × 3000	34.76804	34.54240	24.29289	63.97604	64.02882	63.94772
54482_A_8_1	3000 × 3000	19.59453	19.12174	15.27504	64.01059	64.07322	63.88637
54482_A_9_1	3000 × 3000	18.52384	18.19927	15.11005	64.00498	64.06246	63.87084
Lena	512 × 512	41.44229	43.78703	27.63417	64.00606	63.97906	64.15340

entropy is,

$$H(m) = \sum_{i=1}^{M} p(m_i) log \frac{1}{p(m_i)} \quad (6)$$

where M is the total number of symbols $m_i \in m; p(m_i)$ represents the probability of occurrence of symbol m_i and log denotes the base 2 logarithm so that entropy is expressed. For a random source emitting 256 symbols, image's entropy is $H(m) = 8$ bits. The information entropy values of the plain images and the cipher images are shown in Table 6.

4.2 Encryption Speed Analysis

The encryption speed is an important factor for describing the performance of the algorithm. Though the colour image size is 3000×3000, the algorithm executes the encryption efficiently by means of splitting the image into small chunks. Only one lakh permutation and diffusion sequences are generated to confuse and diffuse the chunks of image iteratively. Hence, the time taken for sequence generation is very less while comparing to generate sequences for whole image. Thereby, the algorithm encrypts faster. Further, the binary search algorithm speeds up the system and it is employed in indexing the sequences during permutation sequence generation. The encryption time for test medical images are calculated and compared with existing literatures (Table 7). The encryption speed is 12.68 Mbit/s which is much better than the existing methods, thereby it proves that the cryptosystem is very fast and efficient.

Table 7: The encryption speed comparisons

Encryption Algorithm	Encrption speed (Mbit/s)
Proposed algorithm	12.68
Ref. [13]	9.89
Ref. [22]	9.39
Ref. [25]	9.12

5 Discussion

The chaotic image encryption literature provides many encryption techniques proposed by targeting grey scale and colour images with smaller sizes [8, 16, 28, 29, 31]. Since the grey scale images are compiled with single channel pixels of 8-bit depth, the image encryption schemes are dealt with pixel shuffling and masking only. In case of 24-bit color images, the encryption is dealt with pixels and color bytes. So the proposed scheme is employing confusion and diffusion operations on both pixels and color bytes of an image. Also by using the key image, it is possible to add more randomness by executing XOR operation with the scrambled pixels and bytes. The resultant cipher image is having a good quality of confusion and diffusion properties. Further, the encryption standard is not only based on the logistic map, it also rigorously depends on the algorithm designed in the manner that involved in step by step operations on pixels and number of iterations in shuffling the pixels and colour bytes. The key space of the proposed system is sufficient and better than the other approaches [3, 26, 35]. The correlation coefficient, NPCR and UACI values of plain and cipher images are optimal and identical with existing approaches [4, 12, 19, 34]. The randomness of the cipher image is proved by better entropy values than the existing methods [12, 15, 25, 35]. The encryption speed is faster than the reported approach [13]. From the numerical results, it is evident that the proposed scheme possesses the highest security against various forms of threats. Hence, it is faster and efficient when compared to other existing methods.

6 Conclusion

This paper presents a faster and efficient symmetric cryptosystem using logistic map to encrypt large size 24-bit colour medical images. The proposed cryptosystem will be able to process different kinds of medical images as well as images of any size. Security analyses and experimental results demonstrated the effectiveness of the given scheme. The large key space supports the system to resist against different brute-force attacks. Statistical analysis reveals that the scheme protects the image from

Table 6: Information entropy values of plain and cipher images

File name	Plain Image			Cipher Image		
	Red	Green	Blue	Red	Green	Blue
109225_A_8_1	3.66064	3.71358	3.66988	7.99993	7.99994	7.99994
109225_A_9_1	4.30507	4.32594	4.15458	7.99995	7.99996	7.99996
7436_A_7_1	5.94937	5.90641	5.69366	7.99989	7.99989	7.99991
7436_A_8_1	4.98705	4.96496	4.79376	7.99978	7.99979	7.99983
7436_A_9_1	5.93084	5.89377	5.63771	7.99988	7.99992	7.99991
54482_A_8_1	3.97933	3.95653	3.86004	7.99948	7.99947	7.99955
54482_A_9_1	3.88024	3.86895	3.79533	7.99944	7.99945	7.99953
Lena	7.25310	7.59403	6.96842	7.99936	7.99926	7.99932
Ref. [12] Image 1	7.4567	7.2863	6.9628	7.9893	7.9879	7.9897
Ref. [35] Lena	-	-	-	7.99936	7.99935	7.99936
Ref. [15] Lena	-	-	-	7.98970	7.98770	7.98960
Ref. [25] Lena	-	-	-	7.99927	7.99924	7.99911

any form of statistical attack. The scheme has high sensitivity to plain image and key, so it can withstand on differential attack. Based on the performance and results of security analsyses, one can conclude that the algorithm is much faster and efficient. On the whole, the proposed encryption scheme has high-level of security and it can be utilized in secure medical image storage and communications.

Acknowledgements

The authors would like to thank the authorities of Bharathiar University, Coimbatore, India for providing the necessary laboratory facilities to carry out this study.

References

[1] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[2] G. Bellandi, M. Giannini, and C. Grande, "Mobile ehealth technology and healthcare quality impacts in italy," *International Journal of Healthcare Management*, vol. 6, no. 3, pp. 192–200, 2013.

[3] A. Diaconu and K. Loukhaoukha, "An improved secure image encryption algorithm based on rubiks cube principle and digital chaotic cipher," *Mathematical Problems in Engineering*, pp. 1–10, 2013.

[4] C. Dong, "Color image encryption using one-time keys and coupled chaotic systems," *Signal Processing Image Communication*, vol. 29, no. 5, pp. 628–640, 2014.

[5] J. Galvez, "Whole slide imaging and telepathology," *Breast Cancer Research*, vol. 5, pp. 1–2, 2003.

[6] T. G. Gao and Z. Q. Chen, "Image encryption based on a new total shuffling algorithm," *Chaos Solitons Fractals*, vol. 38, no. 1, pp. 213–220, 2008.

[7] T. Gulom, "The encryption algorithm GOST28147-89-PES16-2 and GOST28147-89-RFWKPES16-2," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 1–11, 2017.

[8] S. K. A. Hafiz, A. G. Radwan, S. H. Abdel Haleem, and M. L. Barakat, "A fractal-based image encryption system," *IET Image Processing*, vol. 8, no. 12, pp. 742–752, 2014.

[9] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "The study on data hiding in medical images," *International Journal of Network Security*, vol. 14, no. 6, pp. 301–309, 2012.

[10] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images," *Journal of Systems and Software*, vol. 86, no. 3, pp. 716–727, 2013.

[11] I. Hussain and T. Shah, "Application of s-box and chaotic map for image encryption," *Mathematical and Computer Modelling*, vol. 57, no. 9-10, pp. 2576–2579, 2013.

[12] G. Liu, J. Li, and H. Liu, "Chaos-based color pathological image encryption scheme using one-time keys," *Computers in Biology and Medicine*, vol. 45, pp. 111–117, 2014.

[13] H. Liu and C. Jin, "A color image encryption scheme based on arnold scrambling and quantum chaotic," *International Journal of Network Security*, vol. 19, no. 3, pp. 347–357, 2017.

[14] L. Liu and Z. Cao, "Analysis of two condentiality-preserving image search schemes based on additive homomorphic encryption," *International Journal of Electronics and Information Engineering*, vol. 5, no. 1, pp. 1–5, 2016.

[15] L. Liu, Q. Zhang, and X. Wei, "A rgb image encryption algorithm based on dna encoding and chaos map," *Computers and Electrical Engineering*, vol. 38, pp. 1240–1248, 2012.

[16] A. Masmoudi and W. Puech, "Lossless chaos-based crypto-compression scheme for image protection," *IET Image Processing*, vol. 8, no. 12, pp. 671–686, 2014.

[17] R. Matthes, "On the derivation of a chaotic encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, 1989.

[18] A. Mersaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1–11, 2016.

[19] P. Padmapriya, R. Amirtharajan, K. Thenmozhi, and J. B. B. Rayappan, "Triple chaotic image scrambling on rgb a random image encryption approach," *Security and Communication Networks*, vol. 8, no. 18, pp. 3335–3345, 2015.

[20] M. Y. M. Parvees, J. A. Samath, and B. P. Bose, "Secured medical images - a chaotic pixel scrambling approach," *Journal of Medical Systems*, vol. 40, no. 11, pp. 232:1–232:11, 2016.

[21] M. Y. M. Parvees, J. A. Samath, I. K. Raj, and B. P. Bose, "A colour byte scrambling technique for efficient image encryption based on combined chaotic map," in *Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT'16)*, pp. 1067–1072, Mar. 2016.

[22] V. Patidar, N. K. Pareek, and K. K. Sud, "A new substitution-diffusion based image cipher using chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3056–3075, 2009.

[23] Protein Atlas, *The Human Protein Atlas Project*, Apr. 15, 2017. (`http://www.proteinatlas.org`)

[24] E. Ryhan, "Telemedicine: Current and future perspectives," *International Journal of Computer Science Issues*, vol. 10, no. 6, pp. 242–249, 2013.

[25] S. M. Seyedzadeh and S. Mirzakuchaki, "A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map," *Signal Processing*, vol. 92, pp. 1202–1215, 2012.

[26] B. Stoyanov and K. Kordov, "Image encryption using chebyshev map and rotation equation," *Entropy*, vol. 17, pp. 2117–2139, 2015.

[27] M. Uhlen, P. Oksvold, L. Fagerberg, E. Lundberg, K. Jonasson, M. Forsberg, M. Zwahlen, C. Kampf, K. Wester, S. Hober, H. Wernerus, L. Bjorling, and F. Ponten, "Towards a knowledge-based human protein atlas," *Nature Biotechnology*, vol. 28, no. 12, pp. 1248–1250, 2010.

[28] X. Wang and K. Guo, "A new image alternate encryption algorithm based on chaotic map," *Nonlinear Dynamics*, vol. 76, pp. 1943–1950, 2014.

[29] X. Y. Wang, "New chaotic encryption algorithm based on chaotic sequence and plain text," *IET Information Security*, vol. 8, no. 3, pp. 213–216, 2014.

[30] X. Y. Wang and J. F. Zhao, "A new image encryption algorithm based on chaos," *Optics Communications*, vol. 285, no. 5, pp. 562–566, 2012.

[31] Y. Wang, K. W. Wong, X. F. Liao, and G. R. Chen, "A new chaos-based fast image encryption algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 514–522, 2011.

[32] J. D. Webster and R. W. Dunstan, "Whole-slide imaging and automated image analysis: considerations and opportunities in the practice of pathology," *Veterinary Pathology*, vol. 51, no. 1, pp. 211–223, 2014.

[33] M. Yang, M. Trifas, and L. Chen, "Secure patient information and privacy in medical imaging, systemics," *Cybernetics and Informatics*, vol. 8, no. 3, pp. 63–66, 2010.

[34] W. Zhang, K. W. Wong, H. Yu, and Z. L. Zhu, "A symmetric color image encryption algorithm using the intrinsic features of bit distributions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 3, pp. 584–600, 2013.

[35] Y. Q. Zhang and X. Y. Wang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Information Sciences*, vol. 273, pp. 329–351, 2014.

Biography

M. Y. Mohamed Parvees received his M.Sc. (Information Technology) in 2002 from Gandhigram Rural Institute - Deemed University and completed M.Phil. (Computer Science) in 2004 from Annamalai University, India. Presently, he is a faculty in Department of Computer and Information Science, Annamalai University. He pursues his Ph.D. degree in Bharathiar University. He has few international and national publications. His research interests include cryptography, multimedia security and medical information systems.

Dr. J. Abdul Samath received his Ph.D. (Computer Science) from Gandhigram Rural Institute - Deemed University. Currently, he is working as an Assistant Professor in Government Arts College, Udumalpet. He has 10 years of teaching experience and he has published 12 research articles in international journals. His research interests include neural networks, image processing, control theory, cryptography and medical image analysis.

B. Parameswaran Bose received his M.Sc. (Information Technology) in 2002 from Gandhigram Rural Institute - Deemed University. He has 7 years of experience in software research and development mainly in the field of application programming, information security and web technologies with knowledge in analyzing, developing and deploying critical applications. Presently, he does research on cryptography and information security.