REVIEW OF COMPUTER ENGINEERING STUDIES

ISSN: 2369-0755 (Print), 2369-0763 (Online)
Vol. 4, No. 2, June 2017, pp. 70-75
DOI: 10.18280/rces.040206
Licensed under CC BY-NC 4.0

A publication of IIETA

International Information and Engineering Technology Association
Advancing the World of Information and Engineering

http://www.iieta.org/Journals/RCES

Secured image encryption scheme based on DNA encoding and chaotic map

Soumya Paul[1*], Pranjal Dasgupta[2], Prabir Kr. Naskar[3], Atal Chaudhuri[1]

[1] Department of Computer Science & Engineering Jadavpur University, Kolkata 700010, India
[2] Tata Consultancy Services Kolkata Infospace, WB 700010, India
[3] Department of Computer Science & Engineering Govt. College of Engineering & Textile Technology, Serampore, Hooghly 712201, India

Email: soumyapaul5a1@gmail.com

ABSTRACT

Recent research has considered DNA as a medium for ultra-scale computation and ultra-compact information storage. In recent years, the chaos based cryptographic algorithms have suggested some new and efficient ways to develop secure image encryption techniques. In this communication, we propose a new approach for image encryption based on hybrid model of chaotic logistic maps, deoxyribonucleic acid (DNA) masking and DNA replacement algorithm in order to meet the requirements of the secure image transfer. The significant advantage of this approach is improving the quality of DNA masks to obtain the best mask that is compatible with plain images. The experimental results and entropy analysis both confirm that the proposed scheme not only demonstrates excellent encryption but also resists various typical attacks.

Keywords: Image Encryption, Chaotic Map, DNA Encoding, Histogram Analysis, Entropy Analysis.

1. INTRODUCTION

Because of the rapid development of the internet and innovation in technologies, the security of multimedia data content, such as videos and images, has become a serious problem [1]. In recent years, several chaos-based algorithms have been proposed [1, 2] and have found wide popularity among researchers. The plain image is partitioned into blocks, and spatiotemporal chaos is used to shuffle the blocks while the diffusion step is simultaneously applied. One of the latest and most successful image encryption methods is DNA-based image encryption [3]. Because of the large data requirement and high correlation among adjacent pixels in an image, preliminary techniques, such as AES, DES, IDEA, and RSA, are not efficient for proper encryption [5]. To solve this problem, researchers have focused on methods that satisfy confusion and diffusion requirements [6, 7].

In recent years, several chaos-based algorithms have been proposed [1, 4, 7, 8] and have found wide popularity among researchers. Because of the inherent features of chaos systems, such as sensitivity to initial value and randomness, the chaos system-based image encryption method appears to be suitable for high-security encryption. Owing to perfect chaotic properties Logistic map has been used as a pseudorandom sequence generator. Both these maps have enormous key space thereby making the image far less vulnerable [8]. One of the latest and most successful image encryption methods is DNA-based image encryption [9, 3, 10]. The fundamental idea of all DNA-based image encryption is categorized in two phases: first, using DNA theory to encode plain image pixels to a DNA sequence and using those rules to generate the key image. In the second phase, the encoded plain image pixels generated a key image based on DNA operation rules and form the cipher image. The specified numbers of cipher images are extracted from the plain image by using a chaotic function. Entropy and the correlation coefficient as a fitness function are used for improving the quality of the cipher image. As a proposal for a powerful image encryption algorithm, in this study, a novel hybrid model of DNA sequence, chaotic mapping has been developed. In the first stage of the algorithm, some DNA masks are created using DNA sequence and logistic map [4, 11, 13, 14]. These mask sizes are equal to the plain image. The initial cipher-images can be more secure when the proposed method use DNA mask and also it reaches to high entropy, quickly. In the second stage, DNA replacement function is used. After creating DNA key value, XOR operation is performed between image and DNA key value to produce cipher image [10, 12].

In fact, we aim to use the power of DNA sequences in information encryption. The rest of paper is organized as follows. The DNA masking and logistic mapping are described in Section 2. Section 3 is dedicated to introducing the proposed method. The experimental results and security analysis are provided in Section 4 and 5. The conclusions are discussed in the last section. In this project, we want to design new image encryption scheme, which is secured against any modern cryptographic attacks.

2. PRELIMINARIES

In this section, logistic mapping function and the DNA masking are explained.

A. Characteristics of logistic map

One of the most popular and useful chaotic functions is the logistic map function [11, 12, 13, 14] described in Equation (1).

$$\left.\begin{aligned} f(x) &= r\times x\times(1-x) \\ x_{n+1} &= f(x_n) \end{aligned}\right\} \qquad (1)$$

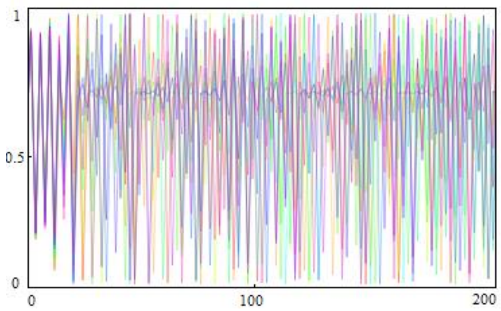


Figure 1. Logistic map function for ***r***=3.999 and x_n=0.66

The values of the logistic map function for r=3.999 are highly randomized and uniformly distributed over 0 to 1 values.

B. Deoxyribonucleic acid sequence

Knowledge of deoxyribonucleic acid sequence (DNA) sequences has become indispensable for basic biological research, and in numerous applied fields such as diagnostic, biotechnology, forensics, and biological systematic. There are four different nucleic acids in a DNA sequence: A (adenine), T (thymine), C (cytosine), and G (guanine). In regards to the rules of base pairing, the purine adenine (A) always pairs with the pyrimidine thymine (T), and the pyrimidine cytosine (C) always pairs with the purine guanine (G). Figure 1 shows a simple DNA structure. It can be concluded that A and T are complementary, and G and C are also complementary [13, 14].

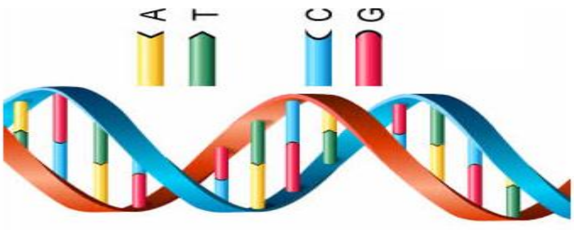


Figure 2. Simple DNA structure

These relationships are often called the Watson-Crick base pairing rules and are named after the two scientists who discovered their structural basis. As we know, in the binary system, 0 and 1 are complementary. Therefore, it can be concluded that 00 and 11 are complementary and also 01 and 10 are complementary. To satisfy the Watson-Crick base pairing rules, Table 1 introduces the coding and decoding map rules for the DNA sequence used in this paper and Table 2 shows XOR operation for DNA sequences.

Table 1. Encoding and decoding map rules for DNA sequences

	A	T	C	G
Rule 1	00	11	10	01
Rule 2	00	11	01	10
Rule 3	11	00	10	01
Rule 4	11	00	01	10
Rule 5	10	01	00	11
Rule 6	01	10	00	11
Rule 7	10	01	11	00
Rule 8	01	10	11	00

Table 2. XOR operation for DNA sequences

⊕	A	T	C	G
A	A	T	C	G
T	T	A	G	C
C	C	G	A	T
G	G	C	T	A

3. PROPOSED SCHEME

A. Generating the value of x₀ and rule for DNA replacement

To have a safe and secure encryption algorithm [4], the value of x_0 in Equation 1 is taken from a 128-bit key as explained by Equation (2) and Equation (3).

$$Key = \{K_1, K_{2,}K_3, \dots, K_{16}\} \dots\dots\dots \qquad (2)$$

$$x_0 = \frac{K_1 \oplus \dots \oplus K_{16} + \sum_1^{16} K_i}{2^{13}} \dots\dots\dots \qquad (3)$$

where K_i represents an 8-bytes character and ⊕denotes an exclusive OR. For generating the rule for DNA replacement Equation (4) is uses

$$Rule = \lfloor x_i \times 256 \rfloor \% 8 \qquad (4)$$

B. Generate 32 byte shuffled hash code with padded noise bits

For secure image transmission 32 bytes hash code is generated. Hash code generation process is depended on the generated value of x_0 from intermediate key (K') and DNA replacement algorithm. For Every consecutive 4 bytes new DNA rule is generated. From the newly generated rule, new position of hash code is calculated. In this way hash code is position wised shuffled. After that, rule wise extra noise bits are padded with newly generated shuffled hash code and all of these are written in a 'key' file (i.e. called Encrypted Key File). This Encrypted Key File is securely transmitted through internet to receiver site for decryption of Cipher image.

C. Secret bytes selection & encrypted byte generation

First four selected secret bytes are position wise shuffled using new rule of Equation 4 and DNA pattern. Consider, selected secret bytes are A = {120, 225, 242, 252} and the rule is 4. The DNA pattern for rule 4 is {11, 00, 01, 10} = {3, 0, 1, 2}. Therefore, the value of A' is A' = {A[3], A[0], A[1], A[2]} = {252,120,225,242}. Consider corresponding ASCII values of each byte of intermediate 16 byte key K' are represented by K'_i = {52, 73,190, 219}. So calculated value of x_0 = 0.522949, x_1 = 0.8841, x_2 = 0.4095, x_3 = 0.96701.

So generated rules are r_i = {5, 7, 4, 2} [i.e. for the value of x_0, r_0 = (0.522949 x 256) mod 8 = 5].

Now, K'_i is generated from K_i using new rule of Equation 4 and DNA pattern. Then, intermediate secret byte A'_i is generated using new rule of Equation 4 and DNA pattern. XOR operation performed between K'_i and A'_i to generate encrypted byte. Write this in encrypted secret file.

D. Retrieval of intermediate Key K' from Shuffled Hash code

At decryption end only cipher image and shuffled hash code are present and x_0 value is generated with the help of original key K. Shuffled Hash code is deshuffled using DNA rule that is used for shuffling. In this way original hash code generated. From original hash code, intermediate K' is retrieved by using reverse DNA replacement algorithm.

E. Decrypted byte generation from encrypted file

Read encrypted byte. Calculate K'_i and A'_i using previous process in time of encryption. XOR operation performed between K'_i and A'_i. 4 bytes shuffle using rule and DNA pattern which is using in time of encryption to generate decrypted byte.

4. MATHEMATICAL IMPLEMENTATION

Consider selected four secret bytes are: -
A[40] = 100, A[41] = 101, A[42] = 102, A[43] = 103, A[44] = 104. Suppose intermediate key K' is "#*uA79%Th@M$". So ASCII values of K' are {117,42,108,65,55,57, 37,84, 104, 64, 77, 36}. x_0 value of K' is generated from Equation 2 i.e. 0. 25332. From Equation 5 DNA rules are generated. In this example DNA Rules {1, 2, 3, 2} are generated. After applying DNA replacement algorithm according to DNA rule-1, bit-1 $(01100100)_2$ is replaced to $(11101100)_2$ i.e. equivalent with $(236)_{10}$. In this way bit 2 bit 3 and bit 4 replaced to 223,34 and 238 respectively. Now new intermediate four bytes are A' [40] = 236, A' [41] = 223, A' [42] = 34, A' [43] = 238.

Now these new values are position wised shuffled. Calculate div = memory location of selected byte of image/4 = (40/4 = 10). If for each calculation rem = memory location of selected byte of image%4 = 0, then new rule is generated from Equation (4). Suppose new rule is 5 for the calculation rem = (40%4 = 0). Then calculate the new position of selected byte of the image using Equation (5), Equation (6).

$$val = DNA[rule][i] \dots\dots\dots \qquad (5)$$

Here value of i is bit position among (0 to 3) of selected byte of image.

$$Newpos = (4 * div + val) \dots\dots\dots \qquad (6)$$

So in this example DNA [5][0] = 2 & Newpos = (10*4) + 2 = 42. So first byte (i.e. 40th location) writes on 42th location. In this way DNA [5][1] = 1 & Newpos = 41, DNA [5][2] = 0 & Newpos = 40 and DNA [5][3] = 3 & Newpos = 43. That means Shuffled secret bytes are A"[40] = 34, A"[41] = 223, A"[42] = 236, A"[43] = 238. In this way cipher image is created.

For generating original image from cipher image just reversed DNA replacement algorithm is used. That means rem = 40%4 = 0, get div = 40/4 = 10. Generate same rule using Equation (3) i.e. rule 5. Use Equation (4) and (5) to get bit position. Write that positional value to original position. But After this process original value is not found. Only intermediate secret bytes are reconstructed. To get original values Equation (7) is used.

$$Val = DNA[rule][bit] \dots\dots\dots \qquad (7)$$

In this example A'[40] = $(236)_{10}$ = $(11101100)_2$. Generated rule is 1. So Applying Equation (7) DNA [1][$(11)_2$] = 01, DNA[1][$(10)_2$] = 10, DNA[1][$(11)_2$] = 01, DNA[1][$(00)_2$] = 00. So original value of A[40] = $(01100100)_2$ = 100.

So above mentioned ASCII values of K' are {117, 42, 108, 65, 55, 57, 37, 84, 104, 64, 77, 36}. Suppose user given original key K at encryption time is "%IcAst@2017#". x_0 values from Equation 3 is 0.212158. Now calculated K'' = 256-K', i.e {139,214,148,191,201…}. Find rem = i%4, for each rem = 0 new rule is generated using Equation (4). Suppose for first four bit rule = 4. mul = rule+1 = (4+1 = 5).

Get value = DNA[rule][rem] = 3, pw = value+1 = 4, cal = 2^{pw} = 16, Sub = cal*mul = 80, bt = K''-Sub = $(64)_{10}$ and after conversion of 'bt' to hex value is $(40)_{16}$. So for 112 converted hex code is 64. For 2nd bit key 114 converted to 84. In this way 32 byte hash code is generated.

Now hash code is shuffled for enhancing security. x_0 calculated from K. Rule is generated (Suppose rule = 6). Get div = pos/4 = (0/4 = 0). Then Equation (8) and Equation (9) are applied for shuffling.

$$Newpos = DNA[rule][div] \dots\dots\dots \qquad (8)$$

$$Wrpos = (div \times 4) + Newpos \qquad (9)$$

So for 1st bit of hash Newpos = DNA[6][0] = 1 & Wrpos = (0*4)+Newpos = 1, likely for 2nd bit of hash Newpos = DNA[6][1] = 2 & Wrpos = 0+2 = 2, for 3rd bit of hase Newpos = DNA[6][2] = 0 & Wrpos = 0+0 = 0, for 4th bit of hash Newpos = DNA [6][3] = 3 & Wrpos = (0*4) +3 = 3. So 1st four bit of original hash (4084) is shuffled to form (8364). In this way 32 bit hash is shuffled.

For enhancing secure transmission again shuffled hash code is padded with extra noise bits. 'Val' is generate using Equation (10).

$$Val = (x_i \times 255)\%15 \dots \qquad (10)$$

In this example for x_0 = 2.1215. Val = (54.06)%15 = 9. So for shuffled hash code 83, new byte = 83-9 = 74. For determining the range of padding bits Pw is calculated using Equation (11).

$$Pw = (x_i \times 255)\%8 \ldots\ldots\ldots \quad (11)$$

So, Pw = (54.06)%8 = 6. Pw = Pw+1 = (6+1=7) and calculate 2Pw = 27 = 128. So 128 bit alphanumeric and special characters are padded along with shuffled hash code using modified random function.

5. EXPERIMENTAL RESULT

Image encryption is completely different from text encryption, because a digital image contains bulk of highly correlated data. As a result, traditional encryption algorithm is not suitable for digital image encryption. Figure 3.b shows encrypted image and lossless decrypted image Figure 3.d for proper secret key. Figure 3.c is also shown that decryption is not possible for wrong key. Some other experimental results for different gray scale images are shown in Figure 4.

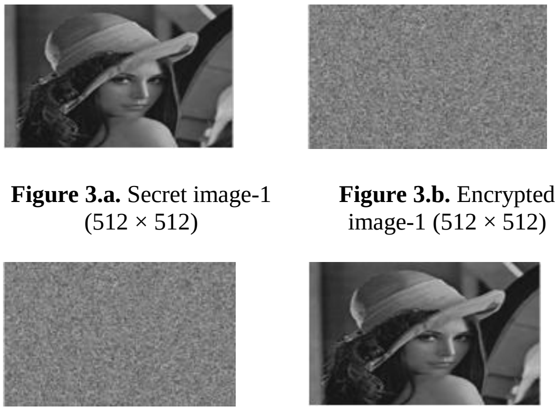


Figure 3.a. Secret image-1 (512 × 512)

Figure 3.b. Encrypted image-1 (512 × 512)

Figure 3.c. Decrypted image using wrong key (512 × 512)

Figure 3.d. Decrypted image using proper key (512 × 512)

Figure 3. Secret image encryption & decryption

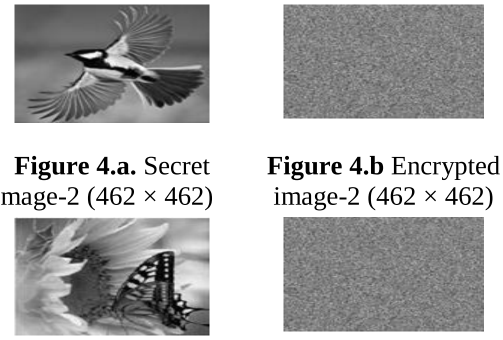


Figure 4.a. Secret image-2 (462 × 462)

Figure 4.b Encrypted image-2 (462 × 462)

Figure 4.c. Secret image-3 (450 × 450)

Figure 4.d. Encrypted image-3 (450 × 450)

Figure 4. Different secret images & encrypted Images

6. STRENGTH & SECURITY ANALYSIS

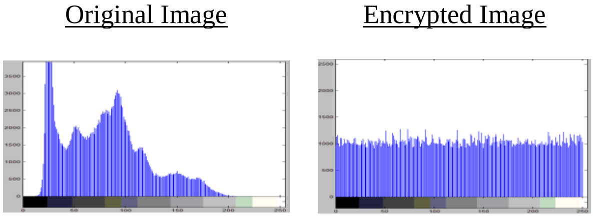
A secure encryption algorithm should be robust against all types of attacks such as cryptanalytic, statistical and brute-force attacks. Here we discuss the security analysis of the proposed algorithm by addressing key space and key sensitivity analysis, statistical analysis, and differential analysis. The resistance against different types of attack is useful measure for the performance of a cryptosystem. Some security analysis results are incorporated in the following section to establish the strength of our proposed scheme.

A. Key space and key sensitivity analysis

A good cryptosystem should have sufficiently large key space to make the brute-force attack infeasible. Key spaces imply the size of keys used for the purpose of encryption and decryption. The proposed scheme uses a secret key as collection of 16 bytes value, i.e. 16×8 = 128 bits key. Therefore total key space of is 2^{128}, which is large enough for brute-force attack according to the present available computational speed. On the other hand the encryption and decryption algorithms are highly sensitive to the secret key. The change of single bit/byte in the secret key produces a completely different encrypted or decrypted image.

B. Histogram analysis

The histogram analysis clarifies how pixels in an image are distributed by plotting the number of pixels at each intensity level. Figure 5 shows histogram of secret image and encrypted image, where Figure 5 (Original Image) shows histogram of secret image Figure 3.a and Figure 5 (Encrypted Image) shows histogram of encrypted image Figure 3.b. The histogram of encrypted image has uniform distribution which is significantly different from original image and has no statistical similarity in appearance.



Original Image

Encrypted Image

Figure 5. Histogram of secret image and encrypted image

C. Differential attack & correlation value

The major requirement of all the encryption techniques is the encrypted image should be greatly different from its original form. Two measures are adopted to quantify this requirement. They are Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI). The NPCR is used to measure the number of pixels in difference of gray level in two images. Let P (i, j) and P' (i, j) be the i$_{th}$ row and j$_{th}$ column pixel of two images P and P', respectively, the NPCR can be defined as

$$NPCR = \frac{\sum_{i,j} C(i,j)}{N} \times 100 \ldots \quad \ldots (12)$$

where N is the total number of pixels in the image.
C (i, j) is defined as

$$C(i,j) = \begin{cases} 0 & P(i,j) = P'(i,j) \\ 1 & P(i,j) \neq P'(i,j) \end{cases}$$

The NPCR values for various images using Equation (12) are shown in Table 3. The high value of NPCR means the pixel values are dramatically randomized. This result indicates that the plain-image and the encrypted image are significantly different from one another, so the proposed algorithm is highly resistive against differential attack. Another quantity, Unified Average Changing Intensity (UACI) measures the average intensity differences between the two images. It can be defined as

$$UACI = \frac{1}{N}\left[\sum_{i,j}\frac{|P(i,j) - P'(i,j)|}{255}\right]\times 100 \ldots\ldots \quad (13)$$

Two quantities, NPCR and UACI are calculated for various images using Equation (12) and Equation (13) respectively. The test results for NPCR and UACI are shown in Table 3.

A secure encryption scheme should generate a completely noisy encrypted image independent of the original secret image. Therefore, they must have a very low correlation coefficient which should be very closer to zero. Here, we calculate the correlation between original and encrypted image using Equation (14).

$$r = \frac{\sum_i \sum_j (M_{ij} - \bar{M})(N_{ij} - \bar{N})}{\sqrt{(\sum_i \sum_j (M_{ij} - \bar{M})^2)(\sum_i \sum_j (N_{ij} - \bar{N})^2)}} \ldots\ldots \quad (14)$$

where M and N are two images of same size and $\overline{M}$ and $\overline{N}$ indicate the mean of the images M and N respectively. A low value of correlation coefficient shows that there is no straight relation between the original and encrypted images. Following Table 3 shows NPCP, UACI and correlation value between gray scale images and encrypted images.

Table 3. NPCR, UACI & correlation value for different images

Input Images	**NPCR**	**UACI**	**Correlation Value**
Original image Vs. Encrypted image			
Figure 3.a & Figure 3.b	99.465	11.676	-0.0046
Figure 4.a & Figure 4.b	99.761	22.671	-0.0023
Figure 4.c & Figure 4.d	99.325	21.938	0.0018
Original image Vs. Decrypted image			
Figure 3.a & Figure 3.d	0.00	0.00	1.0000

Higher value of NPCR and lower value of UACI indicate the absence of any probable statistical relationship between original image and encrypted image. Whereas, the last row of Table 3 shows that the result of NPCR and UACI are zero which proves lossless reconstruction of secret image. Above result shows, a low correlation value exists between original image and encrypted image. Therefore, encrypted image is completely different from original image. We get the correlation value one for secret image (Figure 3.a) and decrypted image (Figure 3.d), which proves lossless reconstruction of the secret image.

D. Entropy analysis

Entropy analysis is an essential statistical measurement, which is used to test the robustness of an image encryption algorithm. Entropy measurement of a source k is defined as,

$$H(k) = \sum_i p(k_i)\log_2\frac{1}{p(k_i)} \ldots\ldots\ldots \quad (15)$$

where p(k$_i$) represents the probability of the pixel value xi. If the probability of occurrence of each pixel value is same, according to the Equation (15) entropy value will be H (k) = 8. This will be the maximum entropy for an image having truly uniform pixel distribution. An encrypted image will be considered robust if its entropy tends to the value 8. Therefore, higher the entropy value of an encrypted image betters the security.

Table 4. Entropy analysis of different images

Original Image	**Image Entropy**	**Encrypted Image**	**Image Entropy**
Figure 3.a	7.344	Figure 3.b	7.990
Figure 4.a	7.033	Figure 4.b	7.989
Figure 4.c	7.546	Figure 4.d	7.994

Table 4 shows entropy of all encrypted images are close to 8, irrespective of entropy of original images, thus the proposed algorithm is robust enough.

7. CONCLUSIONS

To encrypt and properly decrypt an image is very essential. For this purpose, proper key selection is very crucial. There are algorithms in cryptographic system design is the algorithm to generate `key'. It specifies the manner in which the `key' is to be chosen. DNA encryption interprets a binary value into combination of A, T, G, C-these four types of elements, (which are actually nucleic acid elements) and performs operations with corresponding set of rules. In this paper, these operations are performed with corresponding set of 8 rules that define how to replace an element with other. The key encrypts each pixel of image and in receiver side the image could be decrypted with exactly that key only. Slight change in value of key causes huge blunder in interpretation. The proposed method effectively reduces the correlation between two adjacent pixels and increases the entropy of the cipher or encrypted image. The numerical experiments demonstrate the high resistance of the proposed method against common attacks.

ACKNOWLEDGEMENT

We are thankful to the Department of Computer Science and Engineering of Jadavpur University and GCETTS for giving us the platform for planning and developing this work in the departmental laboratories.

REFERENCES

[1] Abdullah A.H., Enayatifar R., Lee M. (2012). A hybrid genetic algorithm and chaotic function model for image encryption, *AEU Int. J Electron Communication*, Vol. 66, pp. 806-816.

[2] Bakhshandeh A., Eslami Z. (2013). An authenticated image encryption scheme based on chaotic maps and memory cellular automata, *Opt. Lasers Eng.*, Vol. 51,

pp. 665-673.

[3] Liu H., Wang X., Nadir A. (2012). Image encryption using DNA complementary rule and chaotic maps, *Applied Soft Computing*, Vol. 12, pp. 1457-1466.

[4] Enayatifar R., Abdullah A.H, Isnin I.V. (2014). Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence, *Optics and Lasers in Engineering*, Vol. 56, pp. 83-93.

[5] Zhang G., Liu Q. (2011). A novel image encryption method based on total shuffling scheme, *Opt. Commun.*, Vol. 284, pp. 2775-2780.

[6] Zhu Z.L., Zhang W., Wong K.W., Yu H.A. (2010). Chaos-based symmetric image encryption scheme using a bit-level permutation, *Inf. Sci.*, Vol. 181, pp. 1171-1186.

[7] Sathishkumar G.A., Bhoopathybagan K., Sriraam N. (2011). Image encryption based on diffusion and multiple chaotic maps, *International Journal of Network Security & Its Applications (IJNSA)*, Vol. 3, No. 2, pp. 181-194.

[8] Shekhar S., Srivastava H., Dutta M.K. (2012). An efficient adaptive encryption algorithm for digital images, *International Journal of Computer and Electrical Engineering*, Vol. 4, No. 3, pp. 380-383.

[9] Zhang Q., Guo L., Wei X. (2013). A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system, *Optik-Int J Light Electron Opt*, Vol. 124, pp. 3596-3600.

[10] Divya V.V., Sudha S.K., Resmy V.R. (2012). Simple and secure image encryption, *IJCSI International Journal of Computer Science Issues*, Vol. 9, No. 3, pp. 286-289.

[11] Pareek N.K., Patidar V., Sud K.K. (2006). Image encryption using chaotic logistic map, *Image Vision Computing*, Vol. 24, pp. 926-960.

[12] Kanso A., Smaoui, N. (2009). Logistic chaotic maps for binary numbers generations, *Chaos, Solitons and Fractals*, Vol. 40, pp. 2557-2568.

[13] Naskar P.K., Chaudhuri A. (2015). A robust image encryption technique using dual chaotic map, *International Journal of Electronic Security and Digital Forensics, Inder Science*, Vol. 7, No. 4, pp. 358-380.

[14] Naskar P.K., Chaudhuri A. (2016). Secured secret sharing technique based on chaotic map and DNA encoding with application on secret image, *The Imaging Science Journal, Taylor & Francis*, Vol. 64, No. 8, pp. 460-470.