International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:6, 2014

Gray Level Image Encryption

Roza Afarin, Saeed Mozaffari

***Abstract*—**The aim of this paper is image encryption using Genetic Algorithm (GA). The proposed encryption method consists of two phases. In modification phase, pixels locations are altered to reduce correlation among adjacent pixels. Then, pixels values are changed in the diffusion phase to encrypt the input image. Both phases are performed by GA with binary chromosomes. For modification phase, these binary patterns are generated by Local Binary Pattern (LBP) operator while for diffusion phase binary chromosomes are obtained by Bit Plane Slicing (BPS). Initial population in GA includes rows and columns of the input image. Instead of subjective selection of parents from this initial population, a random generator with predefined key is utilized. It is necessary to decrypt the coded image and reconstruct the initial input image. Fitness function is defined as average of transition from 0 to 1 in LBP image and histogram uniformity in modification and diffusion phases, respectively. Randomness of the encrypted image is measured by entropy, correlation coefficients and histogram analysis. Experimental results show that the proposed method is fast enough and can be used effectively for image encryption.

***Keywords*—**Correlation coefficients, Genetic algorithm, Image encryption, Image entropy.

I. Introduction

IN the technological world, information security becomes a main concern. Encryption algorithms have emerged as practical tools for ownership authentication and copyright protection [1], [2]. Digital images play an important role in our daily lives as a big source of information and many algorithms have been proposed for digital image encryption [3]. Among them, Genetic algorithm (GA) has been used frequently in conjunction with well established cryptography methods or in a stand-alone fashion.

The aim of combining GA with common encryption algorithms is to enhance randomness of the encrypted image. In [4], first the input image is encrypted by chaotic function. Then GA is applied to obtain an optimum encrypted image. In [5], [6] the input image is divided into four equal sub-images each encrypted by separate keys. Two sub-images with lowest correlation are selected as initial population for GA. Then, each of these sub-images is divided into four smaller parts.

Cross-over operation is defined as substituting pixel values in each portion. After some generation, the best encrypted image with low similarity between its parts is achieved.

GA can also be used directly for image encryption. Cross-over and mutation operations change pixels values to reduce correlation coefficients between adjacent pixels. In [7], pixels located at even positions are considered as first parents. A set of predefined functions give the first parent and determine its corresponding second parent. Cross-over is defined to exchange binary values of parents from the middle point. For mutation, the function which finds out the second parent is altered. GA is also utilized in frequency domain. Abduhaiba et al. proposed an image encryption method in which two frequency components are selected randomly as parents [8]. By cross-over their imaginary parts are dislocated and in mutation, real parts of parents are subtracted from the input key.

Since encrypted image randomness strongly depends on encryption key, GA is also used to optimize the key. A key with n bits can be rearranged by $n!$ cases. The most advantageous key with optimal length is selected by GA in [9], [10]. Proposed methods usually consider the input image as a set of vectors with fixed length. A set of these constructive vectors are accidentally selected as parents and a single point cross-over is applied to obtain next generation. Mutation is defined as pixel value subtraction from a fixed value [11]-[15].

In this paper we introduce a novel method for image encryption. The proposed method does not need any other encryption scheme and is based only on Genetic Algorithm. It has two distinct steps which are called *modification* and *diffusion*. First pixels positions are dislocated in the modification step to reduce similarity between neighboring pixels and obtain better encrypted result. Chromosomes are binary vectors generated by Local Binary Pattern (LBP) operator. To be reversible the algorithm, two parents (rows/columns) are selected by a semi-random number generator and a predefined key. Then, single point cross-over and mutation operators are applied to generate new individuals. If this procedure increases the fitness function then corresponding rows/columns in the original gray scale image are permuted accordingly. Fitness function of the modification step is defined as average transitions from 0 to 1 in rows and columns of the binary image.

In the diffusion step, values of rearranged pixels are altered to obtain cipher image. Similar to the previous step, chromosomes are binary vectors but they are generated by Bit Plane Slicing (BPS). The gray scale image is decomposed into 8 binary images called bit planes. Again, with the help of a semi-random number generator and a predefined key two bit planes are selected. From each bit plane one parent is chosen in an objective manner. These parents are recombined and permutation is performed to generate two offspring. These individuals are located in the same position as their parents and new bit planes are combined together to make gray scale

Roza Afarin is with the Electrical and Computer Engineering Department, Islamic Azad University, Qazvin Branch, Qazvin, Iran (e-mail: afarin.roza@gmail.com).

Saeed Mozaffari is with the Electrical and Computer Engineering Department, Semnan University, Semnan, Iran (e-mail: mozaffari@semnan.ac.ir).

cipher image. If histogram of the encrypted image becomes more uniform this procedure is accepted and two other parents are selected from ongoing bit planes. Otherwise, two other bit planes are picked up and the above process is repeated.

One of the main factors which differentiate our work from previous efforts is the use of fitness function at each cycle. Although this may increase computational burden, due to binary chromosomes the algorithm converges very fast and encrypted image has high entropy. Proposed methods reviewed in the literature select successive rows/columns of the image as parents and do not utilize fitness function at each generation. But our method opt rows/columns in an objective manner and computes children's fitness at each iteration.

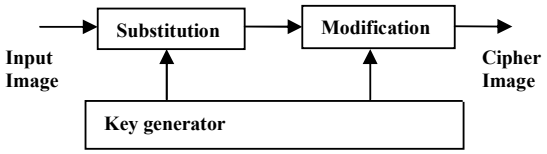


Fig. 1 The proposed scheme

II. Genetic Algorithm

Genetic algorithm, proposed first by John Holland at 1975, is one of the most famous evolutionary algorithms which is inspired from human evolution process. Genetic algorithm (GA) has been used frequently to solve different optimization problems [16].

GA is population based algorithm which starts with an initial population of individual usually randomly generated. These individuals are selected for reproduction based on probability proportional to its fitness. In other words, the fitter the chromosome, the more times it is likely to be selected to reproduce. Then, mutation and crossover operations are applied to the individuals to produce offspring. The rates of mutation and crossover depend on the application. Crossover operator randomly chooses a locus, specific position in the chromosome, and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. Mutation operator randomly alters some of the bits in a chromosome. To overcome local minima problem, mutation is usually used with a very low probability [17].

In this paper, rows/columns of the input image represent initial population. Since size of the image should be constant during encryption process, population is fixed-sized. Thus, only one pair is selected for mating per cycle and two offspring are produced by crossover.

As GA is applied on binary images, chromosomes are binary vectors. The most common crossover operator for binary representation is single point crossover in which two parents are segmented at random positions and new individuals are generated by swapping these segments. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. Mutation lets new trial solutions to be created by making small, random changes in the representation of prior trial solutions. Since binary representation is used, then mutation is achieved by 'flipping' bits at random. For example, the string 00000100 might be mutated in its second position to yield 01000100.

A commonly used rate of mutation is one over the string length [17]. For example, if the chromosome is one hundred bits long, then the mutation rate is set so that each bit has a probability of 0.01 of being flipped. For image encryption, however, a diverse population is more desirable. Therefore, mutation rate is set to higher value of 0.5 and every other bit in the string is flipped.

III. Random Number Generator

Unlike basic form of GA which selects parents, crossover locus, and mutation locations randomly, they are chosen objectively in this paper. This is due to the fact that the algorithm should be reversible and be able to decrypt cipher image.

W used a linear random number generator proposed in [18] which has following parameters: (S, μ, f, U, g) where S is a finite set of states (the state space), μ is a probability distribution on S used to select the initial state (or seed) s_0, $f: S \rightarrow S$ is the transition function, U is the output space, and g: $S \rightarrow U$ is the output function. State of s_i depends to its previous state and the output at step i is $u_i = g(s_i)$. The output values u_0, u_1, u_2, … are the so-called random numbers produced by the following algorithm:

1. Generate the initial state (called the seed) $s_0$according to μ and compute $u_0 = g(s_0)$.
2. Iterate for $i = 1, 2, \ldots$ And compute $s_i = f(s_{i-1})$and $u_i = g(s_i)$.

Having the same parameter μ, the above algorithm generates the same sequence of random numbers. So, parameters μ are used in this paper as key to determine index of rows/columns in the LBP image (needed in substitution phase), bit plane image, and index of bit plane (needed in modification phase). These keys are called key_{sub}, key_{mod} and key_{bit}.

IV. Binary Patterns

As mentioned before, the proposed encryption method is based on GA with binary chromosomes. This is mainly due to simple binary GA operators which accelerates encryption process.

A. Local Binary Pattern

For substitution step which dislocates pixels positions Local Binary Pattern (LBP) operator is used. LBP is one of the most famous and powerful feature descriptors. It has gained increasing attention in many image analyses applications due to its low computational complexity and invariance to monotonic gray-scale changes [19]. The original version of the LBP operator considers a 3×3 rectangular neighboring block around each pixel. These eight neighbors are labeled by thresholding with the central pixel value, weighted with powers of two and then summed to obtain a new value assigned to the central pixel.LBP of the sub-image block

shown in Fig. 2 is calculated as 107 (1+2+8+32+64=107). Fig. 3 shows the input image and its LBP version.

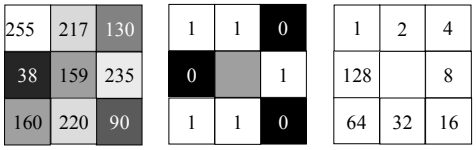


Fig. 2 The LBP operator

(a) (b)

Fig. 3 (a) Input image (b) LBP image

B. Bit Plane Slicing

To change pixels values of the permuted image, generated in the substitution step, and make cipher image with GA, permuted image is decomposed into several binary images called bit planes. Assume we have a gray-scale input image in which each pixel has a gray level value between 0 and 255. Each pixel value can be presented in 8-bit string. So, the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bit-plane 7 for the most significant bit. Fig. 4 illustrates bit plane slicing (BPS), and Fig. 5 shows the various bit planes for the image shown in Fig. 3 (a).

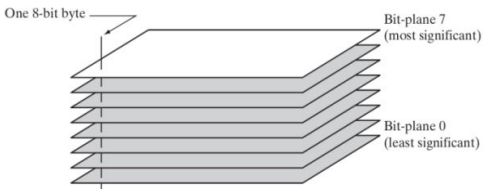


Fig. 4 Bit plane slicing

Fig. 5 Various bit planes for the image Lena

V. Proposed Method

As mentioned before, the proposed method consists of substitution and modification stages. In the former stage position of the pixels are changed while in the latter stage value of pixels are altered. GA with binary chromosomes is used in both stages with single point crossover. Variation between 0 and 1 and histogram uniformity are utilized as cost function for substitution and modification stages, respectively. In the following, details of encryption and decryption steps are presented.

A. Encryption Algorithm

The proposed GA based encryption method needs three keys (key_{sub}, key_{mod} and key_{bit}) as mentioned in Section III. Encryption steps are as follows:

1) Give the plane input image.
2) Generate binary version of input image according to LBP transformation (Section IV *A*).

Substitution stage

3) Generate a semi-random sequence of row/column numbers using key_{sub}. Length of sequence as the same as number of rows and columns.
4) Consider two subsequent rows/columns in the above sequence as parents and apply crossover and mutation operations.
5) If variation between 0 and 1 is increased then perform the same modification on the equivalent gray scale input image. Otherwise, go to the next subsequent pairs of rows/columns.
6) If all rows/columns of the binary LBP image are not processed go to step 4.

Fig. 6 (c) shows result of substitution stage.

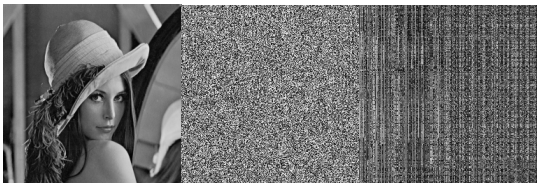
Modification stage

7) Generate binary version of input image according to BPS transformation (Section IV *B*).
8) Generate a semi-random sequence of bitplane pairs using key_{bit}. In each pair, the first element shows one of most-significant bitbplanes (bitplane 8-5) while the second pair indicates one of least-significant bitbplanes (bitplane 1-4).
9) Consider two bitplanes in the above sequence. The reason for choosing one most-significant bitplane and one least-significant bitplane is the fact that least-significant bitplanes are semi-random (Fig. 5) generating more variation in the following steps.
10) Select one row/column from the above bitplanes, using key_{mod} and consider them as parents. Then apply crossover and mutation operations.
11) If histogram of the obtained image becomes more uniform, then perform the same modification on the equivalent gray scale substituted image and go to step 10.
12) Otherwise, go to the next pair of bitplanes and go to step 10.
13) Repeat the above process until the encrypted image is obtained.

Fig. 6 (b) shows the encrypted image.

B. Decryption Algorithm

Having the valid encryption keys (key_{sub}, key_{mod} and key_{bit}), the cipher image can be decrypted. Decryption algorithm is the same as encryption algorithm but in a reverse steps. Fig. 6 (a) shows the encrypted image.



(a) (b) (c)

Fig. 6 (a) Decrypted image (b) Encrypted image (c) Permuted image

VI. Experimental Results

Results of encryption and decryption steps are shown in section 5. In this section quality and security analysis of the proposed method is presented.

A. Quality Analysis

An ideal cipher image should be totally random like a noise pattern. Quality of the encrypted image is usually measured by histogram uniformity, Correlation of adjacent pixels, image entropy.

Image's histogram represents distribution of its gray level values. A random image has uniform histogram. Fig. 7 shows histogram of the original and cipher images.

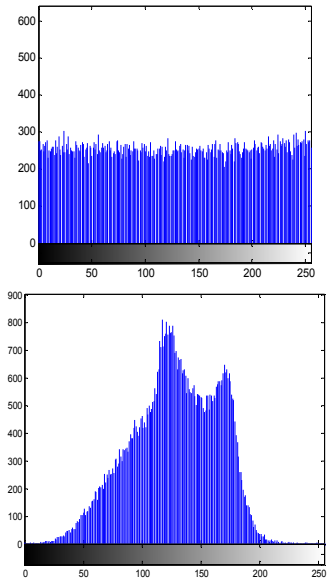


Fig. 7 Histogram of the cipher and original image

Correlation of adjacent pixels in horizontal, vertical and diagonal directions is another criterion for randomness evaluation of the encrypted image. The more randomness the cipher image to be, the less correlation coefficients it has. We select 500 pixels within the cipher image and calculate their correlation coefficients as follows:

$$r_{xy} = \frac{|cov(x,y)|}{\sqrt{D(x)} \times \sqrt{D(y)}} \quad (1)$$

$$\mathrm{Cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}((x_i - E(x))(y_i - E(y)) \quad (2)$$

$$\mathrm{E}(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (3)$$

$$\mathrm{D}(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \quad (4)$$

In the above relation, x and y are the gray levels in two adjacent pixel of image.

Information entropy is the most important feature of randomness. If m be number of gray levels and $p(m_i)$be the probability of m^{th} gray level, information entropy is:

$$\mathrm{I}(m) = \sum_{i=0}^{2^n-1} p(m_i)\log_2\left(\frac{1}{p(m_i)}\right) \quad (5)$$

For a random gray scale image with 256 levels (2^8), entropy is 8 bits. Table I shows correlation coefficient and entropy of the benchmark image. Comparison with other methods is presented in Table II.

TABLE I
Correlation and Entropy of Different Image

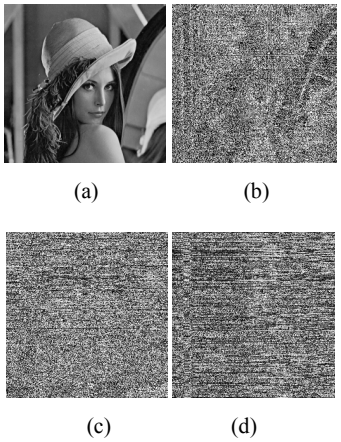
Criterion	Barbara	Lena	Baboon
correlation	-0.00017	-0.00058	0.00088
entropy	7.9932	7.9968	7.9971

TABLE II
Comparison with Other Methods

Method	Correlation	Entropy
[5]	0.0024	7.9660
[9]	-0.0648	7.9793
[10]	0.0027	7.9717

B. Security Analysis

The proposed cryptosystem has three keys for substitution (key_{sub}), modification (key_{mod}), and bitplane (key_{bit}) selection. To evaluate keys sensitivity, the plain image is first encrypted with chosen keys. Then decryption process is performed by different keys with a tiny alteration compared to the original ones. Fig. 8 shows decrypted image with correct keys Fig. 8 (a), with wrong key_{sub} Fig. 8 (b), with wrong key_{mod} Fig. 8 (c), and with wrong key_{bit} Fig. 8 (d).



(a) (b)

(c) (d)

Fig. 8 (a) Decrypted image with correct keys (b) with wrong key_{sub} (c) with wrong key_{mod} (d) with wrong key_{bit}

VII. CONCLUSION

This paper presents a new image encryption method based on genetic algorithm. To accelerate this process, the plane image is first converted into distinct binary images using Local Binary Pattern (LBP) and Bit Plane Slicing (BPS). Experimental results show that our method yields high random cipher image measured by histogram uniformity, correlation of adjacent pixels, and image entropy criteria. Experiments confirm that our approach is sensitive to the keys and small distortion in them does not recover the original image at all.

REFERENCES

[1] ShigueLian, *Multimedia Content Encryption*, CRC press, AUERBACH publication, 2009.
[2] OdedGoldreich, "Cryptography and cryptographic protocols", Distributed Computing, vol. 16, pp. 177-199, 2003.
[3] Komal D Patel, SonalBelani, "Image encryption using different techniques: a review", International Journal of Emerging Technology and Advanced Engineering, vol. 1, pp. 30-34, 2011.
[4] Monish Sharma, Manoj Kumar Kowar, "Image encryption techniques using chaotic schemes: a Review", International Journal of Engineering Science and Technology, vol. 2, pp. 2359-2363, 2010.
[5] RasulEnayatifar, Abdul Hanan Abdullah, "Image security via Genetic algorithm", International Conference on Computer and Software Modeling,vol. 14, pp. 221-226, 2011.
[6] Abdul Hanan Abdullah, Malrey Lee, "A hybrid Genetic algorithm and Chaotic function model for image encryption", International Journal of Electronics and Communication, in press.
[7] V.Srikanth, UditAsati, Viswajit Natarajan, T.Pavan Kumar, Teja Mullapudi, N.CH.S.N.Iyengar, "Bit-Level encryption of image using Genetic algorithm", International Journal of Computing Science and Communication Technologies, vol. 3, pp. 546-550, 2010.
[8] Ibrahim S I Abduhaiba, Maaly A S Hassan, "Image encryption using differential evolution approach in frequency domain", Signal & Image Processing, vol. 2, pp. 51-69, 2011.
[9] Aleksey Gorodilov, Vladimir Morozenko, "Genetic algorithm for finding the key's length and cryptanalysis of the permutation cipher", International Journal Information Theories & Applications, vol. 15, pp. 94-99, 2008.
[10] Sandeep Bhowmik, SriyankarAcharyya, "Image cryptography: the Genetic algorithm approach", IEEE, vol. 3, pp. 223-227, 2011.
[11] Mohammed A.F. Al-Husainy, "Image encryption using Genetic algorithm", Information Technology Journal, vol. 3, pp. 516-519. 2006.
[12] Anil Kumar, M. K. Ghose, "Overview of information security using Genetic algorithm and Chaos", Information Security Journal: A Global Perspective, vol. 18, pp. 306-315, 2012.
[13] Ankita Agarwal, "Secret encryption algorithm using Genetic algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, pp. 216-218, 2012.
[14] Dr. Mohammed Abbas Fadhil Al-Husainy, "Genetic algorithm: tool to encrypt image", International Journal of Advanced Research in Computer Science,vol. 3, pp. 36-42,2012.
[15] Gove NitinkumarRajendra, Bedi Rajneesh Kaur, "A new approach for data encryption using Genetic algorithm and Brain Mu Waves", International Journal of Scientific and Engineering Research, vol. 2, pp. 1-4, 2011.
[16] K.S.Tang, K.F.Man, S.Kwong, "Genetic algorithm and their applications", IEEE Signal Processing Magazine, pp. 22-37, 1996.
[17] Mitchell Melanie, An Introduction to Genetic Algorithm, MIT press, 1999.
[18] Michael D. Vose, "A linear algorithm for generating random number with a given distribution", IEEE Transaction on Software Engineering, vol. 17, pp. 972-975, 1991.
[19] TimoAhonen, AbdenourHadid, MattiPietikainen, "Face recognition with Local Binary Pattern", Springer, pp. 469-481, 2004.