Journal of Theoretical and Applied Information Technology
15th June 2017. Vol.95. No 11
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SUGGESTED SCHEME FOR IMAGE CRYPTOGRAPHY EMPLOYING GENETIC ALGORITHM

[1]MOHAMMED A. F. AL-HUSAINY, [2]HAMZA A. A. AL-SEWADI

[1]Middle East University, Faculty of IT, Computer Science Department, Amman, Jordan

[2]Middle East University, Faculty of IT, Computer Information System Department, Amman, Jordan

E-mail: [1]dralhusainy@gmail.com, [1]mal-husainy@meu.edu.jo, [2]alsewadi@hotmail.com

ABSTRACT

The extensive use of digital information storage and transfer over communication channels nowadays have raised tremendous hazards. Basically, it has exposed the information confidentiality, integrity, authenticity, copyright, and ownership protection to the danger of being breached. Of the most important concerns is the security of digital images, hence they either be protected by the embedding of watermarks if they are to be publicly visible or by converting them into an unintelligible cipher form. This conversion can be achieved by many existing encryption algorithms with proper secret keys, however, due to the increased computation efficiency, new and stronger algorithms are continually sought. In this paper, a novel image encryption scheme that is based Genetic Algorithms is developed and implemented to convert the image into cipher-image. This approach is made possible by exploiting the Genetic Algorithm (GA) properties, namely crossover and mutation that fulfill Shannon's principle of information diffusion and confusion. Obtained experimental results of the proposed scheme has succeeded in protecting digital images against noise and attacks, and is promising as compared with currently used standard cryptographic algorithms, such as Data Encryption Standards (DES) and Advanced Encryption Standard (AES).

Keywords: *Image security, Genetic Algorithm, Secret key cryptosystems, Crossover, and Mutation.*

1. INTRODUCTION

For almost all applications where data security is on the stack, cryptography techniques became mandatory. Such applications involve storage and transfer over the internet for personal information, enterprise data, civilian and military information, etc. Security in any applications is crucial to provide leakage prevention, integrity, authenticity, and accuracy. Cryptography in general means converting clear and intelligible information (plaintext) into a vague and unintelligible form (ciphertext) to everybody except for the intended parties. Hence, any data file, being a text, audio, image or video is first converted into digital form prior, to be treated by one of the cryptographic algorithms in order to encode it into ciphertext. A variety of cryptographic algorithms are available utilizing various mathematical and transformational tools to achieve the encoding goal [1, 2].

Cryptographic encoding systems must be reversible, i.e., encoded data in the encryption process must be recoverable at decryption process. Data encoding is performed using a variety of cryptographic algorithms that implement certain key(s). Currently, cryptographic processes are achieved by two approaches or types of algorithms; symmetric algorithms, where the same key is used for both encryption and decryption processes and asymmetric algorithms that use different (but related) keys for the two processes [3]. Examples of the former type are the widely used Data Encryption Standard (DES) and Advanced Encryption Standard (AES), while a good example of the latter type is the RSA (Rivest, Shamir, and Adleman1) algorithm [4]. Each of the two approaches has its own advantages and disadvantages. However, all work in the finite field and use modular mathematical operations in achieving their goals. Moreover, increasing interests in the use of Genetic Algorithms (GAs) in cryptographic algorithms have proved applicable and efficient as alternative tool for encryption and decryption algorithms [5, 6, 7].

GA is usually useful for heuristic search problems that look for optimum solutions based on natural selection and genetic operations that means weak and useless traits can be eliminated while new generations with acceptable fitness are reproduced. This behavior is achieved due to the three GA properties, namely reproduction, crossover, and mutation. A new population results after the application of some stochastic processes [5] which either have transformed members or newly generated members.

The crossover and mutation properties of GA may be considered to correspond to the transposition and substitution processes of cryptographic algorithms, respectively. This paper proposes a new and efficient design for the image cryptographic algorithm scheme by exploiting these properties of Genetic Algorithms. This scheme will be referred to hereafter as GAICS (which stands for Genetic Algorithm Image Cryptographic Scheme). After the brief introduction in section 1, few of the most recent related work is listed in section 2, followed by a description of the proposed encryption/decryption scheme. Then experimental results are included in section 4 together with their discussions. Finally, section 5 concludes the work.

2. RELATED WORK

Although so many image encryption algorithms based on GA were proposed and investigated, only recent related work will be listed here.

The early use of GA for data cryptography was first suggested by Spillman in 1993 [6] using only the substitution for cryptanalysis to discover the key simple substitution cipher. Then, almost at same time Mathew [7] suggested the use of transposition for cryptanalysis.

Other versions of GAs involved chaotic maps proposed by Li and Zhang in 2002 [8, 9]. A novel image encryption algorithm called BRIE (Bit Recirculation Image Encryption) was proposed, but it was not secure enough against known/chosen plain-image. Then scattering according to some chaotic function proposed by Li et al. in 2002 [10]. Also, Bao et al. in 2002 [11] proposed Magic cube transformation, and Guibin et. al. in 2003 [12] used Affine transformation. It shows that for digital image scrambling, give better results than geometric transformations. Moreover, Zhao in 2003 [13] suggested two-dimensional baker map pixel manipulation. However, all these schemes were not efficient in separating the secret-key from the algorithm.

Kumar and Rajpal in 2004 [14] implemented GA crossover operator with a pseudorandom sequence generator for image encryption. Then Kumar et. al. in 2005 [15] incorporated mutation after encryption to enhance the security, using the concept of GA with NLFFSR (Non Linear Forward Feedback Shift Register) transforming the image into completely disordered data. They claim the achievement of high throughput rate that makes it suitable for real time application, however, the security of the technique was not thoroughly proved.

Geetha et al. in 2006 [16] suggested an effective and adaptive genetic algorithm for audio stego-analysis resulting in flexible steganography implementation. It relies on audio quality metrics and the construction of a two-class classifier. It involves fuzzy logic, probabilistic reasoning, and artificial neural network and did not prove suitable for still images and videos.

Symmetrical blocks image ciphering system was proposed by Tragha et al. in 2006 [17] which allows choosing the size of the blocks and the length of the key and it has been experimented with for most popular symmetric systems, such as IDEA, AES, and DES.

Al-Husainy in 2006 [18] proposed a GA-based Image Encryption using the mutation and crossover concept. After splitting the image data into a set of vectors having a same number of bytes. Different values are extracted from each vector to be used as parameters to implement crossover and mutation operations on the bytes of the vector. But this technique needs to keep some bytes in the data vectors without change to be used as key in the encryption and decryption operations.

Still image and video encryption using GAs with the help of physical model were proposed by Bhandari et al. in 2009 [19]. Their design was based on FPGA (Field programmable Gate Arrays) concept rather than DSP (Digital Signal Processing) for signal and image processing, such model has the inherent speed advantage, but practical limitations due to the hardware requirement.

Recently, image encryption based on bivariate polynomials has been examined and self-regressive function was used in image encryption technique to study the chaotic effect as suggested by Bhowmik and Acharyya in 2011 [20].

Enayatifar and Abdullah in 2011 [21] mixed GA with the chaotic function logistic map to achieve image encryption. They claim the achievement of high efficiency and stability.

Afarin et al. in 2013 [22] describes a method which depends on two phases; substitution and modification. The randomness of these phases measured three factors; histogram analysis, entropy, and coefficient correlation. They claim that their technique is fast enough and effective for image encryption.

Kumar et al. in 2014 [23] reviewed and gave a concise description of security criteria for many data encryption techniques of digital multimedia including genetic algorithms (GAs).

Hence, more research efforts are still required to achieve image encryption techniques that provide, faster encryption/decryption processes, high image quality, and high security against attackers. This paper suggests and tests an efficient and highly secure algorithm for image encryption base on genetic algorithm. It utilizes both crossover and mutation properties of GA to achieve Shannon's security criteria requirement of diffusion and confusion, as will be seen in the next sections.

3. GENETIC ALGORITHM IMAGE CRYPTOGRAPHIC SCHEME

A Genetic Algorithm Image Cryptographic Scheme is proposed here. It exploits the two main operations of genetic algorithms (namely crossover and mutation) in order to perform the two main operations of contents transposition and substitution that are essential in any good encryption algorithm for image security. Figure 1 outlines the block diagram for the proposed image encryption algorithm in this paper. Any of the two users in this model shall be able to encrypt an image using a shared secret encryption/decryption key combination. This key combination consists of two parts; the image block dimension D and a carefully chosen secret key. These two parts are agreed upon between the users beforehand and exchanged over a secure channel.

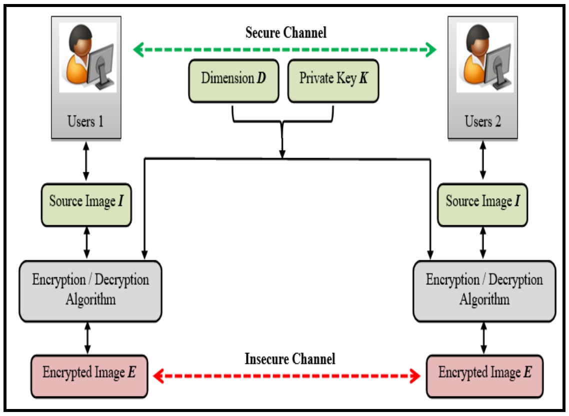


Figure 1: The general model for the Genetic Algorithm Image Cryptography Scheme

3.1 Definitions

In order to have a good understanding of the proposed encryption algorithm, related terminologies and definitions are summarized below

- **Source Image (*I*):** bitmap colored image pixels, each pixel represents three color components; Red (R), Green (G) and Blue (B). Each color is represented by one byte having a value in the range 0 to 255. The algorithm treats the image file as a collection of bytes.
- **Source Image Size (*ISize*):** the image size ***I*** (in byte) is equal to (*Width*×*Height*×*Palette*). Where the *Palette* = 3 for the RGB image.
- **Encrypted Image (*E*):** bitmap colored image pixels resulted from the application of the proposed encryption algorithm, i.e. by performing the crossover and mutation operations of the genetic algorithm on the bytes of the source image ***I***.
- **Key (*K*):** the chosen secret key to be used for the encryption algorithm. It may be any type of digital files such as text, audio, Image, or video. The algorithm treats the user key as a collection of bits. The users should choose a key that contains as much as possible random numbers to ensure that the parameters in the key block ***KBlock*** in figure 3 be as different as possible in each ***KBlock*** used.
- **Key Size (*KSize*):** size of the chosen secret key (in byte). The minimum size of this key shall be **(1 byte**).
- **Block Dimension (*D*):** is an integer number, greater than 1, used by the algorithm to determine:

- The image block dimension (the number of rows and columns) for each block (segment) of the image ***I***. Each Image Block ***IBlock*** is a square matrix of **(*D*×*D*)** dimension and ***IBlock*(*b*, *i*, *j*)** is an element in the image block ***I***, where ***b*** is the block number, ***i*** is the row number, and ***j*** is the column number as shown in the example in figure 2, when **D** = 8.

D×D	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0								
1								
2								
3								
4								
5								
6								
7								

Figure 2: Example of an image block of dimension D

- The total number of bits that are needed to be read from the secret key is ***K***, which is used to set the parameters in the key block ***KBlock***. These parameters are used in the implementation of the crossover and mutation operations of the genetic algorithm used in the encryption algorithm for encrypting each block of the image ***I***. Figure 3 shows the structure of the key block ***KBlock***. The number of bits *N* related to the image block and used in the key block structure is calculated by equation 1.

$$D = 2^{N \text{ bits}} \qquad (1)$$

Genetic Algorithm Operations

Crossover Operation Parameters (bits)			***Mutation Operation Parameters (bits)***			
SV_C	VL_C	SRC_C	SV_M	VL_M	MV_M	SRC_M
N	*N*	1	*N*	*N*	*N*	1

Note: The subscript letters ***C*** refers to Crossover operation and ***M*** refers to Mutation operation.

Figure 3: The structure of the key block ***KBlock****.*

The definitions of the parameters shown in figure 3 are defined as follows

- SV_C: Start Value (i.e., seed) for the random generation algorithm used during the Crossover operation. This will help to implement the crossover operation on different vectors within each image block ***IBlock***.
- VL_C: Vector Length used in Crossover operation. VL_C values range from **1** to ***D***. Use a different length of the vector in each image block ***IBlock*** adds much more difficulties against the attackers. Where it become the implementation of the crossover operation completely different from image block to another.
- SRC_C: a value represents whether the first implementation of the Crossover operation, in each image block ***IBlock***, Starts with either Row or Column. SRC_C value is either **0** or **1**. An additional difficulty against attackers when we use a different sequence of rows and columns in the implementation of the crossover operation in each image block ***IBlock***.
- SV_M: Start Value (i.e., seed) for the random generation algorithm used during the Mutation operation. This will help to implement the mutation operation on different vectors within each image block ***IBlock***.
- VL_M: Vector Length used in Mutation operation. VL_M values range from **1** to ***D***. Use a different length of the vector in each image block ***IBlock*** adds much more difficulties against the attackers. Where it become the implementation of the mutation operation completely different from image block to another.
- MV_M: Mutation Value used in Mutation operation. This value plays a major role in the implementation of the ***XOR*** operation and the effect on the values of bytes in each vector.
- SRC_M: a value represents whether the first implementation of the Mutation operation, in each image block ***IBlock***, Starts with either Row or Column. SRC_M value is either **0** or **1**. An additional difficulty against attackers when we use a different sequence of rows and columns in the implementation of the mutation operation in each image block ***IBlock***.

3.2 The GAICS Algorithm Processes

GAICS algorithm is designed to perform image encryption and decryption. For encryption process it involves the following steps:

Step1: Read the value of the block dimension ***D*** which is supplied by the user. Calculate the number of bits ***N*** from ***D*** using equation 1.

Step2: Read the secret key ***K*** of size ***KSize*** supplied by the user.

Step3: Read the source image ***I*** of size ***ISize***. Divide the image ***I*** into a set of square blocks (two dimensional matrices) of dimension ***D***. Where the number of blocks of the image ***I*** is calculated by equation 2.

$$\boldsymbol{NoOfBlocks = ISize / (D \times D)} \qquad (2)$$

Step4: For each image block, from ***b* = 0** to ***NoOfBlocks*-1**, perform the following:

(i) Read the required number of bits from the key ***K*** and set the parameters of the key block ***KBlock***.

(ii) Implement the following genetic algorithm operations:

<u>Crossover Operation:</u>

1. Use ***SV***$\boldsymbol{SV_C}$ as a seed to generate random numbers in the crossover operation for the current ***IBlock***.
2. For each element ***IBlock*(*b*, *i*, *j*)**
   - Choose another element in ***IBlock*** by randomly selecting row ***r*** and column ***c***.
   - Exchange the elements of the two vectors (of length $\boldsymbol{VL_C}$) at the two elements ***IBlock*(*b*, *i*, *j*)** and ***IBlock*(*b*, *r*, *c*)** either as a row or as a column based on the value of $\boldsymbol{SRC_C}$. The ***IBlock*** is treated as a circular block in this operation.
   - Change the value of $\boldsymbol{SRC_C}$ to its opposite value.

<u>Mutation Operation:</u>

1. Use $\boldsymbol{SV_M}$ as a seed to generate random numbers in the mutation operation for the current ***IBlock***.
2. For each element ***IBlock*(*b*, *i*, *j*)**
   - Change the value of the elements of the vectors (of length $\boldsymbol{VL_M}$) at the element ***IBlock*(*b*, *i*, *j*)** either as a row or as a column based on the value of $\boldsymbol{SRC_M}$. The ***IBlock*** is treated as a circular block in this operation. The new value of each element in the vector is calculated by applying an XOR operation using the formula of equation 3.

$$\boldsymbol{NewValue} = \boldsymbol{OldValue} \text{ XOR } (\boldsymbol{MV_M} \times \boldsymbol{VL_M}) \qquad (3)$$

   - Change the value of $\boldsymbol{SRC_M}$ to its opposite value.

Step5: Use the resulted encrypted image blocks obtained from step 4, reconstruct an image ***E*** which is the cipher of image ***I***.

On the receiver side, for the recovery of the original image ***I***, decryption process is implemented. The same operations (crossover and mutation) apply to the encrypted image ***E*** using the same key block but in reverse order.

4. RESULTS AND DISCUSSION

In this section, the proposed scheme will be examined in its encryption and decryption processes for some selected images of different sizes, colors, and natures. Then apply some standard metrics such as Normalized Mean Absolute Error (NMAE), Peak Signal to Noise Ratio (PSNR) using the formulas given in equations 4 and 5. Moreover, the encryption computation time is measured, too.

$$NMAE = \frac{\sum_{k=0}^{Sizeofimage-1} \left| I(k) - E(k) \right|}{Sizeofimage} \times 100 \qquad (4)$$

$$PSNR = 10.\log_{10}\left(\frac{Max_I{}^2}{NMAE}\right) \qquad (5)$$

Where: Max_I is the maximum possible pixel value of the image *I*.

To show the enhancements gained by the proposed scheme, all these measurements are compared with DES and AES encryption technique after being programmed and run on the same computing environment.

Three different images were selected; Balloon, Nemo-Dory, and a Buildings were selected with sizes 128×128 pixels, 256×192 pixels, and 256×170 pixels, respectively as shown in figure 4. They are selected in order to evaluate the suitability of GAICS algorithm for image encryption.

The resulting cipher images of the selected images of figure 4 using the proposed GAICS algorithm are illustrated in figure 5. For comparison with other widely and practically used encryption techniques, figure 5 also illustrates the results of encrypting the same images with Data Encryption Standard algorithm (DES) and the more powerful Advanced Encryption Standard algorithm (AES). The same computing environment is used for running all these algorithms.

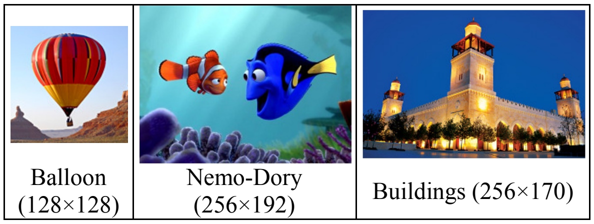


Figure 4: Sample images used for GAICS experiments

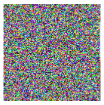
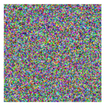
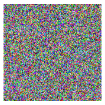
Encryption Algorithm	Balloon	Nemo-Dory	Buildings
GAICS algorithm (Proposed)			
DES algorithm			
AES algorithm			

Figure 5: Encrypted images by GAICS, DES, and AES algorithms

To measure the cryptographic efficiency of the proposed GAICS algorithm and to be able to compare with other established cryptographic techniques, the Normalized Mean Absolute Error (NMAE), Peak Signal Noise Ratio (PSNR) were evaluated for the selected image samples of figure 4, using equations 4 and 5, then the results are listed in table 1-a and 1-b. The computation time is also measured and displayed in figure 1-c. It is included in order to give a notion of the time complexity effect of the new GAICS algorithm as compared with DES and AES algorithms.

Table 1: Comparison of GAICS algorithm with DES and AES algorithms for PSNR, NMAE and Computation time

(a) PSNR

Image	PSNR (dB)		
	GAICS	DES	AES
Balloon	6.10	6.39	6.84
Nemo-Dory	7.20	7.19	7.24
Buildings	6.93	6.96	6.93

(b) NMAE

Image	NMAE (%)		
	GAICS	DES	AES
Balloon	71.08	66.19	65.99
Nemo-Dory	71.15	71.29	70.85
Buildings	95.13	94.82	95.17

(c) Computation time

Image	Time (Sec)		
	GAICS	DES	AES
Balloon	1.17	0.89	0.82
Nemo-Dory	1.45	0.22	0.24
Buildings	0.39	0.21	0.20

Moreover, for the same selected image samples, the source images histograms and those of the encrypted images are depicted in figure 6 for the proposed GAICS algorithm together with those for DES and AES algorithms. Histogram comparison showed that proposed algorithm histogram gives better distribution of colors than the other algorithms.

It must be added that the block dimension *D* of the image blocks in GAICS algorithm plays an important role in the produced encrypted image. Experimenting on this parameter shows this effect clearly. An example is shown in figure 7, where image encryption is performed for various values of D, namely D = 2, 8, 64, and 384 pixels. This figure illustrates that the encryption efficiency improves considerably with the increase in the block dimension D.

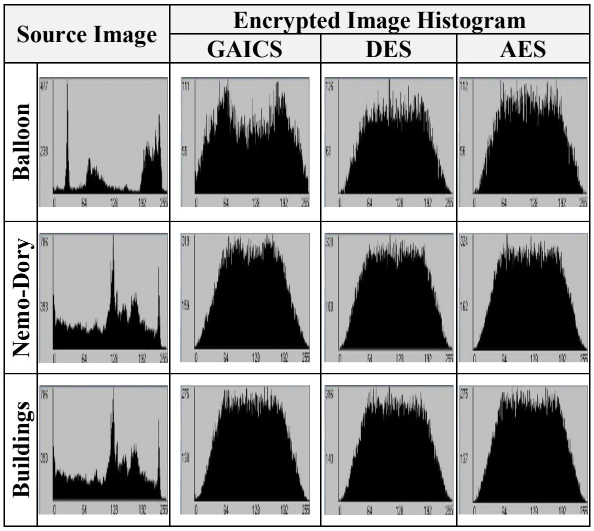


Figure 6: Source and encrypted images histograms using GAICS, DES and AES algorithms

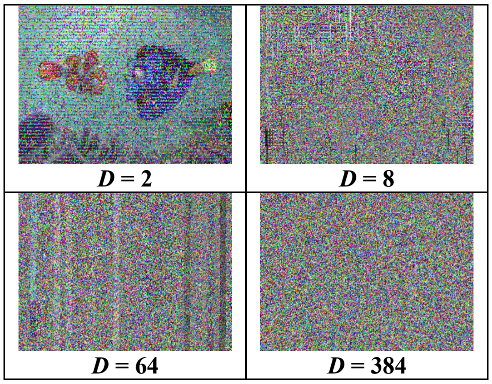


Figure 7: Block dimension D effect on encryption for Nemo-Dory image.

Comparison of the obtained visual observations of the image encryption distribution of figure 5 and the histograms of figure 6 for the proposed algorithm with those for DES, and AES algorithms have shown encouraging results. Then numerically obtained values of NMAE, and PSNR, and measured computation time for GAICS, DES, and AES illustrated in tables 1, shows that the proposed scheme gives comparatively good protection as shown. Although, the use of GA in various fields of data processing usually takes much more time, its use in the proposed GAICS for image encryption succeeded to achieve good results.

5. CONCLUSION

The implementation of the proposed image encryption scheme that involved GA has resulted in a promising approach as compared with existing well established cryptographic techniques currently employed for image protection. This is attributed to the large key space used, comparatively uniform histogram generated, and satisfaction of Shannon's security concept through crossover and mutations features of GA which are utilized to achieve the substitution and transposition operations essentially required for data security. Obtained experimental results of the proposed scheme, GAICS algorithm has shown remarkable success in employing the GA operations to produce excellent diffusion and confusion effects in the source image. Furthermore, GAICS causes positive changes in the colors histogram of the image, all that is done in an acceptable time period.

6. FUTURE WORKS

As the efficiency of computation machines is steadily thriving, image encryption processes are needed to be continuously enhanced, and require wider bandwidth. Therefore, more research work will be extended seeking to enhance the encryption time. However, the comparable results of table 1, suggests that GAICS algorithm is a secure approach and can be recommended for applications involving image protection in the information security field.

REFRENCES:

[1] N. Koblitz, "A Course in Number Theory and Cryptography", Springer-Verlag, New York, Inc., 1994.

[2] A. J. Menzes, C. Paul, V. Van Dorschot, and A. A. Vanstone, "Handbook of Applied Cryptography", CRS Press 5th Printing; 2001.

[3] B. Schneier, "Applied Cryptography: protocols, algorithms and source code in C", John Wiley & Sons, 1996.

[4] W. Stallings, "Cryptography and Network Security, principles and practice", Pearson Education, Inc., publishing as Prentice Hall, 5th Ed., 2011.

[5] R. Jhingran, V. Thada, and S. Dhaka, "A Study on Cryptography using Genetic Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 118 – No.20, May 2015.

[6] R. Spillman, M. Janssen, B. Nelson, and N. Kepner, "Use of Genetic Algorithm in Cryptanalysis of Simple Substitution Cipher" Cryptologia, Vol.17, No.4, 1993, PP 367-377.

[7] J. Mathews, “The Use of Genetic Algorithm in Cryptanalysis”, Cryptologia, Vol. 17, 2010, PP 187-201.

[8] S. Li, and X. Zhang, ”Cryptanalysis of Chaotic Image encryption Method”, Proceeding of the 2002 IEEE Int. conference on circuits and systems (ISCAS 2002), Scotland, Arizona, Vol. 2, 2002, PP 708-711.

[9] S. Li, and X. Zhang, “On the Security if an Image Encryption method”, Proceeding of the 2002 IEEE Int. conference on circuits and systems (ISCAS 2002), Scotland, Arizona, Vol. 2, PP925-928.

[10] C. G. Li, H. Zheng-Zhi, and Z. Hao-Ran, “Image Encryption Techniques: A Survey”, Journal of Computer Research and Development, Vol. 39, No. 10, 2002, pp. 1317-1324.

[11] G. J. Bao, Ji. Shi-ming, and Shen Jian-bin, “Magic Cube Transformation and Its Application in Digital Image Encryption”, Computer Applications, Vol. 22, No. 11, pp. 23-25, Nov. (2002).

[12] Z. Guibin, C. Changxiu, and H. Zhongyu, “An Image Scrambling and Encryption Algorithm Based on Affine Transformation”, Journal of Computer-Aided Design & Computer Graphics, Vol. 15, No. 6, pp. 711-715, June. 2003.

[13] X. Zhao, “Digital Image Scrambling Based on the Baker’s Transformation”, Journal of Northwest Normal University (Natural Science), Vol. 39, No. 2, Feb. 2003, PP 26-29

[14] A. Kumar, and N. Rajpal, “Application of Genetic Algorithm in the Field of Steganography”, in Journal of Information Technology, Vol. 2, No.1, 2004, PP12-15.

[15] A. Kumar, N. Rajpal, and A. Tayal, “New signal security system for multimedia data transmission using genetic algorithms”. NCC 2005, January 20-28, IIT Kharagpur, PP579–583.

[16] S. Geetha, S. S. Sindhu, and A. Kennan., “An active rule based approach to audio steganalysis with a genetic algorithm” at IEEE, 2006.

[17] A. Tragha, F. Omary, and A. Kriouile,”Genetic Algorithm Inspired by Cryptography”, Association for the Advancement of Modeling & Simulation Techniques in Enterprises (A.M.S.E), Series D: Computer Science and Statistics, 15- November 2007.

[18] M. Al-Husainy, “Image encryption using genetic algorithm”. Information Technology Journal, Vol. 5, No.3, 2006, PP516–519.

[19] S. U. Bhandari, S. Subbaraman, S. S. Pujari, and R. Mahajan.,”Real time video processing on FPGA using on the fly partial configuration” at IEEE, 2009.

[20] S. Bhowmik, and S. Acharyya,” Image cryptography: The genetic algorithm approach” Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on June 2011, vol.2, PP. 223 – 227

[21] R. Enayatifar, and A. H. Abdullah, “Image Security via Genetic Algorithm”, International Conference on Computer and Software Modeling IPCSIT, Singapore, Vol. 14, 2011, PP198-203.

[22] R. Afarin, and S. Mozaffari, “Image encryption using genetic algorithm and binary patterns”, 10th International conference on Information Security and Cryptology (ISCISC), 2013.

[23] M. Kumar, A. Aggarwal, and A. Garg, “A Review on Various Digital Image Encryption Techniques and Security Criteria”, International Journal of Computer Applications, Vol. 96, No. 13, June 2014.