ISSN 2321 3361 © 2021 IJESC

Research Article Volume 11 Issue No.04

An Optimal Digital Image Encryption System using The Genetic Algorithm

P. A.Wumnaya[1], P. A. Agbedemnab[2], M. A. Agebure[3]
School of Computing and Information Science, University of Technology and Applied Sciences, Navrongo, Ghana

Abstract:
The advancement in technology has made it very easy to gain access to an avalanche of items, both tangible and intangible. As this advancement takes on different strides, the ease with which the privacy of individuals and institutions at large is exploited has exponentially increased. This has injected a serious threat to the lives and activities of people and institutions with an amalgamated effect. Image Encryption is just one of the numerous strategies that have been adopted in order to return sanity and security in the digital ecosystem, however little the success has been in the field. Nonetheless, some extremely and talented dubious fellows manage to by-pass various encryption methods that are used to encrypt images. This paper presents an optimal encryption technique for encrypting images of various types and sizes. This approach of encryption is achieved by leveraging on the chaotic nature of the Genetic Algorithm (GA) and a programming tool, which is Visual Basic in Visual Studio 2013. The algorithm employed is very simple without overhead cost but produced very chaotic results thereby making this technique an optimal one.

1. INTRODUCTION

Over the past decade, technology has progressed in every field imaginable. From nanotechnology to remote-controlled contraceptive chips, technology really has expanded and changed at a rate faster than ever (Germanovic, 2014). This evolution has not only affected the professional lives of people but also the social lives and all aspects of people's lives. The presence of the internet and other forms of communication networks has made the transfer of information very easy. This has reached a very critical point that the presence of the internet in one's life has now become a need. One of the forms in which data and information are transferred over networks in our world today is digital images. In fact, almost all the users of the internet share digital images in one way or another. One form of information that is most vulnerable to the cyber-crimes are digital images. In the digital world, nowadays, the security of digital images becomes more and more important since the communications of digital products over open network occur more frequently. Furthermore, special and reliable security for storage and transmission of digital images is needed in many applications, such as pay-tv, medical imaging systems, military image communications and confidential video conferencing. In order to fulfil such a task, many image encryption methods have been proposed, but some of them have been known to be insecure, (Li & Zheng, 2002). Image encryption is a process in which a finite set of instructions, called an algorithm, is used to convert an original image into a cipher image or into an encrypted form, (Jim & Merkow, 2014). The encrypted image is then sent over a medium towards the destination. At the receiving end, the image is then decrypted using the required key. For Image encryption to occur, the pixels of the original image must be obtained. A strong encrypted image should be created such that it cannot be hacked easily. There should be a faster encryption time. The image obtained after decryption should be in perfect condition as it was before encryption. Genetic algorithms (GAs) are search methods based on principles of natural selection and genetics, (Kumari & Goyal , 2016). The GA goes through the following cycle: Evaluate, select, mate, and mutate until some stopping criteria are reached. Reproduction and crossover together give genetic algorithms most of their searching power. Digital image security has gained, in recent years, a lot of attention (Al-Husainy, 2006). This is because, digital images are now stored in memories and also sent through communication networks. With the advent of image encryption, users and organizations have been able to secure the digital images they share over networks. Many image encryption methods have been proposed to keep digital images secure but many of the algorithms that are implemented to achieve this encryption do not provide optimal results. This has led to many hackers and cyber criminals finding backdoors and ways to crack these encryptionsystems. It is in this light, that this paper seeks to implement the GA on images for encryption purposes in a simple but robust manner to achieve optimal results. The rest of the paper is organised as follows: Section 2 reviews literature on the subject matter relating to encryption and GA. The proposed scheme is presented in Section 3 with its results presented in Section 4 in detail showing the functionality of each button in the developed application. The paper is concluded in Section 5.

2. LITERATURE REVIEW

Image encryption schemes have been increasingly studied to meet the demand for real-time secure image transmission over the Internet and through wireless networks. Traditional image encryption algorithms such as data encryption standard (DES), has the weakness of low-level efficiency when the image is large (Sakthidasan Sankaran &Santhosh Krishna, 2011). Recent researches of image encryption algorithms have been increasingly based on chaotic systems, but the drawbacks of small key space and weak security in one dimensional chaotic cryptosystems are obvious (Sakthidasan Sankaran &Santhosh Krishna, 2011). Zhang & Karim, (1999) proposed a method to encrypt colour images using existing optical encryption systems for gray-scale images. The colour images were converted to their indexed image formats before they were encoded. In the encoding subsystem, the image was encoded to stationary white noise with two random phase masks, one in the input plane and the other in the Fourier plane. At the decryption end, the colour images were recovered by

converting the decrypted indexed images back to their RGB (Red-Green-Blue) formats. This method of image encryption is multichannel in nature; however, the proposed single-channel colour image encryption method is more compact and robust than the multichannel methods. Chang, Hwang, & Chen, (2001) used one of the popular image compression techniques, vector quantization to design an efficient cryptosystem for images. The scheme is based on vector quantization (VQ), cryptography, and other number theorems. In VQ, the images are first decomposed into vectors and then sequentially encoded vector by vector before the traditional cryptosystems from commercial applications can be used. However, the approach proposed in this paper simplifies these procedures into an algorithm that produces optimal results using only the GA. Aloha & Kehar, (2013) also proposed a technique to encrypt an image for secure image transmission. In that paper, the digital signature of the original image was added to the encoded version of the original image. The encoding was done by using an error control code, such as a Bose-Chaudhuri Hochquenghem (BCH) code so that at the receiver end, after the decryption of the image, the digital signature can be used to verify the authenticity of the image.
Abdullah, Enayatifar, & Lee, (2012) combined GA and a chaotic function to form what was referred to as a hybrid model for image encryption. In the method, a number of encrypted images were constructed using the plain image and the chaotic function to form the initial population of a GA. A fitness function was defined to mean an encrypted image with the highest entropy and lowest correlation coefficient among adjacent pixels; such that at each stage of the GA, the result obtained from each iteration is optimised to produce superior generations. Thus, the chaotic function was employed for initial encryption, and the GA used to improve the encryption process of the image. This model was robust but not optimal since the procedure was complex with some overhead cost. Recently, (Agbedemnab, Baagyere & Daabo, 2019) also proposed an image encryption scheme based GA and Residue Number System (RNS). The scheme presented a three-layer approach which produced chaotic results. The trade-off in combining these approaches to using only the GA as demonstrated in this paper turns not to be optimal.

3. PROPOSED SYSTEM

The proposed system is an application developed using Visual Basic in Microsoft Visual Studio 2013. The purpose is to simplify the GA into an algorithm that is implementable on encrypting images to obtain optimal but chaotic images after encryption. The general procedure for any GA process is as follows:

i) Initialize Population: Here the parents that shall be used in the procedure are initialized. It is from the parents the subsequent generation shall be created.
ii) Selection: Two parent chromosomes are selected.
iii) Crossover: The chromosomes are crossed over to form new offspring
iv) Mutation: The new offspring are subsequently manipulated to obtain features that are new and are either superior or were absent in parents.

The following steps are the specific procedures that were followed in the development of the application.
i) Initialization:
o Load the Parent (Image)
o Determine width and height of the Image

ii) Selection and Crossover:
o Divide the Parent into a set of chromosomes(blocks)
o The blocks should be of preferred sizes
o Select randomly two sets of chromosomes
o Swap the selected chromosomes
o Do this for all chromosomes to achieve offspring (crossed-over encrypted Image)

iii) Mutation:
o Divide the offspring into a set of chromosomes
o These chromosomes should also have preferred sizes
o Select randomly two chromosomes and swap
iv) Repeat above stated process until desired or preferred results are obtained.

4. RESULTS AND DISCUSSION

The application is a single interface application and performs the following functions:
i) Load Image(parent)
ii) Display loaded Image
iii) Generate chromosomes
iv) Crossover chromosomes
v) Display Crossed over offspring (first encrypted Image)
vi) Mutate offspring and display final offspring (final encrypted Image)
vii) Reverse mutation
viii) Reverse crossover
ix) Decrypt Image

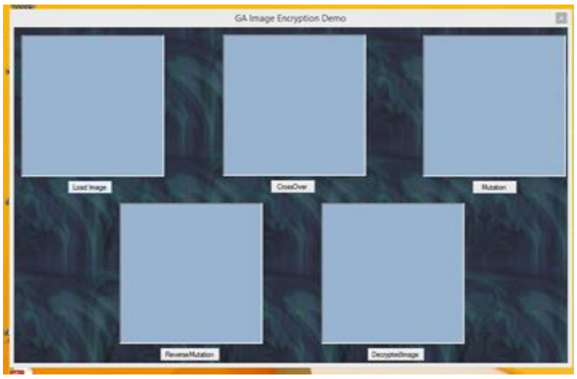

Figure. 1. GAIE Launched Interface

Figure 1 shows the interface of the Genetic Algorithm Image Encryption (GAIE) demo application when launched. From Figure 1, it can be noted that the application has five controls:
i) *Load Image*
ii) *Crossover*
iii) *Mutation*
iv) *ReverseMutation*
v) *DecryptedImage*
Each of the above listed controls has a picture box to display the final results of the operations they perform. Next, the functionality of each control button is demonstrated from the plain image to achieve an encrypted image with the GA.

4.1 The Load Image Button
The Load Image Button as shown in Figure 2, when clicked performs the initialization process and also a part of the selection process. When this button is clicked, the user is able to select an image of his choice. This Image is then loaded after which the height and width of the Image are extracted. The Image is then split into blocks which serve as the chromosomes of the parent Image. The loaded parent Image is then displayed in the picture box.

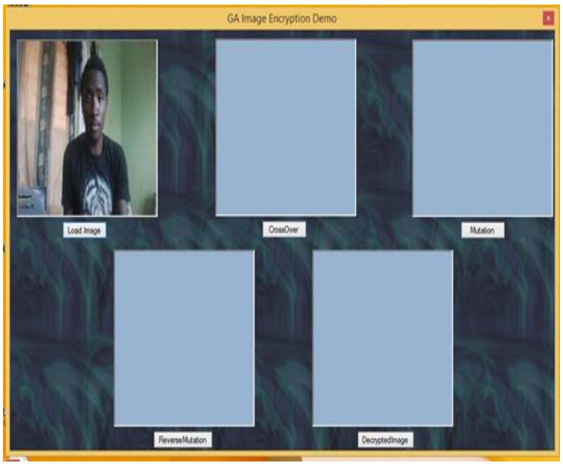

Figure.2. Load Image Control

4.2 The Crossover Button

This button when clicked performs the crossover function as shown in Figure 3. In this operation, the various chromosomes that where generated in the first operation are randomly selected and arranged in rows and columns. The uniqueness of this particular crossover operation is that the positions of chromosomes are not just swapped. The reason is that if this process operated by just swapping positions of two selected chromosomes, one would be able to find the position of two chromosomes by finding the previous position of just one chromosome.

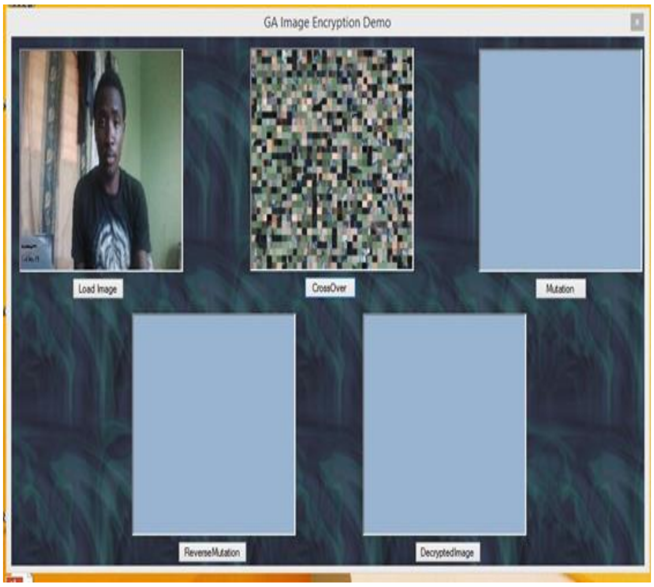

Figure. 3. Crossover Control

4.3 The Mutation Button

The Mutation button triggers the mutation process as shown in Figure 4. This process takes the encryption to a higher level than that of the Crossover function. It is realised that the offspring produced after the mutation attains features that are specifically not present in the parent and possesses it unique form of chromosomes. This brings to mind the purpose of the Mutation process in GA. That is to manipulate offspring chromosomes to obtain a better and optimal offspring. It is the same offspring but with stronger, finer and better features. In this particular case, the mutation process was executed by extracting portions of the previously generated chromosomes to create a new set of chromosomes, which can be uniquely attributed to the offspring created after crossover, by blending them with one another. When that was done, their positions were further randomly changed.

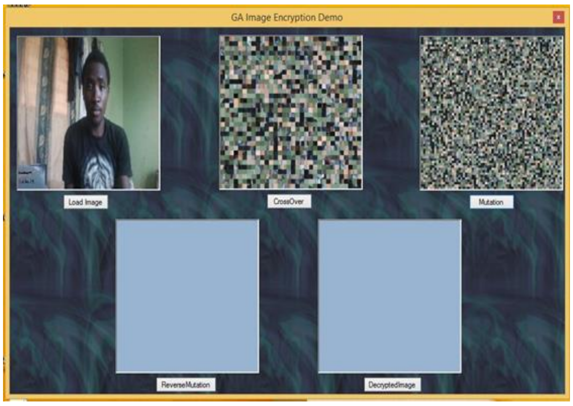

Figure.4. Mutation Control

4.4 Reverse Mutation

As shown in Figure 5, this button reverses the entire Mutation process. This is the beginning of the decryption process. This process is almost the same as the mutation operation but the difference is that it executes the process in the reverse direction.

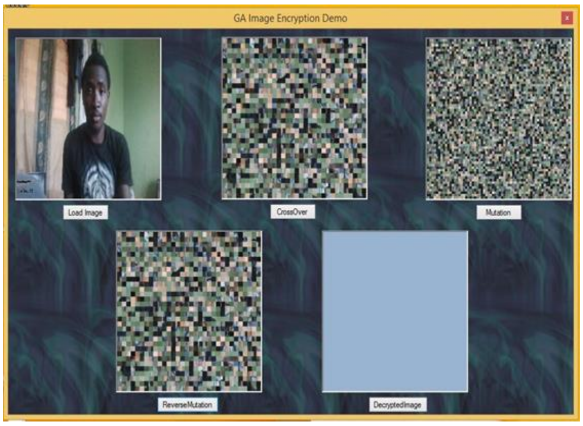

Figure.5. Reverse Mutation Control

4.5 The Decryptimage Button

The Decrypt Imagebutton finally reverses the crossover function using the Image generated after the ReverseMutation operation, to obtain the final decrypted Image.

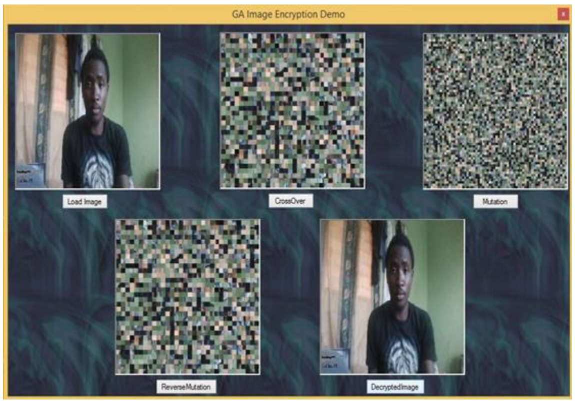

Figure. 6. Decryptimage Control

4.6 Discussion

From the results that have been illustrated above, it is very clear that the security of digital images are assured, if they are achieved with the kind of implementation demonstrated in this paper with GA. An analysis of the crossover function, it is noteworthy that without a key to facilitate the reverse function, the number of permutations that it might take to attain the original image can be unbounded. As an illustration, if the crossover operation is carried with just 8 chromosomes, the number of permutations that will be required to reverse the operation to obtain the original image shall be:

8! = 40320

The probability of using just 8 chromosomes for an encryption is very unlikely even in a puzzle game.

Subsequently, if the mutation operation also executed with 4 chromosomes (which is unlikely), the permutations required will be a multiplication of the permutations for crossover and that of mutation:

8! X 4! = 967680

Thus, the decryption process will be very tedious if not impossible for any evil-minded person to break this encryption process with any form of attack either brute force or Dictionary with a known plain/similar image.

5. CONCLUSION

Genetic Algorithm has gained huge popularity in various scientific areas in recent times. This algorithm has been applied to solve numerous problems due to it characteristics and features. Image encryption is also in a very high demand in numerous societies and all areas of lives and studies. Documents and confidential information are transferred over communication networks all the time. In this paper, an optimal encryption system based on the GA was presented to obtain highly secure and optimal transfer of digital images over communication networks. The results show that the GA is a powerful algorithm for securing digital data, in this case digital images where the encrypted data is chaotic and robust with minimum cost. In future, a research can be carried out using this work as a foundation on how to encrypt other forms of data using only the Genetic Algorithm.

6. REFERENCES

[1]. P. A. Agbedemnab, E. Y. Baagyere and M. I. Daabo, (2019) "A New Image Encryption and Decryption Technique using Genetic Algorithm and Residual Numbers," 2019 IEEE AFRICON, Accra, Ghana, pp. 1-9, doi: 10.1109/ AFRI CON 46755.2019.9133919.

[2].Al-Husainy, M. (2006). Image Encryption Using Genetic Algorithm. Information Technology Journal, Vol.5, No.5, 516-519.

[3]. Aloha, S., & Kehar, S. (2013). A technique for image encryption using digital signature. Optics Communications, Vol-2 I 8 (2203), 223-234.

[4]. Chang, C.-C., Hwang, M.-S., & Chen, T.-S. (2001). A new encription algorithm for image cryptosystems. The Journal of Systems and Software 58, 83-91.

[5]. Enayatifar, R., Abdullah, A. H., & Isnin, I. F. (2014). Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Optics and Lasers in Engineering, 56, 83–93. https://doi.org/ 10.1016/ j.optla seng.2013.12.003

[6]. Germanovic, S. (2014, july 11). How Technology Has Evolved in The Last Decade. Retrieved from Filtered: learn.filtered.com/blog/hpw-technology-has-evolved-in-the-last-decade

[7].InterPol. (2018, january 2). Cybercrime. Retrieved from INTERPOL: www.interpol.int/crime-areas/ cybercrime/ cyber crime

[8]. Jim, B., & Merkow, M. S. (2014, July 4). Information Security Principles of Success. Retrieved from Pearson IT Certidication: www.pearsoniotcertification.com

[9]. Kumari, A., & Goyal , S. (2016). Encryption and Code Breaking of Image Using Genetic Algorithm. International Journal of Advance Research in Computer Science and Management Studies, 356-361.

[10].Li, S., & Zheng, X. (2002). Cryptanalyis of Chaotic Image Encryprion Method. The 2002 IEEE International Symposium on Circuits and Systems(ISCAS 2002), (pp. 708-711). Arizona.

[11]. Öztürk, I., & Sogukpınar, I. (2007). Analysis and Comparison of Image Encryption. World Academy of Science, Engineering and Technology, Vol.1, No.3, 814.

[12]. Sakthidasan Sankaran, K., &Santhosh Krishna, V. S. (2011). A New Chaotic Algorithm for Image Encryption and. International Journal of Information and Education Technology, Vol. 1, No. 2, 137.

[13]. Zhang, S., & Karim, M. A. (1999). Color image encryption using double random phase encoding. MICROWAVE AND OPTICAL TECHNOLOGY LETTERS / Vol. 21, No. 5, 318-322.

APPENDIX – SAMPLE CODES (LOAD IMAGE)

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        If OFGSelectImage.ShowDialog = Windows.Forms.DialogResult.OK Then
PrntImg = Image.FromFile(OFGSelectImage.FileName)
        End If

        PictureBox1.Image = PrntImg
        'extracting the dimension of parent Image
        Width0 = PictureBox1.Image.Width.ToString
        Height0 = PictureBox1.Image.Height.ToString

        'running down width and heght to mod 8
HMod = Height Mod 8
WMod = Width Mod 8

        If HMod<> 0 Then
            Height = Height + (8 - HMod)
```

```
  HMod = Height Mod 8
          End If

          If WMod<> 0 Then
              Width = Width + (8 - WMod)
  WMod = Width Mod 8
          End If

  getGene(8, PrntgeneName, PictureBox1.Image)
      End Sub


  Public Sub getGene(geneDim As Integer, geneName As String, imgSource As Image)
  wid = Width0 / (Width0 / geneDim)
  hgt = Height0 / (Height0 / geneDim)

          'a new bitmap is created to hold individual genes
          Dim Splitted As New Bitmap(wid, hgt)
          Dim dest As New Rectangle(0, 0, wid, hgt)
  num_rows = CInt(Height0 / hgt)
  num_cols = CInt(Width0 / wid)
          Using gr As Graphics = Graphics.FromImage(Splitted)
              Dim sourcer As New Rectangle(0, 0, wid, hgt)

              For row As Integer = 0 Tonum_rows - 1
  sourcer.X = 0
                  For col = 0 Tonum_cols - 1
  gr.DrawImage(imgSource, dest, sourcer, GraphicsUnit.Pixel)
                      Dim filename As String = geneName + row.ToString("00") + col.ToString("00") +
  ".png"
  Splitted.Save(filename, ImageFormat.Png)
  sourcer.X += wid
                  Next
  sourcer.Y += hgt
              Next
          End Using
End Sub
```