IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. IV (Mar – Apr. 2015), PP 17-23
www.iosrjournals.org

An Effective Method to Hide Texts Using Bit Plane Extraction

Mayukh Das
Department of Electronics & Communication Engineering,
Techno India (West Bengal University of Technology), India

Abstract: *The work is to show how simply a data can be hidden inside an image and be easily extracted using Matlab. The work focuses on bit plane extraction and how it can be used to complete the above mentioned task. Both encryption and decryption procedure has been discussed. Here the data to be hidden refers to any piece of writing that may contain valuable information. It does not include any audio or video file. The main focus has been given to grayscaled images, which acts as the reference image that covers the data.*

Keywords: *Bit plane slicing, Decryption, Encryption, Histogram, Steganography*

I. Introduction

With the advancement of digital technologies and communication systems the need for privacy and security has enhanced. Hackers are always ready to exploit any sort of impuissance inside a network. Hence, data shared over any communication media must be concealed from them. Steganography, or data hiding is a significant process for this purpose as it allows an image to shroud a data to be shared. Hence the original data becomes invisible in the image. This invisible data are transmitted across the channel and finally decoded in the receiver end in order to extract the actual data back. This method intensifies the security of the system.

II. Background

A lot of research related to steganography is going on currently. This research steers one's attention towards hiding texts, scripts or any kind of writing in a simple way. Grayscale image or an intensity image is an image in which the value of each pixel is a single sample ranging from 0 to 255. Hence it carries only intensity information. Any value between 0-255 can be represented as an eight bit binary value, example (168)10 = $(10101000)_2$ [1]. Hence an intensity image can be sliced up into 8 bit planes which give a sequence of binary images. This procedure is called bit plane slicing that has been exploited in this work. The slicing can be done with RGB images also.

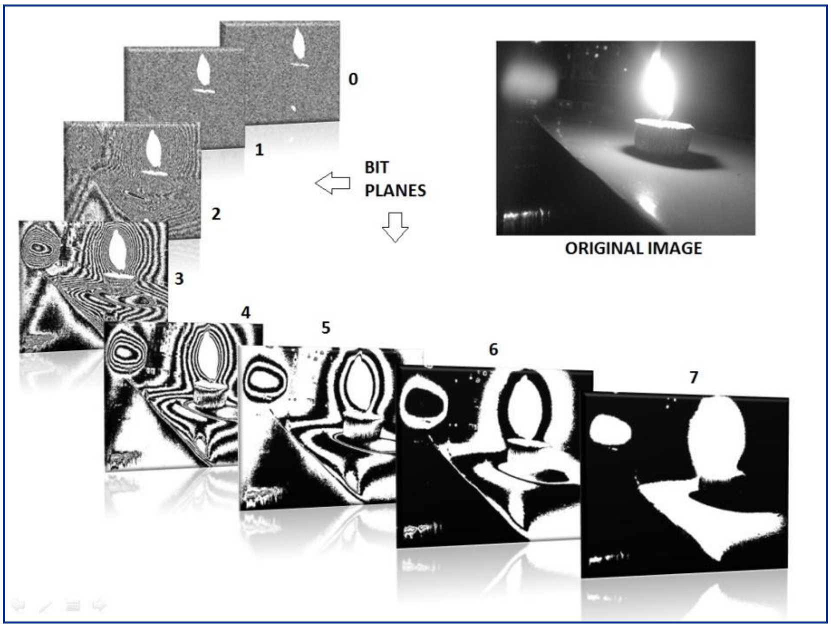


Fig 1: Bit plane slicing of a grayscale image

Hence a grayscale image may be considered as a combination of eight bit-planes where each bit-plane can be represented by a binary matrix [2].

The formation of bit plane is given by

$$Bp(i,j) = r\{(1/2)*floor[(1/2^{i})*o(i,j)]\} \qquad ……[2]$$

Where o(i,j) = original image
Bp(i,j) = bit plane info
r = remainder
floor(X) = rounds the elements of X to the nearest integers towards minus infinity.

III. Methodology

3.1 Encryption

The information to be hidden is first converted into an image file whose file format must be similar to the reference image. The MSB (most significant bit) plane contains utmost information of the image. This information content decreases as we go down and is least in the LSB (least significant bit) plane. The target is to slice both the reference as well as the data image and then overlay a particular data plane with the reference planes so as to form a single image with the desired data concealed in it. But the data plane selected must also contain adequate info so that the original data can be extracted after decryption. Hence, choosing a particular combination is the main challenge in the encoding part.

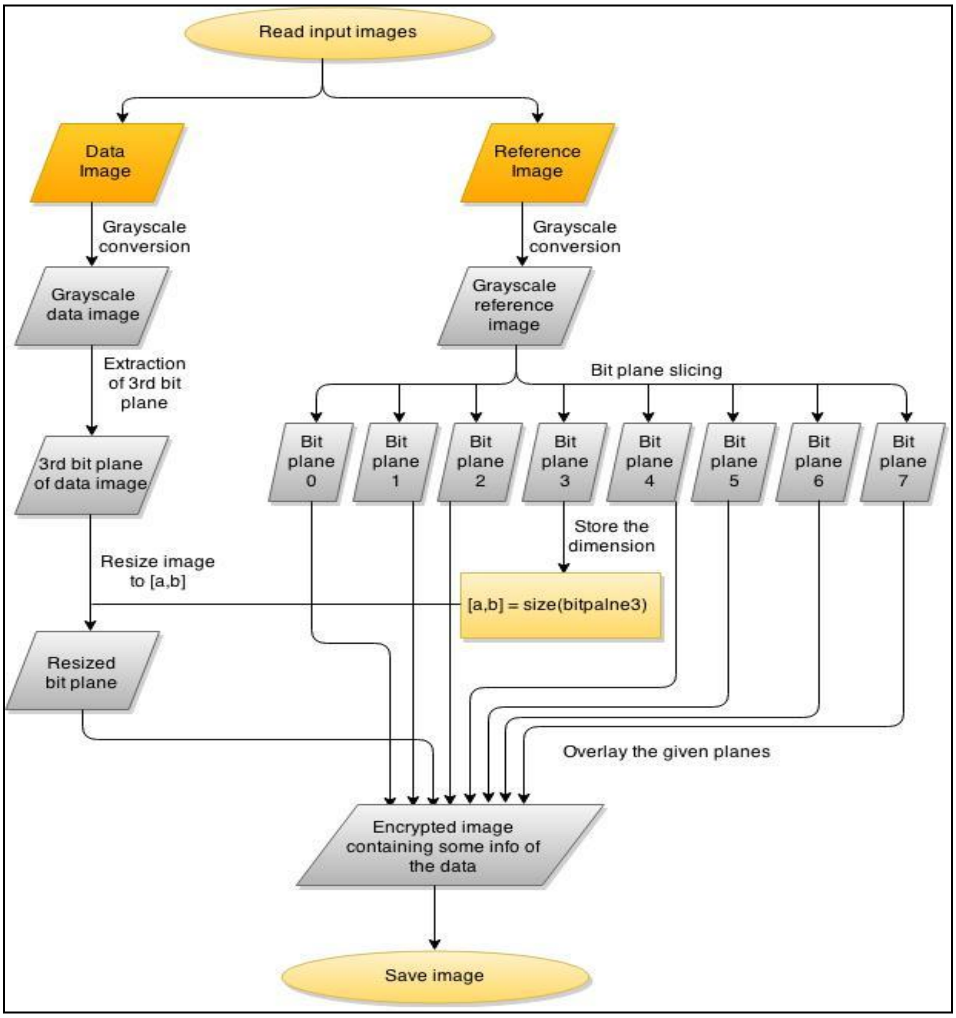


Fig 2: Block diagram of encryption process

Both the encoding and decoding parts were carried out in Matlab and the results were analyzed. The encoding starts with converting all the images into grayscale, because if we separate out R, G and B color components and create 8 bit planes for each color component then we can create total twenty four bit planes for each color image [2]. Therefore, it is quite convenient to deal with grayscale as it has only 8 bit planes. The Matlab command used to slice into bit planes is given by

$$C_j = \mathrm{mod}(\mathrm{floor}(refer/2^j),2);$$

Where C_j is the j^{th} bit plane

Then the 3^{rd} plane (the plane containing the 4^{th} bit information) of the data is extracted to be replaced by 3^{rd} bit of reference data. Direct replacement generates a dimension error. Hence the dimensions of the 3rd bit plane data were resized in accordance with the reference image (3^{rd} bit plane). The final encrypted image was obtained by amalgamating the rest of the reference bit planes with the 3rd bit data. Reason behind preferring the 3^{rd} plane is further discussed in the next section.

3.2 Decryption

The decoding process was done in two steps. It includes bit plane extraction and morphological image processing. The data plane that was blended with the reference image was pulled out and subsequent processing was done. Binary image contains various imperfections. Particularly our extracted bit plane is actually a bleached out image as it contains only the 4^{th} bit info about the original data. Hence morphological image processing removes a fair amount of imperfection by accounting for the form and structure of the image [3]. A pragmatic approach with this process helps us to erode the image in order to get an unclouded view of the data. Erosion with small square structuring elements shrivels an image by discarding away a layer of pixels from both the inner and outer boundaries of regions. The holes and gaps between different regions become larger, and small details are eliminated [3]. Here erosion was done using a 3×3 structure. The image is then complemented to give a convenient look. Processing has been done mainly to make the data decipherable.

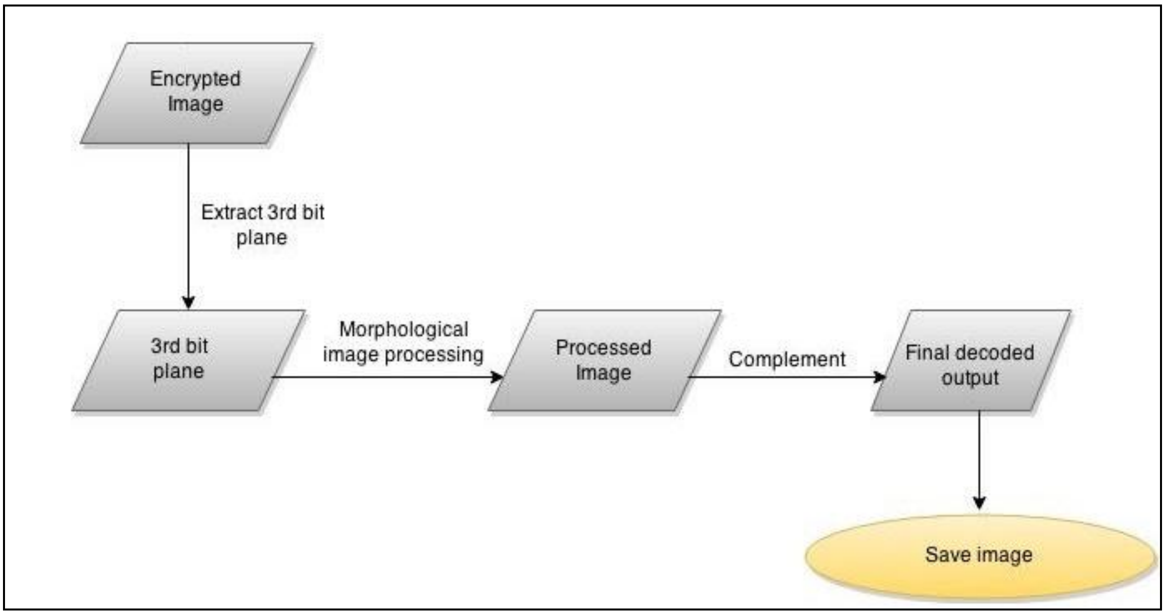


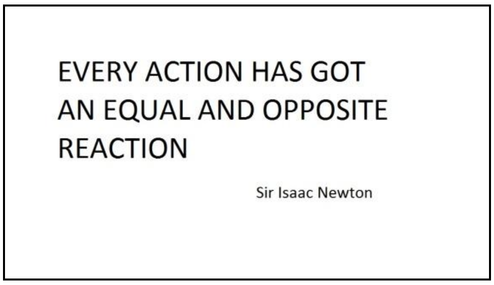
Fig 3: Block diagram of decryption process

IV. Results And Observations

4.1 Encryption

Many pairs of reference and data images were conflated into a single image and their histogram plots were compared with that of the original of which one is shown.

Fig 4: Reference image (R)



EVERY ACTION HAS GOT
AN EQUAL AND OPPOSITE
REACTION

Sir Isaac Newton

Fig 5: Data image (D)

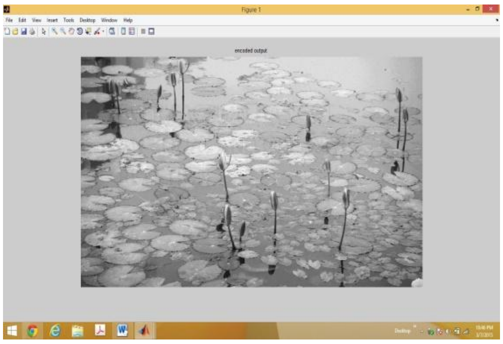
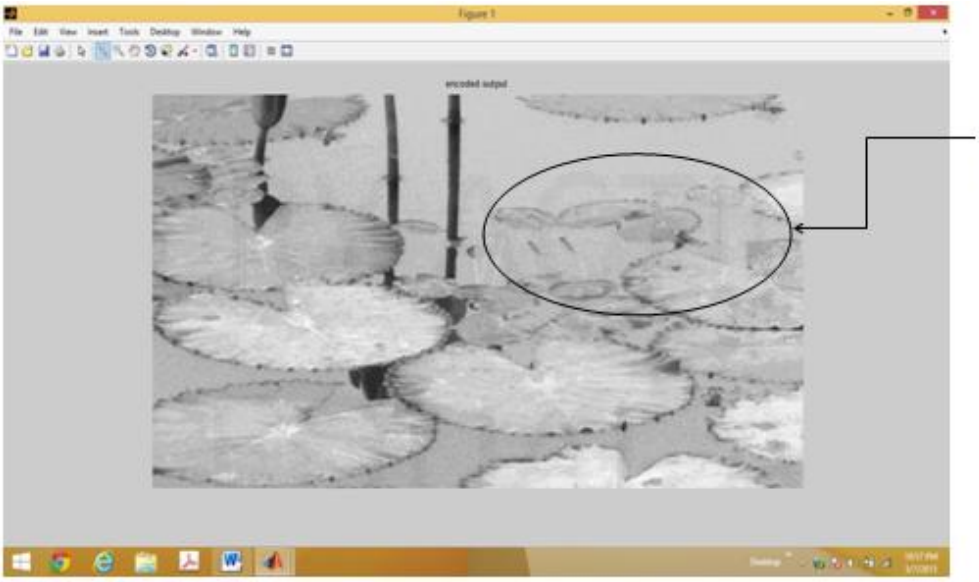


Fig 6: Encrypted image (E)

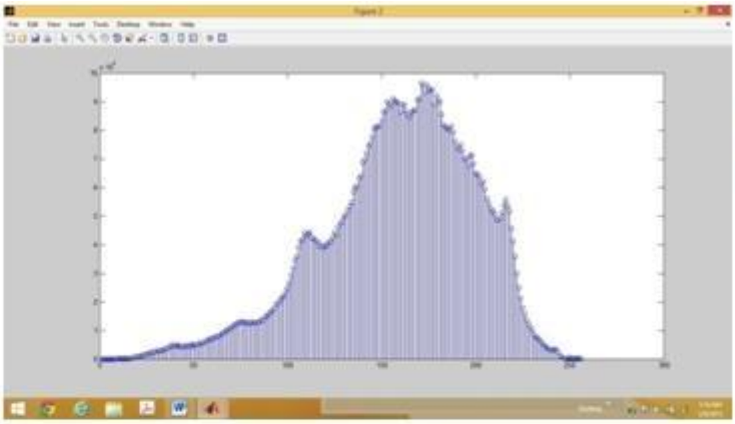
In the figure below a zoomed view of the encoded image is shown to substantiate the fact that the data is perfectly hidden inside.



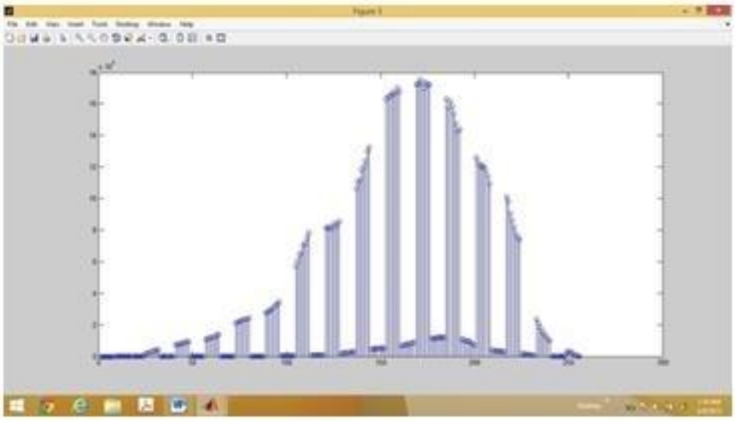
A faded imprint of "CTI" (ACTION) is visible

Fig 7: Zoomed view of E

Histogram plot of an intensity image shows the number of pixels in each and every intensity value. For a grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers(0-255) showing the distribution of pixels amongst those grayscale values [4].



Histogram plot of the reference image (when converted to grayscale)



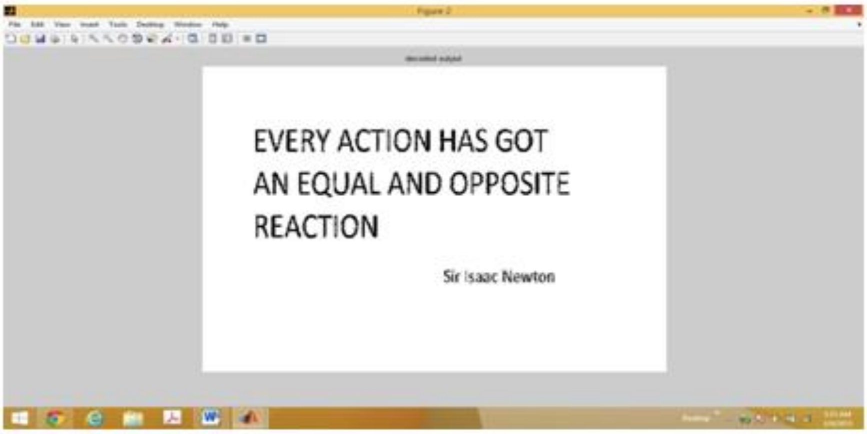
Histogram plot of the encrypted image

Fig 8: Histogram plots

On comparing the two histogram graphs we see that the envelope of the curve is similar but in the encrypted image some pixel values are suppressed. This is because the 3rd bit plane is removed from the image which actually contained a fair amount of info. Hence the histogram graphs emphasize the fact that the bit plane switching has been performed successfully.

4.2 Decryption

The encrypted data was decoded and the information was retrieved. Although the final output slightly differs from the actual data input, the info is clearly readable. Decoder part has mainly two sections i.e. bit plane extraction output and the final output after being processed.

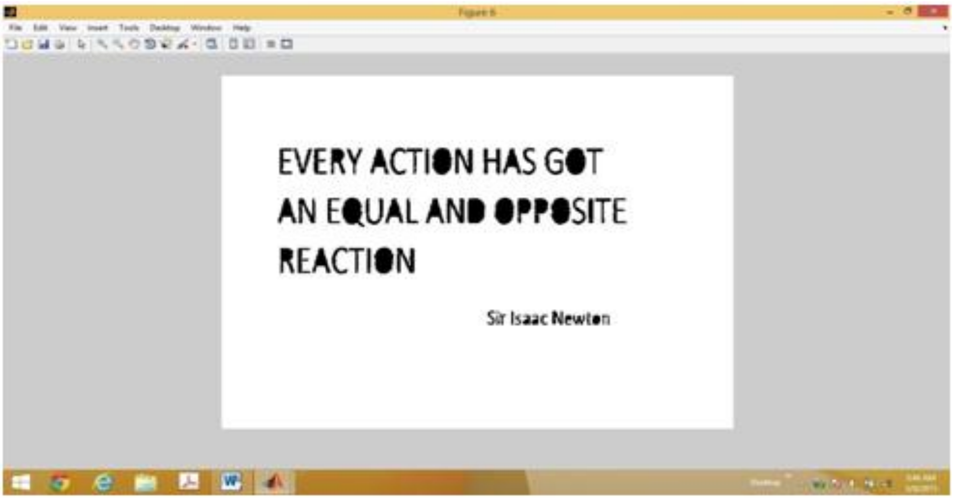


decoded output

EVERY ACTION HAS GOT
AN EQUAL AND OPPOSITE
REACTION

Sir Isaac Newton

Fig 9: Output on extracting the 3rd bit plane of the encrypted image



EVERY ACTION HAS GOT
AN EQUAL AND OPPOSITE
REACTION

Sir Isaac Newton

Fig 10: Final image

Therefore, we see that the decoded output is clearly decipherable. However, in the bit plane extracted output the smaller font characters are a bit obscured. Hence it has been processed to get a distinct appearance in the final image. The larger figures are easily readable even from the first output image. So finally each alphabet is clearly visible irrespective of font their font size.

4.3 Reason Behind Switching 3rd Bit Plane

In order to understand the usefulness of this plane, the encryption part has been carried out by switching other planes and corresponding outputs were observed.
Let us observe the encoded image when encryption is done by interchanging the 4th bit plane.

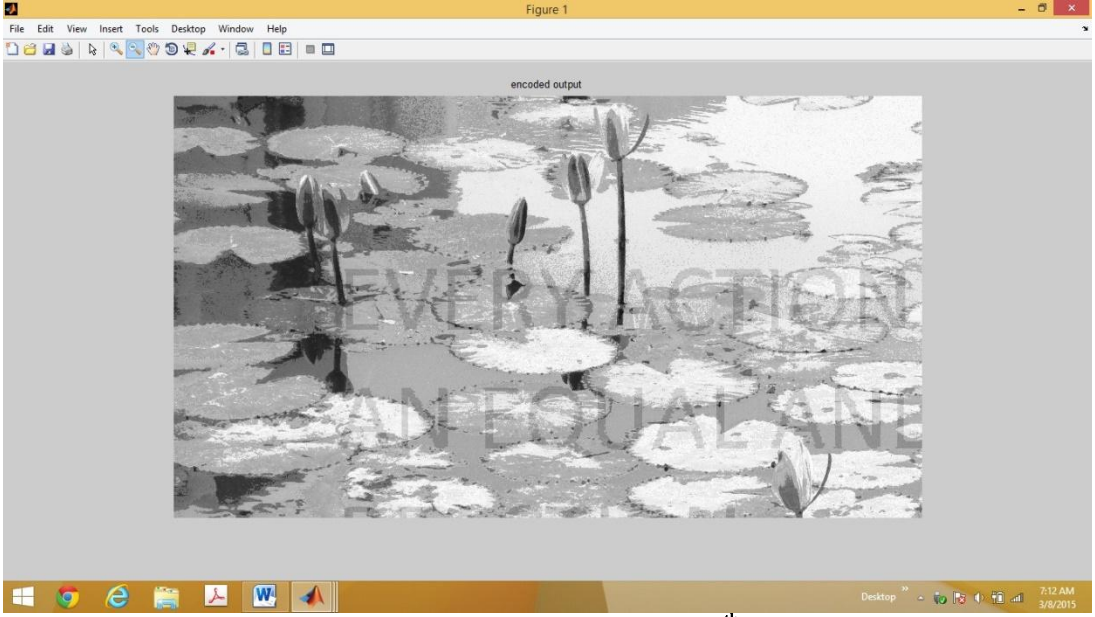


Fig 11: A zoomed view of the encrypted image when the 4th bit plane is interchanged

A significant amount of data is visible over the reference image. This is because bit plane 7, 6, 5 and 4 contains a substantial amount of information about the parent image. When the 4th plane is interchanged, a rich content of image R is replaced by a data plane (denser than the 3rd plane). Hence an imprint of the data is visible throughout the encrypted image. This impression goes darker when higher planes are switched.

Now let us observe the encoded image when encryption is done by switching the 2nd bit plane.

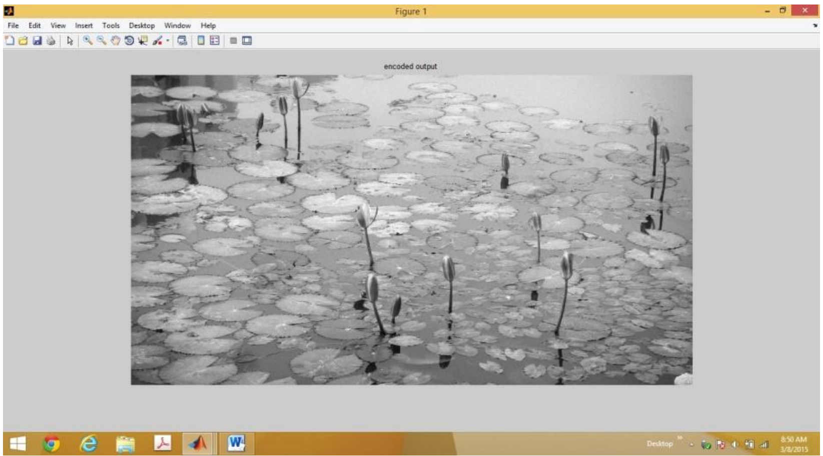
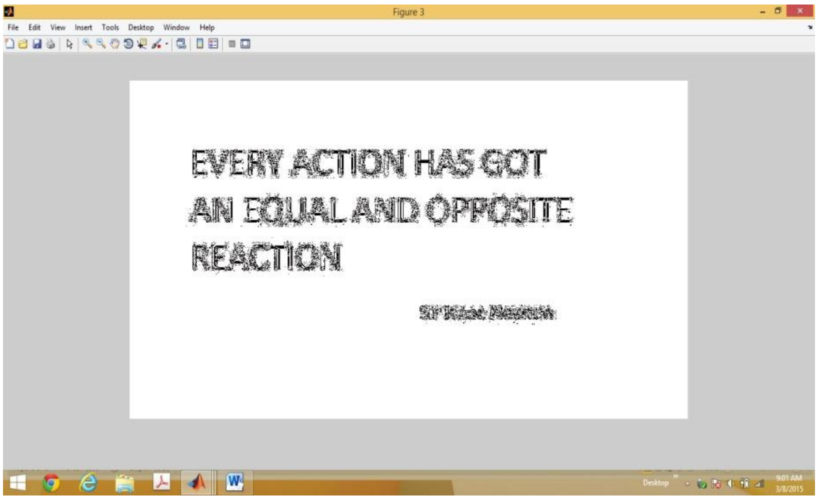


Fig 12: Encrypted image when 2nd plane is switched



EVERY ACTION HAS GOT AN EQUAL AND OPPOSITE REACTION

[illegible]

Fig 13: Decoded output

Here we see that encrypted image is fine, but the smaller font figures of the decoded output are too hazy to be read. This image when processed produces a more blurred picture. Specially the smaller alphabets are never distinctly readable. If the actual data never appears after decrypting then there is no point to choose this combination. Then decoded output gets more obliterated as we interchange further lower bit planes. Only 3rd plane switching yields a positive result for both encoding and decoding processes.

V. Conclusion

With the development of computer security, more and more research methodologies are also developing for the purpose of data hiding. I have tried to implement one methodology for this purpose. One of the main strengths of this work is the simplicity of the code. It allows a user to understand the method and develop their own data hiding techniques. There are however a few limitations also. The main drawback is that although readable the final decoded image is a bit distorted. An advance image processing technique is needed to solve this problem. However, data hiding technologies have advanced from limited use to global deployment. A new technique can further intensify the depth of exploration.

References

[1]. H. Faheem Ahmed and U. Rizwan, Embedding Multiple Images in an Image Using Bit Plane, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 1, January 2013, ISSN: 2277 128X

[2]. N S T Sai and R C Patil, IMAGE RETRIEVAL USING BIT-PLANE PIXEL DISTRIBUTION, International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 3, June 2011

[3]. Morphological Image Processing, Available at: https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm#morpho

[4]. Intensity Histogram, Available at: http://homepages.inf.ed.ac.uk/rbf/HIPR2/histgram.htm

[5]. A.M.Raid, W.M.Khedr, M.A.El-dosuky and Mona Aoud, IMAGE RESTORATION BASED ON MORPHOLOGICAL

[6]. OPERATIONS, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol. 4, No.3, June 2014.

[7]. W. Bender, D. Gruhl, N. Morimoto and A. Lu, Techniques for data hiding (IBM SYSTEMS JOURNAL, VOL 35, NOS 3&4, 1996)

[8]. Faheem Ahmed, H and Rizwan. U, An Alternative Technique in Data Embedding, Advanced Materials in Physics, pp 233 – 242, 2012.

[9]. Cachin. C., An information-theoretic model for steganography, Information and Computation, Vol. 192 (1), Ed. Academic, USA, pp. 41 – 56, 2004.

[10]. Rizwan. U and Faheem Ahmed. H, Comprehensive study on various types of steganographic schemes and possible steganalysis methods for various cover carrier like image, text, audio and video, International Journal of Scientific and Engineering Research, Volume 3, Issue 11, November 2012, pp 151 – 154.

[11]. Eric Cole, Ronald D. Krutz, “Hiding in Plain Sight: Steganography and the Art of Covert Communication”,(Wiley Publishing Inc. (2003).n).

[12]. Mr. Vikas Tyagi, Data Hiding in Image using least significant bit with cryptography, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, April 2012, ISSN: 2277 128X

[13]. Fridrich, J and M. Goljan and D. Hogea and D. Soukal, Quantitative steganalysis of digital images: estimating the secret message length, Multimedia Systems Journal - Special issue on Multimedia Security, Vol. 9 (3), pp. 288 – 302, 2003.