MANTECH Publications

Journal of Embedded Systems & Its Applications
Volume 2, Issue 1, January-June 2017

# *VHDL Implementation of Public key Cryptography based Image Encryption and Decryption using Reversible Data LSB Algorithm*

***B.Aruna[1], K. Shanmuga priya[2]***
*Department of EEE*
*RVS College of Engineering*
*Dindigul ( Anna University, Chennai)*
***Corresponding Author:*** *shanmugak4@gmail.com[2]*

***Abstract***

*This paper proposes a lossless, image encryption and decryption using LSB algorithm, and combined reversible data hiding schemes for cipher text images encrypted by public key cryptosystems with probabilistic and homomorphic properties. In the lossless scheme, the cipher text pixels are replaced with new values to embed the additional data into several LSB-Planes of cipher text pixels by multi-layer images. Then, the embedded data can be directly extracted from the encrypted domain, and the data embedding operation does not affect the decryption of original plaintext image. In the reversible scheme, a pre-processing is employed to shrink the image histogram before image encryption, so that the modification on encrypted images for data embedding will not cause any pixel oversaturation in plaintext domain. Although a slight distortion is introduced, the embedded data can be extracted and the original image can be recovered from the directly decrypted image. Due to the compatibility between the lossless and reversible schemes, the data embedding operations in the two manners can be simultaneously performed in an encrypted image. With the combined technique, a receiver may extract a part of embedded data before decryption, and extract another part of embedded data before decryption, and extract another part of embedded data and recover the original plaintext image after decryption. The proposed architecture of this paper will be planned to implement in video and also analysis the logic size, area and power consumption using Xilinx 14.2.*

***Keywords:*** *Reversible data hiding, Data embedding, LSB shifting algorithm*

## 1. INTRODUCTION

Now a days the data security and data integrity are the two challenging areas for research. There are so many research is progressing on the field like internet security, steganography, cryptography. Encryption and data hiding are two effective means of data protection. While the encryption techniques convert plaintext content into unreadable cipher text, the data hiding techniques embed additional data into cover media by introducing slight modifications. In some distortion-unacceptable scenarios, data hiding may be performed with a lossless reversible manner and Least Significant Bit (LSB) Algorithm.

Commonly the data hiding method is lossless if the display of cover signal containing embedded data is same as that of original cover even though the cover data have been modified for data embedding. For example, the pixels with the most used color in a palette image are assigned to some unused color indices for carrying the additional data, and these indices are redirected to the most used color. This way, although the indices of these pixels are altered, the actual colors of the pixels are kept unchanged. On the other hand, a data hiding method is reversible if the original cover content can be perfectly recovered from the cover version containing embedded data even though a slight distortion has been introduced in data embedding procedure. A number of mechanisms, such as difference expansion, histogram shift and lossless compression have been employed to develop the reversible data hiding techniques for digital images. Recently, LSB shifting algorithm has been introduced to improve the performance of reversible data hiding.

In this project the reversible data hiding and LSB shifting algorithms are used. In the reversible scheme, a preprocessing is employed to shrink the image histogram, and then each pixel is encrypted with additive homomorphic cryptosystem by the image provider. When having the encrypted image, the data-hider modifies the cipher text pixel values to embed a bit-sequence generated from the additional data and error-correction codes. Due to the homomorphic property, the modification in encrypted domain will result in slight increase/decrease on plaintext pixel values, implying that a decryption can be implemented to obtain an image similar to the original plaintext image on receiver side. Because of the histogram shrink before encryption, the data embedding operation does not cause any

overflow/underflow in the directly decrypted image. Then, the original plaintext image can be recovered and the embedded additional data can be extracted from the directly decrypted image.

The LSB shifting algorithm is applied here on the encrypted image and on the input data which is to be secured. Here Least Significant Bits are shifted the shifting process is same as an 8-bit shift register of Parallel input and serial output method (PISO). The bit shifting is performed according to the clock cycle, each bit require one clock cycle to complete their shifting process. Data embedding is the very important process in our project. Here the Pixels are selected based on the Pixel ratio which depends on the input image and cipher text size. Due to this data embedding based on pixel ratio it's possible to get the encrypted image as same as the input image.

In the reversible scheme, a histogram shrink is realized before encryption so that the modification on encrypted image for data embedding does not cause any pixel oversaturation in plaintext domain. Although the data embedding on encrypted domain may result in a slight distortion in plaintext domain due to the homomorphic property, the embedded data can be extracted and the original content can be recovered from the directly decrypted image. The marked image generated by this method versus the original image is guaranteed to be above 48 dB. This lower bound of PSNR is much higher than that of all reversible data hiding techniques reported in the literature.

## 2. LITERATURE REVIEW

### *2.1 Reversible Image Data Hiding With Contrast Enhancement*

Hao-Tian Wu, Member, IEEE, Jean-Luc Dugelay, Fellow, IEEE, and Yun-Qing Shi, Fellow, IEEE

A novel reversible data hiding (RDH) algorithm is proposed for digital images. Instead of trying to keep the SNR value high, the proposed algorithm enhances the contrast of a host image to improve its visual quality. The highest two bins in the histogram are selected for data embedding so that histogram equalization can be performed by repeating the process.

The side information is embedded along with the message bits into the host image so that the original image is completely recoverable. The proposed algorithm was implemented on two sets of images to demonstrate its efficiency.

### *2.2 Reversible Data Hiding*

Zhicheng Ni, Yun-Qing Shi, Nirwan Ansari, and Wei Su,IEEE transactions on circuits and systems for video technology, vol. 16, no. 3, march 2006.

A novel reversible data hiding algorithm, which can recover the original image without any distortion from the marked image after the hidden data have been extracted, is presented in this paper. This algorithm utilizes the zero or the minimum points of the histogram of an image and slightly modifies the pixel grayscale values to embed data into the image. It can embed more data than many of the existing reversible data hiding algorithms.

It is proved analytically and shown experimentally that the peak signal-to-noise ratio (PSNR) of the marked image generated by this method versus the original image is guaranteed to be above 48 dB. This lower bound of PSNR is much higher than that of all reversible data hiding techniques reported in the literature.

### *2.3 Separable Reversible Data Hiding In Encrypted Image*

Xinpeng Zhang,IEEE transactions on information forensics and security, vol. 7, no. 2, april 2012

This work proposes a novel scheme for separable reversible data hiding in encrypted images. In the first phase, a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data.

With an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

### *2.4 Data Hiding In 2-Color Images*

Yu-Chee Tseng, Member, IEEE Computer Society,and Hsiang-Kuang Pan In

In this paper, we propose a new scheme that improves in its capability to maintain

higher quality of the host image after data hiding by sacrificing some data hiding space. The new scheme can still offer a good data hiding ratio. It ensures that, for any bit that is modified in the host image, the bit is adjacent to another bit which has a value equal to the former's new value. Thus, the hiding effect is quite invisible.

## 3. PROPOSED SYSTEM

The proposed system of this paper is shown below; here the LSB algorithm and reversible byte loading are added,

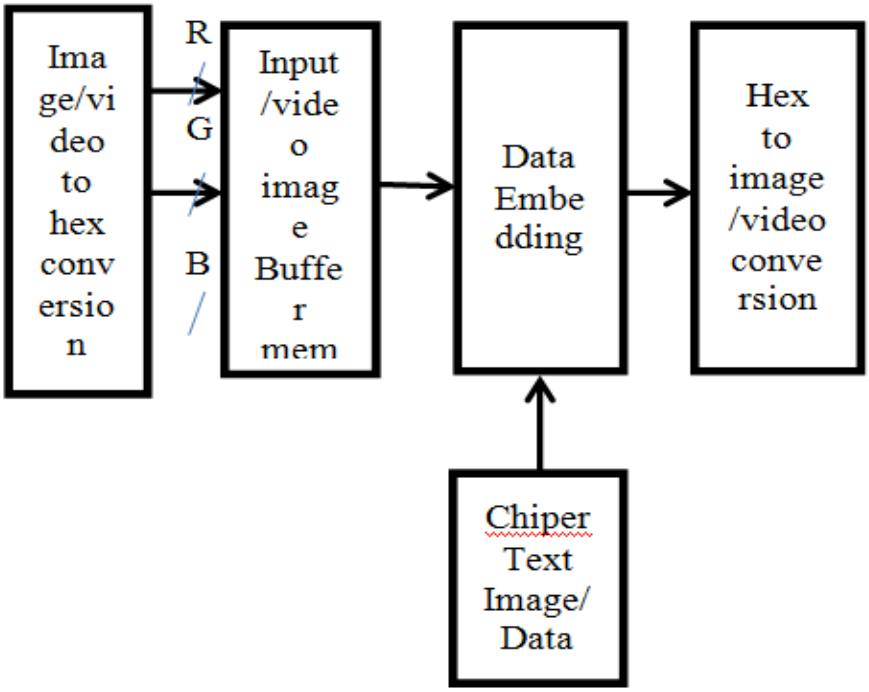



***Figure 3.1 Block Diagram of Proposed System***

The proposed diagram contains the image to hex data conversion block, this block is used to convert the input image in to hex data. it will generate RGB value of 8-bit, totally it contains 24 bit output, here the 24 bit output will be loaded to buffer memory and then the data is applied to the Data embedding block, the data embedding block is designed based on reversible data shifting and LSB algorithm, it will read the data from Cipher Text to image/data block,

after embedding the hex data will be converted to image.

***Data Embedding***

The data embedding block is designed using LSB algorithm and bit reversible shifting, the architecture is shown below, **(See Fig 3.2 below)**

In the block diagram the cipher text of key data is read by Byte reversible, and it will generate the output of 8-bit in reversible order, which means if the input is 48h it's equivalent of "01001000" in binary, after reversible the order of the output is "00010010" i.e. 12h. The LSB bit shifting block is same as a 8-bit shift register of Parallel input and serial output method (PISO) the diagram is shown below,

**(See Fig 3.3 below)**

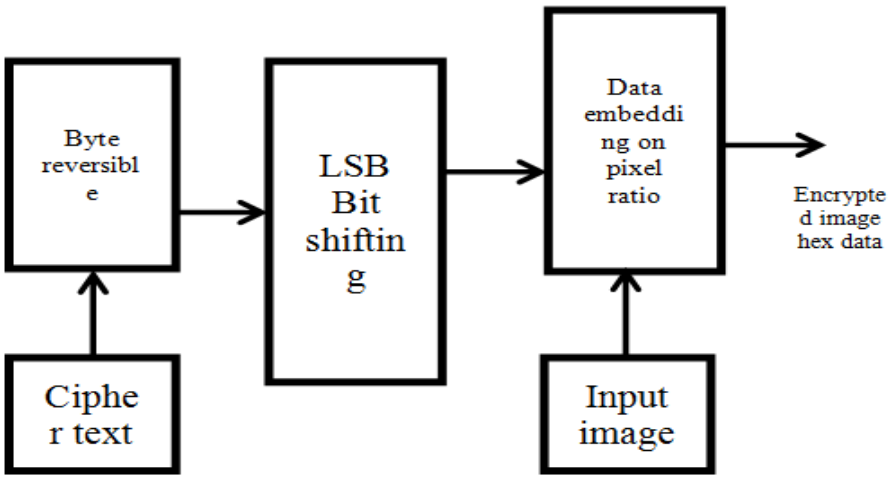



***Figure 3.2Block Diagram of Data Embedding***

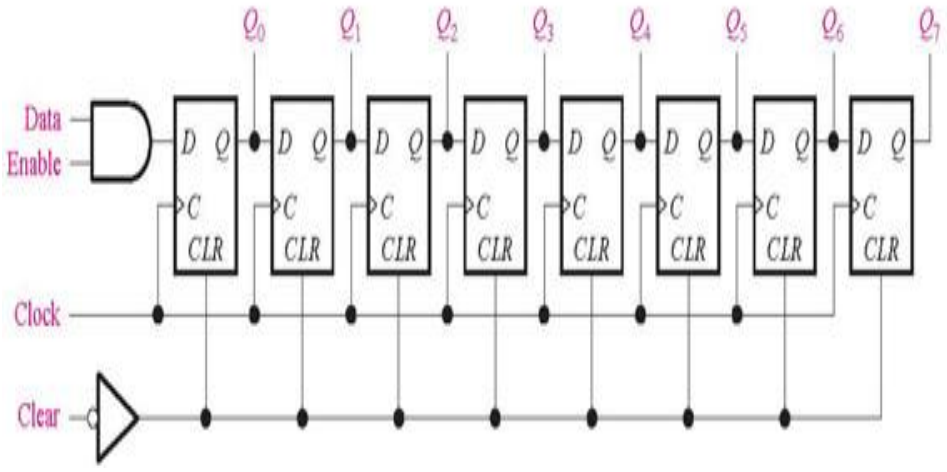



***Figure 3.3 8-bit shift register of Parallel input and serial output method***

Here the bit shifting is performed according to the clock cycle, each bit require one clock cycle to complete their shifting process. The process continues for the remaining bits in the input. The output of the LSB shifting block is applied to the data embedding block here the output of previous block is embedded according to the pixel ratio. Here we are selecting the Pixels based on the Pixel ration which depends on the input image and cipher text size.

***Advantages of Proposed System***

- Reduced distortion
- LSB bit shifting
- Ratio based pixel selection

**4. EXISTING SYSTEM:**

Encryption and data hiding technique are two effective means of data protection. While the encryption techniques convert plaintext content into unreadable ciphertext, the data hiding techniques embed additional data into cover media by introducing slight modifications. In some distortion-unacceptable scenarios, data hiding may be performed with a lossless or reversible manner.

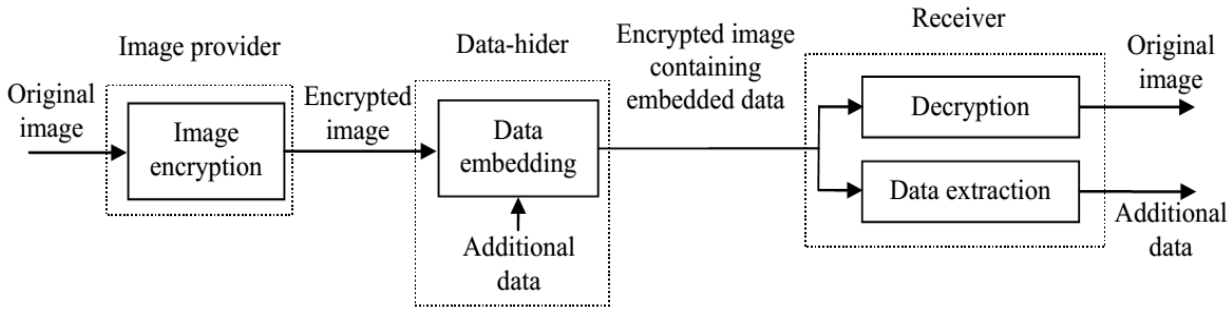



***Fig.4***

The above diagram contains, the original image is the input image data of hex values, it will be using to convert JPEG image to hex data conversion, which means every JPEG file contain header and data, we are read the data (ASCII) in the image file and the data to be converted into HEX value, and it will be stored it into the memory buffer.

After memory filling the data to be read and given to the Data-hider, here we are adding the additional data i.e. the key data of image encryption, here the LSB byte loading technique will be handled and it

will be loaded by random of 8-pixel's once. The decryption part of diagram is vice versa.

## 5. MODULE DESCRIPTION

### *5.1 Reversible Data Hiding Algorithm*

Reversible data hiding (RDH) approach in image processing is an innovative technique, where the information related to original cover recovered lossless approach, this lossless data extraction is done once the extraction of embedded message is successfully completed. The applications related to reversible data hiding are medical imagery, military imagery and law forensics, where no distortion of the original contents is allowed.

### *5.2 Histogram-Based Reversible Data Hiding*

Reversible data hiding enables the embedding of messages in a host image without any loss of host content, which is proposed for image authentication that if the watermarked image is deemed authentic, we can revert it to the exact copy of the original image before the embedding occurred.

We present an improved histogram-based reversible data hiding scheme based on prediction and sorting. A rhombus prediction is employed to explore the prediction for histogram-based embedding. Sorting the prediction has a good influence on increasing the embedding capacity. Characteristics of the pixel difference are used to achieve large hiding capacity while keeping low distortion.

The proposed scheme exploits a two-stage embedding strategy to solve the problem about communicating peak points. We also present a histogram shifting technique to prevent overflow and underflow. Performance comparisons with other existing reversible data hiding schemes are provided to demonstrate the superiority of the proposed scheme.

The histogram modification technique involves generating histogram and finding the peak point and the zero point and shifting histogram bins to embed message bits. For a given host image, we first generate its histogram and find a peak point and a zero point. A peak point corresponds to the grayscale value which the maximum number of pixels in the given image assumes. On the contrary, a zero point corresponds to the grayscale value which no pixel in the given image assumes.

For example, the histogram of the grayscale image (512×512×8) is illustrated in figure 4.1, in which the peak point is at

154 and the zero point is at 255. Let P be the value of peak point and Z be the value of zero point. The range of the histogram, [P+1, Z-1], is shifted to the right-hand side by 1 to leave the zero point at P+1.

Once a pixel with value P is encountered, if the message bit is "1," increase the pixel value by 1. Otherwise, no modification is needed. We note that the number of message bits that can be embedded into an image equals to the number of pixels which are associated with the peak point.

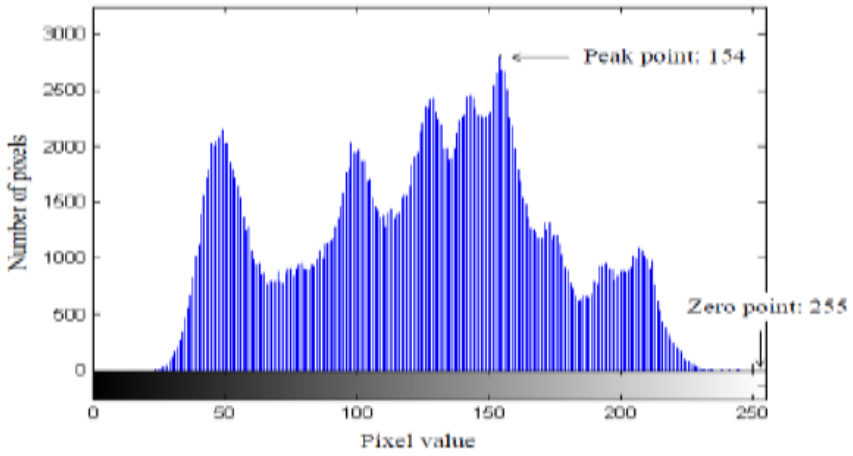



***Figure 5.1. Histogram of the grey scale image***

### *5.3 Rhombus Prediction And Sorting*

Use of pixel difference histogram introduced a significant performance advantage over previous methods. To improve our previous work, we present the prediction sorting to enhance the correlation of neighbouring pixels. In order to predict the pixel value of position ui, j in Figure 4.2, we use a rhombus prediction by considering four neigh boring pixels vi, j-1, vi-1, j, vi, j+1, vi+1, j. All pixels of the image are divided into two sets: the "White" set and "Gray" set.

The pixel value u of the White set can be predicted by using the four neighbouring pixel values of the Gray set and to hide data.

Note that the two sets are independent, which means changes in one set do not affect the other set, and vice versa. The

centre pixel ui, j can be predicted from the four neighbouring pixels vi, j-1, vi-1, j, vi, j+1, vi+1, j.

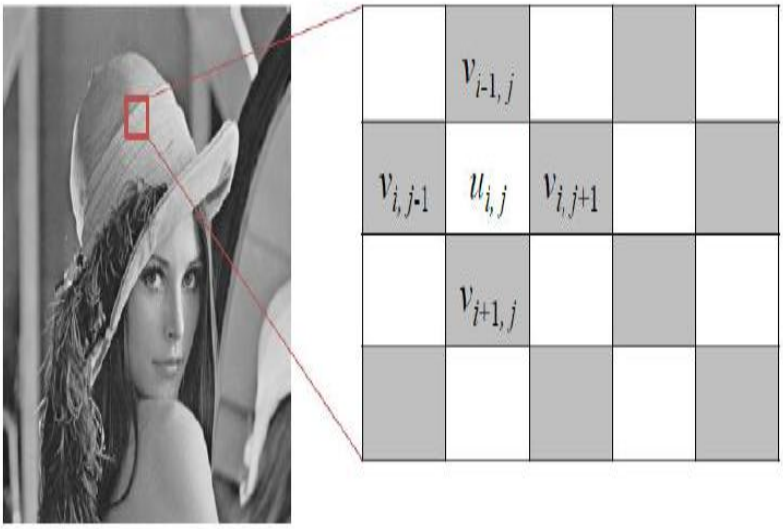



***Figure 5.2. Rhombus prediction***

**5.3.1** ***Histogram Modification on Pixel Differences***

The reversible data hiding scheme for White set is designed as follows.

1) Predict the pixel value ui, j in White set

2) Sort the host pixel ui, j according to the prediction value u'i, j, and produce the sorted pixels {x0, x1…xi} for 0 ≤ i ≤ N-1 where N is the pixel number of White set.

3) Calculate the pixel difference di between pixels

4) Determine the peak point P from the pixel differences

5) If di >P, shift xi by 1 unit:

$$y_i = \begin{cases} x_i, & \text{if } i = 0 \text{ or } d_i < P, \\ x_i + 1, & \text{if } d_i > P \text{ and } x_i \geq x_{i-1}, \\ x_i - 1, & \text{if } d_i > P \text{ and } x_i < x_{i-1}, \end{cases}$$

Where yiis the watermarked value of pixel i.

6) If di = P, modify xi according to the message bit:

$$y_i = \begin{cases} x_i + b, & \text{if } d_i = P \text{ and } x_i \geq x_{i-1}, \\ x_i - b, & \text{if } d_i = P \text{ and } x_i < x_{i-1}, \end{cases}$$

Where yiis the watermarked value of pixel i.

7) Construct the watermarked White set according to the sorted pixels {y0, y1…yi} for 0 ≤ i ≤ N-1 where N is the pixel number of White set.

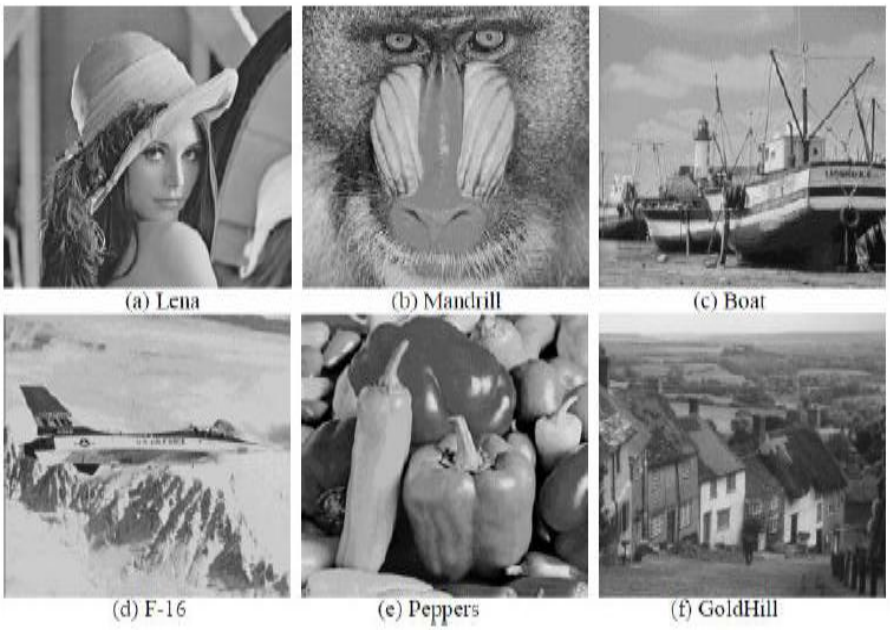



***Figure 5.3 Six test images used for performance evaluation***

## 6. LEAST-SIGNIFICANT BIT (LSB) TECHNIQUE

The least significant bit of some or all of the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types (i) 24 bit images and (ii) 8 bit images. In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image; much so the resultant stego image looks almost same as the cover image. In 8 bit images, one bit of information can be hidden.

The hidden image is extracted from the stego-image by applying the reverse process. If the LSB of the pixel value of cover image C(i,j) is equal to the message bit m of secret massage to be embedded, C(i,j) remain unchanged; if not, set the LSB of C(i, j) to m. The message embedding procedure is given below

S(i,j) = C(i,j) - 1, if LSB(C(i,j)) = 1 and m = 0

S(i.j) = C(i,j), if LSB(C(i,j)) = m

S(i,j) = C(i,j) + 1, if LSB(C(i,j)) = 0 and m = 1

where LSB(C(i, j)) stands for the LSB of cover image C(i,j) and m is the next message bit to be embedded. S(i,j) is the stego image. As we already know each pixel is made up of three bytes consisting of either 1 or a 0.

***6.1Data Embedding***

The embedding process is as follows.

Inputs: Cover image, stego-key and the text file

Output: stego image

***6.1.1 Procedure***

***Step 1:*** Extract the pixels of the cover image.

***Step 2:*** Extract the characters of the text file.

***Step 3:*** Extract the characters from the Stego key.

***Step 4:*** Choose first pixel and pick characters of the Stego key and place it in first component of pixel.

***Step 5:*** Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.

***Step 6:*** Insert characters of text file in each first component of next pixels by replacing it.

***Step 7:*** Repeat step 6 till all the characters has been embedded.

***Step 8:*** Again place some terminating symbol to indicate end of data.

Step 9: Obtainedstego image

**7. ENCRYPTION ALGORITHM:**

1) Initially reading the file of JPEG and converted it's to RAW Data Conversion with support of MATLAB

2) After Data conversion, the RAW Data format to store into the Memory module of input data memory, Parallely the same process will be done it for key data memory but it's a normal data.

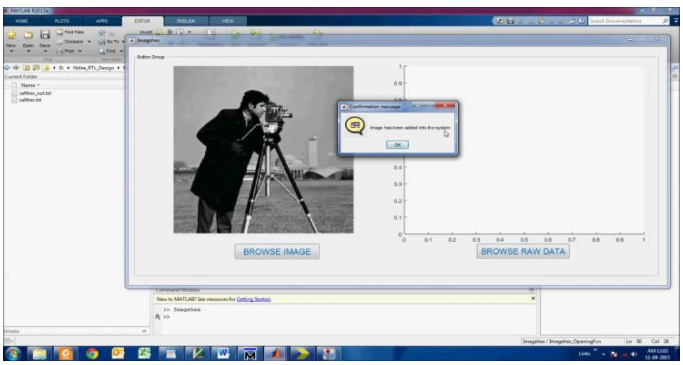
3) Then Read the input data memory and it to be loaded input output memory.

4) Based upon the ratio of 12:8, 4:1 format, the key data will be read and shift the data, finally the LSB data will be loaded into the LSB bit of output memory data.

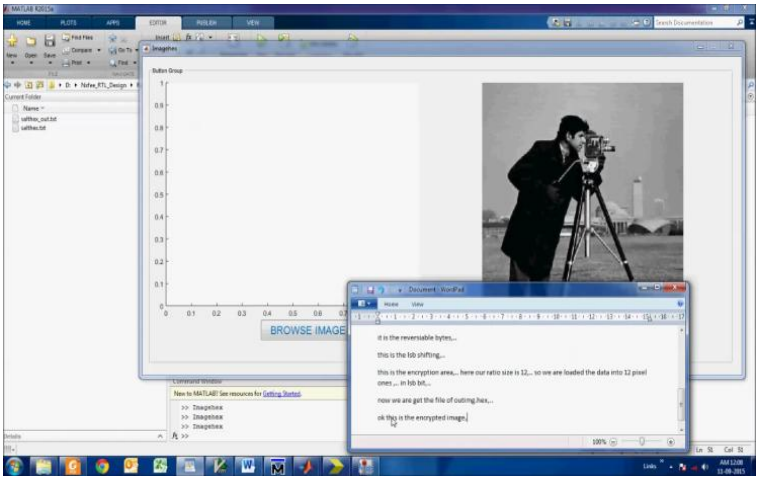
5) Read the output memory data and the RAW data to be converted into the file format of JPEG.

***Output Screenshots***

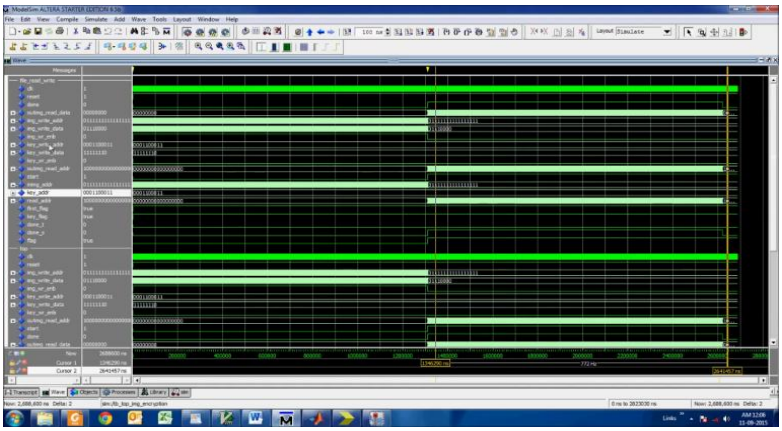
***Input image:***



***Output image:***




***Simulation output:***



## CONCLUSION

In cryptography the message being sent at one end remains confidential and should be received only by the intended receiver at the other end. Even though many cryptographic algorithms are developed, The Reversible Data Hiding algorithm with LSB Shifting is used here for our system due to its extended security and high speed operation.

This extended security is achieved by introducing the LSB Shifting and Pixel based data embedding. Due to this pixel based data embedding the quality of the encrypted image does not change.

The encryption part of the projectis completed for phase one. In future the decryption section will be implemented.

**ACKNOWLEDGMENT**

This project is funded for Master of Engineering in Embedded System Technologies under the Anna University, Chennai, In the academic year 2016-2017.

**REFERENCES**

1) J. Tian, "Reversible data embedding using a difference expansion,"IEEE Trans.Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896,Aug. 2003.

2) K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption," IEEE Trans. Information Forensics & Security, 8(3), pp. 553-562, 2013.

3) Kamstra, L., Heijmans, H.J.A.M.: 'Reversible data embedding into images using wavelet techniques and sorting', IEEE Trans. Image Process., 2005, 14, (12), pp. 2082–2090

4) W. Zhang, X. Hu, X. Li, and N. Yu, "Optimal Transition Probability of Reversible Data Hiding for General Distortion Metrics and Its Applications," IEEE Trans. on Image Processing, 24(1), pp. 294-304, 2015.

5) X. Zhang, "Commutative Reversible Data Hiding and Encryption," Security and Communication Networks, 6, pp. 1396−1403, 2013.

6) X. Zhang, "Reversible Data Hiding in Encrypted Image," IEEE Signal Processing Letters, 18(4), pp. 255−258, 2011

7) Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted Signal-Based Reversible Data Hiding with Public Key Cryptosystem," Journal of Visual Communication and Image Representation, 25, pp. 1164-1170, 2014.

8) Z. Ni, Y. Q. Shi, N. Ansari, andW. Su, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.