A Data Embedding Scheme usingEncrypted Images

Athira M A
PG Scholar
Cochin College of Engineering

Sreebala P
Asst.Prof
SIMAT COLLEGE

Vipin Krishnan C V
Asst.Prof
Cochin College of Engineering

ABSTRACT

The use of computer networks for data transmission has created the need of security. Transmission of images is a daily routine and it is necessary to find an efficient way to transmit them over networks. This work describes the concept of Separable Reversible data hiding technique. When it is desired to send the confidential data over an insecure channel, it mustbe encrypted as well as compress the cover data and then embed the data into that cover data.Also it is important that the data hiding should be reversible in nature, and should be suitable for the encryption/decryption domain. In the Modified System, a content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. The data hider can hide the data in the encrypted image compressing the least significant bits of the encrypted image to obtain the space to hide the data by using data hiding key. The data can be retrieved at the receiver side using the data hiding key without the need for decrypting the image. But, the encrypted image will remain unchanged till it is decrypted using the encryption key,to reveal the original image content. The receiver who has both the encryption and data hidings keys can access the data embedded as well as the original image. .

Keywords

Reversible Data hiding, Image Encryption, Data Embedding,Data hiding, Data extraction, Decryption

1. INTRODUCTION

Information has been valuable since the dawn of mankind. As access to computer has been increased, Security for such informations has become correspondingly important. Most corporate assets were hard or physical: factories, buildings, land, raw materials, etc In the past, But in present more and more assets are computer-stored information such as proprietary formulas, customer lists, marketing, sales information, financial data etc.. Some assets exist as bits only which stored in various computers. In the current trends of the world, technologies have advanced so much that most of the individuals prefer using the internet as the primary medium to transfer data from one end to another across the world. There are many possible ways to transmit data using the internet: via emails, chats, etc. The data transition is made very simple, fast and accurate using the internet. However, The main problems with sending data, is the security threat it poses i.e. a personal data can be stolen or hacked by different ways. Therefore it must be so important to take data security into consideration, as it is one of the most essential factors that need attention during the data transfer.

Because of the increase in demand for information security, the field of encryption is become very important and it posses broad applications such as in multimedia systems, internet communication, medical imaging, military communication, etc. Image security is of that much concern as web attacks have become more and more serious.

Data security means protection of data from an unauthorized users or hackers and providing high security to prevent any further modifications. This area of security has gained more attention over the recent period of time due to the mass increase in data transfer rate over the internet. For improving the security features in transferring the data over the internet, many techniques like: Cryptography, Steganography have been developed. The Cryptography is a method to conceal information by encrypting it to ciphers and transmitting it to the receiver using an unknown key[6], Steganography provides more security by hiding the ciphers into a invisible image or other formats.[3]

In the present scenarios, a new challenge is to embed data in the encrypted images. Previous work proposes an embed data in an encrypted image by using an irreversible approach of data hiding[1]. A new idea is to apply, reversible data hiding algorithms on encrypted images by wishing to remove the embedded data before the image decryption. .

A. **System Definition**

Proposed scheme involves image encryption, data embed-ding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider will process the encrypted image using a data-hiding key to create spaces to accommodate the additional data.The data is then stored in these created spaces. At the receiver side, the data embedded in the created spaces can be easily retrieved from the encrypted image using the data-hiding key. A decryption with the key that was used for encryption can result in an image similar to the original version. Using both the encryption key and the data-hiding key, the embedded data can successfully extracted and the originalcover image can be perfectly recovered from the encrypted image. Both these processes are independent of each other and hence is said to be Separable.[1]

2. RELATED WORK

In Reversible data hiding, the image is compressed and encrypted by using the encryption key and the data to be hidden is embedded in to the image by using the data hiding key. At the receiver side, he first needs to extract the image using the encryption key and after that the user uses the data hiding key to extract the embedded data. It is a serial process and is not separable.[2]

Limitations include the following: Content of the image is revealed before data extraction and also in this scheme; the data extraction is not separable from the content decryption. That means if someone have the data-hiding key but not have the encryption key, he cannot extract any of the in formations from the encrypted image with embedded data. It is possible to combine the techniques of Cryptography and Steganography by encrypting a message using cryptography and then hiding the encrypted message using

Steganography [3]. This paper was studied as it helped in acquiring ideas on secure data hiding in digital images and understanding their features.

A. Features

1. Hiding Capacity: The size of information that can be hidden is related to the size of the cover. A larger hiding capacity decreases the bandwidth required.

2. Perceptual Transparency: It is important that the embedding occur without significant degradation or loss of perceptual quality of the cover.

3. Robustness: the ability of embedded data to remain intact if the stego-image undergoes transformations.

4. Tamper Resistance: refers to the difficulty for an attacker to alter a message once it has been embedded in a stego-image, such as a copyright mark was replaced with a pirate, by one claiming legal ownership. This paper gives information on various applications and features of data hiding technique.

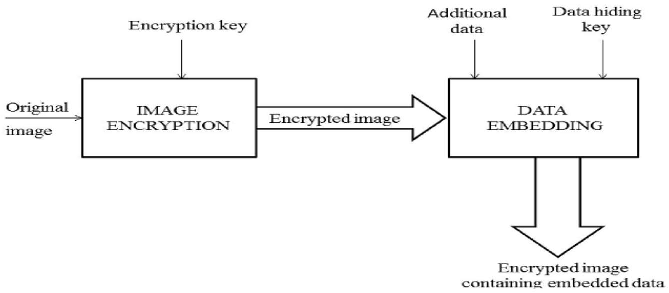
B. Applications

Copyright Protection: A secret watermark can be embed-ded inside an image to identify it as intellectual property. Detection of an embedded watermark is performed by a statistical, correlation, or similarity test.

In Medical field - to safely encrypt a patient's image and then embed his/her medical details onto it;

In Defense - to store and send the highly confidential details regarding a country's welfare and progress;

To manage Authentication Credentials of a Company; regarding Employee Details, Future Projects etc.



3. TASKS AND ARCHITECTURE

3.1 Tasks

The Modified System includes the following key features:

1. Image Encryption

The image is encrypted using an encryption key. A key is a piece of information that determines the output of an algorithm or process. The encryption key is provided by the user and the user uses same key at the receiver side for decrypting the image.

2. Data Embedding

In the data embedding phase, data is embedded into pixels of the encrypted image. The data-hider uses a data hiding key for embedding data.

3. Image Recovery and Data Extraction

In this phase, we will consider the three cases, First the receiver has only the data-hiding key,second having only the encryption key, and third case,both the data-hiding and encryption keys. When the receiver has the data-hiding key alone he could extract the embedded data only, cannot decrypt the image.when the receiver has the encryption key alone he could decrypt the image but cannot extract the data. Finally, when the receiver has both the data hiding key as well as the encryption key he could extract the data as well as the image from the encrypted data with the data being embedded.

3.2 Architecture

The proposed scheme involves image encryption, data em-bedding and data-extraction/image-recovery phases. The con-tent owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider will process the encrypted image using a data-hiding key to create spaces to accommodate the additional data.The data is then stored in these created spaces. At the receiver side, the data embedded in the created spaces can be easily retrieved from the encrypted image using the data-hiding key. A decryption with the key that was used for encryption can result in an image similar to the original version. Using both the keys, the embedded data can be successfully extracted and the original image can be perfectly recovered from the encrypted image. Both these processes are independent of each other and hence is said to be Separable.

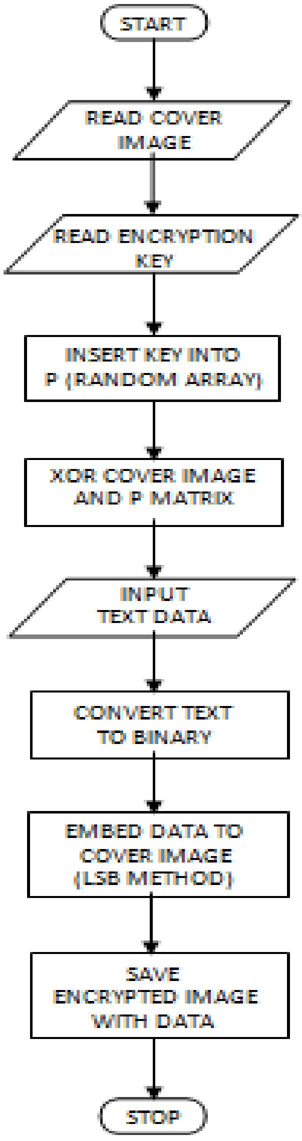

START
READ COVER IMAGE
READ ENCRYPTION KEY
INSERT KEY INTO P (RANDOM ARRAY)
XOR COVER IMAGE AND P MATRIX
INPUT TEXT DATA
CONVERT TEXT TO BINARY
EMBED DATA TO COVER IMAGE (LSB METHOD)
SAVE ENCRYPTED IMAGE WITH DATA
STOP

Fig. 1. Proposed System (Image Encryption and Data Embedding)

Encoding part is pictured in the following flow chart (fig.3).

4. IMPLEMENTATION

Encoding

1) Image Encryption:

a) : First, the input cover image is resized and converted into gray scale image to reduce the complexity.The image encryption key is then entered and is embedded in a random matrix of size 100x100.This is then xored with the cover image to get the encrypted cover image.

2) Data Embedding:

a) : Similarly the data image is resized(with size smaller than that of the cover image) and converted into gray scale.The last two lsbs of the cover image are cleared and the msbs of the data image is embedded instead, in the cover image.

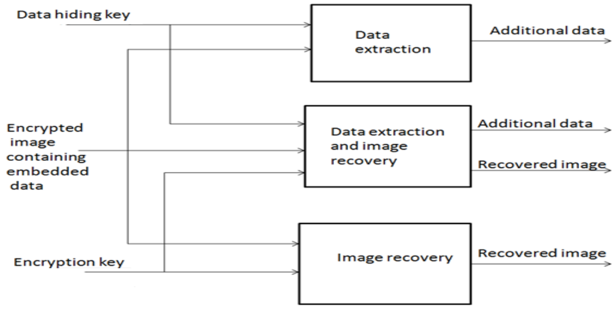


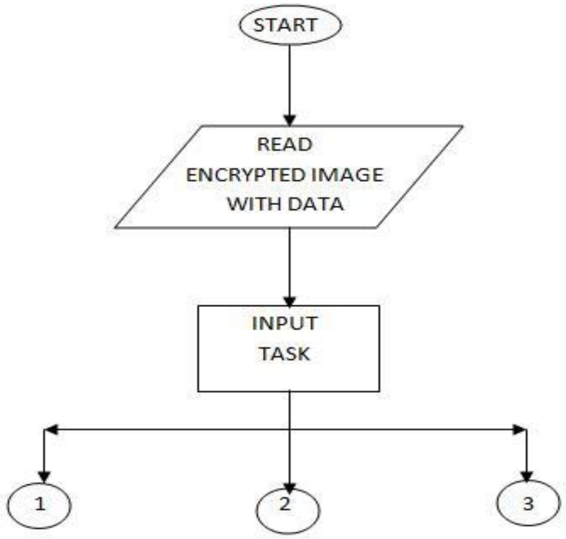
Fig. 2. Proposed System (Image Decryption and Data recovery)

Decoding

For decoding process, the receiver has to frist choose the options to decode Cover image (For correct encryption key only) - fig.5

1. Data image (For correct data hiding key only) - fig.6
2. Both (When both keys are available) - fig.7

Algorithm for Decoding image and Data :



\For extracting the cover image, first the user has to enter the encryption key. If the key is correct, it is xored with the encrypted image to get back the cover image. For retrieving the data,the user is required to enter the correct data hiding key. When a correct match is found the reverse algorithm for data embedding is done to retrieve back the data image.

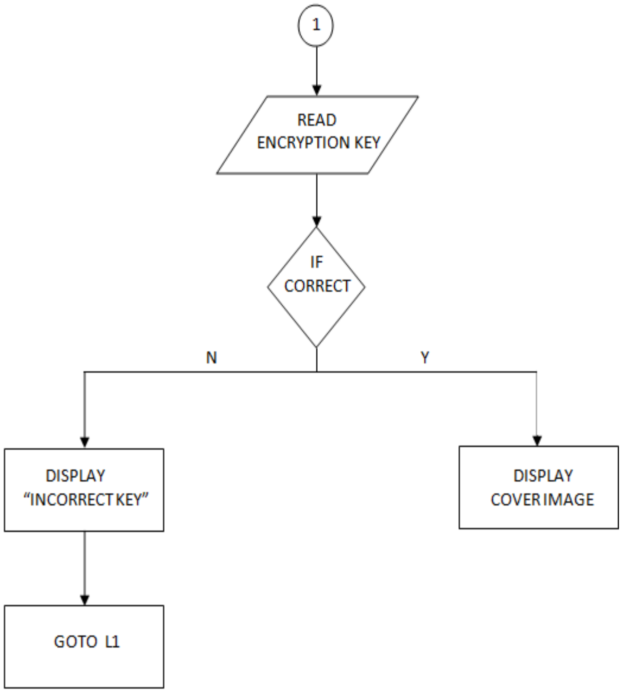


Fig. 5. First case in decoding process - Decryption of Cover image alone

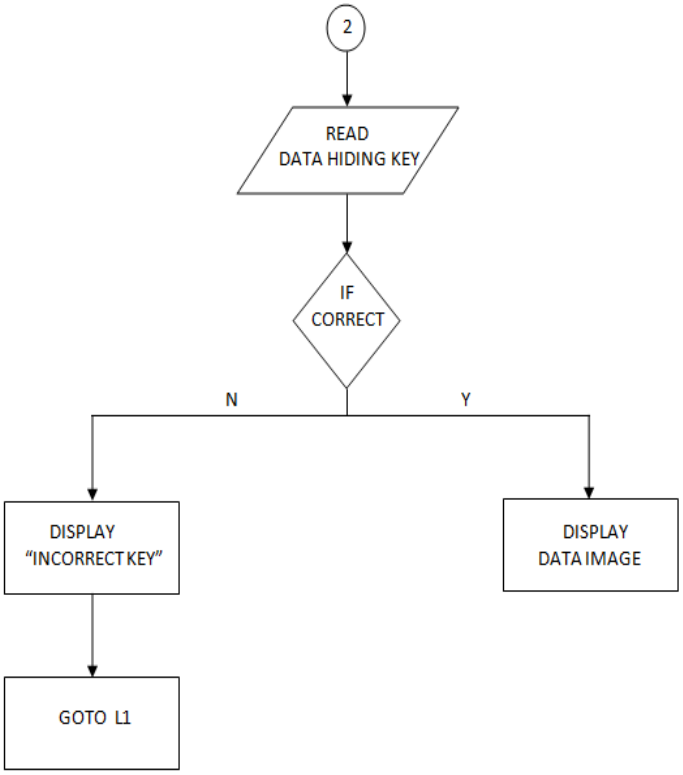


Fig. 6. Second case in decoding process - Extraction of data alone

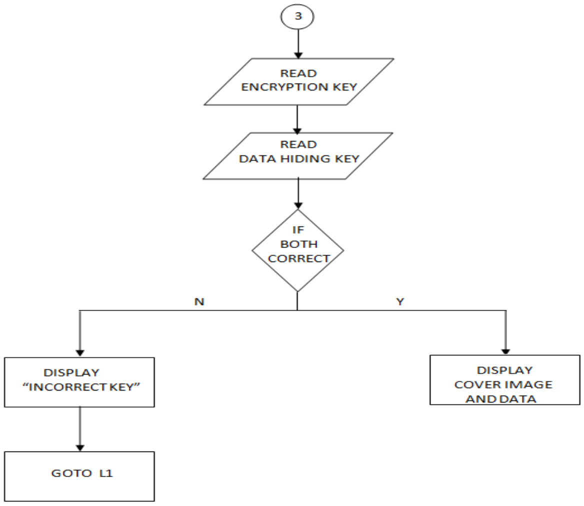


Fig. 7. Third case in decoding process – Both Cover Image Decryption and Extraction of data

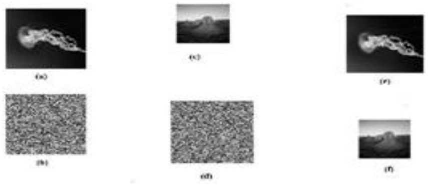
5. RESULTS AND DISCUSSIONS

The work was completed by using the system with following specifications:

Windows7, corei3, 2gb ram and software used was matlab 2012,version 7.1

The results are as follows:

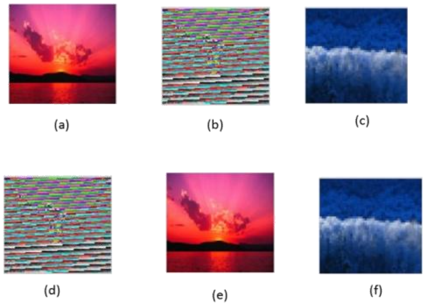
Experiment in Gray Scale Images:



A test image in Gray scale sized 100*100 was first used as the cover image shown in figure 4.1 (a). After Image Encryption, all the 8 bits of each pixel were converted into another value to get the encrypted image as in figure 4.1(b). The Data image, shown in figure 4.1 (c) to be hidden was then resized to half the size of Cover image, i.e, 50*50. It was then embedded into the Encrypted Cover Image by manipulating its 2 LSBs. The Encrypted Image with Hidden Data is shown in figure 4.1 (d). At the Decoder side, both Data Extraction and Image Recovery are done as per the decoders wish if and only if the corresponding keys match perfectly. The Decrypted Image and the Extracted Data are shown in figures, 4.1 (e) and 4.1 (f) respectively

Experiment in RGB Scale Images:

A test image in Gray scale sized 100*100*3 was first used as the cover image shown in figure 4.2 (a). After Image Encryption, all the 8 bits of each pixel of the three colours were converted into another value to get the encrypted image as in figure 4.2(b). The Data image, shown in figure 4.2 (c) to be hidden was then resized to half the size of Cover image (row and column - wise), i.e, 50*50*3. It was then embedded into the Encrypted Cover Image by manipulating its 2 LSBs. The Encrypted Image with Hidden Data is shown in figure 4.2 (d). At the Decoder side, both Data Extraction and Image Recovery are done as per the decoders wish if and only if the corresponding keys match perfectly. The Decrypted Image and the Extracted Data are shown in figures, 4.2 (e) and 4.2(f) respectively.



Graphical User Interface (GUI)

Step1: The file name of the cover image to be encrypted is entered in the edit text box 1 and pressing the image button displays the same in the axes 1 provided. Similarly, we can input the data image into the GUI axes

Step 2: The next push button press encrypts the image and displays the encrypted image in the corresponding axes which is axes

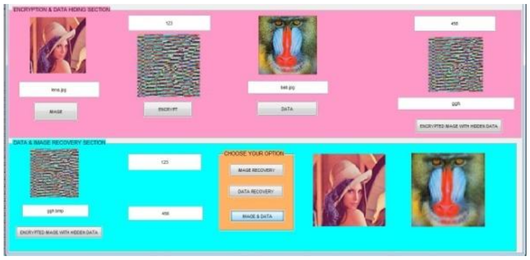
Step3: The Encrypted image with hidden data should be saved. So, a file name should be entered by the user in the provided field along with the data hiding key to save the encrypted image containing hidden data.

Step 4: at the Decoding side, the name of the file to be decoded has to be entered first pressing the button below, displays the same in the axes 4 provided. Now, the decoder has the option to choose, that is whether to

1) Decrypt the Cover image; 2) Extract Data image; or

3) Both. All the three cases would be possible only if the keys entered in corresponding Edit Boxes are equal. The out images are correspondingly displayed in axes 5 (Decrypted Cover image) and axes 6 (Extracted Data image).

Fig4.3 shows the layout of gui for hiding data image in encrypted image and the fig 4.4 shows the gui layout of hiding text data in image.


ENCRYPTION & DATA HIDING SECTION
IMAGE
ENCRYPT
DATA
ENCRYPTED IMAGE WITH HIDDEN DATA
DATA & IMAGE RECOVERY SECTION
CHOOSE YOUR OPTION
IMAGE RECOVERY
DATA RECOVERY
IMAGE & DATA
ENCRYPTED IMAGE WITH HIDDEN DATA

lena.jpg
123
bab.jpg
456
ggh
ggh.bmp
123
456

Step 1: The file name of the cover image to be encrypted is entered in the edit text box 1 and pressing the image button displays the same in the axes 1 provided.

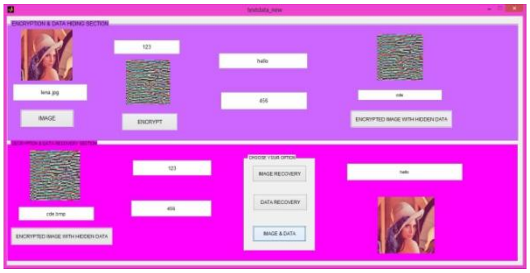
Step 2: The next push button press encrypts the image and displays the encrypted image in the corresponding axes which is axes no.2. The Encryption key is entered at Edit Box no.2;

Step 3: The data text is then input into the GUI Edit Box No.3 and the Data Hiding Key in Edit box No.4

Step 4: The Encrypted image with hidden data will be then displayed in axes 3 which should be saved. So, a file name should be entered by the user in the provided field along with the data hiding key to save the encrypted image containing hidden data.

Step 5: at the Decoding side, the name of the le to be decoded has to be entered rst and pressing the button below, displays the same in the axes 4 provided. Now, the decoder has the option to choose, that is whether to

1) Decrypt the Cover image; 2) Extract Text Data; or 3) Both. All the three cases would be possible only if the keys entered in corresponding Edit Boxes(no.7 and no.8) match. The decrypted image will be displayed in axes 5 and Hidden Data can be seen in output box no.9


ENCRYPTION & DATA HIDING SECTION
IMAGE
ENCRYPT
ENCRYPTED IMAGE WITH HIDDEN DATA
DECRYPTION & DATA RECOVERY SECTION
ENCRYPTED IMAGE WITH HIDDEN DATA
CHOOSE YOUR OPTION
IMAGE RECOVERY
DATA RECOVERY
IMAGE & DATA

lena.jpg

123

hello

456

cde

cde.bmp

123

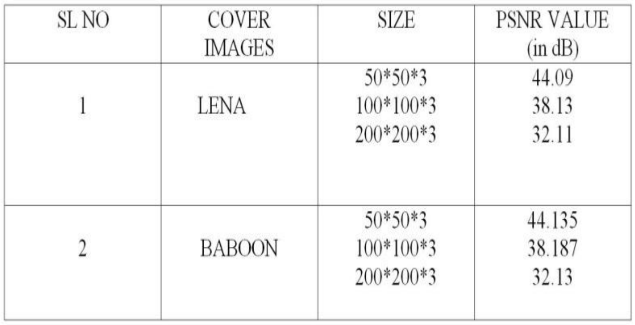
456

hello

Fig. 11. Modified System (Layout of Gui)

Evaluation

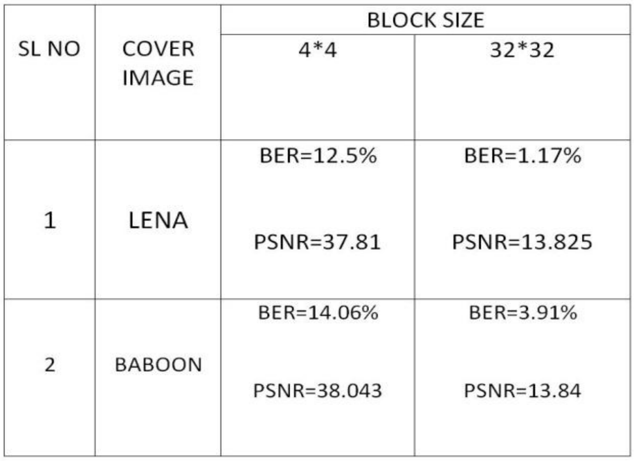
Table: 1 The variation in PSNR according to the variation in the size of the Cover images



SL NO	COVER IMAGES	SIZE	PSNR VALUE (in dB)
1	LENA	50*50*3 100*100*3 200*200*3	44.09 38.13 32.11
2	BABOON	50*50*3 100*100*3 200*200*3	44.135 38.187 32.13

It can be clearly seen from the tabular column that the PSNR values are inversely proportional to the Size of the Cover Images; i.e., as the Cover Image size increases, PSNR value goes on decreasing. But still, reasonable PSNR values could be achieved in the Modified System when compared to the Existing System.

Table2 :The Bit-error rates and the PSNR values for the Existing System



SL NO	COVER IMAGE	BLOCK SIZE	
		4*4	32*32

1	LENA	BER=12.5%
		PSNR=37.81
			BER=1.17%
			PSNR=13.825

2	BABOON	BER=14.06%
		PSNR=38.043
			BER=3.91%
			PSNR=13.84

when standard images like Lena and Baboon Face were used as Cover images. The images were first selected and then divided into different blocks of size (i) 4*4 and (ii) 32*32. Data was then embedded into these blocks. In this system, the cover image has to be first decrypted, only then it is possible to extract out the data. However, it was found that errors occurred often in the extracted data and hence, Bit-error rates for the extracted data were computed. PSNR values of the Decrypted images were also noted in comparison with their Original Cover images. These values are tabulated as in the Table 4.2. It can be found that Bit-error rates of data extracted increases but PSNR value improves for smaller sized blocks.But in the Modified System, Bit-error rate is found to be zero since all data bits are preserved.

6. CONCLUSIONS & FUTURE SCOPE

End user awareness is critical, as hackers often directly tar-get them. Users should be familiar with Security Policies and should know where the most recent copies can be obtained. Users must know what is expected and required of them. Typically this information should be imparted to users initially as part of the new hire process and refreshed as needed. The project is divided into four phases, namely, encryption, decryption, data hiding and data extraction parts. The first phase includes encrypting the cover data using a suitable key. The key is a piece of information that determines the functional output of the cipher. It is known only to the sender and receiver. We have programmed all four phases of our project using MATLAB with both image and text as data. The program reads an image and encrypts it using pseudo- random sequence. The pseudo-random sequence is modified with the encryption key provided by the user so that the decryption of the image and data recovery is possible with the help of a known key. The snap shots of the program after execution were obtained. A GUI interface is to be implemented in this system so that our project becomes more user-friendly.

In this research work we reviewed many papers on steganography techniques. These papers are good enough and have wide future scope We reviewed a lot of papers about data embedding schemes, steganography and cryptography. These papers are so informative and have wide varieties of future scope. This paper is so informative and is enough for those people who decided to start their work in this field. In future works, we are going to implement more advanced schemes with hybrid algorithms for preserving the data.

7. REFERENCES

[1] XinpengZhang(2011), Reversible Data Hiding in Encrypted Image, IEEE SIGNAL PROCESSING LETTERS, VOL. 18, NO. 4

[2] VinitAgham and TareekPattewar, A survey on seperable reversible data hiding technique, IMACST: VOLUME 4 NUMBER 1 MAY 2013

[3] ZaidoonKh. AL-Ani, A.A.Zaidan, B.B.Zaidan and Hamdan.O.Alanazi (2010), Overview: Main Fundamentals for Steganography, JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3.

[4] Firas A. Jassim (2013), A Novel Steganography Algorithm for Hiding Text in Image using Five Modulus Method, International Journal of Computer Applications,Volume 72 No.17

[5] Deepthi Barbara Nickolas, Sindhuja.Ba, Sivasankar.A (2013), Enhance-ment of Data Hiding Process in Encrypted Image Using Advanced Encryption Standard, International Journal of Current Engineering and Technology, Vol.3, No.2

[6] Majdi Al-Qdah (2013), Fast and Secure Image Hiding Scheme Based on Cryptographic Techniques, Journal of Emerging Trends in Computing and Information Sciences, Vol. 4, No. 2

[7] Vivek Jain1, Lokesh Kumar2, , Madhur Mohan Sharma3, Mohd Sadiq4, Kshitiz Rastogi5(2012), PUBLIC-KEY STEGANOGRAPHY BASED ON MODIFIED LSB METHOD, Students of CSE department, IMS Engineering College,

[8] Eugene T Lin and Edward J.delp,A Review of data hiding in dig-italimges,Video and Image processing Laboratory (VIPER) School of Electrical and computer Engineering purde university West lafayette,IndianaS

[9] Chi-Kwong Chan, L.M. Cheng(2003), Hiding data in images by simple LSB substitution, Department of Computer Engineering and Information Technology, City University of Hong Kong, Hong Kong

[10] Jun Tian(2003), Reversible Data Embedding Using a Difference Expan-sion, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,VOL. 13, NO. 8

[11] Aman Kumar, Dr. SudeshJakhar, Mr. Sunil Makkar (2012), International Journal of Advanced Research in Computer Science and Software Engineering,vol.2, issue 7 .