Steganography using Cryptography and Pseudo Random Numbers

Unik Lokhande
Research Scholar
Government college of Engineering, Aurangabad

A. K. Gulve
Asst. Professor,
Government college of Engineering, Aurangabad

ABSTRACT

When it comes to information hiding it is always a question whether to use Steganography or cryptography. Cryptography scrambles the message so that is it unrecognizable and without proper key the Encrypted message is useless. With the help of steganography messages can be passed over the network discreetly as it provides secure data hiding. It is very challenging and emerging field of research. Steganography can be simple as LSB technique or can be complicated as transform domain techniques, each technique have their own advantage. Both of these techniques are often confused with each other because in a way both are used to hide data. This paper discusses the important aspects of Steganography and Cryptography. How a simple LSB steganography technique in images can be complicated further using combination of Cryptography and Pseudo-random number generator.

General Terms

Image Steganography

Keywords

Steganography, Cryptography, Pseudo-Random Numbers.

1. INTRODUCTION

Now a day's most of the data is in digital form. Due to spread of internet and various computing devices this data is exceptionally increasing day by day. Some of this data is private like password, ATM pins, biometric data and secret messages. This type of data always encourages the criminals to make use of such data for their malicious purpose. All of this data travels over internet over unsecure networks. Maintaining confidentiality and integrity of sensitive data is really important. From here the need to protect data is emerged. To encode the data as it travels over unsecure networks or sometimes even hiding the existence of data itself. Cryptography and Steganography were introduced to provide data confidentiality. Cryptography scramble the message beyond recognition and the person with valid key can only read this data. Whereas Steganography hides the very existence of message. Passing the message secretly is always priority when it comes to war. Even if the suspicion is raised that the two parties are communicating can lead to disaster. Jeremiah Andrew Denton, Jr. was prisoner of war captured during Vietnam War and kept in prison and solitary confinement for 8 years. He was forced to do television press conference. He used this opportunity to successfully communicate with US Naval Intelligence confirming for the first time that the prisoners were tortured. He repeatedly blinked his eyes during interview conveying message in Morse code spelling out the word 'T-O-R-T-U-R-E'. Confirming that the war prisoners are being tortured. There are so many examples where people tried to establish secret communication by hiding data. This show how crucial is to pass information without arising suspicion at first sight. It is also seen that these techniques are used by criminals and terrorist.

1.1 Cryptography

Throughout the course of history it been seen the art of hiding information is carried out in one form or another. Julies Caesar used some form of encryption to convey messages to his war generals. This was a simple substitution cipher where each character is replaced with some other character from alphabet. This cipher is popularly known as 'Caesar cipher'. As the technology developed more n more complex form of ciphers came into existence. Cryptography also played an important role in World War I and II

The process of converting Plain text into cipher text is known as Encryption, and the process of converting the cipher text into original plaintext is known as Decryption. Encryption is generally follows a pattern or algorithm with specific key (or set of keys) to encrypt data. .Cryptography becomes a necessity when data transfer takes place over any entrusted medium, including any network, especially the Internet. Cryptography can also be used a tool for user authentication as well as for shielding the data from alteration and the prime issue of data theft. The Kerckhoff's principle states that the secrecy of message must be dependent on secrecy keys rather than the crypto-system or algorithm [1].

There are three types of cryptographic schemes

- Secret key cryptography or symmetric key cryptography
- Public key cryptography or asymmetric key Cryptography
- Hash functions

In symmetric key or secret key cryptography both sender and receiver use same set of keys to encrypt and decrypt data. Its usefulness decreases when user wants to exchange keys on unsecure channel. Different set of keys must be maintained for different user. If there are 'N' number of peoples on group then it is necessary to administer N*(N-1)/2 set of keys [1].

Public key cryptography or Asymmetric key cryptography uses different set of key to encrypt data. When sender wants to send data over an unsecure channel he first encrypts data using receiver's public key. After receiving the encrypted data receiver uses his private key to decrypt data. The important aspect here is use of public and private key. Because of this there is no need to exchange the keys before communication is carried out In this scheme public key may be known to everyone but secret key is different for every user. The system is designed in such a way that even if the public key is revealed to attacker without the private key the encrypted data is useless [1].

A hash function takes a relatively arbitrary amount of input and produces a fixed size output. The properties of some hash functions can also be utilised to verify and confirm the source

of a file, or the integrity of any network packet, or arbitrary data, hence contributing to an increase in the security level of a system administrator's network. By using the message digests generated by a cryptographic hash function a system administrator can detect unauthorized changes in files. This is especially important when safeguarding critical system binaries and sensitive databases. It is seen often that hash functions are used with standard cryptographic schemes to verify the source of data. Mathematical transformations are used to irreversibly "encrypt" information [1].

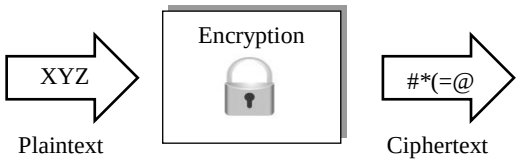


Figure 1.1: Basic process of Encryption

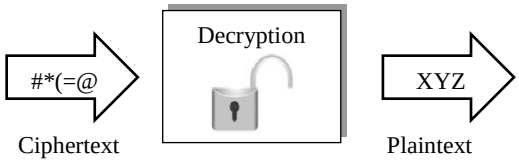


Figure 1.2: Basic process of Decryption

1.2 Steganography

Steganography comes from ancient Greek words stegnos and graphia which means 'concealed writing'. The goal of Cryptography is to protect message caught by unauthorized person. While goal of steganography is concealing the existence of message. It can be said steganography obtain Security through obscurity. If the message is not detected it makes both sender and receiver invisible. The goal here is to obtain invisibility rather than security. It is possible to apply steganography on any digital media like audio, video, images, network packets etc. The medium used to hide data is known as Cover object or cover data and the result of steganography is Stego object which contains secret information.

Depending on the strength of stego object against attack, Steganographic techniques can be classified as 'Fragile' or 'Robust'. The fragile technique implies that if the stego object is modified the information inside it is lost. The aim of robust technique is to embed information which is invulnerable to file modifications. In case of images the modification can simply be sharpening, blurring, cropping, rotating etc. [2]

Most commonly used steganography techniques are LSB substitution and Transform Domain techniques. These techniques are described below.

LSB Substitution is a very simple technique in which the Cover is broke into bytes and LSB of each byte is replaced with 1 message bit. The message must be carefully hidden in LSB such that Human Sensory System will not be able to pick message. The Human Sensory systems are Human Visual system (HVS) and Human Auditory systems. One can change up to 4th LSB in images without distorting image pixels. Above 4th LSB human eye can easily pick up the changes made in image. LSB substitution is fragile as it is susceptible to various File modification attacks. [3]

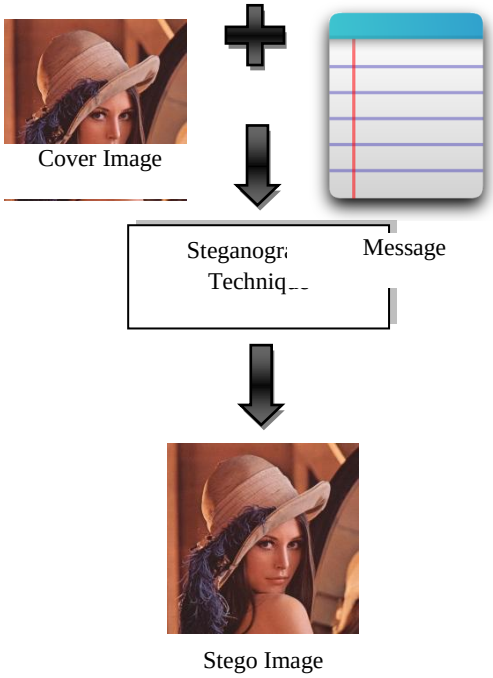


Figure 1.3 Basic Steganography Process

Transform Domain techniques make use of complex transformations to hide message in significant areas of cover object. These transformations can be DCT (Discrete Cosine Transform), Wavelet transform or Fourier Transform to hide data in Cover object. Transform domain techniques are more resistant to image processing attacks [4]

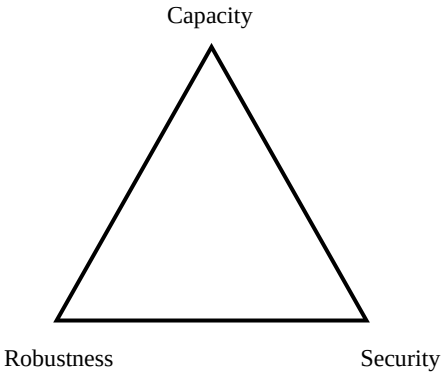


Figure 1.4: Steganography Triangle

Several critical points need to be scrutinized while studying Steganographic systems. They are Steganographic robustness, capacity, and security [5, 6]. The conjugation between these characteristics can be depicted by the steganography triangle shown in Figure 1.4. It represents the balance of the desired characteristics in relation with the Steganographic method. They are co-dependent on each other and in order to enhance the quality of an element, one or both of the other elements need to be degraded. Robustness refers to an embedded message's ability to survive either a deliberate attack by a suspecting third-person or the random corruption by noise in some transmission phase. Capacity gives an account to the maximum number of bits which could be inserted in the image, without the stego image being indistinguishable and visually intact. Security depends on the ability of an embedding carrier to remain undetected.

2. PROPOSED METHOD

Out of both Cryptography and steganography neither of them provides efficient security. Each one has their own unique advantages. But combining both provides with better solution to security. In this paper AES-128 is used to encrypt the message before it is inserted into image. After the message is encrypted then it is embedded in to image using Pseudo random numbers in LSB of image.

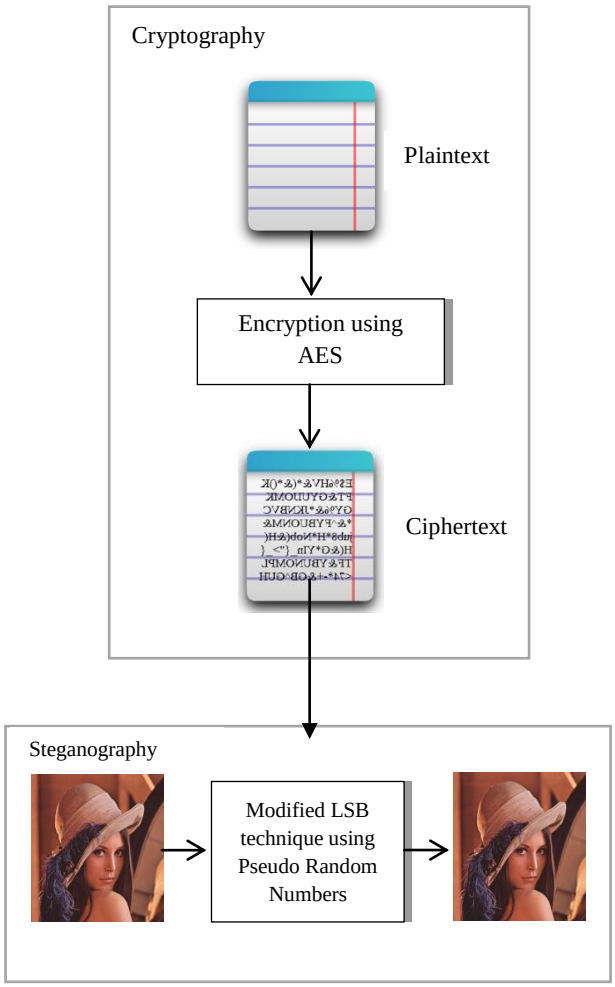


Figure 2.1: Proposed Method

2.1 Selecting Image

Selection of cover image is very important and mostly depends on the Steganography algorithm. The images falls under two main categories: Lossy and Lossless. The image formats which removes extra noise after certain compression algorithm are considered as lossy. JPEG is an example of lossy compression; it removes the unwanted noise from image to high compression. To a computer an image is simply a file that represents different intensities of light throughout the image.

Graphic Interchange Format (GIF) is one of the machine independent file format for images. It has a bit depth of 8 bit. But this is the main problem with GIF format it uses 256 color palette, so even if a single LSB is changed color of the palette changes significantly. Which make it difficult to use LSB Steganography in 8 bit color GIF. On the other hand greyscale GIF contains 256 grey color palettes which make it suitable to use in LSB steganography [7].

Portable Network Graphics (PNG) is a lossless image which is a type of bitmapped format. PNG was created to improve GIF. This format is widely used on web so there is less chance to arise suspicion. It can carry large amount of data. LSB in PNG is suitable where there is stress on data hiding capacity rather than security [7].

Bitmap (BMP) also known as Device Independent Format (DIB). 8 bit color BMP is not good for steganography as they only contain 256 color palettes. On the other hand 24 bit bitmap are very are ideal medium to use LSB steganography. It contains 3 color values for each color which are Red, Green and Blue (RBG). 24 bit color bitmap is ideal for this application as we are modifying out of 4 LSB's in image. There are 16,777,216 color variants in 24 bit Bitmap. If the change is made up to 4th LSB in bitmap from all RGB values of a pixel. It is very likely that it will distort the pixel by huge difference. But if we use only one color component out of RGB say blue in this case, the distortion of pixels will be very less. Hence only blue pixel will be used in this paper in order to achieve the LSB Steganography.

2.2 Description of AES-128

AES-128 is used in this paper to encrypt the text before it is embedded into image. AES-128 is a block cipher. It has fixed block size of 128 bit i.e. 16 bytes. AES used fixed key length of 16 bytes. The number 128 in AES represents a transformation round which are 10 rounds in AES-128, 12 in AES-196 and 14 in AES-256. There are four types of transformation in AES which are explained below.

Byte Substitution (SubBytes) transformation is a non-linear byte substitution which runs separately on each byte of the state using a Substitution Table (S-Box). This steps uses look up table which find a replacement byte for a chosen byte from input state array. In the Byte Substitution method, the input byte is used to index a row/column in a table to retrieve the substituted value this method works upon each byte without any dependency. A better substitution algorithm is suggested for this stage to get better output [7].

The Shift Rows (ShiftRows) stage provides a simple "permutation" of the data, whereas the other steps involve substitutions. Other steps involve substitutions whereas the shiftrows stage offers only a permutation of data. Further, since the state is treated as a block of columns, In this step a diffusion of values between columns occurs as the row shift moves an individual byte from one column to the other, which in terms of linear distance is a multiple of 4 bytes, it also ensures that the 4 bytes of a single column are scattered to four separate columns [7].

Mix Columns (MixColumns) can also be implemented by expressing the transformation on each column as 4 equations which will be used to calculate the new bytes for that particular column. Computation using this method is restricted to XORs, conditional XORs & shifts (for the modulo reduction).The aim of this process is to scatter the 128 bit input block. Inverse of the matrix containing large coefficients is used for decryption computation, making the process slower to implement and comparatively harder [7].

Add Round Key (AddRoundKey) stage involves a bitwise XOR operation of the current block coupled with a piece of the expanded key. This step can only be used at start and end of each round as it utilises a portion of the key and hampers the result, non-adherence to this rule can lead to an undo effect of other previous steps [7].

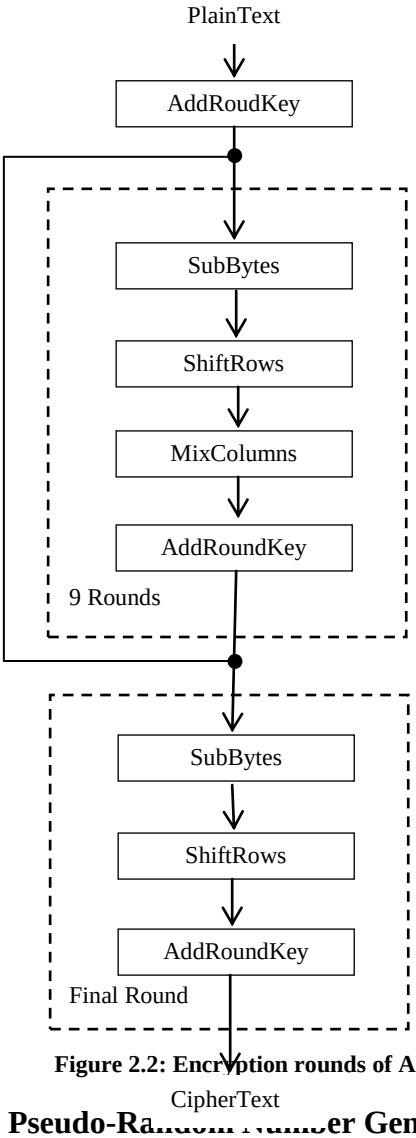


Figure 2.2: Encryption rounds of AES

2.3 Pseudo-Random Number Generation

For this application the random number generator in C# is used which is based on Donald E. Knuth's subtractive random number generator algorithm [8]. But it is always better to use cryptographically strong Random number generator. The goal here is to get Pseudo Random number sequence of numbers 4, 5 and 6 using the same seed/key whose length is equal to message bits. The numbers 4, 5 and 6 here represents which LSB will be used to insert the message bits. Once the Number sequence is generated the process is moved to next step which is replacement of message bits with Image LSB's.

3. LSB Steganography Using Pseudo-Random Numbers

To Perform LSB steganography a 24-bit bitmap is used in this paper. As only Blue component from RGB values is needed. One can use any of the color components from the RBG pixels. So rather than replacing LSB from all color components to insert message bits only 4th, 5th or 6th LSB is replaced from Blue color.

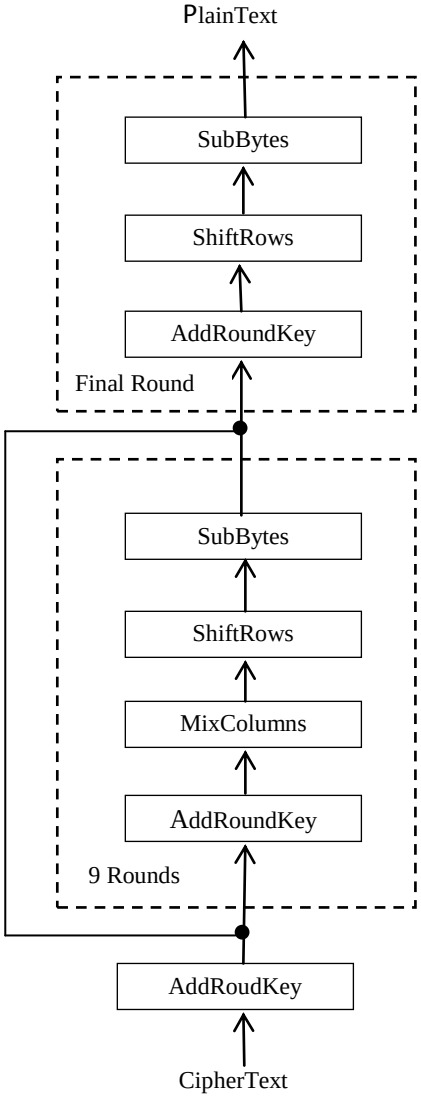


Figure 2.3: Decryption rounds of AES

For example if the letter 'a' is to be inserted in image. Letter 'a' has binary value '01100001' will utilizes only a single color component. To achieve this one can use any color component out of red, green or blue color. Consider the following grid of 8 pixels which are adjacent in 24bit Bitmap image. In the greed each color represents corresponding pixel color.

Now the sequence number 45546456 is generated by Pseudo-Random Number generator. Then the corresponding LSB of the Blue pixel is replaced. Suppose 0 from message bit is to be adjusted from the pixel value 11101011 the 4th LSB is replace by 0. The modified pixel value will be 11100011. And grid will be changed as shown in figure below (Bold numbers represents the changed value of LSB).

10001101	01110110	11101011
11010010	10000011	00110011
10111010	11101011	10000001
10001101	01110110	11101011
11010010	10000011	00110011
10111010	11101011	10000001
10001101	01110110	11101011
11010010	10000011	00110011

Figure 2.3: original grid 24 pixels from Image

Here only 5 LSB's are replaced with message bits as the rest of message bits were already present in the pixels. There is always fifty percent chance that the required value is already present. So there is no need to change those values. Even if the changes are made in LSB's of all blue pixels the change will be very little.

10001101	01110110	11100011
11010010	10000011	00110111
10111010	11101011	10000101
10001101	01110110	11100011
11010010	10000011	00110001
10111010	11101011	10000001
10001101	01110110	11101011
11010010	10000011	00110011

.

Figure 2.4: Modified grid 24 pixels from Image

3.1.1 Proposed Algorithm for Steganography

1. Select a 24-bit Bitmap Image.
2. Generate a seed using Key used for AES.
3. Generate a pseudo- random Number sequence containing only 4, 5 and 6 using seed whose length equal to Message bits.
4. Select only blue Pixel to insert data.
5. Select a bit from message to insert into LSB.
6. Select the LSB position from Random Sequence.
7. Replace the message bit with LSB position.
8. Repeat the same Process for the rest of message.

4. RESULT ANALYSIS

When RGB histogram of both cover image and Stego image are compared it doesn't show any significant difference. To test this 28KB of data is embedded in Lena.bmp image of size 512x512 using the proposed LSB technique in this paper. The histograms of Cover Image and Stego image are shown in figures below.

(a) **(b)**

Figure 3.1: (a) Cover Image (b) Stego Image

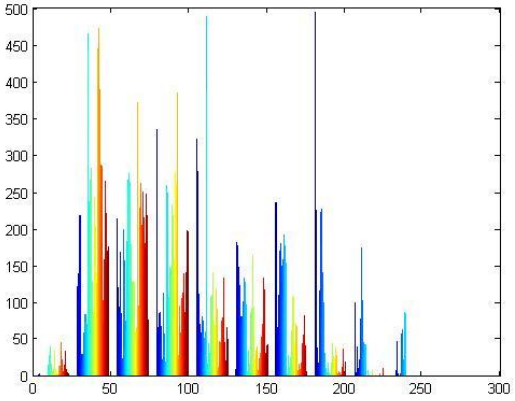


Figure 3.2: RGB Histogram of Cover Image

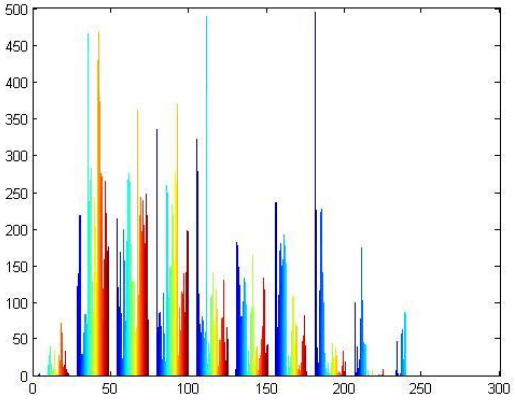


Figure 3.3: RGB Histogram of Stego Image

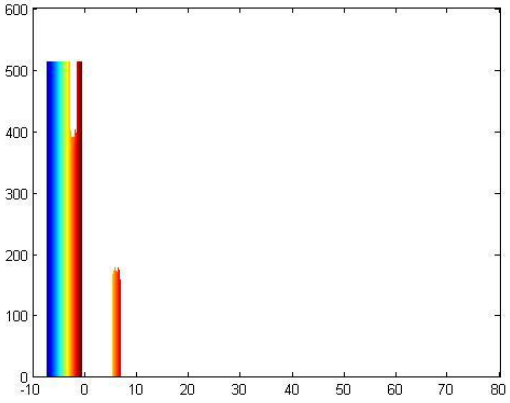


Figure 3.4: Difference Histogram

4.1 Mean Square Error (MSE)

The MSE between images I_1 (m,n) and I_2 (m,n) is given by:

$$MSE = \frac{\sum_{m,n}[I_1(m,n) - I_2(m,n)]^2}{M * N}$$

M and N are the number of rows and columns of respective image [9]. When MSE is calculated for lena.bmp with 28kb of data embedded in the image MSE is 3.0147

4.2 Peak Signal-to-Noise Ratio (PSNR)

PSNR is calculated using MSE. PSNR is a good measure to compare restoration of images. It is measured in decibel (db). PSNR of an image can be calculated as:

$$PSNR = 10log_{10}\frac{R^2}{MSE}$$

Here the maximum fluctuation in the input image data type is represented by R [9]. The PSNR of the same image lena.bmp is 43.335 db.

4.3 Results with other Images

The proposed method was tested against other several Images with 28kb of data, whose results are shown in table below.

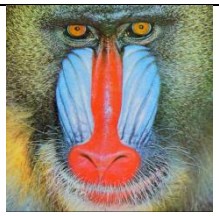
Image (512x512)	PSNR in db.
	43.335
	44.902
	43.026
	43.893
	44.036

Table 1: PSNR of Various Images Using Proposed Method

5. CONCLUSION

The proposed method provides security of both Steganography and Cryptography. Use of Pseudo-Random Number sequence helps the message bit to spread across 3 (4^{th}, 5^{th} and 6^{th}) LSB's randomly. Without the proper key it is almost impossible to detect which LSB holds the message bits. By using only a single RBG component from the pixel the distortion produced by Steganography is negligible, But at the cost of capacity to hold the data. By removing the need of last LSB to insert data, the proposed method holds strong against LSB removal attack. Even if the message is extracted from image, it won't make any sense without key used for AES to encrypt message. PSNR of the images is acceptable value and shows that the Image quality is preserved at good level.

Security can be increased by selecting random color component from RBG of pixel every time when message bit is inserted.

6. ACKNOWLEDGEMENT

I sincerely thank my research guide Prof A. K. Gulve Department of Computer science and engineering, Government College of engineering Aurangabad, for his guidance, encouragement, valuable suggestions and moral support throughout the progress of this research.

7. REFERENCES

[1] Unik Lokhande. "An Effective Way of using LSB Steganography in images along with Cryptography." International Journal of Computer Applications 88(12):25-29, February 2014. Published by Foundation of Computer Science, New York, USA.

[2] Shashikala Channalli, Ajay Jadhav, "Steganography an Art of Hiding Data" International Journal on Computer Science and Engineering Vol.1 (3), 2009, 137-141.

[3] J. Cummins, P. Diskin, S. Lau and R. Parlett, "Steganography and Digital Watermarking", School of Computer Science, the University of Birmingham, 2004.

[4] N. F. Johnson and S. Jajodia, "Steganalysis: The Investigation of Hidden Information", IEEE Information Technology Conference, September 1998.

[5] W. Luo, F. Huang, J. Huang, Edge adaptive image steganography based on LSB matching revisited, IEEE Trans. Inf. Forens. Security 5 (2) (2010)201-214.

[6] Ge Huayong, Huang Mingsheng, Wang Qian, "Steganography and Steganalysis Based on Digital Image", IEEE Trans. International Congress on Image and Signal Processing, (2011) 252-255.

[7] Parag Kadam, Mangesh Nawale, Akash andhare, Mukesh Patil "Separable Reversible Encrypted Data Hiding in Encrypted Image Using AES algorithm and Lossy Technique" Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering February 21-22 2013 IEEE.

[8] Donald E. Knuth "The art of computer programming Vol. 2 (3^{rd} Edition): Seminumerical Algorithms"

[9] V. Lokeswara Reddy, Dr. A. Subramanyam, Dr.P. Chenna Reddy, "Implementation of LSB Steganography and its Evaluation for Various File Formats" Int. J. Advanced Networking and Applications Volume: 02, Issue: 05, Pages: 868-872 (2011).