Volume 5, Issue 1, January 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Novel Approach of LSB Based Steganography Using Parity Checker

Tahir Ali
Department of Computer Science &Engg.,
MIET, Meerut, India

Amit Doegar
Department of Computer Science,
NITTTR, Chandigarh, India

Abstract— For hiding secret information in images, there exist a large variety of steganographic techniques, some are more complex than others and all of them have respective strong and weak points. In the proposed method, all pixels of the cover image can be used but message bit is stored in LSB of one of the three color components, Red(R), Green(G), Blue(B) based on the parity of three LSBs of R, G, B components of 24-bit color image.The proposed work uses the concept of parity method for hiding and recovering secret data or information. The proposed method can hide large volume of data in a single RGB image with changes in only few pixels of input image retaining the advantages and discarding the disadvantages of traditional LSB method.

Keywords— Steganography, cover image, stego image, even odd Parity.

I. Introduction

Since the rise of the Internet one of the most important factors of information technology and communication has been the security of information. Cryptography was created as a technique for securing the secrecy of communication and many different methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately it is sometimes not enough to keep the contents of a message secret, it may also be necessary to keep the existence of the message secret. The technique used to implement this, is called steganography[3].

Steganography is the art and science of invisible communication. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The word steganography is derived from the Greek words “stegos” meaning “cover” and “grafia” meaning “writing” defining it as “covered writing”. In image steganography the information is hidden exclusively in images[3].

The four main categories of file formats that can be used for steganography are text, image, audio and video. Given the proliferation of digital images, especially on the Internet, and given the large amount of redundant bits present in the digital representation of an image, images are the most popular cover objects for steganography.

Image Steganography

As stated earlier, images are the most popular cover objects used for steganography. In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganographic algorithms exist.

Image steganography techniques can be divided into two groups: those in the Image Domain and those in the Transform Domain. Image – also known as spatial – domain techniques embed messages in the intensity of the pixels directly, while for transform – also known as frequency – domain, images are first transformed and then the message is embedded in the image.

Image domain techniques encompass bit-wise methods that apply bit insertion and noise manipulation and are sometimes characterised as “simple systems”. The image formats that are most suitable for image domain steganography are lossless and the techniques are typically dependent on the image format.

Steganography in the transform domain involves the manipulation of algorithms and image transforms. These methods hide messages in more significant areas of the cover image, making it more robust. Many transform domain methods are independent of the image format and the embedded message may survive conversion between lossy and lossless compression

Least Significant Bit Steganography

Least significant bit (LSB) insertion is a common, simple approach for embedding information in a cover image. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data. For example a grid for 3 pixels of a 24-bit image can be as follows:
(00101101 00011100 11011100)

(10100110 11000100 00001100)
(11010010 10101101 01100011)

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

(0010110**1** 0001110**1** 1101110**0**)
(1010011**0** 1100010**1** 0000110**0**)
(1101001**0** 1010110**0** 01100011)

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye - thus the message is successfully hidden.

In its simplest form, LSB makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion. For this reason, LSB steganography has also been developed for use with other lossless image file formats such as PNG.

II. RELATED WORK

An Many research works have been carried out on LSB based Steganography. Different researchers employed different techniques for the purpose of hiding secret data in a cover image. Following are the few related works carried out by various research groups:

Md. Olioul Islam in [1] presented a new steganography technique to hide large data in Bitmap image using stream builder and parity checker. This method uses the concept of odd and even parity for embedding and extracting of secret message. This method is an improvement of Least Significant Bit (LSB) method for hiding information in images.

Rajkumar Yadav et al in [2] presented a new steganography technique for hiding data in images using parity checker. This method uses the concept of odd and even parity for insertion and retrieval of message. This method is an improvement over earlier methods like least significant bit method and 6th, 7th bit method for hiding information in images.

Alam et al in [7] presented a description on how one can use the human vision system and pure steganography to increase the size of the data that we want to embed in the image. They focus on the property of human vision system that help to increase the amount of data hiding in the bitmap (.bmp) and JPEG (.jpg) images practically. They enhance the work of LSB and try to come out with a better result for both image quality and the amount of data can be hidden inside it. They come out with two approaches; first one is the 3-3-2 approach without any limitations on the type of images being used and can reach up to 33.3% of size of hidden data, and the second one is the 4-4-4 approach which increase the amount up to 50% of hidden data from the size of image but with certain limitations on the type of images chosen.

Chaudhary et al in [9] proposed an improved steganography approach for hiding text messages in lossless RGB images. The objective of this work is to increase the security level and to improve the storage capacity with compression techniques. The security level is increased by randomly distributing the text message over the entire image instead of clustering within specific image portions. Storage capacity is increased by utilizing all the color channels for storing information and providing the source text message compression.

Roy et al in [10] presented the evaluation of different algorithms for digital image steganography both in the spatial and transform domain like LSB substitution, OPAP, Pixel Indicator Technique, F5 etc. and tries to put light on some possible future research directions in the topic of consideration.

Amitava Nag et al in [14] presented a novel technique for Image steganography based on LSB using X-box mapping where we have used several Xboxes having unique data. The embedding part is done by this steganography algorithm where we use four unique X-boxes with sixteen different values (represented by 4-bits) and each value is mapped to the four LSBs of the cover image. This mapping provides sufficient security to the payload because without knowing the mapping rules no one can extract the secret data (payload).

Chen et al in [15] presented Multi-bit minimum error replacement (MER) method that can embed multi-bit logo/secret data into k least significant bits (LSBs) of cover data only introduces minimum embedding error (MEE). However, k-LSBs MER suffers from weak anti-forensics. Moreover, it is unfortunate because other previous steganography works have seldom considered both large embedding capacity and high image quality. Therefore, this work proposes an anti-forensic steganography system using multi-bit adaptive embedding algorithm with flexible bit location to overcome the problem of forensics and to achieve high performance includes both large embedding capacity and high image quality.

III. PROPOSED TECHNIQUE

The proposed method uses non-filtering steganographic algorithm which is a most popular and least vulnerable steganographic technique based on LSB. In the proposed method, all pixels of the cover image can be used but message bit is stored in LSB of one of the three color components, Red(R), Green(G), Blue(B) based on the parity of three LSBs of R, G, B components of 24-bit color image. The message to be embedded is converted into bits by representing each character by 8 bits ASCII code, for example “HELLO” will be embedded in the cover image as “01001000 01000101 01001100 01001100 01001111”.

The overall operation includes the following steps:

1. Determine the message to be embedded
2. Calculate the message size
3. Select a cover image of sufficient size
4. Embeds the size of embedding message using the first P pixels(P depends on the size of the message).
5. Embeds the message.

Algorithm Design:

Every pixel of 24-bits has three color components i.e R, G, B of 8-bits each. First it collects the LSB of three color components and makes a group of three bits. Now the sequence of these three bits may have either even number of 1's or odd number of 1's. If the sequence of three bits contains even number of 1's then it is called as even parity otherwise it is called as odd parity. The embedding in the proposed algorithm depends on the message bit and the parity generated by the LSB of each color components.

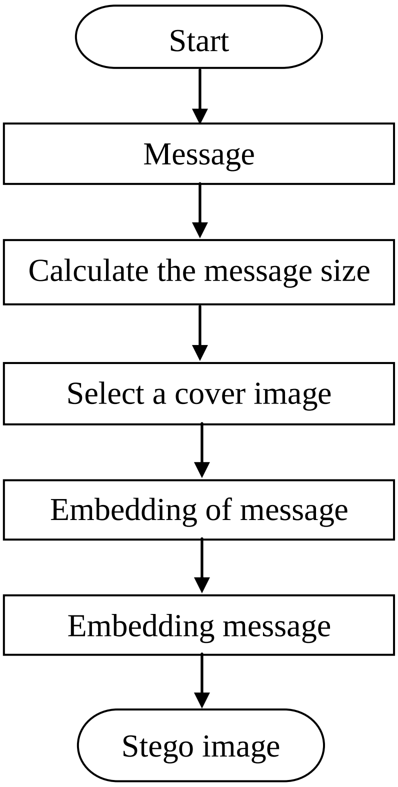


Figure 3.1: Overall Operation Of Proposed Method

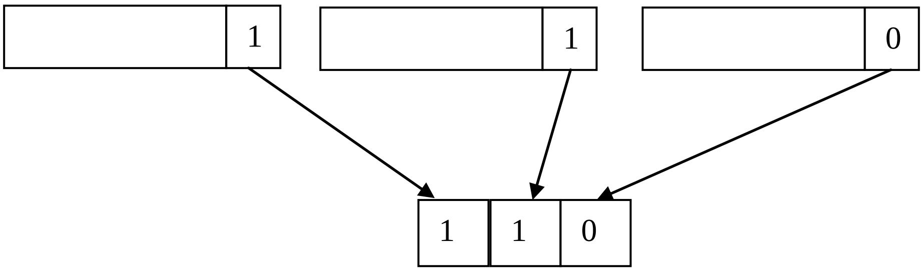


Figure 3.2: Even parity

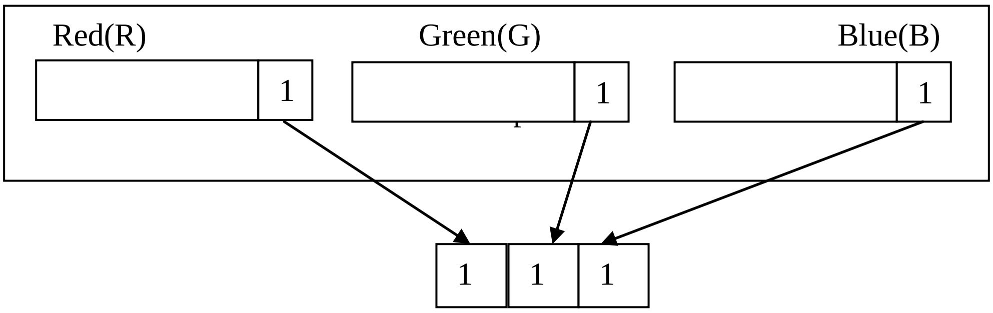


Figure 3.3 Odd parity

A. Embedding Process

The embedding process can be depicted by the following table:

Table 3.1 Embedding process

LSBs of Color Component of a pixel			Parity	Message Bit	Action Performed	Resulting LSBs of Color Components			Resulting Parity
R_{LSB}	G_{LSB}	B_{LSB}				R_{LSB}	G_{LSB}	B_{LSB}	
0	0	0	Even	0	No Change	0	0	0	Even
0	0	0	Even	1	Reverse R_{LSB}	1	0	0	Odd
0	0	1	Odd	0	Reverse G_{LSB}	0	1	1	Even
0	0	1	Odd	1	No Change	0	0	1	Odd
0	1	0	Odd	0	Reverse B_{LSB}	0	1	1	Even
0	1	0	Odd	1	No Change	0	1	0	Odd
0	1	1	Even	0	No Change	0	1	1	Even
0	1	1	Even	1	Reverse R_{LSB}	1	1	1	Odd
1	0	0	Odd	0	Reverse G_{LSB}	1	1	0	Even
1	0	0	Odd	1	No Change	1	0	0	Odd
1	0	1	Even	0	No Change	1	0	1	Even
1	0	1	Even	1	Reverse B_{LSB}	1	0	0	Odd
1	1	0	Even	0	No Change	1	1	0	Even
1	1	0	Even	1	Reverse R_{LSB}	0	1	0	Odd
1	1	1	Odd	0	Reverse G_{LSB}	1	0	1	Even
1	1	1	Odd	1	No Change	1	1	1	Odd

B. Embedding Algorithm

```
I. Get the message to be embedded
II. Calculate the size of the message(no. of character or byte)
III. Assume P= no of pixels to store the size of the message
IV. Represent the size of the message using P bits
V. Convert the message into bit stream using 8-bits ASCII codes
VI. Append the size of the message (P bits) with the message bit stream
VII. Let N=length of resultant message in bits
VIII. Select a cover image
IX. I = 1
X. Collect three LSBs say $R_{LSB}$, $G_{LSB}$, $B_{LSB}$ for R, G, B color components of $I^{th}$ pixel
XI. Determine the parity of $R_{LSB}$, $G_{LSB}$, $B_{LSB}$ (Even or Odd)
XII. Get message bit (0 or 1)
XIII. If message bit is 0 and parity is even, do nothing
XIV. If message bit is 0 and parity is odd, reverse the value of any of the LSB($R_{LSB}$, $G_{LSB}$, $B_{LSB}$) in alternate fashion
XV. If message bit is 1 and parity is even, reverse the value of any of the LSB($R_{LSB}$, $G_{LSB}$, $B_{LSB}$) in alternate fashion
XVI. If message bit is 1 and parity is odd, do nothing
XVII. I= I+1
XVIII. If (I <= N) go to step X else go to step XIX
XIX. END
```

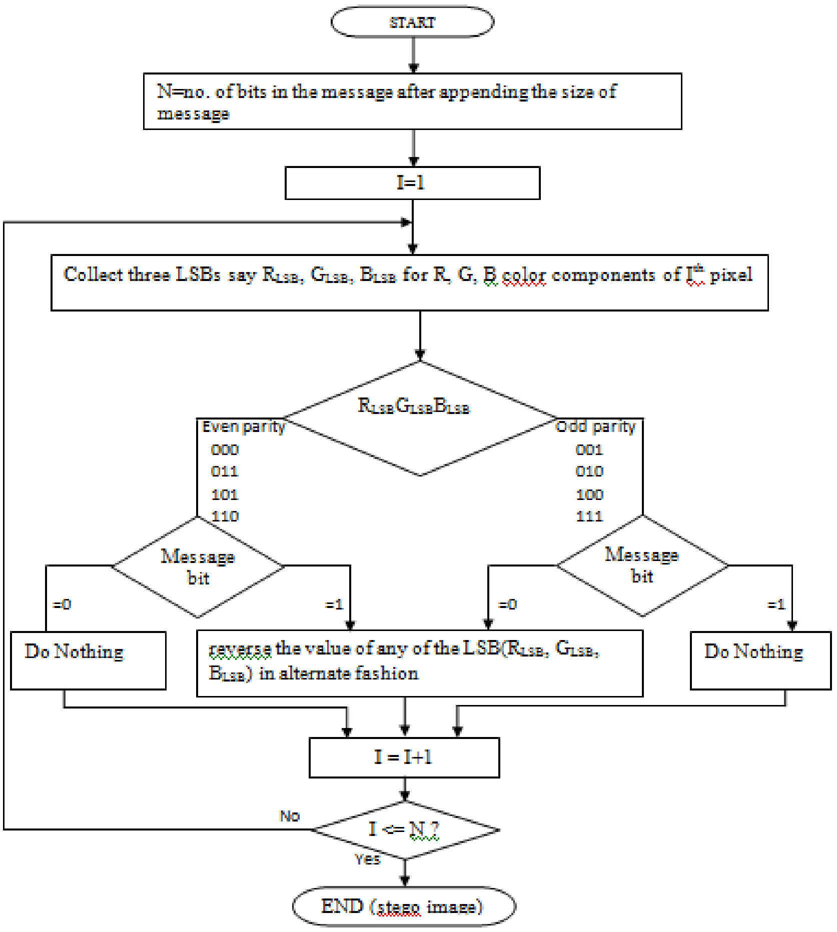

START
N=no. of bits in the message after appending the size of message
I=1
Collect three LSBs say RLSB, GLSB, BLSB for R, G, B color components of Ith pixel
RLSB GLSB BLSB
Even parity
000
011
101
110
Odd parity
001
010
100
111
Message bit
=0
=1
Message bit
=0
=1
Do Nothing
reverse the value of any of the LSB(RLSB, GLSB, BLSB) in alternate fashion
Do Nothing
I = I+1
No
I <= N ?
Yes
END (stego image)

Fig.3.4 Flowchart showing the insertion of bit "1" and "0"

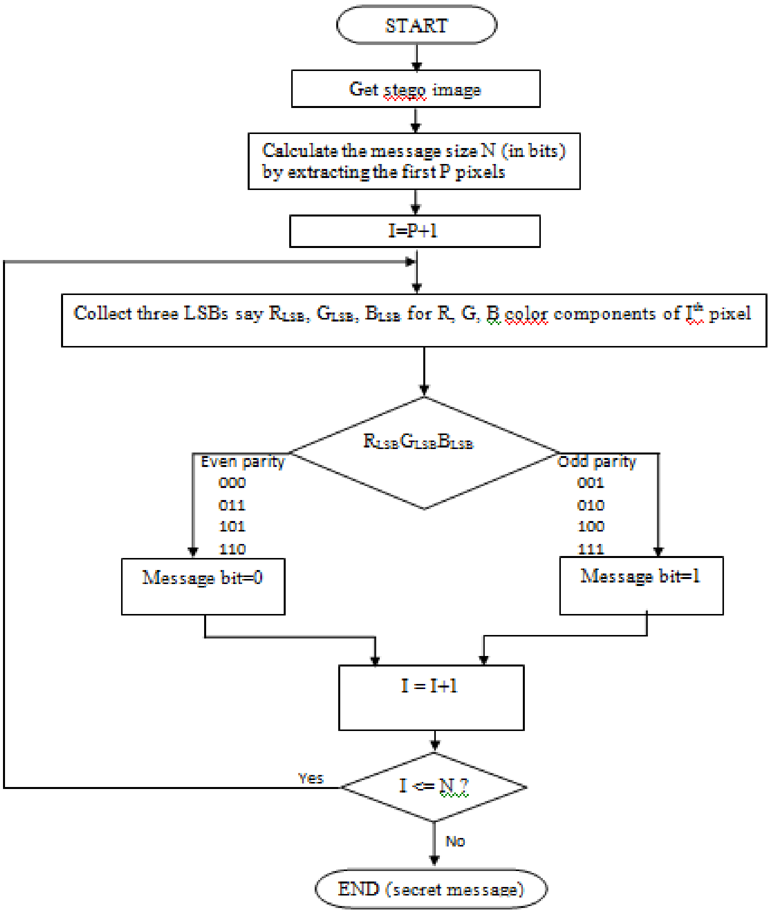

START
Get stego image
Calculate the message size N (in bits) by extracting the first P pixels
I=P+1
Collect three LSBs say RLSB, GLSB, BLSB for R, G, B color components of Ith pixel
RLSB GLSB BLSB
Even parity
000
011
101
110
Odd parity
001
010
100
111
Message bit=0
Message bit=1
I = I+1
Yes
I <= N ?
No
END (secret message)

Fig. 3.5 Flowchart showing extraction of hidden message

C. Extracting Process

The extracting process can be depicted by the following table:

Table 3.2 Extracting process

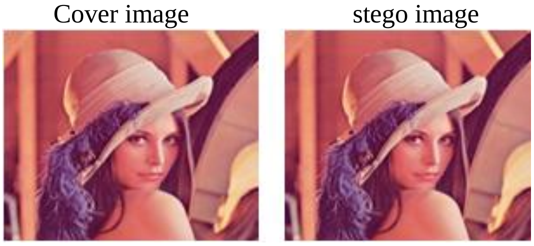
LSBs of Color Component of a pixel of stego image			**Parity**	**RetrievedM essage Bit**
R_{LSB}	**G_{LSB}**	**B_{LSB}**		
0	0	0	Even	0
0	0	1	Odd	1
0	1	0	Odd	1
0	1	1	Even	0
1	0	0	Odd	1
1	0	1	Even	0
1	1	0	Even	0
1	1	1	Odd	1

D. Extracting Algorithm

```
   I.  Get stego image
  II.  I=1
 III.  Collect three LSBs say $R_{LSB}$, $G_{LSB}$, $B_{LSB}$ for R, G, B color components of $I^{th}$ pixel
  IV.  Determine the parity of $R_{LSB}$, $G_{LSB}$, $B_{LSB}$ (Even or Odd)
   V.  If the parity is even store the 0 as the message bit else store 1 as the message bit
  VI.  I = I+1
 VII.  If ( I<= P) then go to step III else go to step VIII
VIII.  Convert the received message into a integer value say n
  IX.  Set N = n*8
   X.  Collect three LSBs of $I^{th}$ pixel
  XI.  Determine the parity of $R_{LSB}$, $G_{LSB}$, $B_{LSB}$ (Even or Odd)
 XII.  If the parity is even store the 0 as the message bit else store 1 as the message bit
XIII.  I = I+1
 XIV.  If ( I<=N) go to step X else go to step XV
  XV.  Convert the received bit stream into message
 XVI.  END
```

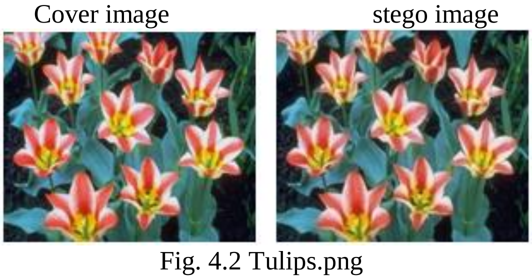
IV. EXPERIMENTAL RESULTS

The proposed algorithm was implemented in MATLAB(R2012a) running on Windows 8 operating system. This chapter provides details of the result obtained by executing proposed algorithm with different cover images given in Figure 4.1 and 4.2(left side). The corresponding stego images are also shown in Figure 4.1 and 42 (right side).



Cover image stego image

Fig. 4.1 Lena.png



Cover image stego image

Fig. 4.2 Tulips.png

Table 4.1

COVER IMAGE	**IMAG E SIZE**	**HIDDEN MESSAGE**	**MESSAGE SIZE**	**NO. OF ALTERED BITS IN COVER IMAGE**
Lena.png	462 KB	The Proposed algorithm was run to hide this message in the test image Lena.png	78 Characters or 624 Bits	316/ (462* 1024*8) = **0.0083%**
Tulips.png	663 KB	Steganography is the art and science of invisible communication.	64 Characters or 512 Bits	263/ (663* 1024*8) = **0.0048%**

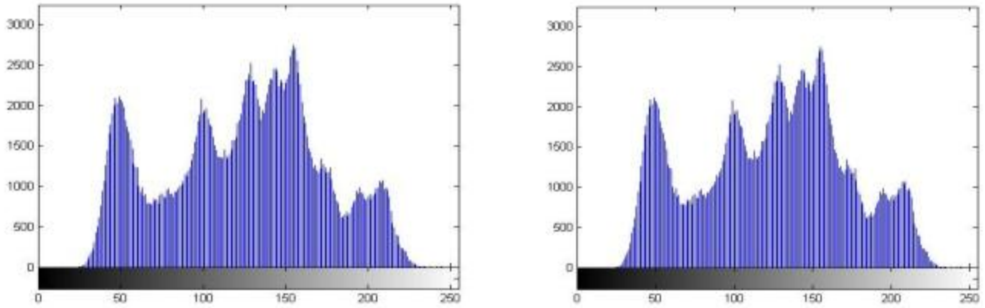


Fig. 4.3 Histogram of Lena.png(cover and stego image)

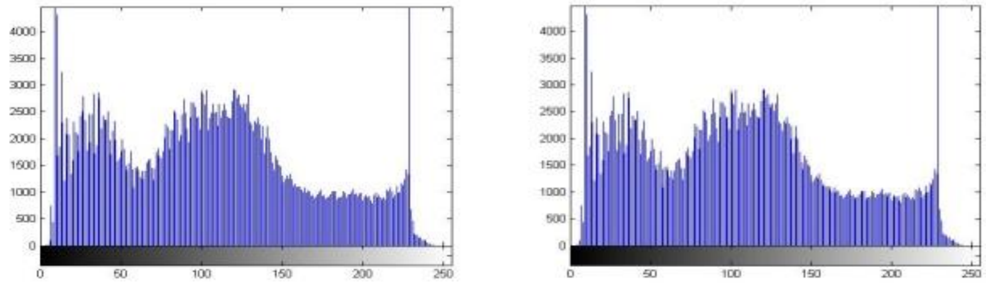


Fig. 4.4 Histogram of Tulips.png(cover and stego image)

V. CONCLUSIONS

In the proposed work, we provide a new concept of image steganography based on modified LSB approach. The proposed algorithm is based on the parity of LSBs of three color components i.e R, G, B. The main goal of the proposed method is to increase the size of the message to be embedded with the image and also make the technique difficult to the unauthorized person to determine the presence of secret message.

REFERENCES

[1] Md. Olioul Islam "A High Embedding Capacity Image Steganography using Stream Builder and Parity Checker" 15th International Conference on Computer and Information Technology (ICCIT) on pp. 458-463, IEEE, December 2012.

[2] Rajkumar Yadav, Rahul Rishi & Sudhir Batra, "A New Steganography Method for Gray Level Images using Parity Checker", International Journal of Computer Applications (0975-8887) Volume 11-No. 11, December 2010.

[3] T. Morkel, J.H.P. Eloff, M.S. Oliver, "An overview of image steganography", Pretoria, South Africa, Information and Computer Security Architecture (ICSA) Research Group, pp. 1-11, June 2005.

[4] Neil F Johnson, Sushil Jajodia, "Exploring Stenography: Seeing the Unseen", IEEE Computer, pp 26-34, Feb 1998.

[5] Currie, D.L. & Irvine, C.E., "Surmounting the effects of lossy compression on Steganography", 19th National Information Systems Security Conference on pp. 194-201, Baltimore, Md, October 1996.

[6] D. Artz, "Digital Steganography: Hiding Data within Data", IEEE Internet Computing, vol. 5, no. 3, pp. 75-80, June 2001.

[7] Alam, Shahzad, S. M. Zakariya, and M. Q. Rafiq. "Analysis of Modified LSB Approaches of Hiding Information in Digital Images." Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on pp. 280-285, IEEE, September 2013.

[8] Rong –Jian Chen, Shi-Jinn Horng, “Multi-bit Adaptive Embedding Algorithm for Anti-Forensic Steganography” Biometrics and Security Technologies (ISBAST), International Symposium on pp. 82-89, IEEE, March 2012.

[9] Chaudhary, Ankit, and Jaldeep Vasavada. "A hash based approach for secure keyless image steganography in lossless RGB images." Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on pp. 941-944, IEEE, October 2012.

[10] Roy, Ratnakirti. "Evaluating image steganography techniques: Future research challenges." Computing, Management and Telecommunications (ComManTel), 2013 International Conference on pp. 309-314, IEEE, January 2013.

[11] Akhtar, Nadeem, Pragati Johri, and Shahbaaz Khan. "Enhancing the Security and Quality of LSB Based Image Steganography." Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on pp. 385-390, IEEE, September 2013.

[12] Dagadita, Monica Adriana, Emil Ioan Slusanschi, and Razvan Dobre. "Data Hiding Using Steganography." Parallel and Distributed Computing (ISPDC), 2013 IEEE 12th International Symposium on pp. 159-166, IEEE, June 2013.

[13] M. Zawawi, R. Mahmod, N. Udzir, F. Ahmad, and J. Desa, "Active warden as the main hindrance for steganography information retrieval" International Conference in Information Retrieval Knowledge Management (CAMP), pp. 277-280, March 2012.

[14] Amitava Nag, Saswati Ghosh, Sushanta Biswas, Debasree Sarkar, Parta Pratim Sarkar " An Image Steganography Technique using X-Box Mapping", International Conference on Advances in Engineering, Science and Management (ICAESM -2012) on pp. 709-713, IEEE, March 2012.

[15] Chen, Rong-Jian, Jui-Lin Lai, and Shi-Jinn Horng. "Anti-forensic Steganography Using Multi-bit Minimum Error Replacement with Flexible Bit Location."Computer, Consumer and Control (IS3C), 2012 International Symposium on pp. 175-178, IEEE, June 2012.

[16] Nguyen, Luong Viet, Trinh Nhat Tien, and Ho VanCanh. "The method of hiding steganography without key exchanging and original image." Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on Vol. 2, pp.408-412, IEEE, May 2012.

[17] Karaman, H. B., and S. Sagiroglu. "An Application Based on Steganography."Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), pp. 839-843, IEEE Computer Society, August 2012.

[18] Avinash, K. Gulve, and M. S. Joshi. "A Secured Five Pixel Pair Differencing Algorithm for Compressed Image Steganography." Computer and Communication Technology (ICCCT), 2012 Third International Conference on pp. 278-282, IEEE, November 2012.

[19] Bansod, Smita P., Vanita M. Mane, and Leena R. Ragha. "Modified BPCS steganography using hybrid cryptography for improving data embedding capacity." Communication, information & computing technology (ICCICT), Chicago, 2012 international conference on pp. 1-6 IEEE, October 2012.

[20] Khosravi, Mahdi, Simin Soleymanpour-moghaddam, and Maryam Mahyabadi. "Improved pair-wise LSB matching steganography with a new evaluating system." Telecommunications (IST), 2012 Sixth International Symposium on. IEEE, 2012.

[21] Samidha, Diwedi, and Dipesh Agrawal, "Random image steganography in spatial domain" Emerging trends in VLSI, embedded system, nano electronics and telecommunication system (ICEVENT), 2013 international conference on. IEEE, pp. 1-3, January 2013.