A New Method for Text Hiding in the Image by Using LSB

Reza tavoli
Department of computer, Islamic Azad University, chalus Branch, chalus, Iran

Maryam bakhshi
Department of computer, poyandegan danesh, chalus, Iran

Fatemeh salehian
Department of electronic, hadaf university,sari, Iran

***Abstract*—an important topic in the exchange of confidential messages over the internet is the security of information conveyance. For instance, the producers and consumers of digital products are keen to know that their products are authentic and can be differentiated from those that are invalid .The science of encryption is the art of embedding data in audio files, images, videos or content in a way that would meet the above security needs. Steganography is a branch of data-hiding science which aims to reach a desirable level of security in the exchange of private military and commercial data which is not clear. These approaches can be used as complementary methods of encryption in the exchange of private data.**

Keywords—Hiding text inside an image; image processing; Steganography; image compression; LSB

I. INTRODUCTION

From the time that humans became able to communicate, developing a secret connection was one of the main demands. In past, despite having minute means, people had tried to hide data to not be discovered easily. These information that often their security mattered, usually were associated with war or military information and details of the countries' borders, which were hided in various frames according to the level of their importance.

In ancient Rome for instance, they used to shave the herald's head and tattoo the desired text on his skull. The herald was in quarantine until his hair grows, then he moved to the destination, that again by shaving his head they would read the hidden content.In addition, Italians in the medieval era used a sort of ink that could penetrate the egg shell and give color to the egg white. Thus by peeling the egg, they could easily read the data. In ancient Persia also, they wrote crucial data with the use of onion juice on the paper, hence when it dried, there was no sign of the content. Then with slightly heating the paper, the letters would become clear and the information could be readable [1].

On the other hand, these days the considerable progress of internet and the rapid growth of its use have propelled human to the digital world and communicating by the use of digital data. Meanwhile, the communication security is a critical need and is felt more and more every day. Today the modern techniques of steganography have found many users. In the terrorist operation of 11 of September also steganography was used for information conveyance of this operation. Furthermore, other quite useful applications for steganography that are in this area are in public TV posts, network, controlling the products copyright, search engines, image, and bank cards. Even nowadays medical science and DNA use steganography [2].

Generally there are three approaches for hiding text in an image. First method is encryption, in which information is encrypted in a way that is not understandable for the third party; however the receiver and transmitter can decrypt the data with a common key [3],[4]. The encryption or decryption operations are performed by programming algorithms in the digital domain and occasionally one can realize confidential data, depending on the level of the algorithm's security. The second method is steganography in which not only the information remains secret but also the existence of confidential connection is hidden. In fact steganography is the art and science of hiding communication and its purpose is to hide the existence of any connections between the receiver and transmitter. Often it is thought that the connection is secured by coding the exchanged message; however sometimes practically coding is not enough. Accordingly several methods were proposed for hiding data instead of coding. The third method is watermarking which will be further explained bellow. Assume that a legal owner of a photo embeds a series of messages in an image. Whenever such an image is stolen and put in a website, its legal owner can provide this confidential message to the court as a proof of his ownership. This type of hiding is called watermarking [5].

Prior to the explanation of the LSB method, we should clarify the main difference between encryption and steganography. In fact, the difference between these two terms is the purpose of encryption which is the concealment of the message content and generally not the existence of the message. Whereas steganography aims to hide any sign of the existence of the message [5]. In cases that the exchange of encrypted information is problematic, the existence of the communication should be hidden. For instance if one accesses encrypted content in any way, he will know that this content contains encrypted messages. However in steganography the third person does not obtain any information about the existence of the hidden message at all. The steganography methods were developed for protecting the property rights of multimedia products. In other words this technique was designed to protect the media itself [6].

Prior to further explanation, a brief discussion concerning the LSB method is necessary to simplify next topics. Most steganography approaches that embed the data within the pixel space take advantage of the LSB method. When a file is made, usually some of its bytes are not usable or are worthless [7]. These bytes can be changed without harming the file

considerably. This allows us to write some information in these bytes without anybody being aware that the process has taken place. As it was mentioned before, each video file is merely a binary file that contains colors and light intensity of each pixel according to the binary number [8].

Images normally use an 8-bit or 24-bit format. In the 8-bit format we solely can use 256 color for each pixel (In these 8 bits, each bit is one of the values of 0 or 1 which totally provides 2 ^ 8 or 256 different colors). In 24-bit format also every pixel have the capacity of 2 raised to the 24 power. In this format each pixel uses of 3 bytes of 8 bits. Each byte shows the light intensity of three main colors of red, blue and green. For instance, colors in format html3.0 are according to the 24-bit. Each color in this format has a code based on 16 which comprises of 6 characters. The first two characters are associated with the color red; also the second and third characters are respectfully associated with the colors blue and green. For example for creating the color orange, the intensity values of the colors red, green and blue are respectively 100%, 50% and 0 which is definable with #FF7FOO in html. Furthermore, the size of an image depends on the number of pixels of that image. For instance for an image with the resolution of 640*480 that uses the dynamic range of 8-bit, the image size should be 640*480**3 = 900 KB. Suppose that three neighbor pixels are coded as below [9]:

	Red, Blue, Green
pixel1	10010101/00001101/11001001
Pixel2	10010110/00001111/11001010
Pixel3	10011111/00010000/11001011

Now assume that we want to embed the 9 bit of these data 101101101into these pixels (these 9 bits encrypted data are supposed to be a message)

Now if we use the LSB method, these 9 bits are put into the least significant bits of these three pixel's bytes, then we have the below chart:

	Red, Blue, Green
pixel1	10010101/00001100/11001001
Pixel2	10010111/00001110/11001011
Pixel3	10011111/00010000/11001011

It is seen that only four bits have been changed and this would not harm the image greatly. For instance, a change in the blue color bit from 11111111 to 11111110 is never detectable for the eyes.

Now we may want to hide a text in an image. In this case every character, takes up one byte (8 bites). Since we should put these bits into these image pixels, thus we need to divide these eight bits to a 1-bit packages (or larger packages), and each bit are placed in the least significant bits of one of the main three colors of pixels. This way, words of all languages that are compatible with ASCII or UTF-8 (or any other coding), can be embedded within an image [10].

The LSB method with taking advantage of the random factors and secret key enhances the necessary security for hiding the data. However, by investigating the researchers' studies, one can simply show that this method can be broken (decoded). Although the least significant bits of pixels are seemed random, practically they do not have the real random. In general, the type of these bits arrangement in an image represents some features of that image [10], [11].

II. A Review of Previous Studies

Studies on image compression and steganography have been an active area of research from the beginning of the digital image processing. The use of preprocessing methods for improving compression rate and elevating the level of encryption has interested many researchers. Here we briefly explain some articles.

In a research done by $Shatnavy$in 2012, he used a method of embedding in consecutive pixels. According to his technique, the message with the hidden data is saved in the difference between the values of the consecutive pixels' gray levels. Here the gray level range is within 0 to 255. Selecting this range according to the sensitivity of the human visual system leads to the color change. After that the image is divided to anon-overlapping two pixels blocks. Then the difference between the gray levels of the consecutive pair of pixels d is calculated. If d is in r_k range, then r_k indicates the number of hidden bits in these pair pixels (In fact the difference in these pixels).Thus in parts of the image that the difference between the consecutive pixels is high, the sensitivity of the human visual system is low and therefore more information are saved. Then this number of bit is chosen from the bit stream of the hidden message and is summed with the lowest value of the r_k in decimal format. So a new value such as d is eventuated for the difference of the gray levels of the pixels [12].

In an article by Reddy and others in 2004, he offered a steganography method according to singular value decomposition and discrete wavelet transform. In this type of steganography which is driven from the composition and decomposition of the singular value and discrete wavelet transform, two domains of spatial and frequency steganography were compounded. In this method, discrete wavelet transform is applied on both image and $stegano$ image (the image that we want to hide). As we know in discrete wavelet, the image is divided to four frequency areas which are $cA, cH, CV, cD. cA$ isthe approximation signal (the above left side picture), $cD$is the detail signal (the bottom right side picture), cH the horizontal detail signal (above right) and cV the vertical detail signal (bottom left). In this approach, it applies discrete wavelet transform to both images. Then again it applies discrete wavelet transform to cA region. The obtainedcA region is used for continuing the steganography procedure. In continue, the obtained CA's are converted to three matrixes by singular value decomposition. Then the yield singular value is multiplied in a number less than one hundredth and summed with the singular value of the coverage image. After that by multiplying the singular value of these three matrixes and applying the inverse discrete wavelet transform, it converts to an image in which the $stegano$image is hidden in it [13].

In a paper submitted in 2012 by d.rajedi and others, they proposed a simple method of hiding information. This method

includes the involvement of different secret keys in various stages with the implementation of various matrixes and summing a series of handwritten codes. The proposed method in this paper can be thought as a ladder, in which the normal and encrypted texts are embedded upon the first and final steps. Furthermore in this paper, d.rajedi applied his method on the three-dimensional image. First the typical text simply and without any changes enters to this model and then is followed by a series of transformations, operations of changing information and handwritten codes. Finally it converts to an object with the name of RNS coded object. We can use the produced RNS object as the background of images. This approach is implemented on the images with the use of the alpha factor (the alpha character is connected to the clarity character of the images). Ultimately we have a clear image in the background of the main image. This scheme consists of three main parts which are the simple text encryption, the method of decryption of the encrypted text and the RNS model [14],[15].

III. The Introduction of the Proposed Approach in Hiding Data in an Image

The method that is suggested in this paper uses a stage of the textual data compression and then coding it prior to steganography. In other words, first it applies a preprocessing technique on the desired text, and then puts the text into the image. The proposed method encodes the compressed text and then with the use of a 4*4 mask performs snake scan ordering. After that it loads the eventuated compressed and coded text on image pixels. The below block diagram depicts the stages and procedures. The following parts of this section explain the encryption and decryption stages of the image.

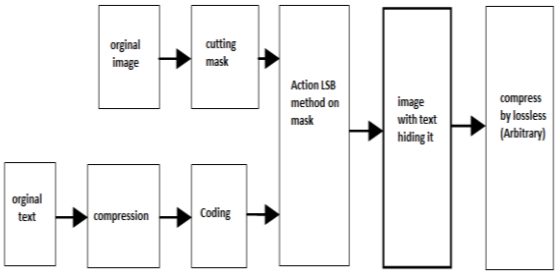

Fig. 1. The final block diagram of the proposed technique for steganography

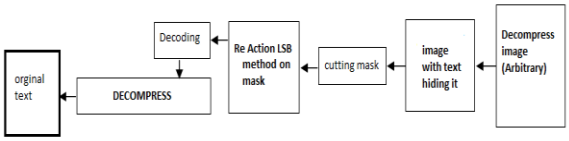

Fig. 2. The final block diagram of the proposed technique for steganography (Text mining)

A. Algorithms (the encryption steps for an image)

- Initially it decreases the volume of the raw image with the use of one the compression methods (Differential method for instance). In fact, this compression not only increases the capacity of number of saved characters in the image, but also it is thought as a type of coding.
- Then it divides text to an arbitrary length of segments. For example each 5 characters of the below array consists of the raw image and the presentation of the ASCCI cods of the desired image.

  Text = $gdcba$
  Text $_{(ASCII)}$ = 103,100,99,98,97
- The next step is the formation of the differential array from the previous scans in which it writes the first array element, and for remained numbers, it subtracts each array from its previous one.

  Array $_{(sub)}$ = 103, 3,1,1,1
- The maximum number from index one to the next ones (not the index zero) are specified for determining the required bit space for their storage. In previous step the maximum array number is three. The obtained maximum number requires two bits for storage. Now in decoding it should be specified that how many bits should be allotted to the data byte (which is two bits here). The storing format of data is as below:

Byte Sequence = (the first byte)(the length of data bytes)(the data bytes sequence)

- Data bytes are accompanied by a sign bit, which represents the sign of the difference from subtraction along an array, for which the succeeding number in the array is larger than the previous; then the bit sign is one, otherwise it is zero.
- For the further reduction to the number of bytes in a sequence, it employs the following code associated with sequence length:

$$if\ (MaxByte <= 1)\ SizeBye = 1, Else\ if\ (MaxByte < = 3)\ SizeBye = 2, Else\ if\ (MaxByte < = 7)\ SizeBye = 3\ \ldots$$
$$if\ (SizeBye = 1)\ insert\ 000\ Else\ if\ (SizeBye = 2)\ insert\ 001\ Else\ if\ (SizeBye = 3)\ insert\ 010\ \ldots$$

The below table values depicts the “data bytes length”

TABLE I. Binary Codes of Byte Size

Size Byte	code
1	**000**
2	**001**
3	**010**
4	**011**
5	**100**
6	**101**
7	**110**
8	**111**

- The output text of the first step is coded by an arbitrary algorithm encryption. An arbitrary key is used to XOR the encoded algorithm with the output text of the first stage and delivered to the next part.
- The XOR operation operates in a way that shows the difference between bits. In other words, if both bits are zero or one then the output is one, otherwise it is zero. For instance the two below characters is supposed:

Clear Text =ab

Text $_{(ASCII)}$ =97, 98 = (01100001, 01100010) $_{Base\ 2}$

- We suppose x as a key: Key $_{(ASCII)}$ = 120 = (01111000) $_{Base\ 2}$
- After applying XOR operation, the text is coded as below:

TABLE II. XOR TABLE FOR HIDING TEXT IN THE IMAGE

	a								b							
XOR	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0
	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
Coded	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	0

- For converting the coded text to the normal mode (decoding), the key is XOR-ed with the data codes.

TABLE III. XOR TABLE FOR HIDING DECODES

XOR	0	0	0	1	1	0	0	1	0	0	0	1	1	0	1	0
	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
Decoded	0	1	1	0	0	0	0	1	0	1	1	0	0	0	1	0
	a								b							

- Obviously it is necessary for the receiver to have the key which the transmitter used for data coding.
- The output text from the second step is scanned with the LSB method and 4*4 masks (the snake scan is applied for more security).

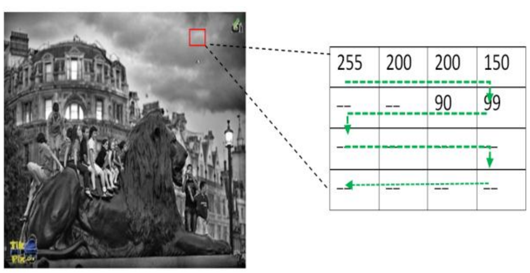

Fig. 3. snake ordering in mask

- If required, this step can compress the output image with lossless methods (such as LZW or differential).

B. Algorithm (the decoding stages of an image)

The decoding is the exact opposite of the above steps, thus preventing obvious explanations.

C. Evaluation

In order to evaluate our algorithm, we used the following arbitrary text and images:

An embedded text which includes 3681 characters and 4 images size 300*300 pixels.

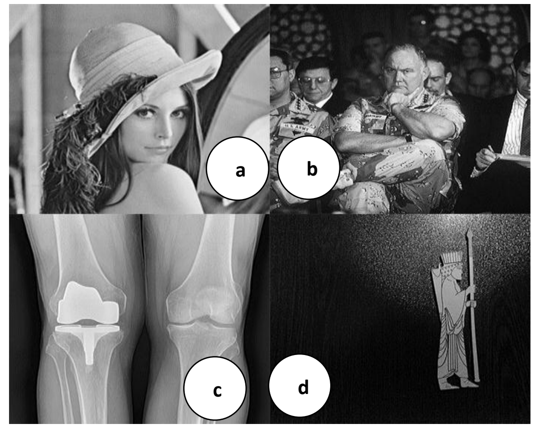

Fig. 4. Four used pictures in *steganography* (a) Lena's picture (b) USA's war minister picture (c) leg's scan picture (d) *Achaemenian* man picture

- Test results validation:

In this section, we validate our proposed method. Before encrypting a message, it has to transform into the binary form. This transformation is done based on 16 bites codes of UNICODE. For instance, each Farsi word in this transformation changes into the binary code of 16 bites. Thus, for obtaining the binary equivalent of a message, we have to put all 16 bites codes of characters in together. According to the proposed method, the equivalent binary message is encrypted by permutation technique, and then is placed in the image. The advantage of bit permutation over character permutation is that while changing encrypted bits into characters, difference characters are displayed rather than the main message. In this method the length of key is dynamic. In other word, it is arbitrary. As mentioned, each UNICODE is 16 bites. Therefore, if the length of a key is not a factor of 16, bites of difference characters are displaced while encrypting, and this considerably increases encryption power and causes that decryption becomes difficult. When encrypted bits are displaced in the image, recognizing whether any information is embedded into the image would become difficult for one who controls the connection in an unauthorized way.

As an instance, for validating, we encrypt the phrase of "this is a secret massage" in above image with the following key:

Key={17,5,14,20,1,18,3,10,4,16,0,2,6,11,13,7,12,8,9,15,19}

After that, it places it in the image and then extracts it by key and without key. Table below presents extracted values for both forms.

TABLE IV. COMPARE EXTRACTED TEXT WITH KEY AND WITHOUT KEY

	Information extracted
With key	This is a private message
Without key	炦 ⌋ **G** 焐 䛳'裋ᄆघ≬ 堤□〈鄏⒤讋•鄏≥絨**â**□□

As observed from the table, if an unauthorized person who controls the connection, suspected to the sending image, cannot recognize any information from it. Because while extracting LSB of an image without any information confronts with ambiguous data like those data in third row of table.

- Test results of peak signal to noise ratio:

Peak signal to noise ratio or PSNR is an engineering term for the ratio between maximum power of a signal and the maximum power of noise that affects the correctness of displaying an image. Or more simply, the less PSNR, the more noise which is due to *stagnography* in the image. For calculating PSNR, first we should obtain medium square error or MSE between main image and *stagnography* image. We use following expression for calculating MSE[16]:

$$\text{MSE} = \frac{1}{\text{n}}\sum_{\text{i=0}}^{\text{i=n}}(\widehat{Y}\text{i} - Yi)^2 \quad (1)$$

In which $\hat{Y}i$ is the main image and Yi is the *stagnography* image, respectively. Moreover, I is the length and width of both images. N is the number of image pixels. After MSE calculation, now we can calculate PSNR. The formula is as follow [16]:

$$\text{PSNR} = 10 * \log10\left(\frac{\text{M0}}{\text{M}}\right) \quad (2)$$

Where M_0 is thage has 32 bits, maximum value of a pixel is 2^{32}. Above expression can simply be presented as [17], [18]:

$$\text{PSNR}=20*\text{Log10}\left(\frac{\max \text{c}}{\sqrt{\text{MSE}}}\right)=20*\text{Log10}\left(\frac{2^{32}}{\sqrt{\text{MSE}}}\right) \quad (3)$$

Test results of selected images are given in below table:

TABLE V. MSE RESULT OF IMAGES WITH THE USE OF TRADITIONAL LSB METHOD IN EACH R,G,B CHANNEL

images	MSE B	MSE G	MSE R
a	37.49	37.77	37.64
b	112.91	113.17	113.02
c	52.47	52.72	52.57
d	50.42	50.65	50.56

TABLE VI. MSE RESULT OF IMAGES WITH THE USE OF PROPOSED LSB METHOD IN EACH R,G,B CHANNEL

images	MSE B	MSE G	MSE R
a	0.1075	0.4314	0.2803
b	0.1082	0.4304	0.2764
c	0.1049	0.4063	0.2592
d	0.1070	0.4326	0.2829

As the MSE criteria approaches zero, the less frequently the output image changes from the primary image, which is good. Accordingly, above tables show the fidelity of both methods.

TABLE VII. PSNR CALCULATION RESULTS OF IMAGES WITH THE USE OF SIMPLE LSB BY SEPARATING EACH R,G,B CHANNEL

images	PSNR B	PSNR G	PSNR R
a	32.39	32.35	32.37
b	27.60	27.59	27.59
c	30.93	30.91	30.92
d	31.10	31.08	31.09

TABLE VIII. PSNR CALCULATION RESULTS OF IMAGES WITH THE USE OF THE PROPOSED METHOD BY SEPARATING EACH R,G,B CHANNEL

images	PSNR B	PSNR G	PSNR R
a	57.81	51.78	53.65
b	57.78	51.79	53.71
c	57.92	52.04	53.99
d	57.83	51.76	53.61

For PSNR criteria, it is better that it approaches 100. The acceptable range in stagnography are within 50 to 100. In a simple LSB method, the criteria range is within 0 to 30; however table 4, simply presents that how these criteria were improved.

- Image Histogram:

Another way of recognizing a message in an image is comparing histogram of the main image with the *stagnography* image. In traditional LSB method, *stagnography* is done on sequential pixels, thus abnormality is created in the image histogram.

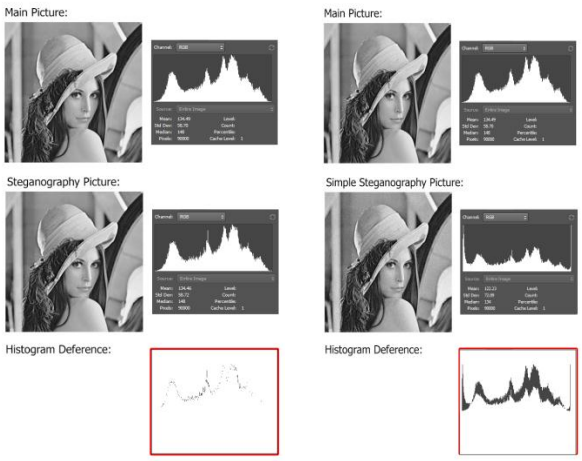

Fig. 5. Comparison between images and their histograms along with their difference histograms (right image is by simple traditional LSB and left image is by proposed method)

As it can be seen from Lena's picture, the histogram of a simple LSB image has a considerable difference with the main image, but in proposed method this difference is trivial due to recompressing words and using fewer image bits.

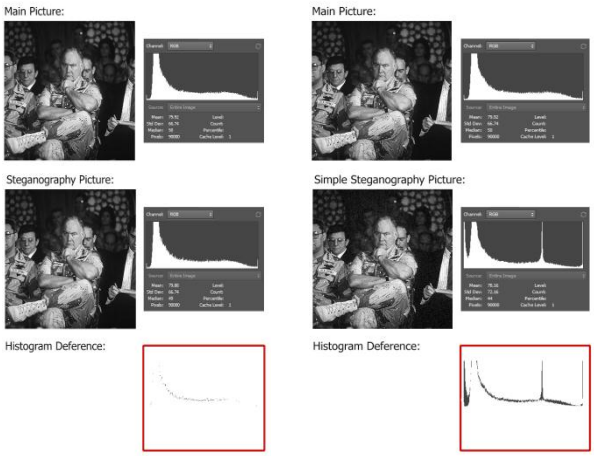

Fig. 6. Comparison between images and their histograms along with their difference histograms (right image is by simple traditional LSB and left image is by proposed method)

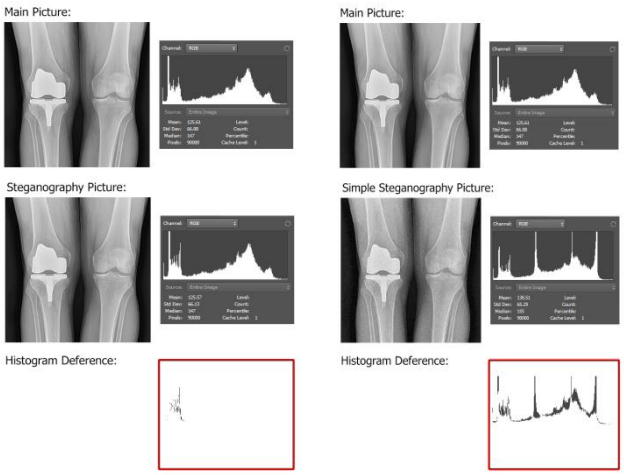

Fig. 7. Comparison between images and their histograms along with their difference histograms (right image is by simple traditional LSB and left image is by proposed method)

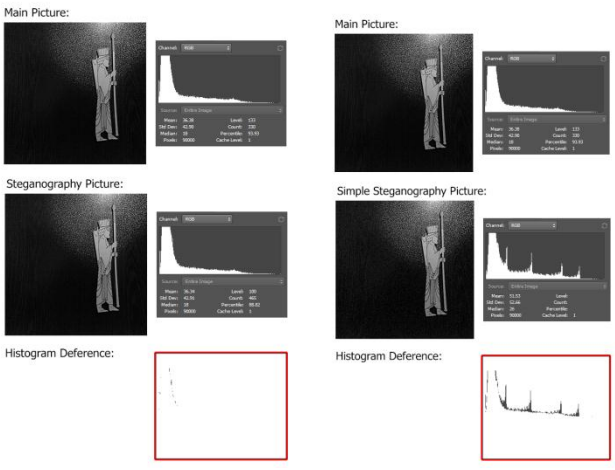

Fig. 8. Comparison between images and their histograms along with their difference histograms (right image is by simple traditional LSB and left image is by proposed method)

IV. Implementation

Regarding the primary desired aims of this paper that the most important one is the capability of commercialization, hence the implementation of the above steps was carried out in c# and under .NET FRAME WORK 4. Therefore it can be utilized in operating systems in a widespread range.

V. Conclusion

This paper initially investigates the multiple approaches of steganography in an image . It has shown that the space pixel provides more capacity than the frequency domain. In addition, the type of an image is effective in achieving the desired result in steganography.In comparison to today's proposed methods in frequency domain; our algorithm has the ability of storing a larger amount of information.

Compression before hiding step is more appropriate in invisibility of the steganography. Furthermore it increases the image capacity for data inclusion. Mixing the use of an appropriate mask with the application of a particular scan of an image, and moreover adding a step of encryption with each, allows for multiple separate phases of information security. By combining the mentioned stages with LSB approach, a desirable percentage of steganography was yielded. Therefore these steps decrease the odds of discovering the hidden data. In fact, by putting together a number of methods and designing an efficient algorithm, we have achieved an innovation and a relative improvement in LSB method. although performing all of the steps above successfully obtains a higher level of security using the LSB method, it also contributes to the problem of increased load to the processing system. For this shortcoming a solution should be devised which does not fit within the scope of this article. The second proposal is to have a random place for masks in the image. In detail, it chooses a fixed position for the first mask, but for the position of the next mask, it selects randomly. Like the structure of a linked list, the address of the next mask is saved in its previous mask. Hence we will reach to a higher level of security.

Reference

[1] Abas Chedad,Joan Condell,Kevin Curran,Paul McKevitt (2010), Digital image steganography: Survey and analysis of current methods, Signal Processing, 727–752

[2] Udda Lavanya,YangalaSmruthi,Srinivasa Rao Elisala, Data hiding in audio by using image steganography technique, Volume 2, Issue 6, November – December 2013

[3] Weiqi Luo,Jiwu Huang,Fangjun Huang (2010), "Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Transactions on Information Forensics and Security, vol. 5, pp. 201-214

[4] HadVitthal S.,BhosaleRajkumar S.,PanhalkarArchana R)2012(, A Novel Security for Secret Data using Cryptography and Steganography" International Journal Computer Network and Information Security, vol. 2, pp. 36-42

[5] Vanya,2 YangalaSmruthi,3 Srinivasa Rao Elisala(2013), "Data hiding in audio by using image steganography technique", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), VOL 2.

[6] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison Publishing Co., 1993.

[7] P.Wayner,Disaapearing Cryptography, 2nd Ed., Elsevier Science: US, 2002.

[8] R.J.Anderson and F.A.P.Petitcolas, "On the limits of steganography," IEEE J. of Selected Areas in Communication, vol. 16, no. 4, pp. 474-481, May 1998.

[9] Al-Shatnawi A.M., "A New Method in Image Steganography with Improved sequential bits", Applied Mathematical Sciences, vol. 6, no. 79, pp. 3907 – 3915, 2012.

[10] Reddy A.Adhipathi,and B.N.Chaterji 2004, "A new wavelet based logo-watermarking scheme", Pattern Recognition Letters, September 2004.

[11] Rajdeep Chowdhury,Nilanjan Dey,Saikat Ghosh" Design and Implementation of RNS Model Based Steganographic Technique for Secured Transmission", International Journal of Advanced Research in Computer Science and Software Engineering, Volume2, Issue 3, March 2012

[12] Phad Vitthal S.,Bhosale Rajkumar S.,Panhalkar Archana R) 2012(A Novel Security for Secret Data using Cryptography and Steganography" International Journal Computer Network and Information Security, vol. 2, pp. 36-42

[13] Nagham Hamid,Abid Yahya,R.Badlishah Ahmad & Osamah M.Al-Qershi,"Image Steganography Techniques: An Overview", International Journal of Computer Science and Security (IJCSS), Volume (6): Issue (3):2012

[14] Weiqi Luo, Jiwu Huang, Fangjun Huang,(2010) "Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Transactions on Information Forensics and Security, vol. 5, pp 201-214

[15] Zhijie Shi and Ruby B. Lee, "Bit Permutation Instructions for Accelerating Software Cryptography", Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp- 138-148, July 2000.

[16] Rohankar, Jayant (Nov 2013)- "SURVEY ON VARIOUS NOISES AND TECHNIQUES FOR DENOISING THE COLOR IMAGE" (PDF)- International Journal of Application or Innovation in Engineering & Management 2 (11).Retrieved 15 May 2015.

[17] Shamim Ahmed Laskar ,Kattamanchi Hemachandran(2012), High Capacity data hiding using LSB Steganography and Encryption ,(IJDMS) International Journal of Database Management Systems, Vol.4, No.6

[18] Strang, Gilbert (July 19, 2005), Linear Algebra and Its Applications (4th ed.), Brooks Cole, ISBN 978-0-03-010567-8