ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

A Novel Image Steganography based on Secured Inversion Technique

Suma. S[1], Dharmambal[2]

M.Tech,4th sem, Dept of ECE(Communication System), NHCE, Bengaluru, Visvesvaraya Technological University, India.

Sr. Asst. Professor, Dept. of ECE, NHCE, Bengaluru , Visvesvaraya Technological University, India

ABSTRACT: In this paper more secured version of inverted LSB (Least Significant Bit) steganography is proposed and implemented. Bit inversion technique is used to get the better quality in stego image. In this technique, Steganalysis is performed on the plain LSB stego-image to analyze the bit pattern of second and third LSBs that co-occurs with LSB. Based on this analysis, LSB of those pixels may be inverted which co-occurs with a specific bit pattern hence number of pixels modified is less compared to plain LSB method therefore we can get the enrichment in PSNR of stego image. Arnold Scrambling Algorithm is applied for secret image before hiding in a cover image therefore third party cannot easily recover the secret image. Secret text, encrypted using RSA algorithm is also embedded in cover image. The advanced method shows good refinement to plain LSB in terms of protection and quality of the image.

KEYWORDS: Steganography, LSB, Arnold Scrambling Algorithm, RSA algorithm, bit inversion, PSNR.

I. INTRODUCTION

Steganography word is originated from Greek words Steganós (Covered), and Graphy (Writing) which literally means "cover writing". In general we can say steganography is "**nonvisual**" communication. Steganography means to hide messages in another medium (audio, video, image, communication). Nowadays steganography systems use multimedia objects like image, audio, video etc as cover media because people often transmit digital images through internet like email or share them with other internet communication application. It is different from protecting the actual content of a message.

Steganography hides the secret message inside a cover-object (carrier object). Cover object and stego-object (carrying hidden information object) are similar after hiding process. Therefore steganography (hiding information) and cryptography (protecting information) both are different from one another. It is difficult to recover information due to invisibility or hidden factor without known procedure in steganography. Detecting procedure of steganography is known as Steganalysis.

Image steganography is method of hiding information into cover-image and produce a stego-image. This stego-image then sent to the receiver by known medium, where the attackers do not know that this stego-image has hidden information. After receiving stego-image, receiver can simply extract the hidden message with or without stego-key (depending on embedding algorithm). Basic diagram of image steganography is shown in below Figure without stego-key. For embedding procedure, embedding algorithm needs a cover image with message image. Stego-image is the output of embedding algorithm which is simply sent to extracting algorithm, where extracted algorithms give the original message that is hidden in the stego-image.

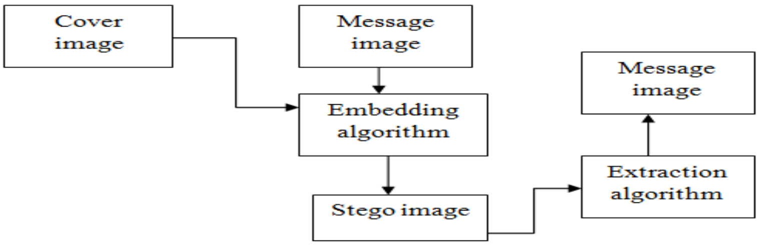


Fig. 1 Image Steganography

Main aim of image steganography techniques includes high payload capacity, high imperceptibility and more robustness. To hide the secret information number of bits used per pixel is called payload capacity. Imperceptibility means intruder is unable to detect the presence of message in the image. This is measured using peak signal to noise ratio (PSNR). If PSNR value is high then higher will the imperceptibility. Robustness means the ability to resist attacks from intruders.

Among the data hiding methods least-significant-bit (LSB) substitution is the simplest one. It embeds the secret data into some LSBs of the pixel value in cover image. In LSB based technique hidden data capacity is low but it is good at imperceptibility because for data hiding only one bit per pixel is used. Secret message can be retrieved very easily by collecting the LSBs once attacker come to know that the image has some hidden secret data hence LSB technique is not robust. Authors have proposed RS Steganalysis technique; this can estimate message size efficiently when the message is embedded randomly. In SPA (sample pair analysis) is a powerful Steganalysis method is proposed, this method uses sample pair analysis to detect the message length[2].

In this paper, author present LSB based steganography scheme which is more secure and robust than plain LSB method. Rather than storing the message image bits sequentially, they are stored in the random order generated by Arnold scrambling algorithm. If the message is text then it is encrypted using RSA algorithm. After that Steganalysis is performed on the plain LSB stego-image to analyze the bit pattern of second and third LSBs that co-occurs with LSB. Based on this analysis, LSB of those pixels may be inverted which co-occurs with a specific bit pattern, this results increase in PSNR of stego-image and this makes the task of Steganalysis difficult[1]. Author also presents secret text hiding in an image and it is encrypted using RSA algorithm before hiding.

II. RELATED WORK

In [3] author uses RGB color image as carrier message. The RGB color image has 24 bits values per pixel and each white pixel is represented as 11111111, 1111111 and 11111111 and black pixel is 00000000, 00000000 and 00000000. It hides the secret message in the two least significant bits directly based on binary coding in RGB color image. This leads to the change in the image resolution & it is easy to attack. In [4] author proposes an optimal LSB substitution method that uses the dynamic programming strategy in order to find out an optimal solution of a bijective mapping function. The bijective mapping function will transform the secret data into another set of values. Then in the cover image the rightmost LSBs of the pixel value are replaced with the transformed values to form the stego-image .Here stego image has minimal distortion. In [5] author uses JPEG technique that divides the input image into non-overlapping blocks of 8x8 pixels and uses the DCT (Discrete cosine transform). The method discussed divides the cover image into non-overlapping blocks of 16x16 pixels. To embed two secret bits, for each quantized DCT block, the least two significant bits of each middle frequency coefficients are modified. The proposed hiding technique achieves better hiding capacity than Jpeg-Jsteg methods which are based on the conventional blocks of 8x8 pixels. In [6] author proposes more secured version and robust than plain LSB method. Here author is not storing the message bits sequentially instead they are stored in a random order generated by RC4 algorithm which uses a stego key shared by both sender and receiver. After that steganalysis is performed on the stego-image to analyze the bit patterns of second and third LSBs that co-occur with LSB. Based on this analysis, LSB of those bytes may be inverted that co-occurs with a specific bit pattern and this improves the PSNR of stego image.

III. IMPLEMENTATION

A. Plain LSB Algorithm:

A raw digital image is a collection of pixels representing the intensity of light at that pixel position .Digital images are typically stored in either 24-bit or 8-bit per pixel. An 8-bit image can represent 256 different levels of light intensities that is called gray image. 24-bit images are called true color images because they can represent a large number of color intensities.24-bit image (color image) will provide more space for hiding information but 24-bit images are generally large. A 24-bit image 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes. As such, large files would attract attention when they were transmitted across a network or the Internet.

Generally gray images are used to hide information such as GIF files and each pixel is represented as single byte and it can correspond to 256 colors. So we can say that pixel value ranges from 0 to 255 and the selected pixels indicates certain colors on the screen. The plain least significant bit technique implies manipulation of LSB plane of cover image by replacing LSBs of cover image with message bits. In this method only one level of intensity differs between original and modified pixel because only LSB is changed, which cannot be detected visually. Hence the eavesdropper will not get the idea that some message is hidden in the image.

Example shows the simple LSB method to hide the character ‘A’ into an 8-bit cover image

Binary equivalence of ‘A’-01100101

Bit pattern of the pixels in cover image:

0010011**1** 1110100**1** 1100100**0** 0010011**1** 1100100**0** 11101001 11001000 00100111

After copying the bit pattern is **-**

0010011**0** 1110100**1** 1100100**1** 0010011**0** 1100100**0** 1110100**1** 1100100**0** 0010011**1**

Bits that are bold represent the changed bits. The probability of modification of a pixel of cover image is 0.5. So, approximately half of the pixels in cover image get changed. This method is vulnerable to attacks.

In simple LSB approach there are several variations. Several approaches modifies two or more bits of cover image so that more amount of data could be hidden in a cover image instead of replacing one bit. Using up to four LSBs for hiding message gives acceptable results but it will affect the quality of cover image as more higher order LSBs are replaced. The LSB replacement allows simply replacing the information behind cover image directly and changing a single bit of a pixel does not cause noticeable, difference in image quality. There is high perceptual transparency of LSB because change in amplitude is very small.

The benefits of Least-Significant-Bit(LSB) steganographic data embedding are that it is straightforward and understandable, easy to implement, and it produces stego- image that is almost same as the cover image and its visual infidelity cannot be judged by naked eyes. The disadvantages of LSB approach is the size of cover image required for a particular message image that is for certain capacity of message, cover image required is 8 times this leads to the increase in bandwidth to send the image. Another disadvantage is that if an attacker finds that some information is hidden behind the cover image, he can easily get information by just collecting LSBs of stego image. Hence this method is not successful.

B. Scrambling using Arnold algorithm:

Plain LSB implementation is easy and it is direct method to hide information in a cover image. In the pixels of cover image the message is embedded with sequence mapping technique. Even though LSB hides the message in such a way that humans do not perceive it, still it is possible for the opponents to retrieve the message because of the simplicity of the technique. Therefore, if malicious people suspects that some secret information is embedded in the image they can easily try to extract the message from the beginning of the image. Therefore to improve the security of LSB scheme this method is proposed. This method overcomes sequence mapping problem by embedding the message that is encrypted using Arnold algorithm. Pixels in cover image are replaced by scrambled bits of the secret image. This method makes very difficult for unauthorized people to get secret message because even though if they collect LSBs of cover image they are not actual secret message but it is scrambled.

Arnold scrambling was proposed by the Mathematician Arnold in the research of ergodic theory. Arnold scrambling algorithm is widely used in digital watermarking technology because it has the feature of simplicity and periodicity [11]. The original image can be restored after several cycles according to the periodicity of Arnold scrambling because the periodicity of Arnold scrambling depends on the image size, it has to wait for a long time to restore an image. Arnold scrambling algorithm is based on square digital image and these images are mostly *N*×*N* pixels of the digital image. Assuming the original image size is N×N, (x, y) is the pixel coordinate, the pixel is moved to (x ', y ') after the geometric transformation .This geometric transformation can be described as follows
[x'; y’]= ([1, 1; 1, 2] * [x; y]) mod (order of image).

Here x, y∈ {0, 1, 2, 3, .N-1}, *N* is the order of digital image matrix. (*x, y*) in the right is the input and (*x ', y'*) in the left is the output.
Arnold scrambling recovery is the pursuit of its inverse matrix to the inverse transformation.
[x';y']= ([2,-1;-1, 1] * [x; y]) mod (order of image)
Advantages of Arnold Scrambling Algorithm are very simple and easy to understand. It is applied for all square digital images.

C. RSA Algorithm

RSA algorithm is developed by Ron Rivest, Shamir & Adleman. RSA algorithm is a message encryption cryptosystem here initially two prime numbers are taken and then the product of these two prime numbers are used to create a public and private key which is further used during encryption and decryption[12].

- Select two large prime numbers p and q
- Compute the system modules n=p*q
- Calculate Euler’s totient value for n Φ(n)=(p-1)(q-1)
- Select integer e (encryption key) at random , Gcd(Φ(n),e)=1
- calculate decryption key such that d=e¯1mod Φ(n)
- Publish the Public Encryption key (KU)=(e, n)
- Keep secret Private Decryption key (KR)=(d, n)
- To encrypt plain text M satisfying M < n, sender obtains public key of recipient KU={e, n}
  Calculate Cipher text C = M ^ e (mod n)

D. Bit Inversion Technique

Here we applied a novel bit inversion technique to improve the stego image quality. Consider the following example to understand this technique. Four message bits 1 01 1 to be hidden into four cover image pixels 1 010 1 **1 0**0, 1 1 1 0 1 **1** **0** 1, 1 0 1 1 1 0 1 1 and 1 1 1 0 1 **1 0**1. Stego-image pixels after plain LSB steganography are 1 010 1 1 0 **1**, 1 1 1 0 1 1 0 **0**, 1 0 1 1 1 0 1 1 and 1 1 1 0 1 1 0 **1**. Two pixels (first and second) of cover image have changed. Now, we can see that second and third LSB of three cover image pixels are 0 and 1 respectively.LSB has changed for two of these three pixels. If we invert the LSB of these three pixels, cover image pixels will be 10101100, 11101101, 10111011 and 1110110**0.** Now, we can see that there is only one pixel of stego image which differs from cover image i.e. the last one. This results in the improvement in PSNR and hence quality of stego image is improved. For correct de steganography, we need to store the fact that we have inverted the LSBs of those pixels in which second and third LSBs are 0 and 1 respectively.

If we consider two bits there are four possible combinations i.e. 00, 01, 10, 11. For each of these combinations, stego image is analyzed to find the number of pixels of first type i.e. whose LSB has changed and second type i.e. whose LSB has not changed. We invert the LSBs of first type pixels if the number of pixels of first type is greater than the number of second type pixels. In this way, less number of pixels of cover image is modified. The total pixel benefit is equal to the difference between the number of first and second type pixels.

E. *Embedding Algorithm for hiding image*

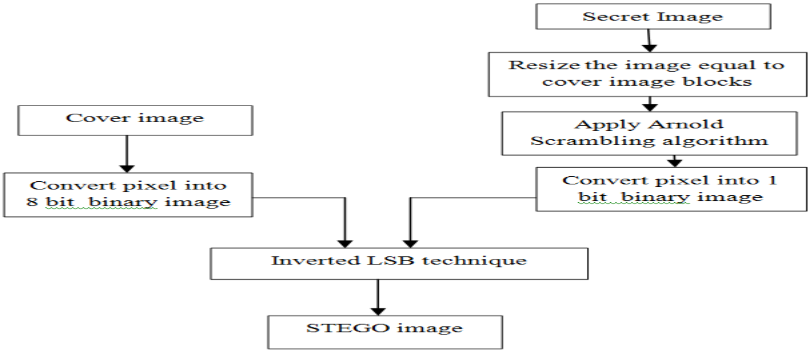

Secret Image
Resize the image equal to cover image blocks
Apply Arnold Scrambling algorithm
Convert pixel into 1 bit binary image
Cover image
Convert pixel into 8 bit binary image
Inverted LSB technique
STEGO image

Fig. 2 Embedding Flowchart for hiding image

D. *Extraction Algorithm*

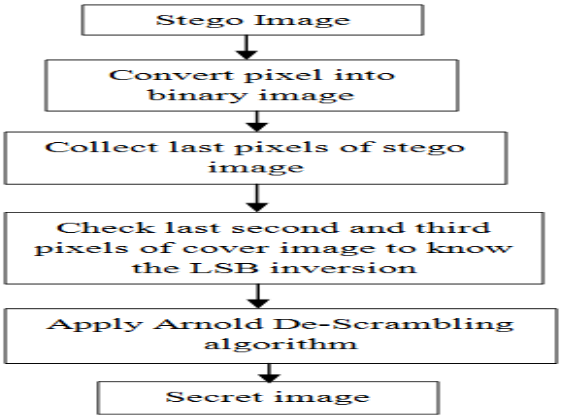

Stego Image
Convert pixel into binary image
Collect last pixels of stego image
Check last second and third pixels of cover image to know the LSB inversion
Apply Arnold De-Scrambling algorithm
Secret image

Fig. 3 Extraction Flowchart

E. *Embedding Algorithm for hiding text*

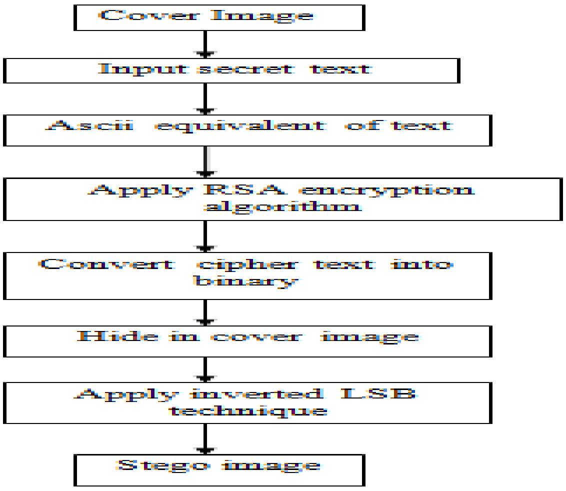

Cover Image
Input secret text
Ascii equivalent of text
Apply RSA encryption algorithm
Convert cipher text into binary
Hide in cover image
Apply inverted LSB technique
Stego image

Fig. 4 Embedding Flowchart for hiding text

III. SIMULATION RESULTS USING MATLAB

A. Results of Embedding algorithm for hiding image

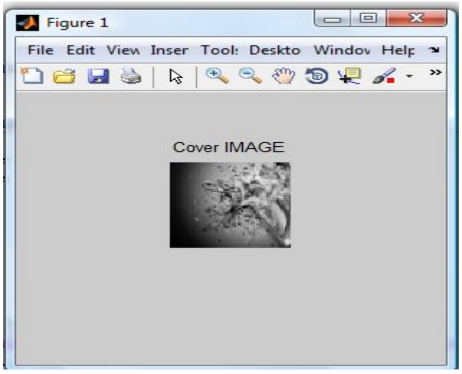


Fig. 5 Input Cover image

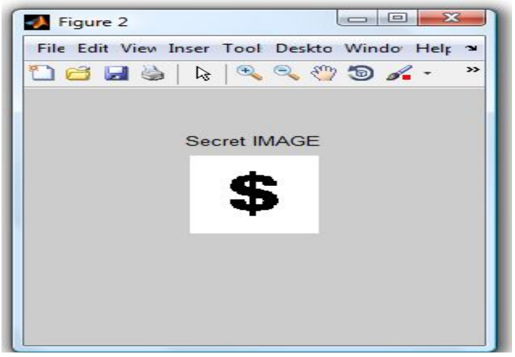


Fig. 6 Message image

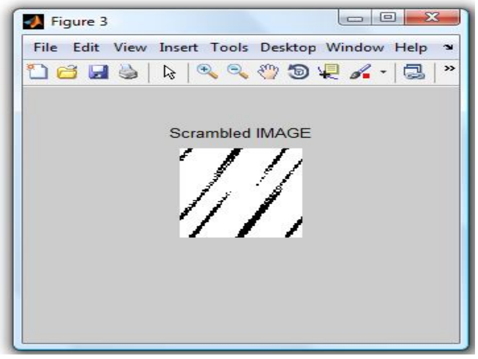


Fig. 7 Scrambled secret image

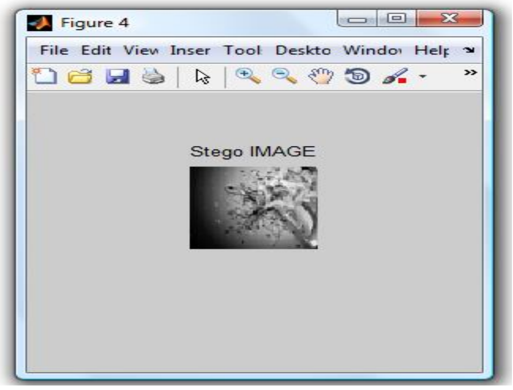
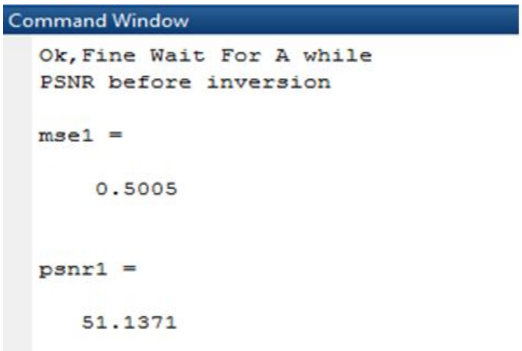
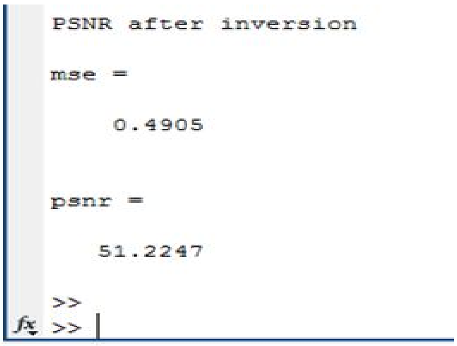


Fig. 8 Output Stego image



```
Ok,Fine Wait For A while
PSNR before inversion

mse1 =

    0.5005


psnr1 =

   51.1371
```

Fig. 9 PSNR output before inversion



```
PSNR after inversion

mse =

    0.4905


psnr =

   51.2247

>>
>> |
```

Fig. 10 PSNR output after inversion

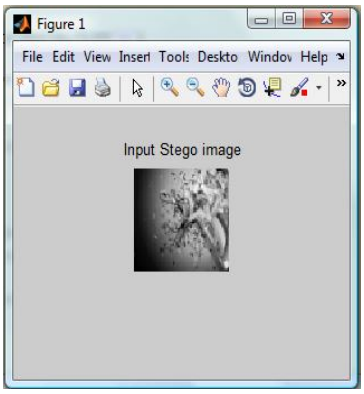
Command window output shows PSNR value after bit inversion technique. PSNR is increased by 0.0876 after applying bit inversion technique. Hence quality of stego image is improved.

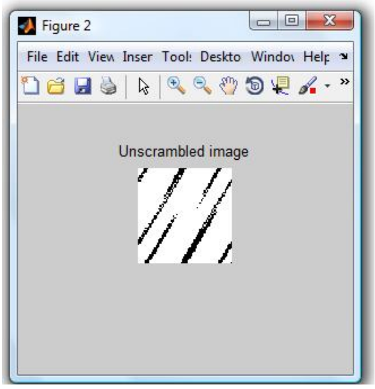
Table shows the Analysis for bit inversion technique for the above cover image

Bit pattern (3^{rd} ,2^{nd} LSB)	Changed bits(cnt00 to cnt11)	Not changed bits(samecnt00 to samecnt11)	Invert	Changed bits in final image
00	8500	7845	YES(8500>7845)	7845
01	8029	8106	NO(8029<8106)	8029
10	8114	8417	NO(8114<8417)	8114
11	8155	8370	NO(8155<8370)	8155

Table 1: Analysis for bit inversion technique for hiding image

B. *Results for Extraction algorithm*





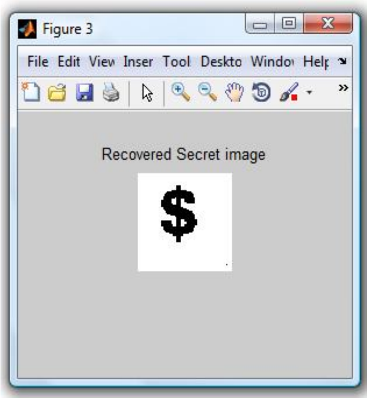
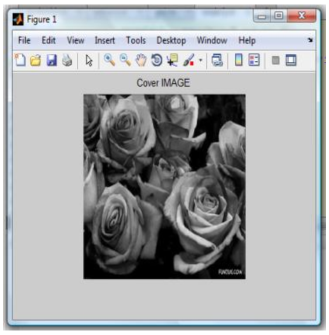


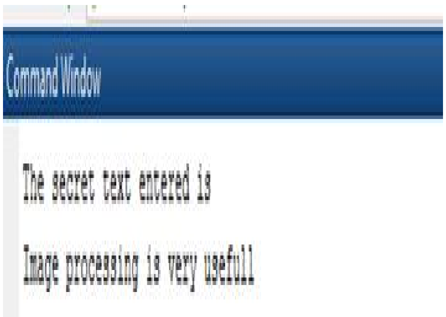
Fig. 11 Input Stego image at Receiver

Fig. 12 Unscrambled secret image

Fig. 13 Recovered Secret Message image

C. *Results of Embedding algorithm for hiding text*





Command Window

```
The secret text entered is

Image processing is very usefull
```

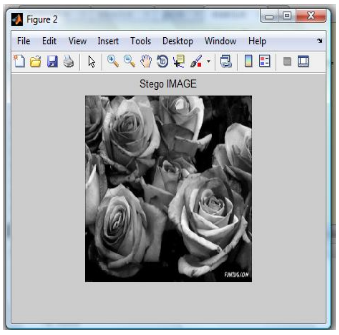
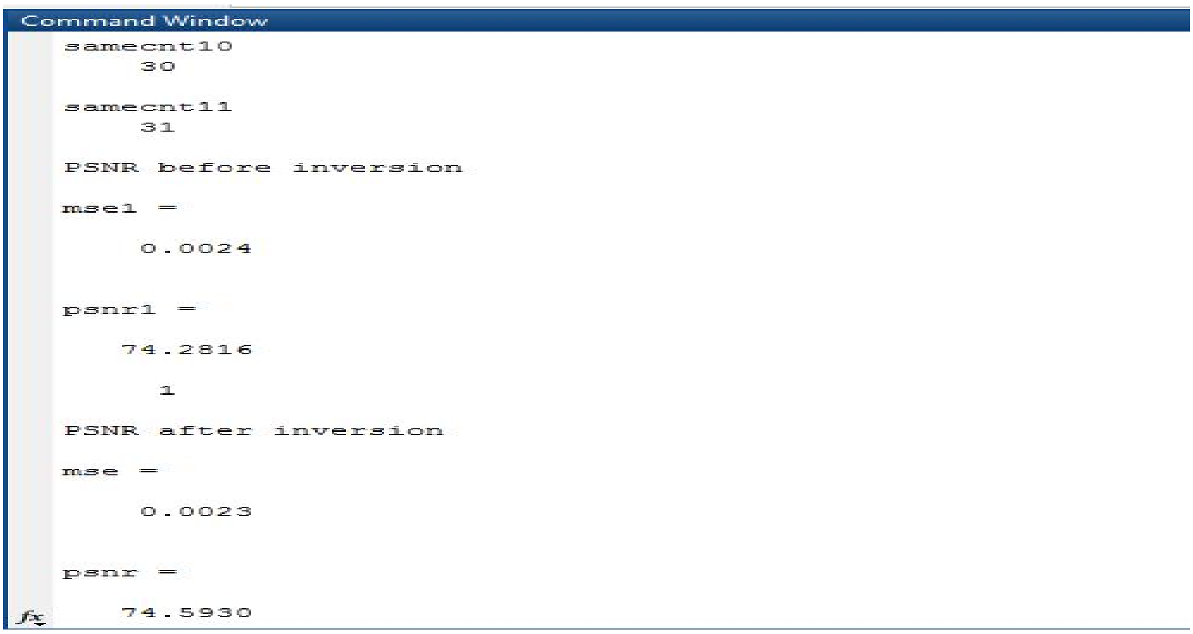


Fig. 14 Input Cover image

Fig. 15 Input secret text

Fig. 16 output Stego image



```
samecnt10
     30

samecnt11
     31

PSNR before inversion

mse1 =

     0.0024


psnr1 =

   74.2816

     1

PSNR after inversion

mse =

     0.0023


psnr =

   74.5930
```

Fig. 17Command window output showing PSNR value before and after applying bit inversion technique

Command window output shows the PSNR value before (only plain LSB technique) and after applying bit inversion technique. We can see that after applying bit inversion technique PSNR is increased by 0.3114 percent. Hence quality of stego image is improved.

Bit pattern (3rd ,2nd LSB)	Changed bits(cnt00 to cnt11)	Not changed bits(samecnt00 to samecnt11)	Invert	Changed bits in final image
00	54	59	NO(59<54)	54
01	38	41	NO(38<41)	38
10	41	30	YES(41>30)	30
11	26	31	NO(26<31)	26

Table 3: Analysis for bit inversion technique for hiding text

V. CONCLUSION

The proposed bit inversion technique improves the stego-image quality. The improvement in PSNR may be very large for some image and for some other image, it may be small. For given a message image cover image is selected for which the improvement is largest. For security enhancement, the message image i.e. is embedded in cover image is scrambled using Arnold algorithm and secret text i.e. is embedded in cover image is encrypted using RSA algorithm. Although the third party could determine that the message bits are embedded, he would have a difficulty to recover it because the message bits are scrambled and encrypted in case of image and text respectively and also some of the LSBs have been inverted; it will misguide the steganalysis process and make the recovery of message more difficult.

Recovery of secret image and text is obtained by using bit inversion technique as well as Arnold descrambling algorithm for image and RSA decryption algorithm for text.

The bit inversion method and use of Arnold scrambling algorithm together makes the steganography better by improving its security, image quality and robustness. Classical LSB algorithm when used with Arnold scrambling algorithm and bit inversion provides user with multilayer of protection so that intended opponent find difficulty to trace hidden information in cover image. Therefore using both cryptography and steganography more security is provided to information.

In future work, other bit combinations of cover image pixels can be considered. Leaving the LSB there are 21 (7C2) bit combinations of two bits in a pixel. We can also consider more bits of cover image pixels for analysis. For example, if we consider three bits it will provide 8 different bit patterns improving the possibility of greater enhancement in PSNR.

REFERENCES

[1] ‘An improved inverted LSB image Steganography’ IEEE 2014 international conference on issues & challenges (ICICT).

[2] S. Dumitrescu, X. Wu, and Z. Wang, “Detection of LSB Steganography via Sample Pair Analysis”, Proceedings of 2003 IEEE Transaction on Signal, vol.51, 2003.

[3] C.S.Tseng and C.C Lin “Hiding data in binary images using LSB” in 2005

[4]Chang, Chin-Chen, and Hsien-Wen Tseng. "Data hiding in images by hybrid LSB substitution." Multimedia and Ubiquitous Engineering, 2009.MUE'09. Third International Conference on. [EEE, 2009.

[5]Almohammad “High capacity Steganography method based upon JPEG” in 2008 third international conference on Availability, Security and Reliability(IEEE),Barcelona

[6] Nadeem Akhtar, Praati Johri “Enhancing the security and quality of LSB based image steganography” 5th international Conference on Computational Intelligence and Communication Networks(IEEE) 2013

[7] ‘A study of various steganography techniques for information hiding’ an international journal of computer science & engineering

[8]Steganography Wikipedia

[9]International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2014 “A proposed method in image steganography to improve image quality with lsb technique “krati vyas, B.L.Pal

[10] Wu, Nan-I., and Min-Shiang Hwang. "Data Hiding: Current Status and Key Issues." IJ Network Security 4.1 (2007): 1-9

[11] Chunxian Song ,Chulin Li, Jing Jing and Guangzhu Xu, “Evaluation of Image Scrambling Degree with Intersecting Cortical Model Neural network” International Journal of Hybrid Information Technology Vol.5,No 2, April 2012

[12] RSA algorithm Wikipedia

DOI: 10.15680/ijircce.2015.0306053