

Optimally Sparse LSB-based Steganography for Secure Transmission of E-learning related Documents

Soumendu Banerjee¹, Akash Nag², Kh Amirul Islam³, Sunil Karforma⁴

¹Assistant Professor, Department of Computer Science, St. Xavier's College, Burdwan

²Faculty of Department of Computer Science, MUC Women's College, Burdwan

³Research Scholar, Department of Computer Science, The University of Burdwan

⁴Professor, Department of Computer Science, The University of Burdwan

Abstract- In an e-learning environment all the transactions have been done via Internet. The messages transmitted between the participants of an e-learning system contain mainly text and images. When sender sends an image to the receiver after having compression, then the extraction makes some sort of distortion of the cover image. So, the receiver will not get the same quality image as sent by the sender. Here we have proposed one model based on LSB steganography, where the receiver will get the cover image without having any distortion. This proposed optimal steganography model is based on authenticating the sender during transmission of text and images among the participants of an e-learning system. We have used LSB steganographic technique to send the secret information by embedding it into the cover image and also calculated the PSNR values. We consider the administrator as the sender and the learner as the receiver for the discussion of our proposed model.

Keywords- E-learning, LSB steganography, Sparse matrix

I. INTRODUCTION

The main participants of an e-learning system are: Administrator, Teachers and Students^[1]. While transmitting documents between the two participants in an e-learning system, in most of the cases the sender sends those after being encoded and when the receiver decodes those documents, they become distorted. There may be some situations, when it is very important that the receiver receives the same documents as send by the sender without having any kind of distortion, in spite of applying encoding and decoding process. To implement our proposed model, we have applied LSB based steganography approach and it also possesses one kind of error detection technique, which helps the receiver recognize if any kind of attack occurs during transmission. Here, we have considered the administrator as sender and learner as receiver. We have wrapped our proposed LSB based^{[2][3]} steganography approach in object oriented model for utilizing the benefits of Object-Oriented analysis and design.

In this paper, we have proposed one LSB based steganography model through which the administrator sends encoded documents to the learner and learner will receive

the same without any changes or distortion. In section II, we have discussed on the organization of the proposed model and section III covers the result and analysis of our proposed model. Finally we have concluded in section IV by showing some future scopes.

II. ORGANIZATION OF PROPOSED OPTIMAL LSB STEGANOGRAPHY

In our proposed optimal LSB steganography model, we have chosen two images, one as cover image and the other as secret image or text (as payload) and the selection must be done in such a way that the cover image will be larger in size than that of the secret image^{[4][5]}. The whole process is divided into two parts, namely, encoding and decoding. Encoding process will be done at the administrator's end and the decoding process will occur at the learner's end. In our proposed model, we have taken all the images in PNG (Portable Network Graphics) format since it follows lossless compression.

The administrator will calculate the length of the payload image and convert the size into bits. Two least significant bits of the first 10 pixels of the cover image are used to store the value of the length of the payload image. Since in our proposed model we have used color images as cover image, 2 bits per RGB channel's color, i.e., 6 bits in each pixel can be used to store the length of the secret image. The limitation of this technique is that the first 10 pixels of the cover image can store maximum of 0 to 2⁵⁹ bits. Thus the size of secret image should not exceed this length and in general, the documents related to an e-learning system do not exceed this range. Since we have considered the cover image as well as the secret image in color image format, so each pixel contains RGB channel, which has 3×8 (=24) blocks to store bit values of which the first 6 bits of each channel are kept unchanged and the last two bits are used to store the length of the payload value. This is shown in details in the Figure 1.1.

Then the administrator will divide the remaining pixels into groups taken 25 pixels at a time. In such a group the first pixel will act as the index where the position will be stored and the remaining 24 pixels are used to store the data. The last 2 bits of each color channel of the first pixel are used to store the index value of the selected position and this is

shown in Figure 1.2. Now, administrator will read the next 6 bits of the payload image and store them at each of the available 24 pixels one at a time and each time will calculate the respective PSNR value. The highest PSNR will give the best result. So, administrator will select that particular pixel where the PSNR value is optimum and place the 6 bits in that pixel by using the LSB steganography technique in the last 2 bits of the pixel at each color channel's LSB. The position of that particular pixel is stored in the first pixel of the group which is used as the index. Since 6 bits of the first pixel are used to store the position, so 2^6-1 (=63) unique locations can be identified using this technique.

For example, after converting the image into binary, suppose we get the first pixel value as the following. Let $R = (130)_{10}$ and its binary equivalent is $(10000010)_2$. Let $G = (210)_{10}$ and its binary equivalent is $(11010010)_2$. Let $B = (70)_{10}$ and its binary equivalent is $(01000110)_2$. Suppose the best PSNR is found in the 12th position. Then we convert it into binary 6 bits (i.e., 001100). Figure 1.2 and Figure 1.3 respectively show how the first pixel will look before and after the embedding.

In our proposed model, since the unique position ranges from 0 to 23, so the remaining values can be used for error detection. While transmitting the document over Internet, if any kind of attack occurs, then this value will be changed and may become greater than 23 which will help the receiver to recognize the attack and the receiver will ask the sender to send the file again. This embedding process will continue until the entire payload has been encoded. If the payload size is not a multiple of 6, then zeros are padded to fulfill the requirement of proposed model. Thus the stego-image is generated and the administrator will send this image to the learner for future use.

After receiving this stego-image, the learner will decode this image for authenticity. We will now discuss this decoding process, which is the reverse of the encoding process. Learner will first read the first 10 pixels of the stego-image and extract 6 bits from each pixel (2 LSBs from each color channel). Through this process he/she will get the total of 60 bits which will indicate the length of the payload image. After finding the length, the learner will divide the remaining pixels of the stego-image into group(s) by taking 25 pixels at a time. The first pixel from each group locates the index value of that particular pixel which informs where the 6 bits of the payload image is hidden. So, the learner can find that value by reading the 6 bits (2 LSBs from each color channel) of the first pixel and identify the position of the hidden data. If this position value is greater than 23, then learner can easily identify that the message is either changed or corrupted. If the position is valid, i.e., ranged from 0 to 23, then reads 6 bits from that particular pixel and this process is continued for all groups. After the completion of reading all the data, concatenate them and if the extra zeros

were padded, which can be recognized according to the actual length of the payload image which was calculated at the very beginning of the decoding process, those can be discarded from the data. After decoding the learner will get the original data sent by the administrator.

Flowchart of proposed Optimal LSBS model

Flowcharts of the encoding and decoding processes, which are discussed in the preceding section, are shown in the Figure 1.4 and Figure 1.5.

III. RESULT AND ANALYSIS OF PROPOSED MODEL

The output of our proposed model is certainly measured by the values of PSNR. Our proposed optimal LSBS model is compatible with image documents as well as text documents. We have taken two sufficiently large size images as cover images which are shown in the Figure 1.6.

The list of secret images is shown in the Figure 1.7. We have taken 10 different images as payload images which are embedded in the above two cover images.

After applying the proposed methodology, the values of PSNR (Rounded up to 4 decimal places), along with their dimensions, are shown in the Figure 1.8 and Table 1.1. In the name column, we have followed the cover_image_secret image format to distinguish between cover and payload images.

For the three images as shown in the first row of Figure 1.8, the PSNR ratio is 64.3800 and the execution time is 9.887 seconds and for remaining three images in the second row as shown in Figure 1.8, PSNR ratio is 68.0262 and the execution time is 17.405 seconds. From the Figure 1.8, we observe that there is no change in the dimensions in both the cases and changes in the cover images are very difficult to recognize. The size of the stego images are also same (24.4MB for Nasa.png and 45.5MB for Deer.png). So, in all other cases, instead of showing the stego images, we use only a table which contains the names of cover images as well as secret images, PSNR values and execution times.

From the Table 1.2, we find the values of PSNR and the execution times of our proposed optimal LSB steganography model of corresponding images. We observe that if the size of a secret image is sufficiently smaller than that of a cover image, then the corresponding value of PSNR is acceptable. If the cover image is not sufficiently large compared to the secret image, then it will be displayed on the screen. From Table 1.1 we see that the cover image Nasa.png is not sufficiently large enough compared to the secret image Home.png. As a result of this, embedding of Home.png into Nasa.png is not possible and consequently PSNR cannot be calculated. We have also applied our models on text documents, which are also as important as images in e-learning. We have chosen 'Nasa.png' as cover

image whose details are given in the Figure 1.6 and the list of secret documents along with their dimensions are shown in Table 1.2. This table also contains the time taken for encoding the text documents into cover image and the respective PSNR values. We have checked our proposed model in respect of various types of text documents, for example, PDF, DOC, TXT and RTF.

Here the dimensions of the output images are not shown in the table because all those have the same dimension and size as the cover image. From Table 5.20, we observe that the PSNR values of the encoded images are within a satisfactory range which means our proposed model is acceptable for various types of documents. The PSNR values are the proof of efficiency of our proposed model.

In this section, we have discussed the decoding part of our proposed optimal LSB steganography model which occurs at the student's end. After applying the decoding process to the embedded images, the student will get the hidden image or document. The retrieved image or document can be kept for further use. This watermark image will authenticate the administrator. So in case of e-learning, our proposed LSB steganography based model may be useful for authentication purpose. The following Table 1.3 shows the retrieved watermarks which have been encrypted in Table 1.2.

From the Table 1.3, we observe that if the size of secret image is fixed, the decoding time varies with the size of cover image proportionally i.e., if the size of cover image increases, then decoding time increases and vice-versa. Again, if the size of cover image is fixed, then the decoding time varies proportionally with the size of secret image. In the same way, we can show that the receiver can also get back the documents like certificate, study material and other essential study related documents associated with e-learning, without having any changes. Another advantage of our proposed model is that the student will get back the document from the stego image in the same format as it was embedded in cover image. So, if the secret document is a text file, then the receiver if needed can also edit that one, and if the administrator doesn't want to give permission to the student to edit the documents, then he/she will send the documents in PDF format.

IV. CONCLUSION

Our proposed model is not only applicable for e-learning system, it is also applicable for other online systems like e-commerce, e-governance etc. It is also very useful for secure transmitting of medical images, since the receiver will get the image without any sort of distortion and due to encoding the size of the files will be less and will be easy to attach within the mail. Future work may include more security while transmitting documents or images over Internet.

V. REFERENCES

[1] Weippl, E. R. (2005). *Security in E-learning*. USA: Springer.
 [2] F.Y.Shih. (2008). *Digital watermarking and Steganography*. London: CRC Press.
 [3] Halder, T., & S.Karforma. (2013). A lsb-indexed steganographic approach to secure e-governance data. *Second International Conference on Computing and Systems-2013* (pp. 158-163). Burdwan: Department of computer science, The University of Burdwan.
 [4] Improved Detection of LSB Steganography in Grayscale Images. *International workshop in information hiding* (pp. 97-115). Springer.
 [5] S.Banerjee, S.Karforma, & A.Nag. (2017). Applying LSB steganography for disseminating academic testimonials in e-learning and its authentication purpose. *International journal of computer trends and technology* , 170-175.

	<p>Soumendu Banerjee has completed his B.Sc(H) in Mathematics, MCA and Ph.D. in Computer Science from The University of Burdwan. He is currently acting as a faculty member in the Department of Computer Science at St. Xavier's College, Burdwan.</p>
	<p>Akash Nag completed his Bachelors in Computer Applications from the University of Burdwan, and his Masters in Computer Science from the University of Calcutta. He received his Ph.D. in Computer Science from the University of Burdwan. He is currently a faculty member in the Dept. Of Computer Science at M.U.C. Women's College, Burdwan. His research interests include algorithms and bioinformatics.</p>
	<p>Kh Amirul Islam has completed his BCA from Dumkal Institute of Engineering and Technology, WBUT and MCA from The University of Burdwan. He has published 4 research papers in journals and conferences.</p>
	<p>Dr. Sunil Karforma has completed B.E. (Computer Science and Engineering) and M. E. (Computer Science and Engineering) from Jadavpur University. He has completed Ph. D. in the field of Cryptography. He is presently holding the post of Professor and the Head of the Department in the Department of Computer Science, The University of Burdwan. His research interests include Network security and e-commerce. He has published numerous research papers in both National and International journals and conferences.</p>

Figures and Tables

R	1	2	3	4	5	6	1	2
G	7	8	9	10	11	12	3	4
B	13	14	15	16	17	18	5	6

Figure 1.1. Bit arrangement: Sample of storing payload length

R	1	0	0	0	0	0	1	0
G	1	1	0	1	0	0	1	0
B	0	1	0	0	0	1	1	0

Figure 1.2. Bit arrangement: cover image before embedding

R	1	0	0	0	0	0	0	0
G	1	1	0	1	0	0	1	1
B	0	1	0	0	0	1	0	0

Figure 1.3. Bit arrangement: cover image after embedding

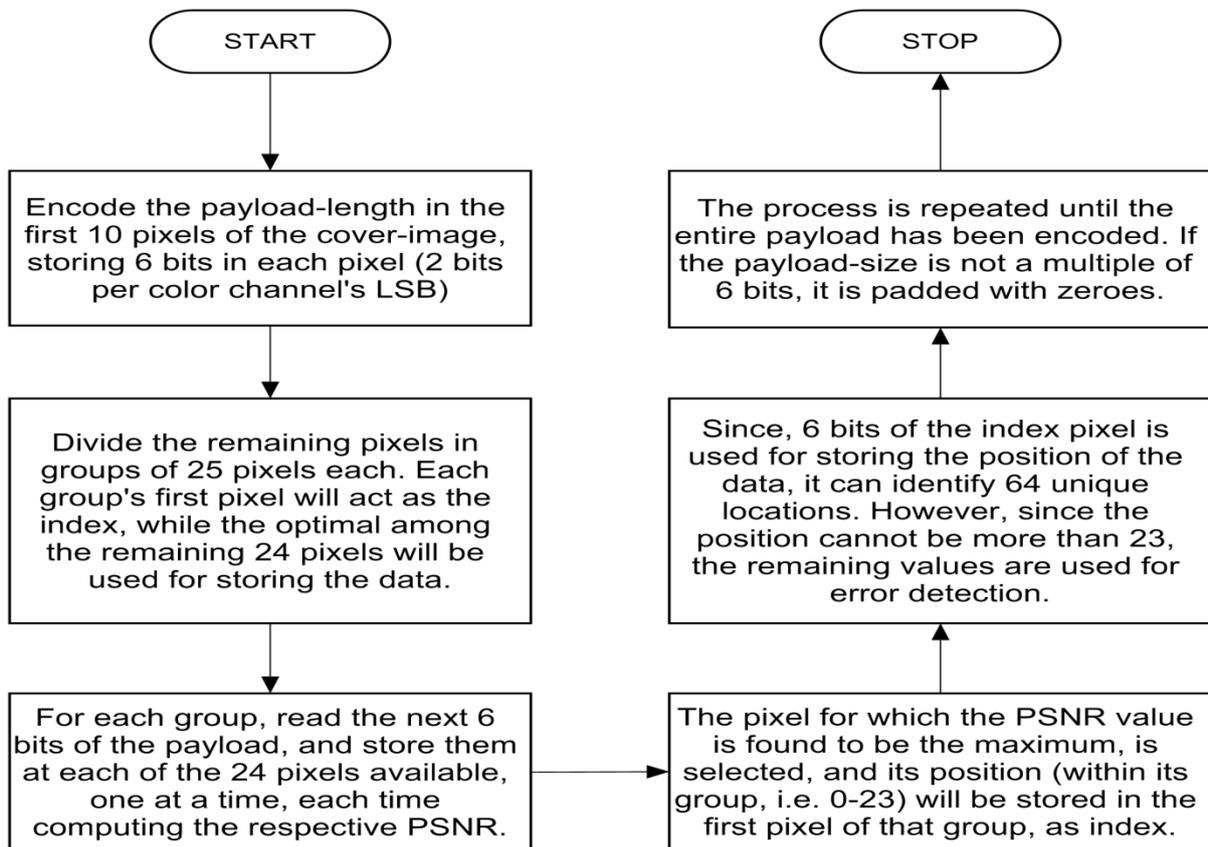


Figure 1.4. Flowchart of encoding process of Optimal LSBS

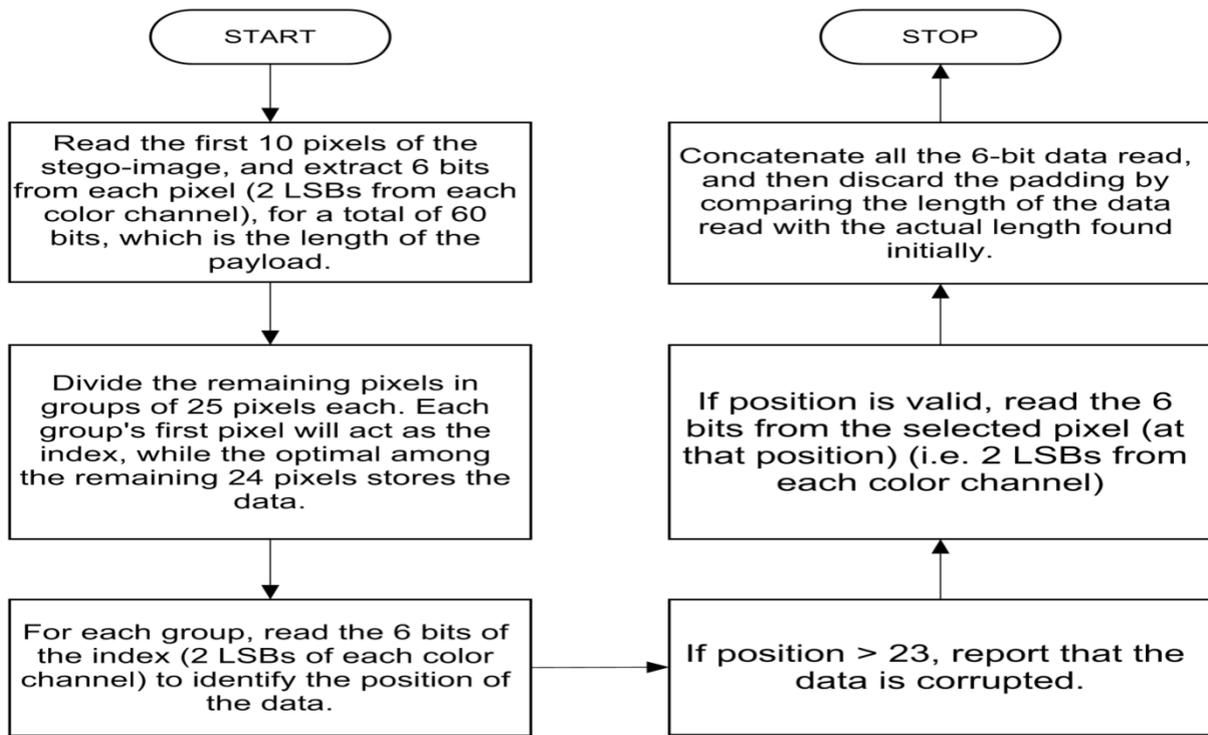


Figure 1.5. Flowchart of decoding process of Optimal LSBS



(a) Deer .png (4608×3456) (27326 KB)



(b) Nasa.png (3444×2484) (12705KB)

Figure 1.6. Selected cover images for Optimal LSBS



(a) Cat.png (300×158) (91.1KB)



(b) University.png (401×154) (141KB)



(c) University_2.png (400×300) (239KB)



(d) Lena.png (256×256) (161KB)



(e) Home.png (420×560) (443KB)



(f) CMS.png [PC: S.De] (292×389) (209KB)



(g) Periwinkle.png [PC: A.Choudhury] (960×720) (89.5KB)



(h) Platform.png (192×256) (109KB)



(i) Road.png (360×270) (216KB)



(j) Sunset.png (320×240) (137KB)

Figure 1.7. Secret images for Optimal LSBs



Cover image(nasa.png) (3444×2484) (12705KB)



Secret image(cat.png) (300×158) (91.1KB)



Stego image(nasa_cat) (3444×2484) (24.4MB)



Cover image (4608×3456) (27326 KB)



Secret image(cat.png) (300×158) (91.1KB)



Stego image(deer_cat) (4608×3456) (45.5MB)

Figure 1.8. Stego images of Optimal LSBs

Table 1.1. List of stego images and related information

SI No.	Cover image	Payload image	Completion time (seconds)	PSNR value
3.	Nasa.png	University.png	11.846	62.5426
4.	Deer.png	University.png	26.586	66.1163
5.	Nasa.png	University_2.png	16.928	68.3811
6.	Deer.png	University_2.png	23.71	63.3581
7.	Nasa.png	Lena.png	11.575	62.0043
8.	Deer.png	Lena.png	19.329	65.5824
9.	Nasa.png	Home.png	0.858	X
10.	Deer.png	Home.png	30.373	61.1881
11.	Nasa.png	CMS.png	14.087	60.9052
12.	Deer.png	CMS.png	21.247	64.4255
13.	Nasa.png	Periwinkle.png	8.674	64.4329
14.	Deer.png	Periwinkle.png	16.536	68.0998
15.	Nasa.png	Platform.png	10.031	63.6468
16.	Deer.png	Platform.png	17.192	67.2261
17.	Nasa.png	Road.png	14.165	60.7821
18.	Deer.png	Road.png	21.372	64.2593
19.	Nasa.png	Sunset.png	10.749	62.7032
20.	Deer.png	Sunset.png	18.33	66.2612

Table 1.2. List of secret documents with respective PSNR values and execution time

Secret documents	Dimension	Completion time	PSNR
Doc_sample1.pdf	7.75KB	5.616 seconds	74.8671
Doc_sampl2.pdf	53.5KB	7.363 seconds	66.4050
Doc_sample3.pdf	127KB	10.296 seconds	63.0376
Doc_sample4.pdf	468KB	0.858 seconds	Cover image is not sufficiently large enough
Doc_sample5.doc	38.5KB	6.692 seconds	67.7405
Doc_sample6.doc	133KB	10.468 seconds	62.7882
Doc_sample7.docx	250KB	15.086 seconds	60.1655
Doc_sample8.txt	244KB	14.945 seconds	60.3398
Doc_sample9.rtf	12.8KB	5.741 seconds	72.8638
Doc_sample10.docx	203KB	13.385 seconds	61.0585

Table 1.3. List of retrieved watermark images

Stego Image	Retrieved Watermark	Decoding Time (Second)
 Nasa_Cat.png	 Cat.png	1.436
		1.482

Deer_Cat.png



Cat.png



1.279

Nasa_Church.png



Church.png



1.466

Deer_Church.png



Church.png



1.342

Nasa_Church2.png



Church_2.png



1.42

Deer_Church2.png



Church_2.png



1.217

Nasa_Lena.png



Lena.png



1.529

Deer_Lena.png



Lena.png



2.886

Deer_Home.png

Home.png



Nasa_CMS.png



CMS.png

2.886



Deer_CMS.png



CMS.png

1.248



Nasa_Periwinkle.png



Periwinkle.png

0.671



Deer_Periwinkle.png



Periwinkle.png

0.889



Nasa_Platform.png



Platform.png

1.233



Deer_Platform.png



Platform.png

1.513



Nasa_Road.png



Road.png

1.319



Road.png

1.279

Deer_Road.png



Road.png

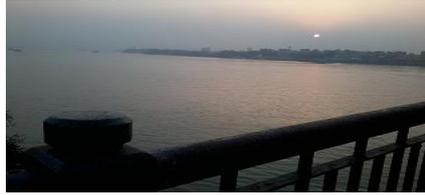


0.874

Nasa_Sunset.png



Sunset.png



0.936

Deer_Sunset.png

Sunset.png