International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 3
ISSN: 2454-4248
536 – 540

Efficient LSB Embedding Using Boosted Steganography

Sukhjit Singh, Research Scholar,
Sant Longowal Institite of Engg. and Technology, Longowal, India
sukhjit_singh@rediffmail.com

Dr. V.K Jain
Sant Longowal Institite of Engg. and Technology, Longowal, India
vkjain27@yahoo.com

Dr. Amanpreet Singh
IKG PTU university, Kapurthala
Amanpreet1970@gmail.com

***Abstract*—**Steganography is the science of hiding information in a multimedia carrier such as an image, audio or video file. The method used to hide information prevents attackers from revealing it and sharing it with others. Digital images such as bitmap files are the mostly used carriers for steganography. This paper presents a new technique that embeds secret message in the LSB of the cover image. In boosted steganography, the cover image is preprocessed before the message is embedded into the LSBs. The secret data is first converted into stream of bits and compared with the LSBs of cover image. The level of distortion is noted. The combination of brightness and contrast of the image is varied and then LSBs of cover image are compared with the secret data bit stream. The point at which distortion level is reduced to the minimum is determined. Finally the secret data is embedded in the image having a particular selected value of brightness and contrast. The experiment measures total number of matches and mismatches between LSB bits and secret data for different levels of brightness and contrast of the image. The results show improvement in the existing scheme of LSB embedding.

Keywords*-Boosted;Steganography; algorithm; extraction;*

*****

I. INTRODUCTION

Information security is one of the main issue that needs to be dealt with today. Digital steganography can be used to protect the secret data. We can hide the secret data within images, audio or video files. These carrier files can be transmitted through different public channels without any indication that information is hidden in these files. There is another field called steganalysis that deals with how to detect a hidden message in any media such as these carrier files.

The main aim of steganalysis is to detect whether a given media contains secret information. Some approximation techniques are used to determine the abnormal behavior of a regular image or video file to determine the existence of secret message. The existence of anomalies can vary from small unnoticeable magnitude to large visible effects.

The word steganography comes from the Greek Steganos, which means covered or secret and graphy means writings or drawing. The main goal of steganography is to communicate in a secure and undetectable manner without any suspicion. If a steganography method causes someone to reveal the secret or existence of message in the carrier then the method become useless. Steganography provides high level of secrecy and security by combining with cryptography. There are many examples of using steganography in the history. In world war 2, invisible ink was used to write on a piece of paper. The paper appeared blank to the average person. Instead of ink, liquids such as mil, vinegar and fruit juices were used. When such substances are heated they darken and become visible to the human eye. The Greece used to select messengers and shave their head, they would then write a message on their head. After the message had been written, the hair was allowed to grow back. After the hair grew back the messenger was sent to deliver the message. The recipient would shave off the messenger's hair to see the secret message.

In image steganalysis the process is opposite to that of image steganography. Image steganalysis is a way to detect presence of secret message in a carrier. It may further estimate the potentially hidden information from a given image.

II. RELATED WORK

In adaptive steganography [1] a variable number of message bits are embedded in different image pixels. First statistical analysis is done and then embedding depth k is found in a image pixel. After this K number of message bits is embedded in k LSBs in image pixel. In this scheme, capacity and security has been enhanced.

Kekre et al. [2] suggested how to embed variable number of secret message bits in different cover image pixels. The method distributes the image pixel values into four different ranges, R1, R2, R3 and R4. Then I number of bits are embedded in range R_i for i=0 to 3. The use of adaptive bits increases the security of algorithm

Mathkour et al. [3] used a method based on spiral LSB substitution technique. In this technique RGB color channel component of cover image is used for LSB substitution. The cover is divided into several parts and a different scheme is used on each part of the cover image. The method uses three mechanisms corresponding to the three sequences. First is 'start from the corner' second is 'start from centre' and third is 'hybrid'. Two sequencing mechanisms used are counter

 ISSN: 2454-4248
536 – 540

clockwise direction and clock wise direction. This method only increases the security.

Mishra et al. [4] suggested a new version involving LSB method. The secret binary message is first divided into 8-bit blocks and image cover is divided into 8-pixel blocks. A random number generator is used to select a 8-pixel block and the 8-bit message is embedded in it. The message is distributed to various cover image blocks.

Swain and Lenka [5] used another technique where substitution takes place alternatively at 7^{th} and 8^{th} bit positions. First the information is encrypted and then embedded to the cover image as follows. The first cover byte uses 7^{th} bit position to hide a message bit and second cover byte uses 8^{th} bit position, third cover byte again uses 7^{th} bit position and then 8^{th} bit position and so on. This scheme provides two levels of security.

Rosziati and Teoh Suk [6] use steganography imaging system (SIS) technique. The method is used to hide data in cover image. At the receiver end, reverse method is used to retrieve the secret data back using a key. During the process of hiding message data, a secret key is calculated. This method provides accuracy, privacy and confidentiality by the use of secret key.

A moderate bit substitution method is proposed by Pharwaha [7], in which 4^{th} LSB cover bit is replaced by the secret data bits. After embedding data bit, a post pixel adjustment process is used to improve the quality of stego-image. This scheme only enhances the security.

Jain and Ahirwal [8] use a private stego-key for embedding mechanism. The stego-key uses five dark level ranges. The chosen key uses five territories and substitutes diverse number of bits in LSBs. The technique covers up to four bits in a few pixels.

Rig and Themrichon [9] proposed an image steganography using Huffman encoding. Two pictures of size MxN and PxQ are used as cover pictures and a secret picture. Huffman encoding is performed over the secret message or secret picture before it is embedded inside the cover picture. The Huffman table and length of the bit stream of Huffman encoding is also inserted in the cover image.

III. LSB Embedding Method

A bitmap file embedding, uses a simple method to embed data into a cover file. In this method the file is extracted in a memory buffer where file header is separated from the rest of the pixel bytes. The secret data is then converted into stream of bits and then it is embedded into the cover LSB bits. To embed two characters “A” and “B” for example, first these converted into bits, then the bits directly overwrites the LSB bits of the cover image. Use of simple embedding method is easy at the cost of lesser embedding efficiency. The embedding efficiency can be further increased if additional techniques are adopted as discussed in the next section.

IV. Proposed Work

In the proposed method our main objective is to reduce the distortion of the cover image by preprocessing it before embedding the data. The method involves changing of contrast value of the image in small steps until we find the maximum number of matches between the LSB cover image bits and secret data bits. When the contrast of an image changes, the LSB or higher bits change. As contrast is changed in small steps, intensity of some of the pixels increases while intensity of other pixels decreases. Even if the intensity of pixels continuously increases, the lower bits randomly changes. To calculate the contrast of an image we first need to calculate the contrast correction factor. If C denotes the contrast of an image then the contrast correction factor F, is given by the following formula.

$$F = \frac{259(C + 255)}{255(259 - C)}$$

The value of C is an integer and can vary from -255 through 0 to +255 and the value of F is a real number and varies from 0 through 1 to 128. After calculating the contrast factor, we calculate the contrast adjustment by use of the following formula

$$R' = F(R - 128) + 128$$

Here R is the red component of the pixel before the adjustment of the contrast value and R’ is the red component of the pixel after the adjustment of the contrast value. The contrast of an image decreases if the value of C decreases from 0 to -255 and the contrast of an image increases if the value of C increases from 0 to +255. Figure 1 shows two images, the first one with contrast value decreased from normal 0 to -120 and the same image with contrast value increased from normal 0 to +120. Similarly figure 2 shows another pair of images one with contrast decreased to -120 from 0 value and the same image with contrast increased to +120 from 0 value. In our case the contrast of an image is changed in small steps around the value zero to a range of -10 to +10 so that there is no change in visibility of the image. Only the LSB bits change under these conditions fullfilling our requirement.

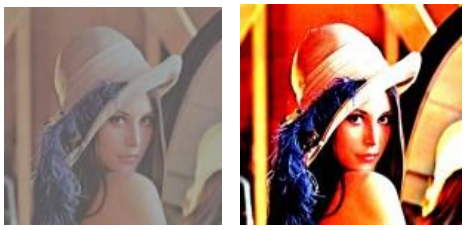

Figure 1. Contrast levels of -120 and +120 respectively

 ISSN: 2454-4248
536 – 540

Figure 2. Contrast levels of -120 and +120 respectively

V. EXPERIMENTAL SETUP

We have developed an algorithm and is implemented and tested over several images. Here we have shown results for three standard images of 256x256 sizes. Embedding and extraction has been done using C++ program implemented using the proposed algorithm. A 16kB of random secret data is embedded in three different images. The contrast of each original image is varied by small amount and re-embedding is done. The contrast level at which the maximum embedding efficiency is obtained, is the optimized value.

VI. RESULTS

Table I shows embedding of 16kB of data bits to corresponding cover Image-1. The table shows variation of contrast around value 0 in the range -10 to +10. The table shows the contrast correction factor corresponding to each value of contrast chosen. The 16kB of secret data is embedded. The LSB Mismatch bits and LSB match bits shows the embedding efficiency at the corresponding contrast level. The last column of the table shows the percentage of data bits match with that of LSB bits of cover image. It is seen that at contrast level -8 we get the maximum matches of 8930 (54.51%) and mismatches of 7454.

Figure 3 shows the graph between the data bits that match with the LSB and the data bits that do not match with the LSB. From the graph it is clear that as the number of matches increase there is corresponding decrease in the mismatches. Similarly Table II and III show the variation of contrast along with variation of matches and mismatches. These tables show that the maximum embedding efficiency is obtained when the contrast level is at -6 and +10 respectively. Figures 4 and 5 show the variation of matches and mismatches of LSB bits as the contrast level is changed.

TABLE I
16KB PAYLOAD EMBEDDED IN IMAGE-1

SNo	Contrast level (C)	Contrast correction factor (*F*)	LSB Mismatch bits	LSB Match bits	%Data Match
1	-10	0.925	7930	8454	51.6
2	-9	0.932	7757	8627	52.66
3	-8	0.94	7454	8930	**54.51**
4	-7	0.947	7806	8578	52.36
5	-6	0.954	8332	8052	49.15
6	-5	0.962	8328	8056	49.17
7	-4	0.969	7798	8586	52.41
8	-3	0.977	8786	7598	46.38
9	-2	0.985	7935	8449	51.57
10	-1	0.992	8366	8018	48.94
11	0	1	7629	8755	53.44
12	1	1.008	7624	8760	53.47
13	2	1.016	8258	8126	49.6
14	3	1.024	8787	7597	46.37
15	4	1.032	7722	8662	52.87
16	5	1.04	8371	8013	48.91
17	6	1.048	8237	8147	49.73
18	7	1.056	7590	8794	53.68
19	8	1.064	7953	8431	51.46
20	9	1.073	8920	7464	45.56
21	10	1.081	7563	8821	53.84

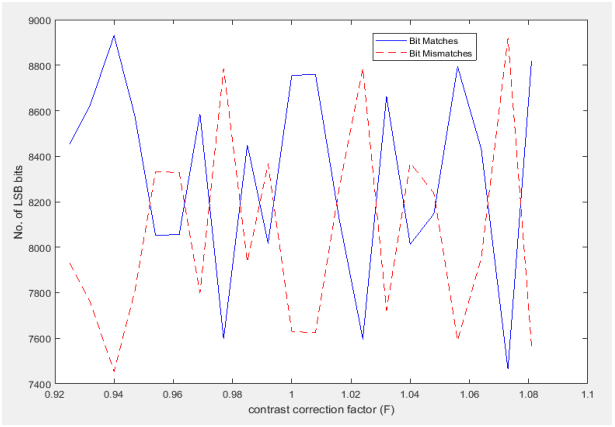

Figure 3. LSB Bit matches and mismatches as contrast changes for Image-1

TABLE II
16KB PAYLOAD EMBEDDED IN IMAGE-1

SNo	Contrast level	Contrast correction factor (*F*)	LSB Mismatch bits	LSB Match bits	%Data Match
1	-10	0.925	7947	8437	51.5
2	-9	0.932	8679	7705	47.03
3	-8	0.94	8945	7439	45.41
4	-7	0.947	8579	7805	47.64
5	-6	0.954	7522	8862	**54.09**
6	-5	0.962	8437	7947	48.51
7	-4	0.969	8078	8306	50.7
8	-3	0.977	8371	8013	48.91

ISSN: 2454-4248
536 – 540

9	-2	0.985	8818	7566	46.18
10	-1	0.992	8766	7618	46.5
11	0	1	8248	8136	49.66
12	1	1.008	7721	8663	52.88
13	2	1.016	7827	8557	52.23
14	3	1.024	8160	8224	50.2
15	4	1.032	7527	8857	54.06
16	5	1.04	7781	8603	52.51
17	6	1.048	8048	8336	50.88
18	7	1.056	8532	7852	47.93
19	8	1.064	7844	8540	52.13
20	9	1.073	8527	7857	47.96
21	10	1.081	7539	8845	53.99

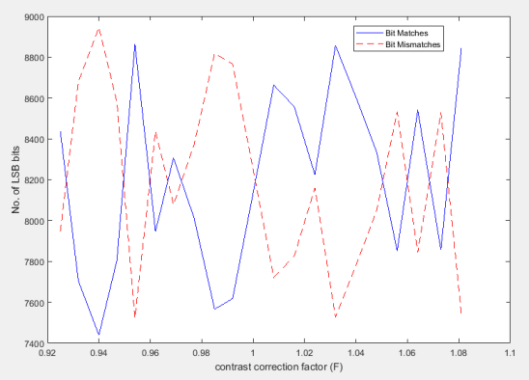

Figure 4. Variation of LSB Bit matches and mismatches as contrast level changes for Image-2

TABLE III
16KB PAYLOAD EMBEDDED IN IMAGE-2

SNo	Contrast level	Contrast correction factor (*F*)	LSB Mismatch bits	LSB Match bits	%Data Match
1	-10	0.925	8517	7867	48.02
2	-9	0.932	7688	8696	53.08
3	-8	0.94	8502	7882	48.11
4	-7	0.947	8864	7520	45.9
5	-6	0.954	7937	8447	51.56
6	-5	0.962	7616	8768	53.52
7	-4	0.969	7855	8529	52.06
8	-3	0.977	8247	8137	49.67
9	-2	0.985	8836	7548	46.07
10	-1	0.992	8150	8234	50.26
11	0	1	8333	8051	49.14
12	1	1.008	8337	8047	49.12
13	2	1.016	8138	8246	50.33
14	3	1.024	8227	8157	49.79
15	4	1.032	8017	8367	51.07
16	5	1.04	8379	8005	48.86
17	6	1.048	7650	8734	53.31
18	7	1.056	8320	8064	49.22
19	8	1.064	7758	8626	52.65
20	9	1.073	7503	8881	54.21
21	10	1.081	7406	8978	**54.8**

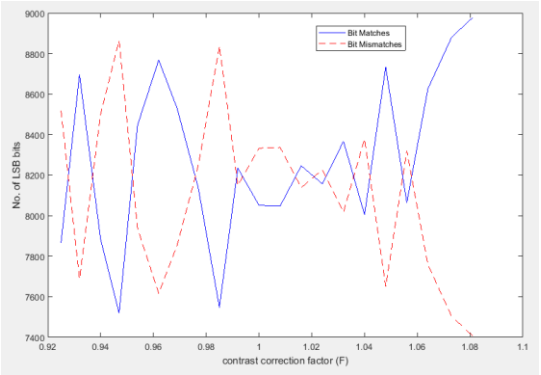

Figure 5. Variation of LSB Bit matches and mismatches as contrast level changes for Image-3

VII. CONCLUSION

This paper presents a new data hiding method based LSB hiding. The contrast of the image is changed before embedding the data. The contrast level is changed within a small range so that visibility of the original image does not change to large extent. From the results it is clear that the proposed method gives better results as compared to the existing LSB embedding scheme. The total number of matches and mismatches changes randomly as shown by figures 4 and 5 by keeping the contrast level within a small range so that there is no visual change in the original image. The results obtained in tables I, II and III from the three images shows that the total number of matches increases and hence improves the PSNR value of the stego images.

REFERENCES

[1] J. He, S. Tang and T. Wu, "An Adaptive Image Steganography Based on Depth-Varying Embedding," 2008 Congress on Image and Signal Processing, Sanya, China, 2008, pp. 660-663. doi: 10.1109/CISP.2008.189

[2] Kekre, Hemant & Athawale, Archana & Halarnkar, Pallavi. (2008). Increased Capacity of Information Hiding in LSB's Method for Text and Image. International Journal of Electrical, Computer, and Systems Engineering.

[3] H. Mathkour, G. M. R. Assassa, A. A. Muharib and I. Kiady, "A Novel Approach for Hiding Messages in Images," 2009

ISSN: 2454-4248
536 – 540

International Conference on Signal Acquisition and Processing, Kuala Lumpur, 2009, pp. 89-93. doi: 10.1109/ICSAP.2009.36

[4] A. Mishra, A. Gupta and D. K. Vishwakarma, "Proposal of a New Steganographic Approach," 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, 2009, pp.175-178. doi: 10.1109/ACT.2009.52

[5] G.Swain and S.K. Lenka, "steganography-Using a Double Substitution Cipher." International Journal of Wireless Communications and Networking, Vol. 2, No. 1, pp.35-39, 2010.

[6] Ibrahim, Rosziati & Suk Kuan, Teoh. (2010). Steganography Imaging System (SIS): Hiding Secret Message inside an Image. Lecture Notes in Engineering and Computer Science. 2186.

[7] A.P.S Pharwaha, "Secure Data Communication using Moderate Bit Substitution for Data Hiding with Three Layer Security" IE (I) Journal-ET, Vol. 91, pp.45-50, 2010.

[8] Y.K. Jain and R.R. Ahirwal, "A Novel Image Steganography Method With Adaptive Number of Least Significant Bits Modification Based on Private Stego-Keys" International Journal of Computer Science and Security, Vol.4, No. 1, pp.40-49, 2010.

[9] R.Das and T. Tuithung, "A Novel Steganography Method for Image Based on Huffman Encoding" IEEE, ISBN: 978-1-4577-0748-3, 2012.