I J C T A, 9(34) 2016, pp. 525-535
© International Science Press

A Survey on Image Steganography Based on Edge Adaptive Least Significant Bit Matched Revisited (EALSBMR) Algorithm

Smitha G. L.* and E. Baburaj**

ABSTRACT

Existing image steganographic methods lack in the complexity, which can be utilized by the radical to decode the images and neutralize the operations. Several methods have been proposed in order to combat this. Perhaps the most efficient method is Edge Adaptive Least-Significant-bit Matched Revisited (EALSBMR)-based approach. It is a famous type of steganographic methods in the spatial domain. In this paper we are reviewing the two

Steganography algorithms - Least Significant Bit and (EALSBMR)-based approach. This paper considers digital images as covers and verifies an adaptive and secure data hiding scheme in the spatial domain.

Keywords: Steganography, Least Significant Bit (LSB) based steganography, Edge Adaptive Least Significant Bit Matched Revisited (EALSBMR) Algorithm.

I. INTRODUCTION

The word Steganography is derived from the Greek words stegos meaning cover and graûa meaning writing [1] deûning it as covered writing. In image Steganography the information is hidden exclusively in images. Steganography is the art and science of secret communication .It is the practice of encoding/embedding secret information in a manner such that the existence of the information is invisible. The original ûles can be referred to as cover text, cover image, or cover audio. After inserting the secret message it is referred to as stego-medium. A stego-key is used for hiding/encoding process to restrict detection or extraction of the embedded data [2].

Steganography diûers from cryptography [6].

- Steganography Hide the messages inside the Cover medium, Many Carrier format.
- Breaking of Steganography is known as Steganalysis.
- Cryptography Encrypt the message before sending to the destination, no need of carrier/cover medium.
- Breaking of cryptography is known as Cryptanalysis.

Watermarking and ûngerprinting related to Steganography are basically used for intellectual property protection. A digital watermark is a kind of marker covertly embedded in a noise-tolerant signal such as audio or image data. It is typically used to identify ownership of the copyright of such signal. The embedded information in a watermarked object is a signature refers the ownership of the data in order to ensure copyright protection. In ûngerprinting, diûerent and specific marks are embedded in the copies of the work that different customers are sup- posed to get. In this case, it becomes easy for the property owner to ûnd

* Research Scholar, Dept of CSE, Sathyabama University, Chennai, *E-mail: smithagl2016@gmail.com*

** Professor, Dept of CSE , Narayanaguru College of Engineering,Manjalumoodu, *E-mail: alanchybabu@gmail.com*

out such customers who give themselves the right to violate their licensing agreement when they illegally transmit the property to other groups [1], [5].

This paper considers digital images as covers and verifies an adaptive and secure data hiding scheme in the spatial domain with the help of EALSBMR algorithm.

II. LITERATURE REVIEW

The term Steganography came into use in 1500s after the appearance of Trithemius book on the subject Steganographia [3].

(a) Past: The word Steganography technically means covered or hidden writing. Its ancient origins can be traced back to 440 BC. Although the term Steganography was only coined at the end of the 15th century, the use of Steganography dates back several millennia. In ancient times, messages were hidden on the back of wax writing tables, written on the stomachs of rabbits, or tattooed on the scalp of slaves. Invisible ink has been in use for centuries for fun by children and students and for serious undercover work by spies and terrorists [7].

(b) Present: The majority of today's steganographic systems uses multimedia objects like image, audio, video etc as cover media because people often transmit digital pictures over email and other Internet communication. Modern Steganography uses the opportunity of hiding information into digital multimedia ûles and also at the net- work packet level [4].

Hiding information into a medium requires following elements [2].

1. The cover medium(C) that will hold the secret message.
2. The secret message (M), may be plain text, digital image ûles or any type of data.
3. The steganographic techniques.
4. A stego-key (K) may be used to hide and unhide the message.

In modern approach, depending on the cover medium, Steganography can be divided into ûve types: 1. Text Steganography 2. Image Steganography 3. Audio Steganography 4. Video Steganography 5. Protocol Steganography.

- Text steganography: Hiding information in text ûles is the most common method of Steganography. The method was to hide a secret message into a text message. After coming of Internet and diûerent type of digital ûle formats it has decreased in importance. Text stenography using digital ûles is not used very often because the text ûles have a very small amount of excess data.
- Image steganography: Images are used as the popular cover medium for Steganography. A message is embedded in a digital image using an embedding algorithm, using the secret key. The resulting stego-image is send to the receiver. On the other side, it is processed by the extraction algorithm using the same key. During the transmission of stego-image unauthenticated persons can only notice the transmission of an image but can't see the existence of the hidden message.
- Audio steganography: Audio Steganography is concerned with embedding information in an innocuous cover speech in a secure and robust manner. Communication and transmission security and robustness are essential for transmitting vital information to intended sources while denying access to unauthorized persons. An audible, sound can be inaudible in the presence of another louder audible sound. This property allows selecting the channel in which to hide information [2]. Existing audio Steganography software can embed messages in WAV and MP3 sound ûles. The list of methods that are commonly used for audio Steganography are listed and discussed below.

- LSB coding
- Parity coding
- Phase coding
- Spread spectrum
- Echo hiding
- Video steganography: - Video Steganography is a technique to hide any kind of ûles in any extension into a carrying Video file.
- Protocol Steganography:- The term protocol Steganography is to embedding information within network protocols such as TCP/IP. We hide information in the header of a TCP/IP packet in some fields that can be either optional or are never used [8].

III. LEAST SIGNIFICANT BIT ALGORITHM

In this section, the LSB steganography algorithm is discussed. It is one of the oldest steganography algorithms that embed the message bits into the stego-image.

(a) LSB Description: It is a well-known data-hiding technique used widely because of its straightforwardness. It conducts a modification to the least significant bit of the stego- image pixels, which change only the tone of the color [10]. This change is so slight that the human eye may not notice it. The LSB hides the message bits into the image pixels either in a sequential or randomized fashion. It creates a path for replacing the least significant bits of the image with the message bits. If the path is randomly generated then the pseudo random number generator PRNG is used [9]. The PRNG should be seeded with some stego-key that is shared between the sender and receiver. In this way the message bits will be spread over the stego-image.

The LSB algorithm is depicted in Figure 1. First a path is created that is used to select the pixels. These pixels are selected in a random order based on a stego-key. For each bit of the secret message, a pixel is chosen from the cover image based on the path. We would then replace the least significant bit of the cover pixel with the bit of the secret message. The algorithm hides the length of the secret message beside the message itself.

```
Input: A cover image C and a message m
Output: A stego-image S
Generate random sequences using seed (stego-key) save as path[]
index = 1
for i = 1 to length(m) do
    n = path[index]
    S_n ← LSB(C_n) = m_i
    index ← index + 1
end for
```

Figure 1: LSB Steganography Algorithm

The extraction phase is the inverse of the embedding phase. At the receiver side the path is created based on the stego-key. First the length of the secret message is recovered by retrieving the least significant bits of the pixels. Then the pixels are traversed based on the path and least significant bit of each pixel is retrieved. This process of traversing all the pixels continues until reaching the end of the message length.

(b) Example of LSB: Let's assume that we want to embed the letter 'A' into a 24-bit cover image. The binary value of 'A' is 10000011. Assume the three adjacent pixels of the image are the following:

(10110100 11010111 10001110)
(00011100 11110110 11010111)
(10001110 00011100 11100101)

After applying LSB steganography algorithm the following pixels of stego-image is acquired. Bits that have been changed because the cover image pixels did not match the message bits are represented in red.

(1011010**1** 1101011**0** 10001110)
(00011100 11110110 1101011**0**)
(1000111**1** 0001110**1** 11100101)

The algorithm first selects a pixel (x_i) sequentially. Then it checks whether the least significant bit of (x_i) matches with the message bit (m_i). Least significant bit of a pixel is the redundant bit which is the most right bit of a byte. If $LSB(x_i) = m_i$, then no change otherwise LSB of x_i is substituted with m_i. Then it selects the next pixel and message bit and checks whether they match or not. This process continues until reaching the end of secret message bits where all secret bits are embedded in the image.

Figure 2 displays two images of gray-scale flower. Figure 2(a) is the original cover image and Figure 2(b) is the stego-image with a message "Sathyabama University Ph.D Thesis Least Significant Bit Approach" is hidden inside it. The LSB algorithm is applied and the stego-image Figure 2(b) is produced. The pixels are selected randomly using PRNG.

(a)

(b)

Figure 2: (a) Cover image of a Flower. (b) Stego-image with a hidden message

(c) Analysis of LSB: The LSB steganography algorithm is straightforward to comprehend. Furthermore, implementation of the LSB algorithm includes low CPU cost and complexity. This section presents some features the algorithm preserves.

Invisibility: The LSB algorithm exploits the fact that human eyes do not perceive the small color modifications. Karaman et al. [11] stated that modifying up to 4^{th} least significant bits are not perceivable by human naked eye. For that reason many algorithms are suggested as improvement to this simple LSB-approach to hide the secret message in different level of least significant bit [9].

Capacity: The hiding capacity rate of the algorithm for an 8-bit depth gray-scale image is at minimum 1 bpp and for a 24-bit image is minimum 3 bpp. "bpp" is a measure used to determine the capacity rate of embedding the message bits into a pixel and stands for bit per pixel. In other words, bpp stands for number of secret bits a pixel of a cover image can hold. It can be noted that as the stego-image size increases, the capacity of embedding the secret message increases. This indicates that a significant amount of information can be hidden using the LSB algorithm. The probability of the expected number of bits modifications is equal to 0.5 in case of maximum embedding rate of 1 bpp.

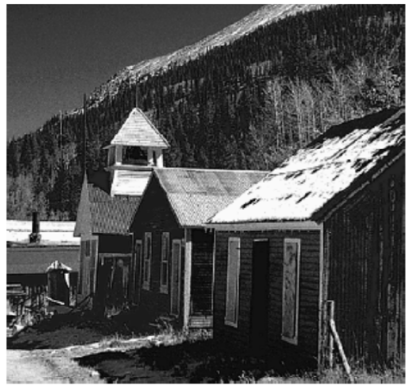
Security: The stego-key is shared between the sender and receiver to assure the correct extraction of the secret message bits. The stego-key ensures that the security is preserved and only the sender and receiver who possess the key can extract the secret message.

(d) Limitation of LSB: The LSB algorithm is widely used as a simple image steganography technique. However, this algorithm has many weaknesses related to undetectability and robustness features.

Undetectability: It is one of the most important features in steganography. The larger the modification is applied to the stego-image, the more detectable elements are presented. If the payload capacity of the LSB algorithm is greater, the statistical properties of the stego-image become different than the cover image. Randomness occurs when the least significant bits are modified in the cover object. This randomness can be detectable by some statistical analysis techniques [11].

Some stego-images can be defeated with visual analysis of the LSB of stego- image. This is called a visual attack. The idea of visual attacks is to separate some parts of the stego-image and present them in a way that helps a person trying to search for noise [12]. One common test is to display Least-significant bits plane of images since these bits are not presented in random [12]. Figure 3(a) displays a cover image and Figure 3(b) the least significant bits plane of the image. The white is displayed when the pixel's LSB=1 and black when LSB=0. From Figure 3(b) you can notice the least significant bits are not random and represent the content of the cover image. Thus, modification of these bits in some parts of the image will leave visual anomalies or noises. For example changing the bits of the door of the first house on the right (black part) will insert some noises and artifacts. This leads to the detection of the secret information by simply analyzing the least significant bits plane of the image.

The stego-images that have hidden messages in their least significant bits cause distortions detectable by steganalysis. The process of analyzing and using the histogram of the image to detect the existence of hidden information is called histogram attack. Histogram attack is a statistical approach for steganalysis applied on LSB approach [14]. One of the most well-known steganalysis techniques is the RS-analysis. RS analysis is a steganalysis method for detecting stego-images that are based on LSB. It is used to estimate the size of the hidden data. RS makes changes to the least significant bit plane. Then the altered bits with some discrimination function are used to classify some sets of pixels. Those sets are counted and some calculations are performed to estimate the message size [13]. It analyzes the asymmetry imbalance introduced to the image when many LSBs are changed. Asymmetry artifacts occur because at the embedding process the even values are always increased while the odd values are always decreased. This occurs when the secret bit does not match the pixels bit. This effect is introduced into the histogram and makes the hidden information detectable.



(a)

(b)

Figure 3: (a) Cover Image (b) LSB plane of the Cover Image

Robustness: LSB is vulnerable to image processing such as cropping, resizing, scaling, rotating and lossy compression which will destroy the hidden message. For instance, a stego-image is converted to another file format; the resulted format uses lossy compression. In that case the hidden information is destroyed and cannot be reconstructed. All the approaches based on LSB are not robust against some image processing. As mentioned previously robustness is not a crucial feature of steganography.

(e) Improvement of LSB: One alternative algorithm that is proposed to improve the undetectibality of the stego-image quality is the LSB matching (LSBM) algorithm. It does not substitute the least significant bits in the stego-image such as in case of LSB algorithm. The LSBM adds -1 or +1 (±1 schema) randomly to the value of the stego-image when the secret information bit does not match the LSB of the stego-image. For instance, the pixel value 63 with the binary number (00111111) and a secret bit 0. After embedding the secret bit, the algorithm randomly adds 1 and it becomes 64 (01000000). This will eliminate the asymmetry artifacts produced by the LSB algorithm because statistically the probability of increasing and decreasing for each modified pixel is the same [15].

However Harmsen and Perlman [15] have proposed to exploit the center of mass (COM) of the histogram characteristic function (HCF) to detect LSBM. It analyzes the histogram of the stego-image and compares it to its cover image. They disclosed that cover images contain more high-frequency component compared to its stego-image histogram. Subsequently Mielikainen [16] proposed LSB matching revisited algorithm (LSBMR). To overcome that limitation in LSB matching revisited algorithm. Luo et al. [15] presented an edge adaptive image steganography based on LSB matching revisited. The proposed method is studied and analyzed in next section

IV. EDGE ADAPTIVE LSB MATCHING REVISITED (EALSBMR) ALGORITHM

The edge adaptive based on LSBMR is one of recent significant achievement in the domain of image steganography based on LSB in spatial domain. The proposed Edge adaptive method [15] uses PVD-approach to select regions and LSBMR as data hiding algorithm. This section presents description of this EALSBMR method, example, limitation of the method and improvements proposed by some researchers.

a) Description of EALSBMR: Several basic processes are used to embed a secret message using edge adaptive based on LSB matching revisited (EALSBMR). First a preprocessing of the image is conducted such as dividing cover image into number of non-overlapping blocks *Bz* and rotating each by some degree. Another parameter threshold T is initialized for region selection. T determines the units of two consecutive pixels to be selected. Where the units are selected whose absolute difference of the two consecutive pixels are greater than or equal T. T is calculated based on the message size and the content of the cover image. The LSBMR is adopted as data hiding technique of secret bits into the selected units. Then adjustment of the units, whose difference drop less than T, is performed. This adjustment is important for correct extraction of the message bit. Post-processing is conducted to obtain the stego-image by rotating back the *Bz* blocks. Finally, *Bz*, T and length of the secret message are embedded into the stego-image. Huang et al. [15] proposed a schema which is depicted in Figure 4.

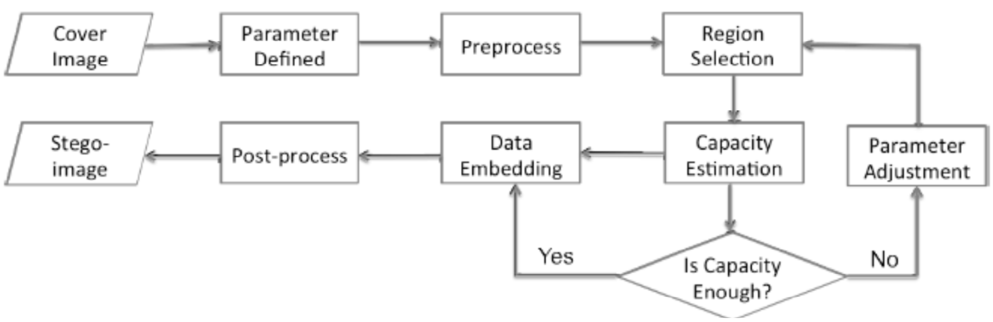


Figure 4: The embedding process of edge adaptive based on LSB matching revisited

The inverse operation is conduced to recover the secret message at the receiver side. By following the succeeding steps. First the parameter *Bz*, T and secret message size are recovered from the stego-image. *Bz* is extracted to do preprocessing of the image. The parameter T is extracted for the regions selection. Units are selected whose absolute difference is greater than or equal T. Finally secret bits are extracted using LSBM revisited algorithm on the selected units.

(b) Example of EALSBMR: To embed a secret message using this method the following stages are applied:

- *Parameter Defining and Preprocessing:* 2 parameters are defined. *Bz* is initialized which is used to divide a cover image into *Bz x Bz* non-overlapping blocks. Then each block is rotated by some degree. The rotation degree is chosen randomly based on a stego-key1 from the following set {0, 90, 180, and 270}. Figure 5 (a) shows a cover image of baboon and the resulting image Figure 5 (b) of dividing it into 12 x 12 non-overlapping blocks and each block is rotated by 180 degree.
- *Region Selection:* The image is then rearranged as a row vector V through raster scanning. Because LSBMR uses unit pair for hiding bits, the V is divided into non-overlapping units of two consecutive pixels (x_i, x_{i+1}). Then the threshold T is estimated to select the region to carry the secret bits. The threshold T depends on the size of secret message (M) and the image content. Then the message bits are embedded in the following set:

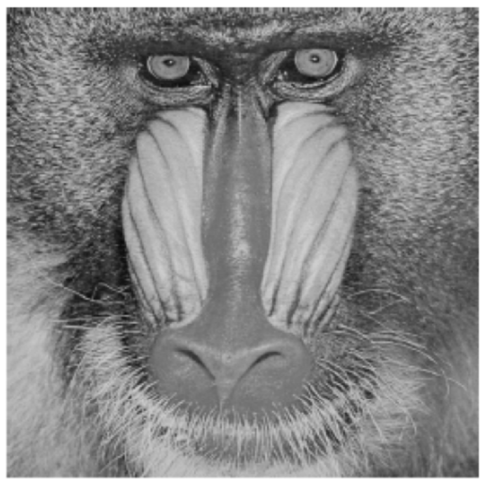
$$EU(t) = \{(x_i, x_{i+1}) \mid |x_i - x_{i+1}| \geq t, \forall (x_i, x_{i+1}) \in V\} \quad (1)$$

Where *EU(t)* is the set of unit pairs that their absolute differences are greater than or equal to parameter t. T can be estimated using the following formula:

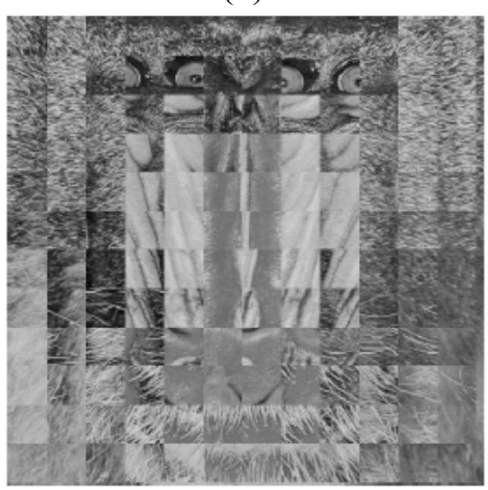
$$T = \arg\max_t \{2 \times |EU(t)| \geq |M|\} \quad (2)$$

Where t – {31, 30,… 2, 1,0}, |EU(t)| represents the total number of pairs in the set EU(t) and |M| is the secret message size. It can be noticed when T=0, the algorithm will achieve the same payload as LSBMR.

In other words, first we check the number of units that their absolute differences are not less than t=31 and save in set EU (t). If the total numbers of pixels in the set is greater than or equal to the total number of secret bits, then T=31; Otherwise t is decremented by 1. We continue in this process until the total number of pixel in the set EU (t) is not less than the message size. This means that all secret bits can be embedded into the selected units. The value of T is adjusted adaptively.



(a)



(b)

Figure 5: (a) Cover Image (b) Image after dividing into 12 x 12 and Rotating by180

- *Data Embedding:* T is calculated and the regions are selected and added to the set *EU(T)*. The secret bits are embedded in those regions in a pseudo random order determined by a stego-key key2. The data-hiding algorithm performed on these regions is the same as LSBMR following the four cases stated in Figure 5.4. However, in some cases after embedding bits into the unit pair (x_i, x_{i+1}) the new difference between them |yi - yi+1| become less than T (|y_i - y_{i+1} | < T). Consequently, the new values should be readjusted to keep their differences greater than or equal to T |y_i - y_{i+1} | e” T. The values are readjusted based on the following formula:

$$
\begin{aligned}
&(v_i^{'}, y_{i+1}^{'}) = \arg\min(e_1, e_2)\{|e_1 - x_i| + |e_2 - x_{i+1}| e_1,\\
&= y_i + 4k_1, e_2 = y_{i+1} + 2k_2, |e_1 - e_2| \geq T, 0 \leq e_1\\
&e_2 \leq 255, 0 \leq T \leq 31, (k_1, k_2) \in Z\}
\end{aligned}
\tag{3}
$$

Figure 6 shows the implemented algorithm for readjusting values based on (3).

Input: new pixel pair values (a , b) , original pixel pair value (ori-a, ori-b), T

Output: readjusted pixel pair a1, b1

```
set_a1[] = { a, a-4 , a+4}
set_b1[] = { b, b-2 , b+2 , b-4 , b+4}

temp_a=a,  temp_b=b, a1=a, b1=a;
min_error = Double.POSITIVE_INFINITY

for i=0 to 2 do
      temp_a = set_a1[i];
      if (temp_a > 255 || temp_a < 0)
      continue;
      for j=0 to 4 do
           temp_b = set_b1[j];

           if (temp_b > 255 || temp_b < 0)
                continue;

           if ((abs(temp_a - temp_b) < T))
                continue;

           new_error = abs(temp_a - ori_a)+abs(temp_b -orig_b);
           if (new_error < min_error)
                     a1 = temp_a; b1= temp_b;
                     min_error= new_error;
           end if

      end for
end for
```

Figure 6: Algorithm for readjustment two consecutive pixels

The algorithm is implemented and the result of hiding information in cover image (Figure 5) is displayed in Figure 7. The hidden information is 1000 byte and the size of the cover image is 512x512. The estimated T = 31. It is clear that the original image and the resulting image looks are the same to human eye. That’s because the selected pixels won’t affect the observation of the human naked eyes whenever changed.

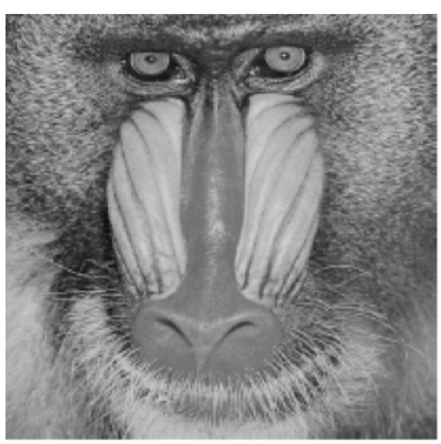


Figure 7: Stego-Image using EALSBMR

(c) Analysis of EALSBMR: *Invisibility:* The most significant property of this technique is that it uses the edges to embed information first and leaves the smooth areas depending on the capacity of the secret message. It has higher invisibility compared to LSB and LSBMR algorithm.

Capacity: In addition the maximum capacity can be achieved is 1bpp in average case when the threshold T=0. In such case the embedding capacity of the Edge-adaptive LSBMR is almost the same as LSBMR except for extra 7 bits.

Security: One of the benefits of this EALSBMR technique is that the security is improved since two stego-keys are required. The stego- key_1 is used to rotate the blocks. The second stego-key_2, which serves as a seed to PRNG, is used to determine the path for embedding the secret bits.

Undetectability: The detectability of existence secret message in the stego-image using RS steganalysis is avoided. The RS steganalysis is ineffective to detect EALSBMR. Since the asymmetry artifacts that introduced to the stego-image is eliminated since ± 1 method is used for embedding information.

(d) Limitation of LSBMR: *Undetectability:* Huang *et al.* [19] stated that the proposed EALSBMR is not sufficient in resisting some steganalysis technique called the DFT spectrum analysis. They conducted an experiment that shows that an alteration is produced to the statistical characteristic of horizontal and vertical histograms in the range from T to 255 and from -255 to T. It can be suggested that the edges determined are not efficient because the proposed algorithm uses a block of 2 pixels to determine the edges. The edges are determined either only in horizontal or in vertical direction. It does not take into account neighboring pixels to detect the edges.

V. CONCLUSION

We have surveyed the two algorithms that are commonly used for image steganography. The application creates a stego image in which the personal data is embedded and is protected with a password which is highly secured. The main intention of the paper is to develop a steganographic application that provides good security.

We are reviewing the Edge Adaptive Least Significant Bit Matching Revisited (LSBMR) algorithm in this paper for developing the application which is faster and reliable and compression ratio is moderate compared to LSB algorithm.

REFERENCES

[1] R. Anderson and F. Petitcolas, "On the limits of steganography" IEEE Journal of Selected Areas in Communications, Vol. 16, No. 4, May 1998.

[2] Niels Provos, Peter Honeyman, "Hide and Seek: An Introduction to Steganography," IEEE computer society, 2003.

[3] K B Raja, Venugopal K R and L M Patnaik, "A Secure Stegonographic Algorithm using LSB, DCT and Image Compression on Raw Images", Technical Re- port, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, December 2004.

[4] T. Morkel , J.H.P. Eloû, M.S. Olivier ,"An overview of image Steganography", Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science University of Pretoria, 0002, Pretoria, South Africa.

[5] Ran-Zan Wang, Chi-Fang Lin, Ja-Chen Lin, "Hiding data in images by optimal moderately signiûcant-bit replacement" IEE Electron. Lett. 36 (25) (2000), 20692070.

[6] Chi-Kwong Chan, L.M. Cheng ,"Hiding data in images by simple LSB substitution", Department of Computer Engineering and Information Technology, City University of Hong Kong, Hong Kong Received 17 May 2002.

[7] Samir K Bandyopadhyay, Debnath Bhattacharyya, Debashis Ganguly, Swarnendu Mukherjee and Poulami Das, "A Tutorial Review on Steganography", Heritage Institute of Technology.

[8] Pratap Chandra Mandal, "Modern Steganographic technique: A Survey", Asst. Prof., Department of Computer Application, B. P. Poddar Institute of Management Technology, International Journal of Computer Science Engineering Technology (IJCSET).

[9] Rodrigues, J. M., Rios, J. R., & Puech, W. (2004). SSB-4 "System of Steganography using bit 4", In 5th International Workshop on Image Analysis for Multimedia Interactive Services.

[10] Mare, S. F.,Vladutiu, M., & Prodan, L. (2011). "Decreasing change impact using smart LSB Pixel mapping and data rearrangement", In The 11th International Conference on Computer and Information Technology, pp. 269-276.

[11] Karaman, H. B., & Sagiroglu, S. (2012), "An Application Based on Steganography",In Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining, pp. 839-843.

[12] Westfeld, A., & Pfitzmann, A. (2000), "Attacks on steganographic systems", In Information Hiding, pp. 61-76. Springer Berlin Heidelberg.

[13] Hempstalk, K. (2006), "Hiding behind corners:Using edges in images for better steganography", In Proceedings of the Computing Women's Congress, Hamilton, New Zealand. [14] Petitcolas, F. A., Anderson, R. J., & Kuhn, M. G. (1999). In Proceedings of the IEEE Information hiding-a survey, 87(7), pp. 1062-1078.

[15] Luo, W., Huang, F., & Huang, J. (2010), "Edge adaptive image steganography based on LSB matching revisited", IEEE Transactions on Information Forensics and Security, 5(2), pp. 201-214.

[16] Mielikainen, J. (2006), "LSB matching revisited", IEEE Signal Processing Letters, 13(5), pp. 285-287.

[17] Anderson, R. J., & Petitcolas, F. A. (1998), "On the limits of steganography", IEEE Journal on Selected Areas in Communications, 16(4), pp. 474-481.

[18] Petitcolas, F. A., Anderson, R. J., & Kuhn, M. G. (1999), In Proceedings of the IEEE Information hiding-a survey, 87(7), pp. 1062-1078.

[19] Huang, W., Zhao, Y., & Ni, R. R. (2011), "Block Based Adaptive Image Steganography Using LSB Matching Revisited". Journal of Electronic Science and Technology, 9(4), pp. 291-296.