ISSN:0975-9646

LZW based Image Steganography using Kekre's Algorithm

Isha Kajal[1], Harish Rohil[2], Abhishek Kajal[3]

[1]*Research Scholar, Department of Computer Science & Applications, CDLU, Sirsa, Haryana*
[2]*Assistant Professor, Department of Computer Science & Applications, CDLU, Sirsa, Haryana*
[3]*Assistant Professor, Department of Computer Science & Engineering, GJUS&T, Hisar, Haryana*

***Abstract* - Steganography is a method in which secret message is embedded into a cover image to protect it from unauthorized access. The challenge of steganography technique is to maintain a rational balance between the quality of file and size of data that can be transferred. This paper presents a secured and robust steganography method capable of embedding high volume of information in digital cover image without incurring and perceptual distortion. This method is based compression and encryption. In this method the message to be transmitted first using Lempel-Ziv-Welch compression technique and is encrypted by using and hide compress data in image using kekre's algorithm. The proposed method is tested with different images and text of various lengths.**

Keywords- Steganography, LZW Compression, LSB Embedding Technique, PSNR, MSE

1. INTRODUCTION

In the world of Internet data protection and security of the information have become critical issue. The techniques available to achieve the goal of confidential information security are cryptography, encryption and steganography. Cryptography scrambles the message so that it cannot be understand while steganography hides the existence of the message by cleverly embedding the message into a cover image. Encryption and Steganography attains the same goal via different methods. The word steganography comes from the Greek word steganos which means covered or secret and the graphy means writing or drawing [2]. Steganography don't alter the message but hides inside a cover object. Combining encryption with steganography allows secure confident information communication hiding the secret message in the least significant bits of pixels of cover image is one of the method of steganography technique using LSB technique the image quality is not deteriorate, The image quality of stego image achieved by applying the LSB technique is very closer to the original one.

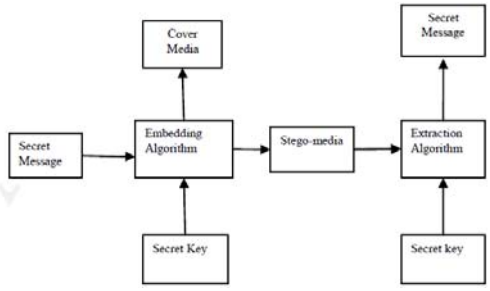


Fig 1. Steganography Process

There have been a large embedding techniques proposed number of steganography in the literature. These techniques modified the cover image with different approaches, Image steganography technique can be divided into two groups:

a) Image domain also called spatial domain
b) Transform domain also called frequency domain

LSB Technique

The most basic and important image Steganographic Technique is Least Significant Bit [1] embedding technique. In this technique data can be hidden in the least significant bits of the cover image and the human eye would be unable to notice the hidden image in the cover file. This technique can be used for hiding images in 24-bit, 8-bit or gray scale format. In this technique, least significant bit of each pixel is replaced with secret message bit until message end. When using a 24 bit image one can store 3 bit in each pixel by changing a bit of each if the red, green and blue color components. An 800 x 600 pixel image can store 1,4400,00 bits or 180,000 bytes of embedded data. For example a 24 bit can be as follows:

(10110101 01101100 10101101)
(10110110 11001101 00111110)
(10110101 01100011 10001110)

The number 150 which binary representation is 10010110 is embedded into the least significant bits of this part of the image, the resulting grid as follows:

(1011010**1** 0110110**0** 1010110**0**)
(1011011**1** 1100110**0** 0011111**1**)
(1011010**1** 0110000**1** 10001110)

Although the number is embedded into the first 8 bytes of the grid, only the 3 underlined bits need to be changed according to the embedded message. On an average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. There are 256 possible intensities of each primary color, so, changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye, thus the message is successfully hidden. If the message is hidden even in the second to least significant as well as in least significant bit then too no difference is seen in the image. In LSB Technique, consecutive bytes of the image data from the first byte to the end of the message are used to embed the information. But this approach is very easy to detect. A more secure system can be in which the sender and receiver share a secret key that specifies only certain pixels to be changed. Even if the intruder suspects that LSB steganography has been used, there is no way of knowing

which pixels to target without the secret key. In its simplest form, LSB makes use of BMP images, since they use lossless compression. To hide a secret message inside a BMP file, one would require a very large cover image. For this reason, LSB method has also been developed for use with other image file formats. This type of information hiding algorithm could be a major risk because eavesdropper can apply sequential scanning based techniques [3] to recover the secret message

2 MODIFIED KEKRE'S ALGORITHM (MKA)

Modified Kekre's Algorithm (MKA) [5] is based on Least Significant Bit (LSB) method. MKA can be applied on 8 bit gray scale images or 24 bit Read Green Blue (RGB) color image. It uses up to five LSB's of a pixel to embed the data. The number of secret data bits that can be embedded in the pixels depends upon the pixel intensity of the pixels of the cover image. To achieve more security MKA uses 8 bit secret key to perform XOR operation to all the bytes of the secret message. While extracting the message XOR operation is also performed using the same key. The embedding algorithm maintains a matrix of pixels where up to 5 bits of message are used to embed, and this matrix is required while extracting the secret hidden message from stego-image. In Table-2 'x' shows don't care bit whose value can be either '0' or '1'. "Pixel intensity" shows the value of pixel. "Data Bit to Embed" shows current message bit used to embed into the cover-image.

Suppose pixel intensity is 245 which exist in the row number 1 and 2 of the Table-2. For embedding the secret data bits its 5 or 4 LSBs can be utilized, depending on current data bit to embed. If current data bit is 0 then 4 LSBs otherwise 5 LSBs are used for embedding data bits. Mark a 1 bit entry into maintained matrix pixel position to identify that if this pixel contains 5 bits of data. Same procedure is applicable for other pixels of the image according to Table 1

TABLE I

S. No	Pixel Intensity	Data bit to Embed	Matrix Entry	Utilize Bit/ bits
1	240-255	1	1	5
2	240-255	0	-	4
3	224-239	0	1	5
4	224-239	1	-	3
5	192-223	X	-	2
6	0-191	X	-	1

Where X: Don't care bit

Same procedure will be run to extract the data bits of message using maintained matrix because it keeps the track of the pixel position where 5 LSBs are utilized. At the end, 8 bit secret key with XOR operations applied on the extracted message to regenerate original message which was embedded.

Hussain [6] discusses a method that is an improvement of MKA [5]. It also applies 8 bit secret key with XOR operation on all bytes of message to change the originality of message. It maintains a matrix for those pixels which will embed 5, 3 and 2 LSBs of data. In Table-2 "Pixel intensity" shows the value of pixel. "Data Bit to Embed" shows current message bit used to embed into the cover-image. "Matrix Entry" maintains a matrix which denotes the 5 LSB are embedded. "Utilize Bits" shows the total number of bits embedded into a pixel.

If pixel intensity is 33 which exist in row number 7 and 8 of Table-2. For data embedding, 2 or 1 bits of pixel can be utilized depending on current data bit to embed. If current message (want to embed data) bit is 0 then 2 bits otherwise 1 LSB are used for embedding data bits. If this pixel contains 2 bits of data, mark a 1 bit entry into maintained matrix pixel position for identification of extra bits. Same procedure is applicable for other pixels of the image according to Table-2.

TABLE 2

S. No.	Pixel Intensity	Data Bit to Embed	Matrix Entry	Utilize Bit/Bits
1	240-255	1	1	5
2	240-255	0	-	4
3	224-239	0	1	5
4	224-239	1	-	3
5	192-223	0	1	3
6	192-223	1	-	2
7	32-191	0	1	2
8	32-191	1	-	1
9	16-31	0	1	3
10	16-31	1	-	2
11	0-15	0	1	5
12	0-15	1	-	4

3. IMAGE QUALITY METRICS

The image quality metrics are figures of merit used for the evaluation purpose of the image quality. These metrics provide some measures of the closeness between two digital images by exploiting the differences in the statistical distribution of pixel values. The most commonly used quality metrics are:

- Mean Square Error (MSE)
- Root Mean Square Error (RMSE)
- Structural Similarity (SSIM)
- Peak Signal to Noise Ratio (PSNR)

3.1. Mean Square Error (MSE)

The mean square error is defined as the square of the difference between the pixel values of the original

image and the stego image and then dividing it by size of the image. The mathematical formula for computing mean square error between x and y images of sizes M*N is given below

$$MSE = \frac{1}{M*N}\sum_{x=1}^{M}\sum_{y=1}^{N}[x(m,n) - y(m,n)]^2$$

The lower value of Mean Square Error (MSE) signifies lesser error in the stego image in other words better quality.

3.2. Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) is calculated by getting the square root of the mean square error (MSE). The RMSE can be calculated as follows.

$$RMSE = \sqrt{MSE}$$

3.3. Structural Similarity (SSIM)

The SSIM metric was given by Wang el. This method is used to measure the similarity between two images [3]. It is designed by modeling any image distortion as a combination of three factors that are loss of correlation, luminance distortion and contrast distortion. Mathematically, the SSIM is calculated as follows:

$$SSIM(\boldsymbol{x},\boldsymbol{y}) = l(\boldsymbol{x},\boldsymbol{y})\cdot c(\boldsymbol{x},\boldsymbol{y})\cdot s(\boldsymbol{x},\boldsymbol{y})$$

where

$$l(\mathbf{x},\mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

3.4. Peak Signal to Noise Ratio (PSNR)

The Peak Signal to Noise Ratio (PSNR) measures the estimates of the quality of stego image compared with an original image and is a very commonly used metric way to measure image reliability or conformity. The mathematical formula to calculate the PSNR value is as follows:

PSNR = 20log10 [MAXPIX / MSE]

Where MAXPIX is the maximum pixel value and MSE is the Mean Square Error.

In PSNR, 'signal' is the original image and 'noise' is the error in the stego image resulting due to encoding and decoding. PSNR is a number that reflects the quality of the stego image and is measured in decibel (dB). Mathematically, PSNR is inversely proportional to the MSE, which implies the lower the value of MSE higher is its PSNR. Thus higher the Peak Signal to Noise Ratio (PSNR) is better.

4. RELATED WORK

Steganography is an area of invisible communication. It is not a modern technique which is used for protecting the unauthorized access of the information but is an ancient technique which is in existence since 440 B.C. The most basic and important image Steganographic Technique is Least Significant Bit [2, 7] embedding technique. In this technique data can be hidden in the least significant bits of the cover image and the human eye would be unable to notice the hidden image in the cover file. This technique can be used for hiding images in 24-bit, 8-bit or gray scale format. In this technique, least significant bit of each pixel is replaced with secret message bit until message end.But this technique has less embedding capacity and easy to detect. Hamid et al. in [8] discussed a texture based image steganography technique. This technique divides the texture areas into two groups. One is simple texture area and other is complex texture area. In Simple texture area 3 LSB bits of Red channel, 3 LSB bits of Green channel and 2 LSB bits of blue channels are used for embedding the secret data. In Complex texture area data is embedded into the 4 LSB bits of the pixel. This method increases the embedding capacity of the covered image. Marvel [9] discusses spread spectrum image steganography technique. In spread spectrum techniques, hidden data is spread throughout the cover-image making it harder to detect. In spread spectrum image steganography the secret message is embedded in noise and then combined with the cover image which results into the stego image. Since the power of the embedded signal is much lower than the power of the cover image, the embedded image is not perceptible to the human eye or by computer analysis without access to the original image. This technique provides high security than the previous techniques. However, this method does not provide sufficient data payload. *Cheddad et al.*'s[10] have proposed a region of interest (ROI) in image based adaptive steganography method. It selects required ROI in the image where it carefully hides the data bits. The selection of these regions is based on human skin tone color detection. Generally adaptive steganography methods are hard to target for attacks especially when the hidden message capacity is too small. [11] *Yang et al.* proposed an adaptive LSB substitution based data hiding method for image. To achieve better visual quality of stego-image it takes care of noise sensitive area for embedding. Proposed method differentiates and takes advantage of normal texture and edges area for embedding. This method analyzes the edges, brightness and texture masking of the cover image to calculate the number of k-bit LSB for secret data embedding. The value of k is high at non-sensitive image region and over sensitive image area k value remain small to balance overall visual quality of image.

5. PROPOSED WORK

In the techniques discussed above, if the data embedded in the image is increased, the image quality deteriorates. So, we cannot embed sufficiently large data into the cover image.

In our proposed technique we overcome this problem. We preprocess the secret data before embedding it into the image. The pre-processing reduces the size of the data by a significantly large amount which permits embedding the large amount of data into the same size

cover image. Our proposed technique is based upon the intensity values of the pixels in the cover image. This technique can be applied to grey scale as well as color images. The preprocessed data is embedded into the cover image based upon the intensity values of the pixels in the cover image. For pre-processing, LZW(Lempel–Ziv–Welch) data compression technique is used. This method generates the stego image which is of very good quality. In this technique sequence of 8-bit secret data is encoded as fixed-length 12-bit codes. The code from value 0 to 255 represents one character sequences consisting of the corresponding 8-bit character. As the data is encoded, the codes with values 256 through 4095 are created in a dictionary depending upon the sequences encountered in the data. A dictionary is initialized to contain the single-character strings corresponding to all the possible input characters. At every step in the compression process, input characters are gathered into a sequence until the next character comes that will make a sequence for which there is no code in the dictionary. The code for the sequence without the character encountered is emitted, and a new code for the sequence with the character encountered, is added in the dictionary. The algorithm works by scanning the input secret data for successively longer substrings until a string is found that is not in the dictionary. When such a string is found, the index for the string without the last character is fetched from the dictionary and sent to output, and the new string including the last encountered character is added to the dictionary with the next available code. The last input character is then used as the next starting point to scan for substrings. In this way, successively longer strings are added in the dictionary and made available for subsequent encoding as single output values.

In the decoding algorithm value is read from the encoded input and corresponding string is outputted from the initialized dictionary. The next value is read from the input secret data string, and adds to the dictionary the concatenation of the string just output and the first character of the string obtained by decoding the next input value. The decoder then proceeds to the next input value and repeats the process until there is no more input. The final input value is decoded without any more additions to the dictionary. The decoder builds up a dictionary which is similar to the dictionary used by the encoder. And this dictionary is used to decode subsequent input values. Full dictionary is not required to send to the receiver. The initial dictionary containing the single-character strings needs to be sent only. If the dictionary's initial values are decided beforehand by the sender and receiver, the initial dictionary too needn't be sent. This compression technique gives best results on the secret data with repeated patterns.

6. EXPERIMENTAL RESULT

In this section, we discuss the experimental results of our proposed steganography method and compare with that of the MKA. We take various color images for hiding the secret data. The image quality metrics used are Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Root Mean Square Error (RMSE). High PSNR value and low MSE value signifies good quality image. PSNR is measured in decibel (db). The images taken in our experiments include Barbara, Baboon, Football, Lena and Pepper each of different dimensions. The reason for making comparison with these methods is that they are more recently developed and have good performance. The secret data taken in our experiments is Abraham Lincoln's letter to his son's teacher that is embedded into each of these images which is of size 1785 bytes. The resultant stego images with hidden secret message, employing our proposed method are shown in Figures below.

The performance results are shown in Table 3, 4, 5 and 6.

Table 3 PSNR values of Different Approaches on different Images

Cover Image	Modified Kekre Algorithm	Proposed Algorithm
Barbara.jpg	60.9421	63.7030
Baboon.jpg	73.3934	76.0391
Football.jpg	68.2248	70.8976
Peppers.png	72.0161	74.8801
Lena.bmp	73.2938	76.0108

Table 4 MSE values of Different Approaches on different Images

Cover Image	Modified Kekre Algorithm	Proposed Algorithm
Barbara.jpg	0.0523	0.0277
Baboon.jpg	0.0030	0.0016
Football.jpg	0.0098	0.0053
Peppers.png	0.0027	0.0013
Lena.bmp	0.0030	0.0016

Table 5 RMSE values of Different Approaches on different Images

Cover Image	Modified Kekre Algorithm	Proposed Algorithm
Barbara.jpg	0.2288	0.1665
Baboon.jpg	0.0546	0.0402
Football.jpg	0.0989	0.0727
Peppers.png	0.0529	0.0402
Lena.bmp	0.0552	0.0404

Table 6 CAP (Maximum Embedding Capacity) values in bytes of Different Approaches on different Images.

Cover Image	Modified Kekre Algorithm	Proposed Algorithm
Barbara.jpg	7371	13104
Baboon.jpg	117631	208550
Football.jpg	39780	70720
Peppers.png	114718	204355
Lena.bmp	127402	226026

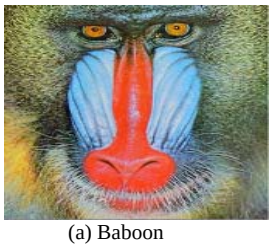
It is evident from the above tables that the proposed technique is better than the existing technique and produces better results. For every image the value of PSNR, MSE and CAP i.e. maximum embedding capacity of our proposed technique is more than the MKA technique. The Capacity of all the cover images to

embed the secret data increases by applying the proposed technique. The security of the secret data also increases due to its pre-processing.


(a) Barbara


(b) Barbara

Figure 2 Barbara (a) Cover image and (b) Stego image


(a) Baboon

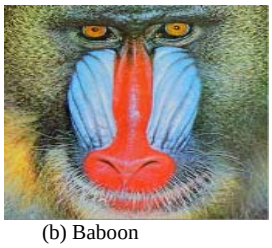
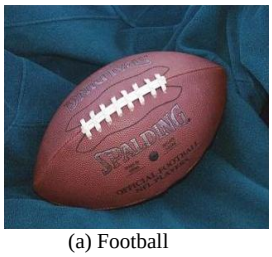

(b) Baboon

Figure 3 Baboon (a) Cover image and (b) Stego image


(a) Football

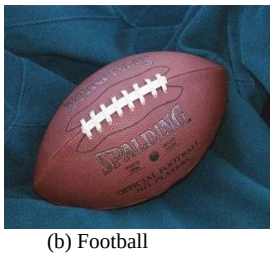

(b) Football

Figure 4 Football (a) Cover image and (b) Stego image


(a) Pepper


(b) Pepper

Figure 5 Pepper (a) Cover image and (b) Stego image


(a) Lena


(b) Lena

Figure 6 Lena (a) Cover image and (b) Stego image

7. CONCLUSION & FUTURE SCOPE OF WORK

In this work we explored the existing image steganography techniques. We proposed an efficient image steganography technique. In image steganography, image is used as a carrier for transmission of the secret information or data. The image used can be either gray scale or color image. In this technique data is firstly preprocess. This preprocessing reduces the size of the data by a significantly great amount. This preprocessed data is then embedded into the LSBs of the pixels of the image depending upon the intensity of the pixel values. Our proposed algorithm is targeted to achieve very high image embedding capacity into the cover image and more security of the secret data. The proposed technique performs better than MKA [5]. It has high PSNR value and low MSE value as compared to MKA. This preprocessing reduces the size of the secret data by a significant amount and thus permits more data into the same image. The embedding capacity of the proposed technique is very high as compared to MKA. The work can be extended to provide an alternative strategy for data compression which can further reduce the size of the data. Efficient cryptographic techniques can also be used along with the steganographic techniques to provide more security to the data.

Data security and high embedding capacity is there due to the pre-processing of the data before embedding into the cover image. This method does not require the original image while extracting the secret data from stego image.

The work can be extended to provide an alternative strategy for data compression which can further reduce the size of the data. Efficient cryptographic techniques can also be used along with the steganographic techniques to provide more security to the data.

REFERENCES

[1] Deshpande Neeta, Kamalapur Snehal, *"Implementation of LSB Steganography and Its Evaluation for Various Bits*" K.K.Wagh Institute of Engineering Education & Research, Nashik India

[2] Moerland, T., "*Steganography and Steganalysis*", Leiden Institute of Advanced Computing Science, www.liacs.nl/home/ tmoerl/privtech.pdf

[3] Johnson, N.F. & Jajodia, S. (1998), "*Exploring Steganography: Seeing the Unseen*", Computer Journal

[4] N. Tiwari and M. Shandilya, "*Secure RGB Image Steganography from Pixel Indicator to Triple Algorithm-An Incremental Growth"*, International Journal of Security and Its Applications Vol. 4(4), Oct. 2010.

[5] H. B. Kekre, A. Athawale, P. N. Halarnkar, "*Performance Evaluation of Pixel Value Differencing and Kekre's Modified Algorithm for Information Hiding in Images*", International Conference on Advances in Computing, Communication and Control, 2009 pp 342-346

[6] M. Hussain, M. Hussain., "*Pixel Intensity Based High Capacity Data Embedding Method*", Information and Emerging Technologies, International conference 978-1-4244-8003 June 2010

[7] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang, Member, IEEE, and Hung-Min Sun, "*Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems*", IEEE Transactions on Information Forensics and Security, vol. 3, no. pp. 488-497. 3rd September 2008.

[8] Hamid, A. M., M. L. M. Kiah, et al. (2009). "*Novel Approach for High Secure and High Rate Data Hidden in the Image Using Image Texture Analysis*" Internationa Journal of Engineering and Technology (IJET).

[9] L.M. Marvel "*Spread Spectrum Image Steganography*," IEEE transactions on image processing, vol. 8, no. 8, pp. 1075-1083, August 1999.

[10] A. Cheddad, J. Condell, K. Curran and P. McKevitt, "*Enhancing Steganography in digital images*", IEEE - 2008 Canadian conference on computer and Robot vision, pp.326-332, 2008.

[11] H. Yang, X. Sun, G. Sun. "*A High-Capacity Image Data Hiding Scheme Using Adaptive LSB Substitution*". Journal: Radio engineering Year: vol. 18, 4 Pages/record No.: 509-516, 2009.

[12] H. B. Kekre, Archana Athawale, Pallavi N. Halarnkar, "*Performance Evaluation of Pixel Value Differencing and Kekre's Modified Algorithm for Information Hiding in 88 Images*", International Conference on Advances in Computing, Communication and Control, pp 342-346, 2009.