Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Data Hiding In Encrypted Compressed Image Using Improved LSB Technique

Y. M. Kamble[1], K. B. Manwade[2]
[1]P.G.Student, Department of CSE, D.Y.Patil College of Engg., Kolhapur, India
[2]Asso. Professor, Department of CSE, Ashokrao Mane College of Engg., Vathar tarf Vadgaon, India

Abstract*— Steganography is a technique where, the secret message is embedded into an image called cover image, and then sent to the receiver. A data hiding scheme by improved LSB substitution is proposed. In simple LSB technique only single LSB bit is used from cover image to hide data bit. But in improved LSB technique, in proposed system, three bits from a color pixel used to hide data bits. Before embedding data bits into color image first encrypt the image and then compress encrypted image for additional space. At the receiver end when content owner receives this stego image he can extract the hidden data bits and recover the image. The substitution improves the security and quantity of secret data bits and also significantly improves the quality of recovered image. Experimental results show that the quality of recovered image significantly improved and cannot is visually indistinguishable from the original cover-image.

***Keywords*— Stego Image, Encryption Compression, Cover Image, Improved LSB**

I. INTRODUCTION

Data hiding is used in many cases for secure communication. One of the forms of data hiding is steganography which embeds data into digital media. Several constraints may need to take into the consideration for embedding secrete data e.g.: the quantity of data to be hidden, what percentage of distortion may occur in cover image due to embedded data, data embedding in image with a minimum amount of data bits loss in cover image, e.g., lossy compression, embedded data should be invisible to a human observer. Depending on the type of application, the quantity of embedded data and degree of distortion may vary. Hence different data hiding techniques are used for different applications. An application that requires a minimal amount of secrete data to be embed is placed on LSB bits of cover image [1]. The use of LSB bits of cover image for secrete data embedding is the most simple method for data hiding. LSB substitution has some drawbacks along with its simplicity. For example any addition of noise or lossy compression may occur in the cover image. There is possibility that some well known attacks may simply set the LSB bits of each pixel to one, which will fully defeats the embedded bit of secrete data in cover image. The distortion of secrete data by setting it one; will set very negligible impact on the cover image. Furthermore, once the algorithm is discovered, the embedded secrete data could be easily modified by an opponent. Security of the hidden data would be significantly improved, when the encrypted hidden data would no longer be easily viewed by the opponent. In the literature, many techniques about data hiding have been proposed [2-6]. One of the common techniques is based on manipulating the least-significant-bit (LSB). Where secrete data bits will be directly replaced in to the LSBs of the cover-image. In this paper an improvement on basic LSB substitution has been proposed by using improved LSB bits. In the color image for single pixel three bits are used to embed the data bits from secrete message. For secure communication encrypted and compressed color image is used as a sample cover image. This encrypted compressed cover image referred as stego image. In this stego image every color pixel has 24 bits. From these 24 bits three bits are used to embed the secrete data. As each color pixel from stego image containing combination of RGB, three bits are used as one from red one from green and one from blue. The secrete message will be hidden into cover image in encrypted format such that an unintended observer will not be aware of the existence of the hidden messages. The rest of the paper is organized as follows. Section 2 briefly describes the proposed scheme. Experimental results are given in Section 3. Finally, Section 4 concludes this paper

II. PROPOSED SCHEME

In this section, the general operation of data hiding by improved LSB substitution method is described.
Let C be the original 24-bit depth color cover-image of Mc × Nc pixels represented as,

$$C= \{Xij \mid 0 \le i < Mc,\ 0 \le j < Nc, Xij \in \{0,1,\ldots 255\}\{0,1,\ldots 255\}\{0,1,\ldots 255\}\} \quad (1)$$

M be the n-bit secret message represented as,

$$M= \{mi \mid 0 \le i < n,\ mi \in \{0,1\}\} \quad (2)$$

2.1 Image encryption:

Image encrypted by using blowfish, symmetric, block cipher technique. In this system first the given image is read out from source. Then for symmetric encryption secrete key is given as an input to blowfish algorithm along with input mage.

Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. This algorithm takes input as block size of 64 bit from input image and the key of length from 32 bit to 448 bits. Then it divides the input block into two halves of size 32 bit each. LE, RE. The pseudo code for algorithm is as follows,

for i=1 to 16 do

$$RE_i = LE_i \oplus P_i;$$

$$LE_i = F[RE_i] \oplus RE_{i-1};$$

$$LE_{17} = RE16 \oplus P_{18};$$

$$RE_{17} = LE16 \oplus P_{17};$$

Here P is array of subkeys derived from input key to the algorithm. Complex function F is used for increasing complexity of the algorithm. After the sixteenth round, swap LE and RE again to undo the last swap.

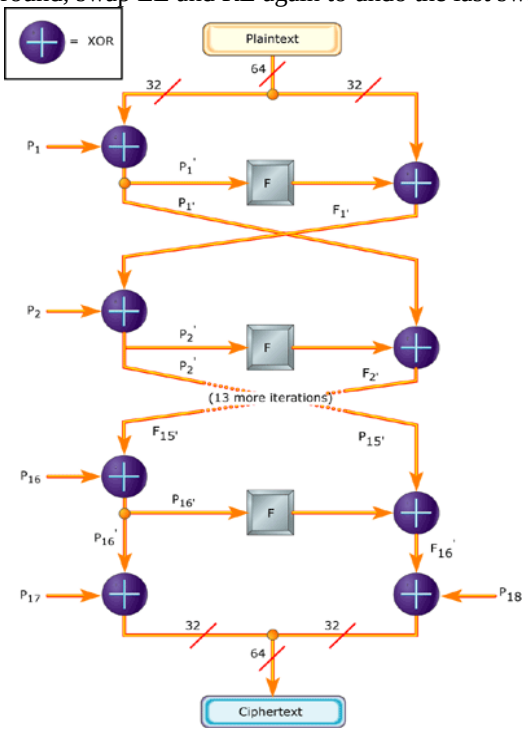

Fig.01 Feistel Structure of Blowfish

This function F contains four S-Boxes containing 256 entries in each. The 32 bit of data halve is divided into 4 parts as 8 bit each and given as input to the one S-Box as input. Each S-box gets an input of 8 bit and results 32 bit output. Again the output from all four S-boxes will be XORed with each other and finally 32 bit output will be generated as shown in following figure.

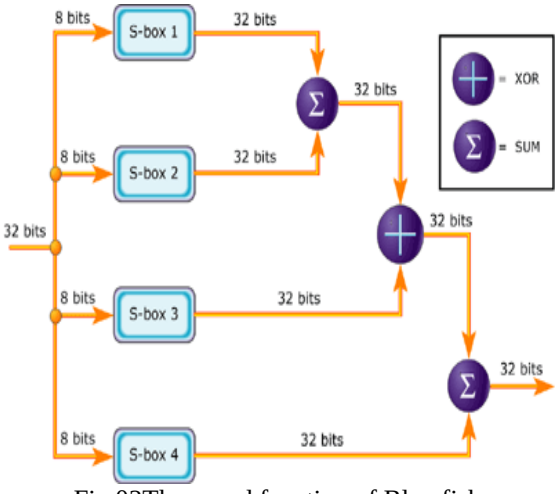

Fig.02The round function of Blowfish

2.2 Compression of encrypted image

As the channel provider for secure communication my compress the data due to limited bandwidth of communication media. So the important factor for image storage or transmission over any communication media is the image compression. Compression makes file sizes manageable, storable and transmittable dimensions. So in the second phase the given encrypted image get compressed by using lossy compression technique. Hence in the proposed scheme VQ (Vector Quantization) [7, 8]-LBG lossy compression algorithm is used to compress given encrypted image.

In the image encoding procedure of VQ for compression, the image is partitioned into a set of non-overlapped image blocks of n x n pixels. Each image block of n x n pixels can be ordered to form a k-dimensional image blocks, where k = n x n. The closest codeword in the codebook for each image vector is to be determined. The compressed codes of VQ are the indices of the closest codewords in the codebook for all the image blocks [7]. Vector Quantization (VQ) methods allow arbitrary compression of digital images.

In this way VQ-LBG algorithm forms a single block by collecting adjacent pixels, which is mapped into a finite set of codewords. In decoding stage the codewords are replaced by corresponding vectors. The set of codewords and the associated vectors together is called a codebook The LBG algorithm [7] works as follows,

Linde-Buzo-Gray (LBG)[7] algorithm used a mapping function to partition training vectors into N clusters.

The mapping function is defined as:

$R^k \rightarrow CB$

Let $X = (x_1, x_2 \ldots x_k)$ be a training vector and d(X; Y) be the EuclideanDistance between any two vectors.

The iteration of LBG for a codebook generation is given as follows:

Step 1: Randomly generate an initial codebook CB_0.

Step 2: i = 0.

Step 3: Perform the following process for each training vector.

Compute the Euclidean distances between the training vector and the codewords in CBi. The Euclidean distance is defined as

$d(X; C) = (\sqrt{\sum}^k t=1(x_t - c_t)^2)$…….. (1)

Search the nearest codeword among CB_i.

Step 4: Partition the codebook into N cells.

Step 5: Compute the centroid of each cell to obtain the new codebook CB_{i+1}.

Step 6: Compute the average distortion for CB_{i+1}.

If it is changed by a small enough amount since the last iteration, the codebook may converge and the procedure stops. Otherwise, i = i + 1 and go to Step 3.

Codebook consists of a set of vectors, called codevectors. A vector quantization algorithm uses this codebook to map an input vector to the codevector closest to it. Image compression, the goal of vector quantization, can then be achieved by transmitting or storing only the index of the vector.

2.3 Data Hiding using Improved LSB

To hide secrete data into encrypted compressed image content owner will provide a data hiding key and secrete data as input. When secrete data will be placed on LSB bits it will be XORed with the data hiding key. Even when opponent will attempt to recover the hidden data from these LSB bits he cannot succeed until having data hiding key. Secrete data will be embedded at LSB of pixel [2]. In the proposed system we are using improved LSB bits where it embeds three bits of secrete data in one pixel. The structure of embedded secrete data in to encrypted compressed image is as shown in following figure 1.

R	1bit	G	1bit	B	1bit

Fig.03 One RGB pixel to store three secrete data bits. One bit per each rgb pixel.

Thus three rgb pixel will be used to embed one character from secrete message. In this way three rgb pixels accommodate the 8(3+3+2) bits from a secrete data and next secrete data bits will be placed in next three rgb pixel. Therefore we can embed approximately 5000 bytes in a image of size 128*128 pixels.An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

III. EXPERIMENTAL RESULTS

For secrete data extraction user provide data hiding key. When proposed system receives data hiding and image containing embedded secrete data it results the hidden data. As if user has only data hiding key only data can be extracted, if he has encryption key he can recover image and if user has both the keys he can extract the data as well as recover the image successfully. In this way we can embed secrete data into image separable.

When content owner extracts hidden data without loss to measure the distortion in directly decrypted image we have used PSNR as quality metrics.

Typical values for the PSNR in lossy image are between 30 and 50 dB. PSNR (Peak signal to noise ratio) can be calculated using following standard formula

$$PSNR=10 \log_{10} ((255*255*255)/MSE$$

Where MSE is mean square error calculated using the square of difference between original image and recovered image divided by image size & by three.

Following table shows the PSNR values of directly decrypted image and recovered images. The PSNR values are calculated for different vector size as shown in table and which is approximately 37dB.Following table no.01 shows the PSNR values of directly decrypted image and recovered image for some standard pixel size as 128*128, 256*256, and 512*512. These standard pixel sizes of images gives the following results as mentioned in table no.01.

Table.1 PSNR values in dB for directly decrypted image and recovered image for standard size images

Image	Decrypted/ Recovered	Vector Size of Compression Algorithm			
		64 bit	128 bit	256 bit	512 bit
Sample 128*128	Decrypted	36.784	37.0046	37.0046	37.004
	Recovered	37.086	37.1065	37.1356	37.118
Sample 256*256	Decrypted	36.618	36.6182	38.9472	38.947
	Recovered	36.869	38.8971	36.8675	36.911
Sample 512*512	Decrypted	46.571	46.5716	46.5716	46.571
	Recovered	36.697	36.7105	36.6758	36.679

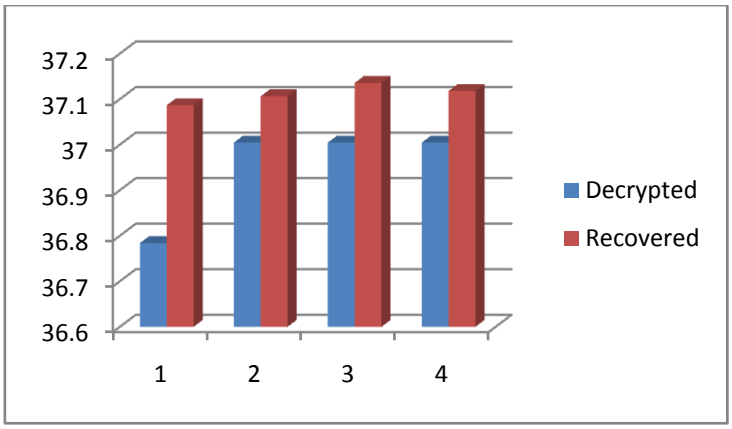

Fig04. PSNR values in dB for directly decrypted and recovered image of some standard size pixels. e.g. 128*128

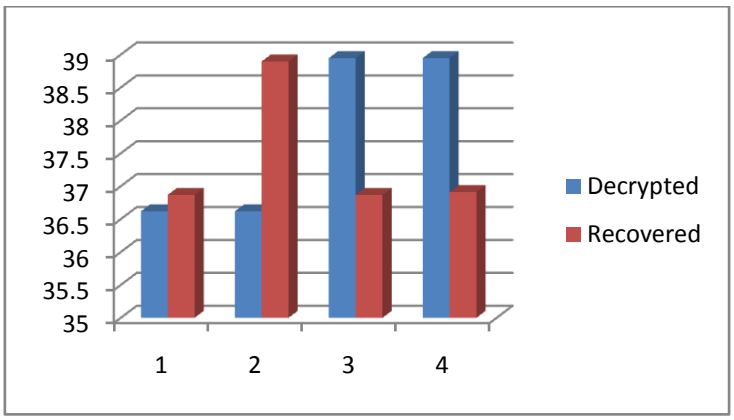

Fig05. PSNR values in dB for directly decrypted and recovered image of some standard size pixels. e.g. 256*256

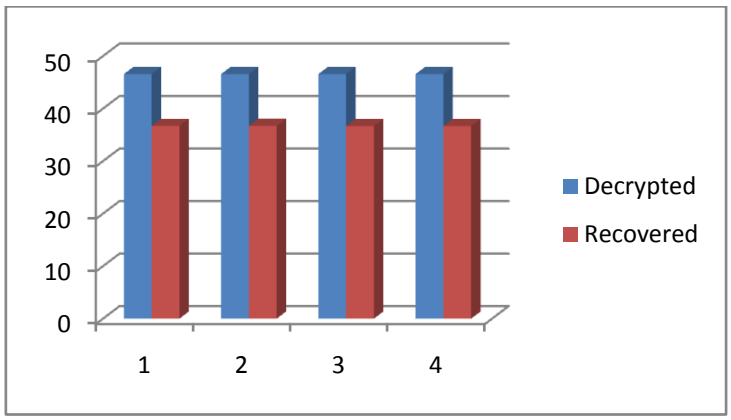

Fig.06 PSNR values in dB for directly decrypted and recovered image of some standard size pixels. e.g. 512*512

IV. CONCLUSION

In the proposed system we can achieve an improved secrete data hiding and extraction technique from encrypted image. With an encrypted image containing secrete data the receiver may extract the secrete data using only the data hiding key, or obtain an original image using only the encryption key. When receiver has both of the keys he can extract the secrete data and recover original content without any error. The standard PSNR values of recovered image are between 30dB to 50dB, and in the proposed system it is achieved approximately 37dB to39 dB, as shown in the experimental results. Also the data embedding rate in cover image, as data embedded rate is 3 bit per pixel, has almost 5000 bytes in an image of 128*128 pixel sizes.

REFERENCES

[1] Chi-Kwong Chan∗, L.M. Cheng "Hiding data in images by simple LSB substitution" Pattern Recognition 37 (2004) 469– 474, ElsevierComputerScience, Aug2003.

[2] A.Z. Tirkel, R.G. Van Schyndel, C.F. Osborne, A digital watermark, Proceedings of ICIP 1994, Austin Convention Center, Austin,Texas, Vol. II, 1994, pp. 86– 90.

[3] W. Bender, N. Morimoto, A. Lu, Techniques for data hiding, IBM Syst. J. 35 (3/4) (1996) 313–336.

[4] T.S. Chen, C.C. Chang, M.S. Hwang, A virtual image cryptosystem based upon vector quantization, IEEE Trans. Image Process. 7 (10) (1998) 1485–1488.

[5] L.M. Marvel, C.G. Boncelet, C.T. Retter, Spread spectrum image steganography, IEEE Trans. Image Process. 8 (8) (1999) 1075– 1083.

[6] K.L. Chung, C.H. Shen, L.C. Chang, A novel SVD- and VQ-based image hiding scheme, Pattern Recognition Lett. 22 (9) (2001) 1051–1058.

[7] A survey of VQ codebook generation by Tzu-Chuen-Lu Ching-Yun Chang.Journal of Information Hiding and Multimedia Signal Processing @2010 ISSN 2073-4212, Ubiquitous InternationalVolume 1, Number 3, July 2010

[8] Yu-chen Hu,Bing-Hwang Su,Chih-Chiang Tsou. "Fast VQ codebook search algorithm for grayscalimagecoding",ScienceDirect,Image and vision computing 26(2008) 657-666.

