Overview of Image Steganography

[1]Anees B. Kazi, [2]Trupti Kumbhar, [3]Kulkarni Kapil P, [4]Deshmukh Gajanan, [5]Deore Sneha R,

B. Tech Student, Associate professor
Email : anees.kazi7@rediffmail.com,nalbalwar_sanjayn@yahoo.co.in,truptikumbhar7@gmail.com

Abstract—Steganography is the technique for information hiding. Secret data is embedded into cover object. Steganography is mainly of four types viz, Text, digital audio, video and image. In this paper we will concentrate more on image Steganography. Counterpart is stegnalysis i.e. detection of presence of message in the cover. In this paper evolution of various methods for Steganography are described starting from Jsteg and ending with LSB+ matching.

Index Terms—Steganography, LSB, DCT, Steganalysis.

I. INTRODUCTION

Steganography is a Greek word which comes from stego means covered and graphy means writing. It is a art of hiding the fact that communication is taking place. Steganography and cryptography are closely related. Cryptography scrambles message so they cannot be understood. Similarly watermarking is also another technique of data hiding, but here image is overlapped and on cover image such that message becomes undetectable. Steganography is mainly of four types viz. 1.Text 2. Audio 3. Image 4. Video. Steganalysis is the counter process to Steganography to detect hidden messages . In this paper we will concentrate on image Steganography with JPEG image format.[1]

II. HISTORY

Form long ago steganographic techniques have been used. Messenger used to tattoo their message on their shaved head then after the hair is grown they used to deliver it. Next was wax table were message was written on the wooden table and then a layer of wax was put on so that message used to pass undetected. Another ancient method is invisible ink. These methods were used to send message in world war II.

III. JPEG IMAGE FORMAT

The file format defined by the Joint Photographic Expert Group (JPEG) stores image data in lossy compressed form as quantised frequency coefficients. First the JPEG compressor cuts the uncompressed bitmap image into parts of 8 by 8 pixels. The discrete cosine transformation (DCT) transfers 8x8 brightness values into 8x8 frequency coefficients. After DCT, the quantisation suitably rounds the coefficients to integers in the range -2048.....2047.

IV. DISCRETE COSINE TTRANSFORM

Discrete Cosine Transform, transfers the pixel values in time domain to frequency domain. They are used by the JPEG compression algorithm to transform successive 8x8 pixel blocks of image into 64 DCT coefficients each. Each DCT coefficient F(u,v) of an 8x8 block of image pixel f(x,y) is given by

$$F(u,v)=\frac{1}{4}C(u)C(v)[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)*\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)u\pi}{16}]$$

Where C(x) = $1/\sqrt{2}$ when x equals 0 and C(x)= 1 otherwise. After calculating the coefficient the following quant sizing operation is performed.

$$F^{Q}(u,v)=[\frac{F(u,v)}{Q(u,v)}]$$

Where Q(u,v) is a 64- element quantisation table. The least significant bits of the quantised DCT coefficients are used mostly for data hiding.

V. JSTEG

In this method after quantization the least significant bits (LSB) of the frequency coefficients is replaced by secret message. The embedding mechanism skips all coefficients with values 0 or 1. If the attack is statistical on JSTEG it mostly discovers the existence of embedded message, because JSTEG replaces the bits and thus it introduces a dependency between the values 's frequency occurrence, that only differ in the bit position. The assumption for a modified image is that adjacent frequencies c2i and c2i+1 are similar. We take the arithmetic mean to determine the expected distribution and compare against the observed distribution ni= c2i.

VI. F5 ALGORITHM

F5 was advanced version of JSTEG, F3 and F4. F5 works with only JPEG image format. F5 overcomes the

drawback of its previous versions i.e security against Steganalysis attacks: Straddling and matrix coding. Straddling randomly but uniformly spreads the message over the cover image to equalize the change density. With matrix embedding F5 improves the embedding efficiency .It can be defined as the number of bits embedded bits per change of DCT coefficient.[5]

VII. LSB ALGORITHM

In the normal LSB hiding ,a string of bit values are replace the original LSBs of pixel values. In this embedding scheme only the LSB plane of the cover image is over written with the secret bit stream according to a pseudorandom number generator. As a result, some structural asymmetry (never decreasing even pixel and increasing odd pixel when hiding the data) is introduced and thus it is very easy to detect the existence of hidden message even at a low embedding rate using some reported steganalytic algorithms such as the chi-squared attack, regular/ singular group(RS) analysis, sample pair analysis etc.

LSB has two main drawbacks:

1. Because of the exclusive shrinkage of Steganography zeroes,F3 effectively embeds more zeroes than ones, and produces- as well as Jsteg, but in different way- statically detectable peculiarities in the histogram.
2. The histogram of JPEG files contains more than even coefficients (excluding 0). Therefore, unchanged carrier media contain (from Jsteg's or F3 perspective) more steganographic ones than zeroes.

The algorithm F4 eliminates these two weakness in stroke by mapping negative coefficients to the inverted steganographic value: even negative coefficients represent a steganographic one odd negative a zero; even positive represent a zero (as before with Jsteg and F3), and odd positive a one.[1] In fig.1 each Two bars of the same height represent coefficients with inverse steganographic value (steganographic zeroes are black, steanographic ones white.)

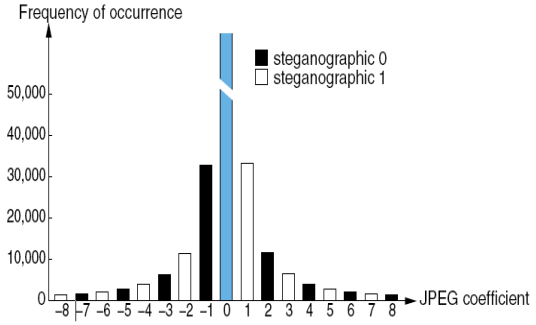

Fig 1.Histogram for JPEG coefficients with F4's interpretation of steganographic values.

VIII. LSB MATCHING

This method employs a minor modification to LSB replacement. If the secret bit does not match the LSB of the cover image then +1 or -1 is randomly added to the corresponding pixel value. Statistically, the probability of increasing or decreasing for each modified pixel value is the same and so the obvious asymmetry artifacts introduced by LSB replacement can be easily avoided. Therefore the common approaches used to detect LSB replacement are ineffective at detecting the LSB matching.[2]

With the LSB matching a bit value bϵ (0,1) is embedded by modifying an integer (such as pixel value or quantized DCT coefficients) i ϵ $[V_{min}, V_{max}]$ to i' :

$$i' = \begin{cases} i; & \text{if LSB}(i) = b \\ i+1; & \text{if LSB}(i) \neq b \text{ and } i = V_{min} \\ & \text{if LSB}(i) \neq b \text{ and } i = V_{max} \\ i+1 \text{ or } i-1 & \\ & \text{if LSB}(i) \neq b \text{ and } V_{min} < i < V_{max} \end{cases}$$

e.g i= pixel intensity value.

IX. LSB$^+$ MATCHING

The evaluated DCT coefficients are arranged in a Zig-zag manner first as shown in figure.

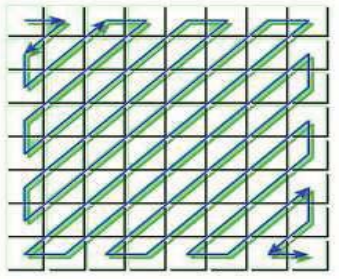

Fig 2. Zig-Zag ordering of DCT coefficients

The LSB$^+$ matching is proposed by dynamically determining the embedding operation in last case of LSB matching i.e, LSB(i) $\neq$ b and $V_{min} < i < V_{max.}$ The no of times of modifying i to i+1 and i-1 to i is counted during the embedding process and denoted by t_i^{i+1} and t_i^{i-1} respectively. Similarly the no times of modifying i+1 to i and i-1 to i will be $t^i_{i+1 \text{ and }} t^i_{i-1.}$ Then the values of $t_i^{i+1} - t^i_{i+1}$ and $t_i^{i-1} - t^i_{i-1}$ are calculated to guide the embedding operation.[3]

$$i' = \begin{cases} i+1 \text{ or } i-1; & \text{if } t_i^{i+1} - t^i_{i+1} = t_i^{i-1} - t^i_{i-1} \\ i+1; & \text{if } t_i^{i+1} - t^i_{i+1} < t_i^{i-1} - t^i_{i-1} \\ i-1; & \text{if } t_i^{i+1} - t^i_{i+1} > t_i^{i-1} - t^i_{i-1} \end{cases}$$

Note that LSB$^+$ matching is not particularly fitting for a specific distribution. It may be applied to a variety of cover object.

X. PROPOSED ALGORITHM:

In proposed algorithm first, DCT of pixel is calculated. Then quantization is done using JPEG Quantization table.

This is followed by arrangement of coefficient in Zig- zag order. Then data is embedded by using LSB algorithm for embedding. The coefficients are divided in four frequency bands viz; DC (1), low frequency (2:10), middle frequency(11:21), high frequency(22:36). Here we use high frequency coefficients for data embedding.

XI. RESULTS

As per the aim of this paper many images were tested for preservation of histogram. Out of them one histogram of one image is shown below.

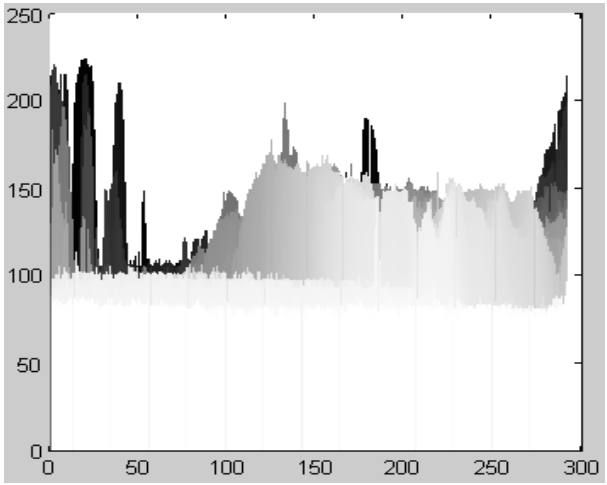

Fig.3 Shows the histogram of cover image.

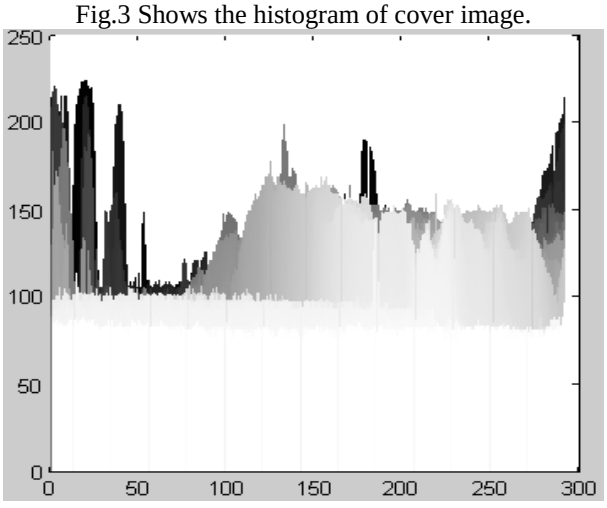

Fig 4 shows the histogram of embedded image.
comparing the above images the histogram is preserved

XII. CONCLUSION

The main aim of all the steganographic methods is data should be sent to the destination without being detected. Various steganalytic methods use different parameters to detect whether the message is embedded in it or not. These parameters are such as comparing histogram, hiding rates, quality factors etc. in this paper we have aimed in preserving the histogram of cover image. As can be seen from fig 2 and fig 3 the histogram of original image is very well preserved.

REFERENCES

[1] Steganography and stegnalysis by J R Krenn.

[2] Edge Adaptive Image Steganography Based on LSB Matching Revisited. Weiqi Luo, Member, IEEE, Fangjun Huang, Member, IEEE, and Jiwu Huang, Senior Member, IEEE

[3] "Data mapping method for Steganography and its application on images.Hao-tain Wu1,Jean-Luc Dudelay1 and Yiu-ming Cheung2

[4] Secure JPEG STEGANOGRAPHY BY LSB+ MATHING AND MULTIBAND EMEDDING.

[5] "MARKOW process based approach to effective attacking JPEG Steganography. "Yun Q.Shi , Chunhua Chen, Wen Chen.

[6] "F5- A Steganographic algorithm high capacity despite better Steganalysis." Andreas Westfeld.

[7] W.luo.W Li, F. Huang, ' Adaptive Steganography for JPEG Images,'' submitted to ACM Transactions on multimedia Computing, Communications and Applications.

[8] "Break our watermarking system 2nd Ed."

[9] C. Chen, and Y.Q. Shi," JPEG Image Stegnalysis Utilizing Both Intrablock and Interblock Correlations," Proceeding of the IEEE International Symposium on Circuits and systems, pp.

[10] J. Fridrich, M. Golian and D. Soukal, " Wet paper codes with improved embedding efficiency," IEEE Transactions on Information Security and Forensics, vol. 1, no.1, pp. 102-110,2006.

◈◈◈