A PERFORMANCE EVALUATION OF JPEG STEGANOGRAPHY TECHNIQUES

Smriti Gupta* Ranjeet Biswas[1]
M.Tech (CSE) IV Sem, Manav Rachna Internationl University*
Assistant director, Manav Rachna International University[1]
guptasmriti09@gmail.com*

***Abstract*— With the rapid application growing of internet and wireless network, information security becomes significant to protect commerce secret and personal privacy. Steganography plays crucial role for information security guarantee. There have been number of steganography embedding techniques proposed over last few years. In this paper, our goal is to evaluate number of JPEG steganography techniques proposed in the literature. Experiment are done on large sets of images for two most important aspects of steganography system i.e imperceptibility and the capacity of stego image and results show that Outguess and F5 are most reliable techniques among universal JPEG steganography techniques.**

Keywords-steganography: JPEG Compression; Steganalysis; Jsteg; F5; Outguess; Matrix encoding; quantization.

I. INTRODUCTION

The word Steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" defining it as "covered writing"[2]. It essentially means "to hide in plain sight". Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected. It hides the existence of a secret message and in the best case nobody can see that both parties are communicating in secret [1]. Many different file format can be used to achieve steganography, but digital images are the popular carrier files by far. It is because the way images are stored creates a great amount of redundant space which is the ideal place to hide information .

Steganography and cryptography are cousins in the spy craft family. They both used to ensure data confidentiality. While cryptography is about protecting the content of messages, steganography is about concealing their very existence.

In this paper we discuss image steganography and how to hide information in jpeg image and we discuss results obtained from evaluating available steganographic software.

II. IMAGE STEGANOGRAPHY

Image steganography is the art of hiding information into a cover image. Each image hiding system consists of an embedding process and an extraction process. An innocuous-looking original image is used as the cover image
to conceal the secret data. The secret data are embedded into the cover image to form a stego image. The embedding process may use an embedding key so that the legal user can successfully extract the embedded data by using the corresponding extraction key in the extraction process. The embedding key and the extraction key are referred to as stego keys. If they are the same, the image hiding system is symmetric, otherwise asymmetric. The two most important aspects of any image based steganographic system are the quality of the stego image and capacity of the cover image.

Data hiding have general requirements;

- Imperceptibility: marked and original data source should be perceptually identical.
- Robustness : The embedded data should survive any attacks.
- Capacity : Maximum data embedding payload
- Security : Security is in the key.

Image steganography schemes can be classified into two broad categories:

a) Spatial-domain based

b) Transform-domain based

In spatial domain approaches, the secret messages are embedded directly. On spatial domain, the most common and simplest steganographic method [2] is the least significant bits (LSB) insertion method. The schemes of second category embed the secret data within the cover image that has been transformed such as DCT, DWT and DFT. The capacity, amount of data embedded within a given image, of spatial domain schemes is better than that of second category. However, the frequency domain schemes have better robustness than that of the first category. Since the watermarking schemes require to be robust, most of the watermarking schemes used are frequency domain. However, steganographic methods need not be robust and instead the capacity and quality are important.

Most image hiding use uncompressed images (eg., BMP)
Or losslessly compressed images (eg., GIF) as cover images. These images potentially contain much visual redundancy so that they can provide large capacity to hide secret data. Many image hiding systems have been developed based on lossless image formats (eg., EzStego [3,4]).

For reducing transmission bandwidth and storing space, the JPEG image is currently the most common format used on the internet. In this paper, steganography in the Discrete cosine Transform (DCT) domain of the JPEG image is focused.

III. REVIEW OF JPEG COMPRESSION

Because steganography in DCT domain of the JPEG image is focused so we describe the JPEG compression in this section.

*She is the author of this paper.

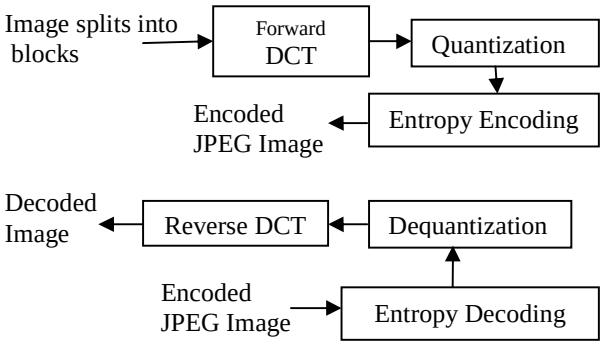

Fig 1. The block diagram of the JPEG compression process

JPEG is an international standard for continuous-tone still image compression which has been approved by International standard organization. The JPEG compression is based on the DCT and allows substantial compression to be achieved while producing a reconstructed image with high visual fidelity.

Fig. 1 shows the JPEG encoding process which compress three major steps; forward DCT, quantization and entropy coding. The encoding process consists of several steps:

1. The representation of the colors in the image is converted from RGB to $Y'C_BC_R$, consisting of one luma component (Y'), representing brightness, and two chroma components, (C_B and C_R), representing color. This step is sometimes skipped which allows greater compression without effect on quality of the image.
2. Due to brightness sensitive receptors in eye, the resolution of chroma data is reduced by a factor of 2 which is known as 'Downsampling'.
3. The image is split into blocks of 8×8 pixels, and for each block undergoes a discrete cosine transform (DCT) and converted to frequency domain.
4. Before computing the DCT of the 8×8 block, 128 is subtracted from each entry to shift from a positive range to one centered around zero.
5. Calculate DCT coefficients by formula given by-;

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

Where

$$C(u) = \frac{1}{\sqrt{2}} \text{ ,if u} \leq 0$$

1, if u > 1

6. These coefficients are then quantized using a quantization table with 64 entries. This step is lossy because of the rounding error. A useful feature in JPEG process in this step varying image compression and quality is obtainable through the selection of specific quantization table. The standard quantization matrix JPEG uses quality factor 50 that as shown in Fig 3.
For a quantity level greater than 50, less compression and high quality is obtained and otherwise vice versa is obtained.
Quantization is achieved by dividing each element in the DCT coefficient block by the corresponding value in the quantization matrix, and the result is rounded to nearest integer. The quantized DCT coefficients $F^Q(u,v)$ are computed by

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

7. The resulting data for all 8×8 blocks is further compressed with a lossless algorithm, a variant of Huffman encoding.

$$F = \begin{bmatrix} 162 & 40 & 20 & 72 & 30 & 2 & -1 & -1 \\ 30 & 108 & 10 & 32 & 27 & 5 & 8 & -2 \\ -94 & -60 & 12 & -43 & -31 & 6 & -3 & 7 \\ -38 & -83 & -5 & -22 & 3 & 5 & -1 & 3 \\ -31 & 17 & -5 & -1 & 4 & -6 & 1 & -6 \\ 0 & -1 & 2 & 0 & 2 & 2 & 8 & 2 \\ 4 & -2 & 2 & 6 & 8 & -1 & 7 & 2 \\ -1 & 1 & 7 & 6 & 2 & 0 & 5 & 0 \end{bmatrix}$$

Fig 2. DCT block

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Fig 3. Quantization table

$$F^Q = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig 4.Quantized DCT block

*She is the author of this paper.

IV. REVIEW ON STEGNALYSIS

Steganalysis is an attempt of discovering hidden data in stego media. A poor steganography technique will arise suspicion by doing visual observation. Whereas other steganography techniques may only be detected through some statistical testing. Some better steganography techniques may be able to withstand either visual or statistical detection. If there exist a method that can detect the existence of a hidden message with success rate better than random guessing, then the steganography system is considered broken.

Most of the embedding process in the steganography system is using the bits manipulating either in sequential or in some pseudo-random pattern. By enhancing the 1 bit-plane and observing this bit- plane, some suspicious artifacts will reveal the effect from steganographic process. In most cases, the 1 bit-plane is the LSB bit-plan and if the LSB is 1 then it is enhanced to the maximum pixel value otherwise if it is 0 then it is remained 0.This is the simplest steganalysis and known as visual attacks[3].This steganalysis is applicable to detect LSB embedding in spatial domain image and palette images, but not in transform domain image.

A statistical analysis proposed by Pfitzman and Westfeld [8] is Pair of Values (PoVs).The idea of this steganalysis is that during the embedding, it is actually swapping one value, A (*2i*) into another value, B (*2i+1*) and vice versa. This does not change the sum of both values of occurrences in the image. Thus, this will form a pair of values and it is observed that before the embedding, these two values are distributed unevenly, but after the embedding, these values tend to become equal. As a result, statistical Chi-square test is a suitable tool to be used. Fridrich et al. [3] introduced a method known as RS steganalysis. This method uses statistics derived from the spatial domain of an image. According to them, the LSB can be predicted up to certain level from the remaining of the 7 bits.

Apart from those steganalysis methods described above, there are also some other methods like universal blind steganalysis and unique fingerprints that can be found in [4]. There is also some steganalysis method that only attacks on a specific steganography system like the one proposed in [5] and known as targeted steganalysis.

V. JPEG Compression Technique in Steganography.

In steganography using digital images, data embedding into compressed images should be primarily considered since images are usually compressed before being transmitted. The JPEG compression using the discrete cosine transform (DCT) is the most common compression standard for still images, therefore many steganographic methods have been proposed for JPEG images [6-12].

In 1999, a method to hide data into JPEG compressed images that uses DCT was introduced. That method embeds only one secret bit into the 64^{th} quantized DCT coefficient of a DCT block in zigzag order [11]. Besides, a different quantization table is offered to the JPEG decoder so as to reduce the noise caused by the secret data. Since a small change of the quantized DCT coefficient will lead to significant distortion in the decoded image, the value in the quantization table for the position the embedded data is changed to 1. These methods are able to produce a stego-image with a small distortion. However, the information hiding capacity is very limited. A 512 x 512 gray level image can hold only 4096 bits.

A. JPEG-JSTEG

Jpeg-Jsteg was among the first secret data hiding tool for embedding secret information into JPEG compressed images[12]. Jpeg- Jsteg embeds one secret bit in the LSB of the quantized DCT coefficients whose values are 0, 1, or –1. The information hiding capacity of Jpeg-Jsteg is improved but is still limited. The number of bits that can be embedded becomes smaller when the compression ratio gets higher. In the literatures concerned, there are few data hiding methods that hide the secret data in JPEG-compressed images. The main drawback of these methods, are that the information hiding the capacity is low and their neither bit rate nor the distortion of the stego-image can be controlled. Chi-square attack provides very reliable results when the LSB placement is known therefore, the data embedded by the J-steg can be detected by chi-square attack.

Embedding Algorithm of Jpeg-Jsteg:
Input: Secret message, cover image
Procedure:
Step 1: Encode the message using Huffman codes.
Step 2: Divide the cover image into 8x8 blocks.
Step 3: Calculate DCT coefficients for each block.
Step 4: Quantize the coefficients
Step 5: while complete message do not embedded do
5.1 get next DCT coefficient
5.2 if DCT $\neq 0$, DCT $\neq 1$ and DCT $\neq -1$
Then
5.3 get next bit from message
5.4 replace DCT LSB message bit

*She is the author of this paper.

End {if}
End{while}
Step 6: De-quantize and take inverse DCT
to obtain stego image.

Extracting Algorithm
Input : Stego image
Procedure:
Step 1: Divide the stego image into 8x8 blocks.
Step 2: Calculate DCT coefficient for each block.
Step 3:Quantize the coefficients
Step 4: while secret message not completed do
4.1 get next DCT coefficient
4.2 if DCT ≠ 0 , DCT ≠1 and DCT ≠ -1
Then concatenate DCT LSB to
Secret message
End {if}
End{while}
Step 5: Decode secret message bits using
The Huffman codes.
End.
Output: Secret message.

B. F3 & F4 ALGORITHM

In F3 algorithm [7], decrease absolute value of the coefficients to be matched between secret message bit and LSB of JPEG coefficient. If the coefficient becomes zero after decrease operation, we exceptionally skip decrease operation and check next coefficient.
F4 algorithm developed to complement the weekness of F3 [7].A negative coefficient is inversed. To match LSB of coefficient with secret message bit, negative coefficient should be increased and positive coefficient should be decreased depends on the secret message bit.

C. F5 ALGORITHM

To provide a secure and high capacity JPEG steganography, Westfeld [6] proposed the F5 algorithm in 2001.It is a widely used algorithm now. Instead of replacing the LSB of the quantized DCT coefficients with the secret bits, the absolute value of the coefficient is decreased by 1. Besides, the F5 algorithm randomly chooses DCT coefficients to embed the secret bits and employs the matrix embedding that minimizes the necessary number of changes to embed a message of certain length. This algorithm successfully defends both the chi-square attack and extended chi-square attacks.

D. OUTGUESS ALGORITHM

In the same year, Provos [8] also proposed the outguess steganographic algorithm to counter the chi-square attack .The outguess uses two passes to achieve the embedding mission. In the first pass, the secret bits are embedded in the LSB's of the quantized DCT coefficients along a random walk. In the second pass, the histogram of the stego image is adjusted to match that of the cover image is similar to that of the cover image, the an avoid the detection of the chi-square family attack. Because the histogram of the final stego image is similar to that of the cover image, the stego image created by the Outguess algorithm can avoid the detection of the chi-square family attack.

E. MATRIX ENCODING

Crandall proposed Matrix encoding [10] to improve the embedding efficiency by decreasing the number of required bit changes. Westfeld proposed F5 algorithm [20] which implement matrix coding .This algorithm modifies the set of LSBs of quantized DCT coefficients after round step. The notation (1,n,k), where n denotes embedding k message bits into an n bits block by modifying one bit of the set. C consist of blocks of length n and message, M, consists of blocks of length k.

F. MODIFIED MATRIX ENCODING

In previous encoding shows that F5 used (1,n,k) code to embed k bits into an n bits LSB block. Y. Kim et al proposed to use (t, n, k) to increase the possibility of bit change choice in each cover block. It is called MME (Modified Matrix Encoding) [13].

G. MODIFIED JPEG JSTEG

This algorithm evaluates the performance and efficiency of using optimized quantization tables instead of default JPEG tables within JPEG steganography. It was found that using optimized tables significantly improves the quality of stego-images. Moreover, this optimization strategy is used to generate a 16x16 quantization table instead of that suggested in [14]. The quality of stego-images was greatly improved when these optimized tables were used. In this method, for each 16x16 quantized DCT block, the least two- significant bits (2-LSBs) of each middle frequency coefficient are modified to embed two secret bits. Additionally, the Jpeg-Jsteg embedding technique is used for the low frequency DCT coefficients without modifying the DC coefficient but this approach can provide a higher information-hiding capacity than jpeg-jsteg.

Embedding algorithm:

1. The message (M) to be embedded in the cover image is randomly generated.
2. The cover image is divided into non-overlapping blocks of 16x16 pixels and then the DCT is used to transform each block into DCT coefficients.
3. The DCT coefficients are scaled by the optimized and modified 16x16 quantization table (Table 6.B). In this quantization table, the values of (1) represent the middle frequencies to be used for embedding (242

*She is the author of this paper.

bits). The quantized DCT coefficients of each block are rounded to the nearest integers and then set in zigzag scan order.

4. The least two-significant bits of each middle frequency coefficient in the quantized DCT blocks are modified to embed two secret bits.
5. The JPEG entropy coding (DPCM, Run-Length coding and Huffman coding) is applied to compress these resultant blocks, and then the jpeg file is obtained.

H. COMPLEMENTARY EMBEDDING

In this algorithm, a secure and high-capacity JPEG steganography is proposed. Instead of flipping the LSBs of the DCT coefficients, the secret bits are embedded in the cover image by subtracting one from or adding one to the non-zero DCT coefficients. It is achieved by dividing the quantized DCT coefficients and the secret bits into two parts according to a predefined partition ratio [15]. Therefore, this method cannot be detected by both the chi-square and the extended chi-square attacks.

Embedding algorithm:

1. Transform the raw data of the cover image into DCT coefficients and then the quantized coefficients is rounded to the nearest integers D.
2. Use a stego key K1 to permute the quantized coefficients Q.
3. Divide Q into two parts, Q1 and Q2, according to a predefined separation ratio x.
4. Use a crypto-key k2 to encrypt the original message O, and obtain the secret-bit sequence S.
5. Divide S into two parts, s1 and s2, according to a predefined separation ratio x.
6. Let L1 & L2 denote the length of s1 & s2 respectively, concatenate L1 and s1 & L2 and s2 to form the secret message M1 & M2 respectively. That is,
   M1=L1*S1,
   M2=L2*S2, where '*' denote concatenation operation.
7. Embed M1 & M2 into the non-zero coefficients of Q1 and Q2 respectively.
8. Combine the modified Q1 and Q2 to form a single coefficient sequence Q3 and use the stego-key k1 for coefficient de-permutation (D1).
9. Compress the D1 using entropy encoding to obtain the JPEG stego-image.

I. QET BASED JPEG STEGANOGRAPHY

The proposed method hides secret information in JPEG-compressed images according to the QET entries [16]. Data embedding requires three steps: selecting DCT coefficients, information hiding and modification of quantization table.

From the Fig. 4, one can observe that many DCT coefficients in high frequency area tend to be zero after the quantization step in JPEG compression. The DCT coefficients that turn out to be zero after quantization are selected for embedding the secret information. Each blocks having 64 DCT coefficients, as F0 to F63. Mostly in all the DCT blocks the value above F25 is zero. So the DCT coefficients from F26 to F63 are selected in zigzag order for hiding secret data in each block results in better image quality and increased capacity than other jpeg compressed image method as about 58% of the pixels are used for data hiding in every 8 x 8 DCT block.

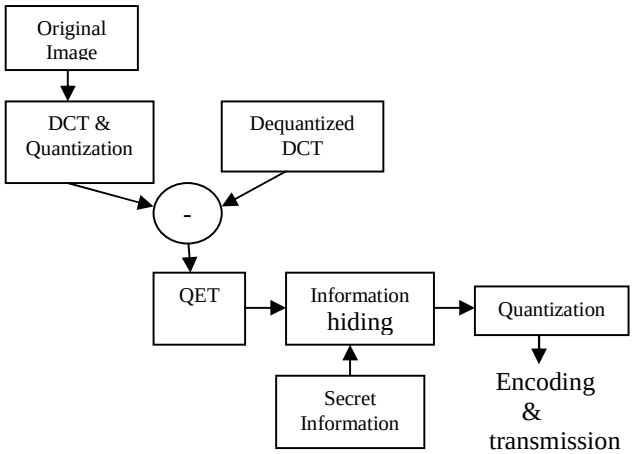

Fig 5. Block diagram of QET method.

V. PERFORMANCE ANALYSIS

As a performance measurement for image distortion, the well known Peak-Signal-to-Noise Ratio, PSNR, which is classified under the difference distortion metrics, can be applied to the stego-images. It is defined as:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{MSE} \text{ dB},$$

where MSE denotes the Mean Square Error between the original image and reconstructed image. For an N x N image, its MSE is defined as:

$$\text{MSE} = \left(\frac{1}{N}\right)^2 \times \sum_{i=1}^{N} \sum_{j=1}^{N} \left(x[i, j] - \bar{x}[i, j] \right)^2$$

Here X[i , j] and X'[i , j] denote the original and decoded gray levels of the pixel [i , j] in the image respectively. A larger PSNR value means that the stego-image preserves the original image quality better. PSNR values falling below 30dB indicate a fairly low quality i.e the distortion caused by embedded image is obvious.

*She is the author of this paper.

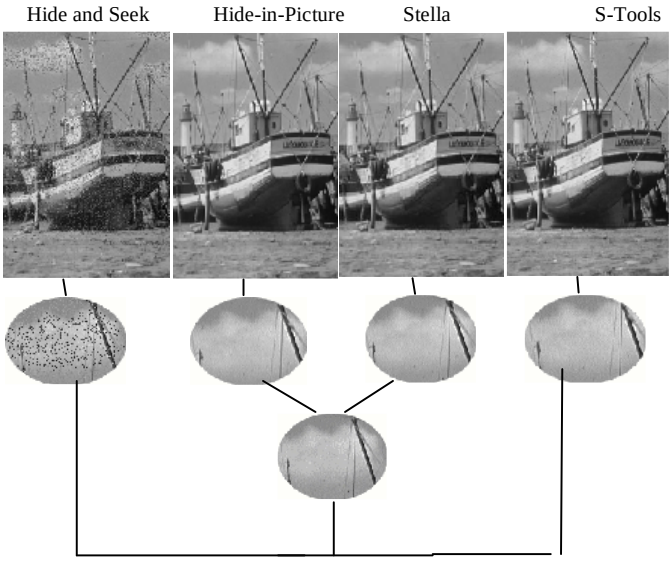

Fig 6.2 Stego images of each software tool appearing in the Table 1.1.

A quality stego image should strive for a PSNR value and above. “Table 1.1” shows the PSNR values spawned by various software based domain methods and applied on the images as shown in Fig 6.1, Fig 6.2, Fig 6.3 and Fig 6.4”,which depict the output of each of the tools

TABLE 1.1 SUMMARY OF PERFORMANCE OF COMMON SOFTWARE

SOFTWARE	PSNR	VISUAL INSEPCTION
Hide & Seek	18.608	Very clear grainy noise in the stego image which renders it the worst performer.
Hide in Picture	23.866	Little Noise. Accepts only 24 bit bmp files. Creates additional color palaette entries. In this case the original boat image has 32 colors and the generated stego-image augmented the number to 256 by creating new colors.
Stella	26.769	Little Noise. Accepts only 24 bit bmp files.
S-Tools	37.775	No visual evidence of tamper.

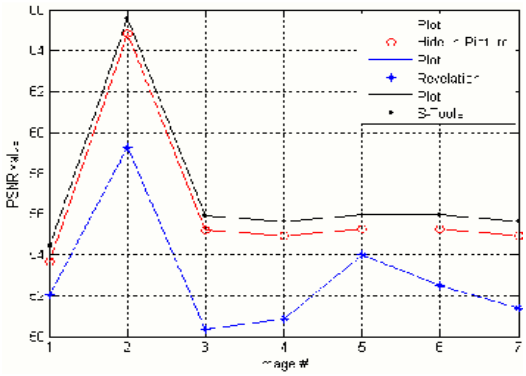

Fig 6.3 Additional experiments on stegnography Software.

A Performance analysis of some steganographic tools is studied in [7].In “Table 1.2”, the sign (√) indicate the charcterstic is present, (-) indicates the unavailability of information while (×) gives the negative response.In the table columns,(1)(2) represent frequency domain (3) encryption support (4) random bit selection.

TABLE 1.2 COMPARISION OF DIFFERENT TOOLS OF JPEG FORMAT

Name	Creator	Year	(1)	(2)	(3)	(4	Detected by
Jsteg	Derek Upham	-	×	DCT	×	×	Chi-square test
JP Hide	Allan Westfeld	2001	×	DCT	Bloufish	×	Chi-square test & steg detect
F5	Andreas Westfeld	2001	×	√	Bloufish	√	Fridrich’s Algo
Outguess	Provos & Honeymoon	2001	×	DCT	Rc4	√	Fridrich’s Algo

Table 1.3 shows the performance analysis of capacity (in bits) for various embedding algorithms applied on various images as shown “Fig 6.5, Fig 6.6, Fig 6.7, Fig 6.8”, which depict the output of each jpeg embedding algorithms.

*She is the author of this paper.

TABLE 1.3 COMPARISION OF CAPACITY (in bits) FOR VARIOUS EMBEDDING ALGORITHMS

Test image	Jsteg	F5	Outguess	Complementary embedding	Modified Jsteg
Lena	32998	33026	16375	44131	255558
Tiffany	31674	31516	15729	43300	255283
Pepper	34295	34074	17016	46346	258977
Babboon	75751	75837	37867	98989	258977

TABLE 1.4 COMPARISION OF PSNR Values (in db) OF THE STEGO IMAGES CREATED BY VARIOUS ALGORITHMS.

Test image	Jsteg	F5	Outguess	Complementary embedding	Modified Jsteg
Lena	36.36	36.94	36.37	35.67	37.1
Peppers	35.45	35.86	35.32	34.75	35
Tiffany	35.93	36.36	35.81	35.07	36


Fig 6.5 Lena


Fig 6.6 Tiffany

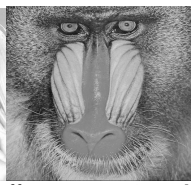
Fig 6.7 Baboon


Fig 6.8 Peppers

VI. \CONCLUSION

In this paper, we have analyzed the various JPEG compression technique and Steganography tools. We observe from various performance analysis of the various embedding algorithms that Among the basic steganography techniques for jpeg format F5 and Outguess are most reliable algorithms (work in DCT domain) although these algorithms are also vulnerable to recompression which is overcome by DWT domain. The embedding process can also be done by modifying some coefficients, which are selected according to the type of protection needed; i.e if we want our message to be imperceptible then the high range of frequency spectrum is chosen but if we want our message to be robust then the low range of frequency spectrum is selected. Usually, the coefficients to be modified belong to the medium range of frequency spectrum, so that a tradeoff between perceptual invisibility and robustness is achieved.

*She is the author of this paper.

REFERENCES

[1] Jhonson, N F and Jajodia,S "Exploring steganography:Seeing The Unseen," IEEE, Computer, pp. 26-34, Feb 1998.

[2] Niels Provos,Peter Honeyman, "Hide and seek: An Introduction to steganography", IEEE, Computer society, 2003, pp. 32-44.

[3] J.Fridrich,M.Golian,"Practical Steganalysis–State of Art",Electronic Imaging 2002,Security and Watermarking of Multimedia Contents, San Jose, California, Proc.SPIE Photonics West,Volume 4675, pp.1-13, January 2002.

[4] J. Fridrich, M. Goljan, D. Hogea, "New Methodology for Breaking Steganographic Techniques for JPEGs",Proc. SPIE Electronic Imaging, Santa Clara,CA, pp. 143-155, January 2003.

[5] J.Fridrich,M.Goljan,D.Hogea,"Steganalysis of JPEG images: Breaking the F5 Algorithm", Information Hiding:5th International Workshop, Book Series Lecture Notes in Computer Science, Volume 2578/2003, 2002, pp.310-323, SpringerLink,Netherlands.

[6] Andreas Westfeld , "F5- A Steganographic Algorithm High Capacity Despite Better Steganalysis," 4th Information Hiding International Workshop, Pittsburgh, USA, vol.2137, April 2001,pp.289-302.

[7] M. Chen, R. Zhang, X. Niu and Y. Yang , "Analysis of current Steganography Tools Classification & Features", Proc. International Conference on Intelligent Conference on Intelligent Information Hiding and Multimedia signal processing, Pasadena.CA, USA, December 2006.

[8] J.Fridrich, M. G Goljan,D.Hogea, "Attacking the OutGuess",Proc. Of the ACM Workshop on Multimedia and security.2002,Juan-les-pins,France, December 6, 2002.

[9] J.Fridrich and D. Soukal, "Matrix Embedding for large Payload",IEEE Tran on Information Forensics and Security.Vol 1,pp. 390-295,September 2006.

[10] Y.H. Kim, Z. Duric and D.Richards, " Modified Matrix Encoding Technique for Minimal Distortion Steganography",Proc.of Information Hiding, 2007,pp.314-327.

[11] Y.S Choi, H.J Kim, and C.M. Park, "The Influence of Quantization Table in view of Information Hiding Techniques Modifying Coefficients in Frequency Domain", Journal of IEEE,Vol46,pp.56-63,January 2009.

[12] Derek Upham: Jsteg, 1997, eg. http://www.tiac.net/users/korejwa/jsteg.htm.

[13] Y.S Choi and H.J Kim, "Improving the Modified Matrix Encoding on steganography Method",Proc. of IEEE 5th International Conference on Information Assurance and security,2009,pp.205-208.

[14] A. Almohammad, R. M. Hierons and G. Ghinea, "High Capacity Steganographic Method Based Upon JPEG", The Third International Conference on Availability, Reliability and Security. *ARES08*, Barcelona, Spain, 4-7 March, 2008, pp. 544-549.

[15] M. Ishaque and Dr.S.A.Sattar, "Quality based JPEG Steganography using balanced Embedding Technique", Proc. of IEEE 2nd International Conference on Emerging Trends in Engineering and Technology,2009,pp.215-221.

[16] Brabin, D.R.D and Sadasivam, V, "QET based Steganography Technique for JPEG Images",Proc.International conference on Control,Automation,Communication and Energy conservation,pp.1-5,June 2009.

*She is the author of this paper.

*She is the author of this paper.