A Discrete Wavelet Transform based Adaptive Steganography for Digital Images

Pooja Rai[1]
Department of CSE, Sikkim Manipal Institute of Technology, Sikkim Manipal University,
Majhitar, 737134, India
[1]poojasampangrai@gmail.com

Sandeep Gurung[2]
Department of CSE, Sikkim Manipal Institute of Technology, Sikkim Manipal University,
Majhitar, 737134, India
[2]gurung_sandeep@yahoo.co.in

Mrinal Kanti Ghose[3]
Department of CSE, Sikkim Manipal Institute of Technology, Sikkim Manipal University,
Majhitar, 737134, India
[3]mkghose2000@yahoo.com

Abstract- Steganography is the scheme of surreptitious communication by hiding the data inside data with the aim of concealing the presence of secret data from inquisitive eyes. Pervasiveness as well as availability of redundant information in an image makes it the alluring carrier medium. However, all parts of an image cannot be used evenly to hide the secret information. In this paper, an approach for image steganography has been made through a method that exploits standard deviation of high frequency components of the carrier image to identify the potential region for secret embedding. Discrete Wavelet Transform (DWT) is used to segregate high frequency and low frequency components of the image. The secret embedding is done in three higher frequency components with non-uniform high embedding efficiency method. The block having standard deviation lower than the mean value is embedded with relatively higher efficiency compared to the blocks with higher standard deviation. Secret image is encrypted using a chaotic mapping which creates diffusion and thus is safe even in the situation where an adversary have the knowledge of embedding technique used. The experimental results for the method were observed with acceptable cover-stego structural similarity, stego image fidelity, embedding efficiency and statistical imperceptibility.

Categories and Subject Descriptors

Security and Privacy~ Formal methods and theory of security

Keywords: Image Steganography; Discrete Wavelet Transform; Standard Deviation; Matrix Encoding; Pairing Function; Arnold Transformation; Secret Extraction Key; Structural Similarity Index Metrics (SSIM)

1. INTRODUCTION

With the rapid growth of Information Technology, sharing of data over the internet has become more desirable by the people all over the world. However, security, authenticity as well as integrity of the data being communicated are vulnerable to various attacks.

Steganography is an approach to secured communication which hides the very existence of the secret information using a host medium acting as an envelope.

Image Steganography is a data hiding technique that allows secret hiding with least visual distortion in the carrier image. The image selected to carry the secret is known as *cover image* and the resultant image with the hidden secret is known as *stego image*. The cover and the stego images are visually indistinguishable and no one can predict the existence of secret except the deliberate receiver.

Image Steganography is categorized into *spatial* and *transform* domain based methods [5]. Spatial domain based methods directly make use of the pixel value to hide the payload whereas the latter uses the transformed components of the image for the same purpose as that of the former. *Adaptive Steganography* is a particular case of two former methods which considers the statistical global features of the cover image prior to secret embedding in the transformed domain [5].

Steganography based on *DWT* have been already proposed, including many block partitioning based techniques [6, 7, 11] with an adequate level of visual distortion in a cover image. Hence, enhancement in the security of the common *DWT* based schemes can be achieved by combining the features of both the frequency domain and the block based techniques. However, the major issues discussed in section I should also be taken into account in the process of data hiding. An approach for object based steganography is proposed in [12]. In this approach an image is segmented into smooth and textured area for variable rate embedding with appealing result for the imperceptibility, stego-image fidelity as well as embedding efficiency in the spatial domain. Inorder to keep the security of the payload being embedded intact, a process of chaotic mapping is introduced in [10] which will keep the payload secure even if the adversary have the knowledge of the embedding mechanism used. Apart from chaotic mapping, edge detection mechanism is also used to identify the edges of the image to be used for embedding in the spatial domain [10]. Thus, it is evident that textured region (including edges) of an image can be mainly used to embed comparatively larger amount of data than smoother region and smoother region requires higher embedding efficiency than that of the textured region [12]. Inorder to identify such regions, image block partitioning followed by some block wise statistical calculation leading to an *Adaptive Steganography* would be the suitable approach. Hence, an attempt has been made to devise such an

 ISBN 978-1-4503-3361-0/15/08...$15.00
DOI: http://dx.doi.org/10.1145/2791405.2791514

approach in frequency domain (using *DWT)* by taking *Standard Deviation* as a measure for segmenting the high frequency components of the cover image block wise and embedding secret with least possible distortion using *Matrix encoding* in a non-uniform manner.

2. PROPOSED METHODOLOGY

2.1 Stage I

2.1.1 Payload Transformation

The secret image is scrambled prior to embedding by using Arnold Transformation (also known as Cat Mapping) [2]. It is a chaotic mapping discovered by Russian mathematician Vladimir I. Arnold, applying which an image will be scrambled by randomizing the original organisation of the pixels and the original image will be back after a specific number of iterations [2]. The iteration at which the original image reappears is known as *period*. The algorithm for payload transformation is given as follows:

Input: Original payload *S*
Output: Distorted payload *D*
Algorithm:
Step1: Find the period i.e. number of Arnold Transform required to regenerate the original payload. Let it be denoted by *p*. Let $t=\left\lceil\frac{p}{2}\right\rceil$ and *req_trns=p-k*
Step2: Generate *D* by applying Arnold Transformation on S for *t* times.

Figure 1. (a) Original secret image (b) Scrambled image after *t* Arnold transforms

2.1.2 Generation of Secret Extraction Key

In order to enhance the security, the dimension of payload, block size as well as the value of *req_trns* are used to create the secret extraction key using Cantor Pairing function [23]. Let m and n be any two integers. The Pairing function is given as follows:

$$P(m,n) = (m+n) \times \frac{(m+n+1)}{2} + n \qquad (1)$$

The key for the proposed algorithm is generated as follows:

$$P(r, c) = u$$
$$P(req\_trns, b\_size) = v$$
$$P(u, v)=V \qquad (2)$$

Where *b_size* is the block size of cover image, *r* and *c* represents the row and column of payload respectively and *V* is the final key. Key decryption is done as follows:

$$v = \left\lfloor \frac{\sqrt{8V+1}-1}{2} \right\rfloor$$
$$q = \frac{v^2+v}{2}$$
$$n = V - q$$
$$m = v - n \qquad (3)$$

The third value from the key is generated by repeating the above process by putting the value of *m* into *V*.

2.1.3 Embedding Algorithm

The embedding of payload is done using *Matrix Encoding* method [3]. For embedding *b* bits of data with *c* number of changes, the embedding efficiency is defined as follows:

$$Embedding\ efficiency = \frac{b}{c} \qquad (4)$$

Let *I* be an image with *n* modifiable bit positions for *m* secret message bits *b*. With a hash function *F* that extracts *m* secret bits from *I,* modified image *I'* for every *m* and *b* with *b=F(m')* can be created with at most d_m Hamming distance between *m* and *m'*. An ordered triple (d_m,*n,m*) can be used to denote the above method [4].
The proposed algorithm makes use of above encoding with d_m *=1, n=3,7and m=2,3*. Let x_i denote the bit positions *(1≤i≤7)* and b_j denote the message bits *(1≤j≤3)*.

Table 1: Embedding process for (1,3,2) scheme

Condition	Action to be taken
$b_1 = x_1 \oplus x_3$ and $b_2 = x_2 \oplus x_3$	Keep original values
$b_1 \neq x_1 \oplus x_3$ and $b_2 = x_2 \oplus x_3$	Change x_1
$b_1 = x_1 \oplus x_3$ and$b_2 \neq x_2 \oplus x_3$	Change x_2
$b_1 \neq x_1 \oplus x_3$ and $b_2 \neq x_2 \oplus x_3$	Change x_3

Table 2: Embedding process for (1,7,3) scheme

Condition	Action to be taken
$b_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7$, $b_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$	No change
$b_1 \neq x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_1
$b_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_2
$b_1 \neq x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_3
$b_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_4
$b_1 \neq x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_5
$b_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_6
$b_1 \neq x_1 \oplus x_3 \oplus x_5 \oplus x_7$ $b_2 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$ $b_3 \neq x_2 \oplus x_3 \oplus x_6 \oplus x_7$	Change x_7

The smoother region is embedded with (1,7,3) scheme i.e. 3 secret bits are embedded into the 7 bits of high frequency coefficients changing only one bit and textured region is embedded with (1,3,2) scheme i.e. 2 secret bits are embedded into the 3 bits of high frequency coefficients changing one bit.

Input: A gray scale Cover Image *I*, Distorted Payload *D*, block size *b_size*.
Output: Stego image *S*
Algorithm
Step 1: Apply *2D-Haar DWT* to *I* in order to get four sub-bands namely *LL, LH, HL*, and *HH*.
Step 2: Calculate the length of payload *D*. Let it be denoted by *L*
Step 3: Convert the payload to its binary equivalent. Let it be denoted by *MSG_BITS.*
Step 4: Construct a matrix *M* by concatenating *HH, LH, HL sub-bands.*
Step 5: Segment *M* into non-overlapping blocks with dimension *b_size* × *b_size.* Let the blocks be stored in *IMG_BLOCK* with *length* denoted by *tot_blocks* .
Step 6: Repeat *for k=1 to tot_blocks*
Step 6.1: Calculate standard deviation of *IMG_BLOCK(k)* and store in the matrix *STD.*
[End of *for* loop]
Step 7: Generate a random number sequence in *RAND_BLOCKS* taking *seed* value as *tot_blocks.*
Step 8: Set i:=1, j:=1, SS:= IMG_BLOCK, Let MEAN_STD denotes the Average of values in *STD*
Step 9: Repeat *while L≠0 //Till all secret bits are embedded*
Step 9.1: Set *block_no:= RAND_BLOCKS(i), i:=i+1,*
Step 9.2: Set block_chosen:= IMG_BLOCK(block_no), convert it into binary form denoted by BIN_BLOCK and find its length. Set *block_len*:= Length(BIN_BLOCK)
Step 9.3: If STD(IMG_BLOCK(block_no)< MEAN_STD then
Step 9.3.1: Repeat *while j<= block_len* //Till all the LSB of the block are chosen
Step 9.3.1.1: Choose *seven* LSB bits from BIN_BLOCK and *three* bits from *MSG_BITS*
Step 9.3.1.2: Embed into BIN_BLOCK according to Table 2 and *Set L:=L-3, j:=j+56.* Let BIN_BLOCK after modification and decimal conversion be denoted by *EMBEDDED_BLOCK.*
Step 9.3.1.3: If L<1 then go to *Step 9.4 //All secret bits are embedded without using all LSB bits from the block*
[End of *if* structure]
[End of *while* loop]
Else
Step 9.3.2: Repeat *while j<=* block_len //Till all the LSB of the block are chosen
Step 9.3.1.1: Choose *three* LSB bits from BIN_BLOCK and *two* bits from *MSG_BITS*
Step 9.3.1.2: Embed into *I* according to Table 1 and *Set L:=L-2, j:=j+24.* Let BIN_BLOCK after modification and decimal conversion be denoted by *EMBEDDED_BLOCK*
Step 9.3.1.3: If L<1 then go to *Step 9.4 //All secret bits are embedded without using all LSB bits from the block*
[End of *if* structure]
[End of *while* loop]
[End of *if* structure]
Step 9.4: Set *SS(block_no):= EMBEDDED_BLOCK //Give back the modified block to SS*
[End of *while* loop]
Step 10: Encrypt the values of *req_trns,* dimension of the payload and block size *b_size* using cantor pairing method. Let the generated key be *K.*
Step 11: Apply *2D-Haar DWT* to *LL* sub band to get further sub-bands *LL1, HL1, LH1, HH1*. Embed *K* into *HH1* sub-band using the same technique mention above and get back modified *LL* by applying *IDWT* using *LL1, HL1, LH1, HH1*.
Step 12: Decompose *SS* into modified *HH, HL and LH* sub-bands denoted by *HH', HL', LH'* respectively.
Step 13: Perform *IDWT* using *LL, HH', HL', LH'* sub-bands to get the stego image *S*.

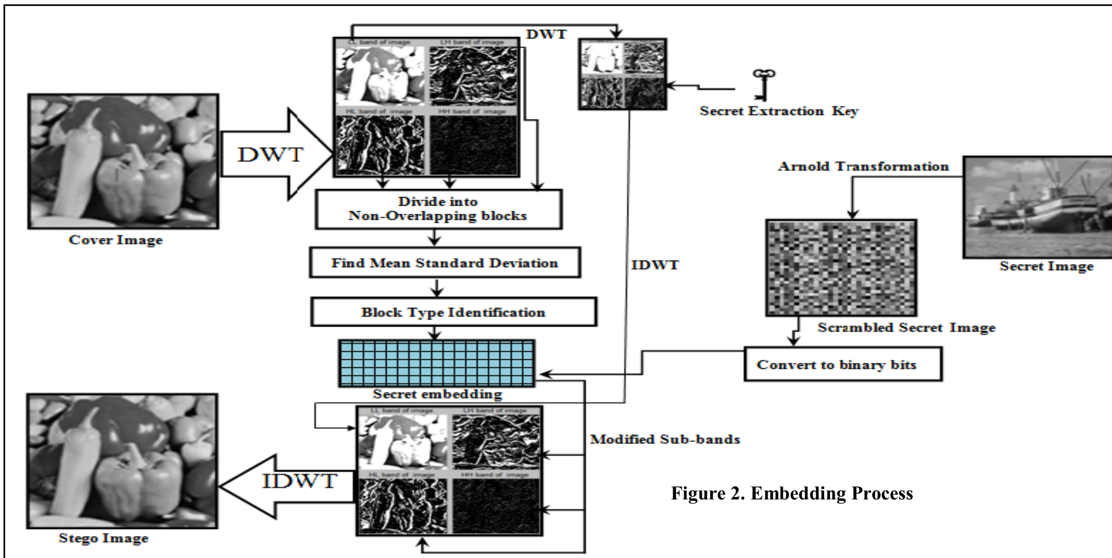

Figure 2. Embedding Process

2.2 Stage II

2.2.1 Payload Extraction

The extraction of the payload is done by performing the reverse operation as that of the embedding algorithm. The algorithm is as follows:

Input: Stego image *S*

Output: Payload *P*

Algorithm

Step 1: Transform *S* to four sub-bands namely *LL, LH, HL*, and *HH by applying 2D-Haar DWT.*

Step 2: Get *HH1* by applying *2D-Haar DWT to LL sub-band.*

Step 3: Extract the vales of *req_trns, b_size* and dimension of payload from *HH1* by applying the Cantor extraction mechanism explained in equation 9. Let the length of payload be denoted by *L*

Step 4: Perform the same operations specified in embedding algorithm from *Step 4* to *Step 8*

Step 5: Let the extracted bits are stored in *SECRET_BITS.* Set *i:=1* and *j:=1*

Step 6: Repeat *while L≠0 //Till all secret bits are extracted*

Step 6.1: Set *block_no:= RAND_BLOCKS(i), i:=i+1,j:=1*

Step 6.2: Set block_chosen:= IMG_BLOCK(block_no), convert it into binary form denoted by BIN_BLOCK and find its length. Set *block_len*:= Length(BIN_BLOCK)

Step 6.3: If *STD(IMG_BLOCK(block_no)< MEAN_STD then*

Step 6.3.1: Repeat *while j<= block_len* //Till all the LSB of the block are chosen

Step 6.3.1.1: Extract *three* bits from *IMG_BLOCK* by just performing the operation specified in *TABLE 2* in reverse order and append into *SECRET_BITS. Set L:=L-3, j:=j+56*

Step 6.3.1.2: If L<1 then go to *Step 6.4 //All secret bits are extracted*

[End of *if* structure]

[End of *while* loop]

Else

Step 6.3.2: Repeat *while j<= block_len* //Till all the LSB of the block are chosen

Step 6.3.2.1: Extract *two* bits from *IMG_BLOCK* by just performing the operation specified in *TABLE 1* in reverse order and append into *SECRET_BITS. Set L:=L-2, j:=j+24*

Step 6.3.2.2: If L<1 then go to *Step 6.4 //All secret bits are extracted*

[End of *if* structure]

[End of *while* loop]

[End of *if* structure]

Step 6.4: Convert the values of *SECRET_BITS* to decimal and save it into *P*

[End of *while* loop]

3. EXPERIMENTAL RESULTS

The simulation of the algorithm has been done in MATLAB 10 and tested on a 64-bit 2.30 GHz Intel Core i5 processor computer. Different standard test images are taken as the carrier image and results for block size of 8×8 has been shown in the following section taking standard lena image as the secret image.

3.1 Analysis of Key Strength

As mentioned in Shabnam et.al. [9], the higher number of digits in the key ensures the safety of the key against Brute Force attack. The probability of the deciding of the key through random guess or Brute Force attack is given as $P_{decode}=\frac{1}{10^k}$ where *k* is the length of the key [9].

A decoding key generated with block size 8×8, period as 30 (for Arnold Transformation) for embedding a secret image of size 110×110 is of 10 digits. The decidability of key is given as follows:

$$P_{decode}=\frac{1}{10^{10}} = 1 \times 10^{-10} \qquad (5)$$

The value given by equation 12 reveals that there is adequately large key space thus providing imperceptibility against Brute Force attack.

3.2 Embedding Distortion Measurement

The amount of alteration produced in the cover image due to secret embedding is known as stego-image fidelity [22].One of the well known stego-image fidelity measurement techniques is *Peak Signal to Noise Ratio (PSNR)* which is calculated using Mean Square Error (MSE) and is defined as follows:

$$MSE = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij}-y_{ij}\right)^2 \qquad (6)$$

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)db \qquad (7)$$

Where x_{ij} denote the pixel value of cover image with size *M×N* and y_{ij} is the pixel value of stego image. Higher values of PSNR indicate lower embedding distortion leading to better fidelity and the expected value is above 40 db. Another important metric to find the quality of the stego image is *Structural Similarity Index Metrics [SSIM].*The stego image quality is assessed by comparing it with cover image [10].The *Structural Similarity* between the two images *X* and *Y* of common size *N×N* is calculated with different window sizes of the image and is given as follows:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x{}^2+\mu_y{}^2+c_1)(\sigma_x{}^2+\sigma_y{}^2+c_2)} \qquad (8)$$

Where the averages of all windows of X and Y are given by μ_x,μ_yrespectively. The covariance of X and Y is given by σ_{xy} *and* $\sigma_x{}^2,\sigma_y{}^2$ denotes the variance of X and Y respectively. Constant values $c_1 = (k_1 l)^2$ and $c_2 = (k_2 l)^2$ where $k_1 = 0.01$, $k_2 = 0.03$ and l is the dynamic length of pixel value with default value 255.

Table 3. Embedding Distortion Measurement Results

Cover image	Secret Image size	PSNR (dB)	SSIM	Average PSNR	Average SSIM
Baboon	32×32	61.42	0.9996	58.01	0.99942
	50×50	60.18	0.9995		
	80×80	56.23	0.9994		
	100×100	56.18	0.9993		
	110×110	56.07	0.9993		
Flowers	32×32	61.31	0.9997	58.40	0.99942
	50×50	60.48	0.9995		
	80×80	56.88	0.9994		
	100×100	56.68	0.9993		
	110×110	56.67	0.9992		
Peppers	32×32	61.68	0.9996	60.16	0.99932
	50×50	61.68	0.9994		
	80×80	60.79	0.9993		
	100×100	58.31	0.9992		
	110×110	58.32	0.9991		

From the above table we can conclude that the embedding rate vary with the type of cover image used.

3.3 Embedding Efficiency

Embedding efficiency defined in section 2.1.3 is an important metric especially when non-linear embedding is done by considering the texture of cover image because the visual distortion will be comparatively more in the smoother region than that of the textured region.

Table 4. Embedding Efficiency Results

Cover Image	Secret Image Size	Embedding Efficiency(bits per change)	Average Embedding Efficiency
Baboon	32×32	2.5840	2.5854
	50×50	2.5898	
	80×80	2.5900	
	100×100	2.5714	
	110×110	2.5919	
Peppers	32×32	2.9949	2.9946
	50×50	2.9955	
	80×80	2.9964	
	100×100	2.9961	
	110×110	2.9903	
Fishing Boat	32×32	2.7359	2.7262
	50×50	2.7250	
	80×80	2.7231	
	100×100	2.7124	
	110×110	2.7349	

The result in the above table reveals that the embedding efficiency changes with the different types of cover images selected. The textured images show relatively higher embedding efficiency. Embedding of at least 2.5 bits is done for change of every single bit in the cover image.

3.4 Histogram Analysis

The cover image and stego image along with their respective histograms is shown in figure 3. The histograms are having negligible difference due to the lesser frequency in bit change through Matrix encoding technique for secret embedding.

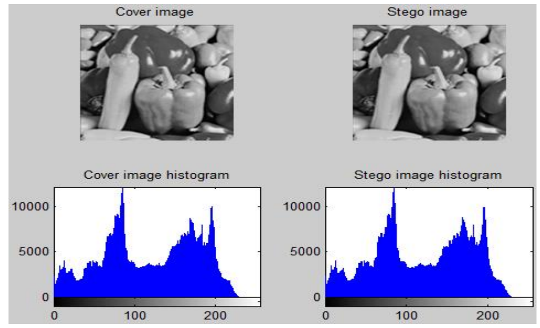

Figure 3. Cover image and stego image with histograms

3.5 Chi-Square (χ2) Analysis :

For checking the statistical imperceptibility, Chi-Square (χ2) test was conducted for the proposed technique [1]. The aim of this test is to check the existence of the degree of randomness of the image pixels even after secret embedding [9]. The result of the test is shown below in figure 4. The probability of detecting the existence of secret information inside the cover image is 0.035 revealing the acceptable disturbance in the random distribution of the pixels. This also reveals that acceptable *Pair of Values (PoVs)* [9] is created by the algorithm.

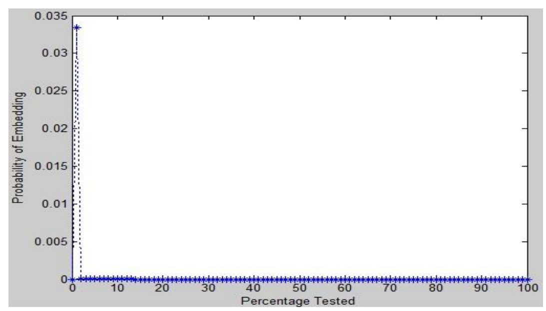

Figure 4. Steganalysis with Chi-Square (χ2) Test

4. COMPARATIVE ANALYSIS

For comparative analysis, the experimental configuration was kept same as that specified in section 3.

4.1 Stego image Fidelity based comparison

The Stego image Fidelity based comparison is with the existing algorithms like JSteg and LSBR .The comparison result is shown graphically in figure 5.

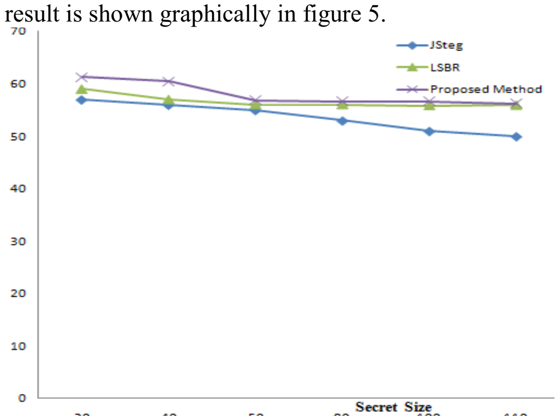

Figure 5. PSNR based comparative results

The result shown by the figure 5 reveals that the proposed method yields PSNR better than JSteg and LSBR.

4.2 Embedding efficiency based comparison

The embedding efficiency based comparison for the proposed method is done with Ratnakirti Roy et al. [12]. Both methods make use of variable rate matrix encoding technique for payload embedding considering smoother and textured regions (identified block-wise). The former technique is implemented in the frequency domain and block-wise standard deviation is used for region identification whereas the latter one in the spatial domain with block-wise entropy as a basis for region identification.

Table 5. Embedding Efficiency based comparison

Cover image	Secret image size	Embedding efficiency (Bits per change)	
		Ratnakirti Roy et. al.[12]	**Proposed Technique**
Baboon	80X80	2	2.59
	100X100	2	2.57
Peppers	80X80	2.10	2.99
	100X100	2.10	2.99
Jet	80X80	2.22	2.83
	100X100	2.28	2.71

4.3 Embedding capacity based comparison

Embedding capacity of a steganographic system implies the amount of data that can be effectively hidden within a selected cover medium by a steganography algorithm. The comparison of the proposed method with few of the existing algorithms in frequency domain is shown in the table below.

Table 6. Embedding capacity based comparison

Algorithm	Capacity	
JSteg	< 1bpnc	bpnc- Bits per non-zero DCT coefficient;
OutGuess	0.4 bpnc	bpsc- Bits per singular value coefficient;
Proposed Technique	(0.43 - 0.66) bpc	bpc- Bits per coefficient

4 CONCLUSION

The proposed method is a DWT based image steganography utilizing the standard deviation of the higher frequency components block wise to identify the smoother as well as textured region of the cover image. Embedding in the higher frequency components in a random fashion leads to an increase of stego image quality and is imperceptible to Human Visual System. An additional security is enforced by scrambling the payload prior to embedding and providing decryption key embedded along with the secret information.

Future work will focus on enhancing the existing method to provide better capacity with the same level of security.

5. REFERENCES

[1] P.E Greenwood, M.S. Nikulin, "A guide to chi-squared testing". Wiley, New York, 1996, ISBN 0-471-55779- X.

[2] Gabriel Peterson, "Arnold's cat map survey", Math 45-Linear Algebra, Fall 1997, pp.1-7

[3] Ron Crandall, "Some Notes on Steganography", Posted on Steganography Mailing List, 1998. Source: http://www.dia.unisa.it/~ads/corso-security/www /CORSO- 203/steganografia/LINKS%20LOCALI/ matrix-encoding.pdf

[4] A. Westfield, "F5-A Steganography Algorithm: High capacity despite better steganalysis", Proc. 4th International Workshop on Information Hiding, vol. 2137, pp. 289-302, Springer, 2001.

[5] Manish Mahajan, Dr. Navdeep Kaur , "Adaptive : A survey of Recent Statistical Aware Steganography Techniques", I. J. Computer and Information Security, Vol.4, No.10, pp.76-92, September 2012

[6] M. Youssef, A.A. Elfarag, R. Raouf, "A Multi-Level Information Hiding Integrating Wavelet-based Texture Analysis of Block Partition Difference Images", 29th National Radio Science Conference, Egypt, 2012, pp203-210.

[7] T.V. Ananthan, B. Mohan, R. Lakshmanan, Image block based Steganography and Encryption System for Business Applications", International Journal of Advanced Technology & Engineering Research, 2012, pp.58-63.

[8] Sarkar, Narayan C Debnath, "Evaluating Image Steganography Techniques: Future Research Challenges", International Conference on Computing and Telecommunications ComManTel 2013) [IEEE], pp.309 –314, January 21 - 24, 2013

[9] Shabnam Samima, Ratnakirti Roy, Suvamoy Changder. Secure Key Based Image Realization Steganography in Image Information Processing (ICIIP), 2013 IEEE Second International Conference on 01/2013 pp.377-382

[10] Ratnakirti Roy, Anirban Sarkar, Suvamoy Changder, "Chaos based Edge Adaptive Image Steganography", 1st International Conference on Computational Intelligence: Modeling, Techniques and Applications, University of Kalyani, September 2013.

[11] Suvarna Patel, Gajendra Singh Chandel, Performance Analysis of Steganography Based on 5-Wavelet Families by 4 Levels –DWT", International Journal of Advance Research in Computer Science and Management Studies, ISSN: 2321-7782 Volume 1, Issue 7, December 2013

[12] Ratnakirti Roy, Suvamoy Changder, "Image Steganography with Block Entropy based Segmentation and Variable Rate Embedding", Business and Information Management (ICBIM), 2014 2nd International Conference on 2014/1/9)[IEEE] pp. 75-80

[13] "Pairing Function", Source: http://mathworld.wolfram.com /Pairing function.html