ISSN No. 0976-5697

Volume 3, No. 4, July- August 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Authentication Technique in Frequency Domain through Daubechies Transformation (ATFDD)

Madhumita Sengupta*
Department of Computer Science and Engineering,
University of Kalyani, Kalyani, Nadia-741235,
West Bengal, India
madhumita.sngpt@gmail.com

J. K. Mandal
Department of Computer Science and Engineering,
University of Kalyani, Kalyani, Nadia-741235,
West Bengal, India
jkm.cse@gmail.com

Abstract: In this paper a Daubechies based steganography in frequency domain termed as ATFDD has been proposed where the cover image is transformed into the time domain signal through Daubechies forward transformation, resulting four sub-image components as, "Low resolution", "Horizontal orientation", "Vertical orientation" and "Diagonal orientation". Secret message/image bits stream in varying positions are embedded in 3rd coefficient of every sub image, along with delicate adjustment followed by reverse transformation to generate embedded/encrypted image. The decoding is done through the reverse procedure. The experimental results against statistical and visual attack has been computed and compared with the existing technique like Yuancheng Li's method, Region-Based method and SCDFT in terms of Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Image Fidelity (IF) analysis, which shows better performances in ATFDD.

Keywords: Steganography; Daubechies; Wavelet; ATFDD; Mean Square Error (MSE); Peak Signal to Noise Ratio (PSNR); Image fidelity (IF);

I. INTRODUCTION

With a rapid development of networking, digital data flow becomes very common whereas ownership protection of digital data becomes that much difficult. Watermarking is a technique of embedding secret message/image in a visible or invisible form, by which we can protect ownership of digital data and provide copyright protection. Secret message transmission is also a process of data exchange through invisible watermarks.

Watermarking can be divided into two major sub domains based algorithms, spatial domain [1] and frequency domain [2, 3]. In case spatial domain a single digital image pixel represents three intensity value of RGB light in color images or single intensity value of light in gray scale images. Thus change in single pixel reflects the single change in intensity. But in frequency domain in case of 2 x 2 mask, single change in coefficient value will reflects the change in four spatial pixel intensity values. And in case of 4 x 4 mask, single change in coefficient value will reflect changes in 16 pixel intensity values. Further stated watermarking algorithm in spatial domain can able to embed big amount of information then algorithm in frequency domain but algorithm of frequency domain having strong anti-attack ability then spatial domain.

Various works have already been done on image watermarking in spatial as well as in frequency domain. In spatial domain starting from LSB manipulation by Chandramouli et al. [4] to entropy based technique by Pavan et al. [5] and N. N. EL-Emam [6] for detecting the suitable areas in the image where data can be embedded with minimum distortion. In STMDF [1] where after embedding image fidelity is readjusted by shuffling the bits of higher order. In frequency domain Chin-Chen Chang in the year 2007 proposed reversible hiding in DCT-based compressed images [7] in this scheme, the two successive zero coefficients of the medium-frequency components in each block are used to hide the secret data, and the scheme modifies the quantization table to maintain the quality of the stego-image. WTSIC [1] is DWT based authentication technique where the secret bits are embedded into three separate sub-image components.

This paper proposed a frequency domain based technique termed as ATFDD where the source image is transformed into its corresponding frequency domain and the authenticated messages/image are embedded into the frequency coefficient. Two phase adjustment is made, called Handling and Fidelity adjustment as intermediate steps to keep the deviation of the embedded image minimum. A reverse transform is performed as a final step of embeddings which added an extra layer of security to the process.

Various parametric tests are performed and results obtained are compared with existing techniques like Yuancheng Li's method [10], SCDFT [11], Region-Based method [12], WTSIC [2] and AINCDCT [3] based on Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Image Fidelity (IF) analysis [8] to show a consistent relationship with the quality perceived by the HVS (Human Visual System).

Section II deals with the ATFDD algorithm, section III emphasis on Daubechies transformation technique, section IV outlined the detail work of ATFDD algorithm; section V analyzed the results of proposed work and comparisons with existing technique. Conclusions are drawn in section VI and references are cited at end.

II. THE SCHEME

ATFDD is divided into five phases. A 4 x 4 mask/window of the original image passes through Daubechies forward transformation to generate frequency coefficients in sliding window based row major order which

converts the image from spatial domain to frequency domain as elaborated in section III. Embedding phases embed 't' number of secret bits into 3rd coefficient of every sub images. A delicate handling has been done followed by an adjustment phase to keep the fidelity of the embedded image close to original one. A reverse transformation technique applied to generate setgo-image. This setgo-image is then broadcasted through transmission channel. The overall schematic representation of ATFDD embedding technique is shown in figure 1.

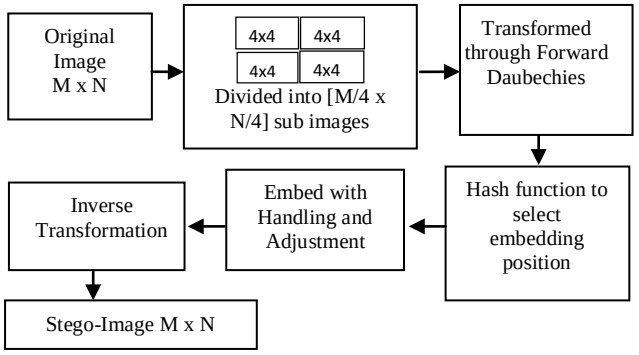


Figure: 1 Schematic diagram of embedding technique of ATFDD

Figure: 2 At the receiving end the embedded image passes through forward Daubechies transformation. 3rd coefficient of every sub images is taken for extraction of 't' number of bits, to regenerate the secret message/image. The schematic representation of the authentication procedure is given in figure 2.

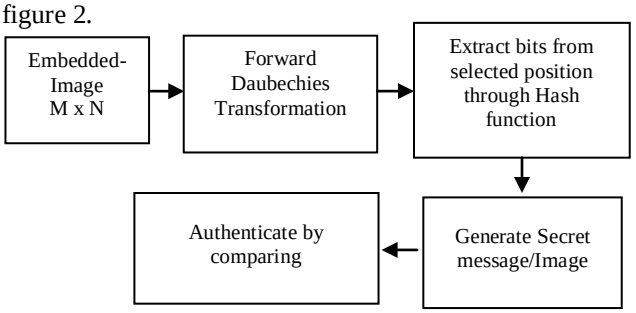


Figure: 2 Schematic representation of authentication process of ATFDD

III. DAUBECHIES TRANSFORMATION TECHNIQUE

Every transformation technique comes with two sets of operation, forward transformation with a counterpart of inverse transformation. Forward transform converts spatial domain image intensity values to frequency domain coefficients and inverse transform generates back to spatial domain values from frequency coefficients.

Daubechies wavelet has no explicit function expression. The scaling functions and wavelet functions are defined by the following two equations eq1 and eq2 respectively.

$$\emptyset(t) = \sum_{n=-\infty}^{+\infty} h[n].\emptyset(2t-n) \quad (1)$$

$$\psi(t) = \sum_{n=-\infty}^{+\infty} g[n].\emptyset(2t-n) \quad (2)$$

Where h[n] is a sequence of lowpass impulse response filter coefficients and g[n] is a sequence of highpass impulse response filter coefficients.

Let us take a sample of image represented in 4x4 matrix shown in figure 3.a. For forward Daubechies transform, image matrix needs to be multiplied by row transformation matrix followed by column transformation matrix as shown in figure 3.b and 3.d respectively. The matrix shown in figure 3.e is the resultant of forward Daubechies coefficients denoted as F_d. These coefficients can be grouped in four subbands similarly as in Haar wavelets [13].

In an inverse Daubechies transform the matrix F_d needs to be multiplied by column transformation matrix followed by row transformation matrix. The resultant will be the original image taken in figure 3.a because of lossless Daubechies transform.

The complete calculation of the forward and inverse Daubechies transformation technique is given in figure 3 and figure 4 respectively. Where the row transformation matrix and the column transformation matrix uses the coefficient value as

H_0 = (1+SQRT (3))/ (4*SQRT (2)), H_1 = (3+SQRT (3))/ (4*SQRT (2)), H_2 = (3-SQRT (3))/ (4*SQRT (2)), H_3= (1-SQRT (3))/ (4*SQRT (2)), $G_0 = H_3$, $G_1 = -H_2$, $G_2 = H_1$ and $G_3 = -H_0$.

71	75	91	149
95	60	62	87
113	109	49	60
72	93	56	53

(a) Original image

H_0	H_2	G_0	G_2
H_1	H_3	G_1	G_3
H_2	H_0	G_2	G_0
H_3	H_1	G_3	G_1

(b) Row transformation matrix

98.144163	174.799055	-21.837357	-22.003264
98.710746	116.249715	-15.896299	22.967367
148.973564	85.078780	-27.043434	22.093686
118.262698	75.484560	-8.914987	-3.812935

(c) Row transformed matrix

148.060273	190.967664	-28.752079	14.031407
180.101626	128.370229	-23.356079	-0.423255
32.676005	-13.963911	-11.927624	18.022673
-11.361019	62.148439	-5.092042	-31.502968

(e) Column transformed matrix (F_d)

H_0	H_1	H_2	H_3
H_2	H_3	H_0	H_1
G_0	G_1	G_2	G_3
G_2	G_3	G_0	G_1

(d) Column transformation matrix

Figure: 3 Daubechies 4x4 forward transformation calculation

148.060273	190.967664	-28.752079	14.031407
180.101626	128.370229	-23.356079	-0.423255
32.676005	-13.963911	-11.927624	18.022673
-11.361019	62.148439	-5.092042	-31.502968

(a) Forward Daubechies transformed matrix (F_d)

H_0	H_1	H_2	H_3
H_2	H_3	H_0	H_1
G_0	G_1	G_2	G_3
G_2	G_3	G_0	G_1

(b) Column transformation matrix

70.9999	74.9997	90.9999	148.9998
95.0000	59.9999	62.0000	86.9999
112.9999	108.9998	48.9999	59.9998
72.0000	92.9999	56.0000	52.9999

(e) Row transformed matrix

H_0	H_1	H_2	H_3
H_2	H_3	H_0	H_1
G_0	G_1	G_2	G_3
G_2	G_3	G_0	G_1

(d) Row transformation matrix

129.77019	98.80962	99.54976	151.32816
118.42429	139.48504	82.88381	95.45179
29.27118	23.11029	-11.72988	-14.18871
-17.25053	-1.19019	27.28612	62.97886

(c) Column transformed matrix

Figure: 4 Daubechies 4x4 Inverse transformation calculation

IV. PROPOSED METHOD

Proposed ATFDD is divided into five processes, Transformation process is describe in section A. Section B deals with watermarking embedding process. Handling process is elaborated in section C, Section D deals with Fidelity adjustment process. Section E elaborated the decoding process to authenticate the image by extracting secret message/image.

A. Transformation:

ATFDD technique works on grayscale images where the images are represented as 8bit integer for intensity value ranges from 0 to 255 with a row as height and column as width. Before transform, whole image is subdivided into 4 x 4 window mask in a row major order figure 5.a. Each mask separately passes through forward Daubechies transform technique as elaborated in section III to generate frequency coefficients figure 5.b.

B. Embedding:

Embedding in consecutive positions will reflect back huge change in spatial domain after inverse transformation. Thus alternate positions are selected in forward Daubechies transformed matrix such as $F_d[1][0]$, $F_d[1][2]$, $F_d[3][0]$ and $F_d[3][2]$ as shown in figure 5.b by color. Embedding is done only in the integer portion of the coefficient value, after embedding the floating portion of the coefficient is added back, 'F_{de}' is shown in figure 5.c, and the binary representations are as follows.

$F_d[1][0]$ = 10110100 on embedding two secret bits it become $F_{de}[1][0]$ = 10111100, likewise $F_d[1][2]$ = 00010111 become $F_{de}[1][2]$ = 00001111, $F_d[3][0]$ = 00001011 become $F_{de}[3][0]$ = 00000110 and $F_d[3][2]$ = 00000101 become $F_{de}[3][2]$ = 00000010.

Algorithm:

Input: F_d forward Daubechies transformed matrix & secret bit stream.

Output: F_{de} forward Daubechies transformed matrix embedded with secret message & S_{ij} Stego-image.

Method: Integer portion of 3rd coefficient of every band in F_d is embedded with secret bits in a random position from LSB based on hash function to generate S_{ij} stego-image.

Step 1: $F_d[1][0]$, $F_d[1][2]$, $F_d[3][0]$ and $F_d[3][2]$ are fetched.

Step 2: Integer portion and floating portions are separated.

integer[i][j]=abs((int) $F_d[i][j]$);

floating[i][j]= $F_d[i][j]$-(int) $F_d[i][j]$;

Step 3: integer[i][j] are embedded with 'B' number of secret bits up to 'P^{th}' position based on the hash function.

newposition=((row*column)+B)%P

Step 4: Embedded integer[i][j] then added with the floating portion to generate F_{de} matrix shown in figure 5.c.

Step 5: Embedding process ends with inverse transformation to generate stego image 'S' as shown in figure 5.d.

With this we calculate difference table 'D' as shown in figure 5.e, by the equation no 3.

$$D_{ij} = I_{ij} - S_{ij} \quad (3)$$

71	75	91	149
95	60	62	87
113	109	49	60
72	93	56	53

(a) Original Image (I)

148.060273	190.967664	-28.752079	14.031407
180.101626	128.370229	-23.356079	-0.423255
32.676005	-13.963911	-11.927624	18.022673
-11.361019	62.148439	-5.092042	-31.502968

(b) Forward Daubechies Transformed Matrix (F_d)

148.060273	190.967664	-28.752079	14.031407
188.101626	128.370229	-15.356079	-0.423255
32.676005	-13.963911	-11.927624	18.022673
-6.361019	62.148439	2.907958	-31.502968

(c) Embedded with secret message (F_{de})

73	78	99	144
94	58	57	90
114	111	52	58
74	97	61	50

(d) After inverse transformation (S)

-2	-3	-8	5
1	2	5	-3
-1	-2	-3	2
-2	-4	-5	3

(e) Difference matrix (D)

Figure: 5 Watermark Embedding calculation

C. Handling:

To minimize the difference between original image and the stego-image the handling process has been applied.

In handling process difference is calculated first between original and the setgo-image then corresponding embedded frequency coefficients are manipulated to minimize the difference.

Algorithm:

Input: I_{ij} Original image, S_{ij} Stego- image and F_{de} forward Daubechies transformed matrix embedded with secret message.

Output: F_{ed} Embedded coefficients after handeling.

Method: Tries to minimize the changes in stego-image as compared to original/cover image by calculating the difference table D_{ij}.

Step 1: Calculate the fresh difference table $D_{ij} = I_{ij} - S_{ij}$.

Step 2: Check for nonzero elements except D_{01}, D_{02}, D_{31} and D_{32}, because this positions are linked with $F_{de}[1][0]$, $F_{de}[1][2]$, $F_{de}[3][0]$ and $F_{de}[3][2]$ which we used to hide secret bits and cannot be tampared. Add the non zero value of 'D_{ij}' table into the corresponding position of '$F_{de}[x][y]$' to generate $F_{ed}[x][y]$.

The corresponding positions are linked in a fashion where:

i. 1st row of D_{ij} is corresponding to 4th row of $F_{ed}[x][y]$.
ii. 2nd row of D_{ij} is corresponding to 1st row of $F_{ed}[x][y]$.
iii. 3rd row of D_{ij} is corresponding to 3rd row of $F_{ed}[x][y]$.
iv. 4th row of D_{ij} is corresponding to 2nd row of $F_{ed}[x][y]$.
v. The same combination is applied for column also 1st column of D_{ij} is corresponding to 4th column of $F_{ed}[x][y]$, 2nd column with 1st column, 3rd with 3rd column and 4th column with the 2nd coulmn of $F_{ed}[x][y]$.

Step 3: Recalculate the difference table 'D'. Run step 2 unelse until no non zero elements exist excpet D_{01}, D_{02}, D_{31} and D_{32}, or it cross the maximum loop limit.

The first and the last step of calculation of the same are shown in figure 6. The first step is shown in figure 6.a and 6.b. And the last step of handling process is shown in figure 6.c and 6.d.

148.060273	190.967664	-28.752079	14.031407
188.101626	128.370229	-15.356079	-4.423255
32.676005	-13.963911	-11.927624	18.022673
-6.361019	62.148439	2.907958	-33.502968

(a) Embedded image (F_{ed}) after step 1 of Handling

0	-4	-9	4
0	3	5	-2
0	-3	-3	1
0	-5	-6	2

(b) Difference table after step 1.

151.060273	193.967664	-23.7520789	12.0314069
188.101626	133.370229	-15.3560792	-5.42325503
27.6760053	-16.9639106	-16.9276237	21.0226734
-6.36101888	67.1484388	2.90795773	-34.5029681

(c) Embedded image (F_{ed}) after handling process over

0	-6	-16	0
0	0	0	0
0	0	0	0
0	-7	-10	0

(d) Difference table after handling process over.

Figure: 6 Calculation for handling process

D. Fidelity Adjustment:

On handling, F_{ed} is ready for inverse transformation to generate stego-image. Sometime this stego-image at receiver end when transformed through Daubechies forward transformation technique generates false frequency coefficients. To overcome, an additional fidelity adjustment is made before inverse transformation for proper decoding.

The situation is depicted in figure 7, the embedded handled image (figure 7.a), if passes through inverse transform generates stego-image (figure 7.b), that on receiver end when passes through forward transformation generates coefficient value as shown in figure 7.c. the coefficient value at position [1][0] changes from 188 to 189, this change will generate noise in the secret message/image. Thus fidelity adjustment is obligatory.

151.060273	193.967664	-23.7520789	12.0314069
188.101626	133.370229	-15.3560792	-5.42325503
27.6760053	-16.9639106	-16.9276237	21.0226734
-6.36101888	67.1484388	2.90795773	-34.5029681

(a) Embedded image after handling process over

71	81	107	149
95	60	62	87
113	109	49	60
72	100	66	53

(b) Stego Image after inverse transform

151.168523	193.816941	-23.816986	12.236858
189.000000	133.210533	-15.000000	-5.447451
27.651804	-16.758465	-16.767934	20.923398
-6.000000	67.083531	3.000000	-34.611219

(c) Forward Daubechies transformation at receiver side

Figure: 7 Calculation without fidelity adjustment

In figure 8 embedded handled image is adjusted by LSB manipulation of $F_{de}[0][0]$ coordinate and rounding off the embedded positions (figure 8.a). Then it passes through inverse transform to generate stego-image (figure 8.b), that on receiver end when passes through forward transform, generates coefficient value as shown in figure 8.c.

152.060273	193.967664	-23.752079	12.031407
188.000000	133.370229	-15.000000	-5.423255
27.676005	-16.963911	-16.927624	21.022673
-6.000000	67.148439	3.000000	-34.502968

(a) Embedded handled image after fidelity adjustment

71	81	107	149
95	61	63	87
113	109	49	60
72	99	66	53

(b) Stego Image after inverse transform

152.164036	194.095947	-23.333733	11.662098
188.000000	133.273032	-15.000000	-4.964198
27.818069	32.375058	-17.013447	20.827404
-6.000000	32.755979	3.000000	-34.423719

(c) Forward Daubechies transformation at receiver side

Figure: 8 Calculation after fidelity adjustment

E. Decoding:

Stego-image on transmission through unsecured channel received by the receiver needs to be authenticated. The algorithm for decoding is given here.

In decoding setgo-image generates frequency coefficients through forward Daubechies transform on a 4x4 mask in a row major order. 3rd coefficients of every band pass through hash function to regenerates the embedded bits and the secret message/image.

Algorithm:

Input: Stego Image (S_{ij})

Output: Secret message/image

Method: 3rd coefficient of every band per 4x4 mask of stego-image S_{ij} after FDT regenerates the secret message/image based on hash function.

Step 1: Stego image 'S' passes through 4 x 4 mask/window in a row major order.

Step 2: Every mask passes through forward Daubechies transform to generate frequency components.

Step 3: 3rd coefficient of every 2 x 2 mask are fetched and based of the hash function secret bits are extracted.

Step 4: 't' number of secret bits extracted will generate the secret message/image.

Step 5: Compare the secret image to authenticate the setgo-image/embedded image in the receiver end.

The stego-image can be generated in three ways based on the requirement and computational complexity. In case the requirement says low computational complexity with only authentication the embedded image F_{de} without any handling or adjustment can pass inverse transformation to generate stego-image (S_1). Secondly if embedded handled image F_{ed} pass through inverse transformation to generate stego-image (S_2). And the third case when the stego-image (S_3) generates after fidelity adjustment phases. Three separate stego images with extracted secret image are shown in figure 9.

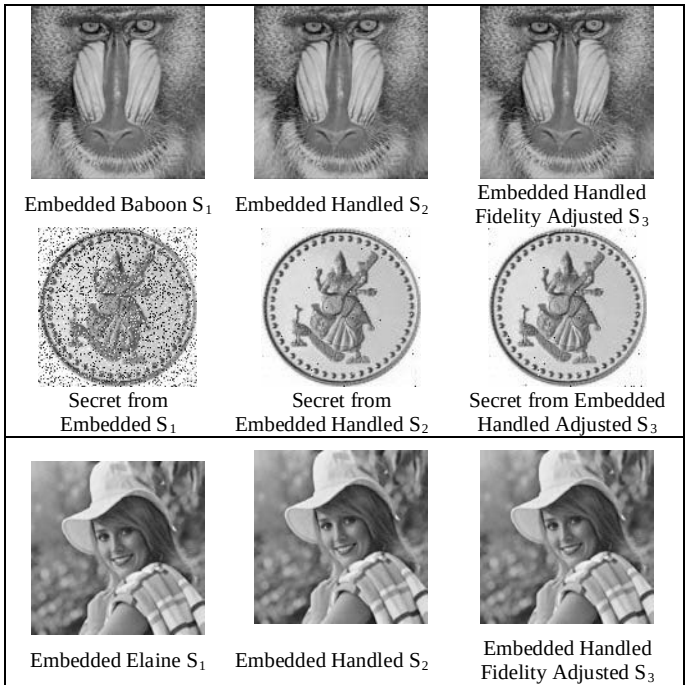
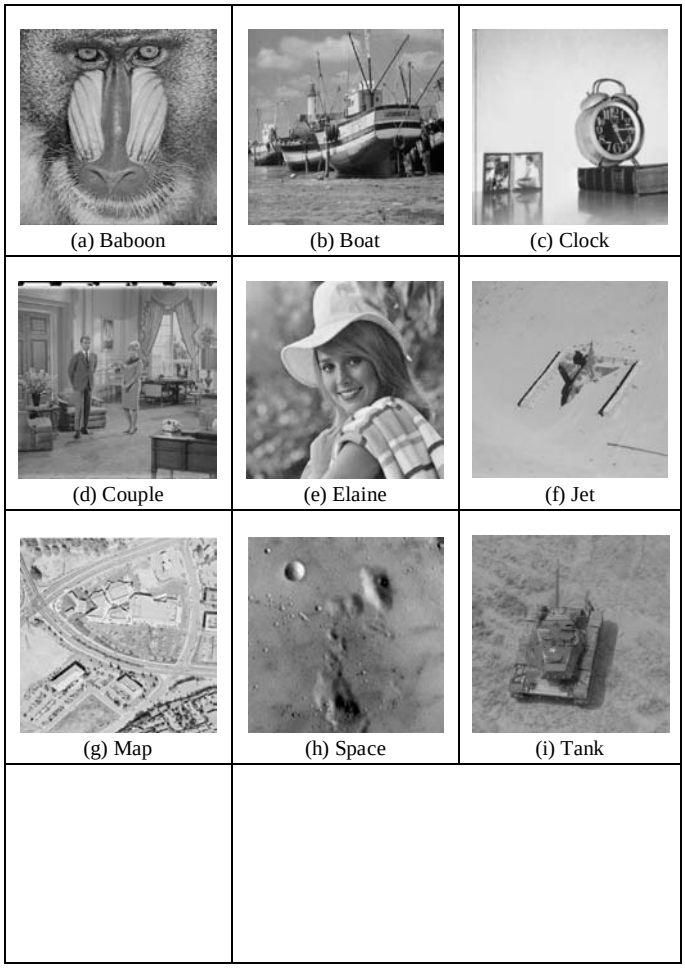


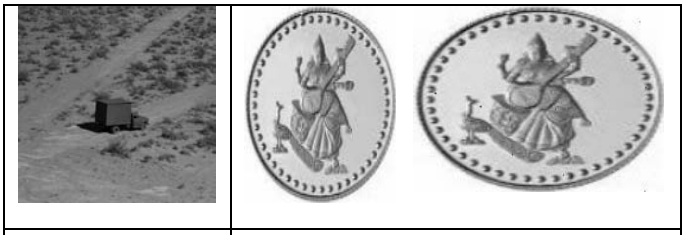
Figure: 9 Example of Embedded, embedded handled and embedded handled adjusted image, with extracted secret image of every stego-image.

V. RESULTS AND DISCUSSIONS

This section deals with the results of computation on embedding hidden data. Ten PGM [9] images have been taken and ATFDD is applied on each. All cover images are 512 x 512 in dimension and a gold coin (k) of 128 x 128 and 256 x 128 is used as authenticating image. The images are given in fig. 10.

Average calculation of MSE, PSNR and IF for ten images with secret image 128 x 128 in dimension, MSE for Stego-image S_1 is 0.656077, for S_2 is 1.149835 and for S_3 is 0.692791 and that of PSNR for stego-image S_1 is 50.110370, for S_2 is 47.664337 and for S_3 is 49.836951 and image fidelity for stego-image S_1 is 0.999968, for S_2 is 0.999943 and for S_3 is 0.999966 as given in table 1, table 2 and table 3. From figure 9 and the calculation shown in table 1-3, it is clear that the embedded image without any adjustment is not at all feasible, due to huge amount of noise in extracted secret image.





(j) Truck (k) secret image

Figure: 10 Cover images of dimension 512 x 512 and secret image of dimension 128 x 128 and 256 x 128

Table:1 Data on applying ATFDD over 10 Images with original image verses stego-image S_1, secret image of dimension 128 x 128

Cover Image 512 x 512	*MSE*	*PSNR*	*IF*
(a) Baboon	0.262009	53.947647	0.999986
(b) Boat	0.715179	49.586653	0.999962
(c) Clock	0.766239	49.287160	0.999980
(d) Couple	0.685902	49.768185	0.999959
(e) Elaine	0.711601	49.608437	0.999966
(f) Jet	0.667007	49.889497	0.999979
(g) Map	0.686550	49.764081	0.999980
(h) Space	0.675129	49.836936	0.999960
(i) Tank	0.702309	49.665523	0.999961
(j) Truck	0.688847	49.749578	0.999943
Average	***0.656077***	***50.110370***	***0.999968***

Table: 2 Data on applying ATFDD over 10 Images with original image verses stego-image S_2, secret image of dimension 128 x 128

Cover Image 512 x 512	*MSE*	*PSNR*	*IF*
(a) Baboon	0.476444	51.350683	0.999974
(b) Boat	1.239418	47.198626	0.999935
(c) Clock	1.364662	46.780552	0.999964
(d) Couple	1.201420	47.333856	0.999928
(e) Elaine	1.238247	47.202731	0.999940
(f) Jet	1.164917	47.467854	0.999963
(g) Map	1.198395	47.344805	0.999965
(h) Space	1.190872	47.372152	0.999930
(i) Tank	1.224697	47.250517	0.999933
(j) Truck	1.199280	47.341598	0.999901
Average	***1.149835***	***47.664337***	***0.999943***

Table: 3 Data on applying ATFDD over 10 Images with original images verses stego-image S_3, secret image of dimension 128 x 128

Cover Image 512 x 512	*MSE*	*PSNR*	*IF*
(a) Baboon	0.318607	53.098246	0.999983
(b) Boat	0.747833	49.392756	0.999961
(c) Clock	0.800278	49.098396	0.999979
(d) Couple	0.721790	49.546693	0.999957
(e) Elaine	0.744541	49.411916	0.999964
(f) Jet	0.701042	49.673362	0.999977
(g) Map	0.722511	49.542357	0.999979
(h) Space	0.710747	49.613655	0.999958
(i) Tank	0.736221	49.460720	0.999960
(j) Truck	0.724335	49.531411	0.999940
Average	***0.692791***	***49.836951***	***0.999966***

Average of MSE, PSNR and IF for ten images with secret image 256 x 128 in dimension have been obtained. MSE for stego-image S_2 is 24.743158 and for S_3 is 14.046639 that of PSNR for stego-image S_2 is 34.241409 and for S_3 is 36.700684. Image fidelity for stego-image S_2 is 0.998790 and S_3 is 0.999313 as given in table IV and V.

Table: 4 Data of ATFDD over 10 Images with original image verses stego-image S_2, secret image of dimension 256 x 128

Cover Image 512 x 512	*MSE*	*PSNR*	*IF*
(a) Baboon	19.617638	35.204337	0.998946
(b) Boat	22.909039	34.530735	0.998791
(c) Clock	32.021030	33.076451	0.999152
(d) Couple	24.974838	34.155777	0.998506
(e) Elaine	22.307346	34.646325	0.998920
(f) Jet	28.417187	33.594993	0.999088
(g) Map	22.661404	34.577936	0.999335
(h) Space	29.030556	33.502250	0.998292
(i) Tank	22.272560	34.653102	0.998776
(j) Truck	23.219986	34.472184	0.998089
Average	***24.743158***	***34.241409***	***0.998790***

Table: 5 Data of ATFDD over 10 Images with original image verses stego-image S_3, secret image of dimension 256 x 128

Cover Image 512 x 512	*MSE*	*PSNR*	*IF*
(a) Baboon	11.130989	37.665466	0.999402
(b) Boat	13.031605	36.980825	0.999312
(c) Clock	18.207264	35.528357	0.999518
(d) Couple	14.157825	36.620838	0.999153
(e) Elaine	12.651691	37.109318	0.999387
(f) Jet	16.133610	36.053488	0.999482
(g) Map	12.843517	37.043964	0.999623
(h) Space	16.503735	35.954981	0.999029
(i) Tank	12.635933	37.114730	0.999306
(j) Truck	13.170216	36.934875	0.998916
Average	***14.046639***	***36.700684***	***0.999313***

A comparative study has been made between Yuancheng Li's Method[10], SCDFT [11], Region-Based method [12], WTSIC [2] and AINCDCT [3] with proposed ATFDD in terms of mean square error, peak signal to noise ratio and image fidelity. Comparison is done on average bases, table VI and figure 11 display the comparision results in details. Comparison shows that ATFDD generates optimized result with respect of PSNR verses bit per byte. With bpB of 0.5 proposed technique ATFDD stands on 49.84 dB better than other compared technique. Where as with bpB of 1.0 proposed ATFDD stand on 36.70 dB of PSNR, that is optimized result with compare to bpB.

Table: 6 Comparison of ATFDD with existing technique

Technique	*Capacity (bytes)*	*Size of cover image*	*bpB (Bits per bytes)*	*PSNR in dB*
Yuancheng Li's Method[10]	1089	257 * 257	0.13	28.68
SCDFT[11]	3840	512 * 512	0.12	30.10
Region-Based[12]	16384	512 * 512	0.5	40.79
WTSIC[2]	16384	512 * 512	0.5	42.04
AINCDCT[3]	16384	512 * 512	0.5	46.34
ATFDD	16384	512 * 512	0.5	49.84
ATFDD	32768	512 * 512	1.0	36.70

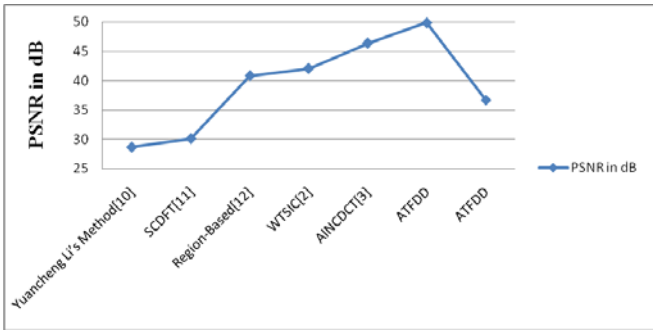


Figure: 11 Graphical representation of comparision tableVI.

VI. CONCLUSIONS

ATFDD is a Daubechies transformation based invisible watermarking process in frequency domain, for ownership verification, copyright protection or secret message transmission. Authenticity is incorporated by embedding secret data in each frequency mask of carrier image in randomly generated position. Handling process reduces the change in stego-image generated after inverse transformation by adding or subtracting small values to/from coefficients not used for hiding secret bits. This process also helps to increases image fidelity. Fidelity adjustment process swap LSB in such a manner that generated stego-image in receiver end on forward transformation can able to regenerate the secret bits intact. The watermarked image in this algorithm is very difficult to detect due to unknown insertion position of the 'authenticating image bits' in the carrier image. Hence, the proposed technique ATFDD is quite secured from almost any possible attacks.

VII. ACKNOWLEDGMENT

The authors express deep sense of gratuity towards the Dept of CSE University of Kalyani where the computational resources are used for the work and the PURSE scheme of DST, Govt. of India.

VIII.REFERENCES

[1] J. K. Mandal, Madhumita Sengupta, "Steganographic Technique Based on Minimum Deviation of Fidelity (STMDF)", IEEE, Second International Conference on Emerging Applications of Information Technology (EAIT 2011), Print ISBN: 978-1-4244-9683-9, DOI: 10.1109/EAIT.2011.24, pp- 298 – 301, 2011.

[2] J. K. Mandal, Madhumita Sengupta, "Authentication /Secret Message Transformation Through Wavelet Transform based Subband Image Coding (WTSIC)", IEEE, International Symposium on Electronic System Design 2010, pp 225-229, ISBN 978-0-7695-4294-2, Bhubaneswar, India, DOI 10.1109/ISED.2010.50. Print ISBN: 978-1-4244-8979-4, Dec, 20th -22nd, 2010.

[3] Madhumita Sengupta, J. K. Mandal, "Authentication of Images through Non Convoluted DCT (AINCDCT)", first International Conference on Communication and Industrial Application (ICCIA 2011), Organised by: Narula Institute of Technology, JIS, IEEE, pp- 1-4, DOI: 10.1109/ICCIndA.2011.6146672, 2011.

[4] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," Proc. of ICIP, Thissaloniki, pp. 1019-1022, Greece, 2001.

[5] S. Pavan, S. Gangadharpalli and V. Sridhar, "Multivariate entropy detector based hybrid image registration algorithm," IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Philadelphia, Pennsylvania, USA, pp. 18-23, March 2005.

[6] Nameer N. EL-Emam, "Hiding a large Amount of data with High Security Using Steganography Algorithm," Journal of Computer Science ISSN 1549-3636, vol. 3, no. 4, pp. 223-232, 2007.

[7] Chang Chin-Chen, et al. Reversible hiding in DCT-based compressed images, Information Sciences, ISSN: 0020-0255, doi:10.1016/j.ins.2007.02.019 Volume 177, Issue 13, pp- 2768-2786, 2007.

[8] Kutter M , Petitcolas F A P. A fair benchmark for image watermarking systems, Electronic Imaging 99. Security and Watermarking of Multimedia Contents. vol. 3657. Sans Jose, CA, USA. January 1999. The International Society for Optical Engineering. pp 226-239 http://www.petitcolas.net/fabien/publications/ei99-benchmark.pdf. (Last accessed on 25th March, 2012).

[9] Allan G. Weber, The USC-SIPI Image Database: Version 5, Original release: October 1997, Signal and Image Processing Institute, University of Southern California, Department of Electrical Engineering. http://sipi.usc.edu/database/ (Last accessed on 25th May, 2012).

[10] Li Yuancheng, Xiaolei Wang, "A watermarking method combined with Radon transformand 2D-wavelet transform", IEEE, Proceedings of the 7th World Congress on Intelligent Control and Automation, June 25 - 27, Chongqing, China, 2008.

[11] T. T. Tsui, X. –P. Zhang, and D. Androutsos, Color Image Watermarking Usimg Multidimensional Fourier Transfomation, IEEE Trans. on Info. Forensics and Security, vol. 3, no. 1, pp. 16-28, 2008.

[12] A. Nikolaidis, I. Pitas, "Region-Based Image Watermarking", IEEE Transactions on Image Processing, Vol. 10, NO. 11, pp. 1721-1740, November 2001.

[13] Madhumita Sengupta, J. K. Mandal, Nabin Ghoshal, "An authetication technique in frequency domain through wavelet transform (ATFDWT), Advances in Modelling Signal Processing and Pattern Recognition(AMSE), vol-54, Issue 2, Published Journal:2011-Vol.54N^{o}1-2, 2011.