Multimed Tools Appl (2018) 77:13721–13752
DOI 10.1007/s11042-017-4986-1

An IWT based blind and robust image watermarking scheme using secret key matrix

Kshiramani Naik[1] · Saswati Trivedy[1] · Arup Kumar Pal[1]

Received: 18 October 2016 / Revised: 20 April 2017 / Accepted: 25 June 2017 /
Published online: 1 September 2017

Abstract In this paper, the authors have proposed a binary watermark embedding approach for protecting the copyright ownership of the gray-scale images. The proposed watermark embedding process is realized in integer wavelet transform (IWT) domain to defend the robustness property. Instead of inserting the watermark bits directly in the coefficients of cover media, an indirect embedding mechanism is proposed with the reference to a logistic map based secret key matrix which enhance the secrecy of the proposed embedding approach. Initially, the approximate sub band of the IWT transformed cover image is selected with the intention to embed the watermark. Later, a secret key matrix of size corresponding to the approximate sub band of the cover image is formed using the logistic map with secret parameters. During the watermark embedding process, the approximate sub band is modified indirectly with reference to the secret key matrix and a proposed division table. The scheme is tested on a set of standard images and satisfactory results are achieved. In addition, the proposed schemes is also able to extract the watermark information in blind manner. Also, the scheme is comparable with some other related schemes. Finally, the proposed watermarking scheme is able to survive the watermark even after performing certain types of image manipulation attacks.

Keywords Blind watermarking · Copyright protection · Integer wavelet transform · Logistic map · Robust watermarking

✉ Kshiramani Naik
kshiramani@gmail.com

Saswati Trivedy
saswatialo12@gmail.com

Arup Kumar Pal
arupkrpal@gmail.com

[1] Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, Jharkhand 826004, India

1 Introduction

The extensive evolution of digital technology facilitates the multimedia data to be transmitted and distributed in digital format over the Internet. As digital data are easily exposed to illegal possession, duplication and dissemination over the Internet, it has become an essential to think about the copyright protection, ownership verification, and tamper-resistance of digital data during their applications. Digital watermarking method is one of the widely used solution for detecting the illegal manipulation occurred in digital data. In digital watermarking method, the information related to the digital data is embedded or hidden in the digital data itself such that the authenticity and integrity can be verified by extracting or detecting the embedded information. The embedded information is termed as watermark. The digital data that contain watermark is termed as cover media. Depending upon the cover media, the watermarking schemes are classified as image, video or audio watermarking.

Depending upon the specific goal, the watermarking method can be categorized into robust, semi-fragile and fragile watermarking. The robust watermarking method is generally considered for copyright protection of the digital data [17, 18]. In robust watermarking scheme, the existence of secret information can be known but it is hard to remove/manipulate the secret information [8]. So in copy right protection scheme robust watermarking is preferred.

Depending upon the embedding domain, again the robust watermarking schemes can be divided into two categories, i.e., spatial-domain schemes and frequency-domain schemes. In Spatial domain, watermark is added directly by modifying pixel values of the cover image. Several robust watermarking scheme in spatial domain have been devised by researchers [10, 15–17, 21]. Embedding the watermark into the cover image in spatial domain is a straight forward method, which has the advantages of low computational complexity and easy implementation. However, the most serious problem of spatial domains is the weakness of robustness i.e. spatial domain watermarking algorithms is able to resist some limited number of attacks. In transform domain, the watermark is embedded by modulating the coefficients of the transformed cover image. However in case of frequency-domain scheme, the computational cost is higher than the ones based on spatial domain, more information can be embedded and better robustness against the common image processing attacks can be survived. The main advantages of using the frequency domain methods are that they can easily be adapted to lossy compression systems, which have the ability to embed data in the compressed representations, and have ability to reveal the watermark even from the modified watermarked image [9, 20]. The transform domain based watermarking schemes can be implemented through various transformation tools such as discrete cosine transform (DCT) [22], discrete wavelet transform (DWT) [23], Discrete Fourier transform(DFT) [12], Integer wavelet transform(IWT) [4], Singular Value Decomposition(SVD) [11] etc.

Various robust image watermarking schemes based on transform domain have shown their effectiveness in image data protection. In [24], Thabit et al. proposed another watermarking scheme based on Slantlet transform matrix to transform small blocks of the original image and hiding the watermark bits by modifying the mean values of the carrier subbands. Fazli et al. [7] proposed a robust watermarking based on a combination of DWT, DCT, and SVD domains. This paper mainly focuses on the geometric attacks. To address this goal, the host image is divided into four non overlapping rectangular segments called sub-images and then watermark is independently embedded into each of them, using the hybrid scheme. The redundancy reduces effect of cropping attack. Moreover, in order to correct main geometric attacks, such as rotation, translation, and affine translation, an inventional synchronization technique is utilized to recover the geometrically attacked image via detection of desired image corners. A binary image in the first experiment and some 1D binary random sequences with different lengths in the next experiments are used as watermarks. Weng et al. proposed another method based on integer Haar wavelet transform (IHWT), which utilizes block selection and difference expansion (DE) (or histogram shifting (HS)) [28]. IHWT has the characteristic that the average of a block remains unchanged before and after watermark embedding. Hence, this invariability can be used for determining whether a block is located in a smooth region or not. In [19], Pal et al. proposed a robust and blind watermarking scheme based on Discrete Cosine Transform (DCT) for protecting the copyright ownership of the digital images. In this work a binary watermark is embedded into the block based DCT transformed cover image by modifying the middle significant AC coefficients using repetition code. The proposed approach ensures the protection of copyright information even in compressed form of the watermarked image. In [13], Kumsawat et al., the watermark has been embeded into the DMT coefficients using multiwavelet tree techniques. Digital watermarking algorithm using integer wavelet transform(IWT) have received wide range of attention in the recent years due to the property that it can map integer to integer without the rounding error, and can obtain good imperceptibility . There are many IWT-based watermarking schemes that have been proposed in recent years. In [25], Verma et al. designed robust digital watermarking scheme using 3-level lifting wavelet transform (LWT) with a block selection procedure. Non-overlapping coefficient blocks from the low pass subband are selected after applying LWT and using certain criterion based on minimum coefficient difference and a threshold value. Ansari et al. proposed another watermarking scheme using IWT and SVD (singular value decomposition) based to address false positive problem that are suffered in SVD based watermarking techniques [3]. The properties of IWT and SVD help in achieving high value of robustness. Singular values are used for the watermark embedding. In order to further improve the quality of watermarking, the optimization of scaling factor (mixing ratio) is performed with the help of artificial bee colony (ABC) algorithm. In [26], Wang et al. proposed an efficient integer transform based reversible watermarking scheme. In this paper, Tian's difference expansion (DE) technique can be reformulated as an integer transform. Then, a generalized integer transform and a payload-dependent location map are constructed to extend the DE technique to the pixel blocks of arbitrary length. In [5], Bohra et al. proposed a technique for robust watermarking of images based on lifting-based integer wavelet transform. The proposed scheme, along with its robustness has got the capability of blind self–authentication of the watermarked images. This paper also utilizes histogram modification to avoid overflow/underflow problem. In [6], the 2-level IWT based watermarking scheme for embedding the compressed version of the binary watermark logo has been developed for robust watermarking. In this paper, the source document image is divided into empty and non-empty segments depending on the absence or presence of the information. Watermarking is applied for non-empty segments. A binary watermark logo is compressed using binary block coding technique of appropriate block-size. IWT is applied on the non-empty segment of the source document image. LL-sub–band of the transformed image is subdivided into blocks of uniform size and compressed watermark bit stream is embedded into it. In [14], Lingamgunta et al. proposed a reversible watermarking based on IWT. The proposed algorithm hides the data and the bookkeeping information in the high frequency subbands of CDF (2,2) integer wavelet coefficients whose magnitudes are similar to a certain predefined threshold. Histogram modification is applied as a preprocessing to prevent overflow/underflow. The embedding technique is based on the parent–child structure of the transformed coefficients called "quadruple wavelet tree" (QWT). In this paper, we develop an invisible robust watermarking scheme based on 1–level IWT domain. Robustness and imperceptibility are strongly achieved in the proposed method through the characteristics of IWT. Before watermark embedding, the cover image is transformed through IWT. We have selected the approximate sub band of the integr wavelet transformed cover image for the watermark insertion. Generally watermark embedding only in the approximate sub band reduce the chance of removing or destroying the watermark from the watermarked image. In [27], Wang et al. proposed a 3-level wavelet based intelligent watermarking scheme using particle swarm optimization (PSO) technique. In this scheme, the high sub–bands of the DWT transformed cover image are considered. The coefficents to contains watermark bits are selected randomly from the different sub–bands. Ali & Ahn presented a DWT–SVD based watermarking algorithm where self-adaptive differential evolution algorithm is used during embedding process [1]. Another work is presented by Ali et al was in wavelet domain and SVD domain. In this work the low frequency sub–band is selected and divided into blocks. Again the blocks were SVD transformed and the left and right singular vector matrix are used for watermark embedding using artificial bee colony (ABC) algorithm [2].

In some existing schemes, the watermark bits are embedded directly on the selected coefficients of the cover image. But in the proposed watermarking scheme, instead of embedding the watermark bits directly to the coefficients of the cover image, an indirect method corresponds to a division method is utilized. However watermark embedding only in the low subband increase the chance of removing or destroying the watermark with the attempt of tampering of that portion. Although this proposed method utilize the low sub–band of the transformed cover image, robustness and imperceptibility are strongly achieved through the proposed embedding method and the characteristics of IWT. Also to increase watermarking security, a generated key matrix using logistic is utilized. The intention of the proposed method is to improve the robustness and invisibility of the watermarked image and this scheme is suitable for extract the watermark information in a blind manner.

The rest of the paper is organized as follows. Section 2 describes the related fundamentals for better understanding of the proposed method. Section 3 contains the details of the proposed method. The experimental results and discussion are in Section 4. Section 5 contains the conclusion of the work done in this paper.

2 Preliminaries

2.1 Integer to integer wavelet transform

Due to the multi-resolution characteristic, the conventional wavelet transform is very popular in signal and image processing field. Also it is a very good computational tool to reduce the digital image files with higher compression ratios which helps to storing images using less memory and for transmitting images faster and more reliably. In the Fig. 1, LL subband represents the approximation part of the image and LH, HH, HL represents the detail part of the image. But the conventional Wavelet transform is not suitable for truly lossless coding because it gives floating point results for any input sequence which generally create problem for reconstruction of the exact signal or image. Due to this problems, a generalized version of conventional wavelet transform, Integer to Integer Wavelet Transform (IWT) is very popular for lossless coding method. It is also known as the second generation of the wavelet transform. The IWT was introduced by Sweldens (1998). The IWT inherits the multi-resolution characteristics of the conventional wavelet transform and that can map integer input sequence to integer output sequence by rounding off the values of wavelet

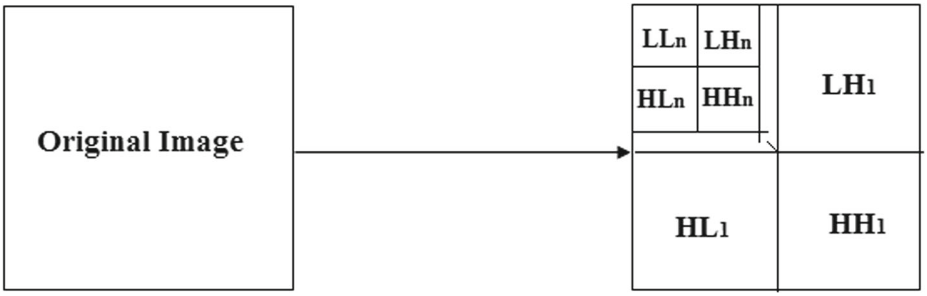


Fig. 1 n-level wavelet transform

transformation. Thus as compared to floating point operation they need less storage space and the implementation is faster than the conventional wavelet coefficients. The IWT was constructed by means of lifting scheme. The schematic diagram of the lifting scheme is shown in Fig. 2:

With a lifting scheme, the forward transform is calculated in three steps.

Split The input sequence is S_j decomposed into an even sequence and odd sequence.

$$\left(Even_{j-1}, Odd_{j-1}\right) \leftarrow Split\left(S_j\right)$$

Where

$$Even_{j-1} = \left\{Even_{j-1,k} = S_{j,2k}\right\}$$
$$Odd_{j-1} = \left\{Odd_{j-1,k} = S_{j,2k+1}\right\}$$

Predict The numbers from one sequence (generally the odd sequence, Odd_{j-1})is predicted on the basis of the other sequence(generally the even sequence, $Even_{j-1}$)by the use of correlation between them.The difference, D_{j-1}between the actual value$\left(Odd_{j-1}\right)$and the predicted value, $P\ \left(Even_{j-1}\right)$ becomes the wavelet coefficients. The operation of

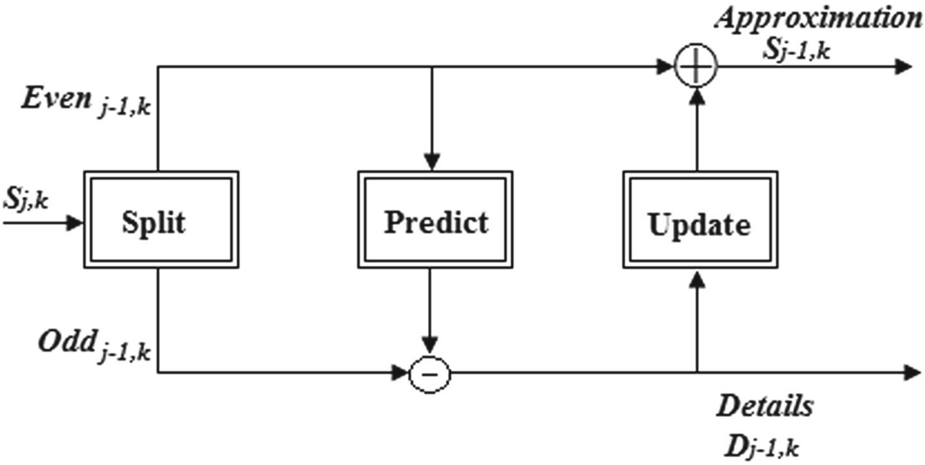


Fig. 2 Forward integer wavelet transform

obtaining the differences from the prediction is called the lifting step.

$$D_{j-1} = Odd_{j-1} - P\left(Even_{j-1}\right)$$

Where

$$P_k\left(Even_{j-1}\right) = \frac{\left(Even_{j-1,k} + Even_{j-1,k+1}\right)}{2} = \frac{\left(S_{j,2k} + S_{j,2k+1}\right)}{2}$$

Update The update step follows the prediction step, where the even values are updated from the input even samples and the updated odd samples. They become the scaling coefficients which will be passed on to the next stage of transform. This is the second lifting step.

$$S_{j-1} = Even_{j-1} + U\left(D_{j-1}\right)$$

Where U is the updated operator and defined as follows:

$$U_k\left(D_{j-1}\right) = \frac{D_{j-1,k}}{2} = \frac{\left(D_{j-1,k-1} + D_{j-1,k}\right)}{4} + \frac{1}{2}$$

The corresponding inverse transform of IWT is calculated as follows:

$$Even_{j-1} \leftarrow S_{j-1} - U\left(D_{j-1}\right)$$
$$Odd_{j-1} \leftarrow D_{j-1} + P\left(Even_{j-1}\right)$$
$$S_j \leftarrow Merge\left(Even_{j-1}, Odd_{j-1}\right)$$

In order to achieve multilevel decomposition,the approximation part, $\left(S_{j-1,k}\right)$is further decomposed into approximate and detail parts using split, predict and update stage and we get $S_{j,2k}$ and $D_{j,2k}$. This process can be repeated n number of times, where $n = log_2(N)$ for the input image of size $N \times N$.

2.2 Logistic mapping

Chaotic signals are a kind of pseudorandom, irreversible and dynamical signals generated by deterministic nonlinear equations, which process good characteristics of pseudorandom sequences .The definition is

$$x_{n+1} = \mu x_n(1 - x_n) \tag{1}$$

Where $x_n \in (0, 1)$is the state of the system for (n = *0, 1, 2,..*) and $\mu \in [0, 1]$. For different values of parameter, μ, the logistic sequence shows different characteristics. For $x \in (0, 1)$ and $\mu \in [3.57, 4]$, the logistic map shows the chaotic behaviour.

3 Proposed method

This section describes some motivating factors that are used to design a robust and blind watermarking method. In the proposed approach, the authors have considered various test images of size $N \times N$ as cover images(C) and a binary logo (W) of size $w \times w$ as watermark. To embed the watermark, a region is selected by applying 1–level IWT on the cover image. As already mentioned, the IWT is an efficient and rapid lifting wavelet transform and its properties are best suited to enhance the robustness and preserve the imperceptibility. Due to this, IWT is very popular in case of digital image watermarking. Also the authors have applied IWT on the cover image to decompose the cover image into four sub-bands, named LL, HL, LH and HH. After 1–level IWT transformation of C, the approximation part i.e. LL sub-band with size $N1 \times N1$ ($N1 = \frac{N}{2}$) is used for watermark embedding. In this paper, LL sub-band is termed as CA. Before embedding the watermark, the CA part is decomposed into non-overlapping blocks of size $n \times n$. In the proposed method, the authors consider the block based watermark embedding procedure. To preserve the watermark bit unchanged, a single bit is embedded repeatedly in the selected coefficients of a particular block. Before embedding process some binary key vectors are generated using a division method (explained in the Key Generation phase) for watermark bit embedding. These binary key vectors are generated from a key matrix generated by utilizing the chaotic logistic map. This binary key also utilized to select the coefficients to be embedded in each block. The detail procedure of the proposed method is carried out through three phases as follows:

3.1 Pre-processing phase

This phase again comprises of two parts. In the first part a key matrix of same size as CA is generated from the chaotic logistic map. Here, instead of taking the direct values of initial condition and system parameter, the authors have considers the calculated initial value and system parameter to generate the logistic sequence. The key matrix is used to generate the binary key vectors those are used for watermark embedding. In the next part the detail procedure of required binary key vectors is described.

Key generation using logistic map

Step 1: Generate a random binary key sequence of l bits long, where $l = t^2$.

Step 2: Divide the key sequence into blocks of 16-bit each.

$$K = \sigma_1 \sigma_2 .....\sigma_{16}$$

Step 3: To calculate the initial condition (x_0) and the system parameter (μ) of the chaotic logistic map, the ASCII key sequence is used. The intermediate values, γ_1 and γ_2 are used to calculate the initial condition and γ_3, γ_4 are used to calculate the system parameter.

$$x_0 = \mathrm{mod}\left((\gamma_1 + \gamma_2), 2\right)$$

$$\mu = \mathrm{mod}\left(\left(3.999 + (\gamma_3 + \gamma_4)/1000\right), 1\right)$$

Where

$$\begin{aligned}\gamma_1 &= (\sigma_1 \parallel \sigma_2 \parallel \sigma_3)_{10}/2^{23}, \\ \gamma_2 &= (\sigma_4 \parallel \sigma_5 \parallel \sigma_6)_{10}/2^{23}, \\ \gamma_3 &= (\sigma_7 \parallel \sigma_8 \parallel \sigma_9)_{10}/2^{23}, \\ \gamma_4 &= (\sigma_{10} \parallel \sigma_{11} \parallel \sigma_{12})_{10}/2^{23}\end{aligned}$$

Step 4: Generate a chaotic sequence, a of length l by using the (1).

Step 5: Reshape the generated chaotic sequence into a square matrix of size $t \times t$.

Step 6: The matrix is concatenated in raster scan order to generate the matrix K_EM of size CA i.e $N1 \times N1$.

Binary key vectors generation for watermark embedding

This phase generates the binary keys for the individual block of the decomposed CA. The detail procedure is described as follows:

Step 1: Calculate the difference matrix, D using CA sub–band and K_EM.

$$D = |CA - K\_EM|$$

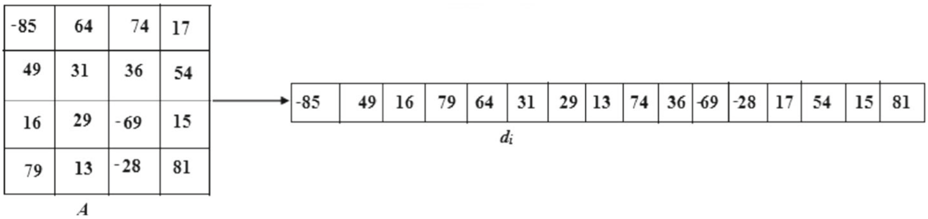
Step 2: Divide the difference matrix, D into the non overlapping blocks, b of size, $n \times n$.

$$b = \left\{ b_1, b_2, \ldots.., b_{\frac{N1^2}{n^2}} \right\}$$

Step 3: Convert each block into row vector.

$$d = \left\{ d_{1 \times n^2}, d_{2 \times n^2}, \ldots..., d_{\frac{N1^2}{n^2}} \times n^2 \right\}$$

For Example Suppose A is the one of the decomposed block of difference matrix, D. Then A is converted into row vector as shown below.



-85	64	74	17
49	31	36	54
16	29	-69	15
79	13	-28	81

A

-85	49	16	79	64	31	29	13	74	36	-69	-28	17	54	15	81

d_i

Step 4: Calculate the adjacent differences, Adj_Diff in each row using the following formula

For $i = 1 : 2 : \frac{N1^2}{n^2} - 1$

$$Adj\_diff(i) = abs(d_i - d_{i+1})$$

Binary Value Generation using Division Method

Step 5: Take a range, R with minimum value, min and maximum value, max. As we are considering gray test images so $min = 0$ and $max = 255$.

Step 6: Divide R with r number of divisions to give r slots.

Then

$$R = \{R_1, R_2, \ldots....R_r\}$$

Where $R_1 = min$ and $R_r = max$

Step 7: Then the elements of Binary key vectors, $Bin\_CA(i)$ reference to the individual block matrix of CA can be generated from the R and $Adj\_diff(i)$ as follows:

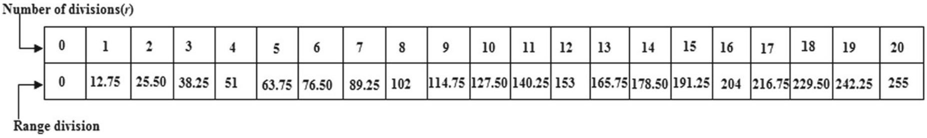
$$Bin\_CA(j) = \begin{cases} 1 & if\ R_k \leq \text{Adj\_diff(j)} \leq R_{k+1} \\ & and \quad \text{mod}\ (k, 2) == 1 \\ \\ 0 & if\ R_k \leq \text{Adj\_diff(j)} \leq R_{k+1} \\ & and \quad \text{mod}\ (k, 2) == 0 \end{cases} \tag{2}$$

Where $k = 1 : r$ and $j = 1 : 2 : n^2$

For example:

Suppose $r = 20$, the range can be divided into different slots as shown in Fig. 3.

The main objective of the generation of the binary key vectors space is to utilize these keys as the reference to the coefficients that are to be modified. The utilization of these reference bits are explained in the next section.



Number of divisions(r)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Range division	0	12.75	25.50	38.25	51	63.75	76.50	89.25	102	114.75	127.50	140.25	153	165.75	178.50	191.25	204	216.75	229.50	242.25	255

Fig. 3 Division of a range

3.2 Watermark embedding phase

This section describes the detail of embedding procedure of watermark of size $w \times w$ to the CA part of size $N1 \times N1$. The main objective is to modify the selected coefficients in such a way that the they fall in the same slot as the watermark bit. The detail procedure is described as follows:

Phase 3: watermark embedding

Step 1: Decompose CA part into non-overlapping blocks of size $n \times n$.

$$b' = \left\{ b'_1, b'_2, \ldots.., b'_{\frac{N1^2}{n^2}} \right\}$$

Step 2: Convert each block into row vectors as follows:

$$d' = \left\{ d'_{1\times n^2}, d'_{2\times n^2}, \ldots..., d'_{\frac{N1^2}{n^2}} \times n^2 \right\}$$

Step 3: Store the watermark bits in a row vector W'.

Step 4: The watermark bits of the vector, W' are embedded in the blocks of CA part according to the value of binary vectors $Bin\_CA(i)$ to generate the watermarked Image WCA as follows:

Algorithm 1 Watermark embedding algorithm

```
for i = 1:$w^2$ do
    for $j = 1 : 2 : \frac{N1^2}{n^2} - 1$ do
        if $W'(i) = 0$ then
            if $Bin\_CA(i, j) == 1$ then
                WCA(i,j) = CA(i,j) + m
            end if
            if $Bin\_CA(i, j) == 0$ then
                WCA(i,j) = CA(i,j)
            end if
        end if
        if $W'(i) == 1$ then
            if $Bin\_CA(i, j) == 0$ then
                WCA(i,j) = CA(i,j)+m
            end if
            if $Bin\_CA(i, j) == 1$ then
                WCA(i,j) = CA(i,j)
            end if
        end if
    end for
end for
```

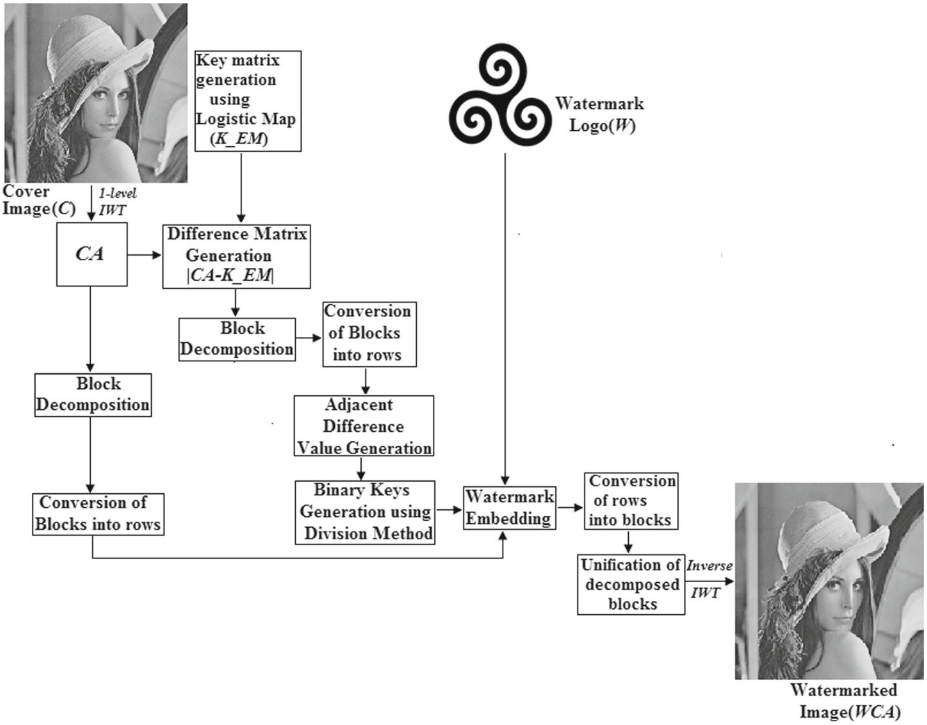

Key matrix generation using Logistic Map (K_EM)
Watermark Logo(W)
Cover Image(C)
1-level IWT
CA
Difference Matrix Generation |CA-K_EM|
Block Decomposition
Conversion of Blocks into rows
Block Decomposition
Adjacent Difference Value Generation
Conversion of Blocks into rows
Binary Keys Generation using Division Method
Watermark Embedding
Conversion of rows into blocks
Unification of decomposed blocks
Inverse IWT
Watermarked Image(WCA)

Fig. 4 Block diagram of the proposed watermark embedding method

The block diagram of the watermark embedding method is shown in Fig. 4.

3.3 Extraction method

At the receiver end the watermark is detected by the intended recipient from the watermarked image using the same secret key that was used in the embedding method. In the proposed method, instead of sending the whole key ,only the initial value of the logistic map (x_0) is need to send at the receiver side to generate the required key. The extraction procedure is just the reverse process of embedding procedure and is given as follows:

Step 1: Generate the key matrix, K_EM from the initial condition(x_0) of the logistic map.

Step 2: Calculate the difference matrix, D' using watermarked image, WCA and K_EM.

$$D' = |WCA - K\_EM|$$

Step 3: Divide the difference matrix, D' into the non overlapping blocks, $b'$of size, 4×4.

$$b' = \left\{b_1{}', b_2{}', ..., b_{4096}{}'\right\}$$

Step 4: Convert each block into row vectors, d'.

$$d' = d_1', d_2', ..., d_{4096}'$$

Step 5: Calculate the adjacent differences, Adj_Diff' in each block of d'.

$$Adj\_diff = d_i{}' - d_{i+1}{}'$$

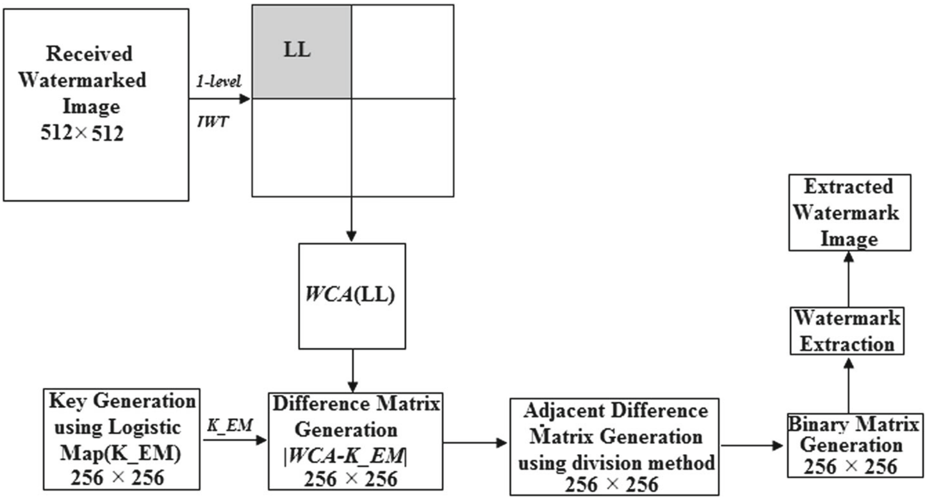

Received Watermarked Image 512×512
1-level IWT
LL
WCA(LL)
Key Generation using Logistic Map(K_EM) 256 × 256
K_EM
Difference Matrix Generation |WCA-K_EM| 256 × 256
Adjacent Difference Matrix Generation using division method 256 × 256
Binary Matrix Generation 256 × 256
Watermark Extraction
Extracted Watermark Image

Fig. 5 Block diagram of the proposed watermark extraction method

Step 6: Generate the corresponding binary matrix of Adj_Diff' i.e. Bin_Adj' using the division method as described above.

Step 7: Calculate the watermarked vectors, WE that contains watermark bits using the following equation:

$$WE = \begin{cases} 0 & if\ Bin\_Adj' = 0 \\ 1 & if\ Bin\_Adj' = 1 \end{cases}$$

Step 8: Extract the watermark bits from WE as follows:

$$WEX(i) = \begin{cases} 0 & if\ \max freq\,(WE\,(i)) = 0 \\ 1 & if\ \max freq\,(WE\,(i)) = 1 \end{cases}$$

Step 9: Reshape WEX into a matrix of size 64 × 64.

The block diagram of the watermark extraction procedure is depicted in Fig. 5.

4 Experimental results

In the proposed scheme, an embedding process using division method (as discussed in the Section 3.1) is employed where the value of the parameter r (r = number of divisions) is given by the user. In this paper, we have considered the value of $r = 20$, 30, 40 and 50. For the result analysis, several images of size 512 × 512, given in Fig. 6 and a binary logo of size 64 × 64, given in Fig. 7 are taken as the cover media and watermark respectively.

The watermark algorithm can be evaluated by considering two important parameters, imperceptibility and robustness (Fig. 8).

Imperceptibility measurement The imperceptibility means that the human visual quality of the host image should not be affected much even after watermark embedding. We have generated different sets of watermarked images according to the value of r and are depicted in Figs. 9, 10, 11, and 12.

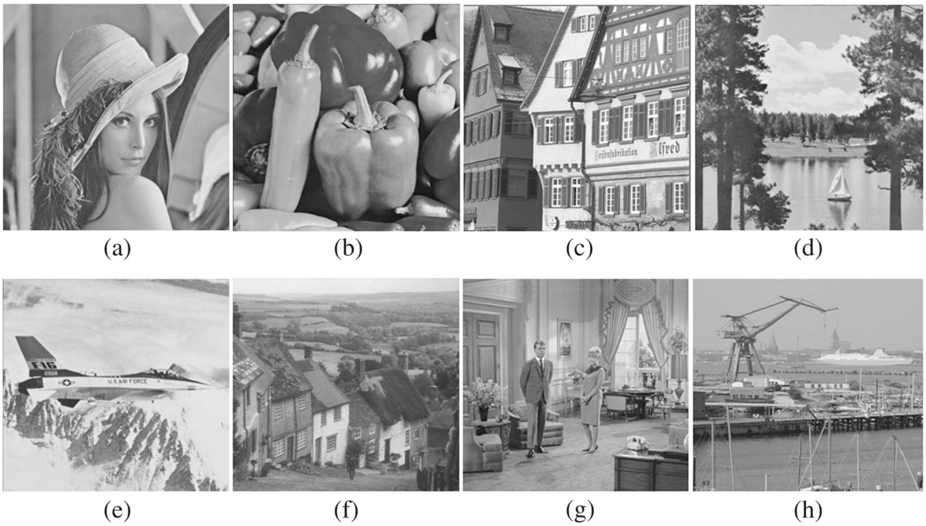


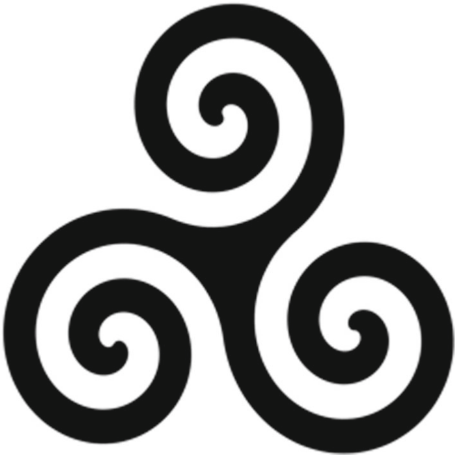
Fig. 6 Original images: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

The corresponding watermarked images with different value of r of the Fig. 6 shown in Figs. 9, 10, 11, and 12. There is very low visual degradation of the watermarked images than original images. Also the histograms of the watermarked images in Fig. 13 are similar to the histogram of original images (Fig. 8) which indicate that the proposed watermark algorithm ensures high degree of fidelity.

An alternation to measure the degree of imperceptibility is Peak Signal to Noise Ratio($PSNR$) and it can be defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$

Fig. 7 Binary logo



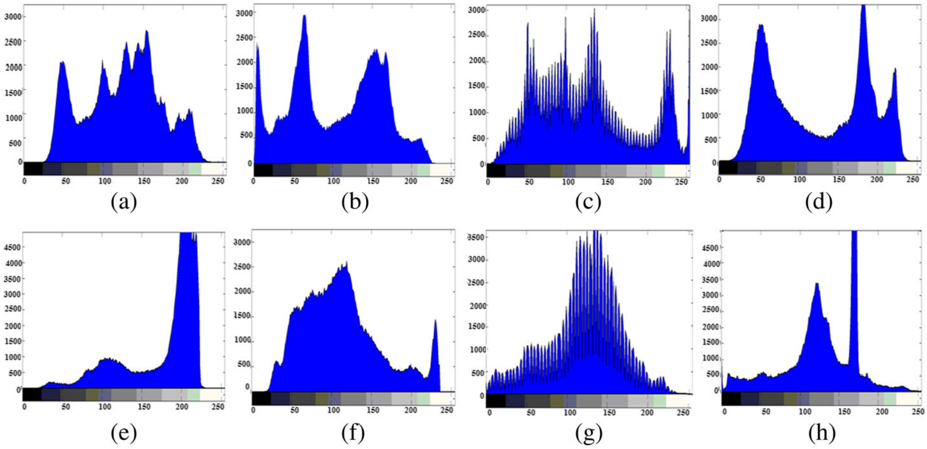


Fig. 8 Histogram of original images: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

Where

$$\mathrm{MSE} = \frac{\sum_{i-1}^{M}\sum_{j-1}^{N} x_{i,j} - \tilde{x}_{i,j}}{M \times N}$$

where $x_{i,j}$ and $\tilde{x}_{i,j}$ denotes the original and encrypted pixel respectively, and the images are of size $M \times N$.

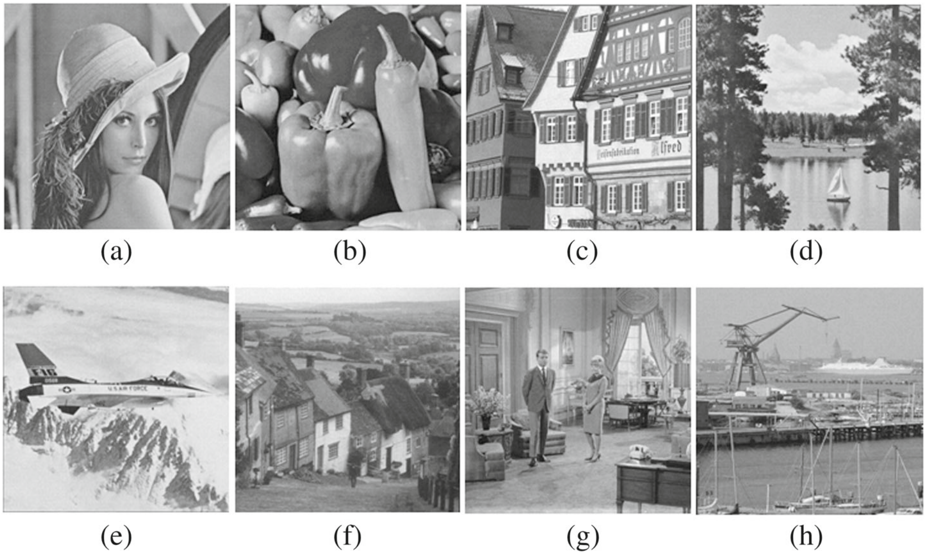
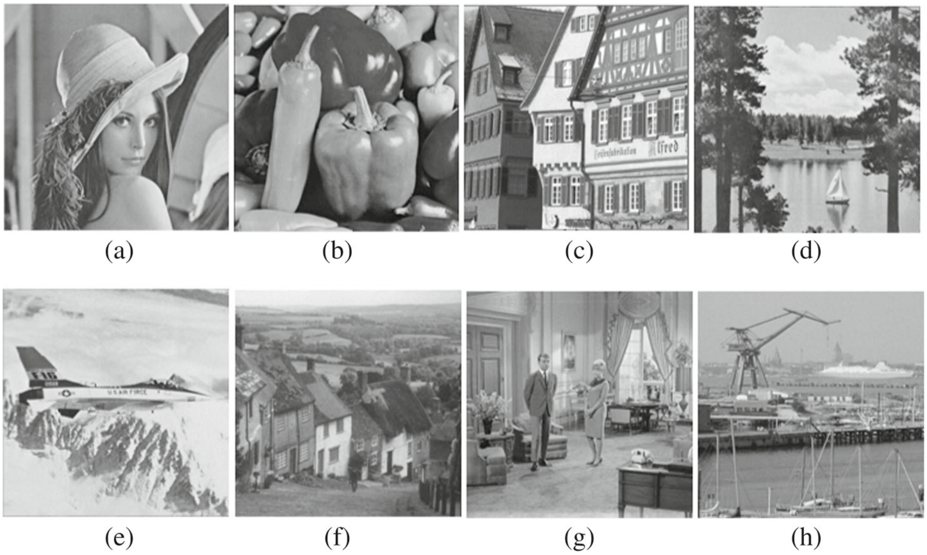


Fig. 9 Watermarked images using 20 divisions: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

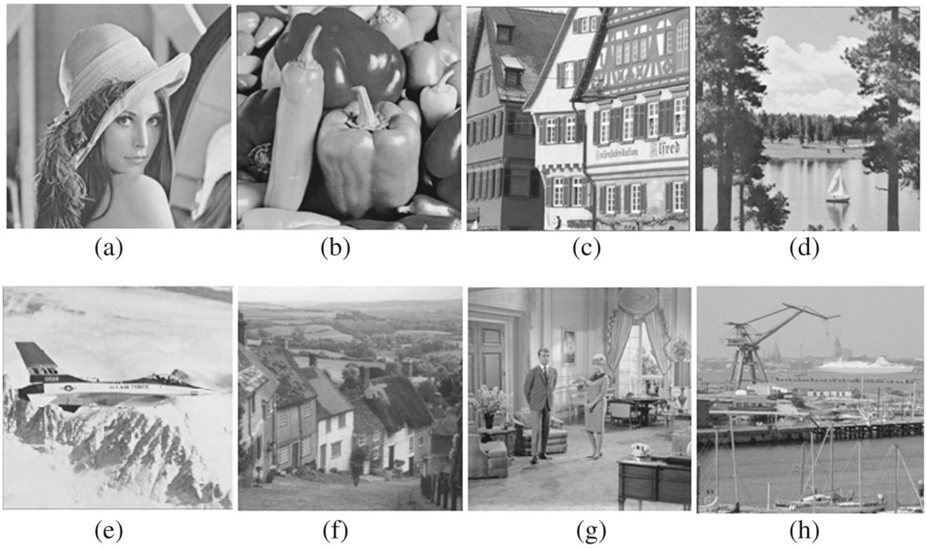


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 10 Watermarked images using 30 divisions: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

A larger $PSNR$ indicate that the watermarked image more closely resembles the original image meaning that the watermark is more imperceptible. Table 1 shows the calculated $PSNR$ values of the different number of division for the proposed watermarking algorithm.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11 Watermarked images using 40 divisions: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

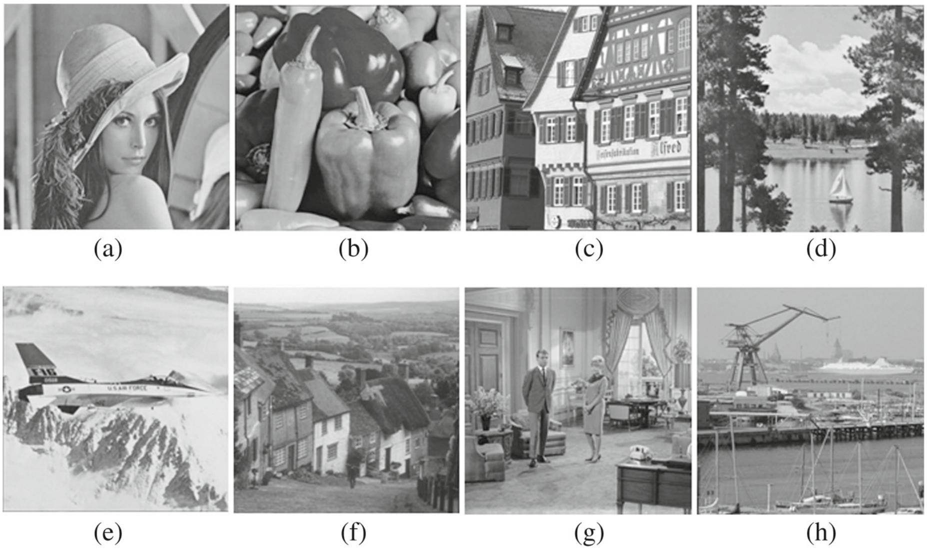


Fig. 12 Watermarked images using 50 divisions: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

From the Table 1 and the Fig. 14 it shows that the $PSNR$ values are higher than 32 dB and also the $PSNR$ values are improving with the increment of number of divisions.

Robustness measurement The robustness of the proposed scheme can be determined in terms of Bit–Error–Rate(BER) and normalized cross correlation(NC) which are measured between the original watermark and extracted watermark (without attack/after applying different types of intended attacks). NC measures the similarities between the original and

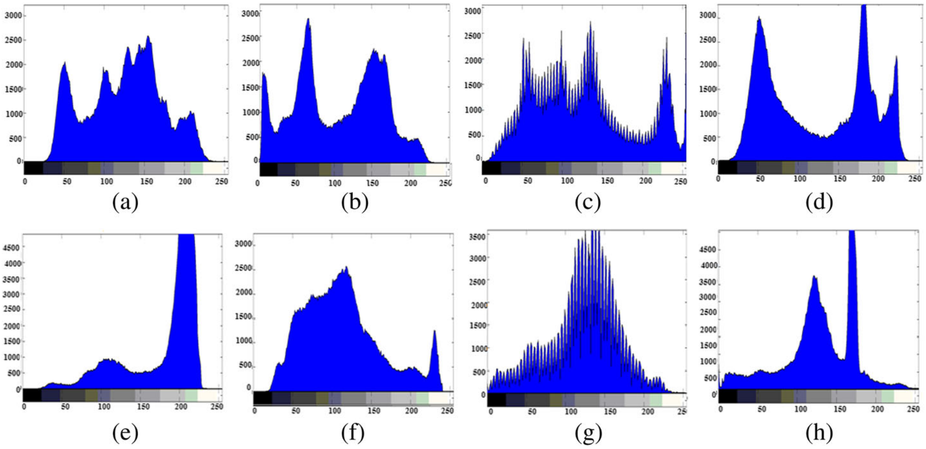


Fig. 13 Histogram of Watermarked Images: **a** Lena **b** Peppers **c** Houses **d** Sailboat **e** Airplane **f** Goldhill **g** Couple **h** Kiel

Table 1 PSNR of the watermarked images with different number of divisions without attack

Image	Using 20 division	Using 30 division	Using 40 division	Using 50 division
Lena	32.7466	36.475	39.0962	40.6873
Peppers	32.8008	36.1867	39.0348	40.7455
House	32.9646	36.3677	39.1705	40.8079
Baboon	32.8328	36.2423	39.0698	40.6771
Tiffany	32.807	36.1754	39.0559	40.6777
Goldhill	32.7808	36.2274	39.057	40.6843
Zelda	32.7901	36.1801	39.0317	40.6968
Kiel	32.753	36.214	39.882	40.6653

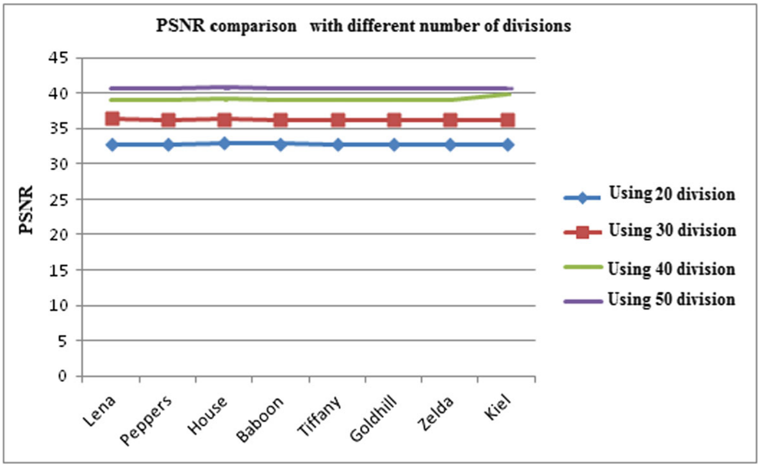


Fig. 14 Imperceptibility comparision with different number of divisions without attack

Table 2 BER and NC of the extracted watermark in different number of division without attack

Image	20 division		30 division		40 division		50 division	
	BER	NC	BER	NC	BER	NC	BER	NC
Lena	0.0413	1	0.0017	1	0.0097	1	0.0004	1
Peppers	0.0012	1	0.0017	1	0.0012	1	0.0004	1
House	0.0068	0.9986	0.0081	0.9993	0.0029	0.9989	0.0015	0.9996
Baboon	0.0032	1	0.0032	1	0.0002	1	0	1
Tiffany	0.0027	1	0.0022	1	0.0007	1	0	1
Goldhill	0.0017	1	0.0015	1	0.0004	1	0.0002	1
Zelda	0.0044	1	0.0012	1	0.0009	1	0	1
Kiel	0.0022	1	0.0034	1	0.0007	1	0	1

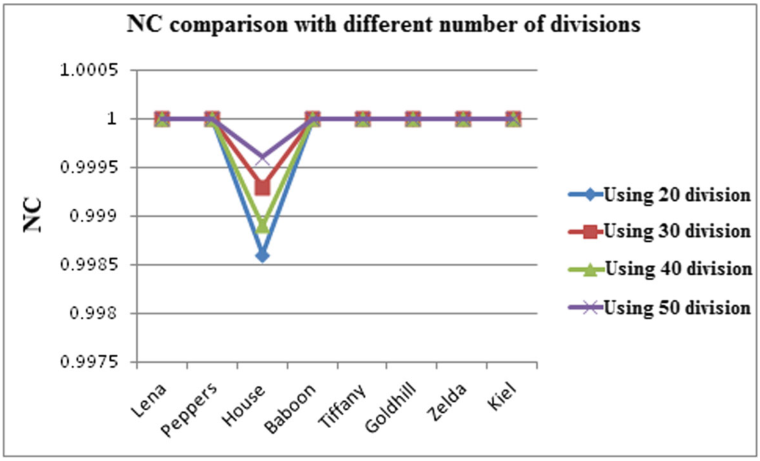


Fig. 15 *NC* comparison with different number of divisions

extracted watermark. For an efficient watermark algorithm the NC values for different test images should be nearly or equal to 1. Generally, $NC \geq 0.75$ is acceptable. The BER values should be very less that means they should be nearly or equal to 0.

The NC can be formulated as follows:

$$NC(w, \bar{w}) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[w(i,j)-\mu_w] \times [\bar{w}(i,j)-\mu_{\bar{w}}]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[w(i,j)-\mu_w]^2 \times \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[\bar{w}(i,j)-\mu_{\bar{w}}]^2}}}$$

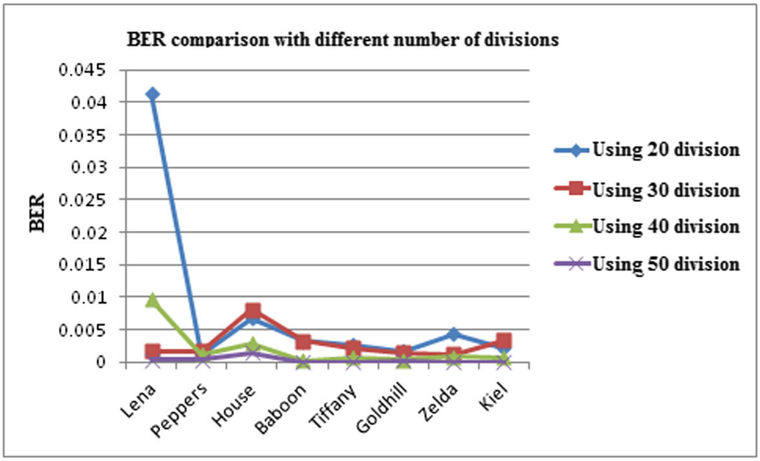


Fig. 16 *BER* comparison with different number of divisions

Table 3 TAF of the watermarked images with different number of divisions without attack

Image	Using 20 division	Using 30 division	Using 40 division	Using 50 division
Lena	0.1709	0.2197	0.0732	0
Peppers	0.1953	0.1709	0.2441	0
House	0.7080	0.6836	0.4395	0.1953
Baboon	0.2930	0.3906	0.1221	0
Tiffany	0.3418	0.1953	0.1953	0.0488
Goldhill	0.1709	0.1465	0.1221	0
Zelda	0.2686	0.2441	0.0977	0
Kiel	0.2197	0.1709	0.0977	0.0244

Where M and N denote the width and the height of the watermark image, $w_{i,j}$ and $\bar{w}_{i,j}$ denote the original watermark pixel and extracted watermark pixel respectively. μ_w and $\mu_{\bar{w}}$ represent the mean of the original watermark and extracted watermark respectively.

The BER can be calculated as follows:

$$BER = \frac{Number\ of\ error\ bits}{Total\ bits\ transmitted} = \frac{Number\ of\ error\ bits\ per\ second}{Data\ rate\ per\ second} \tag{3}$$

Table 2 shows the calculated NC and BER values of the proposed scheme. It is observed that the NC values of the extracted watermark of various test images are equal to 1 or nearly equal to 1 and the BER values are also very less which indicate that the extracted watermark is nearly equal to the original watermark. Figures 15 and 16 shows the performance of NC and BER with different number of divisions. Also the robustness can be measured by Tamper Assessment Function(TAF). It is observed by checking the similarity between the original watermark (w) and extracted watemark($\bar{w}$). Mathematically it is written as:

$$TAF(\%) = \frac{1}{M \times N}\left[\sum_{i=0}^{N-1} w \oplus \bar{w}\right] \times 100$$

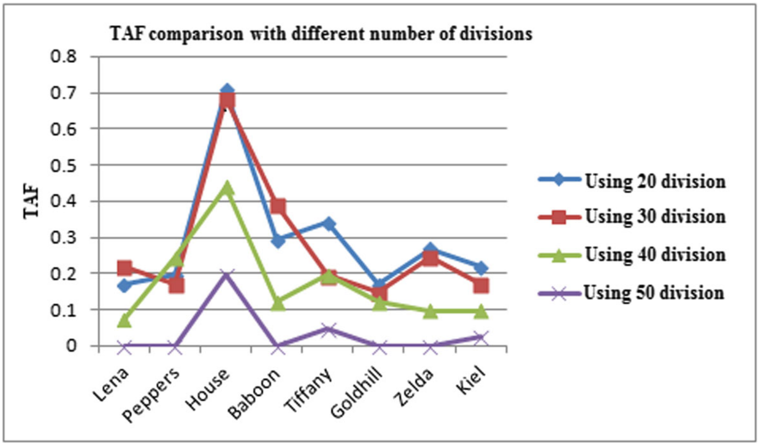


Fig. 17 TAF comparison with different number of divisions

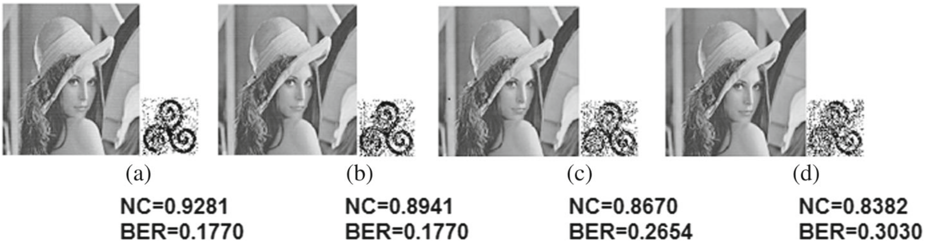


Fig. 18 Result under Gaussian Low pass filtering (2 × 2) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

Where $M \times N$ is the size of watermark image. The value of TAF shoule be closer to zero for the better result.

From the Table 3 and Fig. 17 it is observed that the values are very less which indicates the similarity of the extracted watermark from the proposed approach with the original watermark.

In general during transmission, the watermarked image may be exposed to various attacks on the watermarked images before reaching to the watermark receiver. The attacks include both geometric and non–geometric attacks. Geometric attacks include cropping, resize, rotation, scaling, translation etc and non–geometric attacks include image filtering, averaging, addition of noise, sharpening, brightness, gamma correction compression etc. In the proposed scheme, the watermarked images are verified against both geometric and non–geometric attacks using 20 division, 30 division, 40 division and 50 division. Although in the proposed scheme, number of watermarked images have generated and verified against robustness to various attacks, but as a representative only results of Lena image are presented. The figures below show the watermarked images with attacks and the extracted watermarks are shown below.

Low pass filtering To prove the robustness against the low pass filtering attacks, gaussian low pass filtering attacks with different sizes are applied on the watermarked images with different value of r. The attacked watermarked images and their corresponding extracted watermarks(logo) with calculated NC and BER values are presented in the following figures.

Gaussian Low-pass Filtering(2 × 2) (Fig. 18).

Gaussian Low-pass Filtering(3 × 3) (Fig. 19).

Gaussian Low-pass Filtering(5 × 5) (Fig. 20).

Gaussian Low-pass Filtering(7 × 7) (Fig. 21).

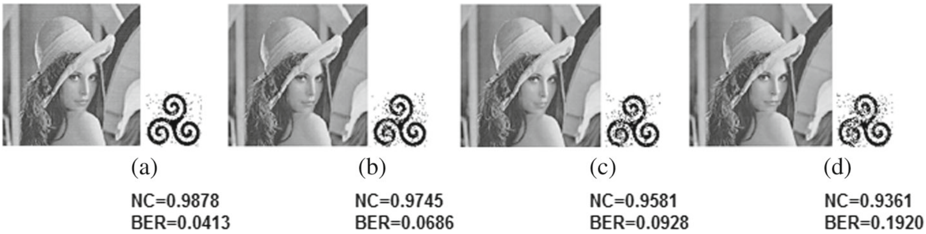


Fig. 19 Result under Gaussian Low pass filtering (3 × 3) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

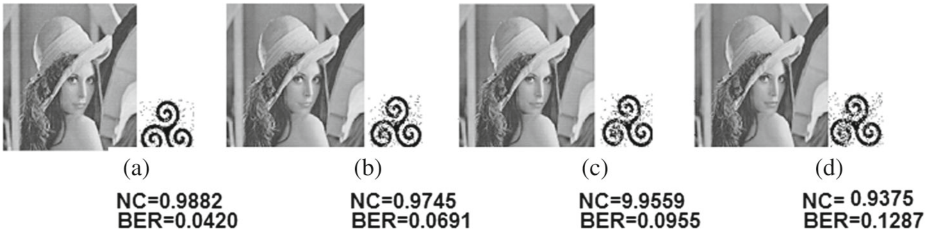


Fig. 20 Result under Gaussian Low pass filtering (5 × 5) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

From the above figures with the NC and BER values of the extracted logos, it is observed that the proposed method is efficient enough to survive the gaussian low pass filter attack.

Median filtering Median filtering attack, the center pixel value is modified by the middle value of the sorted pixel values. For the analysis, we have examined the proposed scheme is examined against median filtering attacks with different window sizes.

Median Filtering (3 × 3) (Fig. 22).

Averaging attack In this type of attack, number of many samples of a precondition data set are inserted with a different secret key or watermark and then are averaged to evaluate the attacked data. If the amount of data set is sufficiently huge, the inserted watermark cannot be discovered any more supposing that it will output zero mean on average (Fig. 23).

Image noising One of the common non-geometrical attack is the addition noise. The proposed method is tested for the robustness against salt and pepper noise attack. Salt & ppepper noise is caused during the transmission due to the pixel's error.

Salt and Pepper noise(var = 0.01) (Fig. 24).
Salt and Pepper noise(var = 0.03) (Fig. 25).
Salt and Pepper noise(var = 0.05) (Fig. 26).

Effect of gamma correction Sometimes watermarked images are enhanced intentionally or unintentionally by power law transformation which may causes the destruction or removal of watermark. The below figures shows the results after the watermarked images are enhanced by gamma correction technique with different gamma value and the recovered watermarks from the enhanced images along with their NC and BER values.

Gamma Correction(gamma = 1) (Fig. 27).

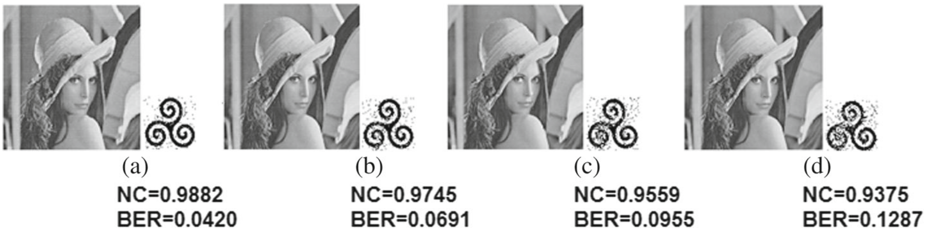


Fig. 21 Result under Gaussian Low pass filtering(7 × 7) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

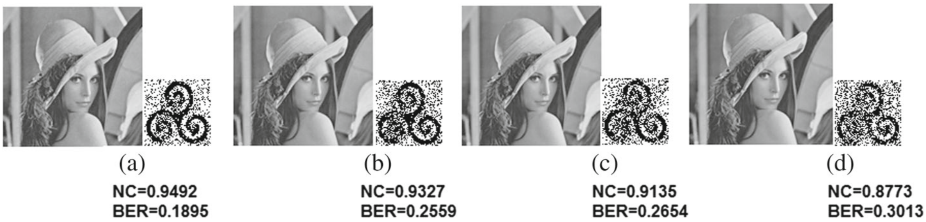


Fig. 22 Result under Median filtering(3 × 3) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

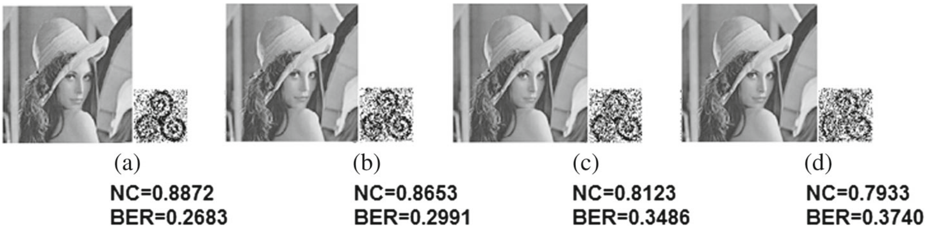


Fig. 23 Result under average filtering attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

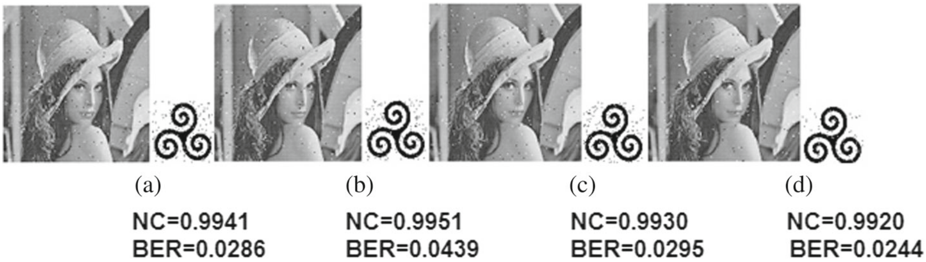


Fig. 24 Result under salt and pepper noise(var=0.01) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

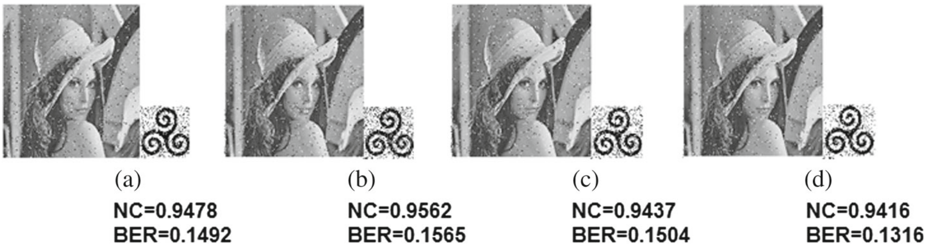
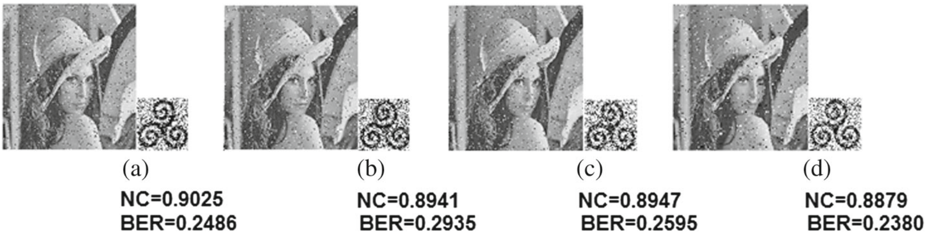


Fig. 25 Result under salt and pepper noise(var=0.03) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division



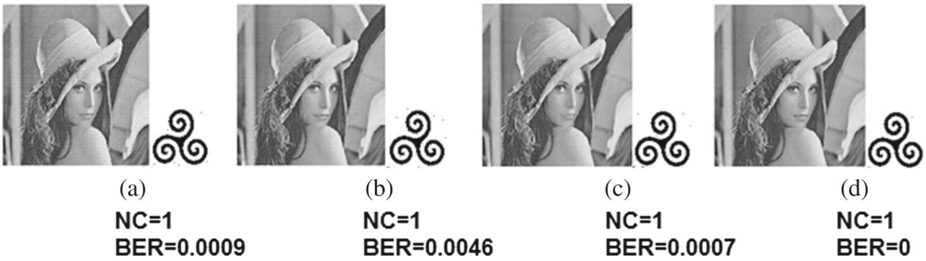
(a) NC=0.9025 BER=0.2486

(b) NC=0.8941 BER=0.2935

(c) NC=0.8947 BER=0.2595

(d) NC=0.8879 BER=0.2380

Fig. 26 Result under salt and pepper noise(var=0.05) attack and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division



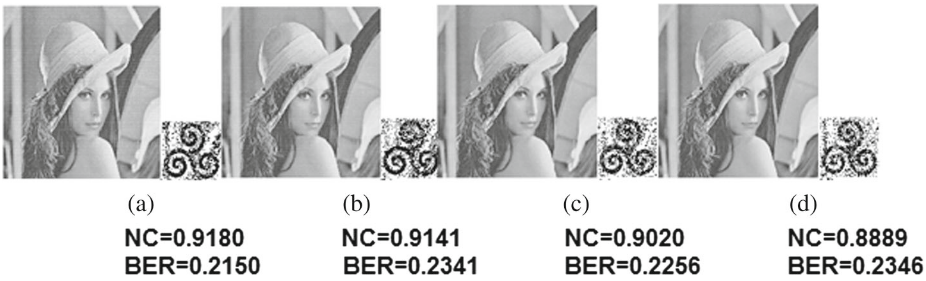
(a) NC=1 BER=0.0009

(b) NC=1 BER=0.0046

(c) NC=1 BER=0.0007

(d) NC=1 BER=0

Fig. 27 Result under effect of Gamma Correction(1) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division



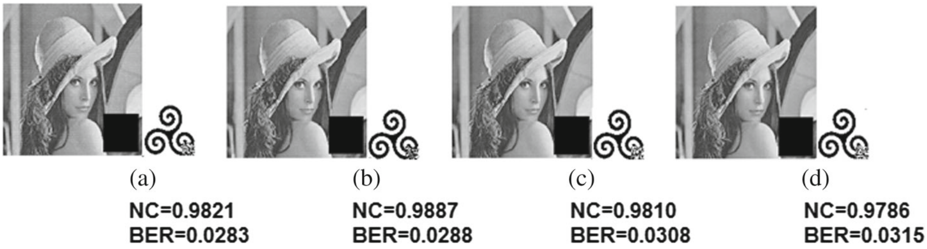
(a) NC=0.9180 BER=0.2150

(b) NC=0.9141 BER=0.2341

(c) NC=0.9020 BER=0.2256

(d) NC=0.8889 BER=0.2346

Fig. 28 Result under effect of Gamma Correction(0.75) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division



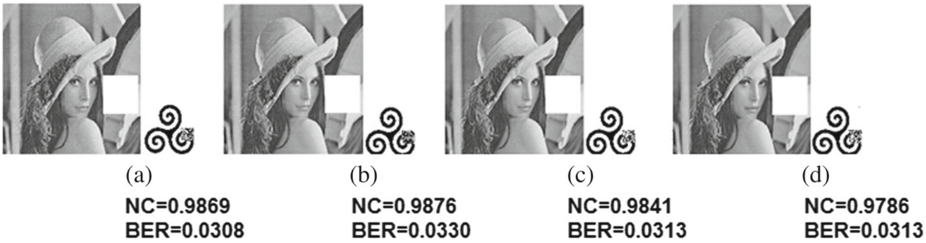
(a) NC=0.9821 BER=0.0283

(b) NC=0.9887 BER=0.0288

(c) NC=0.9810 BER=0.0308

(d) NC=0.9786 BER=0.0315

Fig. 29 Result under cropping attack(128 × 128) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

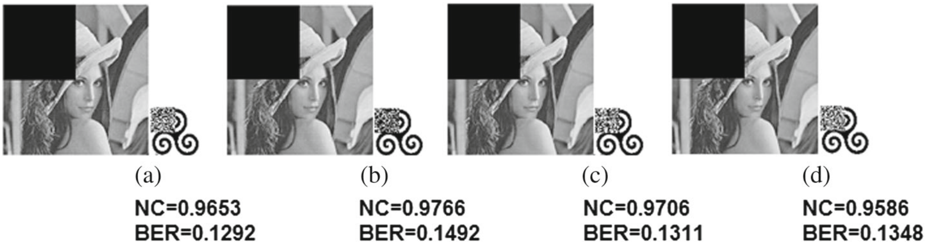

(a) NC=0.9869 BER=0.0308

(b) NC=0.9876 BER=0.0330

(c) NC=0.9841 BER=0.0313

(d) NC=0.9786 BER=0.0313

Fig. 30 Result under cropping attack(128 × 128) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

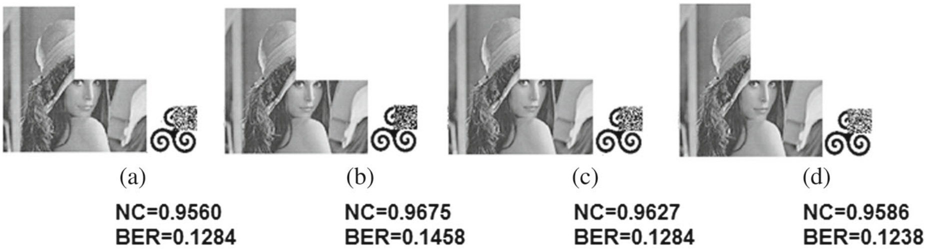

(a) NC=0.9653 BER=0.1292

(b) NC=0.9766 BER=0.1492

(c) NC=0.9706 BER=0.1311

(d) NC=0.9586 BER=0.1348

Fig. 31 Result under cropping attack(256 × 256) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

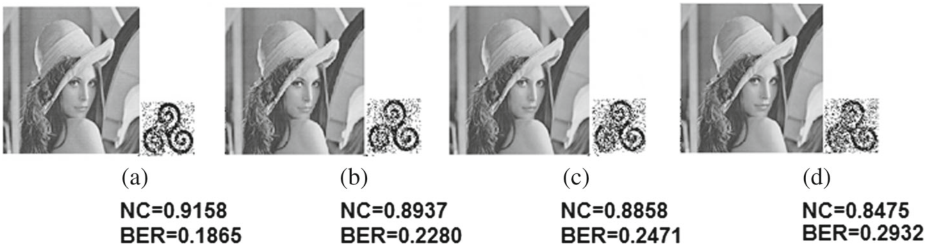

(a) NC=0.9560 BER=0.1284

(b) NC=0.9675 BER=0.1458

(c) NC=0.9627 BER=0.1284

(d) NC=0.9586 BER=0.1238

Fig. 32 Result under cropping attack(256 × 256) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division


(a) NC=0.9158 BER=0.1865

(b) NC=0.8937 BER=0.2280

(c) NC=0.8858 BER=0.2471

(d) NC=0.8475 BER=0.2932

Fig. 33 Result under effect of image resize(512→ 256 →512) and recovered watermark image using **a** 20 division **b** using 30 division **c** using 40 division **d** using 50 division

Table 4 Comparison of NC values with different sub-bands

Sub-bands	LL				LH				HL				HH			
Number of division	20	30	40	50	20	30	40	50	20	30	40	50	20	30	40	50
Gaussian Filtering(33)	0.98	0.97	0.95	0.93	0.75	0.75	0.73	0.73	0.76	0.74	0.75	0.76	0.88	0.83	0.79	0.78
Median Filtering(33)	0.98	0.98	0.99	0.99	0.75	0.74	0.74	0.74	0.77	0.76	0.75	0.75	0.76	0.76	0.76	0.76
Gaussian Noise(0.01)	0.93	0.93	0.94	0.94	0.75	0.75	0.76	0.75	0.74	0.77	0.75	0.76	0.74	0.79	0.73	0.75
Gamma Correction(0.8)	0.87	0.88	0.88	0.88	0.76	0.75	0.74	0.74	0.77	0.75	0.75	0.74	0.99	0.99	0.98	0.93
Rotation(20)	0.88	0.89	0.89	0.89	0.75	0.75	0.75	0.75	0.77	0.73	0.74	0.74	0.77	0.73	0.76	0.74
Histogram Equalization	0.88	0.88	0.88	0.88	0.74	0.74	0.74	0.77	0.75	0.74	0.74	0.74	0.86	0.83	0.79	0.79
Scaling(0.5,2)	0.97	0.98	0.99	0.99	0.75	0.74	0.72	0.73	0.74	0.75	0.74	0.75	0.74	0.74	0.75	0.8
Sharpening	0.94	0.98	0.96	0.96	0.74	0.74	0.74	0.74	0.75	0.76	0.74	0.73	0.77	0.74	0.75	0.75

Table 5 Comparison of BER values with different sub-bands

Sub-bands	CA				CH				CV				CD			
Number of division	20	30	40	50	20	30	40	50	20	30	40	50	20	30	40	50
Gaussian Filtering(33)	0.04	0.08	0.09	0.11	0.51	0.5	0.51	0.49	0.47	0.49	0.48	0.46	0.31	0.39	0.43	0.45
Median Filtering(33)	0.19	0.25	0.27	0.29	0.51	0.53	0.51	0.48	0.46	0.5	0.47	0.46	0.48	0.5	0.46	0.46
Gamma Correction(0.8)	0.25	0.21	0.2	0.19	0.51	0.52	0.5	0.49	0.47	0.51	0.49	0.48	0.02	0.03	0.04	0.19
Histogram Equalization	0.34	0.4084	0.38	0.4	0.51	0.52	0.5	0.47	0.49	0.51	0.48	0.48	0.35	0.42	0.44	0.41
Scaling(0.5,2)	0.18	0.23	0.24	0.26	0.5	0.51	0.51	0.48	0.49	0.5	0.48	0.47	0.47	0.5	0.48	0.26

Table 6 Comparision of NC values of the proposed method with Kumsawat et al. [13] and Lingamgunta et al. [14]

Image	Kumsawat et al. [13]	Lingamgunta et al. [14]	Proposed method			
	NC	NC	NC			
			20 division	30 division	40 division	50 division
Lena	0.97	0.98	**1**	**1**	**1**	**1**
Peppers	0.93	0.95	**1**	**1**	**1**	**1**
Baboon	0.95	0.96	**1**	**1**	**1**	**1**
Tiffany	0.97	0.97	**1**	**1**	**1**	**1**
Cameraman	0.94	0.95	**1**	**1**	**1**	**1**
Barbara	0.95	0.96	**1**	**1**	**1**	**1**

Table 7 The comparison results with Ansari et al. [3] and Ali & Ahn et al. [1], for common image processing operations (NC)

Attack	Ansari et al. [3]	Ali & Ahn [1]	Proposed method			
			Using 20 division	Using 30 division	Using 40 division	Using 50 division
Gaussian Filtering(33)	0.99	0.96	0.98	0.97	0.95	0.93
Median Filtering(33)	0.98	0.94	0.98	0.98	0.99	0.99
Average Filtering(33)	0.97	0.91	0.88	0.89	0.87	0.79
Gaussian Noise(0.01)	0.94	0.85	0.93	0.93	0.94	0.94
Salt & Pepper(0.001)	0.99	0.92	0.97	0.98	0.98	0.99
Gamma Correction(0.8)	0.99	0.96	0.87	0.88	0.88	0.88
Rotation(20°)	0.98	0.94	0.88	0.89	0.89	0.89
Histogram Equilization	0.98	0.92	0.88	0.88	0.88	0.88
Scaling(0.5,2)	0.98	0.94	0.97	0.98	0.99	0.99
Sharpening	0.94	0.88	0.94	0.98	0.96	0.96

Gamma Correction(gamma = 0.75) (Fig. 28).

Cropping attack Cropping attack tries to remove some parts of the image with the intention to destroy the embedded watermark.

Cropping(128 × 128 by black) (Fig. 29).

Cropping(128 × 128 by white) (Fig. 30).

Table 8 The comparison results with Verma et al. [25], for common image processing operations (NC)

Attack	Verma et al. [25]	Proposed Method			
		Using 20 division	Using 30 division	Using 40 division	Using 50 division
Gaussian Filtering(3 × 3)	0.97	0.98	0.97	0.95	0.93
Median Filtering(3 × 3)	0.95	0.98	0.98	0.99	0.99
Average Filtering(3 × 3)	0.86	0.88	0.89	0.81	0.79
Gaussian Noise(0.01)	0.97	0.93	0.93	0.94	0.94
Gaussian Noise(0.02)	0.85	0.91	0.91	0.92	0.92
Salt & Pepper(0.01)	0.75	0.96	0.97	0.98	0.98
Speckle Noise(0.01)	0.74	0.96	0.97	0.97	0.98
Cropping(1/4)	0.93	0.97	0.98	0.98	0.99
Rotation(0.1°)	0.85	0.97	0.98	0.99	0.99
Rotation(0.02°)	0.45	0.97	0.98	0.98	0.98
Histogram Equalization	0.92	0.88	0.88	0.88	0.88
Scaling	0.98	0.97	0.98	0.99	0.99
Sharpening	0.88	0.94	0.95	0.96	0.96

Table 9 The comparison results with Ali et al. [2], for common image processing operations (NC)

Attack	Ali et al. [2]	Proposed Method			
		Using 20 division	Using 30 division	Using 40 division	Using 50 division
Gaussian Filtering(3 × 3)	0.99	0.98	0.97	0.95	0.93
Median Filtering(2 × 2)	0.9	0.98	0.98	0.99	0.99
Gaussian Noise(0.001)	0.98	0.94	0.94	0.95	0.95
Histogram Equalization	0.99	0.88	0.88	0.88	0.88
Scaling(1/2)	0.91	0.97	0.98	0.99	0.99
Salt & Pepper(0.05)	0.81	0.93	0.94	0.94	0.94

Table 10 The comparison results with Thabit et al. [24], for common image processing operations (NC)

Attack	Thabit et al. [24]	Proposed Method			
		Using 20 division	Using 30 division	Using 40 division	Using 50 division
Salt & Pepper(0.002)	0.93	0.97	0.98	0.98	0.98
Salt & Pepper(0.004)	0.96	0.97	0.98	0.98	0.98
Salt & Pepper(0.008)	0.84	0.97	0.97	0.98	0.98
Salt & Pepper(0.01)	0.82	0.96	0.97	0.98	0.98
Salt & Pepper(0.02)	0.81	0.96	0.96	0.97	0.97
Gamma Correction(2)	0.85	0.67	0.66	0.65	0.65
Gamma Correction(1.5)	0.89	0.81	0.80	0.79	0.79
Gamma Correction(1)	1	1	1	1	1
Gamma Correction(0.8)	0.53	0.87	0.88	0.88	0.88
Sharpening	0.92	0.94	0.95	0.96	0.96
Cropping(64 × 64)	0.92	0.97	0.98	0.98	0.99
Cropping(128 × 128)	0.84	0.97	0.98	0.98	0.99
Cropping(200 × 200)	0.82	0.97	0.98	0.98	0.99

Table 11 The comparison results with Wang et al. [27], for common image processing operations (NC)

Attack	Wang et al. [27]	Proposed Method			
		Using 20 division	Using 30 division	Using 40 division	Using 50 division
Gaussian Filtering(3 × 3)	0.96	0.98	0.97	0.95	0.93
Median Filtering(3 × 3)	0.95	0.98	0.98	0.99	0.99
Gaussian Noise	0.94	0.93	0.93	0.94	0.94
Cropping	0.76	0.97	0.98	0.98	0.99
Scalling(1/2)	0.91	0.97	0.98	0.99	0.99

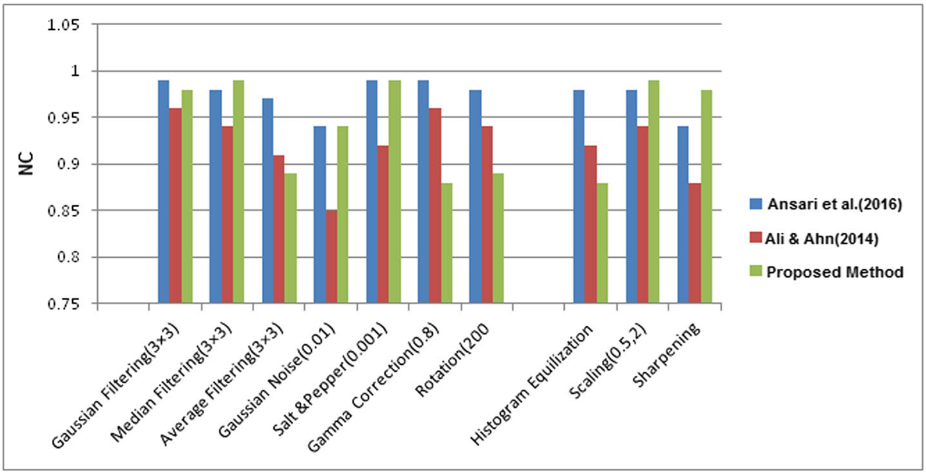


Fig. 34 The comparison results with Ansari et al. [3] and Ali & Ahn [1], for common image processing operations (NC)

Cropping(256 × 256 by black) (Fig. 31).

Cropping(256 × 256 by white) (Fig. 32).

The watermarked images after cropping attacks and the recovered watermarks along with the NC and BER values are shown above. It is observed that the proposed method can strong enough to survive various types of cropping attacks.

Effect of image resize In image resizing attack, the size of the image is reduced by different factors and again resize to the original size.

Image resize(512→ 256 →512) (Fig. 33).

Also the robustness of the proposed scheme is observed by considering the different sub-bands of the images which are indicated in Tables 4 and 5 in terms of the NC and BER values.

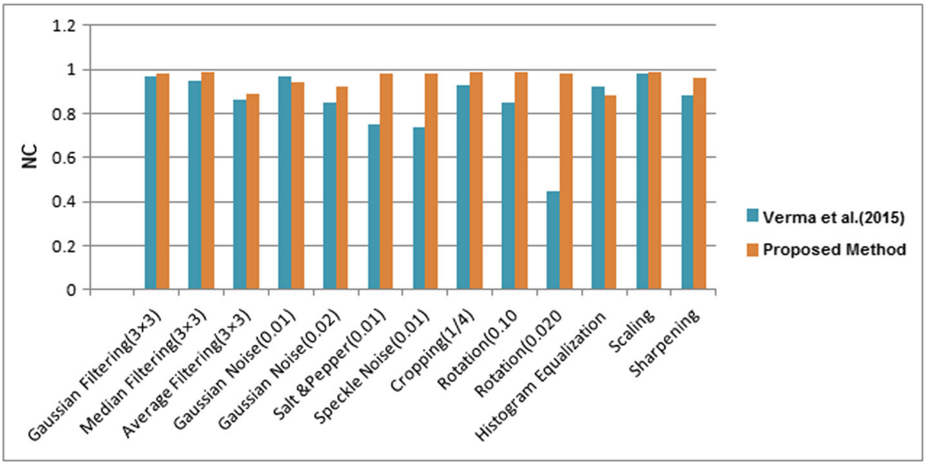


Fig. 35 The comparison results with Verma et al. [25], for common image processing operations (NC)

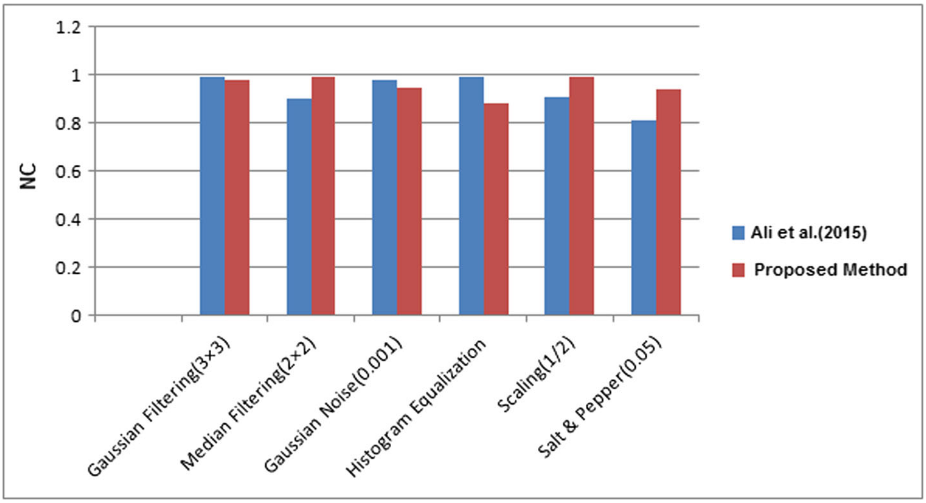


Fig. 36 The comparison results with Ali et al. [2], for common image processing operations (NC)

Comparative analysis The proposed robust watermarking scheme is based on IWT domain. In the previous section we have shown the efficiency of the proposed method in terms of imperceptibility and robustness to different types of attack by calculating $PSNR$, NC and BER values. Also the efficiency of the proposed scheme is compared with some existing robust watermarking schemes in transform domain on the basis of NC values before attack and also after applying some image processing attacks on the watermarked images Tables 6, 7, 8, 9, 10, and 11 and Figs. 34, 35, 36, 37, and 38.

Compared with other existing methods, the proposed watermarking algorithm has better robustness against various geometric and non-geometric attacks.

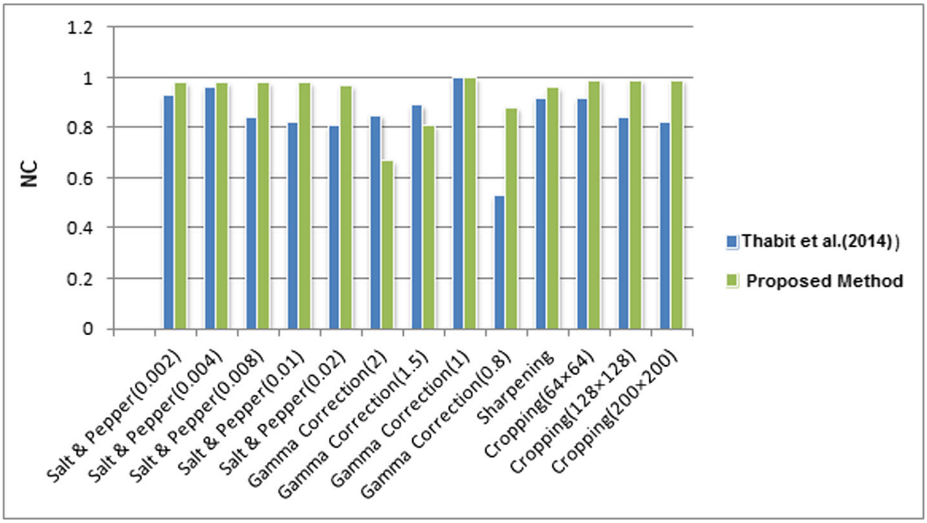


Fig. 37 The comparison results with Thabit et al. [24], for common image processing operations (NC)

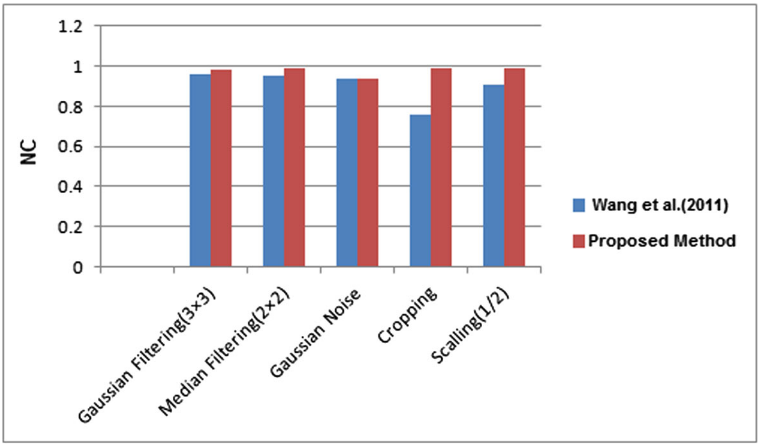


Fig. 38 The comparison results with Wang et al. [27], for common image processing operations (NC)

5 Conclusion

In this paper, a novel robust and blind binary watermarking scheme is proposed. The watermark (logo)is embedded in the low sub–band of the IWT transformed cover image. However in the low sub–band portion, the chance of watermark distortion or removal is high. But in the proposed method even after the various types of attacks applied on the watermarked images, the watermark is able to survive with low BER and high NC values. The quality of the watermarked images are also good in terms of perceptibility and $PSNR$. Also as compared with some existing schemes, the proposed scheme indicate its efficiency with high robustness.

References

1. Ali M, Ahn CW (2014) An optimized watermarking technique based on self-adaptive de in dwt–svd transform domain. Signal Process 94:545–556
2. Ali M, Ahn CW, Pant M, Siarry P (2015) An image watermarking scheme in wavelet domain with optimized compensation of singular value decomposition via artificial bee colony. Inf Sci 301:44–60
3. Ansari IA, Pant M, Ahn CW (2016) Robust and false positive free watermarking in iwt domain using svd and abc. Eng Appl Artif Intell 49:114–125
4. Arsalan M, Malik SA, Khan A (2012) Intelligent reversible watermarking in integer wavelet domain for medical images. J Syst Softw 85(4):883–894
5. Bohra A, Farooq O et al. (2009) Blind self-authentication of images for robust watermarking using integer wavelet transform. AEU-Int J Electron C 63(8):703–707
6. Chetan KR, Nirmala S (2015) An efficient and secure robust watermarking scheme for document images using integer wavelets and block coding of binary watermarks. Journal of Information Security and Applications 24:13–24
7. Fazli S, M Moeini A (2016) Robust image watermarking method based on dwt, dct, and svd using a new technique for correction of main geometric attacks. Optik-International Journal for Light and Electron Optics 127(2):964–972
8. Furht B, Kirovski D (2004) Multimedia security handbook. CRC Press
9. Gunjal BL, Manthalkar RR (2010) An overview of transform domain robust digital image watermarking algorithms. Journal of Emerging Trends in Computing and Information Sciences 2(1):37–42

10. Hwang M-S, Chang C-C, Hwang K-F (1999) A watermarking technique based on one-way hash functions. IEEE Trans Consum Electron 45(2):286–294
11. Jia S-L (2014) A novel blind color images watermarking based on svd. Optik-International Journal for Light and Electron Optics 125(12):2868–2874
12. Kaushik AK (2012) A novel approach for digital watermarking of an image using dft. Int JElectronComp Sci Eng 1(1):35–41
13. Kumsawat P, Attakitmongcol K, Srikaew A (2007) A robust image watermarking scheme using multiwavelet tree. In: World congress on engineering. Citeseer, pp 612–618
14. Lingamgunta S, Vakulabaranam VK, Thotakura S (2013) Reversible watermarking for image authentication using iwt. Int J Signal Process, Image Process Pattern Recog 6(1):145–156
15. Maity SP, Kundu MK (2002) Robust and blind spatial watermarking in digital image. In: ICVGIP
16. Mukherjee DP, Maitra S, Acton ST (2004) Spatial domain digital watermarking of multimedia objects for buyer authentication. IEEE Trans Multimedia 6(1):1–15
17. Nikolaidis N, Pitas I (1998) Robust image watermarking in the spatial domain. Signal Process 66(3):385–403
18. Nin J, Ricciardi S (2013) Digital watermarking techniques and security issues in the information and communication society. In: 27th international conference on advanced information networking and applications workshops (WAINA), 2013. IEEE, pp 1553–1558
19. Pal AK, Roy S (2016) A robust and blind image watermarking scheme in dct domain. Int J Inf Comput Secur, In Press
20. Parashar P, Singh R (2014) A survey: digital image watermarking techniques. Int J Signal Process Image Process Pattern Recognit 7(6):111–124
21. Sebé F, Domingo-Ferrer J, Herrera J (2000) Spatial-domain image watermarking robust against compression, filtering, cropping, and scaling. In: International workshop on information security. Springer, pp 44–53
22. Sverdlov A, Dexter S, Eskicioglu AM (2005) Robust dct-svd domain image watermarking for copyright protection: embedding data in all frequencies. In: Signal processing conference, 2005 13th European. IEEE, pp 1–4
23. Tao P, Eskicioglu AM (2004) A robust multiple watermarking scheme in the discrete wavelet transform domain. In: Optics east. International society for optics and photonics, pp 133–144
24. Thabit R, Khoo BE (2014) Robust reversible watermarking scheme using slantlet transform matrix. J Syst Softw 88:74–86
25. Verma VS, Jha RK, Ojha A (2015) Significant region based robust watermarking scheme in lifting wavelet transform domain. Expert Syst Appl 42(21):8184–8197
26. Wang C, Li X, Yang B (2010) Efficient reversible image watermarking by using dynamical prediction-error expansion. In: 2010 IEEE international conference on image processing. IEEE, pp 3673–3676
27. Wang Y-R, Lin W-H, Yang L (2011) An intelligent watermarking method based on particle swarm optimization. Expert Syst Appl 38(7):8024–8029
28. Weng S, Pan J-S (2016) Integer transform based reversible watermarking incorporating block selection. J Vis Commun Image Represent 35:25–35

Kshiramani Naik is presently working as an Assistant Professor in the Department of Computer Science and Engineering and IT, Veer Surendra Sai University of Technology, Odisha,India. She completed her Ph.D in Computer Science and Engineering from Indian Institute of Technology(ISM), Dhanbad in 2017. She received her BE in CSE and M.Tech in CSE from BPUT Rourkela and NIT Rourkela respectively. Her research interest includes Image Cryptosystem, Steganography and Watermarking.

Saswati Trivedy completed her MTech in Computer Science and Engineering from Indian School of Mines, Dhanbad in 2015. She received her B.Tech in Electronics and Communication Engineering from West Bengal University of Technology, Kolkata. Her research interest includes Image Cryptography and Watermarking.

Arup Kumar Pal is presently working as an Assistant Professor in the Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, India. Prior to join this institute, he was a Lecturer in the Department of Computer Science & Engineering, NIT Jamshedpur during April, 2011 to December, 2011. He did his Ph.D in Computer Science and Engineering from Indian School of Mines, Dhanbad in 2011. He has around 4 years of teaching and research experiences, and contributed a number of research papers in several journals and conference proceedings of National and International reputes. His main research interest includes Vector Quantization, Image Compression, Image Cryptosystem, Steganography, Watermarking and CBIR.