Chapter 4

RWS for Authentication, Tamper Detection

In this chapter, two image watermarking schemes, DRWS-LBP and RWS-LBP-HC have been introduced for authentication and tamper detection. In DRWS-LBP, dual image has been used for the watermarking technique. LBP operator is employed on cover image to generate AC using shared secret key (μ) which is used to check the authenticity and to detect the tamper on watermarked images. In RWS-LBP-HC, an RWS has been proposed with the help of LBP and Hamming code. Here, the LBP operator is used for image authentication, and HC is used to tamper detection and correction. In both the methods, an effort has been made to find the robustness of the schemes against some standard attacks.

4.1 LBP based Dual RWS [3]

Watermarking scheme is an efficient solution to protect multimedia documents. Many watermarking schemes have been developed for various applications, but authentication and tamper detection is still a significant area of research. In DRWS-LBP, dual image based watermarking technique has been developed using LBP to protect multimedia documents from illegal modification. The new method includes the following steps: First, the host image is partitioned into (3×3) non-overlapping blocks. Then system vector (S) is generated using LBP, and an XOR operation is performed with secret watermark bits. Two-bit AC is generated from S vector and embedded within dual image relying on a shared secret key (μ). The watermark and the cover image can be successfully recovered from the dual watermarked image at the time of extraction. After extraction, it successfully performs an authentication check and detects tamper on watermarked image. The experimental outcomes are compared with the state-of-the-art methods to show the effectiveness of DRWS-LBP. Some standard NIST recommended steganalysis and attacks are conducted to evaluate the robustness and imperceptibility. It has been found that DRWS-LBP is robust and secured against different standard attacks. Meanwhile, it can detect the message integrity within the watermarked images.

The scheme DRWS-LBP is described in two subsections (4.1.1 and 4.1.2). The schematic framework of watermark embedding, extraction and authentication verification process have been depicted in Fig. 4.1(a), 4.1(b), and 4.1(c) respectively.

[3] Published in **International Journal of Security and Privacy**, Wiley, 03 February 2019: Online ISSN:2475-6725, with title *Watermarking Scheme using LBP for Image Authentication and Tamper Detection through dual image*.

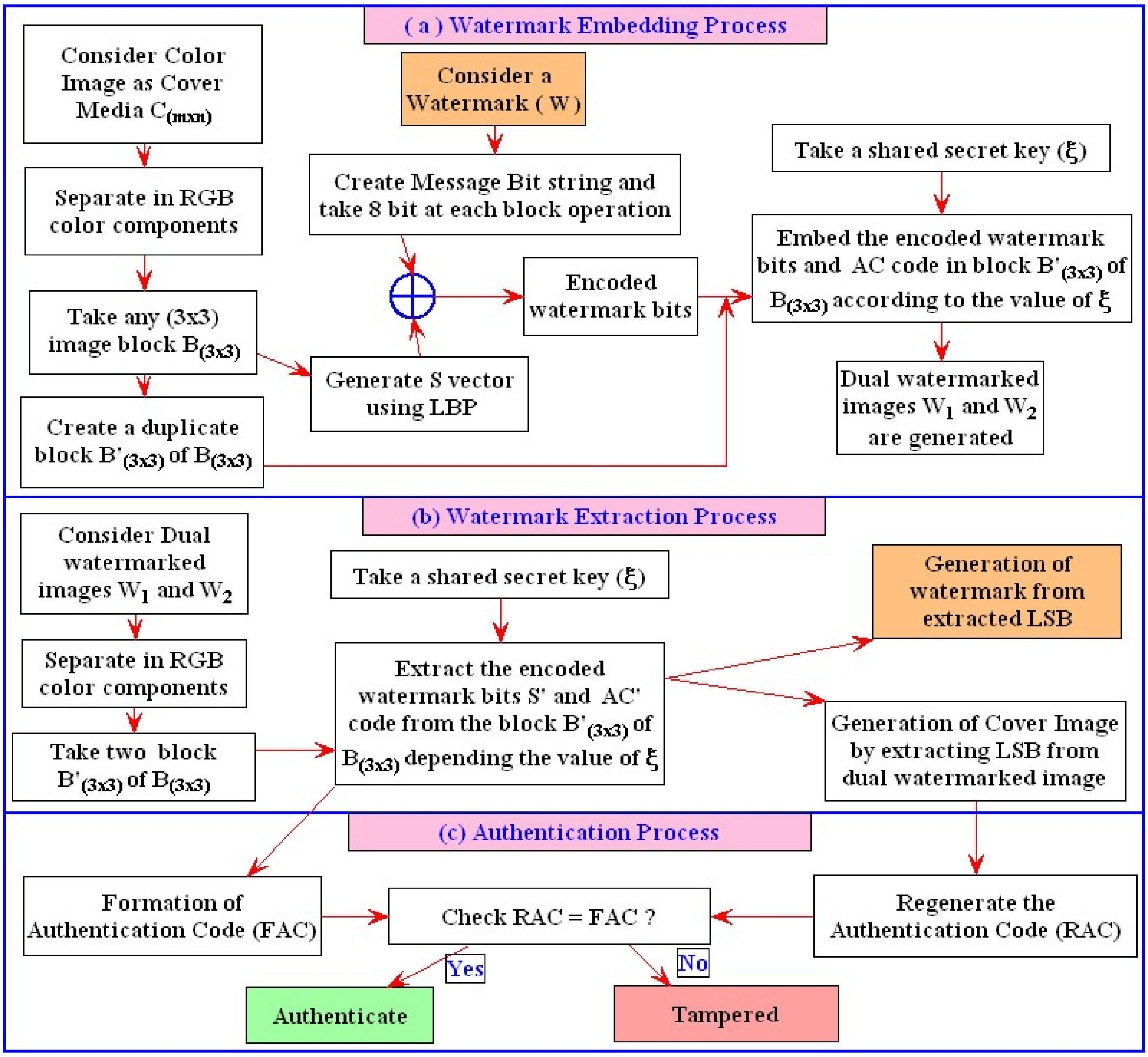
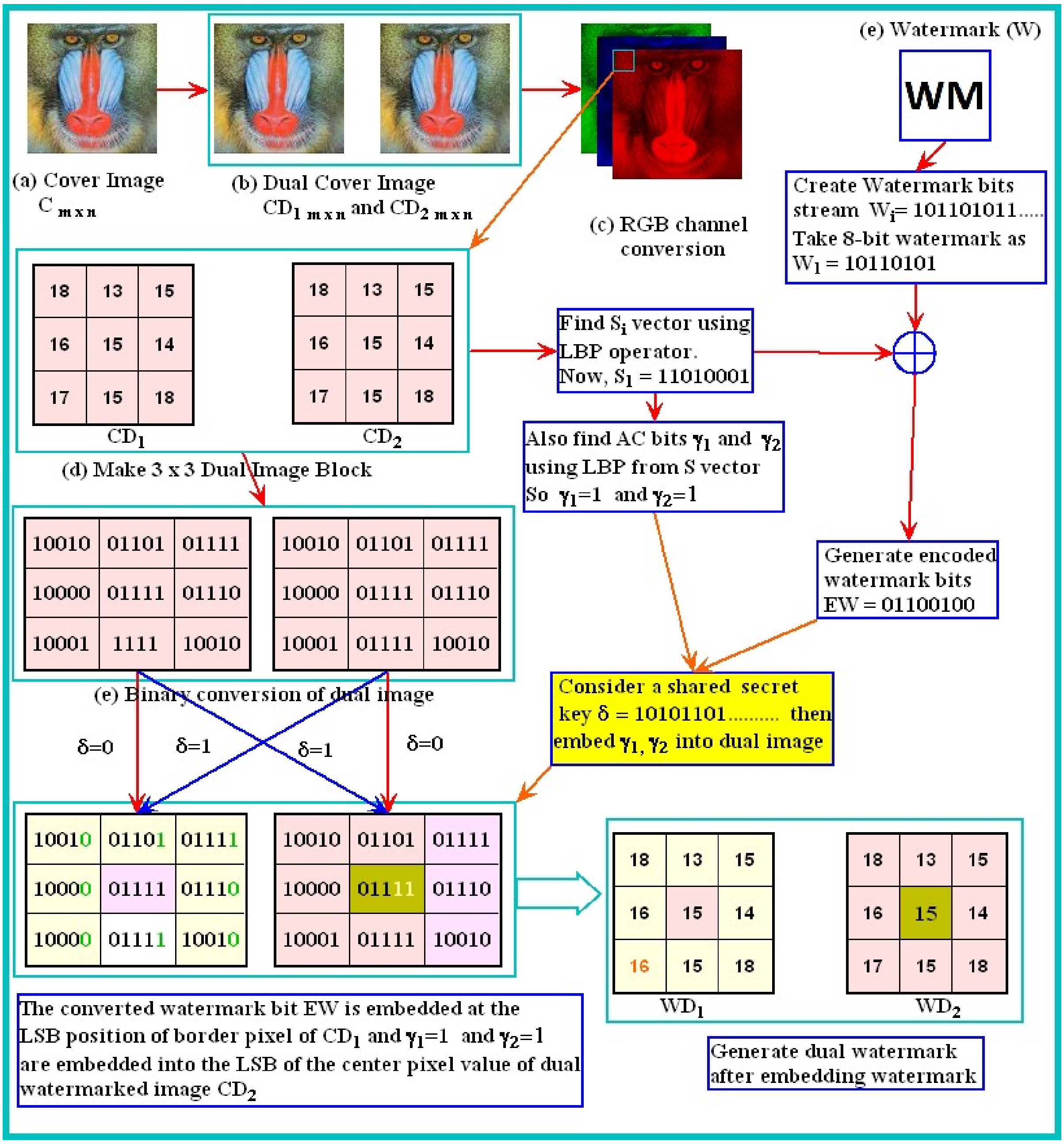

(a) Watermark Embedding Process
Consider Color Image as Cover Media C(mxn)
Separate in RGB color components
Take any (3x3) image block B(3x3)
Create a duplicate block B'(3x3) of B(3x3)
Consider a Watermark (W)
Create Message Bit string and take 8 bit at each block operation
Encoded watermark bits
Generate S vector using LBP
Take a shared secret key (ξ)
Embed the encoded watermark bits and AC code in block B'(3x3) of B(3x3) according to the value of ξ
Dual watermarked images W1 and W2 are generated
(b) Watermark Extraction Process
Consider Dual watermarked images W1 and W2
Separate in RGB color components
Take two block B'(3x3) of B(3x3)
Take a shared secret key (ξ)
Extract the encoded watermark bits S' and AC' code from the block B'(3x3) of B(3x3) depending the value of ξ
Generation of watermark from extracted LSB
Generation of Cover Image by extracting LSB from dual watermarked image
(c) Authentication Process
Formation of Authentication Code (FAC)
Check RAC = FAC ?
Regenerate the Authentication Code (RAC)
Yes
No
Authenticate
Tampered

Figure 4.1: Block diagram of watermarking process in DRWS-LBP.

4.1.1 Watermark Embedding Phase

In this scheme, a dual image based RWS using LBP has been developed. First a color image CI is considered as a cover image. Then a copy of the cover image is generated to construct the dual images $(CI_{o(m\times n)})$ and $(CI_{d(m\times n)})$.

Now, both the cover images are separated into RGB color pixel blocks and then (3×3) image blocks are considered from each. Taking the CI_{oi} pixels into account and all pixel values are converted in the binary form as $CI_{o0}, CI_{o1}, \ldots, CI_{o8}$. Then, from the (3×3) image block S vector is calculated as $S = CI_{o0} \oplus CI_{o1} \oplus CI_{o2} \oplus CI_{o3} \oplus CI_{o4} \oplus CI_{o5} \oplus CI_{o6} \oplus CI_{o8}$ and 8-bit S_i vector is obtained as $S_0, S_1, \ldots, S_7$. Moreover, a 2-bit AC, γ_1 and γ_2 are created by applying the equation $\gamma_1 = S_1 \oplus S_3 \oplus S_5 \oplus S_7; \quad \gamma_2 = S_0 \oplus S_2 \oplus S_4 \oplus S_6$. Now W is considered and a watermark bits stream M_i is generated from W. Then from M_i, an 8-bit watermark is



Create Watermark bits stream W_i= 101101011.....
Take 8-bit watermark as W_1 = 10110101

Find S_i vector using LBP operator.
Now, S_1 = 11010001

Also find AC bits γ_1 and γ_2 using LBP from S vector
So γ_1=1 and γ_2=1

Generate encoded watermark bits
EW = 01100100

Consider a shared secret key δ = 10101101.......... then embed γ_1, γ_2 into dual image

18	13	15
16	15	14
17	15	18

18	13	15
16	15	14
17	15	18

10010	01101	01111
10000	01111	01110
10001	1111	10010

10010	01101	01111
10000	01111	01110
10001	01111	10010

10010	01101	01111
10000	01111	01110
10000	01111	10010

10010	01101	01111
10000	01111	01110
10001	01111	10010

18	13	15
16	15	14
16	15	18

18	13	15
16	15	14
17	15	18

The converted watermark bit EW is embedded at the LSB position of border pixel of CD_1 and γ_1=1 and γ_2=1 are embedded into the LSB of the center pixel value of dual watermarked image CD_2

Generate dual watermark after embedding watermark

Figure 4.2: Numerical illustration of watermark embedding phase in DRWS-LBP

selected. Before the embedding procedure, a 512-bit shared secret key μ has been generated by the SHA-512 encryption algorithm. After getting an encrypted watermark (EW_i), an XOR-operation is performed between $M1_i$ (first 8-bit of M_i) and S_i. Now, first block of dual images $CI_{o(m \times n)}$ and $CI_{d(m \times n)}$ are considered for the embedding purpose of EW_i and AC bits. This EW_i is embedded into LSB of the border pixel of the first block of $CI_{o(m \times n)}$ and two AC bit is embedded into the 2 LSB of the first block of the center pixel of $CI_{d(m \times n)}$. Furthermore, to serve the purpose of security, the blocks are chosen depending on the shared secret key (μ). If

Algorithm 4.1: DRWS-LBP: Watermark Embedding Algorithm

```
Input  : A Cover Image of size $(m \times n)$
Output: Dual Watermarked Image of size $(m \times n)$
Algorithm Embedding():
    for ( y=0; y < imageHeight; y=y+3) do
        for ( x=0; x < imageWidth; x=x+3) do
            for ( color=0; color <= 2; color++) do
                String strLBP;
                // Get LBP data from $3 \times 3$ pixel block of first dual image
                // If the pixel value is greater than middle pixel, append 1 or append 0
                strLBP=getLBP(dualImage1);
                // Get 8 bit secret data
                secretData=get8BitsSecretData();
                // Xor this 8 bit secret data with LBP string
                String strS=XOR(strLBP, secretData);
                // Embed this XOR data in the $3 \times 3$ block of first dual image
                for (i=0; i < 8; i++) do
                    changeLSB1(image1[x+0][y+0][color],strS.getBit());
                end
                // Get w1 by XOR ing 1,3,5,7th bits
                w1=getW1Data(strS);
                // Get w2 by XOR ing 0,2,4,6th bits
                w2=getW2Data(strS);
                // Change dualImage2 LSB1 with w1
                changeLSB1(dualImage2, w1);
                // Change dualImage2 LSB2 with w2
                changeLSB2(dualImage2, w2);
            end
        end
    end
```

the value of μ is “1” then embed EW_i in a clockwise manner into the $(CI_{o(m\times n)})$ image blocks and AC into $(CI_{d(m\times n)})$ image blocks respectively. Otherwise, embed EW_i in an anti-clockwise manner into the $(CI_{d(m\times n)})$ image blocks and AC into $(CI_{o(m\times n)})$ image blocks respectively. The above process is applied to all the pixel blocks of the dual images. After embedding the entire watermark M_i, two dual watermarked image $(WI_{o(m\times n)})$ and $(WI_{d(m\times n)})$ are created. A numerical illustration of the embedding process is shown in Fig. 4.2 and the algorithmic description is shown in Algorithm 4.1.

4.1.2 Watermark Extraction and Recovery Phase

In this section, the detail extraction procedure of DRWS-LBP has been discussed. First, the dual watermarked images WI_o and WI_d are considered as input image then decomposed into R, G, and B color components. A pixel block of size (3×3) is considered from both the images WI_o and WI_d. Then according to the shared secret key μ the image blocks are selected for extraction of the encrypted watermark (EW') and authentication bits. Now, cover image $(RC_{(m\times n)})$ is reconstructed by considering the pixel values of the border pixel of WI_d and center pixel of

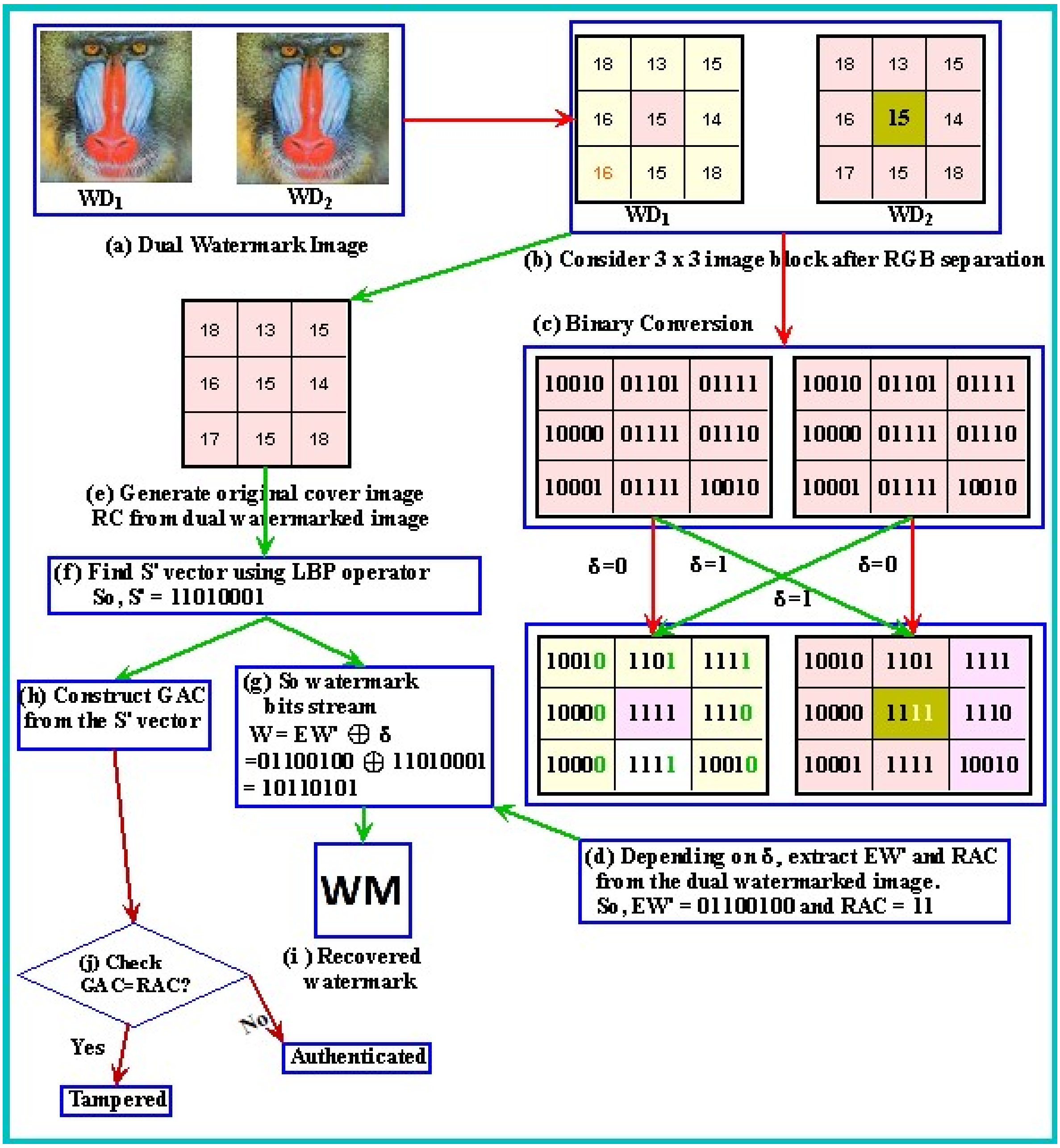


Figure 4.3: Numerical illustration of watermark extraction phase in DRWS-LBP

WI_o. After that, RC_i pixels are taken into account and all pixel values are converted into the binary form as $RC_0, RC_1, \ldots, RC_8$. S' vector is calculated from the (3×3) image block as $S' = RC_0 \oplus RC_1 \oplus RC_2 \oplus RC_3 \oplus RC_4 \oplus RC_5 \oplus RC_6 \oplus RC_7 \oplus RC_8$ and 8-bit S'_i vector is obtained as $S'_0, S'_1, \ldots, S'_7$. Moreover, 2-bit AC γ'_1 and γ'_2 are constructed by applying the equation $\gamma'_1 = S'_1 \oplus S'_3 \oplus S'_5 \oplus S'_7$; $\quad \gamma'_2 = S'_0 \oplus S'_2 \oplus S'_4 \oplus S'_6$. Furthermore, GAC can be constructed by concatenating the γ'_1 and γ'_2. Then the pixel block WI_o is considered and the LSB of the border pixels are extracted and concatenated into EW' to generate extracted encrypted watermark. Also, recovered authentication code (RAC) can be obtained by extracting

Algorithm 4.2: DRWS-LBP: Watermark Extraction Algorithm

```
Input  : Cover Image
Output: Stego image
Algorithm Extraction():
    for ( y=0; y<imageHeight; y=y+3) do
        for ( x=0; x < imageWidth; x=x+3) do
            for ( color=0; color <= 2; color++) do
                String strLSB;
                // Get LSBs from 8 pixels and store it in strLSB
                strLSB=getLSB(stegoImage1);
                // Get LBP data from 3x3 pixel block of first dual image
                String strLBP;
                // If the pixel value is greater than middle pixel, append 1 or append 0
                // XOR this LBP string with 8 bit LSB string
                strLBP=getLBP(stegoImage2);
                // Append this data to secret data string
                data=XOR(strLBP, strLSB);
                SecretData.append(data);
            end
        end
    end
// Generate secret image from secret bits
CreateSecretImage(SecretBits);
// Create cover image from unchanged pixels of stegoImage1 and stegoImage2
CreateCoverImage(stegoImage1, stegoImage2);
```

the 2 LSB of the center pixel of WI_d. After that, an XOR operation is performed between μ and EW' and stored in EXW. Thus, the whole process is repeated for the remaining blocks using the reverse process of watermark embedding. The authentication process can be tested by comparing GAC & RAC, if they are equal then no tamper has occurred and the cover image is authenticated. Otherwise, there is a tamper in the watermarked image. The detail watermark extraction algorithm is presented in Algorithm 4.2 and a numerical illustration of DRWS-LBP is depicted in Fig. 4.3.

4.1.3 Experimental Results and Comparison

A set of benchmark [89], [61], [90], [26] colour images of size (510×510) are considered to assess the effectiveness of DRWS-LBP. Three different sizes of logo images are considered as a watermark, shown in Fig. 4.5 to measure the quality and corresponding capacity. MSE [35], PSNR [35], SSIM [79] and Q-Index are computed using the equation (2.5), (2.6), (2.8) and (2.12) respectively to test the perceptible characteristics after embedding the watermark. Also NCC [94], BER [65], SD [35] and CC [35] are computed using the equation (2.11), (2.13), (2.9) and (2.10) respectively for tamper detection in a watermarked image. Performance of DRWS-LBP is also assessed on the basis of time complexity and it is compared with other existing schemes.

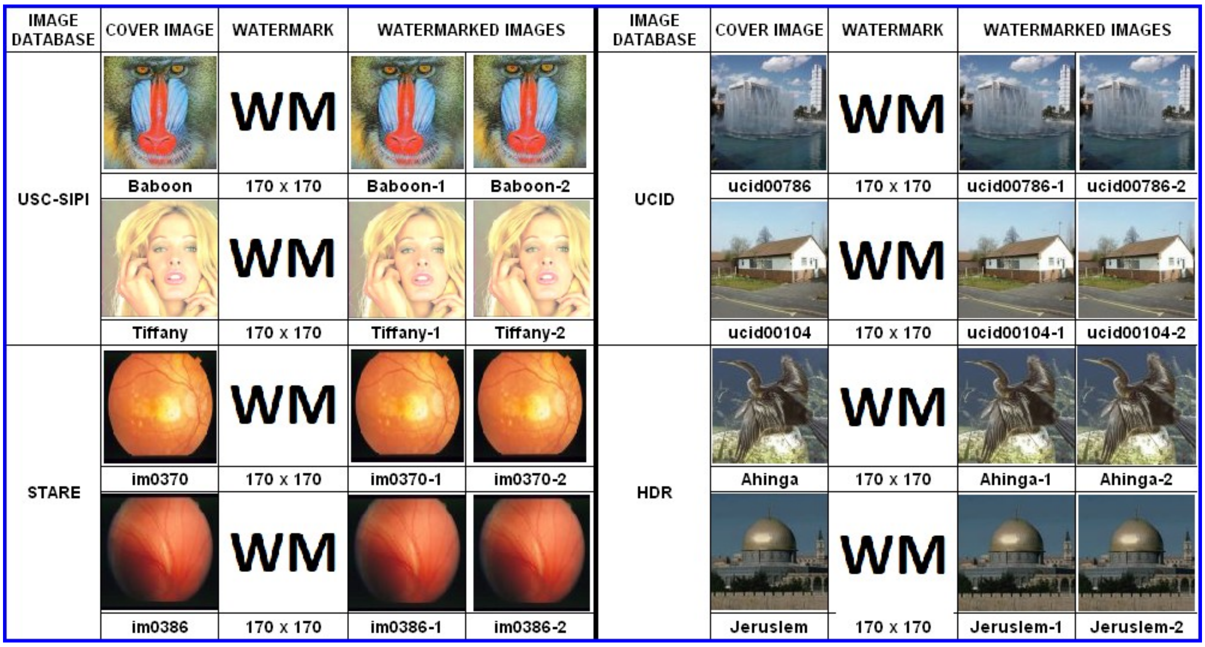
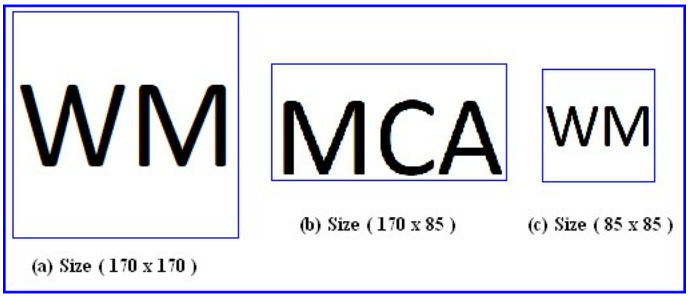


Figure 4.4: Pictorial results of output images in DRWS-LBP



(a) Size (170 x 170)

(b) Size (170 x 85)

(c) Size (85 x 85)

Figure 4.5: Watermark image (logo) with different size used in DRWS-LBP

4.1.3.1 Quality Measurement and Payload Analysis

The fundamental necessities of any watermarking scheme are robustness and imperceptibility. The subjective characteristics of the watermarked images is evaluated in DRWS-LBP and it has been shown in Fig. 4.4. Figure 4.4 illustrate that no visual distortions are detected after embedding maximum payload of $6,93,600$ bits.

DRWS-LBP has been tested by taking sample images from four different standard benchmark image databases and experimental outcomes are noted in Table 4.1. Table 4.1 illustrate that approximately 53 dB PSNR can be achieved after embedding a highest amount of $6,93,600$ bits watermark. Moreover, Q-Index values are close to unity which establishes the acceptability of the proposed scheme.

In addition, the results of objective analysis have been illustrated in Table 4.2 for color images (without any invasion). The Table 4.2 shows the variation of PSNR with respect to number of

Table 4.1: Capacity, PSNR, Q-Index, and Payload values for standard benchmark images in DRWS-LBP

Datasets	Image	Capacity (bits)	Average PSNR (dB)	Average Q-Index	Payloads (bpp)
USC-SIPI [90]	Lena	1,73,400	59.45	0.99999	0.667
		3,46,800	55.35	0.99999	1.34
		6,93,600	53.51	0.99998	2.67
UCID [61]	Jeruslem	1,73,400	59.47	0.99999	0.667
		3,46,800	55.67	0.99999	1.34
		6,93,600	53.17	0.99999	2.67
STARE [89]	Im0005	1,73,400	59.03	0.99999	0.667
		3,46,800	55.39	0.99999	1.34
		6,93,600	53.51	0.99998	2.67
HDR [26]	Taucan	1,73,400	59.38	0.99999	0.667
		3,46,800	55.31	0.99999	1.34
		6,93,600	53.45	0.99999	2.67

Table 4.2: Average PSNR of various yardstick image datasets considering 25 to 100 images DRWS-LBP

Datasets	Image Size	Total Image	Average PSNR
STARE [89]	513×513	25	53.64
		50	53.53
		100	53.34
USC-SIPI [90]	513×513	25	53.64
		50	53.51
		100	53.39
UCID [61]	513×513	25	53.68
		50	53.33
		100	53.12
HDR [26]	513×513	25	53.48
		50	53.24
		100	53.37

images collected from the different image database. From the experimental results, it is clear that, average 53 dB PSNR can be achieved after taking a set of 25, 50 and 100 images at a time respectively.

Moreover, DRWS-LBP is compared with the existing schemes [34, 57, 102, 103] by taking four sample images from SIPI image database [90] and experimental outcomes are noted in Table 4.6. Table 4.6 illustrate that approximately 53 dB PSNR on average can be achieved after embedding a highest amount of 6, 93, 600 bits watermark. Also it has been observed that DRWS-LBP is approximately 18%, 22% and 38% better than Yao et al.'s scheme [103] while achieving 1.2 bpp, 1.6 bpp and 2.2 bpp payload respectively.

Table 4.3: MSE, PSNR, NCC, SSIM, Q-Index and BER results for different benchmark datasets in DRWS-LBP

Image Dataset	Images	MSE	PSNR (dB)	NCC	SSIM	Q-Index	BER
SIPI [90]	Lenna	1.97	53.51	0.99999	0.9871	0.9999	0.01167
	Baboon	1.96	53.51	0.99998	0.9989	0.9999	0.01159
	Tiffany	1.93	53.96	0.99999	0.9960	0.9998	0.01164
	Average	1.95	53.50	0.99999	0.9968	0.9999	0.01163
HDR [26]	anhinga	1.84	53.18	0.99998	0.9980	0.9999	0.01315
	bardowl	1.86	53.45	0.99998	0.9987	0.9999	0.01171
	jeruslem	1.81	53.17	0.99997	0.9971	0.9999	0.01278
	Average	1.84	53.27	0.99998	0.9979	0.9999	0.01327
STARE [89]	im0001	1.86	53.51	0.99999	0.9987	0.9999	0.01155
	im0048	1.84	53.43	0.99999	0.9970	0.9999	0.01163
	im0548	1.87	53.51	0.99998	0.9966	0.9999	0.01156
	Average	1.85	53.48	0.99999	0.9975	0.9999	0.01158
UCID [61]	ucid00148	1.97	53.45	0.99998	0.9922	0.9999	0.01198
	ucid00354	1.92	53.43	0.99998	0.9952	0.9999	0.01154
	ucid00401	1.98	53.49	0.99999	0.9948	0.9999	0.01152
	Average	1.95	53.48	0.99998	0.9936	0.9999	0.01175

Also, the comparison graph concerning PSNR (dB) for Lena, Airplane, Baboon, Boat and Pepper images are shown in Fig. 4.7. From the graphical representation it is seen that DRWS-LBP provides better results in terms of PSNR compared with other existing dual image based schemes [34, 57, 102, 103]. The resemblance with respect to PSNR(dB), Capacity (bpp) and Q-Index with existing techniques based on dual image are presented in Table 4.4.

Table 4.5 represents the experimental outcomes of DRWS-LBP with respect to other existing

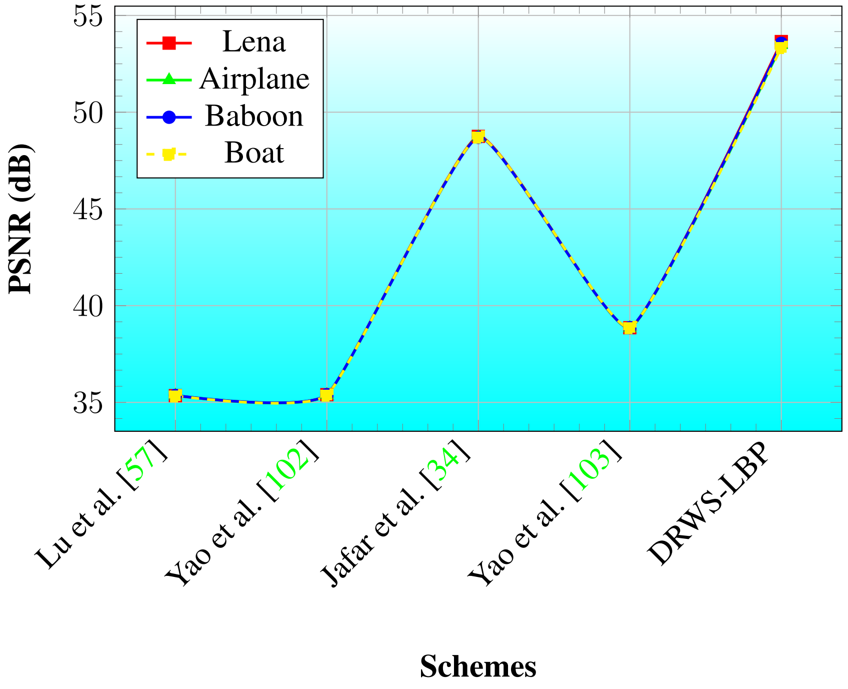


Figure 4.6: Comparison graph in terms of PSNR (dB) with dual image based existing schemes in DRWS-LBP

Table 4.4: Comparison of different dual image based existing methods with respect to PSNR and embedding capacity in DRWS-LBP

Schemes	Lena			Pepper			Barbara			Goldhill		
	PSNR1 (dB)	PSNR2 (dB)	Capacity (bits)	PSNR1 (dB)	PSNR2 (dB)	Capacity (bits)	PSNR1 (dB)	PSNR2 (dB)	Capacity (bits)	PSNR1 (dB)	PSNR2 (dB)	Capacity (bits)
Chang et al. [13]	45.12	45.13	523,264	45.14	45.15	523,264	45.13	45.11	523,264	45.13	45.14	523,264
Chang et al. [10]	48.13	48.14	523,264	48.11	48.11	523,264	48.14	48.14	523,264	48.13	48.13	523,264
Lee et al. [47]	52.38	52.39	393,276	52.38	52.39	393,490	52.39	52.39	393206	52.38	52.39	393212
Lee & Huang [48]	49.76	49.56	1.07 (bpp)	49.75	49.66	1.07 (bpp)	49.75	49.67	1.07 (bpp)	49.77	49.67	1.07 (bpp)
Chang et al. [14]	39.89	39.89	802,895	39.94	39.94	799,684	39.89	39.89	802,888	39.90	39.90	802,698
Qin et al. [71]	52.11	41.58	557,052	51.25	41.52	557,052	52.12	41.58	557,052	52.12	41.58	557,052
Lu et al. [56]	49.20	49.21	524,288	49.19	49.21	524,288	49.22	49.20	524,288	49.23	49.18	524,288
Jung et al. [41]	48.18	47.20	519,180	48.18	48.18	519,180	48.15	48.13	519,180	48.19	47.21	519,180
Jafar et al. [34]	48.70	48.71	650,369	48.70	48.71	627,637	48.70	48.71	650,781	48.72	48.71	650,726
Jana et al. [37]	52.71	52.81	74,752	52.67	52.72	73,728	52.70	52.76	74,752	52.73	52.78	74,752
DRWS-LBP	53.57	53.43	693,600	53.57	53.45	693,600	53.59	53.47	693,600	53.56	53.45	693,600

LBP based schemes. Also, the comparison graph concerning PSNR (dB) for Lena, Airplane, Baboon, Boat and Pepper images are presented in Fig. 4.7. From the graphical representations it is seen that DRWS-LBP provides better result in terms of PSNR compared with other existing LBP based schemes [65, 68, 94, 107].

Table 4.5: Comparison graph in terms of PSNR (dB) with LBP based existing schemes in DRWS-LBP

Schemes	Parah et al. [65]	Wenyin et al. [94]	Pinjari et al. [68]	Zhang et al. [107]	DRWS-LBP	Improvement % w.r.t [107]
Lena	40.53	42.64	43.54	44.02	53.51	21.55
Airplane	40.99	41.37	43.59	44.32	53.51	20.73
Baboon	40.08	42.37	43.55	44.46	53.96	21.36
Boat	41.47	41.28	43.64	43.88	53.50	21.92
Pepper	40.71	42.52	43.53	44.61	53.13	19.09
Average	40.76	42.04	43.57	44.25	53.52	20.93

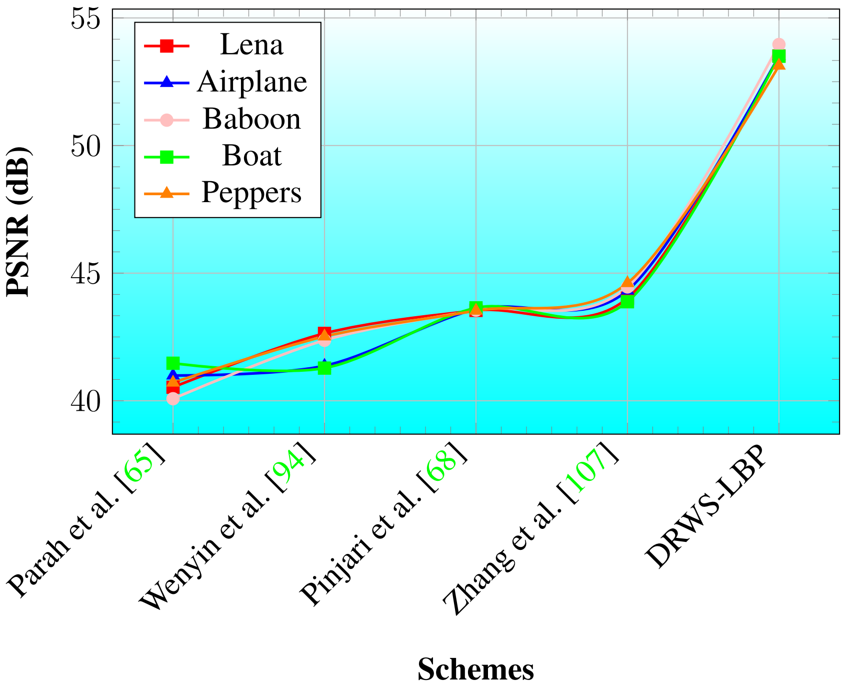


Figure 4.7: Comparison result with LBP based existing schemes in terms of PSNR (dB) in DRWS-LBP

4.1.3.2 Robustness Analysis

Table 4.3 shows the evaluation results of DRWS-LBP in terms of various evaluation schemes like MSE, PSNR, NCC, SSIM, Q-Index and BER on four different benchmark image datasets.

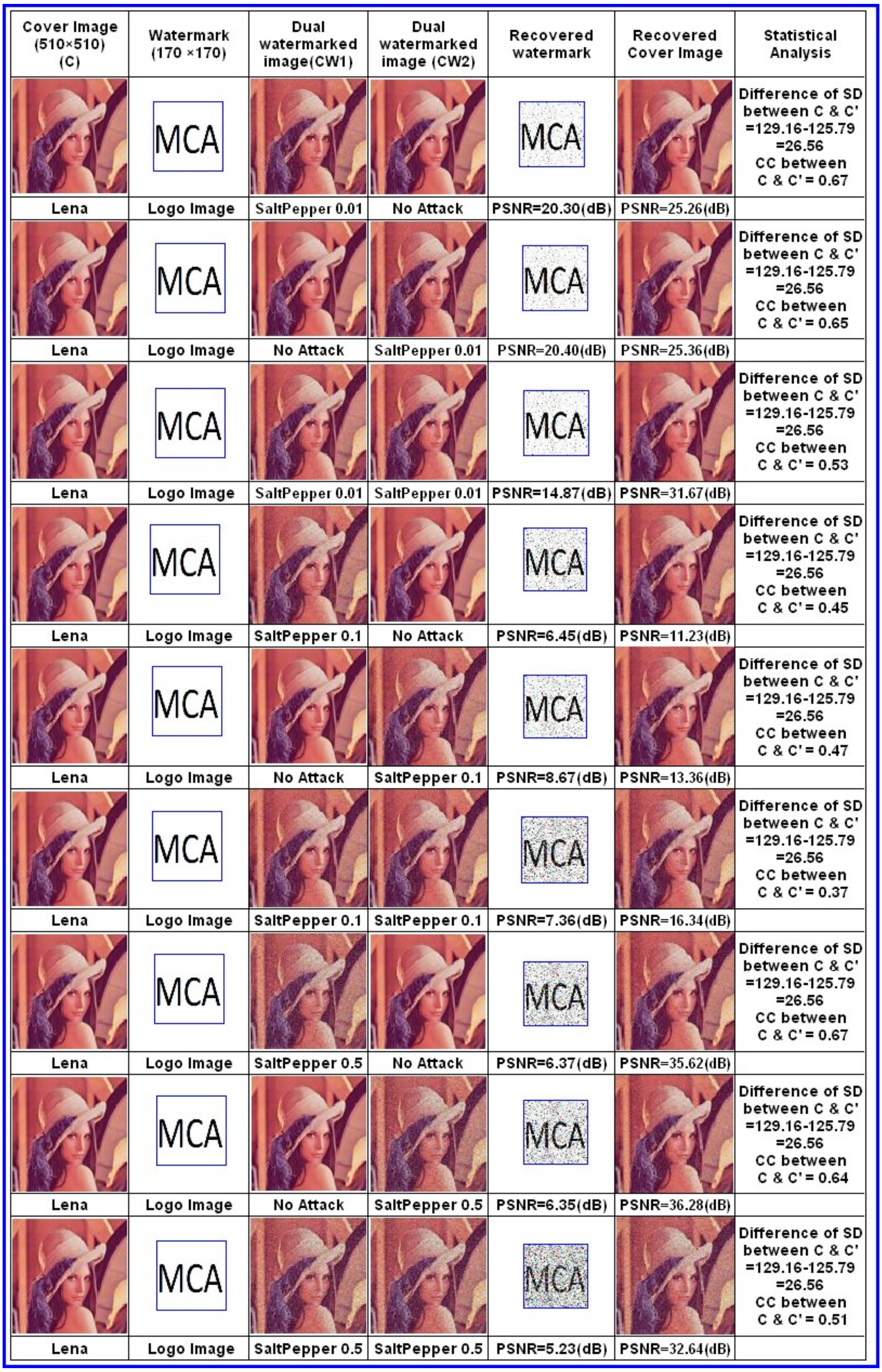
From Table 4.3, it is found that the average visual quality measured by PSNR for the aforesaid image databases is greater than 53 dB. Also the NCC, SSIM and Q-Index values of the DRWS-LBP are nearer to unity, which prove the effectiveness of the designed algorithm. The BER results prove that the developed scheme is robust.

Table 4.6: Comparison results of PSNR and Payload with existing dual image based schemes in DRWS-LBP

Image	Payload (bpp)	Lu et al. [57]	Yao et al. [102]	Jafar et al. [34]	Yao et al. [103]	DRWS-LBP	Improvement % w.r.t [103]
Lena	1.20	47.33	47.51	...	48.71	57.69	18.43
	1.60	41.70	41.75	...	45.34	55.35	22.07
	2.20	35.35	35.40	48.75	38.86	53.65	38.45
Baboon	1.20	47.33	47.52	...	48.72	57.87	18.54
	1.60	41.70	41.75	...	45.34	55.35	22.57
	2.20	35.36	35.39	48.71	38.85	53.47	38.05
Barbara	1.20	47.34	47.52	...	48.72	57.69	18.67
	1.60	41.70	41.78	...	45.33	55.46	22.63
	2.20	35.36	35.40	48.70	38.86	53.56	38.53
Goldhil	1.20	47.33	47.51	...	48.71	57.77	18.49
	1.60	41.70	41.75	...	45.34	55.46	22.41
	2.20	35.34	35.38	48.72	38.87	53.35	38.46

4.1.3.3 Tamper Detection and Recovery

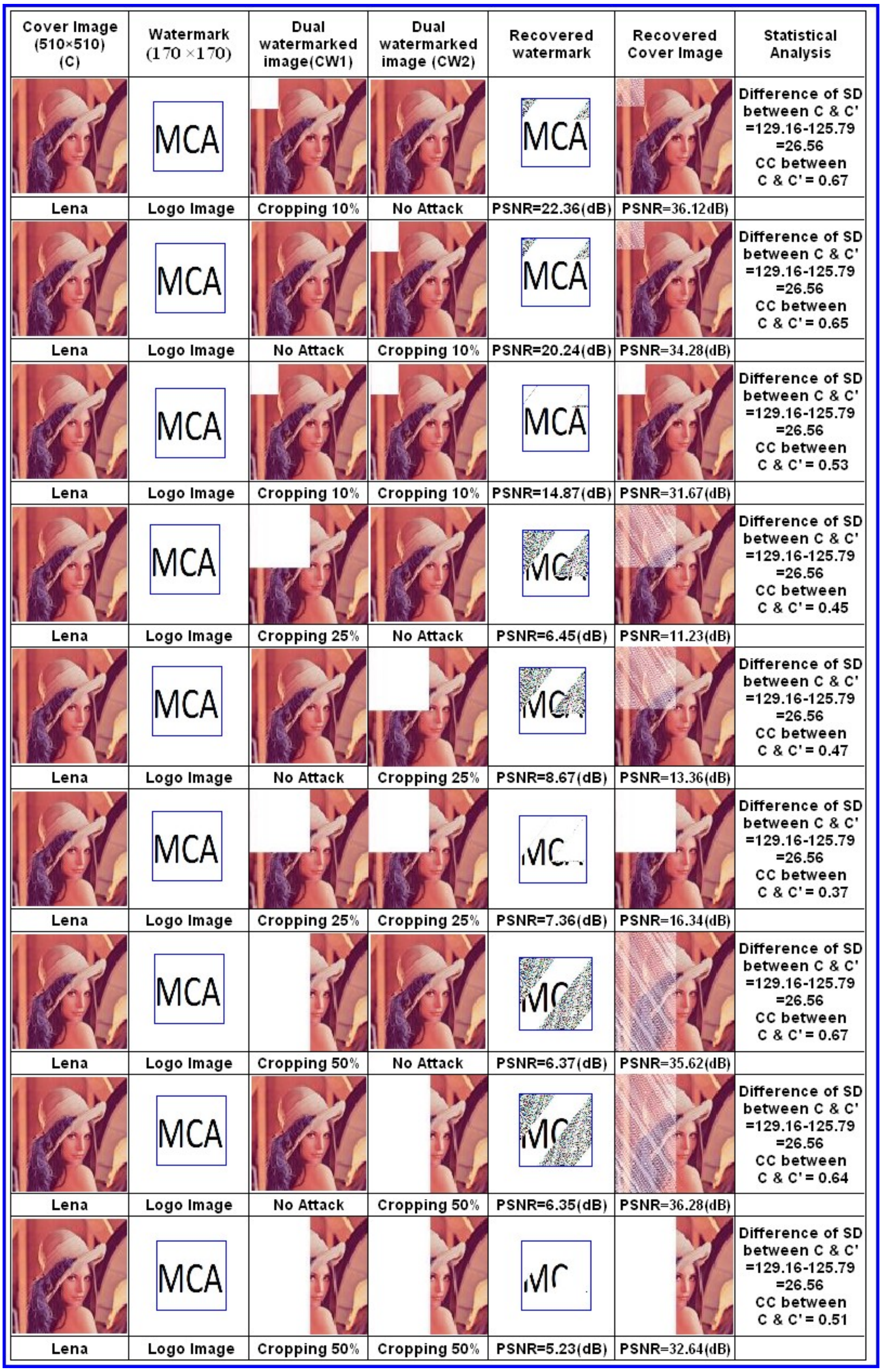
Robustness of proposed scheme is analyzed by evaluating the quality metrics such as PSNR, SSIM, Q-Index, NCC and BER in presence of salt and pepper noise, cropping and copy-move forgery attacks. The Fig. 4.8, Fig. 4.9 and Fig. 4.10 represent the results after applying salt and pepper noise, cropping and copy-move forgery attack with different noise density level respectively. It is clear that after extraction, the objective quality of the extracted watermark



Cover Image (510×510) (C)	Watermark (170 ×170)	Dual watermarked image(CW1)	Dual watermarked image (CW2)	Recovered watermark	Recovered Cover Image	Statistical Analysis
Lena	Logo Image	SaltPepper 0.01	No Attack	PSNR=20.30(dB)	PSNR=25.26(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.67
Lena	Logo Image	No Attack	SaltPepper 0.01	PSNR=20.40(dB)	PSNR=25.36(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.65
Lena	Logo Image	SaltPepper 0.01	SaltPepper 0.01	PSNR=14.87(dB)	PSNR=31.67(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.53
Lena	Logo Image	SaltPepper 0.1	No Attack	PSNR=6.45(dB)	PSNR=11.23(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.45
Lena	Logo Image	No Attack	SaltPepper 0.1	PSNR=8.67(dB)	PSNR=13.36(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.47
Lena	Logo Image	SaltPepper 0.1	SaltPepper 0.1	PSNR=7.36(dB)	PSNR=16.34(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.37
Lena	Logo Image	SaltPepper 0.5	No Attack	PSNR=6.37(dB)	PSNR=35.62(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.67
Lena	Logo Image	No Attack	SaltPepper 0.5	PSNR=6.35(dB)	PSNR=36.28(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.64
Lena	Logo Image	SaltPepper 0.5	SaltPepper 0.5	PSNR=5.23(dB)	PSNR=32.64(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.51

Figure 4.8: Effect of salt pepper noise on Lena image in DRWS-LBP

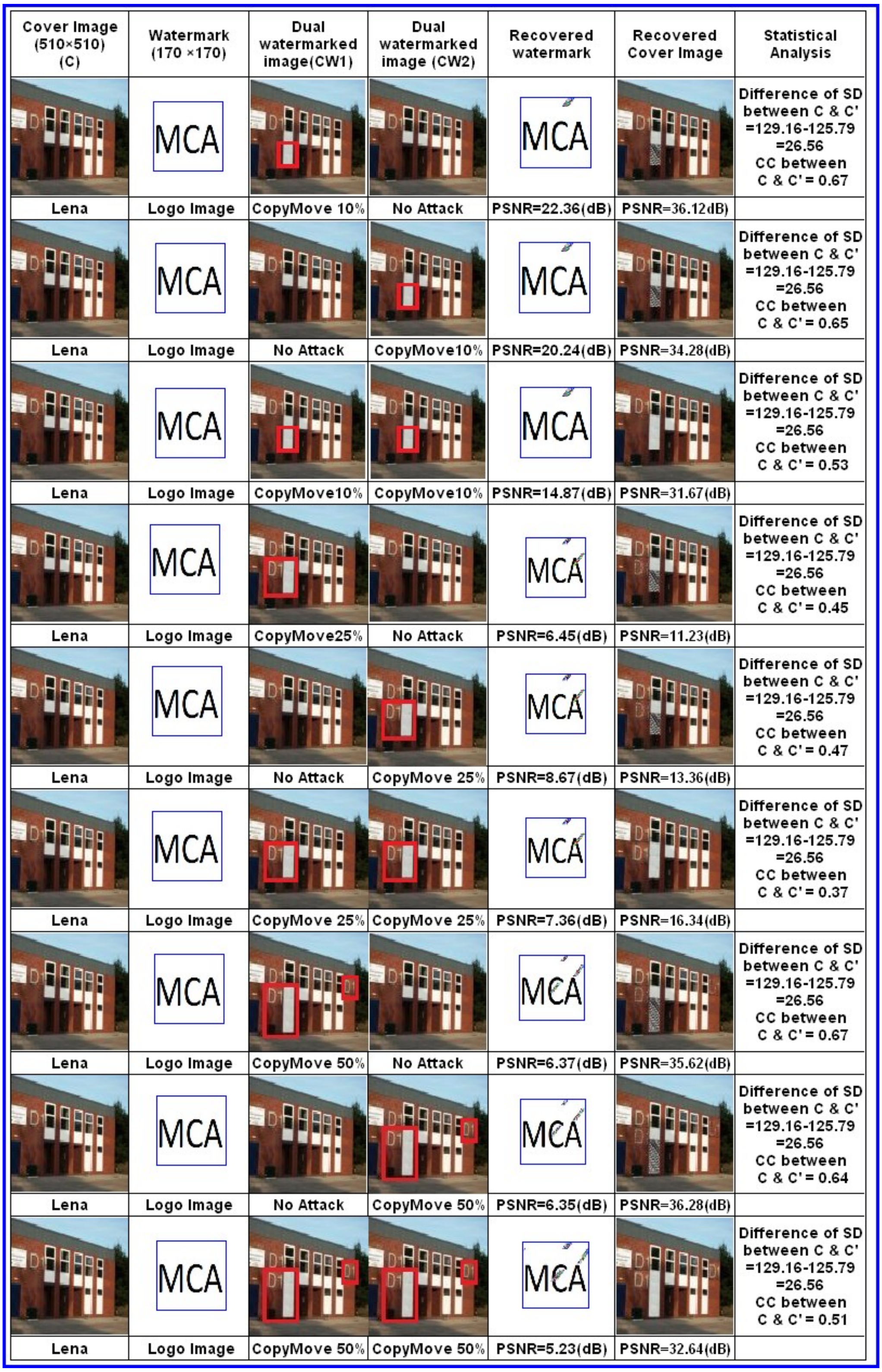
slightly change whereas, the tamper location of the recovered cover image has been identified successfully. Also the results of statistical analysis (SD and CC) shows the robustness of the



Cover Image (510×510) (C)	Watermark (170 ×170)	Dual watermarked image(CW1)	Dual watermarked image (CW2)	Recovered watermark	Recovered Cover Image	Statistical Analysis
Lena	Logo Image	Cropping 10%	No Attack	PSNR=22.36(dB)	PSNR=36.12(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.67
Lena	Logo Image	No Attack	Cropping 10%	PSNR=20.24(dB)	PSNR=34.28(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.65
Lena	Logo Image	Cropping 10%	Cropping 10%	PSNR=14.87(dB)	PSNR=31.67(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.53
Lena	Logo Image	Cropping 25%	No Attack	PSNR=6.45(dB)	PSNR=11.23(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.45
Lena	Logo Image	No Attack	Cropping 25%	PSNR=8.67(dB)	PSNR=13.36(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.47
Lena	Logo Image	Cropping 25%	Cropping 25%	PSNR=7.36(dB)	PSNR=16.34(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.37
Lena	Logo Image	Cropping 50%	No Attack	PSNR=6.37(dB)	PSNR=35.62(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.67
Lena	Logo Image	No Attack	Cropping 50%	PSNR=6.35(dB)	PSNR=36.28(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.64
Lena	Logo Image	Cropping 50%	Cropping 50%	PSNR=5.23(dB)	PSNR=32.64(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.51

Figure 4.9: Effect of cropping attacks on Lena image in DRWS-LBP

proposed scheme. The different objective metrics are presented in Table 4.7 when extraction is performed from tampered image. From Table 4.7, it is noted that the less BER values and near



Cover Image (510×510) (C)	Watermark (170 ×170)	Dual watermarked image(CW1)	Dual watermarked image (CW2)	Recovered watermark	Recovered Cover Image	Statistical Analysis
Lena	Logo Image	CopyMove 10%	No Attack	PSNR=22.36(dB)	PSNR=36.12(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.67
Lena	Logo Image	No Attack	CopyMove10%	PSNR=20.24(dB)	PSNR=34.28(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.65
Lena	Logo Image	CopyMove10%	CopyMove10%	PSNR=14.87(dB)	PSNR=31.67(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.53
Lena	Logo Image	CopyMove25%	No Attack	PSNR=6.45(dB)	PSNR=11.23(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.45
Lena	Logo Image	No Attack	CopyMove 25%	PSNR=8.67(dB)	PSNR=13.36(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.47
Lena	Logo Image	CopyMove 25%	CopyMove 25%	PSNR=7.36(dB)	PSNR=16.34(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.37
Lena	Logo Image	CopyMove 50%	No Attack	PSNR=6.37(dB)	PSNR=35.62(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.67
Lena	Logo Image	No Attack	CopyMove 50%	PSNR=6.35(dB)	PSNR=36.28(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.64
Lena	Logo Image	CopyMove 50%	CopyMove 50%	PSNR=5.23(dB)	PSNR=32.64(dB)	Difference of SD between C & C' =129.16-125.79 =26.56 CC between C & C' = 0.51

Figure 4.10: Effect of copy-move forgery attacks on Lena image in DRWS-LBP

unity Q-Index and NCC indicate the robustness of the proposed method during these invasion. Again it is clear from the Table 4.7 that robustness of the DRWS-LBP varies inversely with the

Table 4.7: PSNR, SSIM, Q-Index, NCC and BER results of distorted watermark images due to salt pepper noise, cropping and copy-move forgery attacks in DRWS-LBP

Noise	Sample	Perturbation	PSNR (dB)		SSIM		Q-Index		NCC		BER	
			CI	WI	CI	WI	CI	WI	CI	WI	CI	WI
Salt and Pepper	C1	0.01	25.26	20.30	77.64	67.92	0.9541	0.9739	0.9949	0.9940	0.0016	0.0065
		0.1	21.36	16.43	57.69	53.63	0.8926	0.9372	0.9876	0.9855	0.0040	0.0157
		0.5	18.62	13.55	39.02	44.93	0.8126	0.8753	0.9769	0.9714	0.0075	0.0308
	C2	0.01	25.36	20.40	77.57	68.32	0.9542	0.9737	0.9950	0.9942	0.0015	0.0064
		0.1	21.34	16.45	57.67	53.66	0.8929	0.9371	0.9877	0.9859	0.0042	0.0159
		0.5	18.62	13.58	39.06	44.96	0.8125	0.8754	0.9762	0.9716	0.0076	0.0307
	C1 & C2	0.01	22.23	17.55	61.37	57.53	0.9110	0.9479	0.9898	0.9888	0.0030	0.0135
		0.1	19.32	15.53	42.14	46.92	0.8926	0.8726	0.9876	0.9855	0.0057	0.0155
		0.5	15.60	12.68	27.54	34.56	0.6961	0.7123	0.9546	0.9483	0.0151	0.0543
Cropping	C1	10 %	21.01	16.20	91.73	90.12	0.8921	0.9240	0.9871	0.9846	0.0043	0.0144
		25 %	17.31	10.27	79.88	73.63	0.7809	0.7184	0.9728	0.9381	0.0109	0.0448
		50 %	13.46	5.82	59.64	47.89	0.5441	0.4087	0.9425	0.8189	0.0231	0.1021
	C2	10 %	21.03	16.21	91.77	90.15	0.8915	0.9241	0.9873	0.9844	0.0045	0.0145
		25 %	17.32	10.26	79.87	73.67	0.7806	0.7182	0.9724	0.9386	0.0107	0.0442
		50 %	13.43	5.81	59.61	47.87	0.5442	0.4089	0.9423	0.8183	0.0232	0.1028
	C1 & C2	10 %	17.80	14.31	90.35	86.44	0.7992	0.8816	0.9751	0.9760	0.0095	0.0183
		25 %	14.56	11.76	73.45	63.49	0.4538	0.7456	0.9456	0.9358	0.0153	0.0453
		50 %	10.46	8.56	56.23	47.84	0.3296	0.5556	0.9109	0.9082	0.0461	0.0761
Copy Move Forgery	C1	5 %	50.98	22.06	99.67	96.51	0.9999	0.9804	0.9999	0.9962	0.0003	0.0037
		10 %	49.35	20.94	99.47	94.56	0.9998	0.9748	0.9999	0.9938	0.0009	0.0048
		20 %	47.38	18.12	99.32	91.56	0.9997	0.9556	0.9999	0.9901	0.0014	0.0081
	C2	5 %	50.97	22.05	99.65	96.54	0.9999	0.9804	0.9999	0.9965	0.0002	0.0038
		10 %	49.33	20.91	99.44	94.54	0.9998	0.9744	0.9999	0.9936	0.0010	0.0046
		20 %	47.36	18.13	99.33	91.54	0.9997	0.9558	0.9999	0.9903	0.0015	0.0082
	C1 & C2	5 %	48.43	21.77	99.52	96.20	0.9998	0.9805	0.9999	0.9957	0.0011	0.0044
		10 %	46.35	18.94	99.12	93.48	0.9997	0.9604	0.9997	0.9902	0.0018	0.0076
		20 %	44.73	16.90	98.89	89.84	0.9995	0.9403	0.9994	0.9868	0.0025	0.0107

noise density.

The algorithmic complexity of any watermarking scheme is a significant parameter in current research scenario. The execution time of the DRWS-LBP has been tested by using system clock and compared with some recent works [10, 13, 14, 34, 56, 65, 71, 79, 91] and the comparative outcomes are presented in Table 4.8. It is observed that DRWS-LBP requires 0.521 seconds for total execution which is relatively better than all the existing schemes. During embedding only 0.423 seconds time is acquired to insert a (170×170) i.e., $6,93,600$ bits watermark into

Table 4.8: Comparison table in terms of execution time in DRWS-LBP

Schemes	Number of blocks	Embedding time (sec)	Extraction time (sec)	Total time (sec)
Chang et al. [13]	131,072	2.22	3.94	6.16
Chang et al. [10]	131,072	1.06	0.26	1.32
Chang et al. [14]	131,072	1.05	0.52	1.57
Qin et al. [71]	262,144	1.18	0.52	1.70
Lu et al. [56]	131,072	1.10	0.65	2.75
Verma et al. [91]	196,608	0.5173	0.5989	1.1162
Jafar et al. [34]	131,072	0.46	0.17	0.63
Parah et al. [65]	12,288	0.59	0.0624	0.6524
Su et al. [79]	87,723	0.1948	5.8023	5.9972
DRWS-LBP	87,723	0.423	0.098	0.521

(512×512) cover image and 0.098 seconds time is acquired at the time of extraction. The lesser execution time in DRWS-LBP is achieved due to simple algebraic manipulations and the threading concept of Java. To determine the algorithmic complexity, a cover image of size $(M \times N)$ is considered. Time complexity for doing the operations described in Algorithm 4.1 is $\mathcal{O}(MN)$. On the other hand, at the time of extraction, the complexity is $\mathcal{O}(MN)$, considering Algorithm 4.2.

4.2 RWS using LBP and HC [4] [5]

In this investigation, an RWS has been introduced for an interpolated color image to verify image integrity, authenticity and to correct errors using LBP and $(15, 11)$ HC respectively. The LBP vector values have been calculated using (2×2) original pixel block of cover image. Watermark is inserted within the LSB of interpolated pixels. Here, the LBP operator is used to solve image authentication and tamper detection problem, whereas HC is used to detect and correct the error which may occurs in embedding phase. Some standard NIST recommended steganalysis have been performed to evaluate the robustness and imperceptibility. It is observed

[4] Review submitted in **Multimedia Tools and Application**, Springer: **Impact Factor: 1.541** with title *A Reversible watermarking scheme for interpolated color image based on Local Binary Pattern and Hamming Code*

[5] Published in **Proceedings of International Conference on Communication, Devices and Computing (ICCDC 2017)**, Springer: pp 59-67, ISBN 978-981-10-8584-0 with title *Hamming Code-Based Watermarking Scheme for Image Authentication and Tampered Detection*

that the RWS-LBP-HC is secure and robust against various attacking environment. It can also detect tampered locations and can verify the ownership of an image. Experimental results are compared with the existing watermarking schemes to establish the superiority of the RWS-LBP-HC. It also shows good perceptible quality with a high payload and less computational cost. The RWS-LBP-HC has been described in three subsection (4.2.1, 4.2.2 and 4.2.3).

4.2.1 Pre-Embedding Phase

Pre-embedding phase describes the detailed image interpolation process that has been applied to enlarge the image. The diagrammatic description of the interpolation technique is shown in Fig. 4.11. Here, a (2×2) cover image block (Fig. 4.11(a)) has been considered to form an interpolated image of size (4×4), shown in Fig. 4.11(b). After interpolation, each block contains 16 pixels. The original pixels are belonging in each corner of blocks (C_i shown in red color). The remaining pixels are separated into two regions central region (X_i shown in yellow color) and border region (P_i shown in green color). The watermark bits are embedded in 2 LSB's of border region pixels. The central region pixels contain 4-bit tamper detection code (TDC) generated through LBP and 4 redundant bits are generated through $(15, 11)$ Hamming code for error detection and correction.

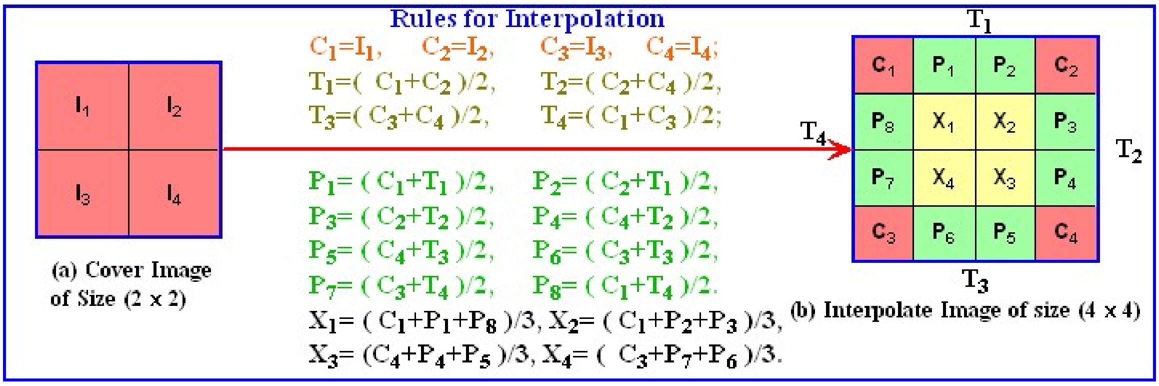

Rules for Interpolation
$C_1=I_1$, $C_2=I_2$, $C_3=I_3$, $C_4=I_4$;
$T_1=(C_1+C_2)/2$, $T_2=(C_2+C_4)/2$,
$T_3=(C_3+C_4)/2$, $T_4=(C_1+C_3)/2$;
$P_1=(C_1+T_1)/2$, $P_2=(C_2+T_1)/2$,
$P_3=(C_2+T_2)/2$, $P_4=(C_4+T_2)/2$,
$P_5=(C_4+T_3)/2$, $P_6=(C_3+T_3)/2$,
$P_7=(C_3+T_4)/2$, $P_8=(C_1+T_4)/2$.
$X_1=(C_1+P_1+P_8)/3$, $X_2=(C_1+P_2+P_3)/3$,
$X_3=(C_4+P_4+P_5)/3$, $X_4=(C_3+P_7+P_6)/3$.
I_1 I_2
I_3 I_4
(a) Cover Image of Size (2 x 2)
T_1
T_2
T_3
T_4
C_1 P_1 P_2 C_2
P_8 X_1 X_2 P_3
P_7 X_4 X_3 P_4
C_3 P_6 P_5 C_4
(b) Interpolate Image of size (4 x 4)

Figure 4.11: Block diagram of image interpolation phase in RWS-LBP-HC

4.2.2 Watermark Embedding Phase

Here, a LBP based RWS for an interpolated cover image has been described. Figure. 4.12 depicts the detail embedding and extraction procedure of RWS-LBP-HC. At first, color cover image CI is divided into R, G and B color components as CI_R, CI_G and CI_B. Then color blocks

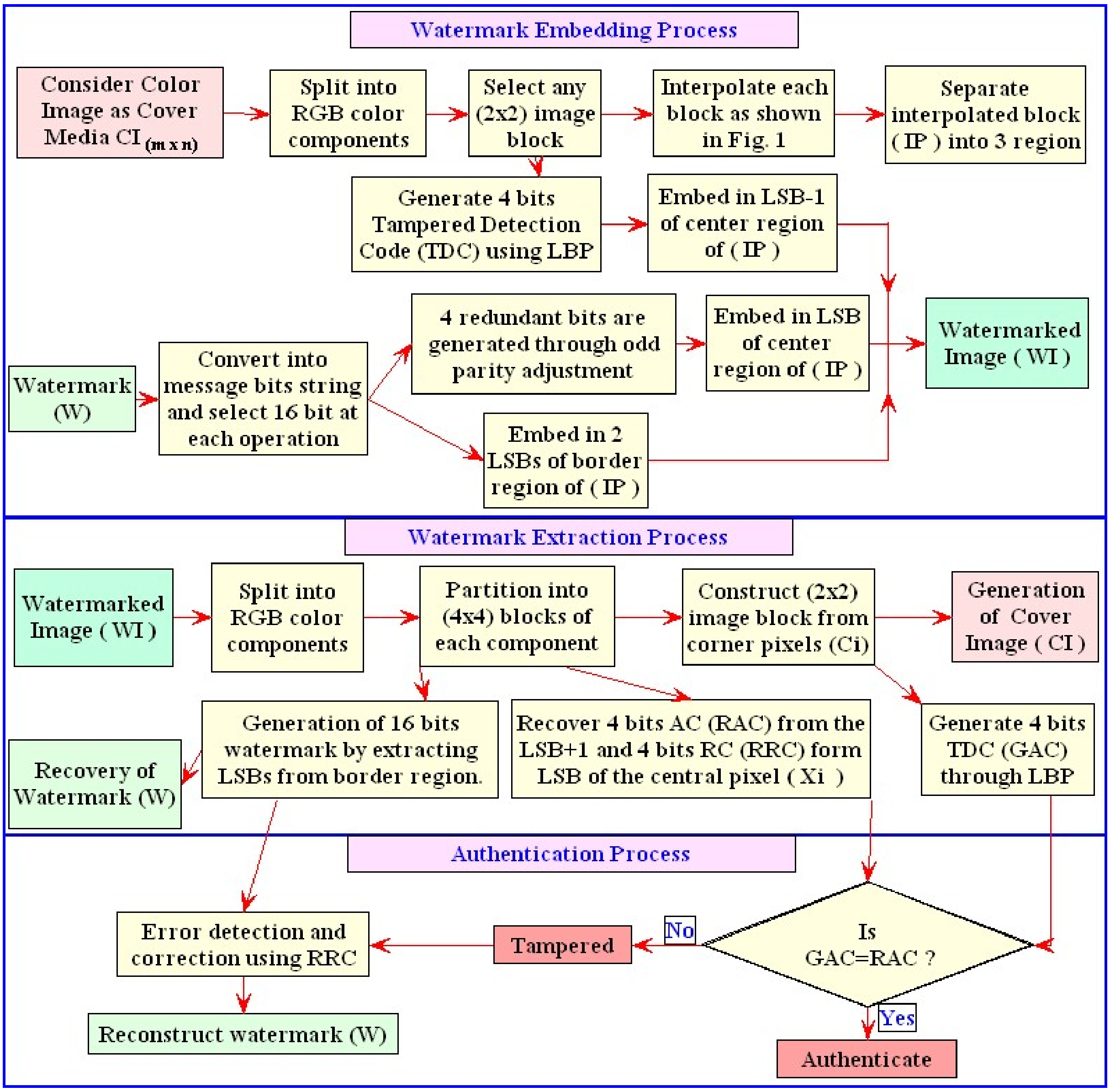
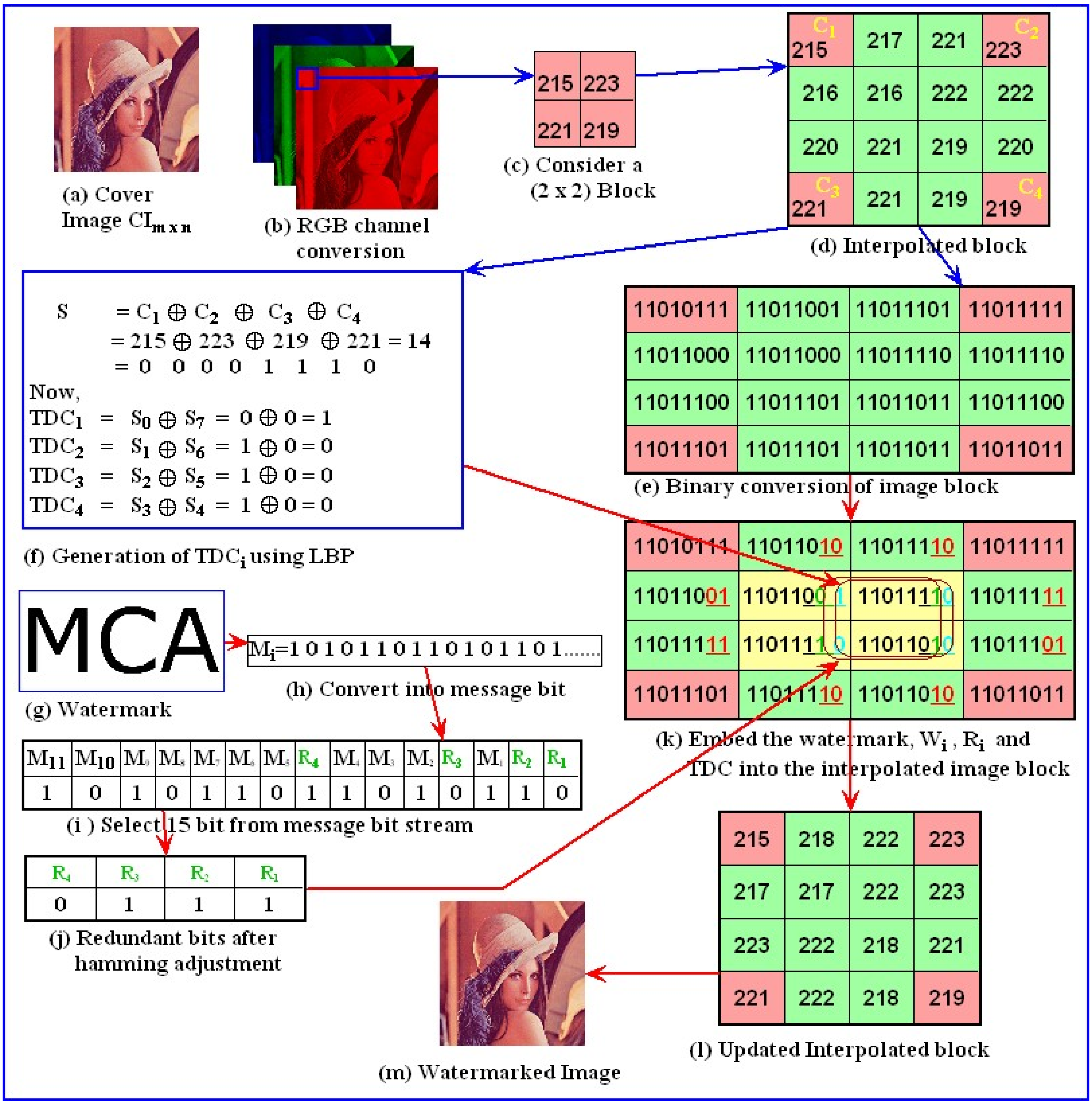

Watermark Embedding Process
Consider Color Image as Cover Media CI (m x n)
Split into RGB color components
Select any (2x2) image block
Interpolate each block as shown in Fig. 1
Separate interpolated block (IP) into 3 region
Generate 4 bits Tampered Detection Code (TDC) using LBP
Embed in LSB-1 of center region of (IP)
Watermark (W)
Convert into message bits string and select 16 bit at each operation
4 redundant bits are generated through odd parity adjustment
Embed in LSB of center region of (IP)
Watermarked Image (WI)
Embed in 2 LSBs of border region of (IP)
Watermark Extraction Process
Watermarked Image (WI)
Split into RGB color components
Partition into (4x4) blocks of each component
Construct (2x2) image block from corner pixels (Ci)
Generation of Cover Image (CI)
Recovery of Watermark (W)
Generation of 16 bits watermark by extracting LSBs from border region.
Recover 4 bits AC (RAC) from the LSB+1 and 4 bits RC (RRC) form LSB of the central pixel (Xi)
Generate 4 bits TDC (GAC) through LBP
Authentication Process
Error detection and correction using RRC
Tampered
No
Is GAC=RAC ?
Yes
Reconstruct watermark (W)
Authenticate

Figure 4.12: Block diagram of watermarking process RWS-LBP-HC

are interpolated into ICI_R, ICI_G, and ICI_B by applying interpolation rules shown in Fig. 4.11. After that a (4×4) pixel block from ICI_R are considered. Then XOR operation is performed among the C_i and stored into S after converting it into 4 bits binary string. Again an XOR operation is performed with two specific bits of S to generate 4-bit TDC by appending one after another. Now, W is converted into a message bits string (M_i) and first 16-bit watermark from M_i are considered and are stored into watermark bits (WB). Two bit from WB are extracted and are embedded in the LSB of the border pixels (P_i for i= 1 to 8) of $(RI_{4\times4})$. After that, from the WB, four redundant bits (R_i) are created using (15, 11) Hamming code. One bit from TDC and 1 bit from R_i are extracted and are embedded in LSB, LSB-1 position of the middle



(a) Cover Image $CI_{m \times n}$

(b) RGB channel conversion

215	223
221	219

(c) Consider a (2 x 2) Block

215	217	221	223
216	216	222	222
220	221	219	220
221	221	219	219

(d) Interpolated block

11010111	11011001	11011101	11011111
11011000	11011000	11011110	11011110
11011100	11011101	11011011	11011100
11011101	11011101	11011011	11011011

(e) Binary conversion of image block

$S = C_1 \oplus C_2 \oplus C_3 \oplus C_4$
$= 215 \oplus 223 \oplus 219 \oplus 221 = 14$
$= 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0$

Now,
$TDC_1 = S_0 \oplus S_7 = 0 \oplus 0 = 1$
$TDC_2 = S_1 \oplus S_6 = 1 \oplus 0 = 0$
$TDC_3 = S_2 \oplus S_5 = 1 \oplus 0 = 0$
$TDC_4 = S_3 \oplus S_4 = 1 \oplus 0 = 0$

(f) Generation of TDC_i using LBP

(g) Watermark

M_i=1 0 1 0 1 1 0 1 1 0 1 0 1 1 0 1.......

(h) Convert into message bit

M_{11}	M_{10}	M_9	M_8	M_7	M_6	M_5	R_4	M_4	M_3	M_2	R_3	M_1	R_2	R_1
1	0	1	0	1	1	0	1	1	0	1	0	1	1	0

(i) Select 15 bit from message bit stream

R_4	R_3	R_2	R_1
0	1	1	1

(j) Redundant bits after hamming adjustment

11010111	11011010	11011110	11011111
11011001	11011001	11011110	11011111
11011111	11011110	11011010	11011101
11011101	11011110	11011010	11011011

(k) Embed the watermark, W_i, R_i and TDC into the interpolated image block

215	218	222	223
217	217	222	223
223	222	218	221
221	222	218	219

(l) Updated Interpolated block

(m) Watermarked Image

Figure 4.13: Numerical illustration of watermark embedding phase in RWS-LBP-HC

pixels (X_i for i= 1 to 4) of ($ICI_{R(4\times4)}$) respectively. Then the above procedure is repeated for all other blocks in ICI_R and $ICI_R^{'}$ is generated from the updated ICI_R pixel blocks. Then, the same operation is performed to generate $ICI_G^{'}$ and $ICI_B^{'}$ from the updated ICI_G and ICI_B pixel blocks. Finally, the interpolated watermarked image (IWI) is generated using the modified $ICI_R^{'}$, $ICI_G^{'}$ and $ICI_B^{'}$ pixel blocks. The embedding algorithm of RWS-LBP-HC is presented in Algorithm 4.3 and a numerical illustration of RWS-LBP-HC is depicted in Fig. 4.13.

Algorithm 4.3: RWS-LBP-HC: Watermark Embedding Algorithm

```
Input  : Cover image (CI), Watermark image (W)
Output: Watermarked Image (WI)

Step 1:  Cover image (CI) are separated into R, G and B color ingredient as ($CI_R$), ($CI_G$) and ($CI_B$).
Step 2:  Each color blocks are interpolated into ($ICI_R$), ($ICI_G$) and ($ICI_B$) by applying interpolation rules shown in Fig. 4.11.
Step 3:  Consider a $(4 \times 4)$ pixel block from ($ICI_R$).
Step 4:  Corner pixels of the selected block as $C_1$, $C_2$, $C_3$ $and$ $C_4$ are taken.
Step 5:  Bits of $C_1$, $C_2$, $C_3$ $and$ $C_4$ are XORed and are stored to form a 8 bit string S.
Step 6:  Two specific bits of S are XORed and are append these to tamper detection code (TDC).
         $TDC_1 = S_0 \oplus S_7; TDC_2 = S_1 \oplus S_6; TDC_3 = S_2 \oplus S_5; TDC_4 = S_3 \oplus S_4;$
Step 7:  First 16 watermark bits from Watermark image (W) are taken.
         WB = getFirstNBitsWatermark(16)
Step 8:  Extract 2 bits from watermark bits (WB) and embed in LSB of the border pixels ($P_i$) of ($RI_{4\times4}$) except the corner pixels ($I_i$) for i= 1 to 8.
Step 9:  Extract 1 bit from AC and embed in LSB position of the middle pixels ($X_i$) of ($RI_{4\times4}$)
Step 10: Redundant bits $R_1$, $R_2$, $R_3$ $and$ $R_4$ from WB are generated using Hamming code
         $R_1$ = XOR of 1,3,5,7,9,11,13 and 15-th bits of WB
         $R_2$ = XOR of 2,3,6,7,10,11,14 and 15-th bits of WB
         $R_3$ = XOR of 4,5,6,12,13,14 and 15-th bits of WB
         $R_4$ = XOR of 8,9,10,11,12,13,14 and 15-th bits of WB
Step 11: $R_1$, $R_2$, $R_3$ $and$ $R_4$ in the LSB-1 position of middle pixels ($X_i$), for i = 1 to 4 are embedded.
Step 12: Step-3 to Step-11 for all $(4 \times 4)$ image blocks of $ICI_R$ are repeated.
Step 13: $ICI'_R$from the updated $ICI_R$are generated.
Step 14: Step-3 to Step-13 for the image blocks of $ICI_G$ and $ICI_B$ are repeated.
Step 15: $ICI'_G$ and $ICI'_B$ from the updated $ICI_G$ and $ICI_B$ are generated.
Step 16: Watermarked Image (WI)using the modified $ICI_R$, $ICI_G$ and $ICI_B$ are generated.
```

4.2.3 Watermark Extraction and Recovery Phase

In this section, the detail extraction procedure of RWS-LBP-HC has been discussed. First, a watermarked image (IWI) is considered as input image and then it is divided into R, G and B color compopnents as ICI'_R, ICI'_G and ICI'_B. A (4×4) pixel block is considered from (ICI'_R). Then, corner pixels of the selected block are taken as C'_i for $i = 1$ to 4. An XOR operation is performed with all C'_i and it is stored in S' after converting it into a 8-bit binary string. Again XOR operation is performed with two selective bits of S' and is appended to TDC'. Then, 2 LSB bit of the border pixels (P'_i for $i = 1$ to 8) of $ICI'_{R(4\times4)}$ are collected and

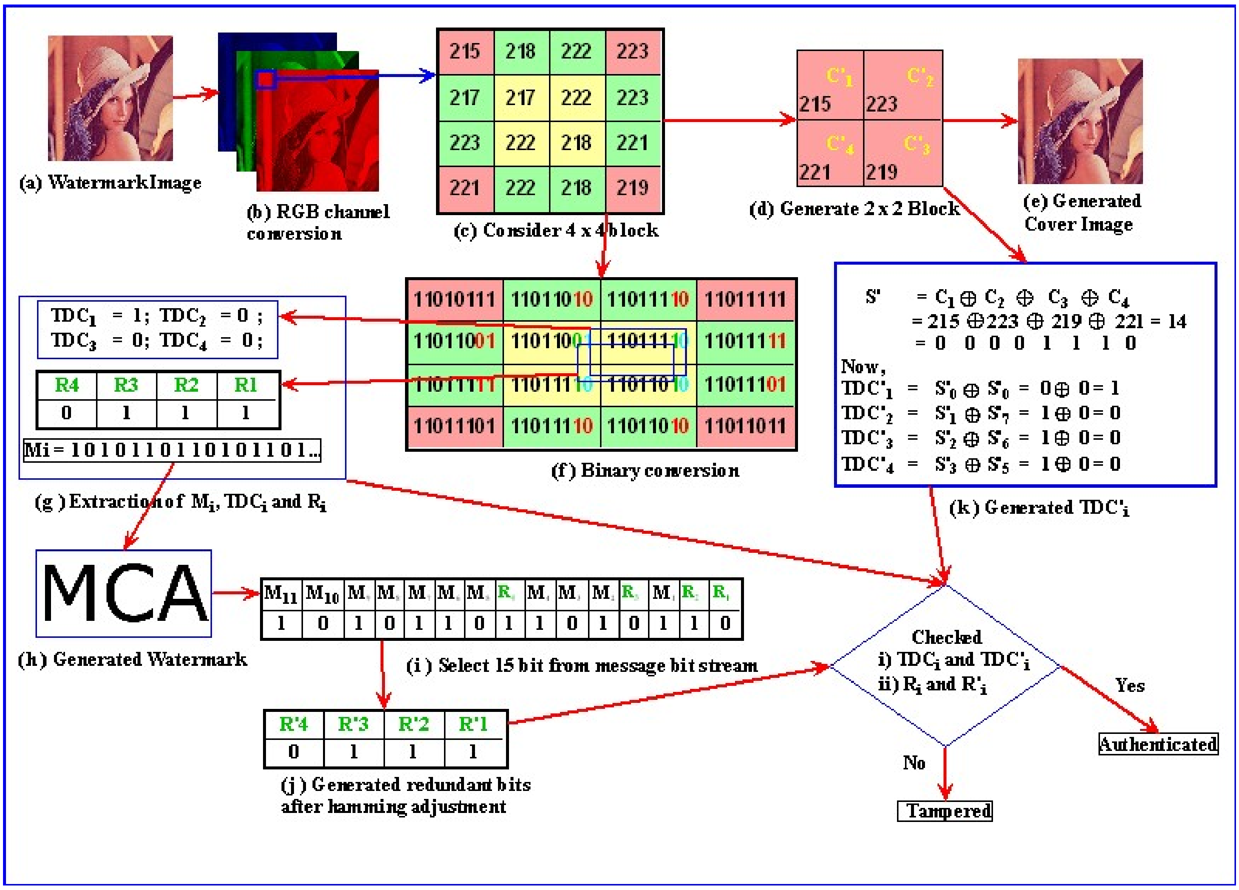


Figure 4.14: Numerical illustration of watermark embedding phase in RWS-LBP-HC

are appended to WB' string to retrieve 16-bit watermark bits. Now, $(15, 11)$ HC is applied to create redundant bits (R'_i) from WB'. Then, R' is generated by appending the redundant bits R'_i. After that, 1 bit from each LSB and LSB-1 position of the middle pixels (X'_i for $i = 1$ to 4) of ($RI'_{4\times4}$) are extracted and are appended to a string TDC_i and R_i respectively for $i = 1$ to 4. The above process has been repeated for all (4×4) image blocks of ICI'_R and watermark bits string (WBS) are produced by appending it in WB' after extraction. The whole process is repeated for the remaining blocks including ICI'_G and ICI'_B color blocks. The tamper detection and authentication process have been tested by comparing TDC_i with TDC'_i and (R_i with R'_i). If they are equal then no tamper occurred and the cover image is authenticated. Then W is generated from the WBS and the cover image is generated by excluding the of interpolated pixels of IWI. The details watermark extraction algorithm is elaborated in Algorithm 4.4 and a numerical illustration of RWS-LBP-HC is depicted in Fig. 4.14.

Algorithm 4.4: RWS-LBP-HC: Watermark Extraction Algorithm

```
Input  : Watermarked Images (IWI)
Output: Cover Image (CI) and watermark (W)
Step 1: Watermarked Image (IWI) are divided into R, G and B color ingredients as $(ICI^{'}_{R})$, $(ICI^{'}_{G})$ and $(ICI^{'}_{B})$.
Step 2: A $(4 \times 4)$ pixel block from $(ICI^{'}_{R})$ are considered.
Step 3: Color blocks are converted into $(CI^{'}_{R})$ by reverse interpolation technique.
Step 4: Corner pixels of the selected block as $C'_1, C'_2, C'_3\ and\ C'_4$ are taken.
Step 5: Bits of $C'_1, C'_2, C'_3\ and\ C'_4$ are XORed and are stored to form a 8 bit string S'.
Step 6: Two specific bits of S' are XORed and these are appended to TDC'.
        $TDC'_1 = S'_0 \oplus S'_7; TDC'_2 = S'_1 \oplus S'_6; TDC'_3 = S'_2 \oplus S_5; TDC'_4 = S'_3 \oplus S'_4;$
Step 7: 2-LSB of the border pixels $(P^{'}_{i})$ of $(RI'_{4\times4})$ are obtained and are appended to WB' string to retrieve 16 bits watermark data.
Step 8: One bit from the LSB and LSB-1 position of the middle pixels $(X^{'}_{i})$ of $(ICR'_{4\times4})$ are extracted and a string $TDC_i$ and $R_i$ respectively for i = 1 to 4 are appended
Step 9: Redundant bits $R'_1, R'_2, R'_3\ and\ R'_4$ are generated from WB' using Hamming code
        $R'_1$ = XOR of 1,3,5,7,9,11,13 and 15-th bits of WB'
        $R'_2$ = XOR of 2,3,6,7,10,11,14 and 15-th bits of WB'
        $R'_3$ = XOR of 4,5,6,12,13,14 and 15-th bits of WB'
        $R'_4$ = XOR of 8,9,10,11,12,13,14 and 15-th bits of WB'
Step 10: GHC is generate by appending the redundant bits $R'_1, R'_2, R'_3\ and\ R'_4$.
Step 11: Step-3 to Step-10 are repeated for all $(4 \times 4)$ image blocks of $ICI^{'}_{R}$.
Step 12: Watermark bits are constructed string (WBS) by appending bits extracted in $WB'$.
Step 13: $ICI^{'}_{R}$ is generated from the updated $RI_{2m\times2n}$.
Step 14: Step-3 to Step-13 are repeated for the image blocks of $ICI^{'}_{G}$ and $ICI^{'}_{B}$.
Step 15: $(TDC_i$ & $TDC'_i)$ and $(R_i$ & $R'_i)$ are compared for tamper detection and authentication.
Step 16: Watermark (W) from the WBS are generated.
```

4.2.4 Experimental Results and Comparison

A set of benchmark [89], [61], [90], [26] colour images of size (512×512) (shown in 2.3) are considered as CI to evaluate the efficiency of RWS-LBP-HC. Three different sizes of logo images have been considered as a watermark (W) as shown in Fig. 4.16 to measure the quality and corresponding capacity. Performance of RWS-LBP-HC is assessed to test its effectiveness. MSE [35], PSNR [35], SSIM [78] and Q-Index are computed using the equation (2.5), (2.6), (2.8) and (2.12) respectively to test the perceptible characteristics after embedding. Also NCC [94], BER [65], SD [35] and CC [35] are computed using the equation (2.11), (2.13), (2.9) and (2.10) respectively for tamper detection in a watermarked image. Performance of RWS-LBP-

HC is assessed on the basis of time complexity and it is compared with other existing schemes.

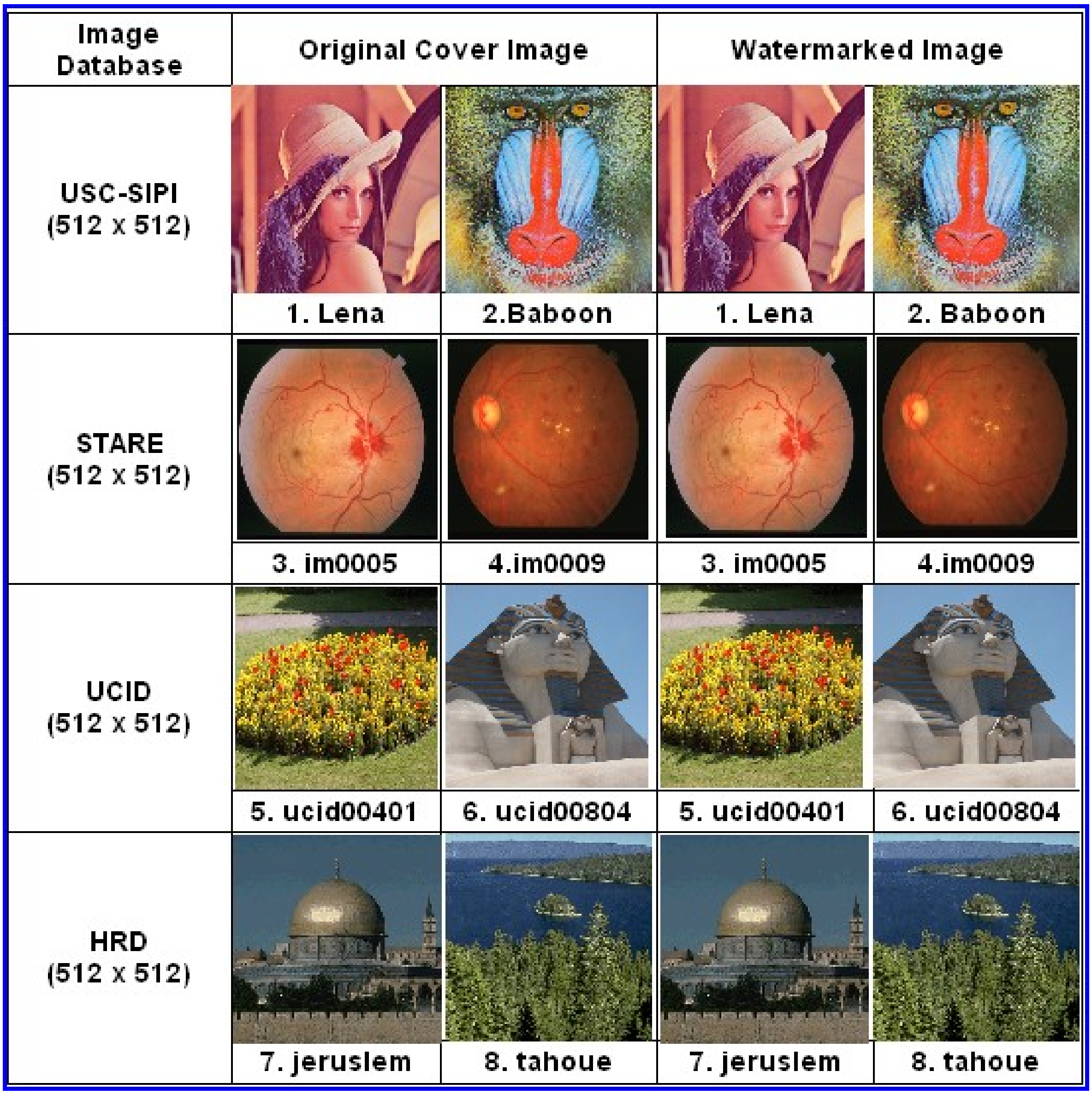


Figure 4.15: Pictorial results of output images in RWS-LBP-HC

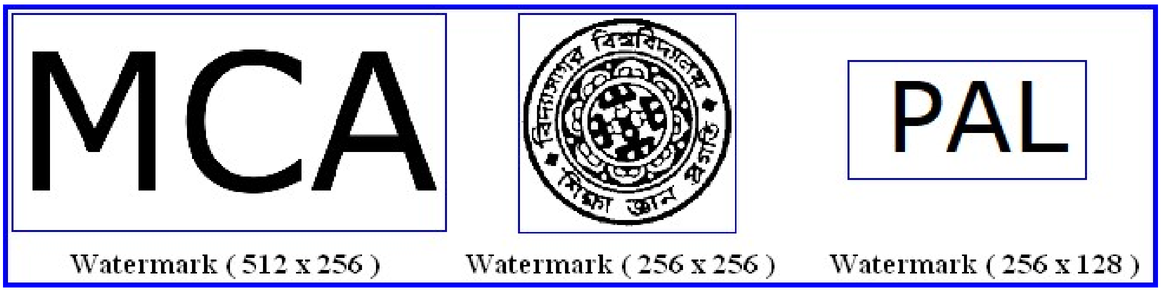


Figure 4.16: Watermark images (logo) with different size used in RWS-LBP-HC

4.2.4.1 Quality Measurement and Payload Analysis

The fundamental necessities of any watermarking scheme are robustness and imperceptibility. Usually, the quality of watermarked images are evaluated from their subjective and objective quality indices. The subjective characteristics of the watermarked images is evaluated in RWS-LBP-HC and it has been shown in Fig. 4.15. The evaluation results of RWS-LBP-HC with respect to PSNR, Capacity and Q-Index after embedding watermark of different bits are presented in Table 4.9. It is observed from Fig. 4.15 that no visual distortions are detected after embedding maximum payload of $3,145,728$ bits.

The RWS-LBP-HC is tested taking sample images from four different standard benchmark image databases and experimental outcomes are noted in Table 4.9. Table 4.9 illustrate that after embedding a highest amount of $3,145,728$ bits watermark, approximately 45 dB PSNR can be achieved on average taking the aforesaid datasets. Q-Index values are also close to unity which establishes the acceptability of proposed scheme.

Table 4.9: Capacity, PSNR, Q-Index, and Payload results are presented for standard benchmark images in RWS-LBP-HC

Database	Image	Capacity (bits)	PSNR (dB)	Q-Index	Payload (bpp)
USC-SIPI [90]	Lena	786,432	51.2392	0.99999	0.75
		1,572,864	48.2345	0.99973	1.50
		3,145,728	45.1615	0.99945	3.00
UCID [61]	Jeruslem	786,432	50.3642	0.99996	0.75
		1,572,864	47.4567	0.99967	1.50
		3,145,728	44.6374	0.99932	3.00
STARE [89]	Im0001	786,432	50.9861	0.99997	0.75
		1,572,864	48.1534	0.99988	1.50
		3,145,728	45.1381	0.99969	3.00
HDR [26]	Medical1	786,432	51.1892	0.99998	0.75
		1,572,864	48.3159	0.99991	1.50
		3,145,728	45.2051	0.99939	3.00

Additionally, the results of objective analysis have been depicted in Table 4.10 for color images (without any invasion). Table 4.10 presents the PSNR results in terms of number of Images

Table 4.10: Average PSNR of various yardstick image datasets considering 25 to 100 images in RWS-LBP-HC

Datasets	Image Size	Total Image	Average PSNR
STARE [89]	512 × 512	25	44.3598
		50	44.3206
		100	44.2424
USC-SIPI [90]	512 × 512	25	45.3101
		50	45.2602
		100	45.1615
UCID [61]	512 × 512	25	43.9717
		50	43.9378
		100	43.9175
HDR [26]	512 × 512	25	45.1439
		50	45.1138
		100	45.0605

used from the different image database. From the experimental results it is clear that Q-value, SSIM and NCC results outperform than the other schemes.

The resemblance with respect to PSNR(dB), Capacity (bpp) and Q-Index with existing techniques based on interpolation are presented in Table 4.11. From experimental results, it is noticed that RWS-LBP-HC attains a maximum embedding capacity $(3,145,728$ bits) with good visual quality (48.66 dB PSNR) which is very important for medical, e-governance and military applications.

Table 4.11: Comparison of different RWT in terms of PSNR, embedding capacity and Q-Index in RWS-LBP-HC

Images	Jung and Yoo [42]			Lee & Huang's [46]			Hu and Li's [30]			Jana et al. [35]		RWS-LBP-HC		
	Capacity (Bits)	PSNR (dB)	Q-Index	Capacity (Bits)	PSNR (dB)	Q-Index	Capacity (Bits)	PSNR (dB)	Q-Index	Capacity (Bits)	PSNR (dB)	Capacity (Bits)	PSNR (dB)	Q-Index
Lena	1,34,850	32.15	0.7601	2,15,506	30.61	0.7463	1,17,190	32.13	0.8495	7,76,224	35.80	3,145,728	46.58	0.9995
Airplane	1,32,339	30.38	0.7945	1,89,611	28.78	0.7735	1,02,615	30.80	0.8585	7,76,224	35.81	3,145,728	49.36	0.9995
Baboon	3,02,991	22.64	0.6875	3,82,265	22.17	0.6681	2,72,513	24.04	0.8087	7,76,224	35.81	3,145,728	49.35	0.9996
Boat	1,90,569	27.91	0.6881	2,78,005	26.75	0.6955	1,73,462	28.39	0.8136	7,76,224	35.80	3,145,728	49.34	0.9996
Peppers	1,39,772	30.47	0.7051	2,30,340	29.11	0.7094	1,30,494	30.91	0.8291	7,76,224	35.82	3,145,728	48.65	0.9994
AVG	1,80,104	28.71	0.72706	2,59,145	27.48	0.7186	1,59,254	29.25	0.8319	7,76,224	35.81	3,145,728	48.66	0.9995

The comparison graph with respect to capacity (in bits) and PSNR (dB) for Lena, Airplane,

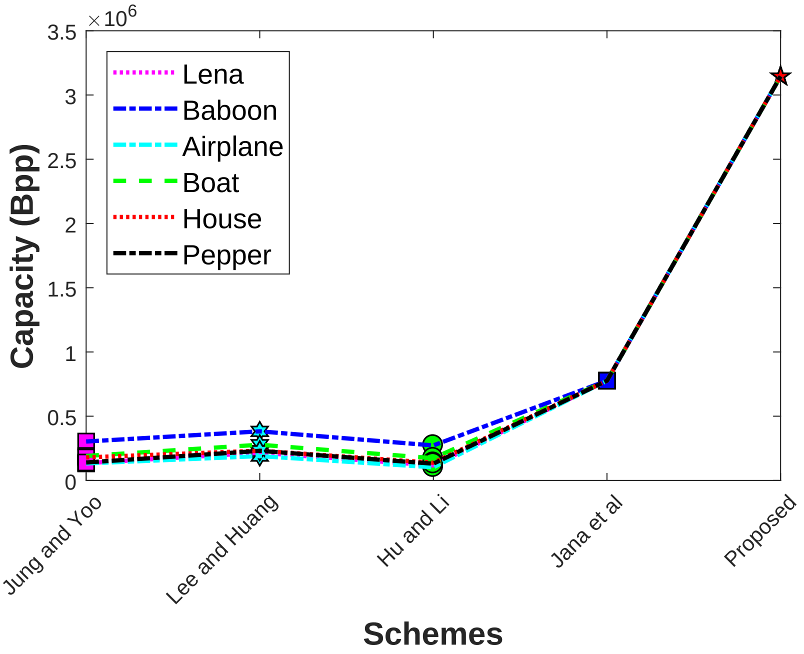


Figure 4.17: Comparison graph in terms of capacity with existing schemes in RWS-LBP-HC

Baboon, Tiffany, Boat and House images are presented in Fig. 4.17 and Fig. 4.18 respectively. From graphical representation, it is found that RWS-LBP-HC provides a higher capacity compared to other existing schemes.

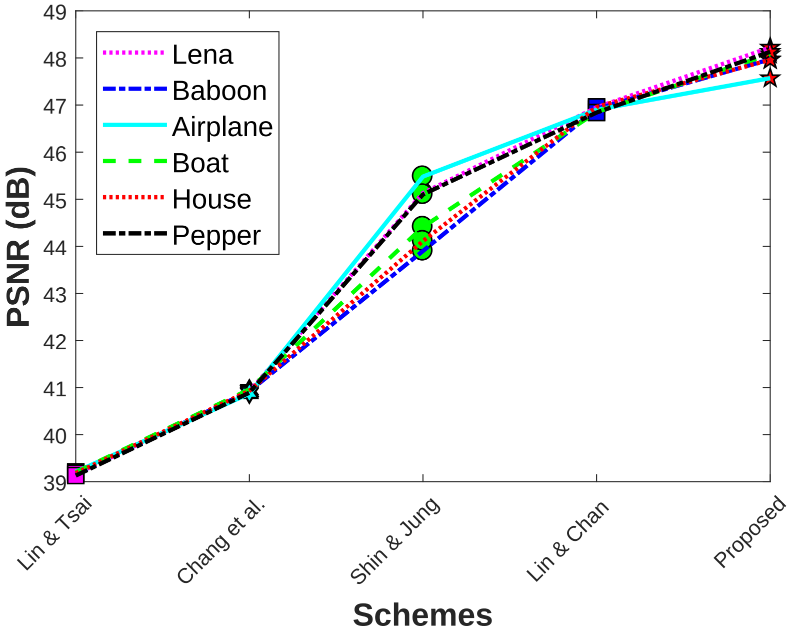


Figure 4.18: Comparison graph in terms of PSNR (dB) with existing schemes in RWS-LBP-HC

Table 4.12: Comparison with existing LBP based scheme in terms of PSNR in RWS-LBP-HC

Schemes	Parah et al. [65]	Wenyin et al. [94]	Pinjari et al. [68]	Zhang et al. [107]	RWS-LBP-HC	Improvement % w.r.t [107]
Lena	40.53	42.64	43.54	44.02	45.16	2.52
Airplane	40.99	41.37	43.59	44.32	45.16	1.86
Baboon	40.08	42.37	43.55	44.46	45.31	1.87
Boat	41.47	41.28	43.64	43.88	45.26	3.04
Pepper	40.71	42.52	43.53	44.61	45.17	1.23

Table 4.12 represents comparison results of RWS-LBP-HC with respect to the other existing LBP schemes. Also, the comparison graph concerning PSNR (dB) for Lena, Airplane, Baboon, Boat and Pepper images are shown in Fig. 4.19. From the graphical representation it is seen that RWS-LBP-HC provides a better PSNR results compared to other existing LBP based schemes [68, 94, 107].

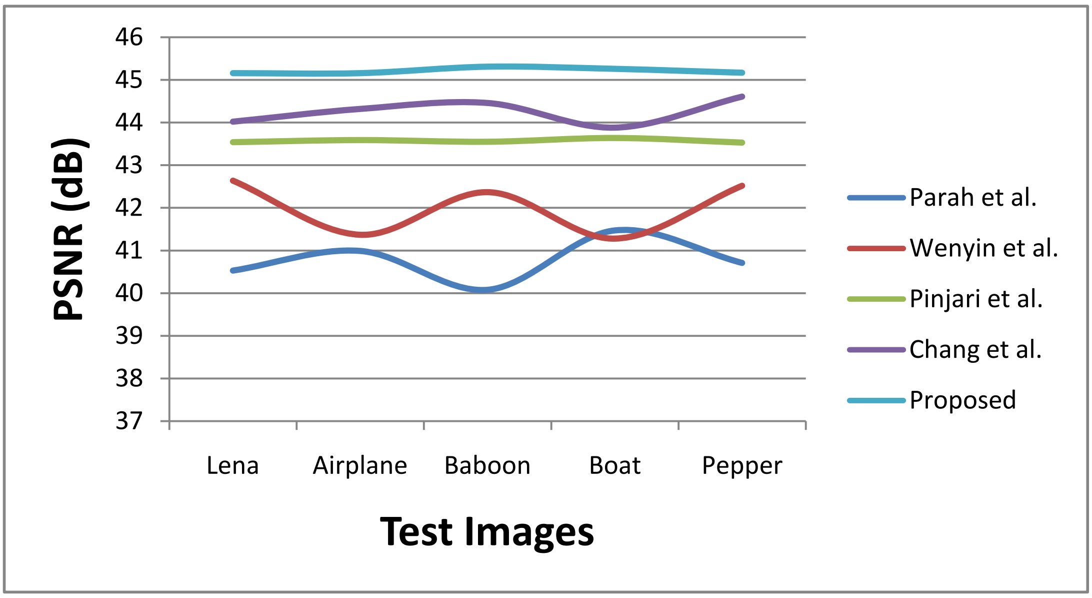


Figure 4.19: Comparison result with LBP based existing schemes in terms of PSNR (dB) in RWS-LBP-HC

4.2.4.2 Robustness Analysis

Table 4.13 shows the evaluation results in terms of various evaluation scheme like MSE, PSNR, NCC, SSIM, Q-Index and BER with color cover images of four different benchmark datasets. From Table 4.13, it is found that the average PSNR for the aforesaid image databases is greater

Table 4.13: Results of MSE, PSNR, NCC, SSIM, Q-Index and BER for different image of four different benchmark datasets in RWS-LBP-HC

Image Dataset	Images	MSE	PSNR (dB)	NCC	SSIM	Q-Index	BER
SIPI	Lenna	2.4942	45.1615	0.999936	0.98039	0.999452514	0.031952
	Baboon	2.4103	45.3101	0.999939	0.991713	0.999477653	0.031233
	Tiffany	2.5493	45.0666	0.999972	0.977551	0.998572325	0.03088
	Average	2.4846	45.1794	0.999949	0.983218	0.999167497	0.031355
UCID	anhinga	3.2421	44.0226	0.999880	0.980390	0.999424632	0.034966
	bardowl	3.3589	43.8689	0.999846	0.977551	0.999491527	0.037210
	jeruslem	2.8141	44.6374	0.999844	0.983030	0.999009357	0.033161
	Average	3.3112	43.9569	0.999865	0.983171	0.99926072	0.036082
STARE	im0001	2.5077	45.1381	0.999938	0.967999	0.999753284	0.031354
	im348	2.5217	45.1138	0.999913	0.967692	0.999593411	0.031304
	im548	2.5043	45.1439	0.999939	0.967411	0.999693760	0.031319
	Average	2.5112	45.1319	0.999930	0.967700	0.999680152	0.031325
HDR	Medical1	2.4693	45.2051	0.999943	0.979987	0.999859235	0.031287
	Medical2	2.4618	45.2183	0.999906	0.977011	0.999478028	0.031248
	Medical3	2.4693	45.2051	0.999943	0.979987	0.999859235	0.031280
	Average	2.4668	45.2095	0.999930	0.978995	0.999732166	0.031271

than 45 dB. Also the NCC, SSIM and Q-Index values of RWS-LBP-HC are nearer to one, which proves the effectiveness of the designed algorithm. The BER results prove that RWS-LBP-HC is robust.

4.2.4.3 Tamper Detection and Recovery

Robustness of RWS-LBP-HC is analyzed by evaluating the quality metrics such as NCC [94], BER [65], SD and CC [35]. The RWS-LBP-HC has been assessed against salt and pepper noise, cropping and copy-move forgery attacks. The experimental outcomes after applying salt and pepper noise, cropping and copy-move forgery attack with different noise density level are depicted in Fig. 4.20, Fig. 4.21 and Fig. 4.22 respectively. It is clear that after extraction, the objective quality of the extracted watermark is slightly changed where as the tampered location of the recovered cover image has been identified successfully. The statistical analysis (SD and CC) shows the robustness of RWS-LBP-HC. The different objective metrics are presented in Table 4.14 when extraction performed from tampered image. From the Table 4.14 it is clear that the less BER values and near unity Q-Index and NCC indicate robustness of RWS-LBP-HC against salt and pepper noise, cropping and copy move forgery attack. Again it is clear from the Table 4.14 that the robustness of our approach varies inversely with the noise density.

The algorithmic complexity and execution time of any watermarking scheme is a significant parameter in current research scenario. The execution time of RWS-LBP-HC has been compared with some recent works [65, 78, 91] and the comparative outcomes have been noted in Table 4.15. It is observed that RWS-LBP-HC requires 0.57 seconds for total execution which is 0.32 seconds, 0.55 seconds, and 0.08 seconds faster than Su et al. [78], Verma et al. [91] and Parah et al. [65] schemes respectively. The lesser complexity in RWS-LBP-HC is achieved due to simple algebraic manipulations and the threading concept of Java.

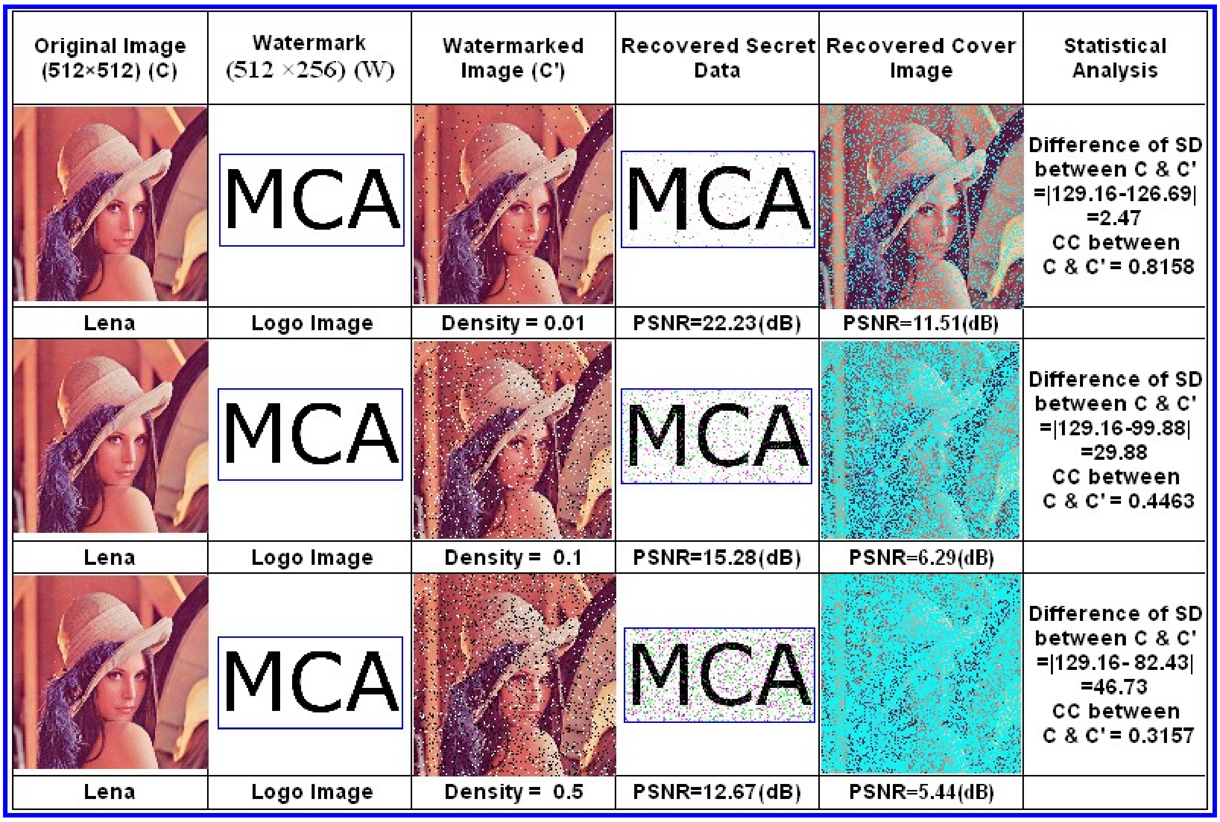


Figure 4.20: Effect of salt pepper noise on Lena image in RWS-LBP-HC

To determine the algorithmic complexity, a cover image of size $(M \times N)$ is considered and 16-bit watermark is embedded within a (4×4) pixel block. Therefore from the $Step-12$ and $Step-14$ of Algorithm 4.3, it has been easily calculated that the time complexity is $\mathcal{O}(MN)$. Moreover, at the time of extraction, the complexity for embedding is $\mathcal{O}(MN)$, considering $Step-12$ and $Step-14$ of Algorithm 4.4. The execution time is reduced by employing thread-

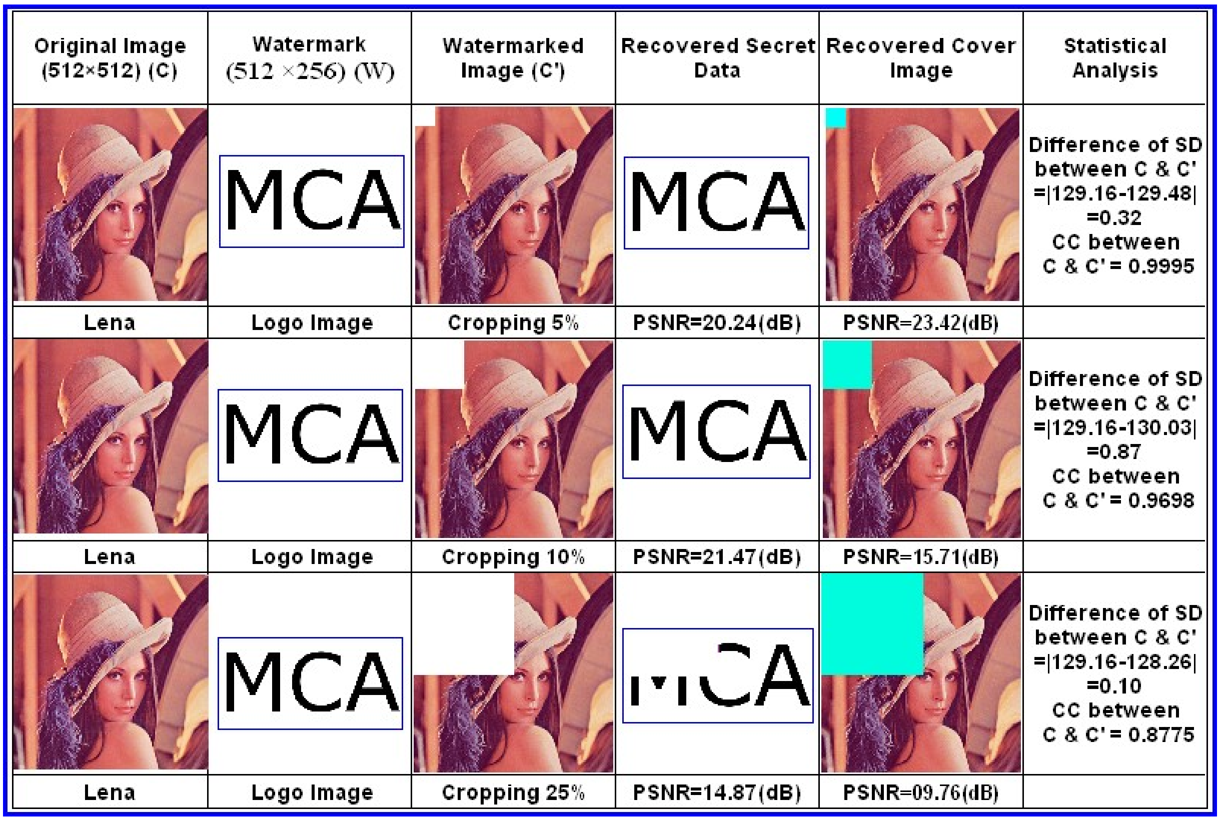


Figure 4.21: Effect of cropping attacks on Lena image in RWS-LBP-HC

Table 4.14: PSNR, SSIM, Q-Index, NCC and BER of distorted watermark images due to salt pepper noise, cropping and copy-move forgery attacks in RWS-LBP-HC

Noise	Perturbation	PSNR (dB)		SSIM		Q-index		NCC		BER	
		CI	WI	CI	WI	CI	WI	CI	WI	CI	WI
Salt and Pepper	0.01	11.51	22.23	0.1549	0.5712	0.8147	0.9893	0.8943	0.9962	0.03298	0.00333
	0.1	06.29	15.28	0.0370	0.1784	0.4241	0.9445	0.7228	0.9812	0.10986	0.01605
	0.5	05.44	12.67	0.0399	0.1298	0.2787	0.9001	0.6822	0.9658	0.13366	0.02824
Cropping	10%	23.42	22.23	0.99159	0.5712	0.99949	0.9893	0.99227	0.9962	0.00177	0.00333
	25%	15.71	15.28	0.95133	0.1784	0.96918	0.9445	0.95598	0.9812	0.01074	0.01605
	50%	09.76	12.67	0.79647	0.1298	0.87436	0.9001	0.84509	0.9658	0.13366	0.04352
Copy move Forgery	5 %	26.41	29.44	0.9932	0.9921	0.9976	0.9970	0.9961	0.9992	0.0010	0.0015
	10%	24.99	23.60	0.9827	0.9871	0.9938	0.9946	0.9980	0.9925	0.0013	0.0019
	20%	18.70	18.74	0.9456	0.9475	0.9649	0.9889	0.9915	0.9775	0.0058	0.0075

ing conception. During embedding only 0.52 seconds time is acquired to insert a (512×256) i.e., $3,145,728$ bits watermark within a (512×512) cover image and 0.0537 seconds is acquired during extraction.

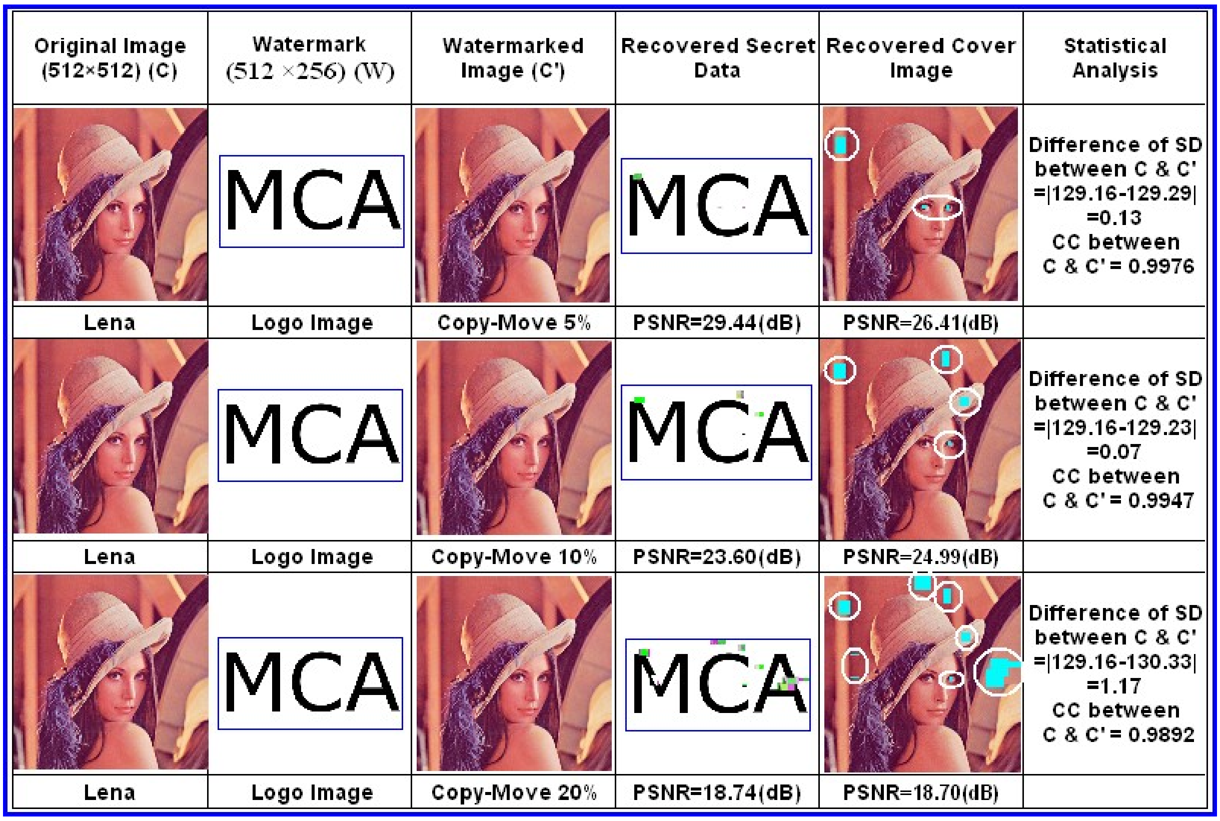


Figure 4.22: Effect of copy move forgery attacks on Lena image in RWS-LBP-HC

Table 4.15: Comparison table in terms of execution time in RWS-LBP-HC

Schemes	Size of Image	Embedding time (sec)	Extraction time (sec)	Total time (sec)
Su et al. [78]	512 × 512	0.5244	0.3701	0.8945
Verma et al. [91]	512 × 512	0.5173	0.5989	1.1162
Parah et al. [65]	512 × 512	0.59	0.0624	0.6524
RWS-LBP-HC	512 × 512	0.52	0.0537	0.5737

4.3 Discussion

In this chapter, the LBP operator has been introduced to improve our work concerning robustness. In DRWS-LBP, LBP operator is employed in the dual image for tamper detection. After analyzing RWS-LBP-HC in terms of robustness, it has been seen that the scheme can resist seven types of attacks and cover image can be recover successfully from nine kinds of attacks after performing ten suggested attacks shown in Fig. 4.23.

So there is a chance to increase robustness and capacity. A new watermarking scheme has been developed in the interpolated image with the help of (15, 11) HC shown in RWS-LBP-HC. Here the Hamming code is used to detect tamper, correct tamper and use of LBP can able to locate

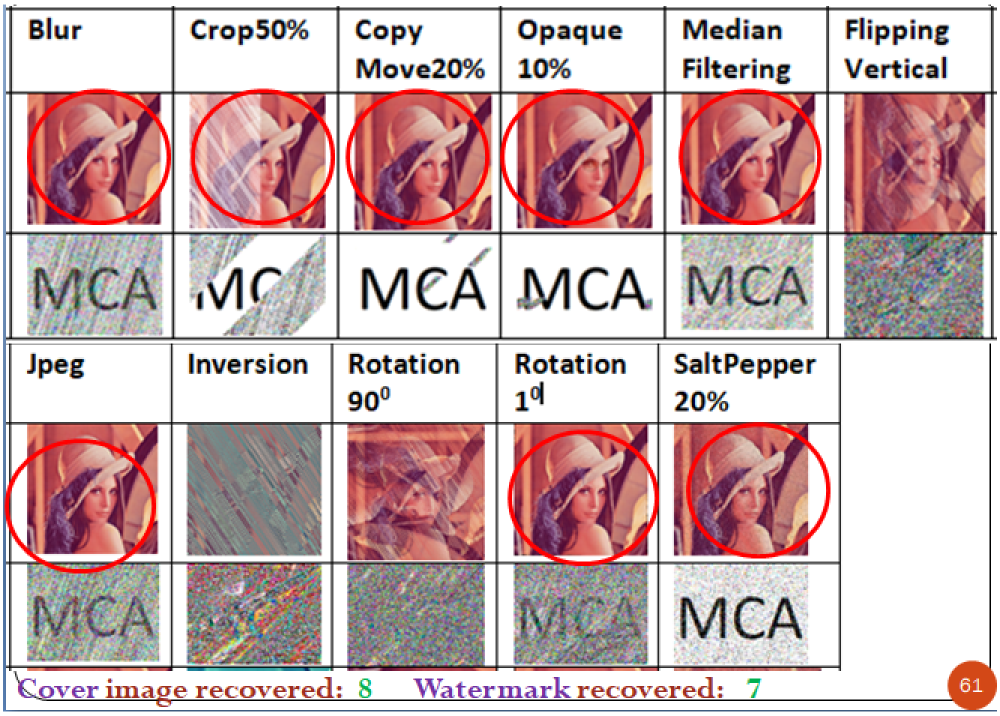


Figure 4.23: Effects of different types of attacks on Lena image for DRWS-LBP.

tampered region. Moreover, in the sense of robustness, it has been seen that DRWS-LBP can resist nine attacks and cover image can be recovered successfully from nine types of attacks after performing ten special types of attacks. But after localization, this scheme can recover cover image only from three cases shown in Fig. 4.24. The overall results are shown in Table

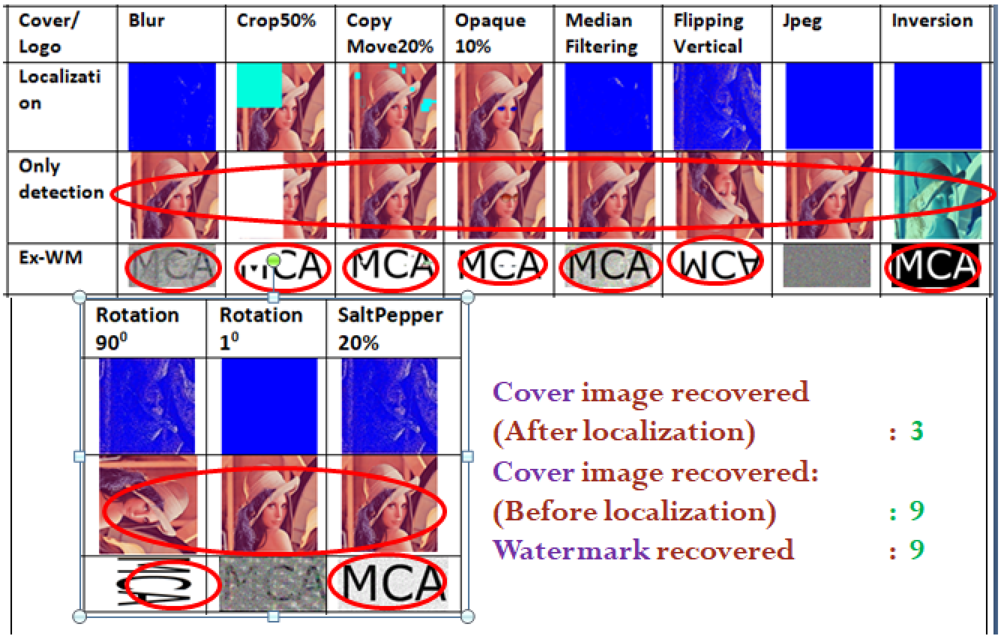


Figure 4.24: Effects of different types of attacks on Lena image for RWS-LBP-HC.

4.16.

Table 4.16: Effects of 10 different types of attacks

Schemes	Image Recovered	Salt Pepper	Cropping	Copy move	Opaque	Median Filtering	Flipping (Vertical)	JPEG Compression	Blurring	Rotation	Invertion
RWS-WM	CI	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗
	W	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
RWS-CA	CI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
	W	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓
DRWS-LBP	CI	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗
	W	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗
RWS-LBP-HC	CI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
	W	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓

4.3.1 Salient Feature of this Chapter

- In this chapter, authentication can be achieved by using Local Binary Pattern with dual image. An authentication code is generated with the help of the LBP operator in both the schemes.

- Tampered location can be identified with the help of LBP operator in DRWS-LBP. Hamming code has been used to detect and correct tampered location in RWS-LBP-HC.

- Shared secret key has been utilized to distribute watermark pixel pairs among dual images to enhance security. At the time of sharing, LFSR gives the randomness of the key.

- Digital watermarking based on LBP method was not reversible. Reversibility has been achieved in proposed watermarking schemes using LBP and Hamming codes.

- Moreover, dual image and image interpolation have been used to increase embedding capacity and to maintain imperceptibility of LBP based watermarking schemes.

So to improve the reversibility after tampering, an effort has been made to develop our scheme and hybridize LBP operator with CA in the watermarking scheme. Moreover, to perform authentication, tamper detection and tampered region localization successfully, we have designed two more schemes RWS-LBP-CA and RWS-LBP-WM-LIP presented in Chapter 5.