Chapter 6

An efficient watermarking scheme for image authentication and localization with two chances for restoration capability

6.1 Introduction

The huge growth in multimedia technologies is actively increasing the production, transmission and publishing facilities of digital multimedia data. The availability of a wide range of processing software has increased the rate of illegal operations such as duplication, modification and forgery without any trace of manipulations. Therefore, verification of the integrity and authentication of digital images have become big issues[9, 111]. Digital signatures and Digital watermarking technique have emerged as solutions to tackle these issues. The digital signature is the most utilized technique in protecting digital content. The digital signature can be either an encrypted or a signed hash value of a set of extracting features from the original image. For the authentication scheme, this signature is attached to the image and the received image verifies by comparing its signature to the attached signature[134, 147, 148, 149]. However, this traditional technique has various limitations [150]: 1) It encodes the signature in a file separate from the original image, thus require extra

bandwidth to transmit it; 2) it can only confirm whether the target image has been tampered but can not indicate the exact location of tampered regions; 3) it can not be used for tampered image restoration. The restoration issue of tampered regions has received much attention from the image and signal processing researchers [3]. On the other hand, digital watermarking is a hiding technique that embeds invisible information into a cover image without affecting the size of image. Therefore, to resolve the aforementioned limitations watermarking based authentication scheme is being encouraged to enhance the efficiency of restoration quality with localization capability[10, 126, 77].

Fragile and semi-fragile watermarking schemes are used for image authentication purpose. Fragile watermarking scheme can be destroyed by any minor unintentional or intentional manipulation [3, 9, 10, 64, 77, 126, 151]. Thus, it can detect any modification (i.e. even a single bit) in the watermarked image. Such type of authentication is called as exact authentication [3]. On the other hand, semi-fragile watermarking scheme differentiates between the content-preserving (non-malicious) processes, including compression, noise, filtering, smoothing etc, and malicious manipulations, including distortion, cropping, inserting, replacing, etc [15, 132, 152, 153, 154]. Such type of authentication is called as selective authentication [3]. In this chapter, we will restrict our attention towards the fragile watermarking scheme for exact tampered localization and manipulation detection with effective restoration capability. In general, fragile watermarking schemes can be classified into two main categories: pixel-based schemes [153] and block-based schemes [78, 126, 151]. In pixel-wise fragile watermarking schemes, the watermark is generated from the gray values of the cover pixels and is embedded into the cover pixels themselves. In block-wise fragile watermarking schemes, the cover image is divided into sub-blocks. The watermark is generated for each block and embedded into corresponding mapped block. If the watermarked image is modified (intentionally or unintentionally), the watermark of a particular block is not retrieved successfully, then that block is identified as tampered block.

The remaining part of this chapter is organized as follows. Section 6.2 gives the related research work with quality-related trade-offs. The methodology of proposed watermark embedding procedure and extraction procedure is discussed in Section

6.3. The experimental evaluation scenario and details of comparison with the existing approach are described in Section 6.4. Finally, the conclusion of this chapter is drawn in Section 6.5.

6.2 Related Work

Although various self-embedding watermarking schemes exist, there is no general model of the tampered restoration of the image because of the trade-off between the image quality and the restoration conditions. The image quality is decided by means of objectively measured distortion of the original image, e.g., using the peak signal to noise ratio (PSNR). The tampered restoration conditions are usually the maximum tampering rate, i.e., maximum tampered regions for which the restoration is still possible[63, 126].

In [65], the authors proposed DCT based content reconstruction flexible watermarking scheme using compressive sensing and composite reconstruction techniques. In this scheme, the embedded watermark bits for content recovery are computed from the DCT coefficients of five MSBs layers of the host image. In this scheme, the restored image quality depends on the tempered area. If smaller the tampered area, the more the amount of available watermark data will be, leading to a better quality of recovered content. This scheme is capable to restore up to tampering rate of 60 %.

Two self-embedding watermarking techniques were proposed by Zhang et al. [66], called as a reference sharing mechanism. In the first technique, the watermark was derived from 5 MSBs of the original cover image. In the second technique, the cover image was decomposed into three levels based on the hierarchical self-embedding scheme. In the first scheme, the original data in five layers of watermarked image can be recovered when tampering rate is no more than 24 %. In the second scheme, the reference sharing methods with different restoration capabilities are employed to protect the image at different levels. So that a better restored image can be obtained when a image tampered with less fake content and this is capable to restore up to 66% tampering rate. Another self-embedding scheme was proposed by Zhang et al. [67] in which, authors first permuted the image pixels based on a secret

key and then divided them into a series of pixel-pairs. The restoration data was generated by exclusiveOR operation within the original MSBs of pixel pairs and authentication data derived from MSBs and restoration data. Finally, watermark data was embedded into the three LSBs planes. In this scheme, the five MSBs of a pixel pair can be either fully or partially recovered by using reference data. The remaining uncertainty is resolved by manipulating local pixel correlations.This scheme is capable to restore up to tampering rate of 54 %.

In [64], Qian et al. presented a scheme based on Discrete Cosine Transform (DCT) to reduce the embedding data for self-restoration. The DCT coefficients of 8×8 blocks were encoded into different numbers of bits and the authentication-bits and restoration-bits were embedded into the three LSBs planes of the cover image. However, this scheme is capable to avoid tampering coincidence problem but the accuracy of tamper localization decreases because of large block size of 8×8.

A DCT and fractal compression coding based self-embedding fragile watermarking scheme was proposed by Zhang et al.[71]. In this scheme, three kinds of watermarks were generated for image authentication and restoration, which was in turn based on an interleaved and overlapped 8×8 image block. Three versions of restoration watermarks for each block were embedded into different quadrants, which provide three chances for block restoration in case of any image modification. However in this scheme the accuracy of tamper localization also decrease because of large block size of 8×8.

A representative of the watermarking technique for tamper detection and recovery is the hierarchical digital watermarking scheme proposed by Lin et al. [134]. In Lin's method, the detection of tampered regions is based on a 3-level hierarchical structure. If a tampered block is not detected in level-1 inspection, it will be detected in level-2 or level-3 inspection with a probability of nearly 1. Lin's method is also storage effective, as it only requires a secret key and a public chaotic mixing algorithm to recover a tampered image.

Lee et al. [111] proposed a dual watermarking scheme for image authentication and restoration. Two copies of watermark of the whole image are embedded in this scheme, so that it can provide second chance for block restoration in case one copy is tampered. Dual watermark can improve the quality of recovered image. However,

there are two main drawbacks in this scheme. Firstly, the whole watermark (i.e authentication as well as recovery data) embedded into mapped blocks. If block is not tampered and corresponding mapped blocks are tampered then we wrongly consider the block is tampered. Secondly, in this scheme recovery data is just average value of the block. So, its restoration quality is not good enough.

In this chapter, we present an effective self -embedding watermarking scheme based on DCT encoding technique. Here, a block-wise mechanism is used for tampered region detection, localization and restoration. The block size in this scheme is 2×2. There are some strong points in the scheme: (1) very less sensitive to error pixels; (2) negligible blocking artifacts because of using small size of blocks; (3) high quality restoration based on DCT coefficients and second chance of restoration (4) the tampered detection and localization accuracy is high because of using three-level hierarchical tampered detection mechanisms. In this scheme to avoid tampering coincidence problem i.e. both copy of restoration bits are not able to extract from the mapped blocks, then restoration in this scheme is based on its 3×3 neighborhood blocks. The proposed scheme is also secured using some predefined keys. Experimental results demonstrate that the proposed scheme is capable to restoration up to 50 % tampering rate with high PSNR and NCC values.

6.3 Proposed Approach

In this proposed scheme, five most significant bits (MSBs) planes of the host image are kept unchanged, while three least significant bits (LSBs) planes of the host image is modified by the watermark. The watermark is generated by five MSBs planes of the host image itself. For each block, twelve bits watermark are generated in which two bits are called authenticated bits and remaining 10 bits are called restoration bits. Authentication bits are used for detection and localization of the tampered host image, whereas the restoration bits are used to restore the extensive content of the corrupted areas of the host or cover image. For color images, all the color channels (i.e. R, G and B) are watermarked similar to gray image by the proposed algorithm. Let the original image I contain N_1 and N_2 numbers of rows and columns respectively, and let N represent the total number of pixels ($N = N_1 \times N_2$). The

intensity range of n^{th} pixel of the host image is denoted by $P^n \in [0, 255]$, where n=1, 2, 3, $\cdots$, N. Any pixel P^n can be represented by eight bits denoted as $P^n(1)$, $P^n(2)$,$\cdots$, $P^n(8)$, where where $P^n(1)$ is the first least significant bit (LSB) and $P^n(8)$ is the most significant bit (MSB). It can be represented in the binary form as follows:

$$P^n(i) = \lfloor \frac{P^n}{2^{i-1}} \rfloor \bmod 2, i = 1, 2, ..., 8 \tag{6.1}$$

and the decimal equivalent can be represented as:

$$P^n = \sum_{i=1}^{8} \left(P^n(i) \times 2^{i-1} \right) \tag{6.2}$$

6.3.1 Watermark Embedding Procedure

Watermark embedding procedure can be divided into three phases. The first one is the restoration bits generation, second one is the authentication bits generation and last one is block mapping. The block diagram of embedding procedure of the proposed scheme has been depicted in Fig. 6.1. For each block of size 2×2 of the cover image, two authentication bits and ten restoration bits are generated.

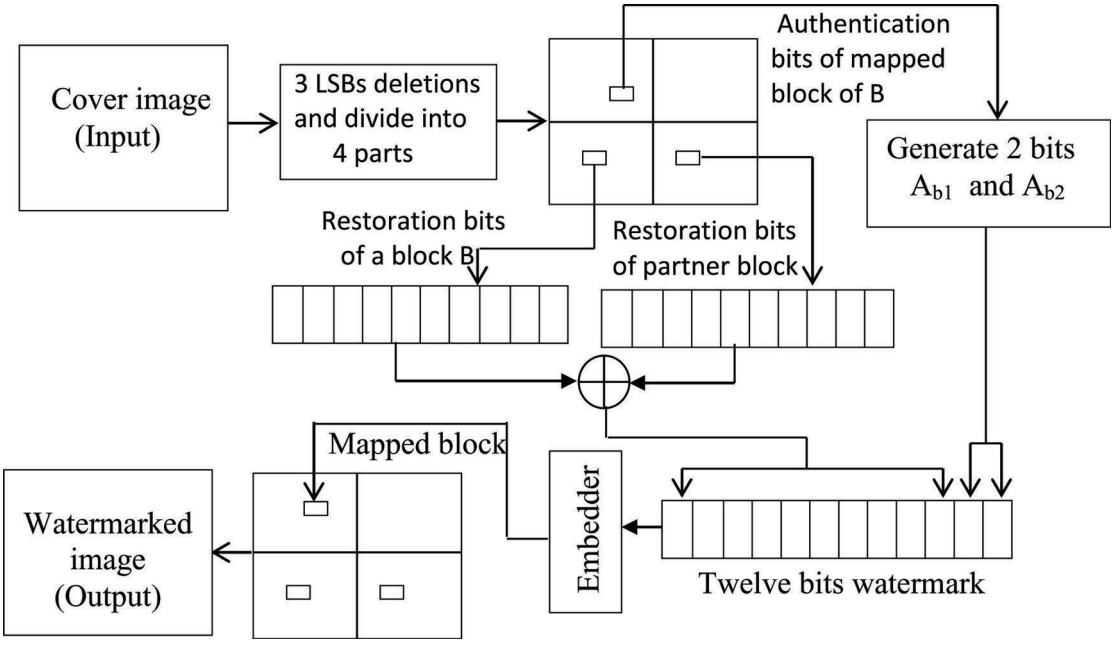


FIGURE 6.1: Block diagram of watermark embedding procedure.

6.3.1.1 Restoration bits Generation Procedure

Algorithm 10 Block Restoration bits Generation

INPUT: Input Block (B_n^s) Pixels: P_n^1, P_n^2, P_n^3, P_n^4
OUTPUT : Vector V of size 1×10

1: **procedure** RESTORATION BITS GENERATION
2: $g_n^b = \lfloor \frac{P_n^b}{2} \rfloor, n = 1, 2, 3, ..., N/16; b = 1, 2, 3, 4$
3: $g_n^b = g_n^b - 8, n = 1, 2, 3, ..., N/16; b = 1, 2, 3, 4$
4: $G_n = DCT2\left(\begin{bmatrix} g_n^1 & g_n^2 \\ g_n^3 & g_n^4 \end{bmatrix}\right) = \begin{bmatrix} G_n^i & G_n^2 \\ G_n^3 & G_n^4 \end{bmatrix}, n = 1, 2, ..., N/16$ ▷ G_n is the DCT coefficients of n^{th} block.
5: $r_j = round(G_n^j), n = 1, 2, 3, ..., N/16; j = 1, 2$
6: $sign(r_j) = \begin{cases} 0 & \text{if } G_n^j \geq 0; \\ 1 & \text{if } G_n^j < 0. \end{cases} \quad j = 1, 2$
7: $V(1) = sign(r_1)$
8: $V(2:5) = dec2Bin(r_1, 4)$ ▷ Dec2Bin(), used to convert decimal value into binary
9: $V(6) = sign(r_2)$
10: $V(7:10) = dec2Bin(r_2, 4)$ ▷ Dec2Bin(), used to convert decimal value into binary
11: $V(1:10) = V(1:10) \oplus K(1:10)$
12: **end procedure**

The restoration bits generation procedure can be done by the following steps:

Step 1. Deletes the three LSBs from each pixel and divide the cover image I into four equal parts I_1, I_2, I_3 and I_4 such that,

$$\begin{aligned} I_1 \cup I_2 \cup I_3 \cup I_4 &= I \\ I_1 \cap I_2 \cap I_3 \cap I_4 &= \phi \end{aligned} \tag{6.3}$$

Now intensity level of each pixel of the cover image has reduced from $[0, 255]$ to $[0, 31]$.

Step 2. Each part of I is divided into non-overlapping blocks of size 2×2 pixels. So, in the each part of the cover image, total number of blocks are $\frac{N}{16}$. The n^{th} block pixels are denoted by P_n^m where n is from 1 to $\frac{N}{16}$, and $m \in \{1, 2, 3, 4\}$, denotes the parts of the host image in which this pixel belongs.

Step 3. Enter a secret non -zero integer Key_1 which is not divisible by 1024, and convert this Key_1 into ten bits binary vector K_1 as follows

$$\begin{aligned} Key_1 &= mod(Key_1, 1024) \\ K_1 &= dec2bin(Key_1, 10) \end{aligned} \tag{6.4}$$

where dec2bin(), is used to convert decimal value into binary.

Step 4. Now generate the 10 bits restoration bits for each block using the "Block Restoration bits Generation" algorithm. The pseudo code of this algorithm is shown in algorithm10.

Step 5. After generating the restoration bits of each block of all four parts I_1, I_2, I_3 and I_4 denoted as V_1, V_2, V_3 and V_4 embed with partner blocks into the corresponding two mapped blocks as follows:

$$\begin{aligned} V_1 \oplus V_2 &\longrightarrow mapped\_In\_Part(I_3) \\ V_2 \oplus V_3 &\longrightarrow mapped\_In\_Part(I_4) \\ V_3 \oplus V_4 &\longrightarrow mapped\_In\_Part(I_1) \\ V_4 \oplus V_1 &\longrightarrow mapped\_In\_Part(I_2) \end{aligned} \tag{6.5}$$

Hence this way, two copies of the restoration bits of each block embedded into the watermarked image.

6.3.1.2 Authentication bits Generation Procedure

Using five MSBs planes and spatial location of pixels, two authentication bits for each block can be generated in the following manner:

Authentication bit A_{b1} Generation

In order to generate first authentication bit A_{b1}, Hamming code technique is used (shown in algorithm 11). The block diagram of A_{b1} generation as shown in Fig. 6.2. Consider a pixel P_n^b is in n^{th} block (where b=1,2,3,4) and five MSBs of P_n^b are represented as $P_n^b(m)$ where $m \in \{4, 5, ..., 8\}$. The first authentication bit A_{b1} can be calculated in the following steps:

Step 1. Binary value of five MSBs of all four pixels of a block arranged in a 1-D array A.

Step 2. Permute array A and split this array into five groups of four bits each.

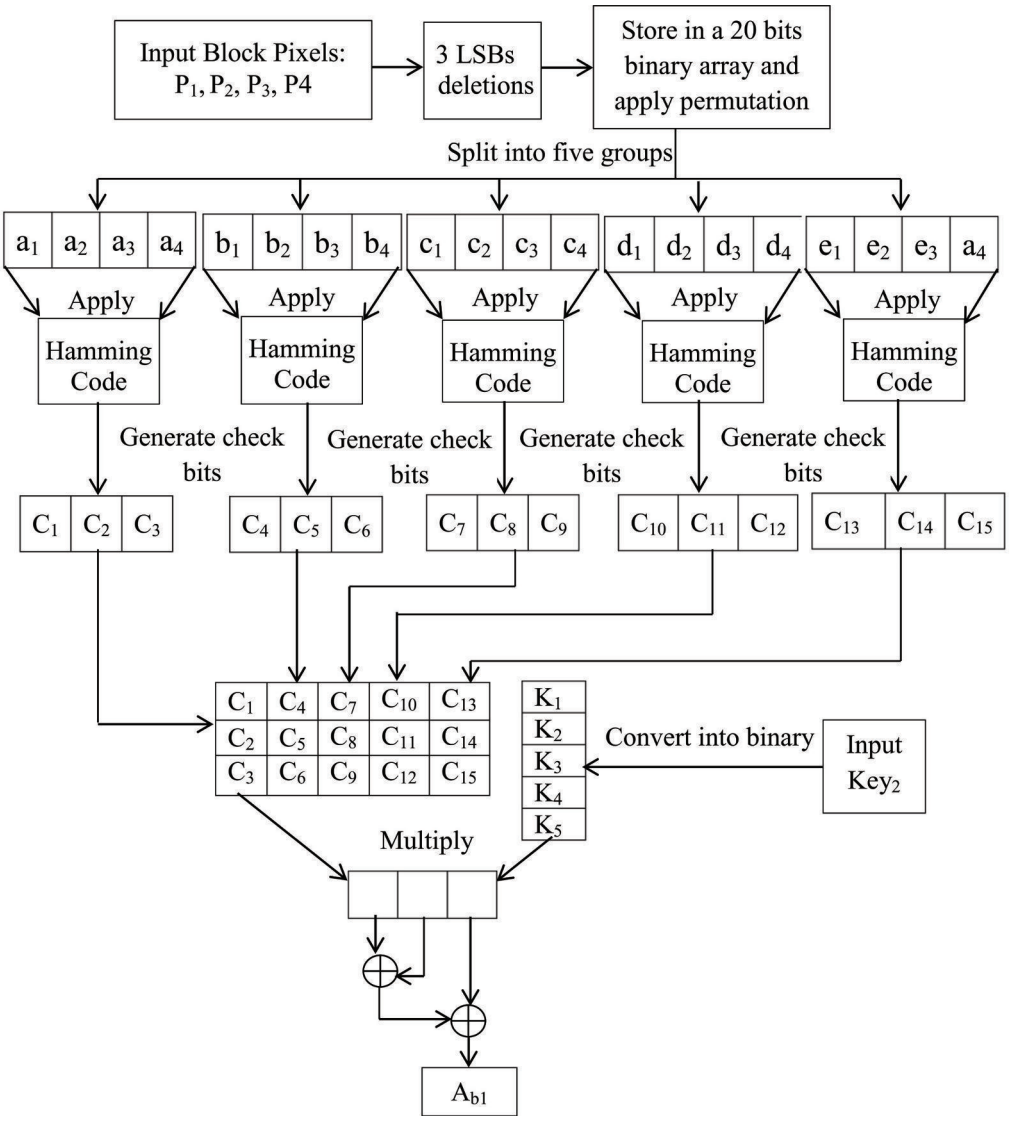


FIGURE 6.2: Block diagram of first authentication bit A_{b1} generation.

Step 3. Generate three bits check bit for each group, using algorithm11(Hamming Code).

Step 4. These check bits store in a matrix M of size 3×5.

Step 5. Input a secret non zero integer Key_2, which is not divisible by 32 and convert it into five bits binary. The five bits binary fill in a 5×1 column vector K_2.

$$\begin{aligned} Key_2 &= mod(Key_2, 32) \\ K_2 &= dec2bin(Key_2, 5) \end{aligned} \tag{6.6}$$

Step 6. Using matrices M and K_2 , Authentication bit A_{b1} generated as follows:

$$M \times K_2 = k_{3\times 1} \tag{6.7}$$

$$A_{b1} = \sum_{m=1}^{3} (k(m))\, mod 2 \tag{6.8}$$

where A_{b1} is the notation of first authentication bit.

Authentication bit A_{b2} Generation

In order to generate second authentication bit, we are using pixel locations. Block Diagram for Authentication bit 2 generation shown in Fig. 6.3. In this authentica-

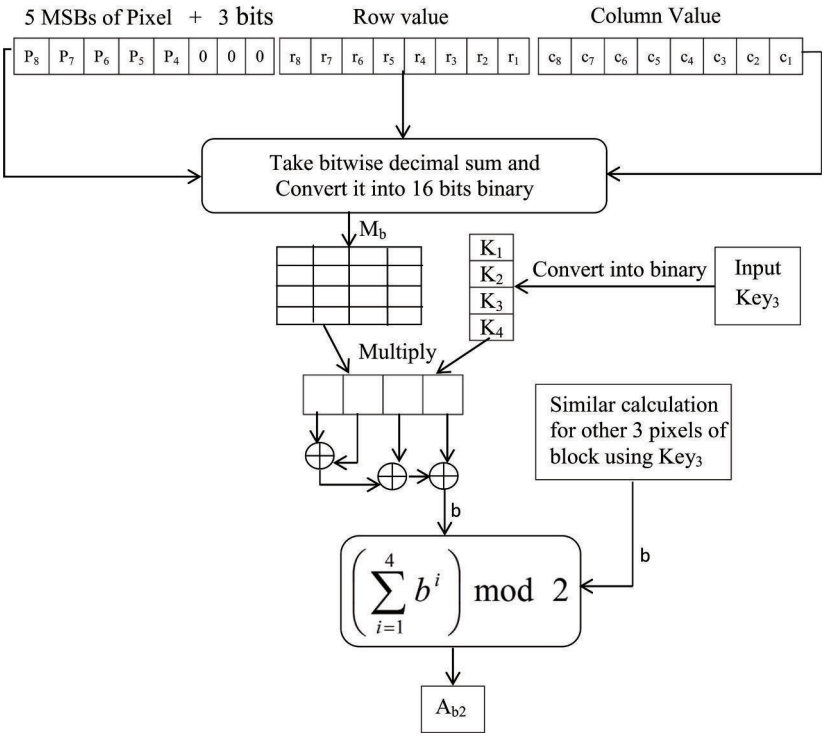


FIGURE 6.3: Block diagram of second authentication bit A_{b2} generation.

tion scheme, we are trying to bind each pixel to the corresponding spatial location. Consider a pixel P_n^b is in n^{th} block and all five MSBs of P_n^b are represented as $P_n^b(m)$ where $m \in \{4, 5, ..., 8\}$. Similarly $P_n^{b,r}$ and $P_n^{b,c}$ are binary value of corresponding row and column value of P_n^b in original spatial image plane. The second authentication bit A_{b2} can be calculated in the following steps:

Step 1. Three dummy bits(i.e. 000) append in the LSB of $P_n^b(m)$ where $m \in \{4, 5, ..., 8\}$ and make eight bits value.

$$f_n^b = AppendLSB(P_n^b, 000), b = 1, 2, 3, 4; \tag{6.9}$$

Step 2. Take the bitwise decimal sum of f_n^b,$P_n^{b,r}$ and $P_n^{b,c}$ and stored in a row vector D of size 1×8. So, value range of each entries of vector D is 0 to 3.

Step 3. Now convert the each entries of the vector D into two bits binary representation and arrange in a binary matrix M_b of size 4×4.

Step 4. Input a secret non-zero integer Key_3, which is not divisible by 16 and convert into four bits binary. These four bits binary fill in a 4×1 column vector K_3.

Step 5. Using matrices M_b and K_3, authentication bit A_{b2} calculated as follows:

$$M \times K_3 = k_{4\times 1} \tag{6.10}$$

$$A_{b2} = \sum_{b=1}^{4} (k(b))\, mod 2 \tag{6.11}$$

where A_{b2} is the notation of second authentication bit.

Algorithm 11 Hamming Code

INPUT: Input the Data Bits : D_3, D_5, D_6, D_7
OUTPUT : Check Bits : C_1, C_2,C_4

1: **procedure** CHECK BITS CALCULATION FOR 7 BITS CODE WORD
2: Determine the number of Check bits by Eq^n: $2^k - 1 \geq m + k$ ▷ m is the data length, k no of check bits
3: Create bit positions, insert check bits and data bits
4: Check bits Calculation ▷ D_3, D_5, D_6 and D_7 is data bits
5: $C_1 = D_3 \oplus D_5 \oplus D_7$ ▷ Here C_1 is the first Check bits
6: $C_2 = D_3 \oplus D_6 \oplus D_7$ ▷ Here C_2 is the second Check bits
7: $C_4 = D_5 \oplus D_6 \oplus D_7$ ▷ Here C_4 is the third Check bits
8: **end procedure**

6.3.1.3 Block Mapping

The restoration bits generated from the block of two partner parts I_i and I_j of the host image should be embedded into the mapped block of other part I_k of the host

Algorithm 12 Smooth Embedding Function

INPUT: Input Block Pixels: P_1, P_2, P_3, P_4, Watermark Vector $V_{1\times12}$,
OUTPUT : Watermarked Block Pixels: X_1, X_2, X_3, X_4

procedure 12 BITS SMOOTH EMBEDDING IN EACH BLOCK
    Split the vector V in four parts i.e. w_1 , w_2, w_3 and w_4 of 3 bits each.
    $W_i = bin2Dec(w_i), i = 1, 2, 3, 4$ ▷ bin2Dec(),used to convert binary value in Decimal
    $P_i = (P_{i,j}\&(1111100)_2), i = 1, 2, 3, 4; j = 8, 7, ..., 1$ ▷ Set 3 LSBs of each pixel to Zero
    for $i = 1$ to 4 **do**
        if $W_i \geq 5$ **then**
            $X_i = P_i + W_i - 8$ ▷ Watermark added
        else
            $X_i = P_i + W_i$ ▷ Watermark added
        end if
    end for
end procedure

image with authentication bits of the mapped block itself. Then the block mapping $\{(i, j) \neq k\}$ condition is required for watermark embedding. For this, the generation algorithm of the block mapping sequence is as follows.

Step 1. Assign a consecutive unique number $i \in \{1, 2, ..., N/16\}$ to each block in the each part of the image in the raster scan order, where $N/16$ is the total number of blocks in a part of host image.

Step 2. Enter a secret non zero integer $Key_4 \in [1, N/16 - 1]$, which is a prime number.

Step 3. For each block number m of part I_i and I_j, apply a 1-D transformation equations 6.12 and 6.13 to acquire n, the number of its mapped block of part I_k.

$$f(m) = (Key_4 \times m) mod \left(\frac{N}{16}\right) \tag{6.12}$$

$$n = f(m) + 1 \tag{6.13}$$

where $m, n(\in [1, N/16])$ are the block number of partner blocks and corresponding mapped block. Here $Key_4 \in [1, \frac{N}{16} - 1]$ must be a prime number in order to acquire a one-to-one mapping; otherwise, the period is less than $N/16$ and a many-to-one mapping may occur [126].

Step 4. The restoration bits of partner block $B_{m,i} \oplus B_{m,j}$ (i.e. block of parts I_i

and I_j) appending with two authentication bits of $B_{k,n}$ are embedded into three LSBs planes of corresponding mapped block $B_{k,n}$ (i.e. block of part I_k) using smooth embedding function (as shown in algorithm 12) and equation 6.5.
In this way, the watermarked image is generated such that each block contains restoration bits of other two partner blocks and authentication bits itself.

6.3.2 Image restoration procedure

In this section, the procedure for tampered image restoration is presented. Suppose that the content of watermarked image is modified or replaced with fake information by anonymous attackers. Receiver's purpose is to first verify the authenticity of the received watermarked image. If the received watermarked image is not authentic, then the receiver will locate the tampered blocks. After identification all tampered blocks, receiver will restore the corresponding extensive content using the restoration bits already extracted from the mapped block. The mapped block is determined with the help of partner blocks. Therefore, the watermark extraction process can be divided into two phases: first is the image authentication and localization procedure, and second one is the tampered block restoration procedure. The block diagram of the image restoration procedure is shown in Fig. 6.4.

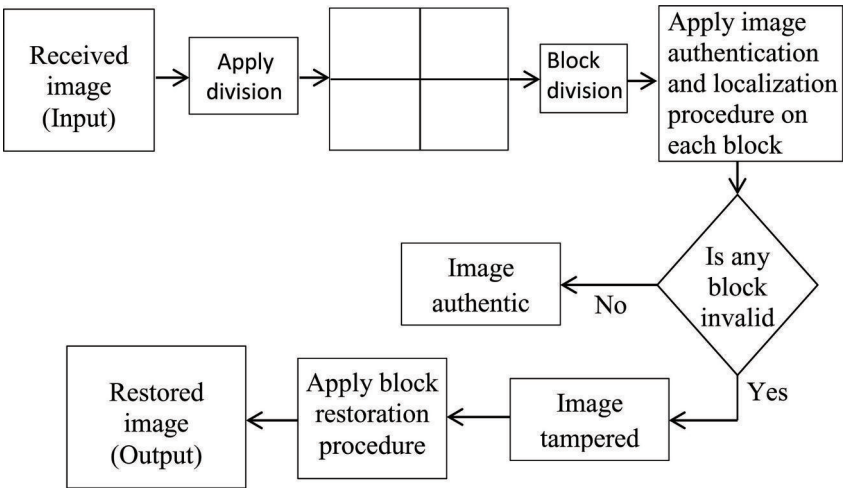


FIGURE 6.4: Block diagram for watermark extraction and image restoration procedure.

6.3.2.1 Image authentication and localization procedure

The modification in the fourth bit position (i.e. fourth LSB) value of each pixel of the received image is required because smoothing function is applied at the time of embedding procedures and fourth bit position value is modified. So, the fourth bit position (i.e. fourth LSB) value of each pixel of the received image is set to "1" , if the decimal value of three LSB of that pixel is greater than or equal to numeric value "5" otherwise fourth bit position value remains unchanged. Further, the received image is divided into non-overlapping blocks of 2×2 pixels, as in the watermark embedding process. The procedure of tampered block identification is described in the following three levels.

Level 1 : On the basis of authentication bits, mark all blocks of the received image "valid" or "invalid". Here, the valid block mark means generated authentication bits of that block totally matched with extracted authentication bits of the block itself, otherwise mark the block invalid. This process will be done in the following steps.

Step 1. First of all, enter the secret integers Key_2 and Key_3 which are used at the time for authentication bits generation.

Step 2. Convert the Key_2 into five bits binary and Key_3 into four bits binary vectors using following equations.

$$\begin{aligned} Key_2 &= mod(Key_2, 32) \\ K_2 &= dec2bin(Key_2, 5) \end{aligned} \tag{6.14}$$

$$\begin{aligned} Key_3 &= mod(Key_3, 16) \\ K_3 &= dec2bin(Key_3, 4) \end{aligned} \tag{6.15}$$

Step 3. Calculate the authentication bits A_{b1} and A_{b2} using the "Authentication bit A_{b1} Generation" and "Authentication bit A_{b2} Generation" procedures .

Step 4. Extracts the authentication bits from 3 LSBs of block itself which were embedded at the time of watermark embedding procedure.

Step 5. Now, compare the calculated authentication bits with the extracted authentication bits, if any mismatch is found, mark that block as "invalid", otherwise, mark the block as "valid".

Level 2 : If a block marks *valid* after *Level* 1, then, its needed to be verified that the block is actually valid or not (i.e. remove the false positive detection problem). It can verify by using the restoration bits in the following steps.

Step 1. Divide the received image I_w into four parts : I_{w1},I_{w2}, I_{w3} and I_{w4} such that,

$$\begin{aligned} I_{w1} \cup I_{w2} \cup I_{w3} \cup I_{w4} = I_w \\ I_{w1} \cap I_{w2} \cap I_{w3} \cap I_{w4} = \phi \end{aligned} \tag{6.16}$$

Step 2. Now identify the valid block in which part of I_w belongs.

Step 3. Identify at least one valid partner block B_{wj} of block B_{wi} and corresponding mapped block B_{wk} from other parts of the cover image.

Step 4. Generate the restoration bits v_i and v_j of blocks B_{wi} and B_{wj}, using "Block Restoration bits Generation" algorithm (as shown in algorithm 10).

Step 5. Extract the embedded restoration bits v_{wk} from 3 LSBs of B_{wk} excluding 2 bits authentication bits of B_{wk}.

Step 6. Now extract the embedded restoration bits of B_{wi} using vectors v_j and v_{wk} by equation 6.17.

$$\hat{v_i} = v_{wk} \oplus v_j \tag{6.17}$$

Step 7. Now compare the generated restoration bits v_i of block B_{wi} with extracted restoration bits $\hat{v_i}$, if any mismatch is found, mark that block as "invalid".

Level 3: In this level, tampered detection is based on 8-neighborhood of the block. After $Level2$, for each block, mark the block "Invalid" if and only if there are five or more invalid blocks in its 3×3 neighborhood otherwise mark the block "valid".

6.3.2.2 Tampered block restoration procedure

After the three level of image authentication and localization procedure, all blocks of the received image I_w are marked either "valid" or "invalid". After identification of all the invalid blocks, these are needed to be restored. The restoration procedure of an invalid block B can be performed from extracted restoration bits. For achieving this goal, embedded restoration bits has to be first extracted from the mapped blocks with the help of partner blocks. Here each block has 2 partner blocks and 2 mapped blocks. So, this way second chance of restoration is possible if one copy of embedded restoration bits is not possible to extract (one mapped block or one partner block are marked as invalid). For restoration of an invalid block, its needed to enter the Key_1 and divide the received image I_w into four parts: I_{w1}, I_{w2}, I_{w3} and I_{w4} such that they follow the condition as given in equation 6.16. After this invalid

block B can restore by considering the following two cases.

Case 1: When at least one copy of restoration bits of B is able to extract First of all, receiver will extract the restoration ten bits V from a valid mapped block of B, with the help of corresponding valid partner block. This extracted restoration bits are decoded by Key_1 as follows.

$$\begin{aligned} Key_1 &= mod(Key_1, 1024) \\ K_1 &= dec2bin(Key_1, 10) \end{aligned} \tag{6.18}$$

$$V(1:10) = V(1:10) \oplus K_1(1:10) \tag{6.19}$$

where dec2bin(), is used to convert decimal value into binary.

After using V, restoration block matrix M_r of size 2×2 of tampered block B is generated by using $BlockRestoration$ algorithm(as shown in algorithm 13). Thereafter, convert all four values of matrix M_r into five binary bits and fill the five MSBs of corresponding pixel of invalid block B of received image and append three zeros in first three LSBs of each pixel to make gray value range from 0 to 255. After the completion of restoration of the block B, mark it as a valid block.

Case 2: When both copy of restoration bits of B are not able to extract In this case invalid block B is restored by its 8-neighborhood blocks of received image I_w and can be performed in following steps.

Step 1. Set three LSBs of only valid 8-neighborhood blocks of block B to zeros and calculate average values of those blocks.

Step 2. Again, take the mean m of calculated average values in step 1.

Step 3. Each pixel of invalid block B is replaced by m.

Step 4. Finally after the completion of restoration of the block B, mark it as a valid block.

Finally, the received image is restored by the above mentioned tampered block restoration procedures. The restored image is very much alike to the corresponding original watermarked image. This is demonstrated by the experimental results.

Algorithm 13 Block Restoration

INPUT: Input extracted restoration bits vector : V

OUTPUT : Matrix M_r of size 2×2

1: **procedure** BLOCK RESTORATION

2: $sign_1 = V(1)$ ▷ $sign_1$ is a variable

3: $R_1(1:4) = V(2:5)$ ▷ R_1 is a vector of size 1×4

4: $r_1 = bin2dec(char(R_1 +' 0'))$ ▷ bin2dec(), used to convert binary values into decimal

5: $sign_2 = V(6)$ ▷ $sign_2$ is a variable

6: $R_2(1:4) = V(7:10)$ ▷ R_2 is a vector of size 1×4

7: $r_2 = bin2dec(char(R_2 +' 0'))$ ▷ bin2dec(), used to convert binary values into decimal

8: **if** $sign_1 \neq 0$ **then**

9: $r_1 = -r_1$

10: **end if**

11: **if** $sign_2 \neq 0$ **then**

12: $r_2 = -r_2$

13: **end if**

14: $M_r = Inv - DCT2\left(\begin{bmatrix} r_1 & r_2 \\ 0 & 0 \end{bmatrix}\right)$ ▷ M_r is a matrix of size 2×2

15: $M_r(i,j) = \lceil (M_r(i,j) + 8) \rceil$ i=1,2; j=1,2

16: $M_r(i,j) = \begin{cases} 0 & \text{if } M_r(i,j) < 0 \ ; \\ M_r(i,j) & \text{if } 0 \leq M_r(i,j) \leq 31 \ ; \\ 31 & \text{if } M_r(i,j) > 31. \end{cases}$ ▷ M_r values are normalized in the range $\in [0, 31]$

17: **end procedure**

6.4 Experimental Results and Discussions

The proposed scheme is fully implemented in MATLAB 13b environment. The computational platform was a Core i7-3770 processor with a speed of 3.40 GHz and 2 GB of RAM. To evaluate the performance of the proposed scheme on color image as well as gray image, two set of test images of size 256×256 are chosen. Fig. 6.5 shows some of the set 1 test images of gray scale. The corresponding watermarked images are shown in Fig. 6.6. The watermark embedding PSNR and NCC value and average embedding time are listed in table 6.1. As the embedding PSNR and NCC values are very high, so it is highly difficult to differentiate between the watermarked and the original image in vision. The embedding time is very less ($\sim$ 6.7 seconds) for an image. This low embedding time indicates that the proposed algorithm is simple and efficient. Similarly, Fig. 6.7 shows some of the set 2 test images of color

FIGURE 6.5: Set 1. Gray scale test images (a) Horse (b)Nature (c) Lena (d) Cameraman (e) Baboon (f) Group Photo 1 (g)Building 1 (h) Building 2

FIGURE 6.6: Watermarked images of Set 1. gray scale images of Fig. 6.5

image. The corresponding watermarked images are shown in Fig. 6.8. In this set of images, the watermark embedding PSNR and NCC value and average embedding time are listed in table 6.2. The embedding PSNR and NCC values are also very high. The embedding time is very less ($\sim$ 20 seconds) for a color image.

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

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

FIGURE 6.7: Set 2. Color images (a) Color Lena (b)Color House (c) Color Couple (d) Color Boat (e) Color Aircraft (f) Color Peppers (g)Color Baboon (h)Color Group Photo

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

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

FIGURE 6.8: Watermarked images of Set 2. Test images of Fig. 6.7

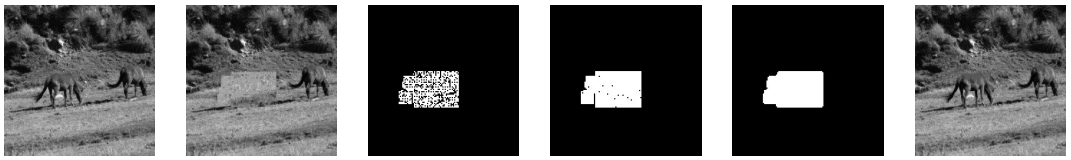
Table 6.1: Essential information observed during watermark embedding in Set 1 Images.

Cover Image (Gray Image)	PSNR(dB) (Embedding)	NCC (Embedding)	Embedding Time (in Sec.)
Horse	40.00	0.9978	6.5832
Nature	39.31	0.9973	6.7080
Lena	39.77	0.9977	6.6456
Cameraman	39.56	0.9986	6.5988
Baboon	39.86	0.9976	6.7548
Group Photo	440.44	0.9988	6.6456
Building 1	40.51	0.9990	6.3960
Building 2	39.15	0.9982	6.6924

Table 6.2: Essential information observed during watermark embedding in Set 2 images.

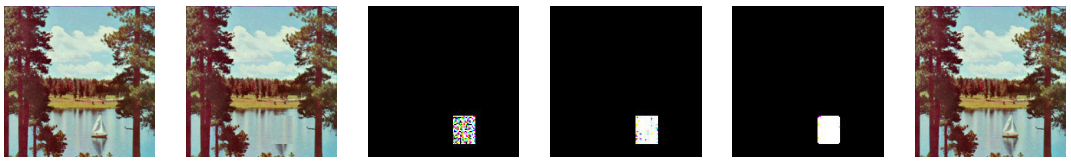
Cover Image (Color Images)	PSNR(dB) (Embedding)	NCC (Embedding)	Embedding Time (in Sec.)
Color Lena	39.7652	0.9971	19.3441
Color House	39.3380	0.9980	19.7497
Color Couple	38.9610	0.9950	19.7341
Color Boat	40.3093	0.9983	19.9993
Color Aircraft	38.9920	0.9967	19.5781
Color Pepper	39.9376	0.9978	19.5781
Color Baboon	40.0445	0.9979	19.8589
Color Group Photo	40.2537	0.9988	19.4689

Fig. 6.9 and Fig 6.10 show the object removal attacks and restored image of gray scale image "Horse" and color image "Color Boat". Figs. 6.9(a) and 6.10 (a) show the original watermarked image. These watermarked images are altered by using object removal attacks. In the "Horse" image, one horse has been removed while boat has been removed from the color boat image. These tampered images are shown in Figs. 6.9(b) and 6.10(b). Figs. 6.9(c) and 6.10(c) show the detected tamper area using level-1 tampered detection method. Figs. 6.9(d) and 6.10(d) show the detected tamper area using level-1and level-2 tampered detection methods. Figs. 6.9(e) and 6.10(e) show the detected tamper area using level-1, level-2 and level-3 tampered detection methods. Figs. 6.9(f) and 6.10(f) show the restored images. The PSNR values, NCC values and average recovered times of the "Horse" recovered image are 41.24 dB, 0.9975, and 7.1604 sec. The PSNR values, NCC values and average



(a) (b) (c) (d) (e) (f)

FIGURE 6.9: Object removal attacks and restored image of gray scale image "Horse" (a) Watermarked Image (b)Object removal attacks (c) Level-1 tampered detection (d) Levels -1,-2 tampered detection (e) Levels-1,-2, -3 tampered detection (f) Restored image

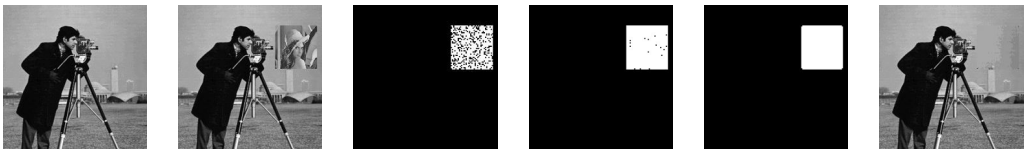


(a) (b) (c) (d) (e) (f)

FIGURE 6.10: Object removal attacks and restored image of color image "Boat" (a) Watermarked Image (b)Object removal attacks (c) Level-1 tampered detection (d) Levels -1,-2 tampered detection (e) Levels-1,-2, -3 tampered detection (f) Restored image

recovered times of the "Color Boat" recovered image are 46.4886 dB, 0.9996, and 20.6233 sec. These high PSNR and NCC values demonstrate that the quality of restored images is very close to original watermarked images.

Fig. 6.11 and Fig. 6.12 show the object addition attacks and restored images of gray scale image "Cameraman" and color image "Color house". Figs. 6.11(a) and 6.12 (a) show the original watermarked image. These watermarked images are altered by using object addition attacks. These tampered images are shown in Figs. 6.11(b) and 6.12(b). Figs. 6.11(c) and 6.12(c) show the detected tamper area using level-1 tampered detection method. Figs. 6.11(d) and 6.12(d) show the detected tamper area using level-1and level-2 tampered detection methods. Figs. 6.11(e) and 6.12(e) show the detected tamper area using level-1, level-2 and level-3 tampered detection methods. Figs. 6.11(f) and 6.12(f) show the restored images. The PSNR values, NCC values and average recovered times of the "Cameraman" recovered images are 43.97 dB, 0.9994, and 7.1136 sec. The PSNR values, NCC values and average recovered times of the "Color House" recovered images are 37.0755 dB, 0.9934, and 22.3861 sec.



(a) (b) (c) (d) (e) (f)

FIGURE 6.11: Object addition attacks and restored image of gray scale image "Cameraman" (a) Watermarked Image (b)Object addition attacks (c) Level-1 tampered detection (d) Levels -1,-2 tampered detection (e) Levels-1,-2, -3 tampered detection (f) Restored image

(a) (b) (c) (d) (e) (f)

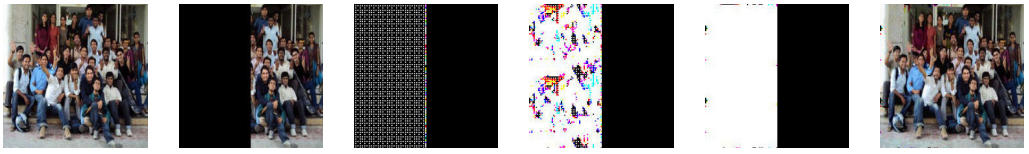
FIGURE 6.12: Object addition attacks and restored image of color image "House" (a) Watermarked Image (b)Object addition attacks (c) Level-1 tampered detection (d) Levels -1,-2 tampered detection (e) Levels-1,-2, -3 tampered detection (f) Restored image

Fig. 6.13 and Fig. 6.14 show the cropping (50%) attacks and restored images of gray scale image "Group Photo" and color image "Color Group Photo". Figs. 6.13(a) and 6.14 (a) show the original watermarked image. The cropped images are shown in Figs. 6.13(b) and 6.14(b). Figs. 6.13(c) and 6.14(c) show the detected tamper area using level-1 tampered detection method. Figs. 6.13(d) and 6.14(d) show the detected tamper area using level-1and level-2 tampered detection methods. Figs. 6.13(e) and 6.14(e) show the detected tamper area using level-1, level-2 and level-3 tampered detection methods. Figs. 6.13(f) and 6.14(f) show the restored images. The PSNR value, NCC value and average recovered time of the "Group Photo" recovered images are 35.75 dB, 0.9958, and 8.6269 sec. The PSNR value, NCC value and average recovered time of the "Color Group Photo" recovered images are 33.8454 dB, 0.9766, and 25.9898 sec.

The essential restoration information (PSNR (dB), NCC, and average recovered time (Tr in Sec.))for various tampering rates for gray scale images and color images are listed in table 6.3 and table 6.4 respectively. Fig. 6.15 and Fig. 6.16 show the PSNR and NCC of the recovered content in the tampered area with respect to the tampering rates for gray scale images and color image respectively. In theses figures,

(a) (b) (c) (d) (e) (f)

FIGURE 6.13: Cropping attacks and restored image of gray scale image "Group photo" (a) Watermarked Image (b)Cropping(50 %) attacks (c) Level-1 tampered detection (d) Levels -1,-2 tampered detection (e) Levels-1,-2, -3 tampered detection (f) Restored image



(a) (b) (c) (d) (e) (f)

FIGURE 6.14: Cropping attacks and restored image of color image "Color group photo" (a) Watermarked Image (b)Cropping(50 %) attacks (c) Level-1 tampered detection (d) Levels -1,-2 tampered detection (e) Levels-1,-2, -3 tampered detection (f) Restored image

TABLE 6.3: PSNR (dB), NCC, Recovered time (Tr in Sec.) of recovered content in the tampered areas with different tampering rates of gray scale images.

Cover Image	perfor-menace	Tampering rate							
		5%	10%	20%	24%	35%	40%	45%	50%
Lena	PSNR	45.12	42.42	38.78	38.17	36.81	36.45	35.92	35.55
	NCC	0.999	0.999	0.998	0.997	0.995	0.994	0.993	0.993
	Tr	6.864	6.879	7.285	7.488	8.049	8.377	8.486	8.517
Horse	PSNR	44.11	40.91	38.02	37.23	35.53	34.95	34.40	33.94
	NCC	0.999	0.997	0.995	0.995	0.991	0.988	0.988	0.986
	Tr	6.676	6.864	7.753	7.550	8.002	8.127	8.408	8.580
Camera -man	PSNR	46.49	43.92	40.77	39.63	36.87	36.06	35.45	34.98
	NCC	0.996	0.996	0.995	0.994	0.993	0.993	0.991	0.989
	Tr	6.957	7.082	7.378	7.41	7.972	8.081	8.268	8.533
Baboon	PSNR	44.05	40.56	37.24	36.44	34.65	34.08	33.63	33.33
	NCC	0.999	0.996	0.992	0.990	0.986	0.984	0.983	0.982
	Tr	6.755	7.176	7.426	7.394	8.034	8.003	8.175	8.409
Group Photo	PSNR	47.75	44.2	40.84	39.98	37.88	37.14	36.3	35.75
	NCC	0.999	0.999	0.999	0.998	0.997	0.997	0.996	0.996
	Tr	6.677	6.926	7.301	7.504	7.972	8.159	8.190	8.299

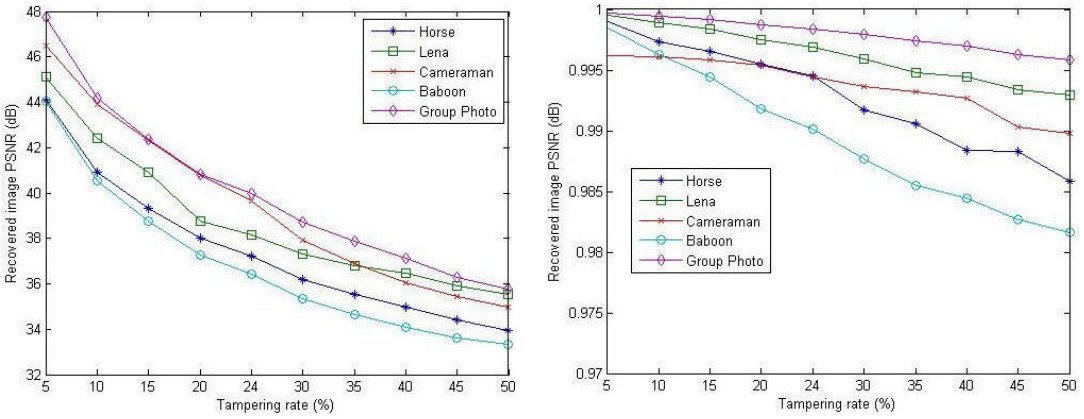


FIGURE 6.15: (a) PSNR of restored content with respect to the tampering rates of the gray images. (b)NCC of restored content with respect to the tampering rates of the gray images.

TABLE 6.4: PSNR (dB), NCC, Recovered time (Tr in Sec.) of recovered content in the tampered areas with different tampering rates of color images.

Cover Image	perfor-mence	Tampering rate							
		5%	10%	20%	24%	35%	40%	45%	50%
Color Lena	PSNR	45.27	42.51	38.89	38.16	36.23	35.56	34.97	34.47
	NCC	0.999	0.999	0.997	0.996	0.994	0.993	0.992	0.991
	Tr	22.18	22.45	23.54	24.10	25.35	25.69	26.41	26.69
Color Group Photo	PSNR	44.71	41.34	37.71	37.01	35.41	34.81	34.32	33.85
	NCC	0.994	0.994	0.983	0.983	0.979	0.977	0.977	0.975
	Tr	21.48	22.51	23.57	23.71	25.59	25.76	26.52	26.91
Color House	PSNR	44.08	40.59	37.61	36.93	35.35	34.82	34.27	33.83
	NCC	0.999	0.997	0.994	0.993	0.991	0.990	0.989	0.988
	Tr	20.09	20.89	22.28	22.59	24.73	24.94	25.87	26.08
Color Boat	PSNR	44.15	41.35	38.43	37.52	35.23	34.57	34.04	33.61
	NCC	0.999	0.999	0.997	0.996	0.994	0.993	0.993	0.992
	Tr	21.04	22.07	22.87	23.99	24.49	24.82	25.55	26.33
Color Aircraft	PSNR	43.99	40.47	37.47	36.75	35.39	34.99	34.59	34.27
	NCC	0.999	0.997	0.993	0.992	0.988	0.987	0.985	0.984
	Tr	21.73	22.49	23.15	23.63	25.83	26.13	26.53	26.55
Color Baboon	PSNR	43.97	40.54	37.43	36.65	35.11	34.58	34.08	33.68
	NCC	0.999	0.997	0.994	0.993	0.989	0.988	0.987	0.986
	Tr	21.81	22.32	23.32	23.89	25.33	25.48	26.02	26.39

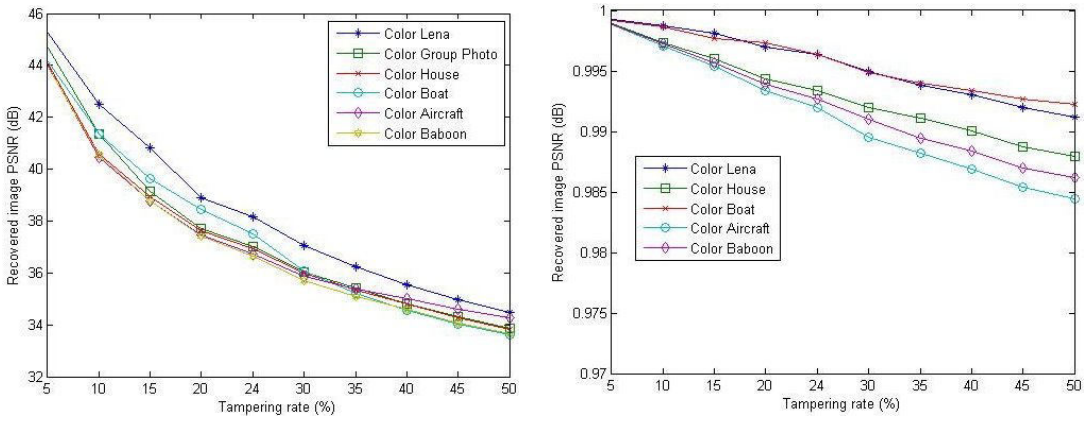


FIGURE 6.16: (a) PSNR of restored content with respect to the tampering rates of the color images. (b)NCC of restored content with respect to the tampering rates of the color images.

the curve of the recovered PSNR and NCC values with respect to tampering rate decreases smoothly but even if the tampering rate is 50%, the restored images have PSNR and NCC values effectively high. So, we can say that the restored image is very similar to the corresponding watermarked image.

The proposed watermarking scheme is compared with six schemes [65, 66, 67, 78, 111, 151]. In [66], authors proposed two schemes called here as [66]-A and [66]-B for simplicity. Here three gray scale images, Lena, Cameraman, and Baboon were used to compare the content restoration performance . Figs. 6.17, 6.18 and 6.19 give the PSNR values of restored image with respect to different tampering rates. The tampering rate varying from 5% to 50% with interval of 5%. These plots validate the efficiency of the proposed scheme and existing schemes. The restored image in the proposed scheme has effectively higher PSNR values than the method used in schemes [65, 67, 78, 111, 151] and [66]-B. The first method [66]-A in the scheme [66], gives high PSNR values, but does not work with a large tampering rate. In this method, the extractable restoration bits are used to retrieve the original values of tampered MSBs by applying Gaussian elimination method. In the Gaussian elimination method, if the number of unknowns is too large, the solution may not be unique and, in this case, we cannot find the true solution in the solution space. Actually, if the tampered area is not too extensive, the reference-bits extracted from reserved blocks can provide sufficient information to recover the first type of MSB.

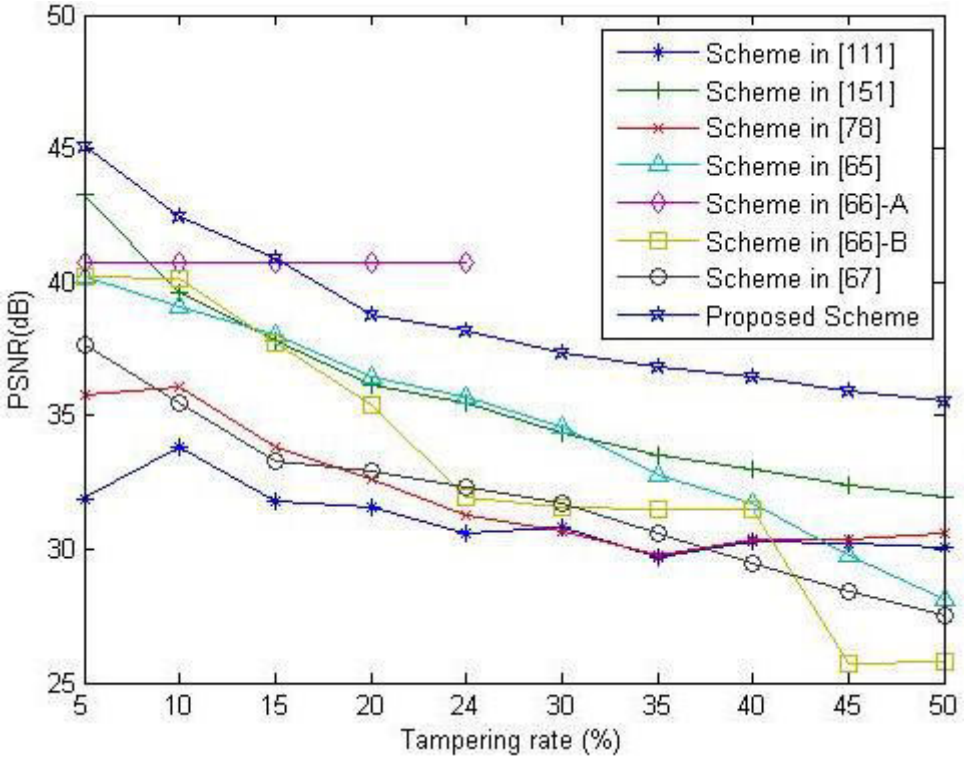


FIGURE 6.17: PSNR(dB) of restored "Lena" image under varying tampering rates.

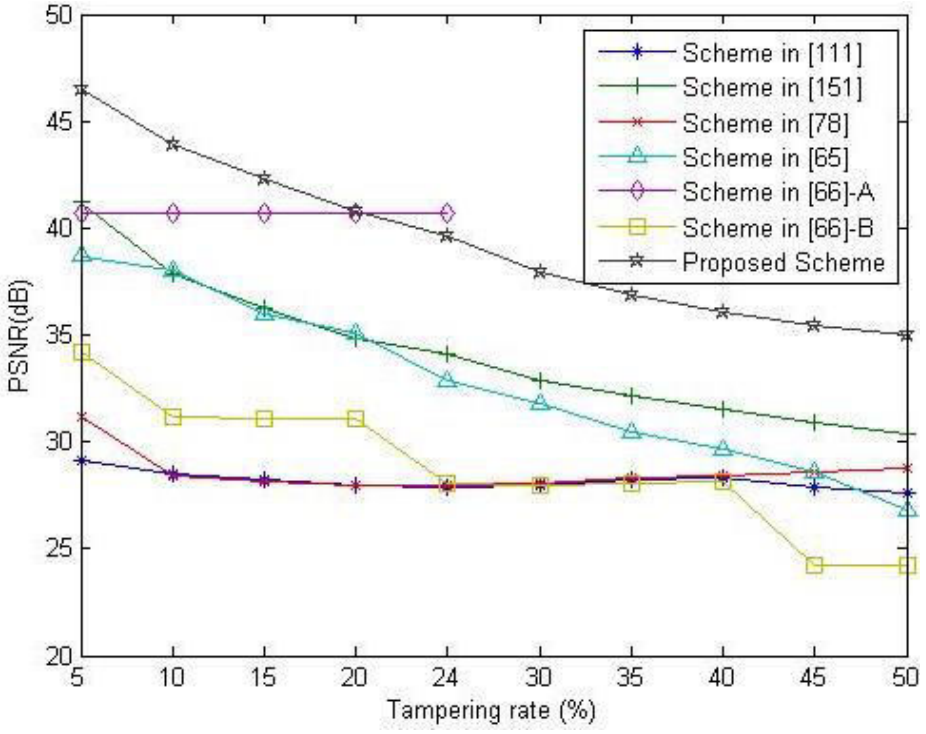


FIGURE 6.18: PSNR(dB) of restored "Cameraman" image under varying tampering rates.

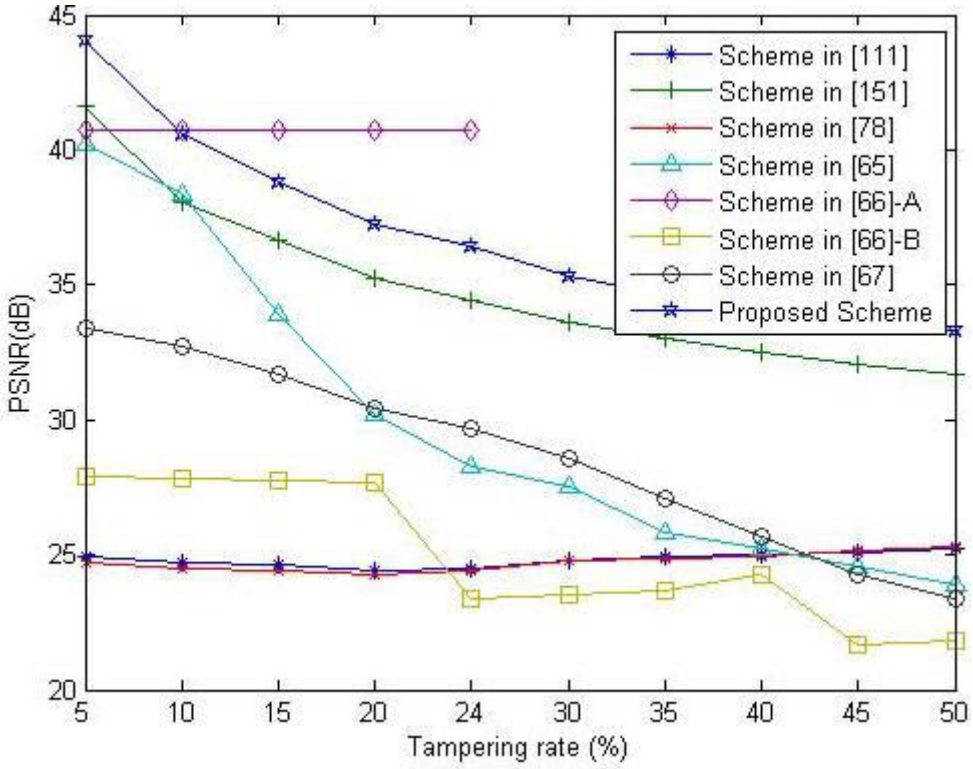


FIGURE 6.19: PSNR(dB) of restored "Baboon" image under varying tampering rates.

It is seen that all the MSB of tampered blocks can be recovered when the tampering rate is no more than 24%.

The schemes [65, 67, 78] and [66]-B are using large size of block (i.e . 8×8). If a single pixel of a block is tampered in block wise watermarking scheme, the whole block pixels treated as tampered. For example only a single pixel in a block of size 8×8 is tampered, the whole sixty four pixels of that block will be treated as tampered region and need to recover the all pixels while 63 pixels are not tampered. Hence, the accuracy of tampered localizations is decreased and also encounters the blocking artifacts. Although, the scheme [151] is good in localization accuracy and free from blocking artifacts, but this scheme provides only single chance for restoration. The scheme [111] provides two chances for restoration, but restoration bits are just average value of the block. So, its restoration quality is not good enough. On the other hand, the proposed scheme is using very small size of blocks, three-level hierarchical tampered detection mechanisms. Its also provides two chances of restoration

capability. Hence, this scheme provides high accuracy in the tampered pixel localization and effectively restored the tampered image. The blocking artifacts are also negligible. Thus, the proposed scheme is superior to the compared schemes.

6.5 Conclusion

This chapter describes DCT based new fragile watermarking scheme for tamper detection, localization and self-recovery. This scheme provides three-level hierarchical tampered detection mechanisms and two chances for tamper restoration. For the authentication bits and restoration bits generation and embedding, a small non overlapping block of size 2×2 is used. Hence, the accuracy of tampered localization is improved. The restoration bits are generated from the DCT coefficients of the cover image. Two copies of restoration bits of each block are embedded into the watermarked image. Experimental results demonstrate that the proposed scheme is capable to restore with high quality upto 50% tampering rate from object removing, object adding and cropping attacks. Future research involves extending this approach to resist some other common image processing operations like JPEG compression, filtering and enhance the quality of the restored image.