Chapter 4

DCT with Quantization Matrix based Fragile Watermarking Scheme for Image Authentication and Restoration

4.1 Introduction

With the powerful image and signal processing technologies and the wide availability of multimedia editing tools, digital image content is prone to illegal manipulations and alterations. The content of a digital image can easily be replaced with fake content. In the critical situations, such as for evidence in a court of law, a small amount of content modification in digital imaging can change the judgment[125]. Therefore, in these days the tampered image detection and localization are important issues. This problem can be overcome by using digital watermarking techniques. Digital watermarking is a technique of imperceptibly altering an image to embed secret message or information about that image. The image, in which secret message is embedded is known as cover or host image and the secret message or information is known as watermark [3, 111, 126]. In addition to the content authentication bits which are used for authentication purposes, recovery (i.e restoration or reference) bits are used to recover the corrupted content of the host image. The recovery bits

are basically the compressed form of the host image called image digest and describe the content of the host image. The watermark scheme in which watermark generated from the cover image is known as self-embedding watermarking scheme.

Due to sensitivity towards the modification, fragile watermarking schemes are used (area like court of evidence) where the exact authentication required. Fragile watermarking schemes for authentication, can be further categorized into two main categories as pixel-wise fragile watermarking schemes [10, 51, 52, 127] and block-wise fragile watermarking schemes[9, 13, 63, 78, 111, 125, 126, 128, 129]. In pixel-wise fragile watermarking schemes, the watermark information generated from the gray values of host pixels and is embedded into the host pixels themselves. In block-wise fragile watermarking schemes, the cover image is divided into sub-blocks. Each sub-block has watermark information and is authenticated by the successful retrieval of the watermark embedded in it. If the watermarked image is modified (malicious or non-malicious), the watermark of a particular sub-block is not retrieved successfully, then that sub-block is identified as tampered or invalid block. The least significant bit (LSB) method in data hiding has the benefit of simplicity, easily detection and high embedding capacity[130, 131]. So, the least significant bit (LSB) method is generally preferred for watermark embedding in the fragile image watermarking.

The rest part of this chapter is organized as follows. Section 4.2 gives a review of related work. Section 4.3 describes the watermark embedding and extraction procedures of the proposed scheme. The experimental results and analysis are given in Section 4.4. Finally, the concluding remarks are given in section 4.5.

4.2 Related work

The performance of self-embedding schemes are generally given in terms of recovered image quality and recovery (or restoration) conditions. The recovered image quality can be decided by a peak signal to noise ratio (PSNR) and the restoration conditions are typically the tampering rate (i.e. amount of modifications). These recovered image quality and tampering rate are strongly interrelated, and the trade-off between them needs to be properly balanced. Due to this trade-off, there is no general model

for the content recovery problem despite the existence of the variety of schemes that are capable to recover the tampered content[63, 126, 129].

In many self-embedding schemes, the recovery bits of a particular sub-block are always hidden in another block (or blocks) of the image. This recovery may fail when the block (or blocks) containing its recovery bits is (are) tampered. This is called tampering coincidence problem. The schemes [9, 111, 128, 132, 133, 134] are not able to handle this problem. Although, in the schemes [111, 133], two copies of recovery bits were embedded into the image and the second chance for block recovery was provided. In [134], authors proposed a hierarchical watermarking scheme. In this scheme, four levels tampered detection procedures were used. It was based on 2 bits authentication watermark and performed in 2×2 sized sub-blocks of a block of size 4×4. This scheme is not robust against the collage and vector quantization(VQ) attacks [135] because the authentication bits of a block were independent of the other blocks. The recovery bits are just an average value of 6 MSB-layers of sub-blocks. So, the quality of recovered image is also not very good in this scheme.

The authors of schemes [63, 65] proposed self -embedding watermarking scheme using a reference sharing mechanism to resolve the tampering coincidence problem. However, the quality of the recovered tampered image in both schemes is improve by embedding the redundant information in the cover image, but both schemes are not able to resist the VQ attack due to the independence between blocks. The localization accuracy is also reduced because of using a large block size of 8×8.

Zhang et al. [66] also proposed two self-embedding watermarking schemes by utilization of reference sharing mechanism. The watermark was derived from 5 MSB-layers of the cover image. In the second technique, the cover image was decomposed into three levels based on the hierarchical self-embedding scheme. The first scheme, the original data in five layers of original 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 data at different levels. So that a better recovered image can be obtained in second scheme, from a tampered version with less fake content. The second scheme is capable to recover up to 66% tampering rate.

To avoid tampering coincidence problem, another self-embedding watermarking scheme was proposed by Zhang et al.[67], in which watermark of a block was scattered in the whole image. In this scheme, the image pixels were permuted using a secret key and then divided them into a series of pixel-pairs. The recovery bits were generated by the XOR operation within the 5 MSB-layers of pixel pairs and the authentication bits were derived from 5 MSB-layers and recovery bits. Finally, watermark data were embedded into the three LSB-layers. 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 of restoring up to 54% tampering rate.

In [64], Qian et al. proposed a watermarking scheme based on Discrete Cosines Transform (DCT) to avoid tampering coincidence problem. 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 the accuracy of tampered localization decreases because of using a large block size.

Zhang et al.[71] proposed a self-embedding fragile watermarking scheme based on DCT and fractal compression coding. In this scheme, the watermark was generated by using interleaved and overlapped block structure. Three versions of recovery bits of each block were embedded into different quadrants, which provide two chances for block recovery in case of one is destroyed. However, in this scheme the accuracy of tamper localization also decreases because of using a large block size.

To improve the quality of recovered image, an effective self-embedding watermarking scheme is proposed in this paper. Here, a block-wise mechanism is used for tampered region detection, localization and recovery. In this scheme authentication bits are embedded in the block itself instead of the mapped block, so the false tampered detection error can be minimized. This scheme is also effective because it is using two level tampered detection procedures. So the accuracy of tamper detection and localization is enhanced. During tampered detection procedures, if tampered areas are not detected at level-1 examination, it will be detected at level-2. So, the tampered regions are detected with the high probability. This scheme is using very small size of the block (i.e. 2×2) and is also secured by using predefined user keys. There are some strong points in the scheme: (1) Very less sensitive to error pixels;

(2) higher localization accuracy; (3) The high quality of recovered image because DCT coefficients are used for recovery; (4) negligible blocking artifacts in recovered image. If the recovery bits are unable to extract from mapped block, then recovery in this scheme is based on its 3×3 neighborhood blocks, Hence, the tampering coincidence problem with this scheme is also avoided. Experimental results show that the proposed scheme allows high quality recovery up to 50% tampering rate.

4.3 Proposed Algorithm

The proposed watermarking scheme can be explained in three steps: watermark generation and embedding, tamper detection and localization and tamper image recovery procedures. Let the cover image I has M_1 rows and M_2 columns where both M_1 and M_2 are multiples of 2. Let M represent the total number of pixels ($M = M_1 \times M_2$).

4.3.1 Watermark generation and embedding procedure

Watermark is generated from the five MSB –layers of the cover image itself. The three LSB –layers of the cover image are substituted with the watermark bits. Two types of the watermark are generated for each block of size 2×2, where the first type of watermark is called authentication bits (2 bits) which are used to locate and detect the tampered blocks and the second type of watermark is called recovery bits (10 bits), which are used to recover the content of the corrupted image. Authentication bits are embedded into the block itself and the recovery bits are embedded into the mapped image block to establish block dependency. Fig. 4.1 depicts the watermark generation and embedding procedures. The details of the watermark generation and embedding procedure are as follows.

4.3.1.1 Recovery bits generation

For recovery bits generation, cover image I is divided into non-overlapped blocks of size 2×2, and denote the n^{th} block pixels as $P_n(i, j)$, where $1 \leq n \leq M/4$, $1 \leq i \leq 2$,

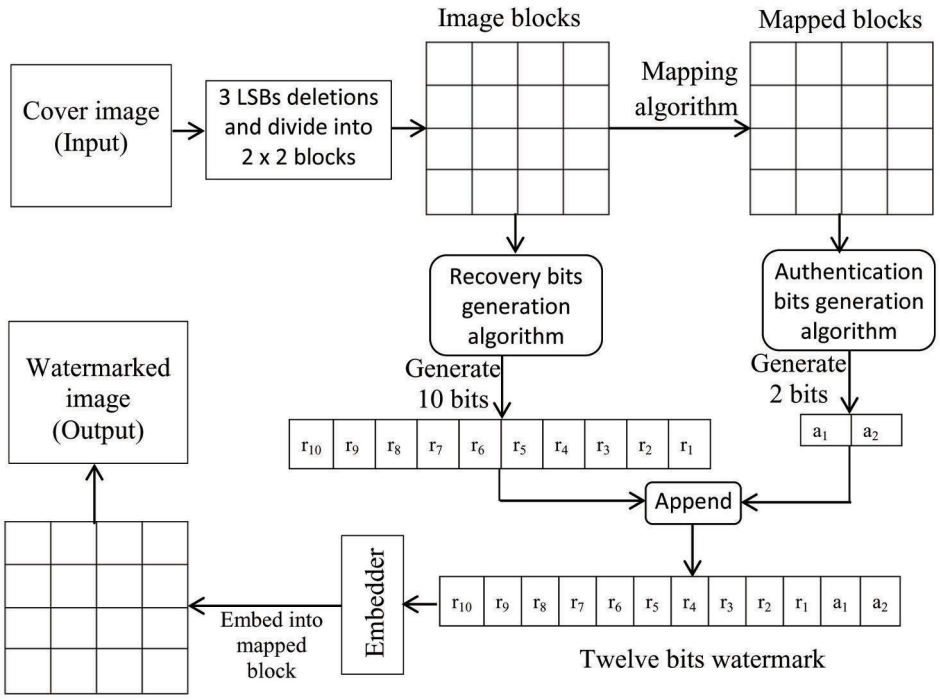


FIGURE 4.1: Block diagram of watermark embedding procedure of a block

$1 \leq j \leq 2$. For each block 10 bits recovery information $R_{1\times 10}$ generated by 5 MSB-layers of the block pixels using "Block recovery bits generation" algorithm. Fig. 4.2 illustrates the block diagram of the recovery bits generation process and the pseudo code of "Block recovery bits generation" is given in the algorithm 1. The algorithm 1 is based on DCT technique and quantization matrix Q, where $Q = \begin{bmatrix} 16 & 11 \\ 12 & 12 \end{bmatrix}$. The sign() function used in algorithm 1 is given by equation(4.1).

$$sign(x) = \begin{cases} 0 & \text{if } x \geq 0; \\ 1 & \text{if } x < 0. \end{cases} \tag{4.1}$$

The Key_1 is a secret non-negative integer which is not divisible by 1024. The Key_1 is used here to increase the security of recovery bits.

4.3.1.2 Authentication bits generation

For each block, two authentication bits are generated as follow:

Algorithm 1 Block Recovery bits Generation

INPUT: n^{th} block pixels of cover image: $P_n(i,j)$, Quantization matrix Q, Key_1

procedure RECOVERY BITS GENERATION

$g_n(i,j) \longleftarrow \lfloor \frac{P_n(i,j)}{8} \rfloor, 1 \le n \le M/4, 1 \le i \le 2, 1 \le j \le 2.$ ▷ Delete the 3LSBs from each pixels

$g_n(i,j) \longleftarrow g_n(i,j) - 16, 1 \le n \le M/4, 1 \le i \le 2, 1 \le j \le 2$ ▷ Subtract each pixel by half the number of maximum possible value

$D_n \longleftarrow DCT2\left(\begin{bmatrix} g_n(1,1) & g_n(1,2) \\ g_n(2,1) & g_n(2,2) \end{bmatrix}\right), 1 \le n \le M/4$ ▷ D_n is the DCT coefficients of the block.

$M_n(i,j) \longleftarrow \frac{D_n(i,j)}{Q}, 1 \le n \le M/4, 1 \le i \le 2, 1 \le j \le 2$ ▷ Generate quantized matrix M_n

$r_n(i,j) \longleftarrow round(M_n(i,j)), 1 \le n \le M/4, 1 \le i \le 2, 1 \le j \le 2$

$R(1:10) \longleftarrow 0$ ▷ Initialization the recovery bits vector R to the zero

for $i \longleftarrow 1$ to 2 **do**

  for $j \longleftarrow 1$ to 2 **do**

    Select first largest value $r_n(i,j)$ from matrix r_n

    $R(1) \longleftarrow i - 1$ ▷ Row value of location $r_n(i,j)$ in n^{th} block

    $R(2) \longleftarrow j - 1$ ▷ Column value of location $r_n(i,j)$ in n^{th} block

    $R(3) \longleftarrow sign(r_n(i,j))$ ▷ Value of $r_n(i,j)$ is positive or negative

    $r_1 \longleftarrow abs(r_n(i,j))$

    if $abs(r_1 > 3$ **then**

      $r_1 \longleftarrow 3$

    end if

    $R(4:5) \longleftarrow dec2Bin(r_1, 2)$ ▷ dec2Bin(), used to convert decimal value into binary

    Select second largest value $r_n(i,j)$ from matrix r_n

    $R(6) \longleftarrow i - 1$ ▷ Row value of location $r_n(i,j)$ in n^{th} block

    $R(7) \longleftarrow j - 1$ ▷ Column value of location $r_n(i,j)$ in n^{th} block

    $R(8) \longleftarrow sign(r_n(i,j))$

    $r_2 \longleftarrow abs(r_n(i,j))$

    if $abs(r_2 > 3$ **then**

      $r_2 \longleftarrow 3$

    end if

    $R(9:10) \longleftarrow dec2Bin(r_2, 2)$ ▷ dec2Bin(), used to convert decimal value into binary

  end for

end for

$Key_1 \longleftarrow mod(Key_1, 1024)$

$K_1 \longleftarrow dec2Bin(Key_1, 10)$

$R(1:10) \longleftarrow R(1:10) \oplus K_1(1:10)$ ▷ Recovery bits encoded by Key_1

end procedure

OUTPUT : Recovery bits vector R of size 1×10

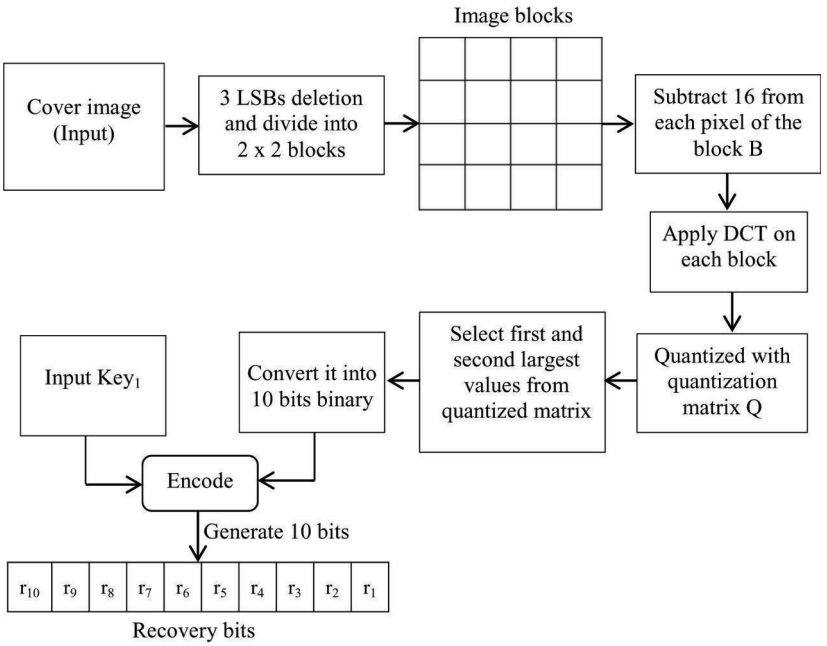


FIGURE 4.2: Block diagram of recovery bits generation procedure

First authentication bit (A_{b1}) generation

The first authentication bit (A_{b1}) of a block is generated from the pixel coordinates along with its 5 MSBs value. Fig. 4.3 illustrates the block diagram of the A_{b1} generation process. Consider n^{th} block pixels as $P_n(i,j)$, where $1 \leq n \leq M/4$, $1 \leq i \leq 2$, $1 \leq j \leq 2$. Similarly $P_n^r(i,j)$ and $P_n^c(i,j)$ are row and column values of $P_n(i,j)$ in the spatial image plane. The pseudo code of first authentication bit generation A_{b1} is given in the algorithm 2.

Second authentication bit (A_{b2}) generation

The second authentication bit (A_{b2}) generation is based on the check bits generated from longitudinal redundancy check (LRC) and the mean value of the pixels of the block. Fig. 4.4 illustrates the block diagram of the A_{b2} generation process. Consider n^{th} block pixels as $P_n(i,j)$, where $1 \leq n \leq M/4$, $1 \leq i \leq 2$, $1 \leq j \leq 2$. Similarly $P_n^r(i,j)$ and $P_n^c(i,j)$ are the row and column values of $P_n(i,j)$ in the spatial image plane. The pseudo code of second authentication bit generation A_{b2} is given in the algorithm 3.

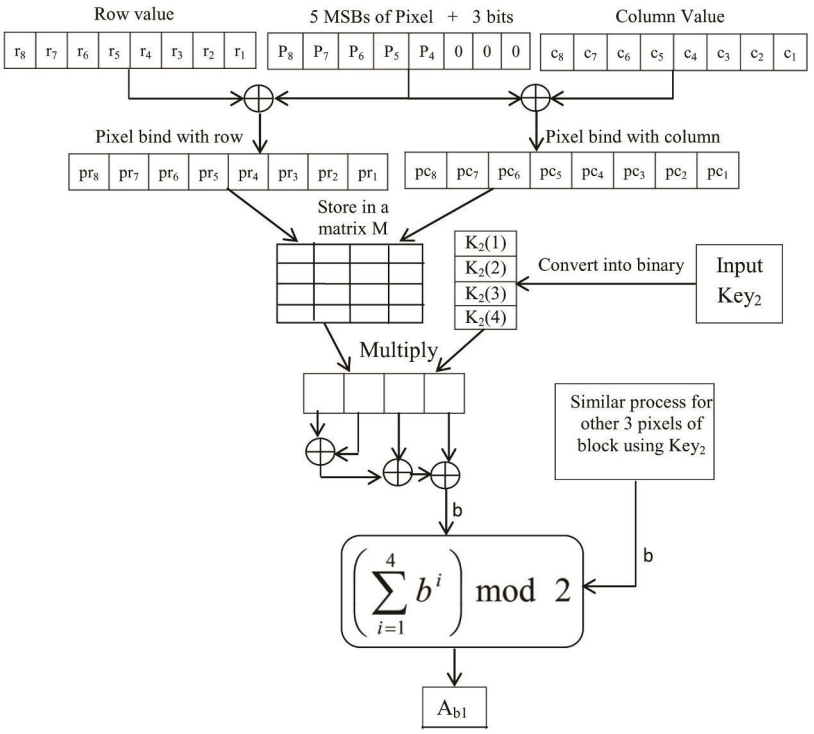


FIGURE 4.3: Block diagram of first authentication bit A_{b1} generation procedure

4.3.1.3 Block Mapping

In the proposed scheme, the recovery bits generated from a block B_i are not embedded into block itself. These recovery bits are embedded in 3 LSBs of the mapped block B_j with authentication bits of the mapped block. Here blocks B_i and B_j are chosen such that $\{i \neq j, \forall i, j | i, j \in [1, N]\}$. where N is the total number of blocks in the cover image. This way 12 bits watermark (as shown in Fig. 4.5) is embedded into 3 LSBs of each block. In the cover image, assign a unique number $i \in \{1, 2, ..., N\}$ to each image block in the raster scan order, where N is the total number of blocks in the image. Then, the mapping block sequence is generated from algorithm 5 using a secret random number Key_3. For one-to-one mapping sequence generation, here $Key_4 \in [1, N-1]$ must be a prime number; otherwise, the period is less than N and many-to-one mapping may occur[126].

Algorithm 2 First authentication bit generation

INPUT: n^{th} block pixels of cover image and the pixel coordinates: $P_n(i,j)$, $P_n^r(i,j)$ and $P_n^c(i,j)$

procedure A_{b1} BIT GENERATION
  $b_n(1:8) \longleftarrow dec2Bin(P_n(i,j),8), 1 \leq n \leq M/4, 1 \leq i \leq 2, 1 \leq j \leq 2$ ▷ 8 bits binary conversion of each pixel
  $b_n(1:3) \longleftarrow 0$ ▷ Set 3 LSBs of each pixel to zeros
  $b_n^r(1:8) \longleftarrow dec2Bin(P_n^r(i,j),8)$ ▷ 8 bits binary conversion of row value of each pixel
  $b_n^c(1:8) \longleftarrow dec2Bin(P_n(i,j),8)$ ▷ 8 bits binary conversion of column value each pixel
  $pr_n(1:8) \longleftarrow b_n(1:8) \oplus b_n^r(1:8)$ ▷ bind pixel value with row value
  $pc_n(1:8) \longleftarrow b_n(1:8) \oplus b_n^c(1:8)$ ▷ bind pixel value with column value
  Matrix $M_{4\times4}$ is filled by $pr_n(1:8)$ and $pc_n(1:8)$
  Generate a random number Key_2 using a seed value
  $Key_1 = mod(Key_2, 16)$
  $K_2 = dec2bin(Key_2, 4)$ ▷ dec2Bin(), used to convert decimal value into binary
  $k_{4\times1} = M \times K_2'$
  $a^1 = \sum_{m=1}^{4} (k(m)) \, mod2$
  Similar to a^1, a^2, a^3 and a^4 calculated for remaining 3 pixels of the block
  $A_{b1} = \sum_{m=1}^{4} (a(m)) \, mod2$
end procedure

OUTPUT : A_{b1}

4.3.1.4 Watermark embedding process

In embedding process, ten bits recovery information of a block B_m and two bits authentication information from corresponding mapped block B_n is stored in a vector W of size 1×12 (as shown in Fig. 4.5). The watermark W is embedded in 3 LSB-layers of block B_j using embedding algorithm. The pseudo code of embedding algorithm is given in algorithm 6. After making embedding process for all blocks in the cover image (I), watermarked image(I_w) is generated. In this way, five MSB-layers of cover image are preserved and three LSB-layers are replaced with the watermark in the generated watermarked (I_w) image.

Algorithm 3 Second authentication bit generation

INPUT: n^{th} block pixels of cover image and the pixel coordinates: $P_n(i,j)$, $P_n^r(i,j)$ and $P_n^c(i,j)$

procedure A_{b2} BIT GENERATION
  $b_n(i,j) \longleftarrow \lfloor \frac{P_n(i,j)}{8} \rfloor, 1 \leq n \leq M/4, 1 \leq i \leq 2, 1 \leq j \leq 2.$ ▷ Delete the 3LSBs from each pixels
  $m \longleftarrow \frac{\sum_{m=1}^{4}(k(m))}{4}$ ▷ mean value of n^{th} block pixels
  for $i \longleftarrow 1$ to 2 **do** ▷ Start for loop for thresholding
    for $j \longleftarrow 1$ to 2 **do**
      if $b_n(i,j) \geq m$ **then** ▷ Apply threshold m on each pixel and generate binary matrix M_b
        $M_b(i,j) \longleftarrow 1$
      else
        $M_b(i,j) \longleftarrow 0$
      end if
    end for
  end for ▷ Thresholding done
  $a_1 \longleftarrow \sum_{i=1}^{2}\sum_{j=1}^{2}(M_b(i,j)) \, mod 2$ ▷ Convert into a single bit
  Organize $b_n(i,j), \leq i \leq 2, 1 \leq j \leq 2$ in a matrix M of 4 rows and 5 columns
  $a(1:5) = LRC(M)$ ▷ Apply LRC function given in algorithm 4
  $a_2 \longleftarrow \sum_{i=1}^{5}(M_b(i,j)) \, mod 2$ ▷ Convert into a single bit
  $A_{b2} = a_1 \oplus a_2$
end procedure

OUTPUT : A_{b2}

Algorithm 4 Longitudinal Redundancy Check (LRC)

INPUT: Matrix $M_{m \times n}$

procedure CHECK BITS GENERATION
  $C(1:n) \longleftarrow 0$ ▷ Initialize check bits vector C of size n
  for $j \longleftarrow 1$ to n **do** ▷ n is the number of columns in matrix M
    $C(j) \longleftarrow M(1,j)$
    for $i \longleftarrow 2$ to m **do** ▷ m is the number of rows in matrix M
      $C(j) \longleftarrow a(j) \oplus M(i,j)$ ▷ Column-wise check bits calculation
    end for
  end for
end procedure

OUTPUT : Check bits vector : $C_{1 \times n}$

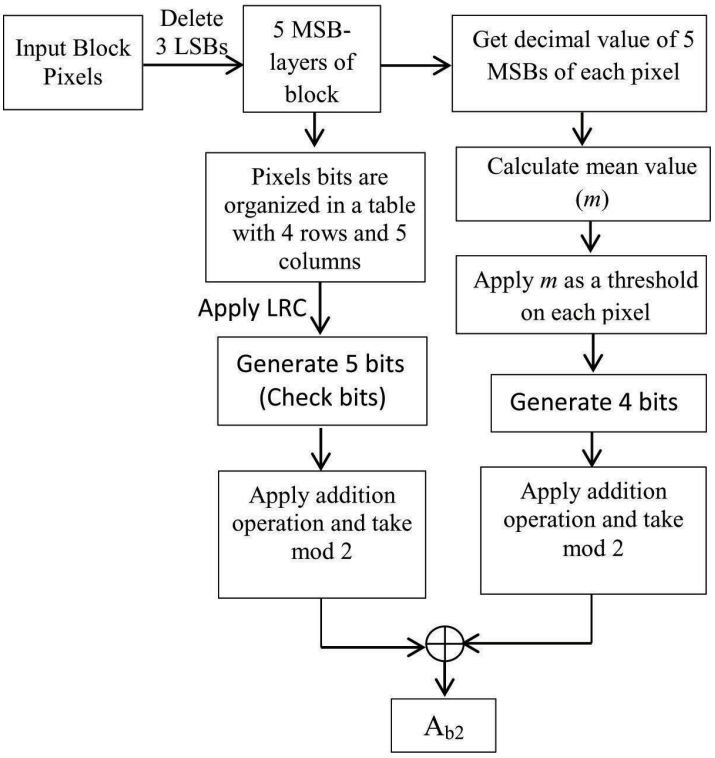


FIGURE 4.4: Block diagram of second authentication bit A_{b2} generation procedure

Algorithm 5 Block Mapping

INPUT: Block B_i index number: i $1 \leq n \leq N$, Key_3

1: **procedure** MAPPING SEQUENCE GENERATION
2: $f(i) = (Key_3 \times i) \bmod N$ ▷ Key_4 is a prime number
3: $j = f(i) + 1$
4: **end procedure**

OUTPUT : Mapped block B_j index number: j

4.3.2 Content recovery procedures

At the receiver side, it is first needed to verify the authenticity of the received watermarked image. If the received watermarked image is not authentic, then the receiver will locate the tampered blocks on the basis of the embedded watermark bits using a three level tampered blocks detection procedures. After detecting all the tampered blocks, the receiver will recover the tampered blocks using the recovery bits extracted from the corresponding mapped block of received image. So, the content recovery procedure can divide into two parts: tampered block detection procedures and tampered block recovery procedures. Fig. 4.6 illustrates the block

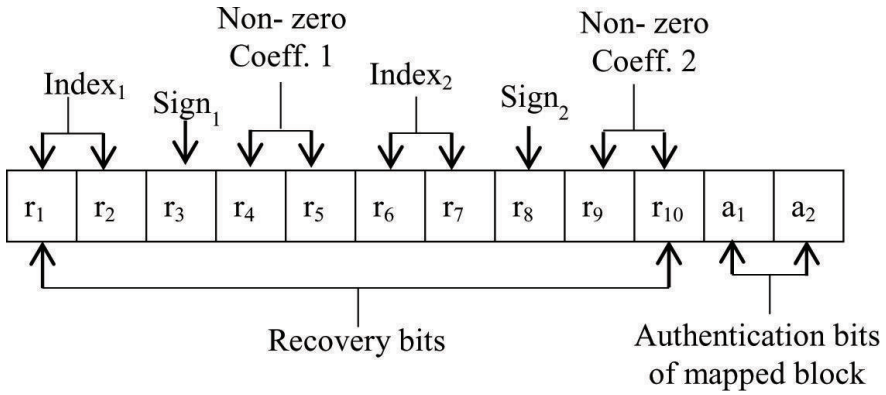


FIGURE 4.5: Block diagram of 12 bits watermark that will embed in each block of cover image to generate watermarked image

Algorithm 6 Embedding Algorithm

INPUT: Mapped Block : B_n, watermark vector: W

procedure WATERMARK EMBEDDING
  Split the W into 3 bits groups : w_1, w_2, w_3, w_4
  $c \longleftarrow 1$ ▷ Initialize counter variable
  for $i \longleftarrow 1$ to 2 **do**
    for $j \longleftarrow 1$ to 2 **do**
      $b_n(1:8) \longleftarrow dec2Bin(P_n(i,j),8)$ ▷ 8 bits binary conversion of each pixel
      $b_n(1:3) \longleftarrow w_c(1:3)$ ▷ 3 LSBs of pixel replaced by watermark bits
      $P_{wn}(i,j) \longleftarrow bin2dec(char(b_n +' 0'))$ ▷ bin2dec(), used to convert binary values into decimal
      $c \longleftarrow c+1$
    end for
  end for
end procedure

OUTPUT : Watermarked block B_{wn} pixels : $P_{wn}(i,j), 1 \leq i \leq 2, 1 \leq j \leq 2$

diagram of the content recovery procedures.

4.3.2.1 Tampered block detection procedures

The received image is first divided into non-overlapping blocks of size of 2×2 pixels. The tampered block detection procedures are described in the following two levels.

Level 1 : In this level, each block of received image is marked as "valid" or "invalid" on the basis of extracted and generated authentication bits. When generated

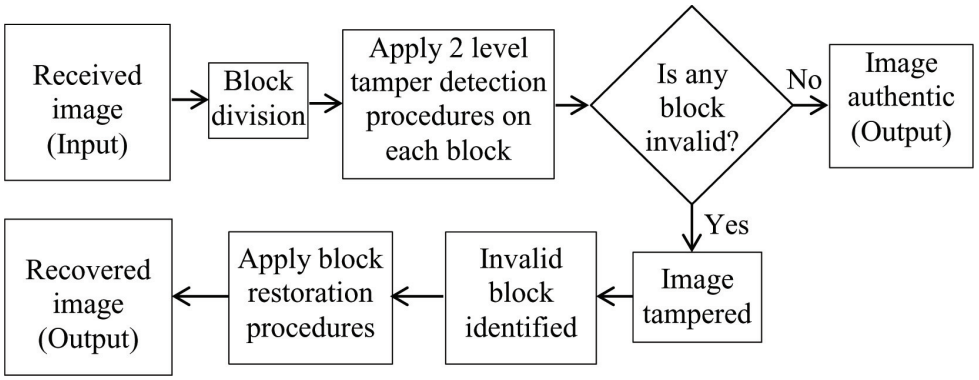


FIGURE 4.6: Block diagram of content recovery procedures of a block

authentication bits of a block are totally matched with extracted authentication bits of the block, then that block marks as "valid" block otherwise marks that block as "invalid". The pseudo code of "level 1" tampered detection is given in algorithm 7.

Level 2 : Level 2 tampered detection procedure is based on restoration bits and 3×3 neighborhood blocks. To reduce false tampered detection, if a block has marked as "valid" in *Level* 1, is needed to examine in level 2 tampered detection procedures and then, marks that block as "valid" or "invalid". The pseudo code of "level 2" tampered detection is provided in algorithm 8. So, the probability of tampered block detection is very high.

Algorithm 7 Level 1 tampered detection

INPUT: n^{th} block: B_n

1: **procedure** LEVEL 1 PROCEDURES
2: Call algorithm 2 and generate G_{a1} ▷ G_{a1} denotes the generated first authentication bits
3: Call algorithm 3 and generate G_{a2} ▷ G_{a2} denotes the generated second authentication bits
4: Extract embedded authentication bits E_{a1} and $E_{a2}$from block.
5: **if** $G_{a1} \neq E_{a1}$ **then**
6: Mark the block : "Invalid"
7: **else if** $G_{a2} \neq E_{a2}$ **then**
8: Mark the block : "Invalid"
9: **else**
10: Mark the block : "Valid"
11: **end if**
12: **end procedure**

Algorithm 8 Level 2 tampered detection

INPUT: "valid" block: B_n

procedure LEVEL 2 PROCEDURES
    Call algorithm 1 and generate a recovery vector G_{rn} of size 10 bits ▷ G_{rn} denotes the generated recovery bits of n^{th} block
    Call algorithm 5 and generate mapped block index number: m
    if Mapped block B_m is marked as "valid" **then**
        Extract embedded recovery bits E_{rn} from 3 LSB- layers of the mapped block B_m. ▷ E_{rn} denotes the extracted recovery bits of n^{th} block from m^{th} block
        for $i \longleftarrow 1$ to 10 **do**
            if $G_{rn}(i) \neq E_{rn}(i)$ **then**
                Mark the block : "Invalid"
            end if
        end for
    else
        Mark the block : "Invalid" if and only if there are five or more invalid blocks in its 3×3 neighborhood.
    end if
end procedure

4.3.2.2 Tampered block recovery procedures

After the two levels of tampered block detection procedures, all "invalid" blocks of the received image I_w are required to be recovered. All the "invalid" blocks B_n can be recovered by block recovery method. The pseudo code of block recovery method, is given in algorithm 9. In the block recovery method, a matrix M of size 2×2 is generated for each "invalid" block B_n. The five MSB-layers of the block B_n, are first filled with binary values generated from the each entry of matrix M. Next, three zeros are appended in the first three LSBs of each pixel of block B_n so as to convert the gray values in the range $\in [0, 255]$. After the completion of recovery of the block B_n, mark it as a "valid" block. These procedures are applied for all "invalid" blocks and at last apply smoothing function in tampered regions of recovered image.

Finally, the received image is recovered by this aforementioned procedure. The recovered image is very similar to the corresponding watermarked image as demonstrated by experimental observations.

Algorithm 9 Block recovery

INPUT: "Invalid" block B_n pixels of received image: $P_{wn}(i,j)$, Quantization matrix Q, Key_1
OUTPUT : Matrix M of size 2×2

procedure BLOCK RECOVERY
  Call algorithm 5 and generate mapped block index number: m
  if Mapped block B_m is marked as "valid" **then**
    Extract embedded recovery bits R_E of size 10 from 3 LSB-layers of mapped block B_m
    $Key_1 \longleftarrow mod(Key_1, 1024)$
    $K_1 \longleftarrow dec2Bin(Key_1, 10)$
    $R(1:10) \longleftarrow R_E(1:10) \oplus K_1(1:10)$ ▷ Decode the extracted recovery bits by Key_1
    $M_r(1:2, 1:2) \longleftarrow 0$ ▷ Initialize a matrix M_r of size
    $L_1(1:2) \longleftarrow R(1:2)$ ▷ Location bits of fist largest value
    $sign_1 \longleftarrow R(3)$ ▷ sign bits of fist largest value
    $V_1(1:2) \longleftarrow R(4:5)$
    $Val_1 \longleftarrow = bin2dec(char(V_1 +' 0'))$ ▷ first non largest value
    if $sign_1 \neq 0$ **then**
      $M_r(L_1) \longleftarrow -Val_1$
    else
      $M_r(L_1) \longleftarrow Val_1$
    end if
    $L_2(1:2) \longleftarrow R(6:7)$ ▷ Location bits of second largest value
    $sign_2 \longleftarrow R(8)$ ▷ sign bits of second largest value
    $V_2(1:2) \longleftarrow R(9:10)$
    $Val_2 \longleftarrow = bin2dec(char(V_2 +' 0'))$ ▷ Second largest value
    if $sign_2 \neq 0$ **then**
      $M_r(L_2) \longleftarrow -Val_2$
    else
      $M_r(L_2) \longleftarrow Val_2$
    end if
    $M_r \longleftarrow M_r * Q$ ▷ Multiplication by quantization matrix Q
    $M \longleftarrow Inv - DCT(M_r)$ ▷ M is a matrix of size 2×2
    for $i \longleftarrow 1$ to 2 **do**
      for $j \longleftarrow 1$ to 2 **do**
        $M(i,j) \longleftarrow \lceil (M(i,j) + 16) \rceil$
        if $M(i,j) < 0$ **then** ▷ M values are normalized in the range $\in [0, 31]$
          $M(i,j) \longleftarrow 0$
        else if $M(i,j) > 31$ **then**
          $M(i,j) \longleftarrow 31$
        end if
      end for
    end for
  else
    Select "valid" 8-neighborhood blocks of B_n
    Set the 3LSB-layers to zeros of selected neighborhood blocks of B_n.
    Calculate average values of those blocks in a vector A_{avg}
    $A_{mean} \longleftarrow mean(A_{avg})$ ▷ Calculate mean value of A_{avg}
    for $i \longleftarrow 1$ to 2 **do**
      for $j \longleftarrow 1$ to 2 **do**
        $M(i,j) \longleftarrow A_{mean}$
      end for
    end for
  end if
end procedure
OUTPUT : Recovery matrix M of size 2×2

4.4 Experimental results and discussions

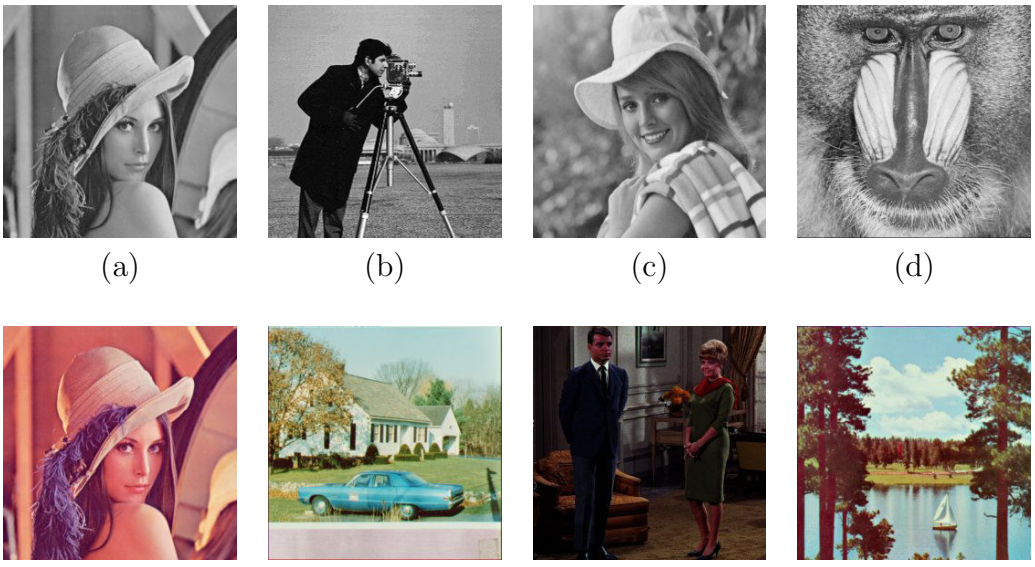
The proposed scheme used square gray scale and color images of size 256×256 as original cover image. The experiment's result of this paper was tested in the MATLAB R2013 b environment. The computing platform was a Core i7-3770 processor with a speed of 3.40 GHz and 2 GB of RAM.

Our evaluation is performed on a set of images selected from the USC-SIPI image database and some random images of our lab. Fig. 4.7 shows some of the test images which were used in the experiments and the corresponding watermarked images are shown in figure 4.8. For color image, the watermark is embedded in the each channel (i.e. R,G,B) similar to grey images. The Peak Signal to Noise Ratio (PSNR) and Normalized Cross Correlation (NCC) values of watermarked image relative to the original cover images are shown in Table 4.1. As the embedding PSNR and NCC values are very high, so it is highly difficult to differentiate between the watermarked image and the original image by human eyes. In other words, the imperceptibility of the proposed scheme is effective enough. Table 4.1 also lists the watermark embedding time in the cover image to generate a watermarked image. This time is very less for an image with a size of 256×256. This low embedding time and high PSNR and NCC values indicate that the proposed scheme is fast and effective.

TABLE 4.1: Essential information observed during watermark embedding.

Cover Image	PSNR (Embedding)	NCC (Embedding)	Average Embedding Time (in Sec.)
Lena	37.5900 dB	0.9979	6.7548
Camera Man	37.1700 dB	0.9988	6.8484
Woman	37.4900 dB	0.9979	6.7860
Baboon	37.4900 dB	0.9980	6.9264
Lena_RGB	38.0650 dB	0.9973	21.1381
House_RGB	38.2492 dB	0.9983	21.7933
Photo_RGB	39.6253 dB	0.9943	21.3705
Boat_RGB	38.5252 dB	0.9986	21.6685

Figs. 4.9 and 4.10, show the tampered area detection and recovery of gray scale image "Cameraman" and color image "Photo_RGB" respectively. Figs. 4.9(a) and 4.10(a) give the original watermarked image. These watermarked images are modified by using object addition attacks. These attacked images are shown in Figs.



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

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

FIGURE 4.7: Test images used in our experiments (a) Lena (b) Cameraman (c) Woman (d) Baboon (e) Lena_RGB (f) House_RGB (g) Photo_RGB (h) Boat_RGB.

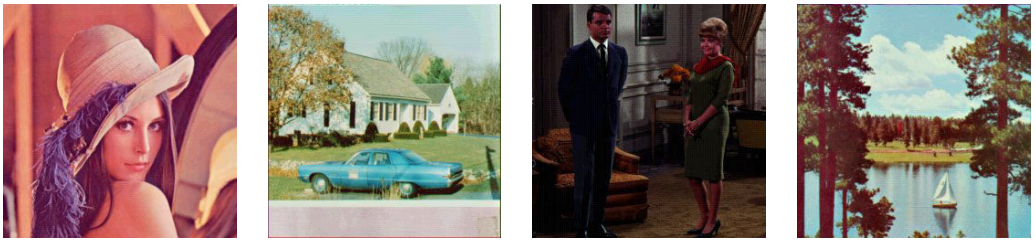
4.9(b) and 4.10(b). By using the tampered block detection procedures, all the tampered blocks were detected as shown in Figs. 4.9(c) and 4.10(c). Only the black areas are indicating authentic blocks, in the Figs. 4.9(c) and 4.10(c). The non-authentic blocks are recovered from using the tampered block recovery procedures. The recovered images are shown in Figs. 4.9(d) and 4.10(d). The PSNR values, NCC values and average recovered times of the both recovered images are listed in table 4.2.

TABLE 4.2: Essential information observed of recovered image after tampering by object addition attacks.

Cover Image	PSNR (Recovered)	NCC (Recovered)	Average recovery Time (in Sec.)
Camera Man	41.27 dB	0.9992	3.7440
Photo_RGB	42.054 dB	0.9928	12.0121

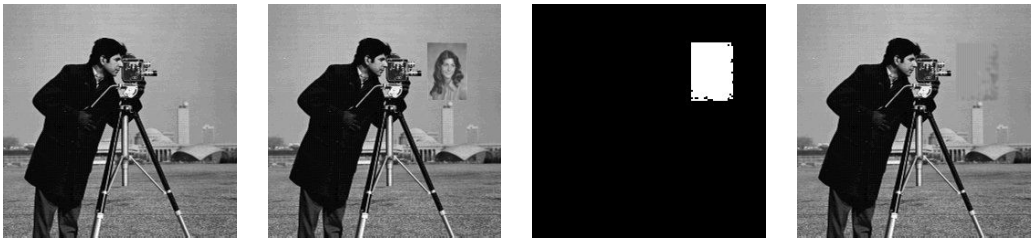
Figs. 4.11 and 4.12, show the tampered area detection and recovery of gray scale image "Cameraman" and color image "Boat_RGB" respectively. Figs. 4.11(a) and 4.12(a) give the original watermarked image. These watermarked images are modified by using object removal attacks. These attacked images are shown in Figs. 4.11(b) and 4.12(b). By using the tampered block detection procedures, all the tampered blocks were detected as shown in Figs. 4.11(c) and 4.12(c). The black

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



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

FIGURE 4.8: Watermarked image of (a) Lena (b) Cameraman (c) Woman (d) Baboon (e) Lena_RGB (f) House_RGB (g) Photo_RGB (h) Boat_RGB.

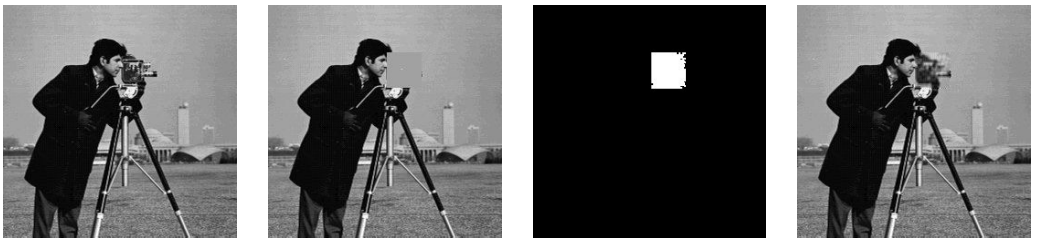


(a) (b) (c) (d)

FIGURE 4.9: (a)Watermarked image of Cameraman image (b) Object addition attack (c) tampered detected area (d) Recovered image.

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

FIGURE 4.10: (a)Watermarked image of Photo_RGB image (b) Object addition attack (c) tampered detected area (d) Recovered image.



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

FIGURE 4.11: (a)Watermarked image of Cameraman image (b) Object removal attacks (c) tampered detected area (d) Recovered image.

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

FIGURE 4.12: (a)Watermarked image of Boat_RGB image (b) Object removal attacks (c) tampered detected area (d) Recovered image.

areas in the Figs. 4.11(c) and 4.12(c), are indicating authentic blocks. The non-authentic blocks are recovered from using the tampered block recovery procedures. The recovered images are shown in Figs. 4.11(d) and 4.12(d). The PSNR values, NCC values and average recovered times of the both recovered images are listed in table 4.3.

TABLE 4.3: Essential information observed of recovered image after tampering by object removal attacks.

Cover Image	PSNR (Recovered)	NCC (Recovered)	Average recovery Time (in Sec.)
Cameraman	45.21 dB	0.9976	3.5412
Boat_RGB	45.3517 dB	0.9990	11.0137

Figs. 4.13 and 4.14, show the tampered area detection and recovery of gray scale image and color image of a "Group Photo". Figs. 4.13(a) and 4.14(a) give the original watermarked image. These watermarked images are modified by using cropping attacks at 50% areas. These attacked images are shown in Figs. 4.13(b) and 4.14(b). By using the tampered block detection procedures, all the tampered blocks were detected as shown in Figs. 4.13(c) and 4.14(c). The black areas in the Figs. 4.13(c)

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

FIGURE 4.13: (a)Watermarked image of Group Photo image (b) Cropping (50 %) attack (c) tampered detected area (d) Recovered image.

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

FIGURE 4.14: (a)Watermarked image of Group Photo_RGB image (b) Cropping (50 %) attack (c) tampered detected area (d) Recovered image.

and 4.14(c), are indicating authentic blocks. The non-authentic blocks are recovered from using the tampered block recovery procedures and corresponding recovered images are shown in Figs. 4.13(d) and 4.14(d). The PSNR values, NCC values and average recovered times of these recovered images are listed in table 4.4.

TABLE 4.4: Essential information observed of recovered image after tampering by cropping attacks at 50% areas.

Cover Image	PSNR (Recovered)	NCC (Recovered)	Average recovery Time (in Sec.)
Group Photo	30.72 dB	0.9827	5.9748
Group Photo_RGB	33.185 dB	0.9890	17.877

Fig. 4.15 shows the various degrees of cropping attacks on "Lena" image and corresponding tampered areas detection results are shown in Fig. 4.16. The recovered images of listed in Fig. 4.15, are shown in Fig. 4.17. The essential information (PSNR (dB), NCC, Recovered time (Tr in Sec.)) of recovered images, are listed in table 4.5. Similarly, cropping attacks apply on other images also and essential information of some images (Cameraman, Baboon, Woman and Group Photo) are listed in table 4.5. It is observed that the essential information like PSNR values

(a) 10% (b) 20% (c) 25% (d) 30%

(e) 35% (f) 40% (g) 45% (h) 50%

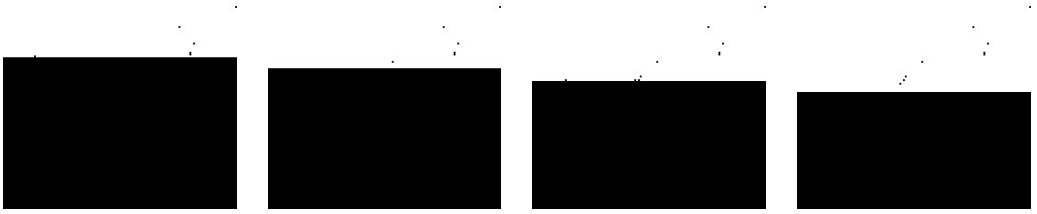
FIGURE 4.15: Tampered images at various tampering rates.

and NCC values are effectively high and Tr (in Sec.) is very less.

Fig. 4.18 shows the curve of the PSNR values of the recovered images (Lena, Cameraman, Baboon, Woman and Group Photo) in the tampered area with respect to the tampering rates from 5% to 50%. It can be seen that the curve of PSNR values with respect to tampering rate decreases smoothly but even if the tampering rate is 50%, the recovered images have PSNR values more than 30.40 dB. So, we can say that the recovered image quality is quite satisfactory.

The proposed scheme is compared with five schemes [65, 66, 67, 78, 111]. Lena image, Baboon image and Cameraman image were used to compare the recovery performance between [65, 66, 67, 78, 111] and proposed scheme. For comparative analysis, The watermarked images were tampered with various degrees and then recovered the image from the tampered images. The Figs. 4.19-4.21 show the graph of PSNR (dB) of the recovered image with respect to the different tampering ratio. These plots demonstrate the effectiveness of the proposed scheme and existing schemes. PSNR values of the recovered images in the proposed scheme are effectively higher than the method used in schemes [65, 67, 78, 111] and [66](two methods called as [66]-A and [66]-B). In the scheme [66], method [66]-A gives high PSNR values,

(a) 10% (b) 20% (c) 25% (d) 30%



(e) 35% (f) 40% (g) 45% (h) 50%

FIGURE 4.16: Tampered detected areas of image listed in figure 15.

(a) 10% (b) 20% (c) 25% (d) 30%

(e) 35% (f) 40% (g) 45% (h) 50%

FIGURE 4.17: Recovered images after tampering at various tampering rates.

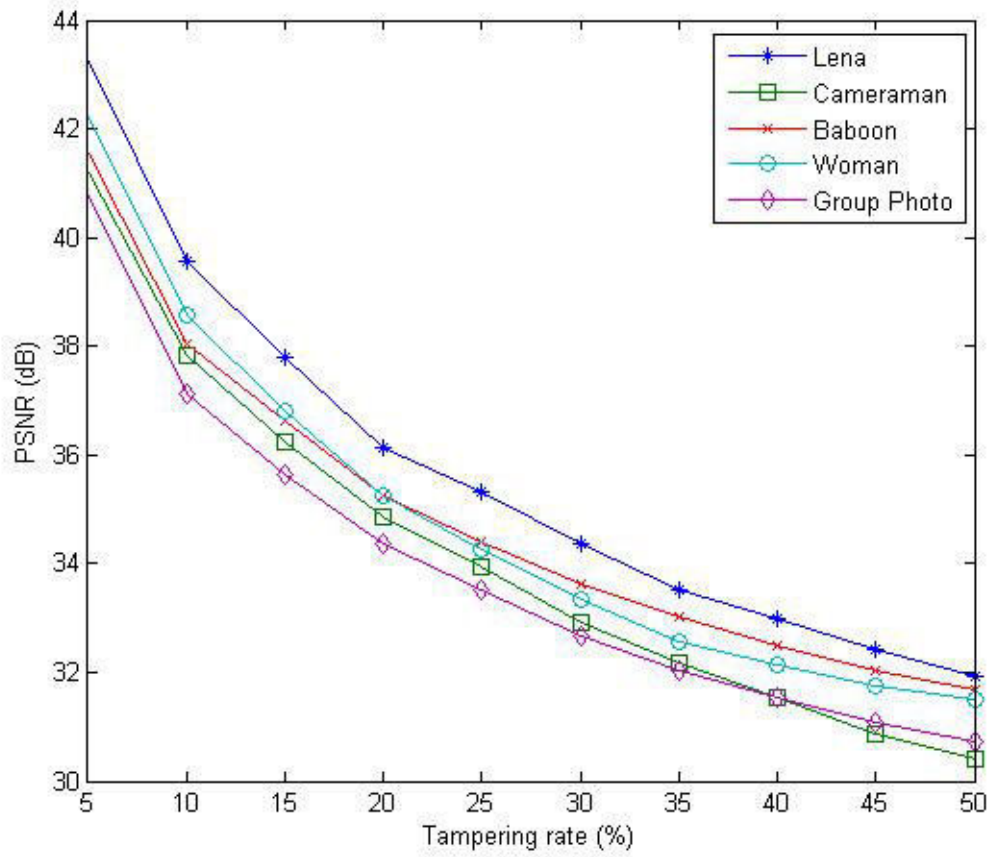


FIGURE 4.18: PSNR of restored content with respect to the tampering rates ∈[5 50].

but does not work with a large tampering rate. The maximum tampering rate for this method is only 24%. The schemes[65, 67, 78] and [66]-B is using large size of block(i.e. 8×8). If a single pixel in a block is tampered then the whole block(i.e. sixty four pixels) will be treated as tampered region. Hence, the accuracy of tampered localizations is decreased and encounters blocking artifacts. In the scheme [111], recovery data is average value of block. So, its restoration quality is not good enough. On the other side, the proposed scheme is using very small size of the blocks(i.e 2×2). Hence, this scheme effectively recovered the image and also provides high accuracy in tampered pixel localization due to use of small size of the block. The blocking artifacts are also negligible in this scheme because of using small size of block and applying the smoothing function on recovered image. Thus the proposed scheme is more flexible than previously existing schemes.

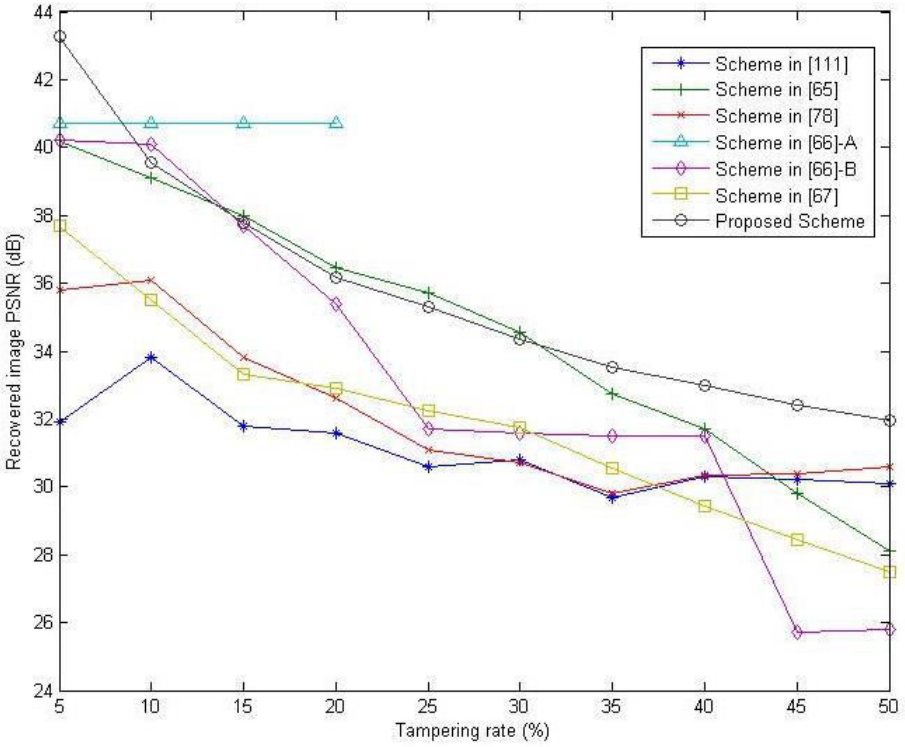


FIGURE 4.19: PSNR values comparison of content recovery with the host image Lena.

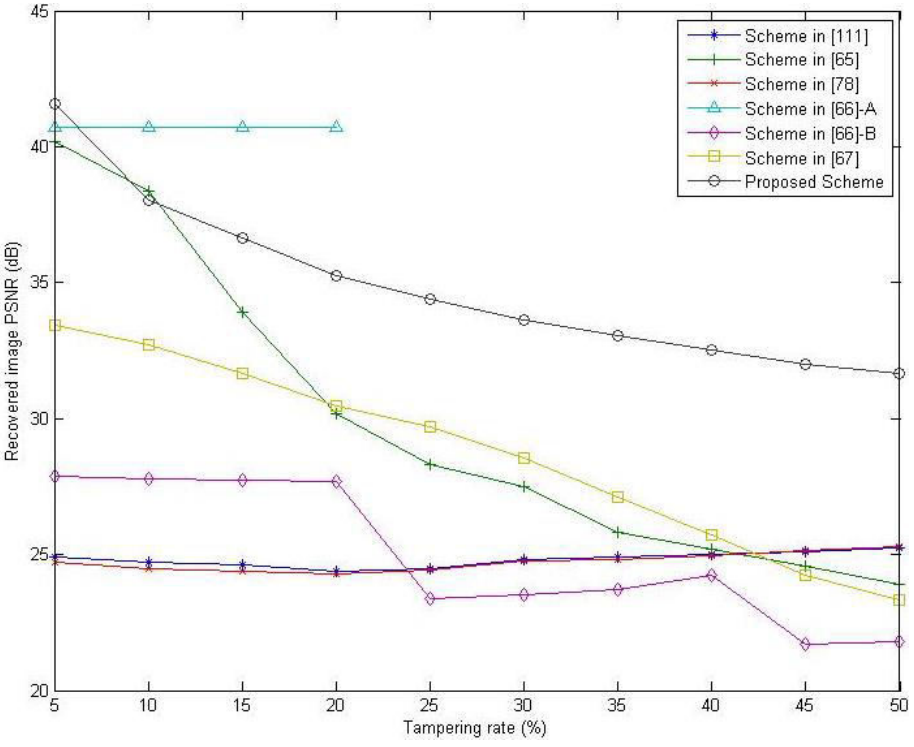


FIGURE 4.20: PSNR values comparison of content recovery with the host image Baboon.

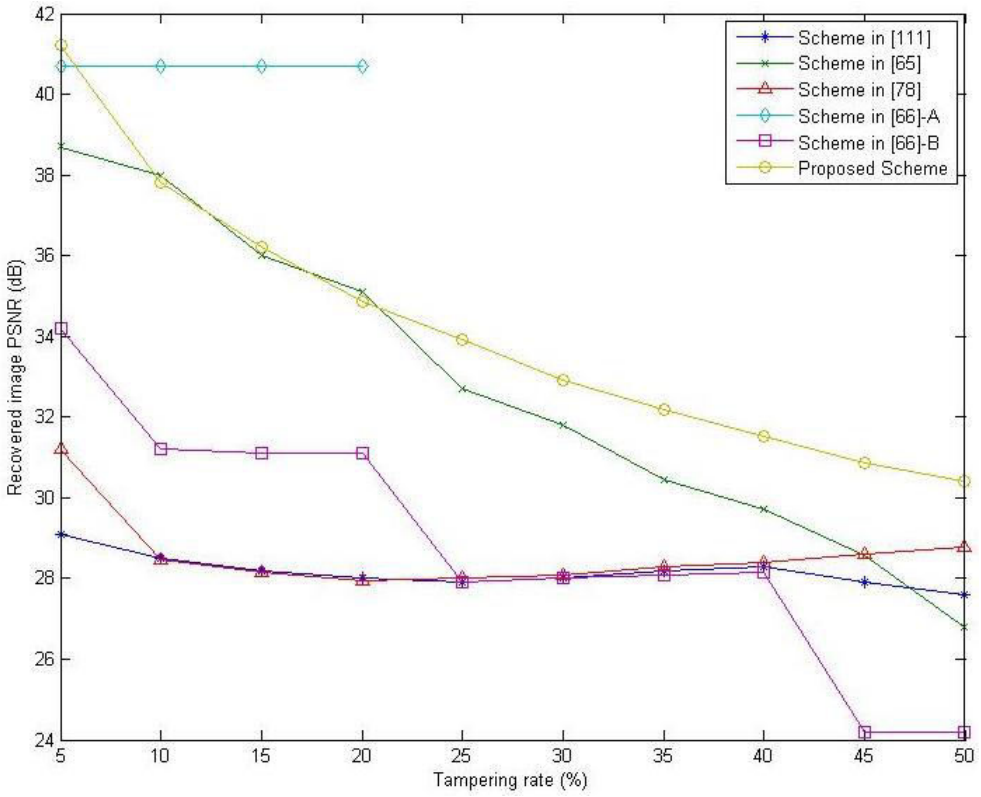


FIGURE 4.21: PSNR values comparison of content recovery with the host image Cameraman.

TABLE 4.5: Essential information (PSNR (dB), NCC, Recovered time (Tr in Sec.)) of recovered content in the tampered areas with different tampering rates.

Host Image	perfor-mence	Tampering rate							
		5%	10%	20%	30%	35%	40%	45%	50%
Lena	PSNR	43.26	39.57	36.15	34.35	33.53	33.00	32.42	31.94
	NCC	0.998	0.995	0.991	0.985	0.982	0.980	0.977	0.975
	Tr	3.682	3.947	4.680	5.054	5.273	5.616	5.834	6.115
Camer -aman	PSNR	41.21	37.82	34.86	32.90	32.18	31.54	30.87	30.41
	NCC	0.998	0.995	0.991	0.986	0.984	0.982	0.980	0.979
	Tr	4.103	4.446	5.001	5.491	5.632	5.959	6.053	6.365
Baboon	PSNR	41.59	38.05	35.26	33.61	33.03	32.51	32.02	31.67
	NC	0.995	0.989	0.980	0.973	0.970	32.510	0.963	0.960
	Tr	4.180	4.399	4.836	5.275	5.865	6.193	6.661	6.864
Woman	PSNR	42.24	38.59	35.25	33.33	32.57	32.15	31.76	31.51
	NC	0.998	0.994	0.989	0.984	0.982	0.980	0.978	0.976
	Tr	3.978	4.430	4.898	5.538	5.788	6.022	6.646	6.895
Group Photo	PSNR	40.81	37.11	34.37	32.68	32.05	31.55	31.07	30.73
	NC	0.999	0.997	0.993	0.990	0.988	0.986	0.984	0.983
	Tr	4.040	4.290	4.789	5.335	5.538	5.959	6.084	6.287

4.5 Conclusion

In this chapter, DCT based effective self-embedding fragile watermarking scheme is presented. This scheme has extreme localization and restoration capability. The accuracy of localization is extreme and blocking artifacts are negligible in this scheme because of using the small blocks of size 2×2. The 5 MSB-layers of the tampered image can still be recovered with high accuracy up to 50% tampering rate. Since the proposed scheme is a fragile watermarking scheme, the watermark information may be destroyed by the common image processing operations likes JPEG compression, contrast enhancement and filtering. Therefore, in the future a semi-fragile watermarking scheme that can simultaneously resist some common image processing operations with good restoration capability needs to be developed. The proposed watermarking scheme can also be extended for audio and video multimedia processing with improved time efficiency by parallel processing on Many-Core Processor platform [136, 137, 138].