CHAPTER 6

REVERSIBLE AUTHENTICATION WATERMARKING
WITH TAMPER LOCALIZATION

6.1 INTRODUCTION

Authentication watermark is very sensitive to any modifications imposed upon an image and can be used for tamper localization with high accuracy. In most conventional authentication techniques based on watermarking, the original image is distorted permanently due to the authentication itself. These distortions are not allowed in some sensitive applications, such as law enforcement, medical and military images. Thus, it is desired to undo the changes introduced by authentication when the image is verified as authentic. As long as the marked image is authentic, the original image can be reconstructed without any distortion. In this chapter, a new Efficient Secure Reversible Authentication Watermarking Technique with Tamper Localization (ESRAWT-TL) based on integer wavelet transform is proposed for images.

6.2 REVERSIBLE WATERMARKING APPROACHES

Several reversible watermarking schemes have been proposed in literature (Alattar 2004, Tian 2003, Lee et al 2007, Tsai et al 2005, Tseng and Hsieh 2009, Fridrich et al 2004). In general, the reversible watermarking schemes can be classified into three categories. The first category of algorithms compress a part of original image, (say LSB bit-plane) and embed
information in the available space (Fridrich 2001, Celik et al 2005, Xuan et al 2004, Feng et al 2006). The second category of algorithms find difference value between the adjacent pixels and the difference value is expanded to embed the watermark (Alattar 2004, Tian 2003). The third category of algorithms uses the concept of histogram shifting to make room for the watermark (Yang et al 2004, Tsai et al 2009, Xuan et al 2006). The three categories of algorithms are discussed in detail in the following sections.

6.2.1 Data Compression

Fridrich (2001, 2002) proposed an invertible authentication system which uses lossless JBIG compression scheme for compressing the bit-planes. The algorithm always starts with inspecting the 5th LSB-plane. Redundancy of the bit-plane is computed as given Equation (6.1).

\[
\text{Redundancy} = \text{Number of pixels} - \text{Compressed data size} \quad (6.1)
\]

When the redundancy is greater than or equal to 128, then next lower bit-plane i.e., the 4th LSB-plane is inspected, otherwise, the next higher bit-plane i.e., the 6th LSB-plane is inspected. This process is repeated until a lower bit-plane with redundancy of 128 or greater is found. This is the lowest bit-plane that provides enough space for lossless embedding of the image hash and called as the key bit-plane.

After finding the key bit-plane, it is losslessly compressed using JBIG lossless compression scheme to generate the compressed bit stream. The hash value of the original image is appended to the compressed bit stream, padding with random bits if necessary, to form the final bit-stream. Then, the key bit-plane of the original image is replaced with the encrypted bit-stream.
During retrieval, the key bit-plane is identified, decrypted and decompressed. The hash value is extracted and the original bit plane is recovered. The key bit-plane is replaced with the decrypted uncompressed bit-plane and the hash of the image is calculated. If the calculated hash matches with the extracted hash, then the image will be verified to be authentic, otherwise it will not.

6.2.2 Difference Expansion

The difference expansion method was first proposed by Tian (2002) and later improved in (Tian 2003). These schemes usually generate some small values to represent the features of the original image. Then, the generated values are expanded to embed the bits of watermark information. The watermark information is usually embedded in the LSB parts of the expanded values. Then the watermarked image is reconstructed by using the modified values. In Tian’s scheme (2002, 2003), the high frequency coefficients of a one level Haar transform is modified. If \( p_1 \) and \( p_2 \) are intensity values of a pair of adjacent pixels, then the average \( l \) and difference \( g \) of \( p_1 \) and \( p_2 \) are defined as given in Equation (6.2) and Equation (6.3) respectively.

\[
1 = \left\lfloor \frac{(p_1 + p_2)}{2} \right\rfloor \tag{6.2}
\]

\[
g = p_1 - p_2 \tag{6.3}
\]

The inverse transform is then calculated using Equation (6.4) and Equation (6.5).

\[
p'_1 = 1 + \left\lfloor \frac{(g + 1)}{2} \right\rfloor \tag{6.4}
\]

\[
p'_2 = 1 - \left\lfloor \frac{g}{2} \right\rfloor \tag{6.5}
\]
where $\lfloor . \rfloor$ denotes the least nearest integer. To embed a watermark bit $w_i \in \{0, 1\}$, the difference $g$ is represented in binary format and shifted to the left by one bit position. Then, the watermark bit $w_i$ is appended into the vacant least significant bit. Let $\text{nb}$ is the length of $g$ (i.e., $g=b_{\text{nb}-1}b_{\text{nb}-2}...b_0$), then the new difference value $g'$ can be obtained as shown in Equation (6.6).

$$g' = b_{\text{nb}-1} b_{\text{nb}-2} ... b_0 w_i + 2 \times g + w_i$$  \hspace{1cm} (6.6)

At the receiver, the watermark bit can be extracted from the LSB of the difference value, and the original difference value $g$ can be restored as in Equation (6.8) and Equation (6.9) respectively.

$$g' = p_1' - p_2'$$  \hspace{1cm} (6.7)

$$w_i = \text{LSB}(g')$$  \hspace{1cm} (6.8)

$$g = \frac{g'}{2}$$  \hspace{1cm} (6.9)

Then the original pixel values can be restored using Equation (6.10) and Equation (6.11).

$$p_1 = \left\lfloor \frac{(p_1' + p_2')}{2} \right\rfloor + \left\lfloor \frac{g + 1}{2} \right\rfloor$$  \hspace{1cm} (6.10)

$$p_2 = \left\lfloor \frac{(p_1' + p_2')}{2} \right\rfloor - \left\lfloor \frac{g}{2} \right\rfloor$$  \hspace{1cm} (6.11)

However, it is recognized that not all coefficients can be used for data embedding because the reconstructed pixels $p_1'$ and $p_2'$ after
watermarking may lie outside of the allowed range. For example, a modified 8-bit bitmap image may have values smaller than 0 (underflow) or greater than 255 (overflow). To prevent overflow or underflow, only a subset of coefficients, called expandable coefficients that do not affect the reconstructed signal are used for data embedding. The new pixel values must satisfy the Equation (6.12).

\[
0 \leq 1 + \left\lfloor \frac{g' + 1}{2} \right\rfloor \leq 255 \quad \text{and} \quad 0 \leq 1 - \left\lfloor \frac{g'}{2} \right\rfloor \leq 255 \quad (6.12)
\]

The locations of the expandable watermarked coefficients have to be recorded, compressed and embedded as header information. In order to embed the location map of the expandable coefficients, Tian (2002) proposed to substitute the least significant bit plane of the high-pass coefficients with the header information. To ensure reconstruction, the original LSBs are compressed and embedded together with the location map. Hence, the final payload consists of the watermark message bits, location map and compressed LSBs.

### 6.2.3 Histogram Shifting

Histogram shifting technique for reversible watermarking was proposed by Leest et al (2003) and Ni et al (2003). It is based on creating a gap in the histogram of an image and embeds information into this gap. Histogram of a sample image is shown in Figure 6.1(a). In order to create a gap at a coefficient which has value equals c1, all coefficients greater than c1 are added by one as shown in Figure 6.1(b). All coefficients that have values equal to c1 are used to embed a binary watermark. If the watermark bit to be embedded is 0, the coefficient will not be modified. If the watermark bit to be embedded is 1, c1 will be changed to c1 + 1. The histogram after embedding is shown in Figure 6.1(c). In order to extract the embedded bits, all
coefficients that have values equal to \( c_1 \) and \( c_1 + 1 \) are selected. Receiving a \( c_1 \) means that 0 has been embedded and \( c_1 + 1 \) mean that 1 has been embedded. The original data can be completely recovered by subtracting 1 to all coefficients greater than \( c_1 \).

![Histogram Shifting](image)

**Figure 6.1** Histogram Shifting (a) original histogram; (b) shifted histogram; (c) histogram after watermark embedding

Except Celik et al scheme (2006), none of the lossless authentication methods in the literature offers tamper localization capability, which is one of the major advantages of authentication watermarks over conventional digital signatures. In the following section, Celik et al scheme (2006) is described briefly.
6.3 LOCALIZED LOSSLESS AUTHENTICATION WATERMARK

Celik et al (2006) proposed a Localized Lossless Authentication Watermarking (L-LAW) framework. Except Celik et al scheme (2006), none of the lossless authentication methods in the literature offers tamper localization capability, which is one of the major advantages of authentication watermarks over conventional digital signatures. L-LAW is an implementation of LAW using hierarchical image authentication scheme and generalized-LSB data embedding. Figure 6.2 is a block diagram of LAW that shows the watermark embedding and verification phases. The watermark embedding phase comprises of two steps: (i) lossless pre-embedding and (ii) authentication watermarking. The verification phase of LAW is shown in Figure 6.1(b) and comprises of two steps: (i) authentication watermark verification, and if the verification step is successful, (ii) original image recovery.

![Figure 6.2: Lossless Authentication Watermarking Framework](image)

(a) lossless authentication watermark embedding;  
(b) watermark verification and recovery
The scheme uses block-based localization. Each block is partitioned into two parts, \( P_A \) and \( P_I \). In the watermarked image, part \( P_A \) carries authentication information and part \( P_I \) carries original image information. The block diagram of localized – LAW framework is shown in Figure 6.3. In the pre-embedding step of the watermark embedding phase, the LSB values in part \( P_A \) is reversibly embedded into the rest of the image \( P_I \) using Lossless Generalized – LSB (LGLSB) data embedding. Next, in the authentication watermarking step, authentication information for data in \( P_I \) is computed and placed in part \( P_A \) using hierarchical image authentication.

![Diagram](image)

**Figure 6.3** Localized LAW (a) watermark embedding; (b) watermark verification and recovery
6.3.1 Generalized Lossless LSB Data Embedding

One of the earliest data embedding methods is the LSB modification. In this well-known method, the LSB of each pixel is replaced by a payload data bit. If additional capacity is required, two or more LSBs may be overwritten. A generalization of the LSB-embedding method is employed here. In the embedding phase, the lowest levels of the pixels are replaced by the watermark payload using a quantization step followed by an addition. During extraction, the watermark payload is extracted by obtaining the quantization error or simply by reading lowest levels of the watermarked signal. Lossless generalized LSB algorithm is depicted in Figure 6.4.

![Figure 6.4 Lossless Generalized LSB Algorithm](image)

In the embedding phase, the pixel value is quantized and the residual is obtained. The compressed residual and the payload data are concatenated and embedded into the host signal via G-LSB modification.

6.3.2 Hierarchical Image Authentication

The image is divided into blocks that correspond to the elementary localization units of the hierarchical authentication watermark used in the subsequent authentication watermarking step. Minimum block size is chosen as $64 \times 64$ pixels. At each successive level of the hierarchy block size is quadrupled i.e., the image is partitioned into blocks which in turn are...
composed of 2 × 2 blocks at the preceding level of the hierarchy, (e.g., 256 × 256, 1024 × 1024, …). In each block, LSBs of the first N pixels (in the raster-scan order) are designated to carry the authentication payload.

In Figure 6.5, L = 0 denotes the lowest level of the hierarchy where the lines depict boundaries of the blocks that form the fundamental localization unit. Blocks at higher levels are composed of four blocks from the preceding level forming a quad-tree. For each block in the hierarchy, a message authentication code is computed.

![Figure 6.5 Organization of Block Hierarchy and Signatures in L-LAW](image)

Authentication information payload for a block is the concatenation of parts of Message Authentication Codes (MAC) computed at different levels of the hierarchy: all of the L = 0 MAC bits, 1/4 of the MAC bits at L = 1, and 1/16 of the MAC bits at level L = 2. Authentication payload is embedded in part $P_A$ of image bits corresponding to LSBs of shaded regions.
in Figure 6.6. Each lowest level block then carries a portion of upper level signatures, together with its independent signature. MAC is computed for each block using MD5 and the 64-bit MAC derived from MD5 is used as the authentication information. In the embedded version of the image, the LSBs carry the compressed bit stream of original LSBs as well as the payload data.

In Figure 6.6, dotted lines are pixel boundaries and thick lines form the blocks, which are tamper localization units of the authentication watermark. In each block, LSBs of shaded areas carry authentication information (forming part $P_A$). All remaining bits in the image carry image information forming part $P_I$. Unshaded areas are modified during the pre-embedding step to allow lossless recovery.

![Figure 6.6 Image Partitioning to Carry Authentication Payload](image)

**Figure 6.6 Image Partitioning to Carry Authentication Payload**

### 6.3.3 Simulation Results of L-LAW Scheme

A $1024 \times 1024$ grayscale image is watermarked using L-LAW algorithm. The watermarked image has been altered. The vehicles around the structure at the center have been digitally removed. When that image is passed through the watermark detector, the image alteration has been
detected. Furthermore, the altered region has been located, as indicated by the darkest blocks in Figure 6.7. In this figure, the shading reflects the level of confidence in the integrity of a particular block, light shading corresponding to high confidence values indicated by the hierarchical image authentication scheme.

![Figure 6.7 Results of L-LAW Scheme](image)

(a) (b)

**Figure 6.7  Results of L-LAW Scheme** (a) watermarked image; (b) tamper detection output

### 6.3.4 Limitations of L-LAW Scheme

The limitations of L-LAW scheme are listed below:

**Size of the Image:** The major drawback of L-LAW scheme is, it works fine only with larger size images, e.g. 1024 x 1024. The minimum block size is 64 x 64. The smaller sized images have only less number of such blocks and hence the scheme is not efficient for small images.

**Localization Accuracy:** Since the minimum block size is fixed as 64 x 64, the localization accuracy is also 64 x 64. This accuracy is very poor when compared with the existing localization schemes which provide a localization accuracy of 8 x 8, 12 x 12 etc. (Lin et al 2004, Wong and Memon
In addition the localization accuracy of $64 \times 64$ is not suitable for smaller images and most of the sensitive images in medical and military applications.

Imprecise Localization: The LSBs of the lowest level blocks hold a portion of upper level signatures, together with its own signature. If one block of the lowest level is modified, it will affect only the signature portion of the top level. But, the L-LAW scheme declared that the top level is also unauthentic with a different confidence level. Hence, the tamper localization is not precise.

6.4 PROPOSED ESRAWT –TL

The proposed AWT verifies the image integrity, identifies and localizes tamper detection and reconstruct the original image. If the image authenticity is verified, then the distortion due to embedding the watermark can be completely removed from the watermarked image so that the original image can be reconstructed. If the image is tampered, then the tampering positions can also be localized. The tamper localization accuracy of the proposed scheme is $5 \times 5$ and hence it is suitable for smaller images also. Further, the tamper localization is also accurate. Hence, the proposed scheme overcomes the limitations of L-LAW scheme.

The proposed ESRAWT-TL scheme uses two layers of watermarks. The first layer of watermark is embedded in spatial domain which verifies image authenticity and the second layer is embedded in transform domain which provides reversibility. Two different techniques are used for the two layers of watermark. In spatial domain, it exploits selective LSB embedding and in wavelet domain, it exploits histogram characteristics of the difference images of the wavelet coefficients. The proposed scheme modifies pixel values slightly to embed the watermark and hence does not produce much distortion to the original image.
6.4.1 Watermark Embedding

In order to verify the integrity of the image, a binary logo image is replicated to the size of the image and is embedded in selective LSB of the original image. The range of pixel value of the original image at the two extremes is narrowed down to avoid underflow/overflow problems. To obtain reversibility, the original values of the modified LSBs and the pixels that are shifted to avoid underflow/overflow problems must also be embedded as side information to the recipient as overhead data. IWT is applied on the watermark embedded image. A reversible watermarking technique using histogram modification is applied on the coefficients of high frequency sub-bands to embed the overhead data. Then inverse IWT is applied to get the final watermarked image.

6.4.1.1 Preprocessing

When the watermark is embedded in the wavelet domain underflow or overflow can occur in the spatial domain, i.e., the pixel values obtained from the watermarked wavelet coefficients can either be smaller than the minimum pixel value \( p_{\text{min}} \) (\( p_{\text{min}} = 0 \) for 8 bit grayscale image) or be greater than the maximum value \( p_{\text{max}} \) (\( p_{\text{max}} = 255 \) for 8 bit grayscale image). Since the reversibility is lost when underflow or overflow occurs, it must be predicted prior to the watermark embedding.

There are two ideas in the literature to avoid overflow/underflow problems. The first one identifies the pixels that cause overflow/underflow and ignores them during watermark embedding (Tian 2003, Alattar 2004). In this case, the payload must include information that provides the unused pixels in the watermark embedding process. The second one uses the concept of histogram shifting (Lee et al 2006). The pixels at the extreme levels are shifted towards the centre. In the proposed scheme, histogram shifting method is used to avoid overflow/underflow problems.
The range of pixel value of the original image is narrowed down before applying IWT. Let T be shifting threshold, t and t’ be pixel value before and after modification. For an 8-bit grayscale image, the modified pixel value t’ is found using Equation (6.13), if the pixel value t is between 0 to T and Equation (6.14) is used if the pixel value t is between 255−T to 255.

\[
t' = t + T, \text{if } t \in [0,T]
\]  

\[
t' = t - T, \text{if } t \in [255-T, 255]
\]

Now the range of pixel value is changed from [0,255] to [T, 255−T]. The pixels that are shifted must be recorded (book keeping data) and embedded along with watermark as an overhead data.

### 6.4.1.2 Watermark generation

The watermark to be embedded is taken as a either a logo or a random sequence of predefined bits. The advantage of using the logo is that cropping can be easily detected. The watermark is embedded in spatial domain, that is, LSB of selected pixels are modified to embed the watermark. The original image is divided into m × m sub-blocks (say, 3 × 3 or 5 × 5). The size of the sub-block defines the localization accuracy. If more accuracy is required, then the sub-block size shall be smaller; otherwise, larger sub-blocks can be used.

The watermark WM is generated by replicating the logo image L so that the size of the watermark matches with the number of sub-blocks in the original image. Figure 6.8 depicts the watermark generation process for lena image (256 × 256). The size of the sub-block is taken as 5 × 5 and hence the size of the logo is 51 × 51. The original logo and the generated watermark are shown in Figure 6.8(b) and Figure 6.8(c) respectively. To provide additional
level of security, the generated watermark is scrambled using a shared
watermark scrambling key \( wk \). Figure 6.8(d) shows the scrambled watermark.

![Figure 6.8](image)

**Figure 6.8** Watermark Generation for Lena Image using ESRAWT-TL
(a)original Image; (b)original logo; (c)generated watermark; (d) scrambled watermark

### 6.4.1.3 Layer – 1 Embedding

The original image \( F \) is divided into sub-blocks of size \( 5 \times 5 \). Let
the sub-block is represented as \( F_{r,c} \) where \( r,c \) represents the row and column of
the sub-block respectively. In each sub-block, one watermark bit is
embedded. The centre pixel of the sub-block is modified to embed the
watermark. The watermark bit is embedded by finding the sum of pixel
values of each sub-block. In Equation (6.15), \( [F_{r,c}]_{i,j} \) is the pixel value at
position \( (i, j) \) of the sub-block \( F_{r,c} \), \( R \) is the remainder in dividing the sum of
pixel values of the sub-block \( F_{r,c} \) by 2.

\[
R = \text{mod} \left( \sum_{i=1}^{m} \sum_{j=1}^{m} [F_{r,c}]_{i,j}, 2 \right) \quad (6.15)
\]

In Equation (6.16), \( w_i \) is the watermark bit, \( \oplus \) represents XOR
operation. If the remainder \( R \) and watermark bit \( w_i \) are same, then inc will be
zero and hence there is no need to make any changes in the sub-block;
otherwise, the centre pixel \( F_{r,c} \) is incremented by inc as given in Equation (6.17). The result of the embedding process is Layer-1 Watermarked Image.

\[
\text{inc} = R \oplus w_i \quad (6.16)
\]

\[
[F_{r,c}]_{a,b} = [F_{r,c}]_{a,b} + \text{inc} \quad (6.17)
\]

To obtain reversibility, the original values of the modified LSBs must also be embedded as overhead data. A 1 in the location map \( \text{lmap}(r,c) \) in Equation (6.18) reveals that the LSB of the middle pixel in the respective sub-block \( F_{r,c} \) was modified. The location map is embedded in wavelet coefficients in reversible manner.

\[
\text{lmap}(r,c) = \text{inc} \quad (6.18)
\]

### 6.4.1.4 Constructing overhead data

After embedding watermark in the LSBs of the image, the overhead data is embedded using reversible watermarking. The overhead data consists of book keeping data which provides information about the pixels shifted to avoid underflow/overflow and location map which provides information about the modified LSBs during watermark embedding process. This overhead data is embedded in wavelet domain in a reversible manner.

The book keeping data consists of three parts: shifting threshold \( T \), number of pixels shifted \( ps \) and row and column position \( r_i \) and \( c_i \) of the \( i^{th} \) pixel shifted. The three parts of the book keeping data are converted into a single bit stream as given in Equation (6.19).

\[
B = T \cup ps \cup (r_1 \cup c_1) \cup (r_2 \cup c_2) \ldots \cup (r_{ps} \cup c_{ps}) \quad (6.19)
\]

The size of location map is \( M/5 \times N/5 \). It is also represented as a single bit stream \( \text{LM} \) by reading it in column wise. The overhead data bit
stream \( O \) is formed by combining \( B \) and \( LM \). The entire embedding process is depicted in Figure 6.9.

\[
O = B \cup LM \tag{6.20}
\]

Figure 6.9 Watermark Embedding Process in ESRAWT-TL
6.4.1.5 Layer - 2 embedding

The L-1 watermarked image is decomposed by applying integer Haar wavelet transform to get LL1, LH1, HL1 and HH1 sub-bands. The overhead data is embedded in high frequency sub-bands only.

Construction of difference images: Let the size of the sub-bands are $X \times Y$. Difference image coefficients, $dD(i,j)$, $dH(i,j)$, $dV(i,j)$ of size $X \times Y/2$ are constructed for HH, HL and LH sub-bands respectively. For $1 \leq i \leq X$ and $1 \leq j \leq \frac{Y}{2}$, the difference image coefficients are constructed using Equation (6.21).

\[
\begin{align*}
    dD(i,j) &= CD(i,2j) - CD(i,2j-1) \\
    dH(i,j) &= CH(i,2j) - CH(i,2j-1) \\
    dV(i,j) &= CV(i,2j) - CV(i,2j-1)
\end{align*}
\]  

(6.21)

In Equation (6.21), $CD(i,j)$, $CH(i,j)$ and $CV(i,j)$ are the coefficients of HH, HL and LH sub-bands respectively. The odd-line coefficients $(i, 2j-1)$ are subtracted from the even-line coefficients $(i, 2j)$ to construct the difference image.

Histogram shifting: The histogram shifting technique can be applied to make room for embedding overhead data (Lee et al 2006). The histogram bins of -2 and 2 are emptied by shifting some coefficient values in the difference images. If the difference value is greater than or equal to 2, then the even-line coefficient in the respective sub-band is incremented by one. If the difference value is less than or equal to -2, then the even-line coefficient in the respective sub-band shall be decremented by one. The modified difference images $dD'$, $dH'$ and $dV'$ are computed using Equation (6.22).
\[ d'D'(i, j) = CD'(i, 2j) - CD'(i, 2j - 1) \]
\[ d'H'(i, j) = CH'(i, 2j) - CH'(i, 2j - 1) \]
\[ d'V'(i, j) = CV'(i, 2j) - CV'(i, 2j - 1) \]  

(6.22)

where,

\[
CD'(i, 2j) = \begin{cases} 
CD(i, 2j) + 1 & \text{if } dD(i, j) \geq 2 \\
CD(i, 2j) - 1 & \text{if } dD(i, j) \leq -2 \\
CD(i, 2j) & \text{otherwise}
\end{cases}
\]

\[
CH'(i, 2j) = \begin{cases} 
CH(i, 2j) + 1 & \text{if } dH(i, j) \geq 2 \\
CH(i, 2j) - 1 & \text{if } dH(i, j) \leq -2 \\
CH(i, 2j) & \text{otherwise}
\end{cases}
\]

\[
CV'(i, 2j) = \begin{cases} 
CV(i, 2j) + 1 & \text{if } dV(i, j) \geq 2 \\
CV(i, 2j) - 1 & \text{if } dV(i, j) \leq -2 \\
CV(i, 2j) & \text{otherwise}
\end{cases}
\]

Embedding process: The overhead data is embedded in sub-bands using the modified difference images. The order of difference images used is \( d'D', d'H' \) and \( d'V' \). If \( d'D' \) and \( d'H' \) embeds the entire overhead information, then \( d'V' \) is not used.

The modified difference images are scanned for embedding. Only the coefficients with the difference value of -1 and 1 are used. If such a value is encountered and if the overhead bit to be embedded is 1, then the difference value of -1 will be made to -2 by subtracting one from the even-line coefficient or the difference value of 1 will be made to 2 by adding one to the even-line coefficient. If the overhead bit is 0, the difference value of -1 or 1 will be left unchanged. Based on the difference values, only the even-line fields of the coefficients are either incremented or decremented by one.
The watermarked even-line coefficients of the three sub-bands are represented as follows: If the overhead data bit \( O(k) = 1 \) and the difference value in the difference images is either 1 or -1, then the coefficients of the sub-bands are updated as given in Equation (6.23).

\[
CD_w(i, 2j) = \begin{cases} 
CD'(i, 2j) + 1 & \text{if } dD'(i, j) = 1 \\
CD'(i, 2j) - 1 & \text{if } dD'(i, j) = -1
\end{cases}
\]

\[
CH_w(i, 2j) = \begin{cases} 
CH'(i, 2j) + 1 & \text{if } dH'(i, j) = 1 \\
CH'(i, 2j) - 1 & \text{if } dH'(i, j) = -1
\end{cases}
\]

\[
CV_w(i, 2j) = \begin{cases} 
CV'(i, 2j) + 1 & \text{if } dV'(i, j) = 1 \\
CV'(i, 2j) - 1 & \text{if } dV'(i, j) = -1
\end{cases}
\]

The coefficients of the three sub-bands are left unchanged in all other cases as shown in Equation (6.24).

\[
CD_w(i, 2j) = CD'(i, 2j) \\
CH_w(i, 2j) = CH'(i, 2j) \\
CV_w(i, 2j) = CV'(i, 2j)
\]

The odd-line fields of the watermarked sub-band coefficients are not affected in the embedding process and are given by Equation (6.25).

\[
CD_w(i, 2j - 1) = CD'(i\neq2j - 1) \\
CH_w(i, 2j - 1) = CH'(i\neq2j - 1) \\
CV_w(i, 2j - 1) = CV'(i\neq2j - 1)
\]

After the embedding process is over, inverse IWT is applied with the watermarked sub-band coefficients \( CD_w, CH_w \) and \( CV_w \) to obtain the L-2 watermarked image.
6.4.1.6 Post processing

The pixel values of the L-2 watermarked image are checked for overflow/underflow. If it does, the value of shifting threshold will be incremented by one and the entire process is repeated once again. If it doesn’t produce overflow/underflow, the watermark embedding process will be over.

6.4.2 Watermark Extraction and Verification

Figure 6.10 depicts the watermark extraction and recovery scheme. During watermark retrieval, the first step is to extract the overhead data in the watermarked image. IWT is applied on the watermarked image and the difference images of HH, HL and LH sub-bands are computed. The overhead data embedded in the coefficients are extracted and the coefficient values are restored back using reversible histogram recovery scheme. Then inverse IWT is applied to get the L-1 watermarked image.

The second step in retrieval process is extraction of watermark. The watermark bits embedded in the centre pixel of sub-blocks are extracted and the extracted watermark is descrambled using watermark scrambling key wk. The original watermark is compared with the retrieved watermark bit-wise. If both the watermarks are same, then the authenticity of the retrieved image will be verified and the image was not tampered.

The third step is based on the result of watermark verification. If the image is authentic, then the watermarked image will be recovered back to its original form using the overhead data retrieved. Otherwise, tamper localization process will be applied to identify the tampering positions. The main advantage of the proposed scheme is that the original image recovery process is carried out after the verification of authenticity whereas most of the existing lossless watermarking schemes perform this step prior to authenticity verification (Fridrich 2002, Ni et al 2003).
Figure 6.10 Watermark Extraction and Verification Process in ESRAWT-TL
6.4.2.1 Extraction of overhead data

The difference images of the three high frequency sub-bands are found. The difference images are scanned in the same order as at the sender side. The overhead data is extracted using Equation (6.26).

\[
O_r(k) = \begin{cases} 
0 & \text{if } dD_w(i, j) = 1 \text{ or } -1 \\
1 & \text{if } dD_w(i, j) = 2 \text{ or } -2 
\end{cases}
\]

\[
O_r(k) = \begin{cases} 
0 & \text{if } dH_w(i, j) = 1 \text{ or } -1 \\
1 & \text{if } dH_w(i, j) = 2 \text{ or } -2 
\end{cases}
\]

\[
O_r(k) = \begin{cases} 
0 & \text{if } dV_w(i, j) = 1 \text{ or } -1 \\
1 & \text{if } dV_w(i, j) = 2 \text{ or } -2 
\end{cases}
\]

(6.26)

where \(O_r(k)\) is the overhead data retrieved, \(dD_w, dH_w\) and \(dV_w\) are the difference images of HH, LH, HL sub-bands of watermarked image respectively. It is not necessary to scan all the three sub-band coefficients. The size of the overhead data is also sent as a header in the payload. The two components of overhead data are separated as book keeping data and location map.

All the three difference images are scanned once again to recover the histogram shifting carried out during watermark embedding process. Since, only the even-line coefficients are manipulated, the odd-line coefficients of recovered image are directly obtained from the watermarked coefficients using Equation (6.27).

\[
CD_r(i, 2j - 1) \quad CD_w(i, 2j - 1) \\
CH_r(i, 2j - 1) \quad CH_w(i, 2j - 1) \\
CV_r(i, 2j -1) \quad CV_w(i, 2j - 1) 
\]

(6.27)
In Equation (6.27), $CD_w$, $CH_w$ and $CV_w$ are the coefficients of HH, LH, HL sub-bands of watermarked image and $CD_r$, $CH_r$ and $CV_r$ are the coefficients of HH, LH, HL sub-bands of recovered image. The even-line coefficients of the recovered image can be expressed as in Equation (6.28). The difference images of sub-bands are used to recover the original coefficients.

$$CD_r(i, 2j) = \begin{cases} 
CD_w(i, 2j) - 1 & \text{if } dD_w(i, j) \geq 2 \\
CD_w(i, 2j) + 1 & \text{if } dD_w(i, j) \leq -2 \\
CD_w(i, 2j) & \text{otherwise}
\end{cases}$$

$$CH_r(i, 2j) = \begin{cases} 
CH_w(i, 2j) - 1 & \text{if } dH_w(i, j) \geq 2 \\
CH_w(i, 2j) + 1 & \text{if } dH_w(i, j) \leq -2 \\
CH_w(i, 2j) & \text{otherwise}
\end{cases}$$

$$CV_r(i, 2j) = \begin{cases} 
CV_w(i, 2j) - 1 & \text{if } dV_w(i, j) \geq 2 \\
CV_w(i, 2j) + 1 & \text{if } dV_w(i, j) \leq -2 \\
CV_w(i, 2j) & \text{otherwise}
\end{cases}$$

The inverse IWT is applied after the coefficients of recovered image are constructed. It is clear that, the resultant image is the L-1 watermarked image which contains only the watermark and the distortions due to overhead data embedding are removed. The next step is to verify the authenticity of the image.

### 6.4.2.2 Image authentication

The watermark is embedded in LSBs of the centre pixel of $5 \times 5$ sub-blocks. It is retrieved using Equation (6.29). The remainder $R$ in dividing the sum of pixel values of each watermarked sub-block $F_{r,c}$ by 2 is computed. The remainder represents the watermark bit retrieved $w_{r,c}$ where $r$ and $c$ represents the row and column of the watermark respectively.
\[ R = \text{mod} \left( \sum_{i=1}^{m} \sum_{j=1}^{m} \left[ F_{r.c,i,j} \right] 2 \right) \quad (6.29) \]

\[ w_{r.c} = R \quad (6.30) \]

Since, the watermark is scrambled during embedding process, the retrieved watermark is descrambled using the same shared secret watermark scrambling key. Then, the retrieved watermark is compared with the original watermark. If there are no distortions in the retrieved watermark, then the authenticity will be verified; otherwise, it is concluded that the image was tampered. Once the image authenticity is verified, the next step is to remove the distortions of watermark embedding and hence the recovery of the original image. If image authenticity is not verified, there will be no need of recovery of original image; instead tamper localization will be performed to identify the tamper positions.

### 6.4.2.3 Recovery of original image

The overhead data retrieved at the earlier stage is used for recovering the original image. Location map provides information about the modified LSBs during watermark embedding process. A 1 in the location map lmap(i,j) identifies that the LSB of centre pixel in \((i,j)^{th}\) sub-block was modified. It is used to recover the original values of LSBs. Book keeping data provides information about the pixels that are shifted to avoid underflow/overflow condition. The book keeping data is in a single bit stream format which is preceded by shifting threshold T and the number of pixels modified. Consecutive 16 bits are taken at a time in which the first 8 bits identify the row and the next 8 bits identify the column of the pixel shifted for a 8 bit grayscale image.
Let \( T \) be shifting threshold, \( p_w \) and \( p_r \) be the pixel values of watermarked and the recovered original image. For a 8-bit grayscale image, the pixel values of the recovered image is computed using Equations (6.31) and (6.32).

\[
p_r = p_w - T \text{, if } p_w \in [0, T] \quad (6.31)
\]

\[
p_r = p_w + T \text{, if } p_w \in [255 - T, 255] \quad (6.32)
\]

Now the range of pixel value is changed from \([T, 255-T]\) to \([0,255]\) and hence the original image is successfully recovered without any distortions.

### 6.4.2.4 Tamper localization

If authentication test is failed, it is clear that the watermarked image was modified and tamper localization will be applied to identify the tampering positions. Watermark is generated by replicating the watermark logo and scrambled using the shared watermark scrambling key \( w_k \). It is compared with the watermark retrieved. If there is a mismatch, then the respective 5 × 5 sub-block in the watermarked image was modified.

The following example illustrates the tamper localization clearly. Figure 6.11(a) and Figure 6.11(b) are the watermarked image and modified watermarked image respectively. The retrieved watermark is shown in Figure 6.11(c). It contains salt and pepper noise from which it is certain that the image was tampered during transmission. The noise is extracted from the retrieved watermark and inverse scrambling is applied to locate the tamper positions in the original image. In Figure 6.11(d), the white blocks depict the tamper positions.
Figure 6.11 Tamper Localization using ESRAWT-TL (a) watermarked image; (b) tampered watermarked image; (c) watermark retrieved; (d) image with tamper localization

6.4.3 Simulation Results

ERSAWT-TL scheme is tested with images of various size and the simulation results are illustrated in Table 6.1. The number of tamper localization blocks identified in L-LAW scheme and ERSAWT-TL scheme are presented. Since the localization block size for L-LAW scheme is $64 \times 64$, the images have only very few localization blocks. ERSAWT-TL scheme provides localization accuracy at $5 \times 5$ block size and hence the number of localization blocks in images also very large compared with L-LAW scheme. Hence ERSAWT-TL scheme localizes very small tampers also. Table 6.1 shows that the scheme produces watermarked images with higher PSNR values. The results of images are shown in Appendix 2.
Table 6.1 Results of ESRAWT-TL Scheme

<table>
<thead>
<tr>
<th>Image</th>
<th>Image Size in pixels</th>
<th>No. of Tamper Localization Blocks</th>
<th>PSNR in db</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L-LAW</td>
<td>ERSAWT-TL</td>
<td></td>
</tr>
<tr>
<td>Lena</td>
<td>256 × 256</td>
<td>16</td>
<td>2601</td>
<td>51.4143</td>
</tr>
<tr>
<td>Baboon</td>
<td>116 × 116</td>
<td>1</td>
<td>529</td>
<td>49.5756</td>
</tr>
<tr>
<td>Peppers</td>
<td>136 × 137</td>
<td>4</td>
<td>729</td>
<td>50.3320</td>
</tr>
<tr>
<td>Barbara</td>
<td>130 × 130</td>
<td>4</td>
<td>676</td>
<td>50.4714</td>
</tr>
<tr>
<td>Camera</td>
<td>256 × 256</td>
<td>16</td>
<td>2601</td>
<td>51.3475</td>
</tr>
<tr>
<td>Angel</td>
<td>98 × 130</td>
<td>2</td>
<td>494</td>
<td>51.9675</td>
</tr>
<tr>
<td>Bridge</td>
<td>131 × 90</td>
<td>2</td>
<td>468</td>
<td>50.4271</td>
</tr>
<tr>
<td>Dragon</td>
<td>135 × 101</td>
<td>2</td>
<td>540</td>
<td>52.2837</td>
</tr>
</tbody>
</table>

6.5 SUMMARY

A wavelet-based reversible watermarking scheme for secure image authentication has been presented. In the proposed scheme, the embedded watermark is generated and scrambled based on the size of the image to be watermarked. This provides more protection to the watermarking system. Integer wavelet transform is applied and the proposed watermarking system provides excellent tamper localization accuracy. At the same time, if the image is deemed to be authentic, the original image can also be restored. The proposed scheme is tested with several images and the results exhibit the efficiency of the scheme.