# Reversible Watermarking based Security

**Major Project submitted in partial fulfillment of the requirements for the award of degree of**

**Master of Technology**

**in**

**Information Systems**

Submitted By:

**Himanshu Agarwal**

**(2K11/ISY/08)**

Under the Guidance of:

**Mr. Manoj Kumar**

**(Associate Professor)**

**(Department of Computer Engineering)**

**Department of Information Technology**

**Delhi Technological University**

**(2011-2013)**

# CERTIFICATE

This is to certify that **Himanshu Agarwal (2K11/ISY/08)** has carried out the major project titled **"Reversible Watermarking based Security"** as a partial requirement for the award of Master of Technology degree in Information Systems by **Delhi Technological University**.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2011-2013**. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

**Mr. Manoj Kumar**

Associate Professor

Department of Computer Engineering

Delhi Technological University

Bawana Road, Delhi-110042

# Abstract

Reversible watermarking technique is used to transfer some small information through the medium of images, and both the image and the data could be recovered at the receiving end. It's widely used in medical imaging to transfer the patient information corresponding to his/her medical image for the purpose of diagnostic. Here we propose a technique based on difference expansion, in which the transferred image consist of the original image embedded with the payload consisting of patient info, hash of original image and original bits required for the recovery. In order to provide confidentiality, the payload is encrypted using symmetric key based encryption algorithm, this provides that the patient information is confidential and the image too could not be recovered for diagnoses under attack. Hash of the recovered image is calculated and compared with the recovered embedded hash to make sure that the image has not been tampered. This provides for the confidentiality, authentication and the integrity of the transferred image. We have also extended the concept of reversible watermarking earlier confined only to images and audio data, to other forms of data formats as well. It could be widely used in copyright protection and could be effectively used to distribute .docx, .xls, .ppt, .pdf, .exe and many other file formats securely over the network. The approach is based on sliding window approach in which the embedding space is not the whole available space of the document or the file but a small window is selected out of the available space.

*Keywords:* Reversible Contrast Mapping, Difference Expansion, Reversible Watermarking, Medical Imaging, Image Authentication and Confidentiality

# ACKNOWLEDGEMENT

I express my gratitude to my major project guide **Mr. Manoj Kumar, Associate Professor, Department of Computer Engineering** for the valuable support and guidance he provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members and my friends for providing their valuable help and time whenever it was required.

Himanshu Agarwal

Roll No. 2K11/ISY/08

M.Tech (Information Systems)

E-mail: himanshuagarwal1989@gmail.com

# Contents

## List of Figures

## List of Tables

# Chapter 1: Introduction

For the purpose of copyright protection the content available on the web do contains watermark embedded into it. The watermark sometimes is completely visible as in case of watermarked pdf documents. The images do contains watermarks that may or may not be visible. But the problem is that of the degradation of the image because of the embedded watermark. These watermarked images could not be used for military or medical imaging. So, for these purposes we use reversible watermarking technique that could allow the recipient to extract both the watermark and original image independently provided the embedding algorithm and other embedding parameters are available to the user. The security issues that are concerned with images used for military purposes and medical imaging are:

- Confidentiality: The patient information should not be disclosed while the data image is transferred through the medium.
- Authentication: The received image should be authenticated, being received from the desired source only.
- Integrity: It should make sure that the received image has not been tampered in between the release and the reception.

In order to fulfill all these necessities a number of reversible watermarking schemes have been proposed and they need to satisfy the following requirements:

- Blind: The recovery process should not require the original image.

- Imperceptibility: The quality of the watermarked image should not be seriously degraded from the original image.
- Higher embedding capacity: The process should provide a high embedding capacity so that, if required there should be minimal requirement for the compression of the payload.

Based on these requirements a number of reversible watermarking techniques have been proposed. The concept appeared for the first time in an authentication method in a patent owned by The Eastman Kodak [1].

### 1.1 Motivation

In conventional watermarking schemes, there was a cover image and the watermark information. In those times the available algorithms gave no importance to the cover image but only the hidden watermark was important. So, by the recipient only the watermark was extracted safely but the image was never recovered. That was an unnecessary overhead. The improvement came with reversible watermarking scheme with a lot of restrictions such as high embedding capacity, imperceptibility, robustness and many more. And its use in many crucial applications such as military and medical imaging made it an emerging area for research.

### 1.2 Research Objective

Reversible watermarking technique was confined to images and audio, video data only. The objective of this work is to present an algorithm that could provide high embedding capacity for embedding the patient information in the image for the purpose of diagnostic and also extend the concept of reversible watermarking for copyright protection of documents and files such as .iso, .pptx, .msi, .cab, .msp, .mp3, .xls, .docx, .exe etc. The algorithm will be explained in the forthcoming chapters with the experimental results.

### 1.3 Scope of work

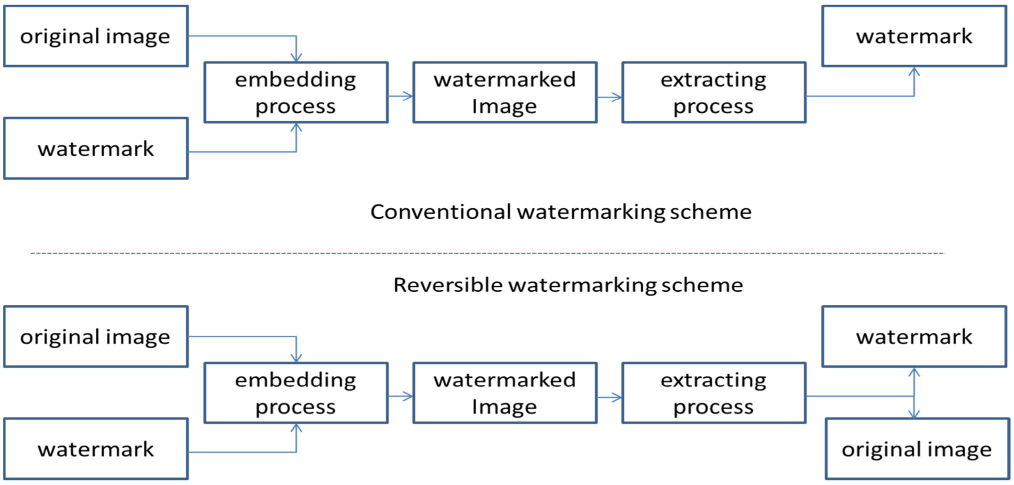
The application of the algorithm to files and documents of different types is still a problem. The technique is based on sliding window selection which requires the user to manually select the embedding space from the available embedding area. The manual process is tedious and it is very difficult to find the maximal possible embedding space from the available area. In future it is possible to find an algorithm that could easily extract the window size based on the type of the file or the document to be watermarked.

### 1.4 Organization of thesis

In this chapter, I have highlighted the concept of reversible watermarking, motivation to do this thesis, my objective, and scope to do the work in same field. Chapter 2 provides a detailed picture of reversible watermarking and the prior work done till date. In chapter 3 I have presented the proposed scheme. Chapter 4 includes the implementation details and experimental results. Finally chapter 5 concludes the thesis.

# Chapter 2: Literature Review

Reversible watermarking is a special type of digital watermarking in which from the watermarked image both the watermark and the original image is extracted without any loss. Figure 2.1 presents a difference between the conventional and the reversible watermarking scheme.



(Figure 2.1: Flowchart of Reversible and Conventional Watermarking scheme)

Reversible watermarking in accordance with the conventional watermarking should satisfy some additional requirement also that is, the process should be blind, the extracting process should be independent of the original image.

There are a number of reversible watermarking techniques in the spatial domain, they are based on the transformation of pixel intensities.

They are as follows:

- Schemes applying data compression
- Schemes based on difference expansion
- Schemes using histogram bin shifting

## 2.1 Schemes applying data compression

In order to recover the original image from the watermarked image some recovery information should also be embedded with the watermark. This increases the size of the total payload to be embedded. In case where the scheme does not provide high embedding capacity the payload is compressed before being embedded in the original image. A well known data compression scheme proposed by Celik et al [5] is presented as:

1. The image pixels are quantified using the following L-Level scalar quantization and remainders are generated: $Q(x) = L \times \left\lfloor \frac{x}{L} \right\rfloor$. Let the original image be

| | | | |
|---|---|---|---|
| 20 | 37 | 7 | 22 |
| 35 | 12 | 32 | 13 |
| 22 | 12 | 18 | 23 |
| 12 | 23 | 12 | 26 |

and the watermark be $\{1000101011\}_2$ and the parameter $L = 5$. The quantified image:

| | | | |
|---|---|---|---|
| 20 | 35 | 5 | 20 |
| 35 | 10 | 30 | 10 |
| 20 | 10 | 15 | 20 |
| 10 | 20 | 10 | 25 |

and the remainders are:

| | | | |
|---|---|---|---|
| 0 | 2 | 2 | 2 |
| 0 | 2 | 2 | 3 |
| 2 | 2 | 3 | 3 |
| 2 | 3 | 2 | 1 |

2. The remainders are then compressed using Calic lossless compression and forms:

| | | | |
|---|---|---|---|
| $x_0$ | $x_1$ | $x_2$ | $x_3$ |
| $x_4$ | $x_5$ | $x_6$ | $x_7$ |
| $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ |

3. The watermark is then converted into $L$-ary format and concatenated with the remainders in the above step. Here $L = 5$. It forms:

| | | | |
|---|---|---|---|
| $x_0$ | $x_1$ | $x_2$ | $x_3$ |
| $x_4$ | $x_5$ | $x_6$ | $x_7$ |
| $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ |
| 4 | 2 | 1 | 0 |

4. The watermarked image is then generated by adding the watermark in their respective cells to the quantified image. The watermarked image formed is:

| | | | |
|---|---|---|---|
| $20+x_0$ | $35+x_1$ | $5+x_2$ | $20+x_3$ |
| $35+x_4$ | $10+x_5$ | $30+x_6$ | $10+x_7$ |
| $20+x_8$ | $10+x_9$ | $15+x_{10}$ | $20+x_{11}$ |
| 10+4 | 20+2 | 10+1 | 25+0 |

In the retrieving phase step-1 of the embedding process will be applied again, the remainders will be generated. The last 4 remainders will represent the watermark and remaining remainders will be decompressed to original 16 remainders and the original image will be regenerated without any loss.

## 2.2 Schemes based on difference expansion

Schemes based on difference expansion embed 1 bit of information per pair of pixels, selected from the image. The pairs of pixels are selected in a particular order, it may be row wise, column wise or any other fixed order. There is a well known difference expansion scheme represented by Tian [2]. Here is a brief description:

For a pair of pixels $(x, y)$, the integer average $l$ and difference $h$ is calculated as:

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor \text{ and } h = x - y \tag{1}$$

The inverse transformation of (1) is defined as:

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor \text{ and } y = l - \left\lfloor \frac{h}{2} \right\rfloor \tag{2}$$

**Embedding Process:**

1. Calculate $l_i$ and $h_i$ for $i^{\text{th}}$ pair of pixels $(x_i, y_i)$ by using equation (1).
2. Calculate $h_i' = 2 \times h_i + w_i$, where $w_i$ is the $i^{\text{th}}$ watermark bit.
3. Using equation (2), substituting $x$ by $x'$, $y$ by $y'$ and $h$ by $h$’. Calculate $x'$ and $y'$.
4. Repeat steps 1-3 for all pixel pairs.
5. Watermarked image can be formed by replacing $(x, y)$ pairs with their corresponding $(x', y')$ pairs.

**Restoration Process:**

1. Using equation (1), substituting $l$ by $l'$, $h$ by $h'$, $x$ by $x'$ and $y$ by $y'$. Calculate $l_i'$ and $h_i'$ for $i^{\text{th}}$ pair of pixel $(x_i', y_i')$.
2. Calculate $w_i$ by extracting the LSB of $h'$, and $h = \left\lfloor \frac{h'}{2} \right\rfloor$.
3. Calculate the original $(x, y)$ pair using equation (2).
4. Repeat steps 1-3 for all pixel pairs.
5. Original image can be formed by replacing $(x', y')$ pairs with their corresponding $(x, y)$ pairs.

In this scheme all the pixel pairs can’t be used for data embedding because pixels are bound to $[0, 255]$. Thus only those pairs which remain within the limits after transformation could be used for embedding. To distinguish some additional information is also embedded with the watermark bit depicting that the particular pair was used for embedding or not.

### 2.3 Schemes using histogram bin shifting

A scheme based on histogram bin shifting utilizes the histogram denoting the frequency of the grayscale pixel values for data hiding. In 2006 Ni et al [10] proposed a histogram bin shifting based data hiding technique. Brief description includes:

**Embedding Process**

The pixel intensity value occurring most frequently is determined from the histogram of the grayscale image and called as peak value. All the pixel values greater than the peak value are shifted one bin to the right, thus the bin next to the bin of the peak value becomes empty. To each pixel value with the peak grayscale value the watermark bit is added. When the watermark bit is “1”, the watermarked pixel will occupy the bin just emptied and in case of watermark bit “0”, there is no modification.

### Extraction and Restoration Process

The watermarked image is scanned in the same sequential order as in the embedding process. Whenever a pixel with the previous peak grayscale value is encountered, that means the watermark bit embedded in that pixel was “0”. If a pixel value with 1 more than the peak value is encountered that means that the watermark bit embedded in that pixel is “1” and the pixel is modified by subtracting 1. Finally all pixel values greater than the peak value are subtracted by 1.

The data embedding capacity of this approach is based on the number of pixels having peak value.

## 2.4 Tian’s data embedding using Difference Expansion

Tian [2] in 2003 proposed a reversible data embedding scheme based on difference expansion. By that time it was considered as one the best scheme providing high embedding capacity with low distortion and low complexity. The detailed explanation of the Tian’s scheme is as follows:

### Reversible Transformation

Forward Transform – For a pixel pair $(x, y)$, where $x$, $y$ are consecutive pixel values of the image. We have $0 \le x, y \le 255$, the forward transform is defined as:

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor \text{ and } h = x - y \tag{3}$$

Inverse Transformation – The inverse transform to calculate pair $(x, y)$ is given by:

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor \text{ and } y = l - \left\lfloor \frac{h}{2} \right\rfloor \qquad (4)$$

To prevent overflow and underflow in equation (4) we have, $0 \le x, y \le 255$. This is equivalent to have $|h| \le 2(255 - l)$, and $|h| \le 2l + 1$ (5)

## Difference Value

Difference value $h$ defined by the equation (3), is used to embed a bit $b$ by difference expansion given by $h' = 2 \times h + b$ (6)

To prevent overflow and underflow $|h'| \le \min(2(255 - l), 2l + 1)$ (7)

$h$ is expandable if and only if $h'$ satisfies equation (7) for both $b = 0$ and $b = 1$. Thus expandable difference can be used for data embedding.

In case of reverse transformation value $h$ is generated back from $h'$ by using $h = \left\lfloor \frac{h'}{2} \right\rfloor$. This means that the last bit of $h'$ is modified from 0 to 1 or 1 to 0. Thus it can be said that $h$ is changeable if and only if $\left| 2 \times \left\lfloor \frac{h}{2} \right\rfloor + b \right| \le \min(2(255 - l), 2l + 1)$ for both $b = 0$ and $b = 1$.

## Embedding Process

From the pair of pixels, difference values are generated and then some expandable difference values are used for data embedding. For complete decoding at the other end, it is mandatory to know which pairs have been used for data embedding. So it is required to embed location information of the embedded pixel pairs along with the watermark. For this purpose a location map is created which contains location information of all selected expandable difference values. For a changeable difference value $h'$, at decoding end it is not possible to find whether $h$ was expandable or not. So for changeable and non expandable difference values, the last bit of the pixel pair could be modified saving its original bit. To guarantee an exact recovery of the original image the original bits should also be embedded in the payload. The data embedding algorithm is divided into six steps:

1. Original image is grouped into pair of pixels, pairs could be selected with pixels adjacent to each other either horizontally or vertically or there may be a key based pair selection. To each pair an integer transform defined by equation (3) is applied. The difference values calculated corresponding to the pixel pairs are placed in one dimensional list $\{h_1, h_2, h_3, ..., h_n\}$.
2. The list obtained from step-1 is then grouped into four different categories:
    i. EZ: It contains all expandable $h = 0$ and $h = -1$.
    ii. EN: It contains all expandable $h \notin$ EZ.
    iii. CN: It contains all changeable $h \notin$ (EZ $\cup$ EN).
    iv. NC: It contains all non-changeable $h$.

    The set of changeable difference values are EZ $\cup$ EN $\cup$ CN.
3. All expandable difference values are changeable. Thus for $h \in$ EZ are the primary difference values for difference expansion. Difference values $h \in$ EN can also be used for difference expansion depending on the size of the payload to be embedded. Thus EN can be divided into EN1 and EN2 with EN1 $\subseteq$ EN for pairs used for difference expansion. The location map is then created with the same size as that of the number of pixel pairs. For $h \in$ (EZ $\cup$ EN1), assign a value 1 in the location map and for $h \in$ (EN2 $\cup$ CN $\cup$ NC) assign 0. The location map is then compressed using run length compression algorithm and is represented by $L$.

4. The original LSB values corresponding to $h \in$ (EN2 $\cup$ CN) except for $h = 1$ and $h = -2$ are collected in a bitstream *C*.
5. The the total payload comprising of the concatenation of *L*, *C*, *P*, where *P* is the payload comprising of watermark, authentication hash etc. is converted into a bitstream *B*.

   $B = L \cup C \cup P = b_1 b_2 b_3 \ldots b_m$, where $b_i \in \{0,1\}$ and $1 \le i \le m$ ($m$ is bit length).

   Embedding will follow as:

```
i)    Set $i = 1$ and $j = 0$
ii)   while ($i \le m$)
          $j = j + 1$
          if $h_j \in$ (EZ $\cup$ EN1)
          $h_j = 2 \times h_j + b_i$
          $i = i + 1$
          else if $h_j \in$ (EN2 $\cup$ CN)
          $h_j = 2 \times \left\lfloor \frac{h_j}{2} \right\rfloor + b_i$
          $i = i + 1$
iii)  end
```

6. After embedding all $m$ bits of *B*, apply inverse transformation defined by equation (4) to obtain the watermarked image.

### Decoding and Restoration Process

In the decoding process the difference values are calculated and grouped into changeable and non-changeable, and the embedded bitstream *B* can be recovered from LSB's of these changeable difference values. From *B*, the location map *L* and the original LSB's *C* will be recovered. For expanded difference values a division by 2 will give back their original values and other changeable difference are recovered from their original LSB's obtained from bitstream *C*.

The process consists of the following five steps:

1. The watermarked image is grouped into pairs of pixels according to the same procedure as in step-1 of the embedding process. For each pair transformation using equation (3) is applied and the difference values are ordered in one dimensional list.
2. The difference values obtained are then grouped into two disjoint sets:
   i. CH: It contains all changeable *h*.
   ii. NC: It contains all non-changeable *h*.
3. Collect LSB's of all difference values in set CH, and form a binary stream $B=b_1b_2\ldots b_m$.
4. Extract the compressed location map *L* from *B* and decompress it using run length decoding algorithm. Restore the original difference values as:

```
i)    Set $i$=1
ii)   for $j$=1 to n
      if $h_j \in$ CH
            if location map value at $h_j = 1$ then $h_j = \left\lfloor \frac{h_j}{2} \right\rfloor$
            else
                  if $0 \le h_j \le 1$ then $h_j = 1$
                  else if $-2 \le h_j \le -1$ then $h_j = -2$
                  else $h_j = 2 \times \left\lfloor \frac{h_j}{2} \right\rfloor + b_i$ and $i$=$i$+1
iii)  end
```

If the location map value is “1”, it means that the difference value was expanded during embedding. If $h \in$ CH, $0 \leq h \leq 1$ and location map value is “0” then original value of $h = 1$. Similarly if $h \in$ CH, $-2 \leq h \leq -1$ and location map value is “0” then original value of $h = -2$. For other changeable difference values where location map value is “0”, restore their original LSB’s from bitstream *C*.

5. After all the original difference values have been recovered, recover the original image by using reverse transformation defined by equation (4). Calculate the hash of the recovered image and compare it with the hash obtained from *P*. If they match, the image content is authentic and recovered image is same as that of the original image.

The scheme provides an embedding capacity of 0.5bpp. The size of the location map is half of the number of pixels of the image. This means that if the location map is not compressed then there would be no space to embed watermark information. In case of compression also sometimes compression is low, in that case data embedding using this scheme would not be possible.

### 2.5 Tseng and Hsieh’s Prediction based difference expansion

Tseng et al [11] scheme is an extension to the difference expansion scheme and does not require a location map. Thus it provides a higher embedding capacity compared with the Tian’s scheme.

### Embedding Process

The embedding process is a four step process:

1. An original image denoted by *I* represents a pixel value *I(x,y)* at position *(x,y)*. Except for the first row and first column of the image *I*, the whole image is scanned in raster scan order and each predictive pixel at *(x,y)* is calculated.

Predictive pixel *I*(x,y)* = $\left\lfloor \dfrac{I(x,y-1)+I(x-1,y)}{2} \right\rfloor$ (8)

2. Then for all pixels except for first row and first column, difference value $d$ is calculated.

$$d = \left| I(x,y) - I^*(x,y) \right| \quad (9)$$

3. $d$ is then classified into four cases and the pixel values are modified accordingly:

case 1: if $\left\lceil \frac{T}{2} \right\rceil \le d < T$ , where $T$ is a predefined variable used to control image distortion. Then secret bit b is embedded.

$$I'(x,y) = \begin{Bmatrix} I^*(x,y) + 2\times d + b, if\,(I(x,y) > I^*(x,y)), \\ I^*(x,y) - 2\times d - b, otherwise. \end{Bmatrix}$$

case 2: if $T \le d \le T + \left\lfloor \frac{T}{2} \right\rfloor$, then no secret bit is embedded.

$$I'(x,y) = \begin{Bmatrix} I(x,y) - \left\lfloor \dfrac{T}{2} \right\rfloor, if\,((I(x,y) > I^*(x,y)), \\ I(x,y) + \left\lfloor \dfrac{T}{2} \right\rfloor, otherwise. \end{Bmatrix}$$

case 3: if $d \ge T + \left\lfloor \frac{T}{2} \right\rfloor$, then also no secret bit is embedded.

$$I'(x,y) = \begin{Bmatrix} I(x,y) + \left\lceil \dfrac{T}{2} \right\rceil, if\,((I(x,y) > I^*(x,y)), \\ I(x,y) - \left\lceil \dfrac{T}{2} \right\rceil, otherwise. \end{Bmatrix}$$

case 4: if $0 \le d < \left\lceil \frac{T}{2} \right\rceil$, then there is no embedding.

$$I'(x,y) = I(x,y)$$

4. The watermarked image is formed by replacing $I(x,y)$ with $I'(x,y)$ for all pixel values except for row one and column one.

### Extraction and Restoration Process

The extraction process resembles embedding process and is a three step process:

1. Watermarked image denoted by $I'$ represents a pixel value $I'(x,y)$ at position *(x,y)*. Except for the first row and first column of the image $I'$, the whole image is scanned in raster scan order and each predictive pixel at *(x,y)* is calculated.

   Predictive pixel *I'*(x,y)* = $\left\lfloor \frac{I'(x,y-1)+I'(x-1,y)}{2} \right\rfloor$ (10)

   Then for all pixels except for first row and first column, difference value *d'* is calculated.

   $$d' = \left| I'(x,y) - I'^{*}(x,y) \right| \tag{11}$$

2. $d'$ is then classified into four cases and the pixel values are modified accordingly:

   case 1: if $T \le d' < 2T$, then the secret bit b is extracted.

   $b = d' - 2\times\left\lfloor \frac{d'}{2} \right\rfloor$ and

   $$I(x,y) = \begin{cases} I'^{*}(x,y) + \left\lfloor \frac{d'}{2} \right\rfloor, if\,((I'(x,y) > I'^{*}(x,y)), \\ I'^{*}(x,y) + \left\lfloor \frac{d'}{2} \right\rfloor, otherwise. \end{cases}$$

   case 2: if $\left\lceil \frac{T}{2} \right\rceil \le d' < T$, then

   $$I(x,y) = \begin{cases} I'(x,y) + \left\lfloor \frac{T}{2} \right\rfloor, if\,((I'(x,y) > I'^{*}(x,y)), \\ I'(x,y) - \left\lfloor \frac{T}{2} \right\rfloor, otherwise. \end{cases}$$

   case 3: if $d' > 2T$, then

$$I(x,y)=\left\{\begin{array}{l} I'(x,y)-\left\lceil \frac{T}{2}\right\rceil, if\,((I'(x,y)>I'^{*}(x,y)), \\ I'(x,y)+\left\lceil \frac{T}{2}\right\rceil, otherwise. \end{array}\right\}$$

case 4: if $0 \le d' < \left\lceil \frac{T}{2} \right\rceil$, then

$$I(x,y)=I'(x,y)$$

3. The original image is formed by replacing $I'(x,y)$ with $I(x,y)$ for all pixel values except for row one and column one.

Lee et al [3] proposed an improvement to the above scheme that provided higher embedding capacity and better visual quality of the watermarked image.

## 2.6 Lee et al’s Prediction based Difference Expansion Scheme

This scheme provided higher embedding capacity by shrinking the boundary pixel to a limit so that the overflow and underflow conditions could be avoided and those pixels could also be used for data embedding. The symbol *T* is a predefined variable used to control image distortion.

### Embedding Process

The embedding process is a four step process:

1. Except for the first row and first column, predictive pixel for an image I at *(x,y)* is calculated using equation (8).
2. For all these pixels difference *d* between predictive pixel value and original value is calculated using equation (9).
3. *d* is classified into two cases and pixel value is modified accordingly:

   case 1: if $d \le T$, then the secret bit *b* is embedded

$$I'(x,y)=\begin{cases} I*(x,y)+2\times d+b, if\,(I*(x,y)\le I(x,y)), \\ I*(x,y)-2\times d-b, otherwise. \end{cases}$$

case 2: if $d > T$ , no secret data embedded

$$I'(x,y)=\begin{cases} I(x,y)+\delta, if\,(I*(x,y)\le I(x,y)), \\ I(x,y)-\delta, otherwise. \end{cases}, \text{ where } \delta = T+1$$

4. Except for the first row and first column, all the pixels are replaced by the modified pixel values computed in the above step to form a watermarked image.

**Extraction and Restoration Process**

1. Except for the first row and first column calculate the predictive pixel for an image $I'$ at *(x,y)* using equation (10).
2. Then the difference $d'$ between the predictive pixel value and original value is calculated using equation (11).
3. $d'$ is then classified into two cases, watermark bit *b* is recovered and the pixel values are modified accordingly:

case 1: if $d' \le 2\times T+1$, the secret bit b is extracted.

$$b = d'\%2$$

$$I(x,y)=\begin{cases} I'*(x,y)+\left\lfloor \frac{d'}{2} \right\rfloor, if\,(I'*(x,y)\le I'(x,y)), \\ I'*(x,y)-\left\lfloor \frac{d'}{2} \right\rfloor, otherwise. \end{cases}$$

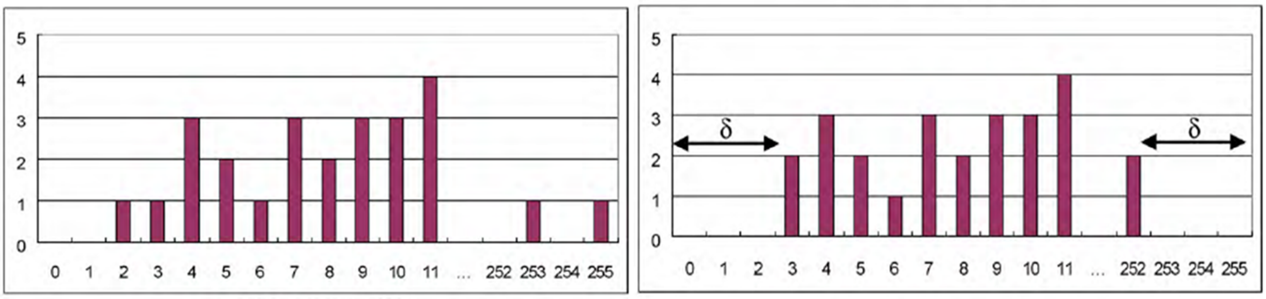
case 2: if $d' > 2\times T+1$, then

$$I(x,y)=\begin{cases} I'(x,y)-\delta, if\,(I'*(x,y)\le I'(x,y)), \\ I'(x,y)+\delta, otherwise. \end{cases}$$

4. Original image is recovered by replacing all pixel values of the watermarked image by the recovered values in the above step (except for the first row and the first column).

**Histogram Modification**

The boundary value pixels are subject to overflow and underflow problems when secret data is embedded. These pixels could not be used for data embedding. To make these pixels available for data embedding, as shown in Figure 2.2 shrinkage is made at both the ends of the histogram by a value of $\delta$. The original values of the modified pixels are to be stored for an exact recovery at the restoration phase. For this purpose n-bit string where $n = \lceil \log_2 \delta + 1 \rceil$ is used to indicate the original value of the pixels lying in the range $[0, \delta]$ or $[255-\delta, 255]$.



(Figure 2.2: Boundary Pixels modification)

For example suppose $T = 2$, then $\delta = 3$ and $n = \lceil \log_2 3 + 1 \rceil = 2$. In this case the values will be represented as shown in Table 2.1.

| Value | Representation | Value | Representation |
|---|---|---|---|
| 0 | 00 | 252 | 00 |
| 1 | 01 | 253 | 01 |
| 2 | 10 | 254 | 10 |
| 3 | 11 | 255 | 11 |

(Table 2.1: Boundary Pixels with their representation)

This information is concatenated with the secret information and thus the total payload formed is embedded in the original image to form the watermarked image.

## 2.7 Coltuc, et al’s Reversible Contrast Mapping Scheme

Coltuc et al’s [4] is a spatial domain reversible watermarking scheme that achieves a high embedding capacity without the requirement of data compression. The scheme is based on reversible contrast mapping that is, integer transformation defined on the pairs of pixels. The scheme provides LSB’s of the pixels as the embedding space and the transformed pixels are perfectly invertible even if their LSB’s are lost.

### Reversible Contrast Mapping

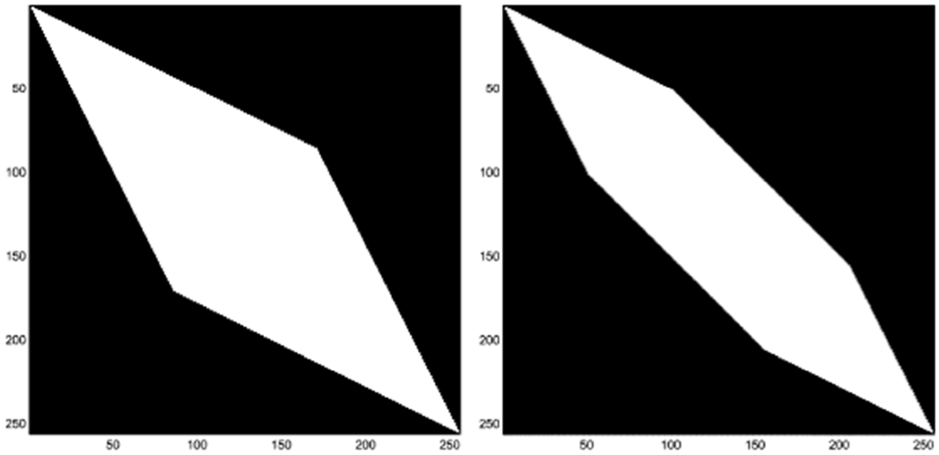
Let $(x, y)$ be the initial pair of pixels, $(x', y')$ be the transformed pair and $[0, L]$ be the range of pixel intensity.

The **forward RCM** transforms the pair $(x, y)$ into $(x', y')$ as follows:

$$x' = 2x - y \text{ and } y' = 2y - x \tag{12}$$

To prevent overflow and underflow the transformation is restricted to a subdomain $D$, where $D \subset [0, L] \times [0, L]$ defined by the equation:

$$0 \le x' \le L \text{ and } 0 \le y' \le L \tag{13}$$



(Figure 2.3 [4]: Domain *D* with and without Control Distortion)

As shown in Figure 2.3 the white colour represent the domain $D$

The **inverse RCM transform** is defined as follows:

$$x = \left\lceil \frac{2}{3}x' + \frac{1}{3}y' \right\rceil \text{ and } y = \left\lceil \frac{2}{3}x' + \frac{1}{3}y' \right\rceil \qquad (14)$$

The reverse transformation fails to recover the odd value pairs $(x', y')$ if the LSB's of both *x'* and *y'* are changed. From (1) it follows that $(x', y')$ is an odd value pair only if $(x, y)$ is an odd value pair. To conclude, on $D$ without the set of odd pairs the inverse RCM is perfectly invertible even if the LSB's of the transformed pair of pixels are changed.

**Algorithm**

The watermark substitutes the LSB's of the transformed pair $(x', y')$. In order to extract both the watermark and the pair $(x, y)$, the transformed pairs should be correctly identified. The LSB of *x* is used to indicate if the pair was transformed or not and *y* is used to insert the watermark bit.

The inverse RCM fails to recover odd value pairs but they could also be used for watermark embedding if they are correctly identified while extraction. Not all odd value pairs could be used for embedding, the pairs subject to ambiguity are found by solving in odd numbers the equation $2x-y=1, 2y-x=1, 2x-y=L$, $2y-x=L$. For $L=255$, there are 170 such pairs. As shown in Fig , $D_c$ be the domain of the transform without the ambiguous odd pixel pairs.

**Embedding Process**

1. Partition the entire image into pair of pixels (on rows, on columns or any space filling curve).
2. For each pair $(x, y)$:
   a) If $(x, y) \in D_c$ and if it is not composed of odd pixel values, transform the pair using the equation (1), set the LSB of *x'* to “1,” and consider the LSB of *y'* as available for data embedding.
   b) If $(x, y) \in D_c$ and if it is composed of odd pixel values, set the LSB of *x* to “0,” and consider the LSB of *y* as available for data embedding.
   c) If $(x, y) \notin D_c$, set the LSB of *x* to “0”, and save the true value.
3. Mark the image by simply overwriting the bits identified in 2(a) and 2(b) with the bits of the watermark (payload and bits saved in 2(c)).

For using the domain $D_c$, use a bit matrix of $L \times L$ where the value “1” indicates the accepted pixel pair and the value “0” indicates rejected pair.

**Extraction and Restoration process**

Watermark extraction and exact recovery of the original image is performed as follows:

1. Partition the entire image into pairs of pixels.

2. For each pair $(x', y')$:

    a) If the LSB of *x'* is"1", extract the LSB of *y'* and store it into detected watermark sequence, set the LSB of *x'*, *y'* to "0", and recover the original pair $(x, y)$ by inverse transform (3).

    b) If the LSB of *x'* is "0" and the pair $(x', y')$ with the LSBs set to "1"belongs to $D_{c,}$ extract the LSB of *y'*, store it into detected watermark sequence and restore the original pair as $(x', y')$ with the LSBs set to "1".

    c) If the LSB of *x'* is '0" and the pair $(x', y')$ with the LSBs set to "1" does not belong to $D_c$, the original pair $(x, y)$ is recovered by replacing the LSB of *x'* with the corresponding true value extracted from the watermark sequence.

Based on these watermarking techniques a lot of work has been done for the secure transfer of medical images. Miaou et al [6] proposed a LSB technique where the host image authenticated the transmission with an embedded message composed of various patient data, the diagnosis report and the doctor's identity. Huang et al [7] proposed a DCT based watermarking scheme for privacy protection and authentication of the transferred image with the aid of the associated patient data.

Memon et al [8] proposed their scheme with a different approach which embeds a robust watermark (electronic patient record) in the region of non interest and fragile watermark in the region of interest. Kundu et al [9] proposed a scheme in which the encrypted patient information and the hash of the ROI is concatenated and inserted using lossless compression and spatial domain watermarking process.

# Chapter 3: Proposed Approach

The proposed approach is based on Coltuc et al's [4] algorithm for data embedding and comprises of two sections:

3.1 Copyright protection of medical images – For secure distribution of patient information along with his or her medical image for the purpose of diagnosis.

3.2 Copyright protection of other data formats – For copyright protection of files or documents other than images.

## 3.1 Copyright protection of medical images

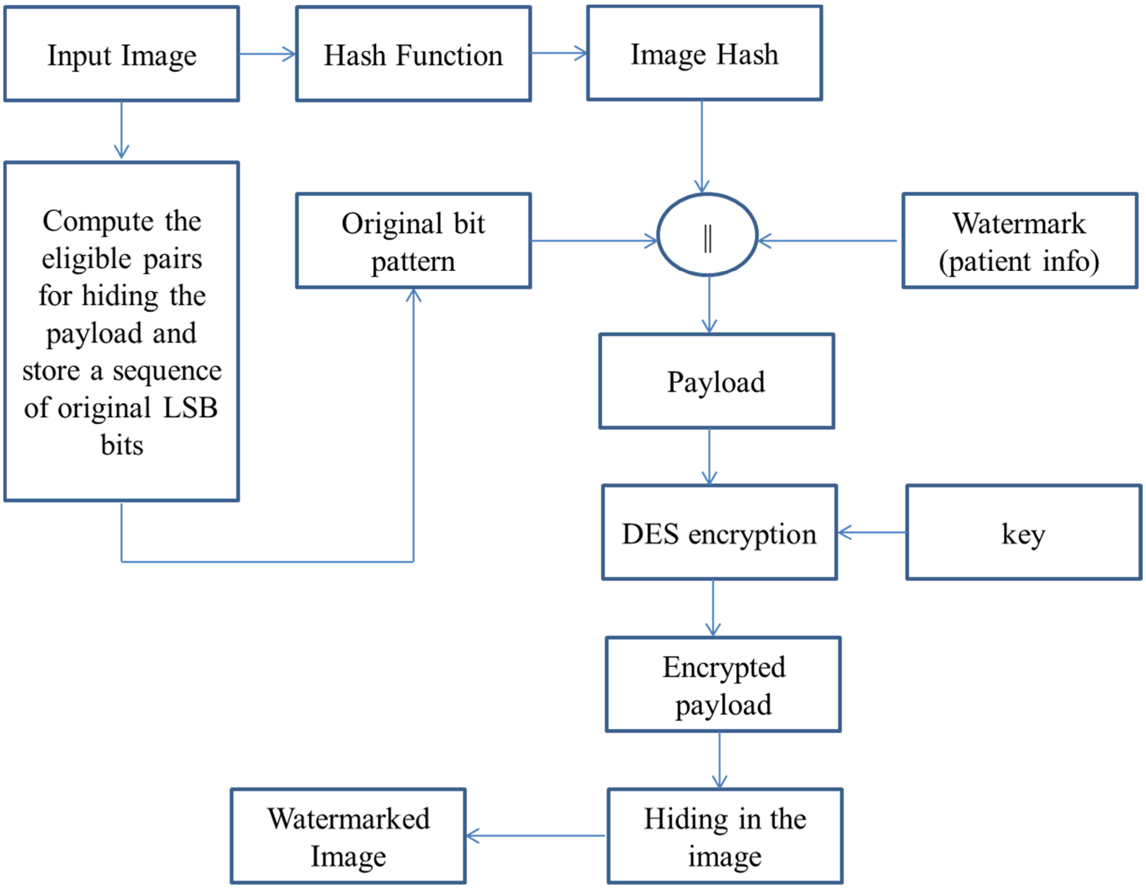
The payload being inserted in the image consist of the concatenation of the hash value of the input image, the patient information and the true LSB values saved for exact recovery. The payload is encrypted using a key (provided by the user) based DES algorithm before embedding.

In case where the numbers of pixel pairs available for inserting the watermark are in large numbers as compared to the length of the payload. In that case the payload is repeated several times as per requirement.

### Embedding Process

The embedding process is as shown in Figure 3.1. The hash of the input image is calculated as image hash and the image hash, the patient information, and the original bit pattern (calculated by applying Coltuc et al [4] scheme) are concatenated to form the payload. The payload is furthur encrypted by applying the DES algorithm and the encrypted payload thus obatined is embedded in the input image to form the watermarked image.

The DES algorithm is used in the feedback mode in which a 64 bit key is initially provided by the user, that key serves only the first block of 64 bits of the payload to be encypted. For the next block of payload we obtain a new key by xoring the initial key with previously encrypted block of 64 bits. This happens for every subsequent block of 64 bits of payload to be encrypted. The advantage of using such an approach is that if a single bit of information is being tampered then there will surely be a change in the recovered watermark from the original watermark and the hash of the recovered image, irrespective of the location of tampering.

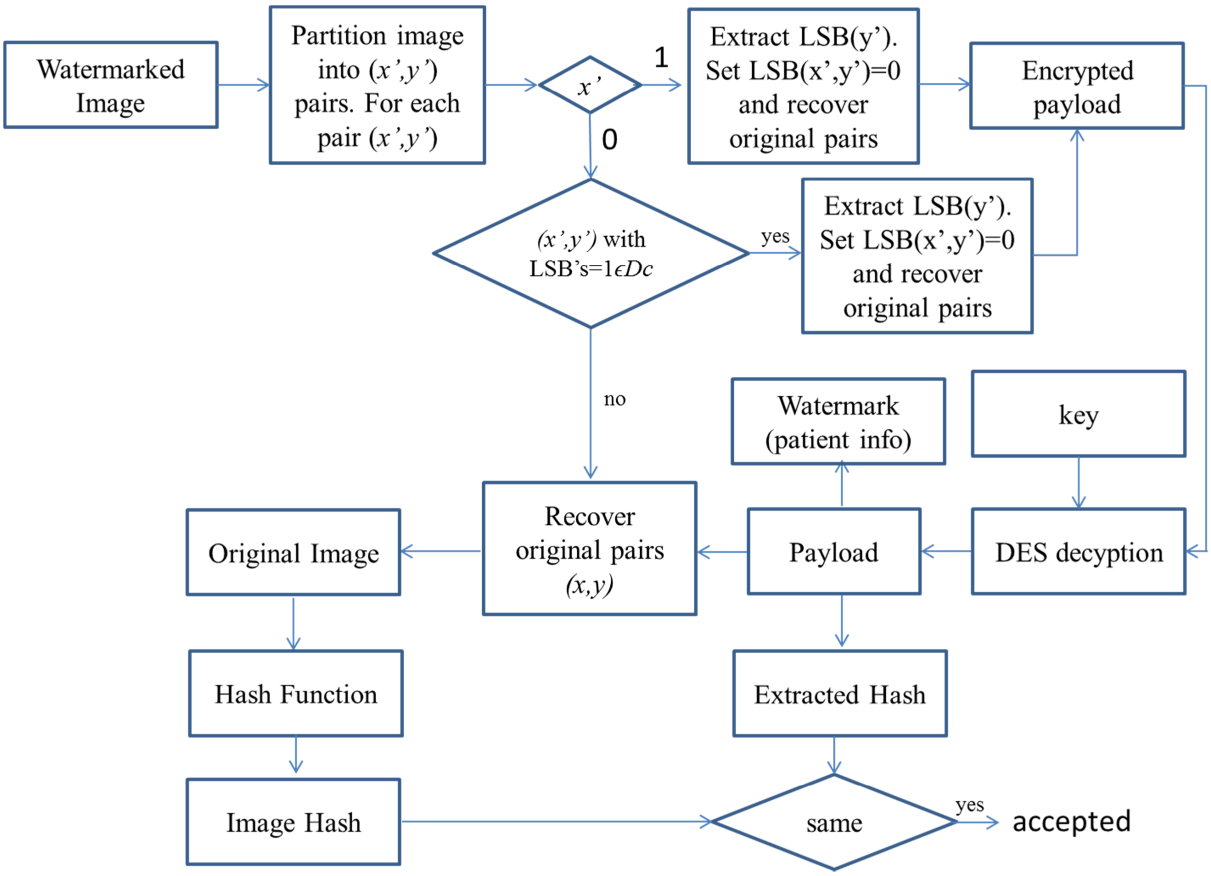


(Figure 3.1: Embedding Process)

**Restoration Process**

In the restoration process we apply the extraction process of Coltuc et al [4] scheme and extract the encrypted payload. Then the encrypted payload is being decrypted by using DES decryption and the same initial key provided again. The watermark and the hash are extracted from the payload and the original image is recovered by using the original LSB bits of the remaining payload. The hash of the recovered image is calculated and compared with the extracted hash. If they both match then the recovered image is accepted else the image has been tampered.

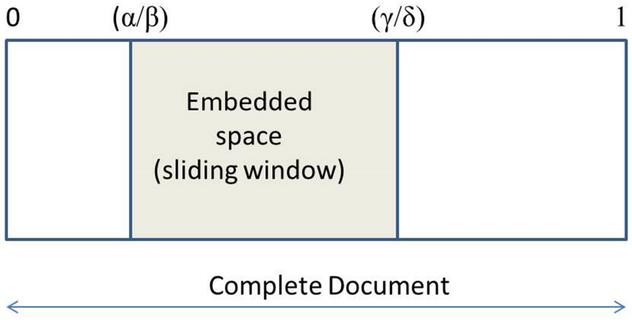
The complete process is as shown in Figure 3.2.


Watermarked Image
Partition image into (x',y') pairs. For each pair (x',y')
x'
1
0
Extract LSB(y'). Set LSB(x',y')=0 and recover original pairs
Encrypted payload
(x',y') with LSB's=1ϵDc
yes
Extract LSB(y'). Set LSB(x',y')=0 and recover original pairs
no
Watermark (patient info)
key
Original Image
Recover original pairs (x,y)
Payload
DES decyption
Hash Function
Extracted Hash
Image Hash
same
yes
accepted

(Figure 3.2: Restoration Process)

### 3.2 Copyright protection of other data formats

The reversible watermarking scheme is applied on data files other than images. The data hiding in images was possible on the whole image but in other file formats data hiding is always not possible on the whole document or file. A window is selected out of the document or file and that window is used for data embedding shown in Figure 3.3.



(Figure 3.3: Sliding Window of the Document)

The shaded region represents the embedding space. The payload can be embedded in this space and it is controlled by two variables (α/β) and (γ/δ). Each ratio decides the ends of the sliding window.

Also since meta data associated with that particular file too gets modified, most of the time the watermarked file is not available for use, for example a watermarked pdf document will report that "There was an error opening this document. The file is damaged and could not be repaired."

The **embedding** and **restoration** process is same as that applied on the images. The term "image" in Fig and is replaced by that particular file in which it is applied.

The scheme was successfully applied on a number of different file types such as .exe files, .pdf files, .docx documents, .pptx documents, cabinet files, disc image files, executable jar files, mp3 format sound, windows installer package, windows installer patch etc.

# Chapter 4: Results

## 4.1 Environmental Setup

The following configuration has been used while conducting the experiments:

**Hardware Configuration**

| | |
|---|---|
| Processor | : Intel core i5 |
| Processor Speed | : 2.30 GHz |
| Main Storage | : 2.00 GB |
| Hard Disk Capacity | : 500 GB |

**Software Configuration**

| | |
|---|---|
| Operating System | : Windows 7 |
| Language used | : C, MATLAB |

## 4.2 Results for copyright protection of medical images

The scheme was applied on 22 images taken from different sources and was compared with lee et al's data embedding scheme. The comparison is as shown in Table 4.1.

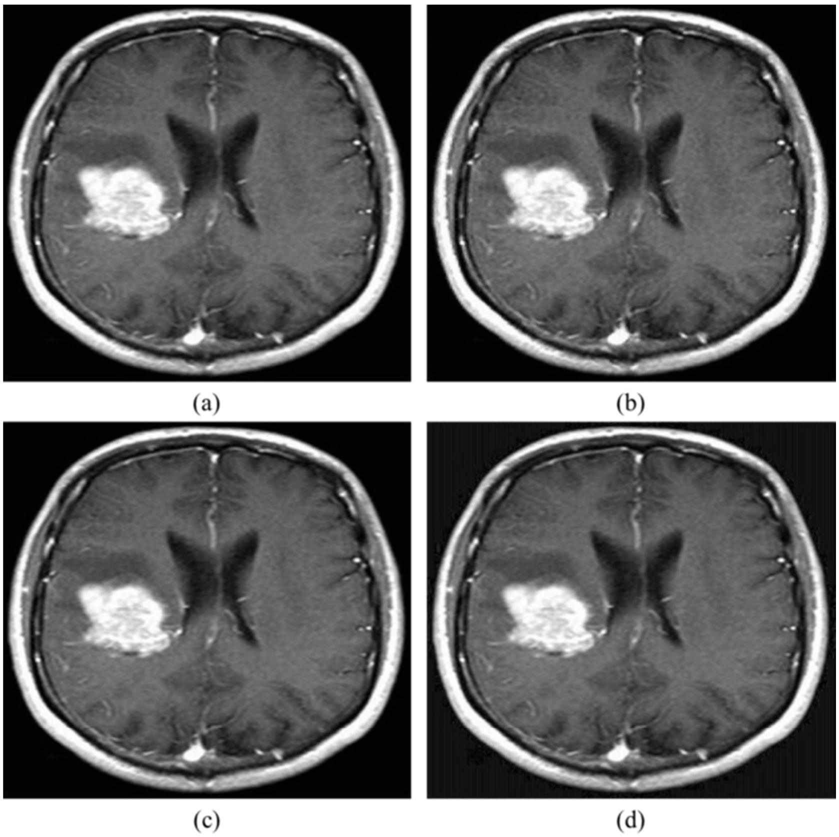
The rows red in colour shows those cases in which Lee et al's [3] scheme failed but proposed scheme could be used for data embedding without any payload compression.

The row in blue shows a case where both the schemes failed and neither could be used for data embedding without payload compression.

| Image Number | Coltuc et al's scheme | | Lee et al's scheme | |
|---|---|---|---|---|
| | Embedding Capacity | Payload size | Embedding Capacity | Payload size |
| 1. | 55816 | 19136 | 47614 | 70840 |
| 2. | 103693 | 1907 | 127805 | 2450 |
| 3. | 103552 | 2048 | 118440 | 2610 |
| 4. | 103362 | 2238 | 108210 | 2942 |
| 5. | 103027 | 2573 | 98069 | 3428 |
| 6. | 102275 | 3325 | 92020 | 3704 |
| 7. | 4492236 | 4916 | 871681 | 19002 |
| 8. | 958422 | 2730 | 160475 | 1504 |
| 9. | 2912986 | 77990 | 558528 | 1016288 |
| 10. | 1572448 | 1568 | 2045408 | 1650 |
| 11. | 149777 | 4975 | 31093 | 11630 |
| 12. | 40577 | 129208 | 298669 | 621404 |
| 13. | 127243 | 4981 | 134459 | 1554 |
| 14. | 308626 | 11990 | 73389 | 5116 |
| 15. | 70519 | 1243 | 66686 | 1154 |
| 16. | 99667 | 7163 | 60727 | 67600 |
| 17. | 146962 | 3048 | 112164 | 9852 |
| 18. | 201331 | 3635 | 57871 | 11424 |
| 19. | 162628 | 8381 | 26277 | 23610 |
| 20. | 159281 | 7857 | 95357 | 34044 |
| 21. | 138240 | 4113 | 155855 | 12694 |
| 22. | 4621173 | 148857 | 1383919 | 1562208 |

(Table 4.1: Comparison of Coltuc et al and Lee et al scheme for images)

Image 4.1(a) is a MRI image depicting a brain tumour. The proposed scheme was applied on the image, from the watermarked image both, the tampered and original image were recovered and the tampered image was easily identified and rejected.



(Fig.4.1: MRI Image Sample (a) Original Image (b) Watermarked Image (c) Recovered image (d) Tampered Image)

Fig.4.1(a) shows an original brain mri image of size 24.2KB and and the image was embedded with 19136 bits of payload comprising of 1024 bits of patient information, 128 bits of image hash, and the remaining original LSB bits. The payload was encrypted using a 64 bit key based DES algorithm. The resulting watermarked image obtained is shown in Fig.4.1(b).

The hash of the original image 4.1(a) is:

297e29d44c84d41b1a5cdab37c50296d

The embedded watermark is:

Name-Mr.Prakash
Age-25yrs
Sex-Male
Doctor-Dr.Ashwani Bansal
Serial-RGH135w02

It was padded with white spaces to complete 1024 bits.

The original image was recovered as shown in Fig.4.1(c) and the same watermark was also recovered.

**In case of Tamepering:**

In case of the tampered image, the watermarked image was tampered by changing the least significant bit of a single byte and the results shows a drastic change while it’s recovery. The recovered image is as shown in Fig.4.1(d).

The hash of the tampered image 4.1(d) is:

285cb932616c9bc6bd519149e4c3036f

The recovered watermark from tampered image is:

```
~é’”¡Õ
________________________________________________________________
,   |-„RQe   -]µ¦z[@B³Â5?åŽ'¤?PŽë6   6çílÝ…pæÚ]G¿éÌyr   Jº
"ËÐE¾ä¥?(¤íý3gœÃÖq   ¶/Ö_§€hdnšÊ?WWŒe?YèíKž|éÒ
vñ-¢Ì   –B
```

The hash obtained from the recovered image doesn’t match with the hash obtained from the decrypted payload. Thus, it was obtained that the image was being tampered.

### 4.3 Results for copyright protection of other data formats

The approach was applied on a number of different file types and the embedding area was detected by hit and trial method. The results shown below represents a comparison between, the embedding in the complete document and embedding using sliding window. The sliding window used is shown by a coloumn representing window size. It represents the embedding area of the document.

The embedding space selected as shown in the tables are not the only embedding space, there may be embedding space available at other locations throughout the document. Finding that manually is a tedious task.

PORTABLE DOCUMENT FORMAT

| Document number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 195982 | 213882 |
| 2 | 678827 | 1283440 |
| 3 | 1715323 | 1914280 |
| 4 | 2825811 | 1541392 |
| 5 | 206926 | 467476 |

(Table 4.2: Embedding Capacity and Payload Size of PDF files)

| Document number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 2327 | 952 | 98-99 |
| 2 | 24159 | 22936 | 97-100 |
| 3 | 97071 | 23916 | 96-100 |
| 4 | 2825811 | 1541392 | 0-100 |
| 5 | 5149 | 1596 | 99-100 |

(Table 4.3: Embedding Capacity and Payload Size of PDF files for a particular window size**)**

WINDOWS INSTALLER PACKAGE (.msi files)

| Document number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 2712918 | 5221034 |
| 2 | 115159 | 156201 |
| 3 | 4437195 | 8195893 |
| 4 | 190518 | 330954 |
| 5 | 2052753 | 2054511 |

(Table 4.4: Embedding Capacity and Payload Size of WIP files**)**

| Document number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 20800 | 10936 | 0.2-0.6 |
| 2 | 3160 | 1363 | 1-3 |
| 3 | 16301 | 15282 | 0.25-0.5 |
| 4 | 7703 | 5334 | 2-5 |
| 5 | 15865 | 4671 | 0-1 |

(Table 4.5: Embedding Capacity and Payload Size of WIP files for a particular window size**)**

MICROSOFT WORD DOCUMENT

| Document number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 165830 | 328871 |
| 2 | 680729 | 1470146 |
| 3 | 106425 | 217826 |
| 4 | 2952 | 3887 |
| 5 | 4538 | 6610 |

(Table 4.6: Embedding Capacity and Payload Size of MW files**)**

| Document number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 5655 | 4240 | 98-100 |
| 2 | 2365 | 1937 | 99.8-100 |
| 3 | 1523 | 1157 | 99.17-100 |
| 4 | 398 | 218 | 9-18 |
| 5 | 453 | 327 | 92-99 |

(Table 4.7: Embedding Capacity and Payload Size of MW files for a particular window size**)**

MP3 FORMAT SOUND

| File number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 395601 | 787199 |
| 2 | 2273790 | 4226578 |
| 3 | 2070300 | 3640640 |
| 4 | 186561 | 347069 |
| 5 | 1128089 | 2145522 |

(Table 4.8: Embedding Capacity and Payload Size of MP3 files)

| File number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 2755 | 1188 | 99.67-100 |
| 2 | 48628 | 16376 | 99-100 |
| 3 | 45865 | 11245 | 99-100 |
| 4 | 534 | 356 | 99.83-100 |
| 5 | 3751 | 1706 | 99.83-100 |

(Table 4.9: Embedding Capacity and Payload Size of MP3 files for a particular window size)

MICROSOFT POWER POINT PRESENTATION

| Document number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 634716 | 1132708 |
| 2 | 113990 | 228538 |
| 3 | 49962 | 77270 |
| 4 | 19581 | 32120 |
| 5 | 47353 | 69895 |

(Table 4.10: Embedding Capacity and Payload Size of Ppt files**)**

| Document number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 10470 | 7205 | 99-100 |
| 2 | 3932 | 2919 | 98-100 |
| 3 | 5021 | 2613 | 94-100 |
| 4 | 3965 | 3348 | 99.01-100 |
| 5 | 5312 | 2896 | 93-100 |

(Table 4.11: Embedding Capacity and Payload Size of Ppt files for a particular window size**)**

APPLICATION (Executable files)

| File number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 422875 | 194217 |
| 2 | 217016 | 333316 |
| 3 | 4647920 | 2254608 |
| 4 | 3482983 | 7019816 |
| 5 | 23422241 | 47457687 |

(Table 4.12: Embedding Capacity and Payload Size of exectable files**)**

| File number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 422875 | 194217 | 0-100 |
| 2 | 14101 | 13415 | 50-55 |
| 3 | 4647920 | 2254608 | 0-100 |
| 4 | 3247 | 1763 | 0.54-0.58 |
| 5 | 10396 | 7324 | 0.52-0.55 |

(Table 4.13: Embedding Capacity and Payload Size of exectable files for a particular window size**)**

EXECUTABLE JAR FILE

| File number | Embedding capacity | Payload size |
|---|---|---|
| 1 | 1208686 | 2355499 |
| 2 | 12701463 | 13146772 |
| 3 | 3740747 | 3805592 |
| 4 | 20443 | 34155 |
| 5 | 230160 | 216939 |

(Table 4.14: Embedding Capacity and Payload Size of exectable jar files**)**

| File number | Embedding capacity | Payload size | Window size |
|---|---|---|---|
| 1 | 21198 | 18405 | 98.89-100 |
| 2 | 345498 | 171467 | 98-100 |
| 3 | 1584858 | 1433678 | 60-100 |
| 4 | 2131 | 1145 | 94-100 |
| 5 | 230160 | 216939 | 0-100 |

(Table 4.15: Embedding Capacity and Payload Size of exectable jar files for a particular window size**)**

# Chapter 5: Conclusion

The current work has been focussed on extending the concept of reversible watermarking beyond images. For that numerous methods have been studied and the one discussed here is based on difference expansion that uses a pair of pixels to embed a unit bit. The bit is inserted in the LSB of one of the component of selected pair. The insertion is not generally possible in the whole document. So for that purpose a sliding window approach is used. The sliding window approach selects a section of the document or file for data embedding. It is selected in such a way so that in that region the embedding capacity is more than the payload that is comprising of the orignal bits for recovery and other information that needs to be hidden.

Beside this I have also suggested a protocol that could be used for secure transfer of medical images containing patient information. The payload comprising of the image hash, original LSB bits, patient information is encrypted using a DES encryption algorithm and then embedded in the image using Coltuc et al’s [] scheme. The DES algorithm is made to work in feedback mode so that if a singe bit of data gets changed the following data gets changed too. And in that case the recovered hash and the hash of the recovered image will never match resulting in a rejected image.

The window selection is really a tedious task and it is not possible to explore the complete embedding space of the document. The future scope of the work is to automate the means of selecting the window. This would add to increasing the embedding capacity of the document resulting in a optimal utilization and will provide better security.

# References

[1] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, Lossless Recovery of an Original Image Containing Embedded Data, US patent:6278791, 2001.

[2] J. Tian, Reversible Data Embedding Using a Difference Expansion, Reversible Data Embedding Using a Difference Expansion, IEEE transaction on circuits and systems for video technology, vol. 13, no. 8, pp. 890-896, Aug 2003.

[3] C. F. Lee, H. L. Chen, H. K. Tso, Embedding capacity raising in reversible data hiding based on prediction of difference expansion, The Journal of Systems and Software, vol. 83, pp. 1864–1872, 2010.

[4] D. Coltuc, J. M. Chassery, Very fast watermarking by reversible contrast mapping, IEEE signal processing letters, vol. 14, no. 4, pp. 255-258, April 2007.

[5] M. U. Celik, G. Sharma, A. M. Tekalp, E. Saber, Localized lossless authentication watermark (LAW), International Society for Optical Engineering, vol. 5020, pp. 689–698, 2003.

[6] S. Miaou, C. Hsu, Y. Tsai, H. Chao, A secure data hiding technique with heterogeneous data-combining capability for electronic patient records, 22nd Annual International conference of the IEEE Engineering in Medicine and Biology Society, pp. 280-283, Jul 23-28 2000.

[7] H. C. Huang, W. C. Fang, S. C. Chen, Privacy Protection and Authentication for Medical Images with Record-Based Watermarking, IEEE/NIH Life Science Systems and Applications Workshop, pp. 190-193, LISSA 2009.

[8] N. A. Memon, S. A. M. Gilani, S. Qayoom, Multiple Watermarking of Medical Images for Content Authentication and Recovery, IEEE 13th international multitopic conference, pp. 1-6, 2009.

[9] M. K. Kundu, S. Das, Lossless ROI Medical Image Watermarking Technique with Enhanced Security and High Payload Embedding, International Conference on Pattern Recognition, pp. 1457-1460, 2010.

[10] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, Reversible data hiding, IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 354-362, 2006.

[11] H. W. Tseng, C. P. Hsieh, Prediction-based reversible data hiding, Information Sciences vol. 179, no. 14, pp. 2460–2469, 2009.