

QR Codes With Watermarking and Security

Major Project submitted in partial fulfilment of the

Requirements for the award of degree of

Master of Technology

in

Information Systems

Submitted By:

Shivi Garg

(2K12/ISY/24)

Under the Guidance of:

Mr. Manoj Kumar

(Associate Professor)

(Department of Computer Engineering)



Department of Information Technology

Delhi Technological University

(2012-2014)

CERTIFICATE

This is to certify that **Shivi Garg (2K12/ISY/24)** has carried out the major project titled **QR Codes with Watermarking and 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 **2012-2014**. 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,

NewDelhi-110042

ABSTRACT

QR code (abbreviated from Quick Response Code) is a type of matrix barcode or two-dimensional barcode.

"Watermarking" is the process of hiding digital information in a carrier signal. 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.

Here we propose multiple schemes where QR codes can be used for security and authentication with image watermarking.

In the first scheme "**QR Codes with Reversible Watermarking**", the QR code is generated for the patient information and this QR code is hidden behind a carrier image. The generated image will be the watermarked image which is sent to the receiver. At the receiver end, the receiver will extract the QR code along with its original image is also extracted which will be same as the carrier image. This provides for the confidentiality, authentication and the integrity of the transferred image.

To extend authentication, second scheme "**QR codes with Authentication and Watermarking**" talks about making patient information more secure. Hash of patient information is generated followed by the key generation that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key. A signing algorithm given a message and a private key produces a signature. The QR Code of this message is generated and is hidden behind an image. At the Receiving end, message is decrypted using the public key and digital signature. The QR code of hashed information is recovered. This information is not hacked by any unintended user which ensures its authentication and non-repudiation.

In the third scheme "**Watermarking and image Processing**", we have extended the concept of watermarking in the field of image processing that is on changing the image attributes, what percentage of pixels match for the watermarked image which helps us to set the particular threshold to detect the watermark.

Keywords: QR Codes, Reversible Watermarking, Medical Imaging, Image Authentication and Confidentiality, Digital Signature.

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.

Shivi Garg

Roll No. 2K12ISY/24

M.Tech (Information Systems)

E-mail: shivi1989@gmail.com

Contents

Certificate	ii
Abstract	iii
Acknowledgement	iv
List of Figures	vii
List of Tables	ix
1. Introduction	1
1.1 Motivation	2
1.2 Research Objective	2
1.3 Scope of work	2
1.4 Organization of thesis	3
2. Literature Review	4
2.1 QR Codes	4
2.1.1 Structure of QR Codes	5
2.2 Reversible watermarking	6
2.2.1 Schemes applying data compression	7
2.2.2 Schemes based on difference expansion	9
2.2.3 Schemes using histogram bin shifting	10
2.2.4 Tian's data embedding using difference expansion	11
2.2.5 Coltuc, et al.'s Reversible Contrast Mapping Scheme	15

3. Proposed Approach	19
3.1 QR codes with reversible watermarking	19
a. QR codes with Authentication and Watermarking	19
b. Watermarking and image Processing	23
4. Results	25
4.1 Environmental Setup	25
4.2 Results for QR codes with reversible watermarking.....	25
4.3 Results for QR codes with Authentication and Watermarking	28
4.4 Results for Watermarking and Image Processing	33
5. Conclusion	45
References	46

List of Figures

2.1 Structure of QR Code	5
2.2 Flowchart of Reversible and Conventional Watermarking scheme	6
2.3 Domain D with and without Control Distortion	16
3.1 Embedding Process	22
3.2 Restoration Process	23
3.3 Watermarking and Image Processing	24
4.1 Main Form showing QR Code Generation and Watermarking	26
4.2 QR Code Generation	26
4.3 QR Code hidden behind a cover image	27
4.4 Original and Recovered Patient information	28
4.5 Hash of patient information	29
4.6 Public and Private key generation	29
4.7 Digital Certificate Generation	30
4.8 Hash Encrypted with Private key and encoded to QR Code	30
4.9 QR Code hidden behind a cover image	31
4.10 Decryption of hash using public key with digital certificate	31
4.11 QR Code decoded to correct patient information	32
4.12 QR Code decoded to wrong patient information	32
4.13 Original image and inverted image	33
4.14 Original image and Gray image	34
4.15 Positive Contrast Image	35
4.16 Negative Contrast Image	36
4.17 Image Brightness Positive	38
4.18 Image Brightness Negative	39

4.19 Gamma Filter	40
4.20 Colour Filter	41
4.21 Resize Image	43
4.22 Cropped Image	44

List of Tables

2.1 Character Storage capacity for different datatypes	19
4.1 Between QR Code hidden image and extracted image	29
4.2 percentage matching for the inverted image	33
4.3 percentage matching for the Gray image	34
4.4 percentage matching for the Contrast image	35
4.5 percentage matching for the Brightness value of an image	37
4.6 percentage matching for the Gamma filtered image	40
4.7 percentage matching for the Colour Filtered image	41
4.8 percentage matching for the Resized image	42
4.9 percentage matching for the cropped image	43

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. The use of QR codes with image watermarking makes it more robust and secure.

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 QR code image for the purpose of diagnostic. This technique is further extended for the authentication and non repudiation purposes. And later on we present an application aspect of using watermarking with image processing. The algorithm will be explained in the forthcoming chapters with the experimental results.

1.3 Scope of work

The application of these algorithm to audio and video files and documents of different types is still a problem. The technique is based on pixel pair 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. This work has been extended in the field of security for the purpose of authentication and non-repudiation. 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 all the three schemes –“QR Code and reversible watermarking”, “QR codes with Authentication and Watermarking” and “Watermarking and image Processing” briefly , motivation to do this thesis, my objective, and scope to do the work in same field. Chapter 2 provides a detailed picture of QR codes and reversible watermarking techniques. In chapter 3 I have presented above mentioned all proposed schemes. Chapter 4 includes the implementation details and experimental results. Finally chapter 5 concludes the thesis.

Chapter 2

Literature Review

Scheme I: - QR Codes with Reversible Watermarking

2.1 QR Codes:

A QR code is a two dimensional information storage tool. It is square shaped and it contains smaller squares. The information is encoded to the position of the small squares. Opposing to regular barcodes which are just one dimensional codes as the length of the lines does not hold information.

QR codes store series of alphanumeric characters and Japanese characters (kanji). The length of the stored information depends on the data type (*mode*, or input character set), version (1... 40, indicating the overall dimensions of the symbol), and error correction level. The maximum storage capacities occur for 40-L symbols (version 40, error correction level L)

Maximum character storage capacity (40-L)			
<i>character</i> refers to individual values of the input mode/data type			
Input mode	max. characters	bit/byte	possible characters, default encoding
Numeric only	7,089	3 $\frac{1}{3}$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Alphanumeric	4,296	5 $\frac{1}{2}$	0–9, A–Z (upper-case only), space, \$, %, *, +, -, ., /, :
Binary/byte	2,953	8	ISO 8859-1
Kanji/kana	1,817	13	Shift JIS X 0208

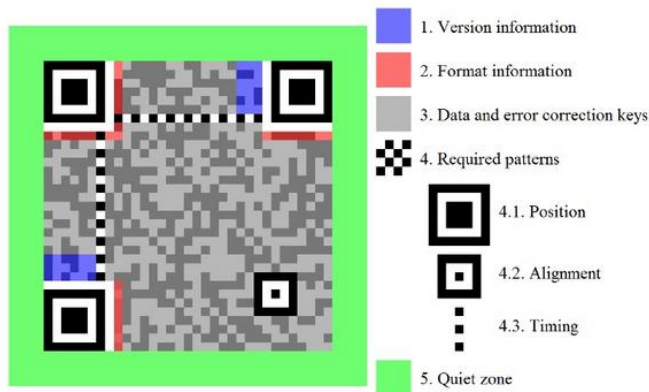
(Table 2.1: Character Storage capacity for different datatypes)

The QR code has a special ability: if you lose a part of the code its content can be recovered. There are four error correction levels[2]:

- Level L 7% of codewords can be restored.

- Level M 15% of codewords can be restored.
- Level Q 25% of codewords can be restored.
- Level H 30% of codewords can be restored.

2.1.1 Structure of QR Codes



(Figure 2.1: Structure of QR Code)

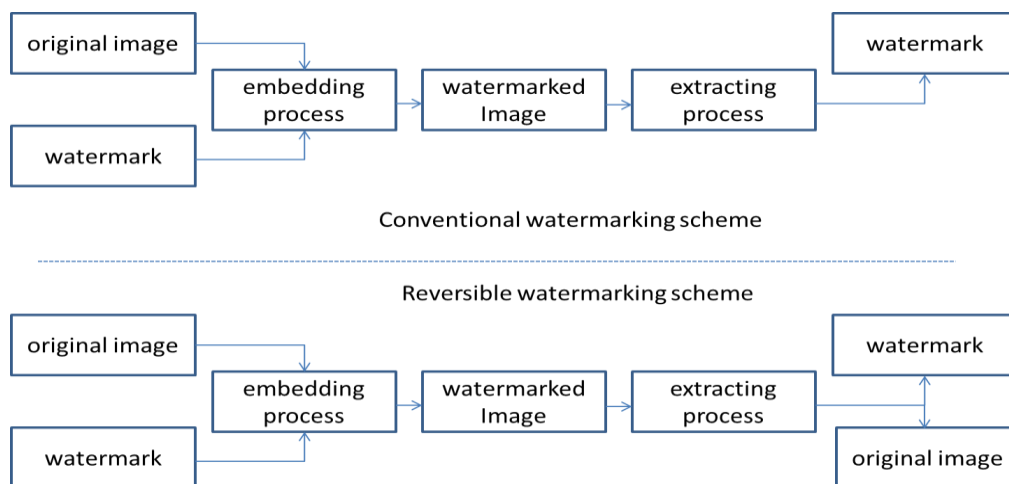
Structure of QR code

- **Finder/Position Pattern:** The finder/position pattern consists of three identical structures which are located in first three corners of the QR Code leaving the bottom right corner. Each pattern is based on a 3x3 matrix of black modules surrounded by white modules. These white modules are in turn surrounded by black modules. Using Finder Patterns, the decoder software recognizes the QR Code and determines the correct orientation of QR code.
- **Timing Pattern:** A pattern for identifying the central coordinate of each cell in the QR Code with black and white patterns arranged alternately. It is used for correcting the central coordinate of the data cell when the symbol is distorted or when there is an error for the cell pitch.
- **Alignment Patterns:** A pattern used for correcting the distortion of the QR Code. It is highly effective for correcting nonlinear distortions. The central coordinate of the alignment pattern will be identified to correct the distortion of the symbol. For this purpose, a black isolated cell is placed in the alignment pattern to make it easier to detect the central coordinate of the alignment pattern.

- **Format Information:** The Formation Information consists of 15 bits next to the separators and stores information about the error correction level of the QR Code and the chosen masking pattern.
- **Data and Error Correction:** The QR Code data are to be stored (encoded) into the data area. The data encoded into the binary numbers of ‘0’ and ‘1’ based on the encoding rule. The binary numbers of ‘0’ and ‘1’ will be converted into black and white cells and then will be arranged. Error correction codes are stored in 8 bit long code words in the error correction section.
- **Quiet Zone:** A margin space necessary for reading the QR Code. The quiet zone makes it easier to have the symbol detected from among the image read by the CCD sensor. Four or more cells are required for the quiet zone.

2.2 Reversible watermarking

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 2.2 presents a difference between the conventional and the reversible watermarking scheme.



(Figure 2.2: 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

2.2.1 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

1. 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}

2. 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

3. 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.

2.2.2 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:

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

The inverse transformation of (1) is defined as:

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

Embedding Process:

1. Calculate l_i and h_i for i^{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^{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^{th} pair of pixel (x_i', y_i') .

2. Calculate w_i by extracting the LSB of h_i , 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.

2.2.3 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 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.

2.2.4 Tian's data embedding using Difference Expansion

Tian[3] 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 $0 \leq x, y \leq 255$, the forward transform is defined as:

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

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

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

To prevent overflow and underflow in equation (4) we have $0 \leq x, y \leq 255$. This is equivalent to have $|h| \leq 2(255-l)$, and $|h| \leq 2l+1$ (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 $|h'| \leq \min(2(255-l), 2l+1)$ (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 $\left| 2 \times \left\lfloor \frac{h}{2} \right\rfloor + b \right| \leq \min(2(255-l), 2l+1)$ 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 $\{h_1, h_2, h_3, \dots, h_n\}$
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 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 .

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

Embedding will follow as:

i) Set $i=1$ and $j=0$.

ii) while ($i \leq m$)

$$j=j+1$$

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 $B=b_1b_2\dots b_m$.
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 $h_j \in \text{CH}$

$$\text{if location map value at } h_j = l \text{ then } h_j = \left\lfloor \frac{h_j}{2} \right\rfloor$$

else

if

$$0 \leq h_j \leq 1$$

then

$$\begin{aligned}
& h_j = 1 \\
\text{else if } -2 \leq h_j \leq -1 & \text{ then } h_j = -2 \\
\text{else } h_j = 2 \times \left\lfloor \frac{h_j}{2} \right\rfloor + b_i & \text{ and } i=i+1
\end{aligned}$$

iii) end

If the location map value is “1”, it means that the difference value was expanded during embedding. If $h \in \text{CH}$, $0 \leq h \leq 1$ and location map value is “0” then original value of $h=1$. Similarly if $h \in \text{CH}$, $-2 \leq h \leq -1$ 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.

2.2.5 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

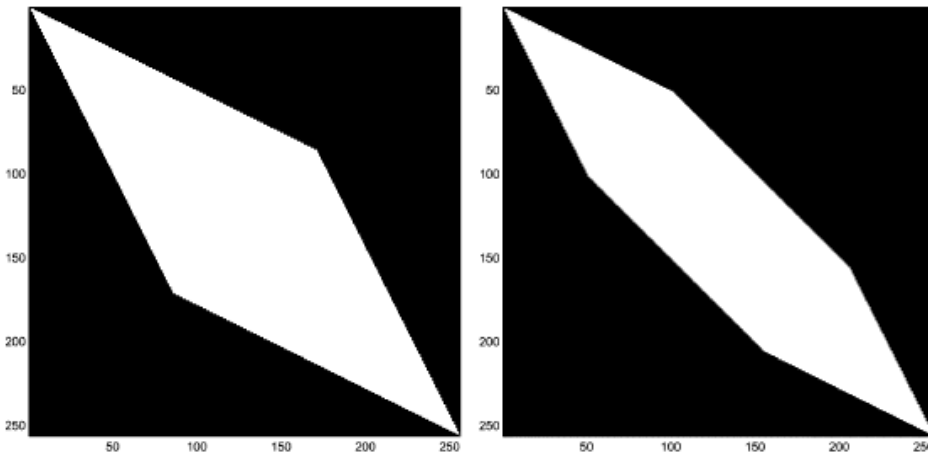
Let (x, y) be the initial pair of pixels, (x', y') be the transformed pair and $[0, L]$ be the range of pixel intensity.

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

$$x' = 2x - y \text{ and } y' = 2y - x \quad (12)$$

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:

$$0 \leq x' \leq L \text{ and } 0 \leq y' \leq L \quad (13)$$



(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:

$$x = \left\lfloor \frac{2}{3}x' + \frac{1}{3}y' \right\rfloor \text{ and } y = \left\lfloor \frac{2}{3}y' + \frac{1}{3}x' \right\rfloor \quad (14)$$

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 2.3, 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 comprises of three schemes:

- 3.1 QR codes with reversible watermarking
- 3.2 QR codes with Authentication and Watermarking
- 3.3 Watermarking and image Processing

3.1 QR codes with reversible watermarking

The patient information is a text file comprising of Name, Age , Sex, Serial number as its attribute. This information is encoded into the QR code. This QR code is watermarked in a Cover image and sent to the receiver. The receiver on the other end will extract this QR Code and along with the original image.

3.2 QR codes with Authentication and Watermarking

The payload being inserted in the QR Code image consists of the hash value of the patient information. The payload is encrypted using a key (provided by the user) based on the Digital Signature algorithm before embedding. This Digitally signed information is encoded into QR Code. The Secure code is then embedded in an image to form the watermarked image.

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 Algorithm

1. Message size is calculated in terms of number of bytes.
2. The secret message is converted into bitstream and appended to the message size.
(Let $M = \text{message size} + \text{message}$)
3. The pixels in which the message is to be hidden is selected based on the key value.
We start from the pixel at location (0,0). Second pixel is selected at a distance of ASCII value of the first character of the key from the first pixel. Then third pixel is selected at a distance of ASCII value of the second character of the key from the

previously selected pixel and so on till we have selected pixels equal to the number of bits in M. If key length (number of characters in key) is less than the number of bits in M then the same key is repeated in the loop starting from the first character of the key.

4. If the number of pixels in the image is less than the number of pixels which are required to hide the message then we will hide the message in cyclic manner with at most three cycles possible where the first pixel in the second and third cycles will be selected according to the key character ASCII value from the last selected pixel in the previous cycle. In the first cycle message bit will be hidden in the LSB of the R component (Red color in RGB model) in the selected pixels. In the second pass bit will be hidden in the LSB of the G component (Green color in RGB model)[10] and in the third pass bit will be hidden in the LSB of the B component (Blue color in RGB model).

Embedding Process

The embedding process is as shown in Figure 3.1. The hash of the patient information is calculated. This hash is encrypted with private key of the sender and digital signature id generated. This Digitally signed information is encoded into QR Code. The Secure code is then embedded in an image to form the watermarked image.

The key generation is based on the RSA algorithm.

RSA being a public key crypto-system has two keys, the Public key and the Private key. The Encryption is done using one and the decryption is done using the other. here, the encryption is done using the Private key and the decryption is done using the Public key. The RSA modulus length is called the key length of the cipher. As the security of RSA depends on the factoring problem, we used a modulus of 1024 bits.

After key generation, Digital signature is being generated.

A digital signature[11]scheme typically consists of:

- A signing algorithm that, given a message and a private key, produces a signature.
- A signature verifying algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.

To create RSA signature keys[12], generate an RSA key pair containing a modulus N that is the product of two large primes, along with integers e and d such that $e d \equiv 1 \pmod{\varphi(N)}$, where φ is the Euler phi-function. The signer's public key consists of N and e , and the signer's secret key contains d .

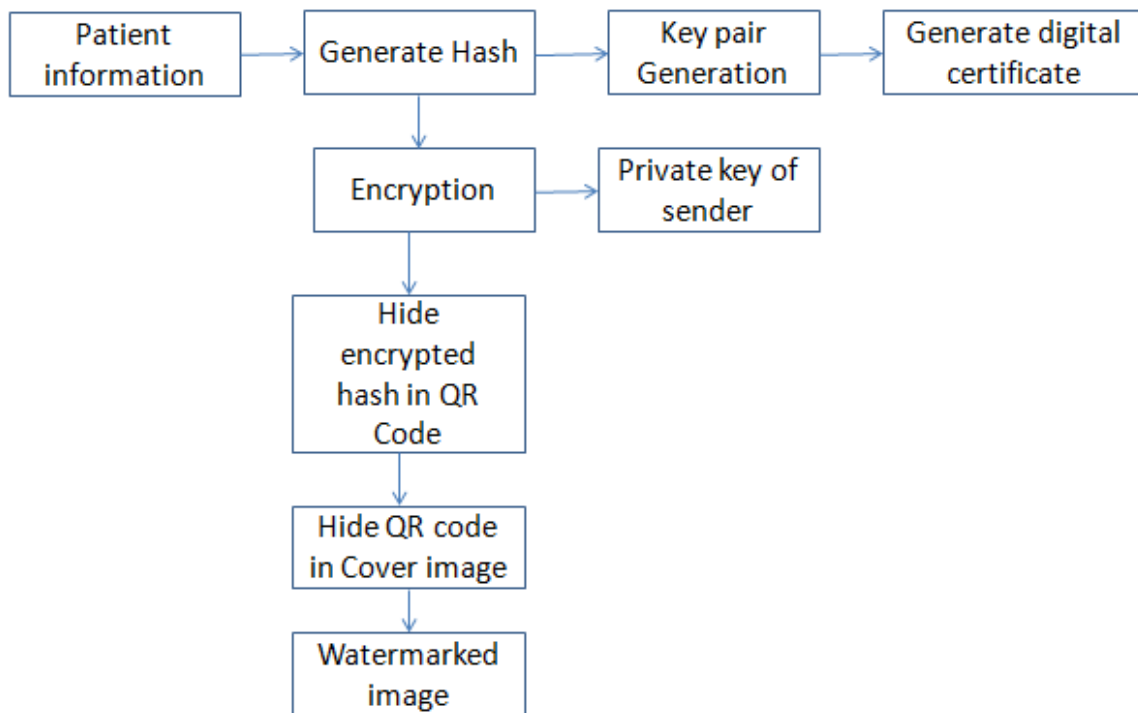
To sign a message m , the signer computes $\sigma \equiv m^d \pmod{N}$. To verify, the receiver checks that $\sigma^e \equiv m \pmod{N}$.

There are several reasons to sign such a hash (or message digest) instead of the whole document.

- **For efficiency:** The signature will be much shorter and thus save time since hashing is generally much faster than signing in practice.
- **For compatibility:** Messages are typically bit strings, but some signature schemes operate on other domains (such as, in the case of RSA, numbers modulo a composite number N). A hash function can be used to convert an arbitrary input into the proper format.
- **For integrity:** Without the hash function, the text "to be signed" may have to be split (separated) in blocks small enough for the signature scheme to act on them directly. However, the receiver of the signed blocks is not able to recognize if all the blocks are present and in the appropriate order.

This Digitally Signed hash of patient information is converted into QR Code.

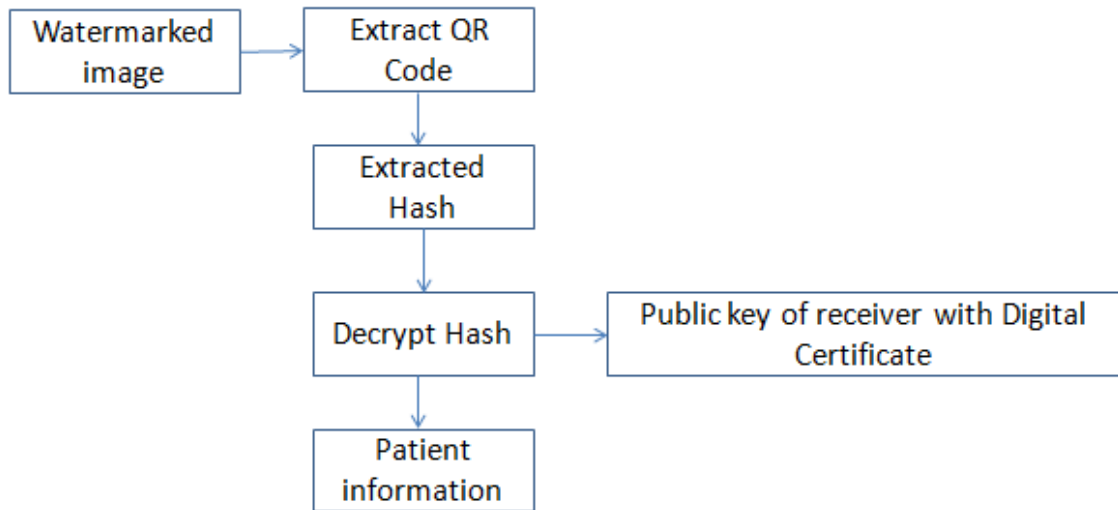
This QR code is hidden in a cover image to create a water marked image by replacing the LSB's of the cover image.



(Figure 3.1: Embedding Process)

Restoration Process

In the restoration process we apply the extraction process of Coltuc et al scheme and extract the encrypted QR Code. To decode the QR Code any QR code decoder can be used and encrypted hash of the patient information is obtained. To recover the original patient information it is decrypted by using the public key and the digital signature. If some different QR is decrypted then this information will not match and it is rejected. Or if the hash is not matched then also it will be rejected confirming that the sender is not authentic and the image has been tampered.



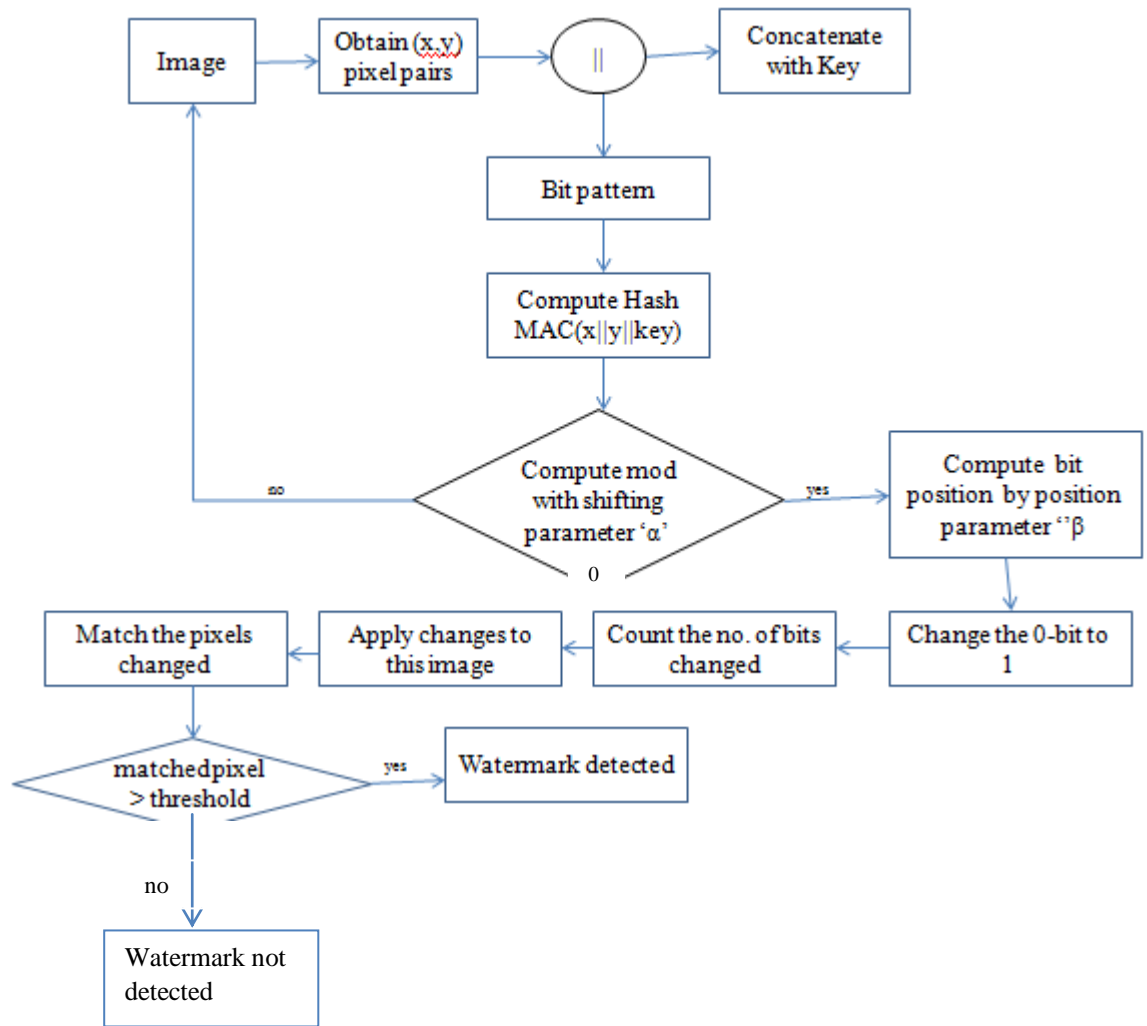
(Figure 3.2: Restoration Process)

3.3 Watermarking and image Processing

We would like to extend this concept in the field of image processing[14]. In this we are trying to hide 1 bit information in the pixels of an image. On changing this image by some image attributes like inverting, gray scale, contrast, brightness, cropping, resizing and colour filtering by varied degree in the pixels of an image, we try to compare how much matching we get in the watermark so that a proper threshold can be set to detect the presence of a watermark[13].

Proposed Scheme:-

In this an image is taken for which (x, y) pixel pairs are obtained. Secret key is concatenated with the pixel pair. Then we compute the hash of the concatenated bit pattern. Compute the mod with the shifting parameter ' α '. This will give the position for the pixels. If the mod result is 0, then compute the position of the LSB bit by taking the mod with the position parameter ' β '. Change the 0 bit to 1 bit and count the number of pixels changed. Next step is to change the image by modifying the pixels by 10 %, 20% or by applying different image processing methods like contrast, colour filter and match the number of pixels with the changed pixels. If the matched pixel is greater than the set threshold, then the watermark is detected else rejected.



(Figure 3.3: Watermarking and Image Processing)

Chapter 4

Results

4.1 Environmental Setup

The following configuration has been used while conducting the experiments:

Hardware Configuration

Processor : Intel core2 Duo

Processor Speed : 1.4 GHz

Main Storage : 4.00 GB

Hard Disk Capacity : 500 GB

Software Configuration

Operating System : Windows 7

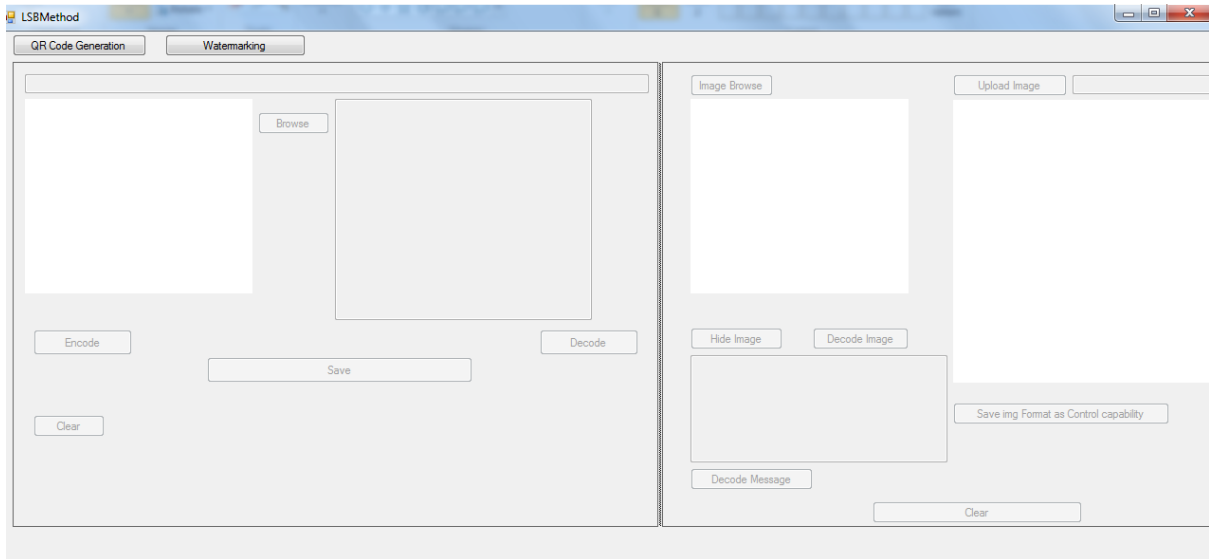
Language used : C#

Software used : Visual Studio 2010 with dot net
Framework 4.5

4.2 Results for QR codes with reversible watermarking

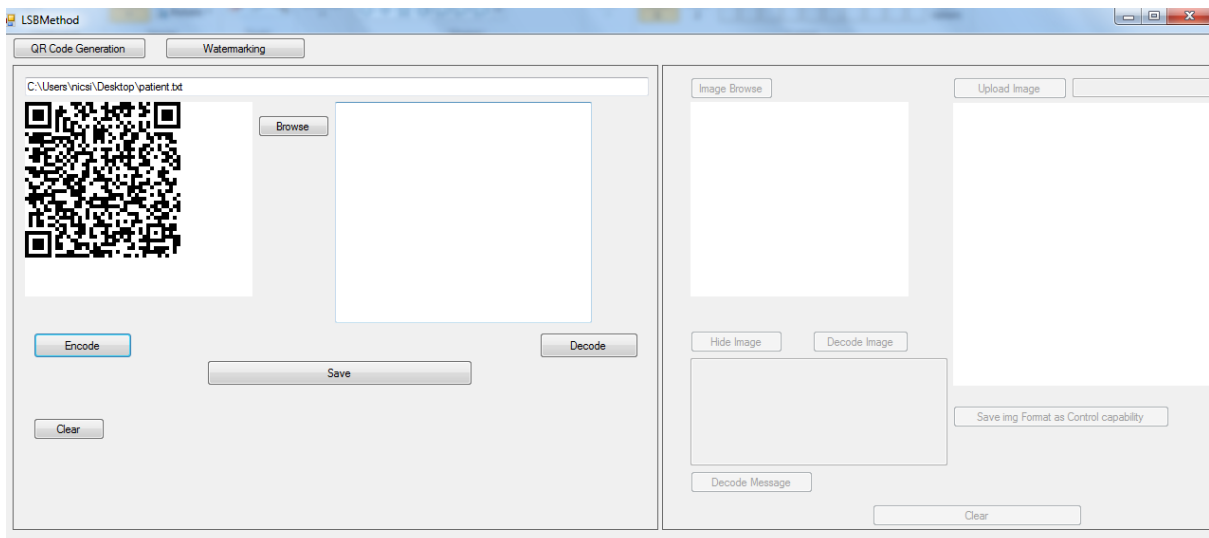
The scheme is applied to the patient information containing the Name, Sex and Age which is converted into the QR Code.

The main form appears to be like as shown in the figure 4.1:-



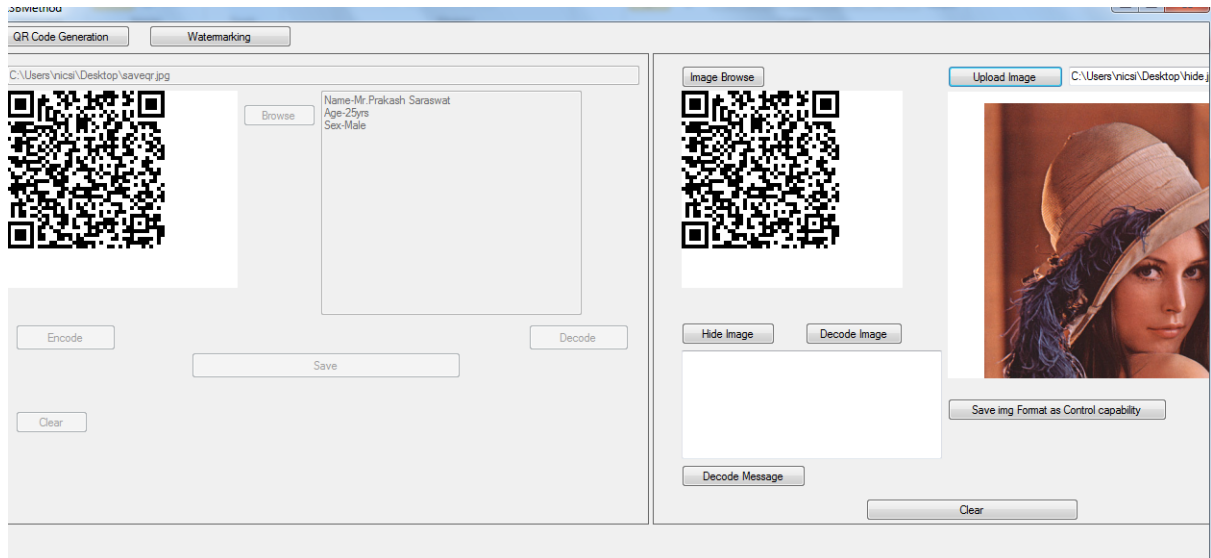
(Figure 4.1: Main Form showing QR Code Generation and Watermarking)

The QR Code generated as shown the figure 4.2:



(Figure 4.2: QR Code Generation)



This QR code is then hidden in an image and image is named as “hide.jpg” as shown in the figure 4.3 given below.



(Figure 4.3: QR Code hidden behind a cover image)

When this image is decoded, QR code will be extracted from the image and new image is saved as “extracted.jpg”.

Comparison is made between the attributes of hide.jpg and extracted.jpg in the table 4.1

Image Attribute	hide.jpg	extracted.jpg
		
Size	631KB	631KB
Dimensions	512 X 512	512 X 512
Resolution	96dpi	96dpi

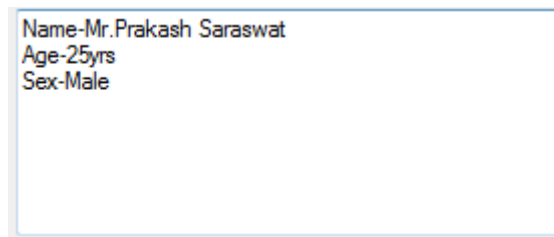
(Table 4.1: Comparison between QR Code hidden image and extracted image)

QR Code is then decoded to get the original patient information.

In case of Tampering:

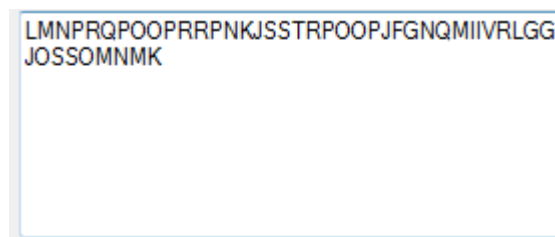
In case of the tampered image, the watermarked image was tampered by changing the least significant bit of a single byte and the results show a drastic change while it’s recovery. The

recovered QR Code will not be decoded into original patient information is as shown in Figure 4.4(b)



Name-Mr.Prakash Saraswat
Age-25yrs
Sex-Male

(a)



LMNPRQPOOPRRPNKJSSTRPOOPJFGNQMIIVRLGG
JOSSOMNMK

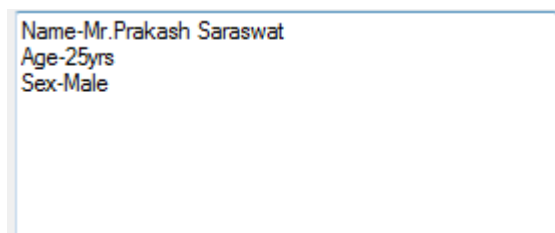
(b)

(Figure 4.4(a): Original Patient information (b) Recovered patient information)

4.3 Results for QR codes with Authentication and Watermarking

This approach is applied for transmission of more secure information for authentication and non repudiation.

Original patient information:



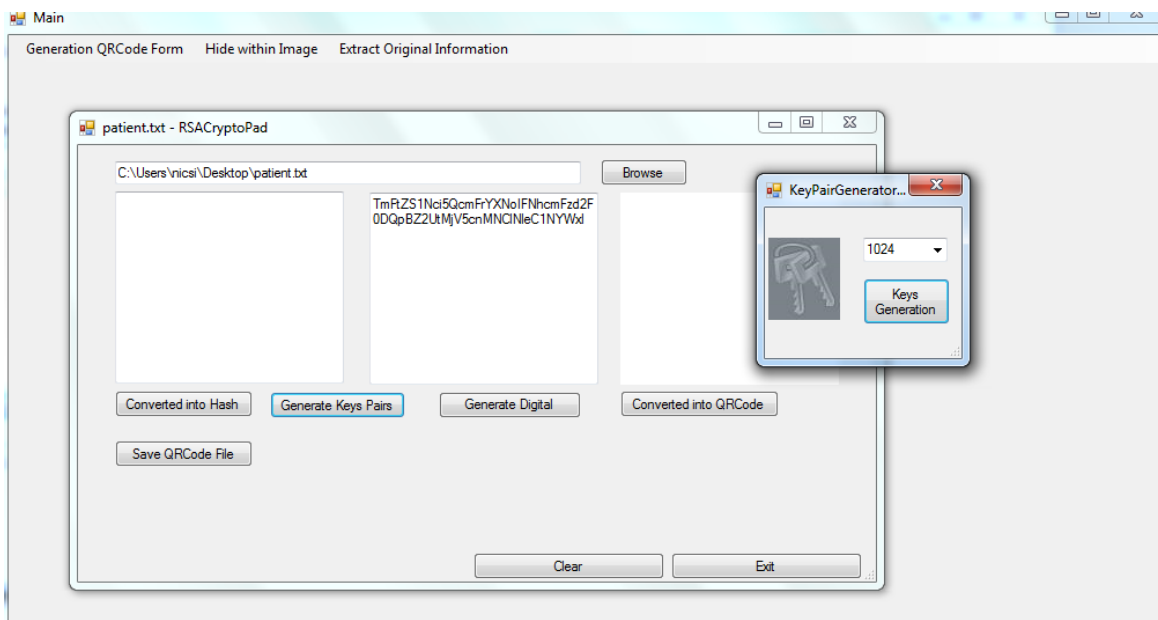
Name-Mr.Prakash Saraswat
Age-25yrs
Sex-Male

Hash of the patient information is generated as shown in the figure 4.5:

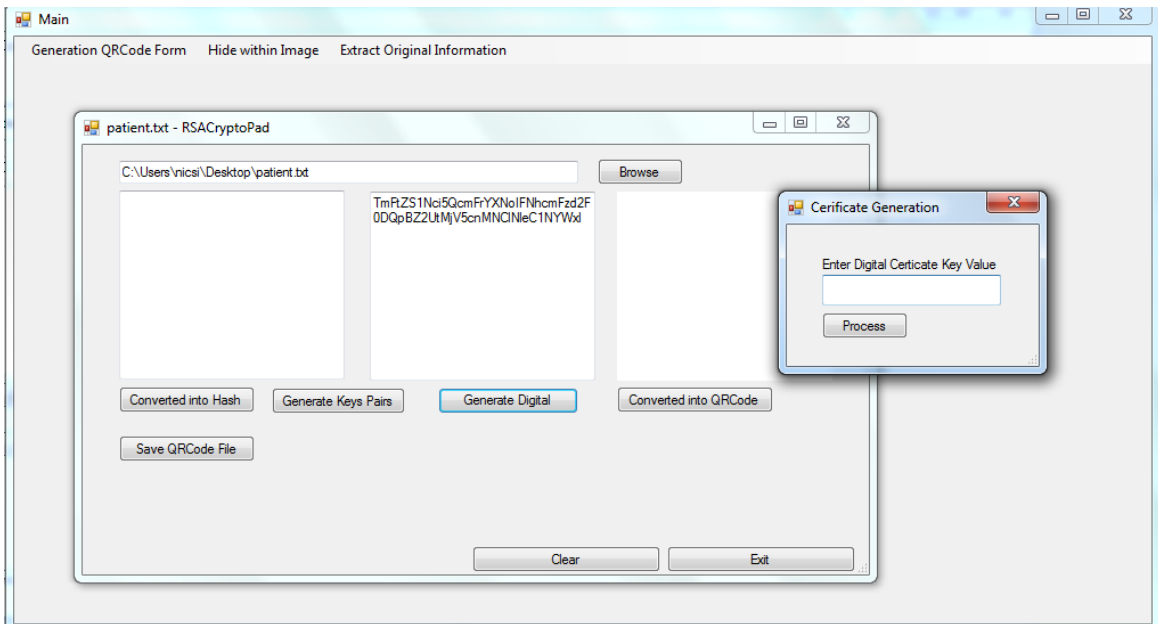
```
TmRtZS1Nci5QcmFrYXNoIFNhcmFzd2F  
0DQpBZ2UtMjV5cnMNCINleC1NYWxl
```

(Figure 4.5: Hash of patient information)

Then key pairs are generated as- Public and Private key as in figure 4.6. After this Digital Certificate is generated as shown in the figure 4.7:

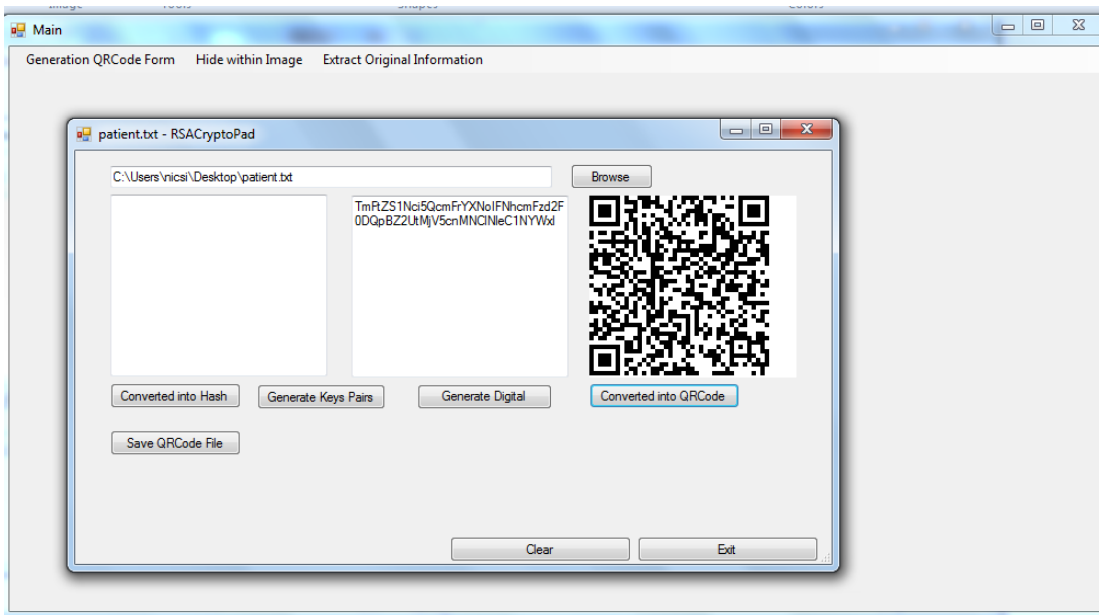


(Figure 4.6: Public and Private key generation)



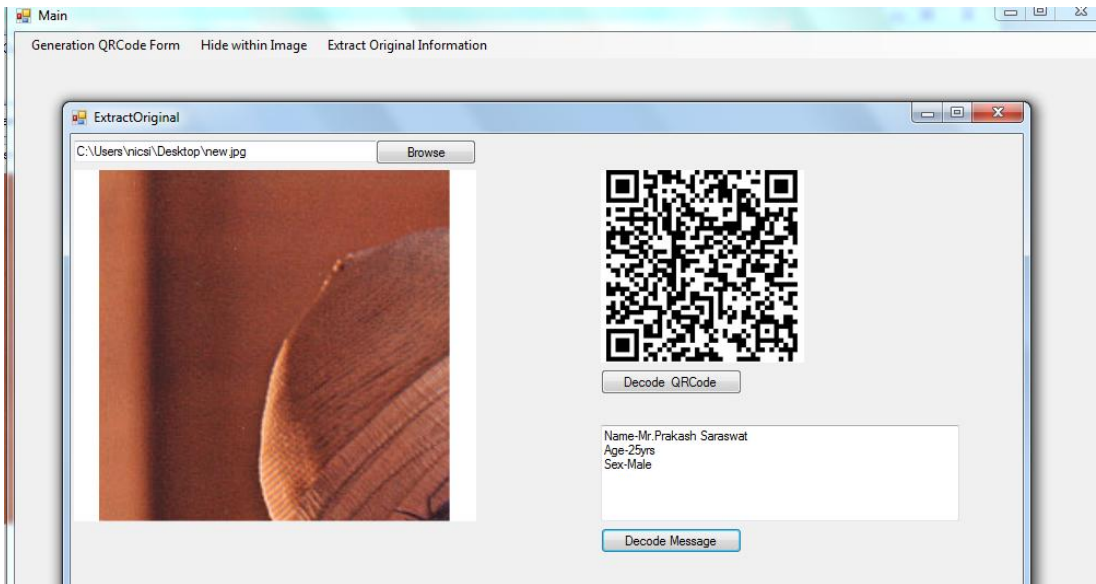
(Figure 4.7: Digital Certificate Generation)

The hashed information is encrypted using the Private key which is then encoded to QR Code as in figure 4.8.



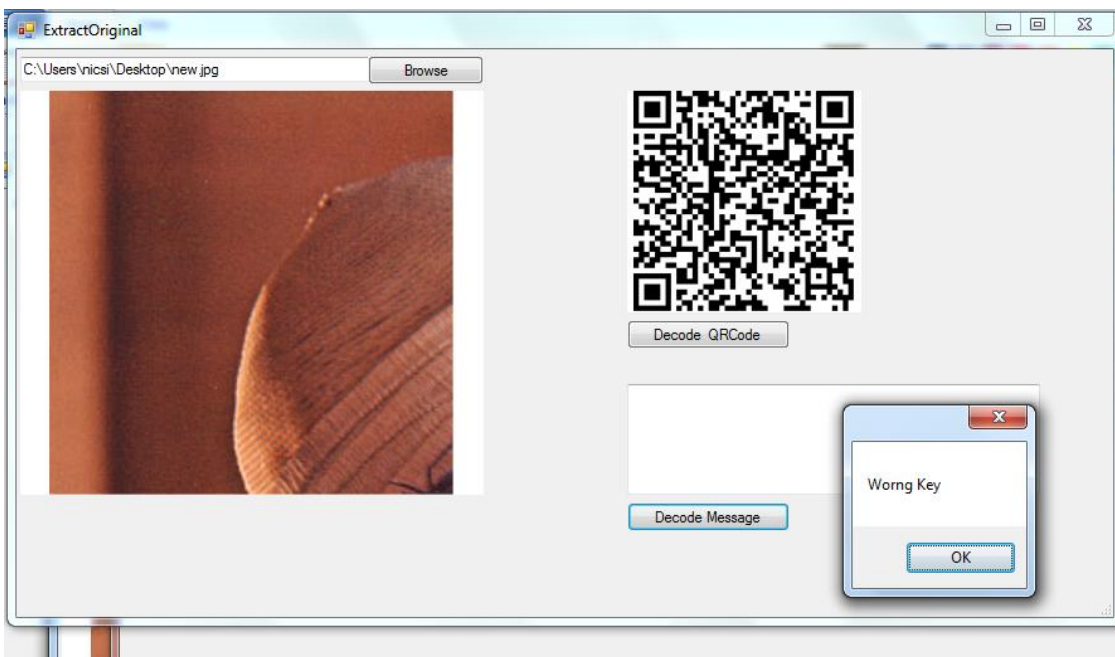
(Figure 4.8: Hash Encrypted with Private key and encoded to QR Code)

This QR code is then hidden behind a cover image as shown in the figure 4.9.



(Figure 4.11: QR Code decoded to correct patient information)

If the key pairs do not match, then QR code cannot be decrypted to the original information as in figure 4.12.



(Figure 4.12: QR Code decoded to wrong patient information)

4.4 Results for Watermarking and Image Processing

In this scheme, we try to apply various digital image processing operations to an image and compare the matching number of pixels in watermarked image to detect a watermark and accordingly set a threshold value.

Digital image processing operations along with the results are listed in a table 4.2 below:

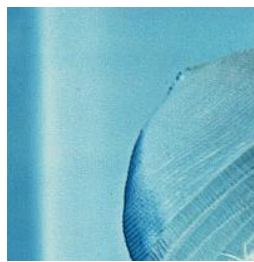
TOTAL PIXELS COUNT OF AN IMAGE: 40804

1) INVERT AN IMAGE:

It simply inverts a bitmap, meaning that each pixel value is subtracted from 255. The Invert command inverts all the pixel colors and brightness values in the current layer, as if the image were converted into a negative. Dark areas become bright and bright areas become dark. Hues are replaced by their complementary colors[15].



(a) Original image



(b) inverted image

(Figure 4.13 (a) Original image (b) inverted image)

Total bits changed	Matching bits	Percentage matched
5158	1804	34.97%

(Table 4.2: percentage matching for the inverted image)

2) GRAY SCALE:

Gray scale filtering is in reference to the colour mode of a particular image. A gray scale image would be a black and white image; any other colour would not be included in it.

Basically, it's a black and white image; the colours in that image, if any, will be converted to the corresponding shade of gray (mid tones between black and white) thus making each bit of the image still differentiable[16].



(a) Original image

(b) Gray image

(Figure 4.14 (a) Original image (b) Gray image)

Total bits changed	Matching bits	Percentage matched
5158	3342	64.79%

(Table 4.3: percentage matching for the Gray image)

3) CONTRAST (values between -100 and 100)

Contrast refers to the amount of colour or gray scale differentiation that exists between various image features in digital images. Images having a higher contrast level generally display a greater degree of colour or gray scale variation than those of lower contrast[17].

Contrast Value	Total bits changed	Matching bits	Percentage matched	Contrast Value	Total bits changed	Matching bits	Percentage matched
+10	5158	3956	76.69%	-10	5158	3267	63.33%
+20	5158	3091	59.92%	-20	5158	3239	62.79%
+30	5158	1381	26.77%	-30	5158	3619	69.98%
+40	5158	1169	22.66%	-40	5158	2722	52.77%
+50	5158	457	8.86%	-50	5158	4867	94.35%
+60	5158	160	3.10%	-60	5158	1858	36.02%
+70	5158	147	2.84%	-70	5158	3662	70.99%

(Table 4.4: percentage matching for the Contrast image)

Images with positive contrast values: As the contrast value increases, the percentage matching of the pixels is reduced. Images with the positive contrast value is shown in the figure 4.15

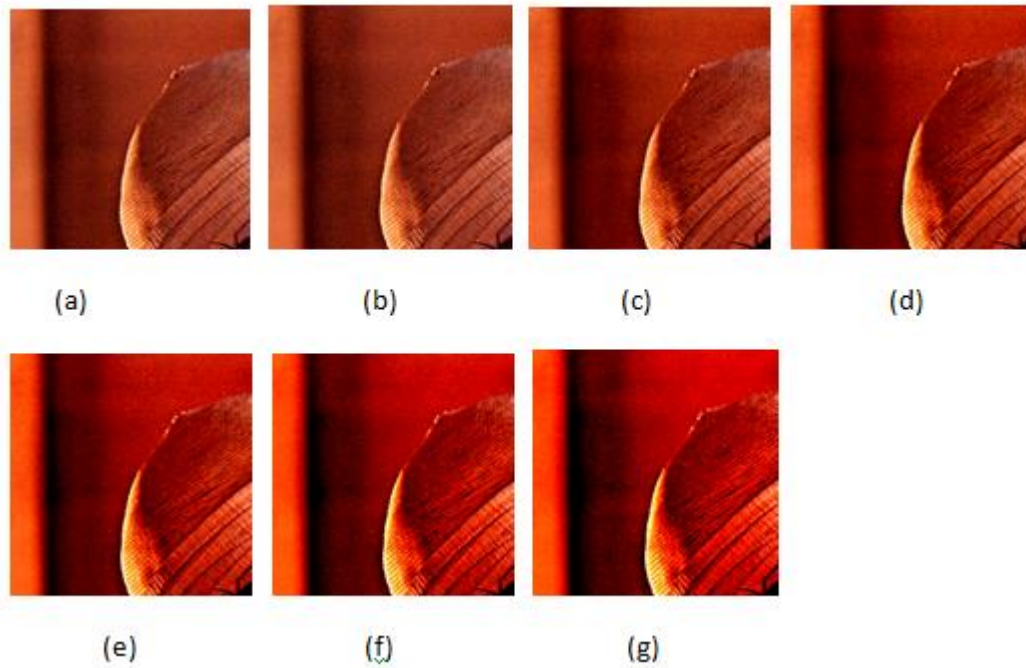


Figure 4.15: Positive Contrast Image: (a) Contrast +10 (b) Contrast +20 (c) Contrast +30
(d) Contrast +40 (e) Contrast +50 (f) Contrast +60 (g) Contrast +70

Images with negative contrast values: As the contrast value is reduced, the pixel matching shows an absurd behaviour of up and down values. Images with the negative contrast is shown in the figure 4.16.

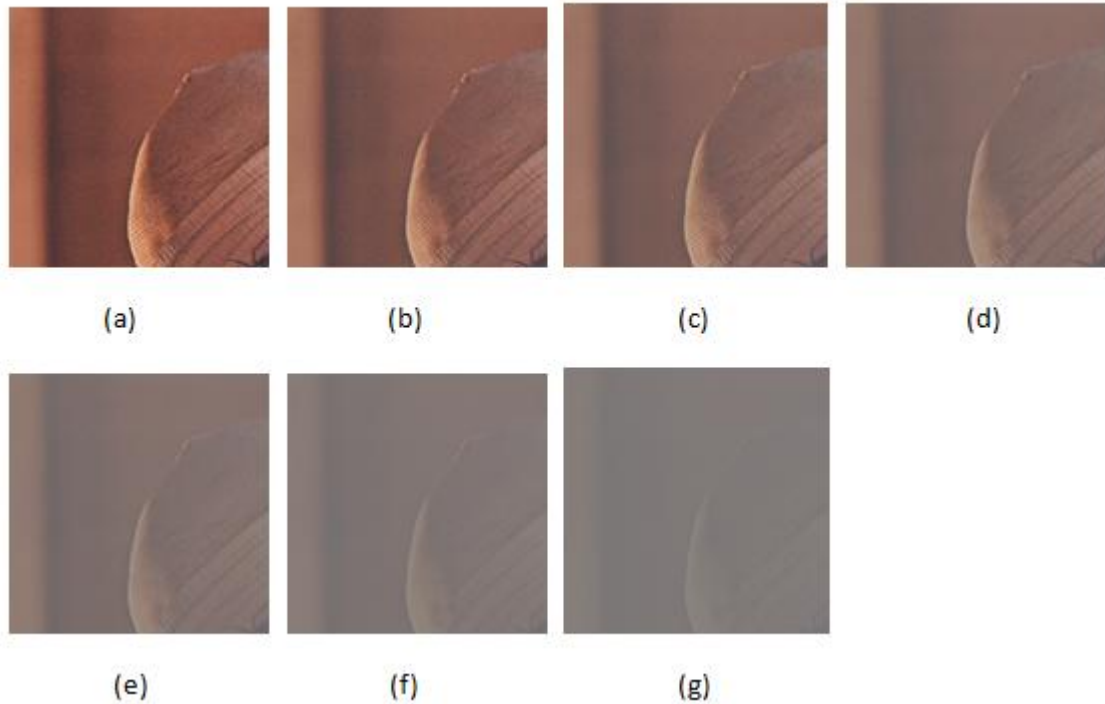


Figure 4.16: Negative Contrast Image: (a) Contrast -10 (b) Contrast -20 (c) Contrast -30
(d) Contrast -40 (e) Contrast -50 (f) Contrast -60 (g) Contrast -70

4) BRIGHTNESS (values between -255 and 255):

Brightness refers to the overall lightness or darkness of the image. The Brightness filter adds a value to each pixel, and if we go over 255 or below 0 the value is adjusted accordingly and so the difference between pixels that have been moved to a boundary is discarded. Doing a Brightness filter of 100, and then of -100 will not result in the original image - we will lose contrast. The reason for that is that the values are clamped[18].

Brightness Value	Total bits changed	Matching bits	Percentage matched	Brightness Value	Total bits changed	Matching bits	Percentage matched
-25	5158	3398	65.87%	+25	5158	3389	65.70%
-50	5158	1985	38.48%	+50	5158	5158	100%
-75	5158	165	3.19%	+75	5158	1715	33.24%
-100	5158	25	0.48%	+100	5158	3258	63.16%
-125	5158	14	0.27%	+125	5158	2567	49.76%
-150	5158	9	0.17%	+150	5158	1814	35.17%
-175	5158	5	0.09%	+175	5158	4864	94.3%
-200	5158	0	0%	+200	5158	1891	36.66%
-225	5158	0	0%	+225	5158	14	0.27%

(Table 4.5: percentage matching for the Brightness value of an image)

Image results with positive brightness are shown in the figure 4.17

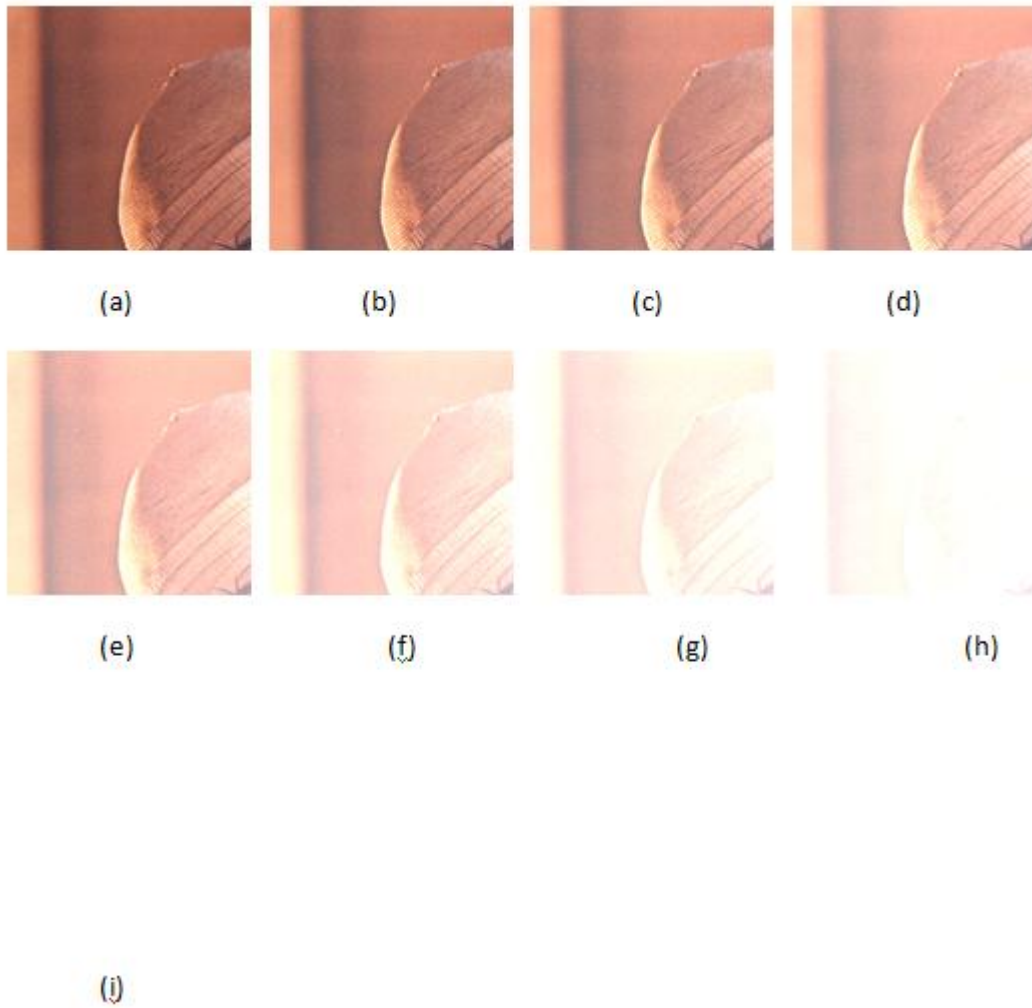


Figure 4.17: Image Brightness Positive: (a) Brightness +25 (b) Brightness +50 (c) Brightness +75 (d) Brightness +100 (e) Brightness +125 (f) Brightness +150 (g) Brightness +175 (h) Brightness +200 (i) Brightness +225

Image results with negative brightness shown as in figure 4.18

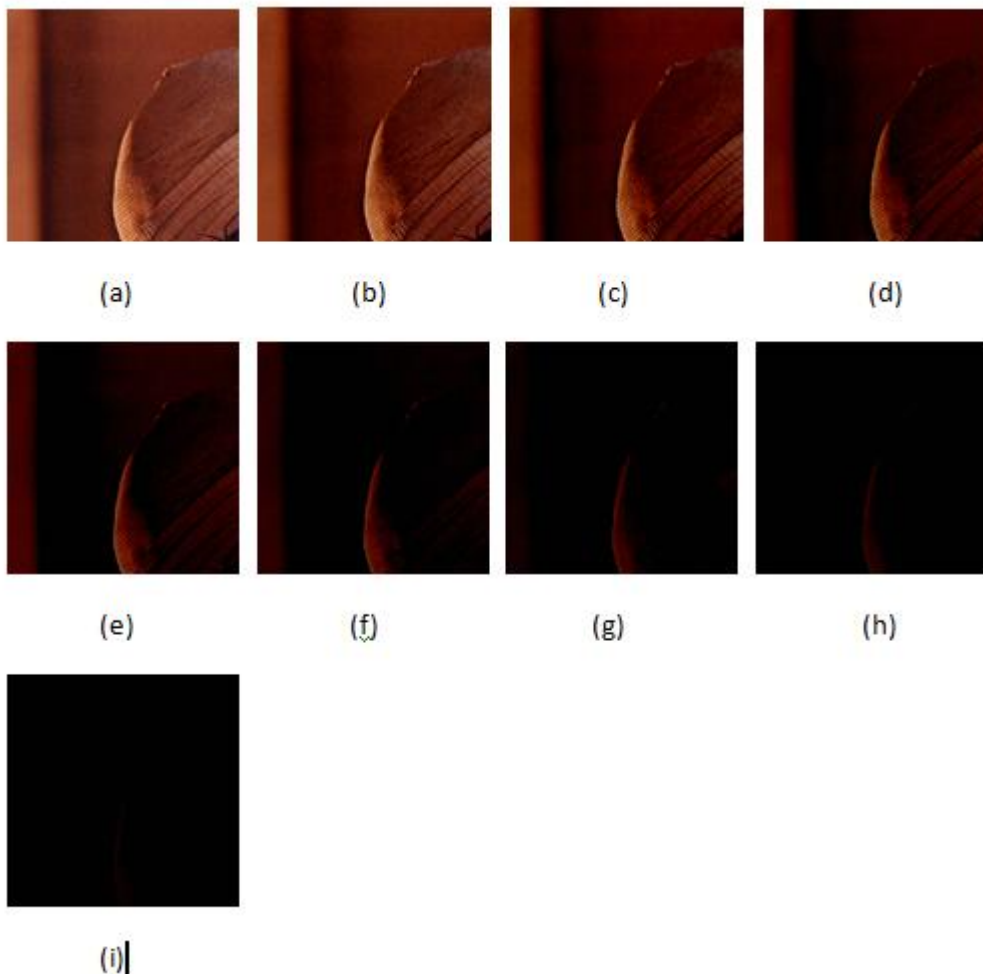


Figure 4.18: Image Brightness Negative: (a) Brightness -25 (b) Brightness -50 (c) Brightness -75 (d) Brightness -100 (e) Brightness -125 (f) Brightness -150 (g) Brightness -175 (h) Brightness -200 (i) Brightness -225

5) GAMMA (values between 0.2 and 5 for RGB)

A gamma filter works by creating an array of 256 values called a gamma ramp for each value of the red, blue and green components [19]. The gamma value must be between 0.2 and 5.

The formula for calculating the gamma ramp is

$$255 * (i / 255)^{1/\text{gamma}} + 0.5.$$

If this value is greater than 255, then it is clamped to 255. It is possible to have a different gamma value for each of the 3 colour components. Then for each pixel in the image, we can substitute the value in this array for the original value of that component at that pixel.

RGB Values	Total bits changed	Matching bits	Percentage matched
(0.2,0.2,0.2)	5158	29	0.56%
(0.5,0.5,0.5)	5158	3419	66.28%
(0.8,0.8,0.8)	5158	4050	78.51%
(1,1,1)	5158	5158	100%
(3,3,3)	5158	4757	92.22%
(5,5,5)	5158	3475	67.37

(Table 4.6: percentage matching for the Gamma filtered image)

Image results for the Gamma filter as shown in the below figure 4.19

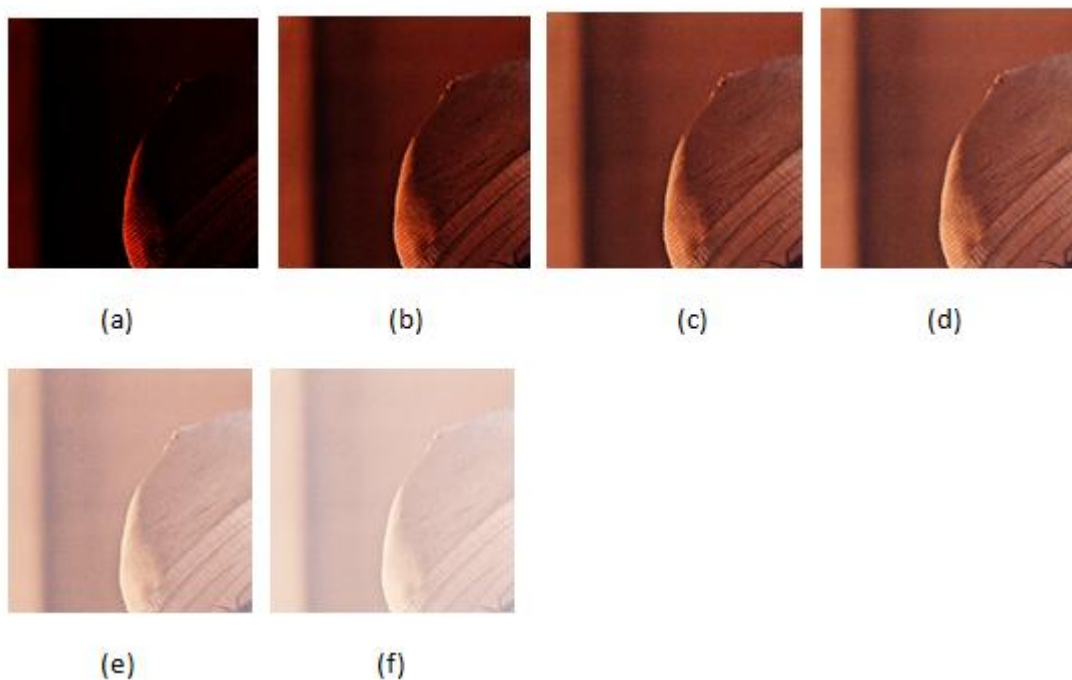


Figure 4.19: Gamma Filter: (a) Gamma 0.2 (b) Gamma 0.5 (c) Gamma 0.8
(d) Gamma 1 (e) Gamma 3 (f) Gamma 5

6) COLOUR FILTER

Colour filters are sometimes classified according to their type of spectral absorption: short-wavelength pass, long-wavelength pass, or band-pass; diffuse or sharp-cutting; monochromatic or conversion. The short-wavelength pass transmits all wavelengths up to the specified one and then absorbs. The long-wavelength pass is the opposite. Every filter is a band-pass filter when considered generally [20].

It just adds or subtracts a value to each colour. The most useful thing to do with this filter is to set two colours to -255 in order to strip them and see one colour component of an image. For example, for red filter, keep the red component as it is and just subtract 255 from the green component and blue component.

Filter	Total bits changed	Matching bits	Percentage matched
RED Filter	5158	0	0%
GREEN Filter	5158	0	0%
BLUE Filter	5158	5158	100%

(Table 4.7: percentage matching for the Colour Filtered image)

Image results:

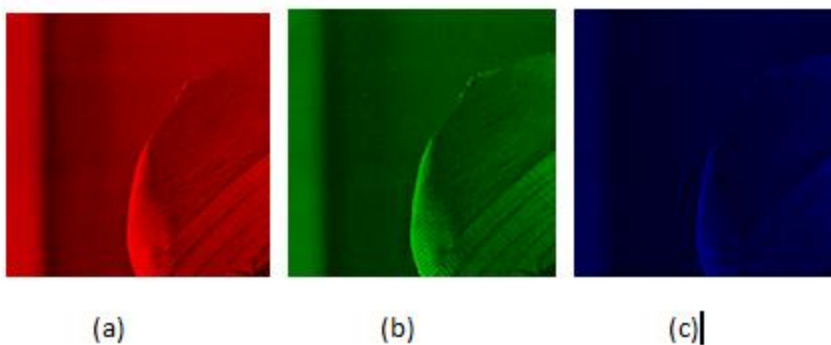


Figure 4.20: Colour Filter: (a) RED Filter (b) GREEN Filter (c) BLUE Filter

7) RESIZE AN IMAGE :

When image is resized, total number of the pixels are reduced.

Original image Dimensions: 202 X 202

Resized Dimensions	Total bits changed	Matching bits	Percentage matched
10 X 10	5158	7	0.135%
30 X 30	5158	85	1.65%
50 X 50	5158	226	4.38%
70 X 70	5158	469	9.09%
90 X 90	5158	708	13.72%
110 X 110	5158	1107	21.46%
130 X 130	5158	1464	28.38%
150 X 150	5158	2000	38.77%
170 X 170	5158	2544	49.32%
190 X 190	5158	3280	63.59%
200 X 200	5158	3634	70.45%

(Table 4.8: percentage matching for the Resized image)

Image results: As the image size reduces, the pixels matching will reduce because bit information is lost will resizing it to a smaller dimension. Image results are shown as in figure 4.21

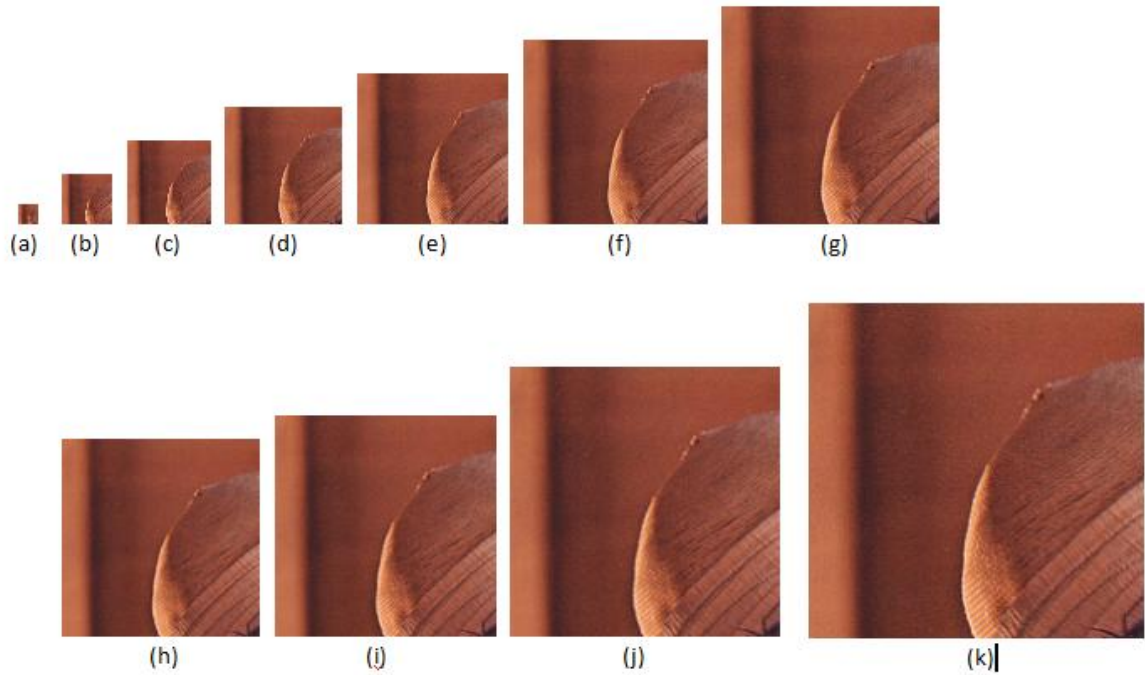


Figure 4.21: Resize Image: (a) 10 X 10 (b) 30 X 30 (c) 50 X 50 (d) 70 X 70 (e) 90 X 90
 (f) 110 X 110 (g) 130 X 130 (h) 150 X 150 (i) 170 X 170
 (j) 190 X 190 (k) 200 X 200

8) CROP AN IMAGE:

When image is cropped, some material from the edges is trimmed to show a smaller area.

XY Coordinates	Total bits changed	Matching bits	Percentage matched
(10 , 10)	5158	3095	60.00%
(30 ,30)	5158	3041	58.95%
(40 , 40)	5158	2150	41.68%
(50 , 50)	5158	2028	39.31%
(70 , 70)	5158	1450	28.11%
(100 , 100)	5158	855	16.57%
(120 , 120)	5158	694	13.45%
(150 , 150)	5158	267	5.17%

(Table 4.9: percentage matching for the cropped image)

As the cropping area increases, the percentage matching in the pixels reduces the large amount of pixels are modified resulting in the loss of bit information thereby difficult to detect the watermark.

Image Results are shown in the figure 4.22 given below:

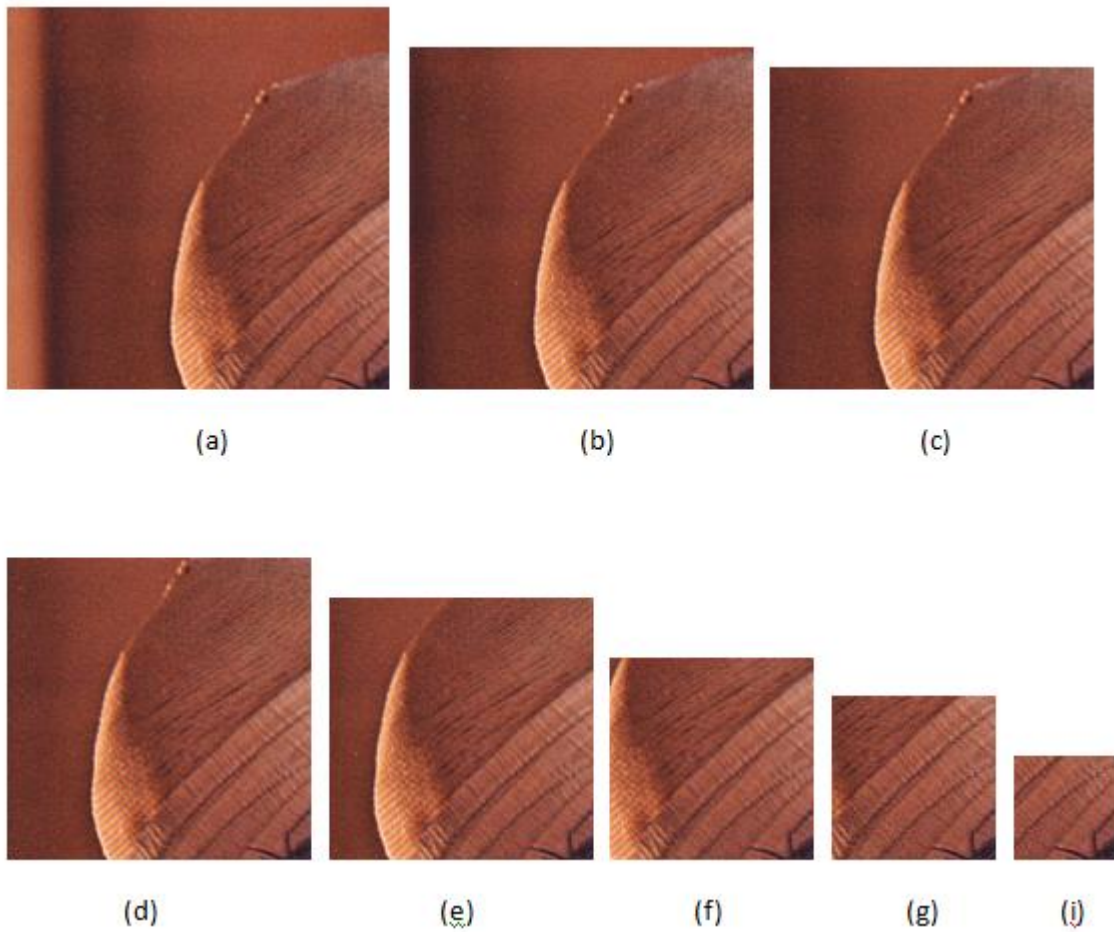


Figure 4.22: Cropped Image: (a) (10, 10) (b) (30, 30) (c) (40, 40) (d) (50, 50)

(e) (70, 70) (f) (100, 100) (g) (120, 120) (h) (150, 150)

Chapter 5

Conclusion

The current work has been focussed on extending the concept of watermarking with images. First approach dealing with **“QR codes with reversible watermarking”** 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.

Beside this I have also suggested a protocol that could be used for secure transfer of medical images containing patient information in the scheme **“QR codes with Authentication and Watermarking”**. The hash of the patient information is encrypted using the RSA encryption algorithm along with the generation of digital signature. This signed and encrypted hash is then embedded in the QR Code. This QR Code then embedded in the cover image using Coltuc et al's [] scheme. The embedding algorithm is made to work in feedback mode so that if a single 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.

Finally I have proposed an application of watermarking in the field of image processing in the scheme **“Watermarking and image processing”**. In this 1-bit information is hidden in an image based on the position parameter ' α ' and hiding parameter ' β '. ' α ' gives the row of the pixel where this bit information is hidden. And ' β ' gives the last four LSB bits where this bit is to be hidden. Then this image is modified resulting the change in the pixel value. Then percentage of matching pixels is calculated and based on that a threshold can be set in the future.

The future scope of this work can be extended in the other multimedia files like audio and video files and other files like database records. This would add to increase the robustness of the watermarking scheme in varied fields and subjects 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] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, Edgar Weippl “QR Code Security
- [3] 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.
- [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] Robert Krenn “Steganography and steganalysis”, Internet Publication, March 2004.

- [11] Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol II MECS 2008, 19-21 March, 2008, Hong Kong.
- [12] Sonal Sharma, Jitendra Singh Yadav, Prashant Sharma Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm, August 2012.
- [13] Feng Bao, Robert H. Deng, Beng Chin Ooi, Yanjiang Yang, Tailored Reversible Watermarking Schemes for Authentication of Electronic Clinical Atlas, National University of Singapore.
- [14] Sangeet Saha, Chandrajit pal, Rourab paul, Satyabrata Maity, Suman Sau A brief experience on journey through hardware developments for image processing and it's applications on Cryptography University Of Calcutta, Kolkata, India.
- [15] docs.gimp.org/en/gimp-layer-invert.html, webpage.
- [16] www.codeproject.com/Articles/33838/Gray/Image-Processing-using-C.
- [17] www.codeproject.com/Articles/33838/Contrast/Image-Processing-using-C.
- [18] www.codeproject.com/Articles/33838/Brightness/Image-Processing-using-C.
- [19] www.smokycogs.com/blog/image-processing-in-c-sharp-adjusting-the-gamma.
- [20] www.workspaces.codeproject.com/saleth-prakash/image-processing-using-matrices-in-csharp.