國立交通大學

電控工程研究所

碩士論文

抵禦影像攻擊之數位浮水印全相關技術之研究

A Study on Full Correlation of Digital Watermarks Under Different Image Attacks

研 究 生：夏拉喬

Student: Koritala Shailaja

指導教授：黃育綸　博士

Advisor: Dr. Yu-Lun Huang

中華民國九十九年一月

January, 2010

A Study on full Correlation of Digital Watermarks Under Different Image Attacks

Student: Koritala Shailaja

Advisor: Dr. Yu-Lun Huang

A Thesis

Submitted to Institute of Electrical Control Engineering

College of Electrical Engineering

National Chiao Tung University

In partial Fulfill of the Requirements for the Degree of Master

In

Institute of Electrical Control Engineering

January, 2010

Hsinchu, Taiwan, Republic of China.

A Study on full Correlation of Digital Watermarks Under Different Image Attacks

Student: Koritala Shailaja **Advisor: Dr. Yu-Lun Huang**

Department of Electrical and Control Engineering

National Chiao Tung University

Abstract

For the surveillance, home security and medical applications a digital watermarking scheme is needed to provide full correlation which is trivial in the normal techniques when any attack is predicted to occur. The correlation of decoded data varies according to the variations of length of the embedding data for the type of attack predicted. We believe that proper selection of length for the embedding data for any watermarking system can assure full correlation. In this thesis, we propose a framework which adds a preprocessor and a splitter for any regular watermark scheme at the embedder and a combiner at the decoder. A length boundary database is constructed with the congregation of various lookup tables which contain the maximum length values for various types of attacks predicted. In the framework, the preprocessor obtains the maximum length for the type of attack predicted from the length boundary database and checks this length with that of the embedding data. If the embedding data length is more than the maximum length obtained, the splitter divides it into pieces of data each of length equal to that of the maximum length chosen. The embedding data is thus embedded with the acceptable length to obtain full correlation.

Acknowledgement

I am grateful to my teacher, Dr. Yu Lun Huang, for teaching me and for her boundless patience and guidance over the course of my Masters. She has given me an incredible opportunity to study the fascinating field of watermarking, to which I could never express my gratitude enough. Without her guidance and moral support I could never achieved this height.

I would like to thank the school of Electrical and Control Engineering of the National Chiao Tung University (NCTU), for all the resources that have been available to aid me in the research. Furthermore, I would like to include in my acknowledgements all the members of the Real Time Embedded systems Lab (RTES), for their encouragement and help and for making my time in graduating memorable.

For all their love and encouragement, I would also like to acknowledge my husband Dr.Raghunath for his faith and for his cheer, my family, especially my Mom and Dad. The enormous support given by my brother, Mr.N.Sharath is unforgettable. Infinitely I would like to thank God, for listening to my worries, and giving me strength and clarity when I have needed then most.

Finally, I would like to thank the anonymous reviewers, for taking the time to review the manuscript. Their constructive and insightful comments have been of tremendous value.

TABLE OF CONTENTS

Page No.

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

List of Figures

List of Tables

CHAPTER 1

Introduction

This chapter of the thesis is organized into three subsections. In the first subsection 1.1 the background of work is given. In the section 1.2 the motivation is presented which includes our contributions and in the final section 1.3 the remaining organization of the thesis is mentioned.

1.1. Background

The idea of digital watermarking is to embed a small amount of imperceptible secret information in the multimedia so that it can be decoded later for the purposes of copyright assertion, copy control, broadcasting, authentication, content integrity verification, and so forth. For example, a string of binary bits generated, which represents the owner of an image, can be taken as a watermark embedded at the least significant bit of the pixels or transformed coefficients by adjusting their value according to a predefined algorithm. Since the secret information is embedded in the content of the media, for the applications related to copyright protection where the watermark is intended to be robust, it does not get erased when the content is manipulated or undergoes format conversions.

In general, digital watermarking has to meet the following requirements [1] ：

1. Perceptual transparency: The perceptual quality of the host signal should not be damaged by the embedding algorithm.
2. Security: The embedded information should not be able to recover by any unauthorized party without the secret key.

3. Robustness: The data embedded should survive the attacks encountered while transmitted through the internet.

The procedure of a general digital watermark system is shown in Figure 1.1 [2]. Referring to the Figure, the secret key K is used to generate the random sequence W in this case. The correlation of the recovered message is checked with the original message.

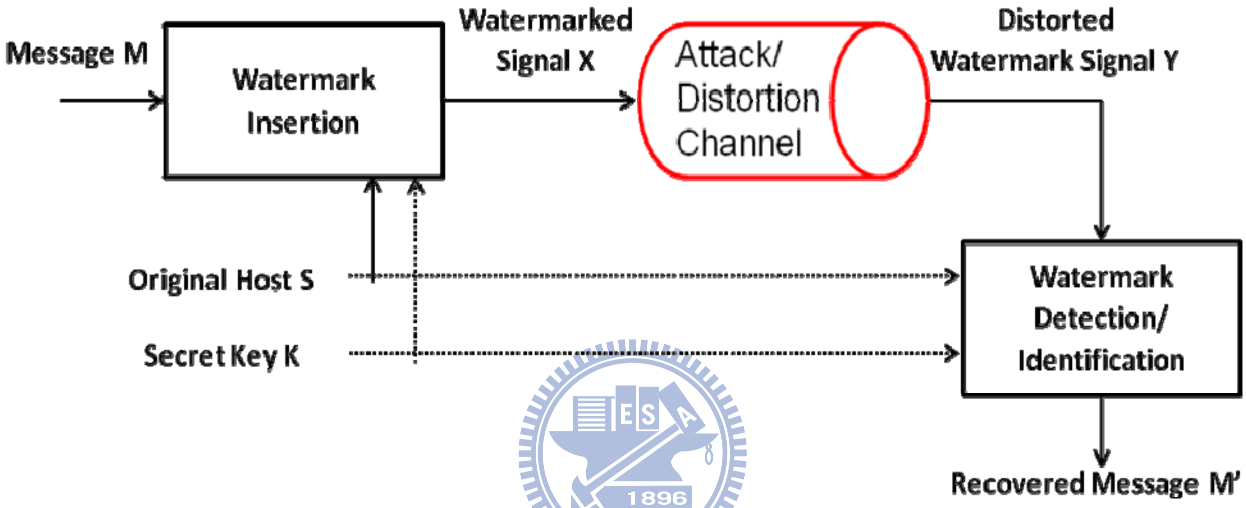


Figure 1.1: Procedure of a digital watermarking system

The applications of the watermarking include copyright protection, fingerprinting, broadcast monitoring, medical application, home security and so on. Based on the application requirements the watermarking techniques are divided into fragile, semi fragile and robust watermarking techniques. In this thesis we discuss about robust watermarking techniques specifically where there is a requirement of recovering the watermark completely. In our writing the term "watermark" is addressed as the term "embedding data". The recovery ratio of the decoded data is calculated and full correlation of the embedding data is considered to be obtained when the recovery ratio is 100%.

1.2. Motivation

Due to the rapid increase of multimedia technology, computer networks, high resolution digital cameras and the available sophisticated photo editing software's the creation and distribution of digital images has become simple. As a consequence, the necessity for the image content protection also increases. So there is a lot of increase in concern about the security and authenticity of the images. In response to these challenges, digital watermarking schemes have been proposed in the last decade. Thereby many works have been done for the purpose of protecting the information which is embedded in the image from modifying or forgery. For the applications of ID cards, home security and legal issues there is a necessity of completely recovering the embedded data. Even though various authors provide watermarking algorithms for embedding data which are robust to certain attack types, no watermarking algorithm guarantee full correlation of embedding data when any attack is predicted. Moreover as the attack dimension increases, the correlation of the decoded data decreases. The decoded data obtains full correlation only in some particular length conditions. So, the study of the maximum length conditions in which we can obtain the full correlation of data is an important issue in the applications like home security and so on. Therefore, this thesis is dedicated to the research for the digital image watermarking systems in the direction of achieving full correlation based on the maximum length conditions of the embedding data. For this purpose we proposed a framework which is the modification of the existing watermarking technique assuring full correlation of the embedding data. We constructed a length boundary database for the host image which provides the maximum length for the embedding data based on certain predicted attack types.

1.3. Synopsis

The thesis is organized in the following way. The next section gives the description of the background of different watermarking techniques. The proposed framework and its functioning details are given chapter 3. Chapter 4 contains the details of the implemented program flow both at the sender and receiver side. The details of the formation of length boundary database comprising various lookup tables are given in chapter 5. Chapter 6 depicts the case study which shows how full correlation is obtained by our framework for a particular image. Finally chapter 7 is our conclusion and future work.

CHAPTER 2

Related Work

In recent years, many methods of digital watermarking have been proposed in literature. These contributions can be divided into categories according to their processing domain of the host image, the variety of hiding position and the type of the embedding data. There are two main processing domains spatial domain [3], [4], [5] and frequency domain. With respect to still image, the most straightforward way to embed a data into an image in the spatial is to add a pseudorandom noise pattern to the luminance values of its pixels. Besides, the digital watermarking techniques are applied in transform domains, such as discrete Fourier transform (DFT) [6], discrete cosine transform (DCT) [7], [8], [9], and discrete wavelet transform (DWT) [10], [11], [12]. Different transform domain has its own advantages and disadvantages. Few works are described below. Since many applications such as medical, forensic, ID cards application require that the error rates be significantly smaller, the error control schemes are thus devised to achieve the goal. Various techniques [13], [14], [15] are used according to the requirements of different types of watermarking algorithms. But these schemes always add a lot of parity bits to the embedding data and therefore limiting the number of original data bits. And also addition of larger data may cause perceptual degradation of the image.

V. Darmstaedter et.al. [16] described a block-based embedding method. This is a multi-bit embedding scheme working in the spatial domain where each bit of the watermark is spread over one 8×8 block of the spatial image. The main purpose of the choice is that 8×8 corresponds to the size of JPEG blocks. Schyndel et al. [17] proposed a method based on bit plane manipulation of the least significant bit (LSB) which offers easy and rapid decoding. The advantages of using LSB watermarking are very simple, fast and providing high capacity. The disadvantages are that it is not robust to noise and not robust to geometrical alterations and easy to be removed. That is, the watermarked image may be easily distorted if image operation is performed. In 1997, Cox et al. developed a new algorithm of using spread spectrum to embed a watermark [18], [19]. The author believes that the watermark must be perceptually significant components of a signal if it is to be robust to common signal distortions and malicious attack. However, the modification of these components can lead to perceptually degradation of the signal. In their early papers, Cox et al inserted a length n watermark into n largest coefficients of the N × N DCT of the image corresponding to the low frequencies, excluding the DC component, to keep imperceptibility. The watermark should be constructed as an independent and identically Gaussian random vector. Although their scheme has the advantage that is robust to noise and to JPEG compression, it also has the disadvantages of decreasing capacity.

Hsu and Wu [20] embedded the watermarks with visually recognizable patterns in the images. The embedding positions were selectively modifying the middle frequency of DCT of the images. J.J.K.O'Ruanaidh and T.Pun [21] propose that Fourier transform based invariants can be used for digital image watermarking. The watermark takes the form of a two dimensional spread spectrum signal in the RST transformation invariant domain. The watermark survives lossy image compression using JPEG at normal setting (75% quality factor). The watermark is also reasonably resistant to cropping and could be recovered from a segment approximately 50% of the size of the original image.

In order to design a watermarking technique robust to a very wide range of image distortions Jiri Fridrich [22] [23] designed an algorithm which the watermark in both low and mid-frequency domain of an image. The author embeds the watermark in low frequency domain by modifying the DCT coefficients of the image and then embeds the same watermark in middle frequency domain using a spread spectrum technique using the modification scheme of Cox et al and O'Ruanaidh et al described above. Thus the benefits of spread spectrum watermarking as well as DCT domain watermarking are attained. The author also proposed various algorithms [24-27] for the application of image authentication. Thus in summary we have given some initial watermarking algorithms which are robust against various attacks given in the Table 2.1 below.

Table 2.1: Robustness of the initial algorithms

Author(s)	**Robustness**
Schyndel et al.	Not much robust to distortions
Cox et al.	Image scaling, JPEG compression, dithering, cropping
V. Darmstaedter et al.	JPEG compression
R. Dugad et al.	JPEG 2000compression
Jiri Fridrich	JPEG compression, Mosaic, Noise Addition, Permutation
X. Xia et al.	JPEG 2000 compression, Image scaling etc..
D. Kundur et al.	JPEG 2000 compression, A/D conversion etc
Hsu and Wu et al.	Cropping, JPEG compression etc.
O'Ruanaidh et al.	Rotation, Cropping, Translation etc.

As mentioned above, there are many kinds of watermarking algorithms existing which are robust against various attacks and can be used for different applications. In some identity cases where any tiny changes to the embedding data are not acceptable, it has to be compensated perfectly. For example, any minor distortion of an embedding data is not acceptable in the case of identity cards or the images in the legal issues and so on which cause a serious debate on the integrity of the embedding data. Therefore for such cases it is desirable that the watermarking schemes are capable of perfectly recovering the embedding data i.e. the decoded data is required to obtain full correlation. But the existing watermarking techniques despite of providing very efficient algorithms, yields poor results based on the variations of length of the embedding data when an attack is predicted. So in our work we have adopted one of the existing watermarking algorithms to analysis the embedding data length conditions under which we can obtain full correlation for different predicted attacks.

CHAPTER 3

Proposed Framework

In this chapter we propose a framework by which any regular watermarking scheme can obtain full correlation under some length conditions. Section 3.1 depicts the objective of the proposed framework and the section 3.2 outlines its design concepts. The detailed description of the components of the framework is given in section 3.3. Finally section 3.4 describes about the length boundary database which plays a crucial role in providing the appropriate maximum length value for the embedding data.

3.1. Objective

It is obvious that there is always a tradeoff between the length of the embedding data and the recovery ratio in any watermarking scheme. For a particular attack type predicted as the length of the embedding data increases the recovery ratio at the decoder decreases. Therefore to overcome this tradeoff we need to embed the data within the boundary of the length where we get 100% recovery ratio. If the recovery ratio is 100%, then full correlation of the embedded data is said to be obtained. So to attain full correlation the maximum length under which is we obtain 100% recovery ratio is considered as a major concern. If the embedding data is more than that length we split the message into pieces and transmit by embedding into the copies of that image with length equal to that of maximum length that can be embedded. Therefore, the major objective of our proposed framework is to obtain full correlation for any existing watermarking scheme.

3.2 Design Concepts

The figure 3.1 below shows the schematic diagram of the proposed framework where a preprocessor and a splitter is added to the watermark embedder at the sender. And at the receiver we add a combiner to the watermark decoder. A length boundary database providing maximum length for embedding data is constructed initially and given to the sender. The major functions of the components are given below.

- Preprocessor: The importance of the preprocessor is that it checks the length of the input embedding data with the maximum length given by the length boundary database corresponding to the attack type predicted and its dimension and decides whether to split the embedding data or not.

- Splitter: The splitter splits the embedding data into equal parts of length equal to the maximum length.
- Combiner: Its purpose is to combine the parts of decoded data obtained at the receiver.
- Length boundary database: It provides the maximum length for the embedding data based on the predicted attack type.

The notations of the framework are given in the Table 3.1 below and the functioning details of each component of the framework are described in the next section.

Table 3.1: Notations for the proposed framework

Notations	**Description**
I_H	Host image
M	Embedding data
A_T	Attack type
A_D	Attack dimension
LBD	Length boundary database
L_M	Length of the embedding data
L_q	Maximum length given by the lookup table
$m_1 ....m_n$	Pieces of embedding data from the splitter where $n = \text{ceil}(L_M/L_q)$
$I_1 ......I_n$	Watermarked images
$I' ...... I'_n$	Retrieved images
$m'_1 ..... m'_n$	Pieces of decoded data from the watermark decoder where $n = \text{ceil}(L_M/L_q)$
M'	Decoded data
N	Number of pieces of the embedding data

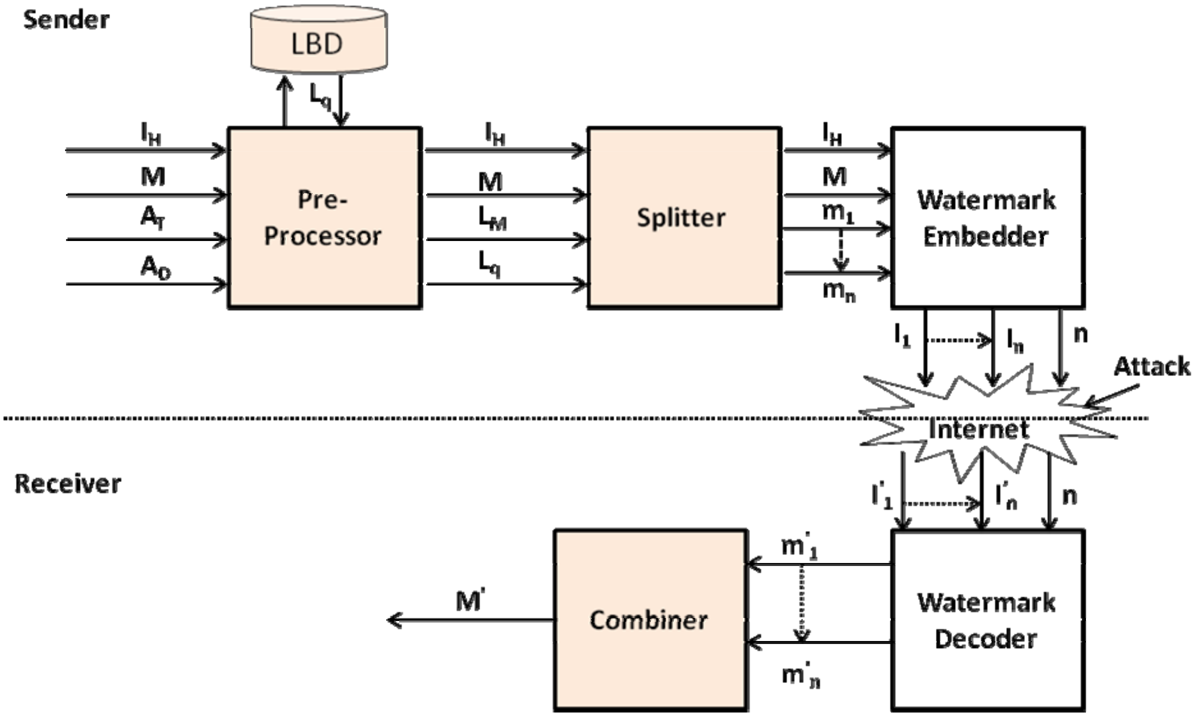


Figure 3.1. Block diagram of the proposed framework.

3.3 Components

The details of how the components of the sender and receiver are functioning are given below.

3.3.1 Sender: The components of the sender are preprocessor, splitter and the watermark embedder.

Preprocessor

According to our prediction of the attack type and its dimension, the preprocessor choose the maximum length for our embedding data from the LBD which contain number of lookup tables providing maximum length for each type of attack predicted according to its dimension. If the length of the sender's embedding data is equal or less than that of the maximum length provided, then the job of preprocessor is pass the embedding data to the watermark embedder directly or it passes the embedding data, its length, the maximum length value and the host image to the splitter. The steps for the functioning of the preprocessor can be given as following.

1. Obtains the embedded data M, attack type A_T and dimension of it A_D and host Image I_H.
2. Calculates the number of bits of the embedding data , L_M
3. According to A_T and A_D, it obtains the maximum length L_q
4. Compare L_M with L_q
5. If $L_M <= L_q$ it passes the embedding data into the watermark embedder
6. If $L_M > L_q$ it passes the embedding data to the splitter.

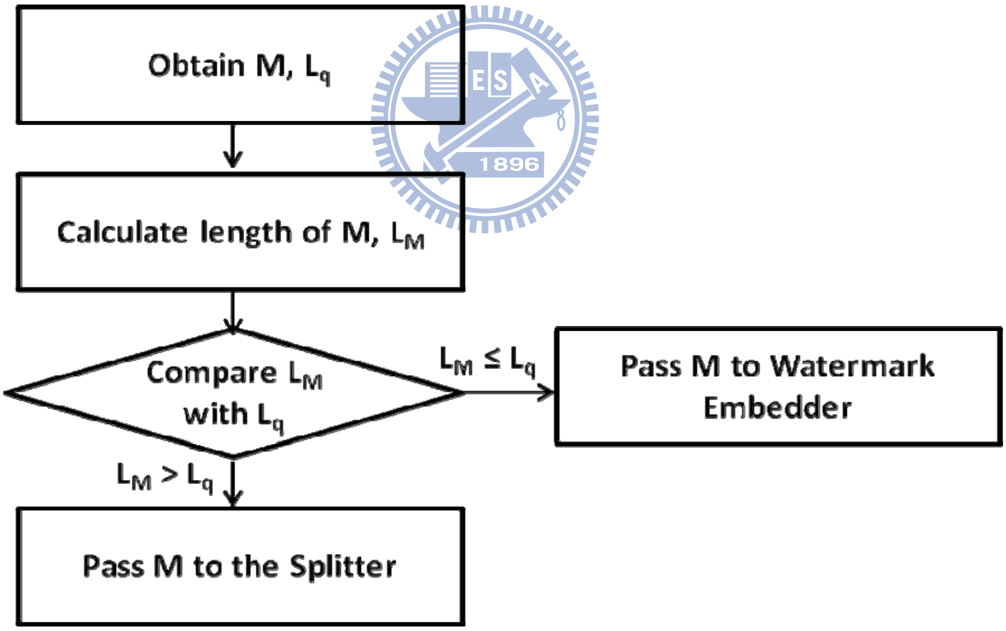


Figure 3.2. Flow chart of the preprocessor

Splitter

The splitter takes the embedding data, its length, the corresponding maximum length and the host image from the preprocessor as the inputs and divides the embedding data into ‘n’ (n = L_M/L_q) number of pieces based on that maximum length value. So we get m_i pieces, each piece containing the embedding data equal to the L_q number of bits. If there are lesser number of bits for the last piece, it will pad zeros in that piece. The algorithm for the splitter mechanism can be given as following. After this process it sends the m_i pieces and the host image to the watermark embedder.

1. Obtain the embedding data M, it length L_M and the maximum length L_q
2. Divide L_M by L_q
3. Obtain the quotient ‘n’ and ceil it to the nearest integer
4. Split the message into ‘n’ number of blocks
5. And for each block L_q numbers of bits from the embedding data are given
6. And thus obtained m_1…..m_n blocks are passed to the watermark embedder.

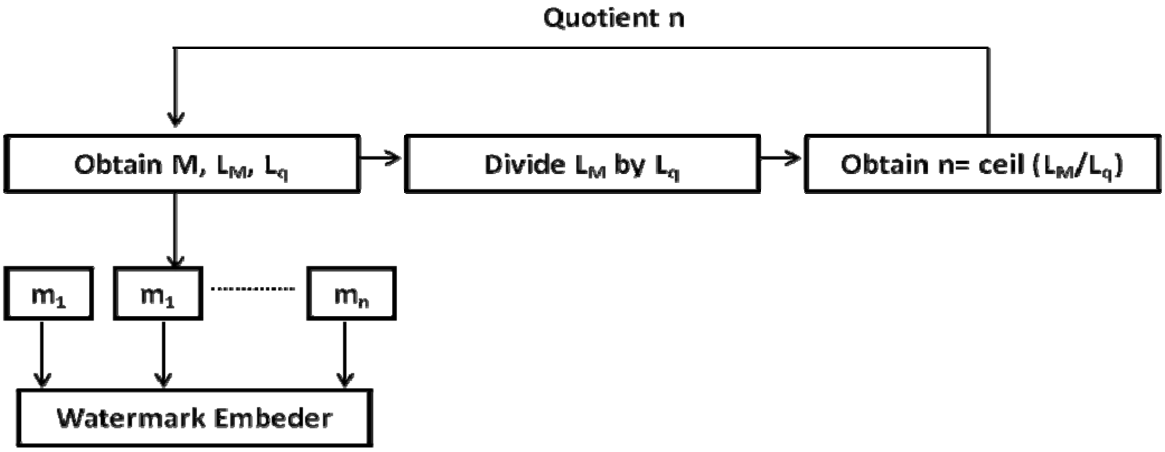


Figure 3.3. Block diagram of the splitter

Watermark Embedder

The watermark embedder either embeds the embedding data directly into the host image or the pieces of embedding data from the splitter are embedded into the copies of the host image by using the required watermarking algorithm. Any certain watermarking algorithm can be used based on which our length boundary database is constructed. Thus the embedded single/copies of image are transmitted. The number of copies 'n' of the images is sent to the receiver.

3.3.2 Receiver: At the receiver, we have the watermark decoder and a combiner.

Watermark Decoder

The embedded data is decoded at the watermark decoder by the corresponding watermark decoding algorithm and we obtain either the single embedded data or the pieces of it according to the embedding process which are further sent to the combiner.

Combiner

The combiner combines the pieces of the decoded data. We have allocated a buffer here to collect the pieces of the decoded data. Since the number of pieces of the decoded data is known to the receiver it combines 'n' number of pieces thus helps in obtaining the full correlation.

3.4 Length Boundary Database

The watermarking algorithms especially in the case of image watermarking do not provide robustness to all the types of attack. Each one, designed for various applications are expected to be robust against some specific attack types. For instance mostly DCT domain watermarking algorithms [28], [29], [30], [31] are designed especially to be robust against JPEG compression whereas many DWT domain algorithms [32], [33], [34], [35] are designed to be robust against JPEG 2000 images. The length boundary database can be formed by predicting those particular attacks. In this thesis we performed the detailed analysis on an existing robust watermarking algorithm and then studied the impact of the input length on its recovery. We have conducted a large number of experiments of embedding and decoding to obtain the maximum length of the embedding data in various conditions of the attack type and its dimensions.

For each predicted attack type and its dimension there will be a limit for the length of the embedding data beyond which we cannot get full correlation. In our writing this limit is termed as maximum length L_q. Based on this fact, maximum length that can be embedded for each predicted attack type and its dimension are found by the experimentation of the required watermarking algorithm. The figure for the procedure of the length boundary database formation is given below.

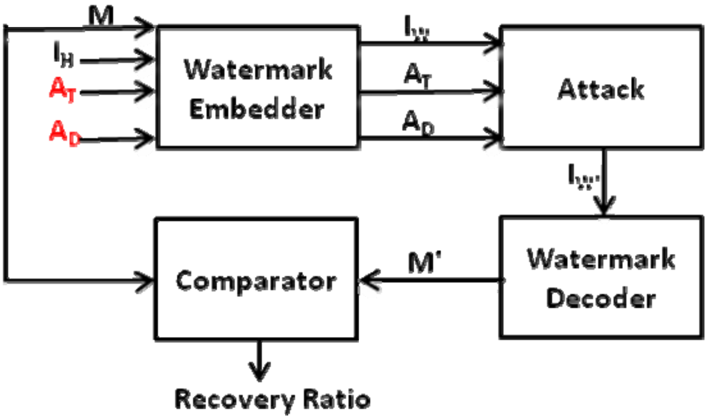


Figure 3.4: Block diagram for the construction of LBD

First we predict some attacks and the steps for constructing the length boundary database are given as

1. For each predicted attack type and its dimension we specify attack type (A_T), attack dimension (A_D), host image (I_H), emdedding data (M) and the length of the data (L_M).

2. Embed the data, distort it accordingly and decode the data.

3. Then we calculate the recovery ratio of the decoded data. The equation for the calculation is given as

$$\text{Recovery Ratio } (\%) = \frac{\text{Total number of bits recovered}}{\text{Total number of bits embedded}} \text{ x } 100\%$$

If the recovery ratio is 100% then the length of the embedding data is increased and again it is embedded, distorted and decoded. This process is continued till the recovery ratio is less than 100%.

4. The maximum length beyond which we cannot obtain 100% recovery ratio is considered as the L_q value for that particular dimension of the attack.

The maximum length values thus obtained are given in the corresponding attack type lookup tables. Thus a length boundary database is formed for the host image initially which contains various lookup tables providing maximum length for each predicted attack type and its dimension. In the case, when the details of the attack dimensions are not predicted, we have given a generalized lookup table for different types of attacks. We suggested the length in the generalized lookup table assuming that the attack dimensions as best and worst conditions denoted by delta (Δ) and (Δ'). The details of the construction with the lookup tables with examples are given in the chapter 5.

CHAPTER 4

Implementation

In this chapter we showed how we implemented our framework both from the sender and receiver side. First we have listed the software module we used in the section 4.1 and described the implemented program flow in the next section.

4.1 Software module

In our experimentation, we used Intel core 2 GHz CPU and developed it on Microsoft Visual studio 2008 on windows platform. The source codes used are open source codes which are written in C language implemented in the windows XP operating system.

For our analysis of watermark embedding and decoding we have adopted the base algorithm of Fridrich [22] whose source code is provided by the author Peter Meerwald at http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/source/. We have modified this source code by written a program in C language for the preprocessor and splitter at the watermark embedder program and for combiner at the watermark decoder program. Initially the original capacity of the image, when there is no predicted attack is found and then for various lengths of the embedding data, a noble analysis on the recovery ratio of decoded data resistant to various attacks like JPEG compression permutation, mosaic and noise addition are done. For each attack according to its dimensions we suggested the maximum length that can be embedded on the basis of our experimental analysis**.** The text message whose length starts from 8 bits is used as the embedding data by varying it according to the dimensions of various attacks.

For JPEG compression attack we have downloaded the source code given by JPEG group at ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz. For the attacks mosaic, noise addition we used Photoshop pro version 4.12 and the attack permutation is done by writing a simple program in MATLAB R2008b. We have adopted some gray-level standard images, for example, Lena.pgm, as the host image. The size of host images is 256 KB and resolution is 512 x 512. Since different images have different capacity to accept the embedding data according to its size, resolution and many internal characteristics of the image, we need to obtain the different lookup tables for different host images. For example a plane image has very less capacity than a complex image when a watermarking algorithm is using perceptually significant components of the image to embed the data. Thereby at the decoder side different percentage of recovery is shown according capacity at the embedding side. For our case study we have adopted the images which have nearer standard deviation since the watermark embedding algorithm converts the image into zero mean and some standard deviation.

In our implementation we have two programs for the sender and receiver side. For the sender we have modified the existing watermarking algorithm, by adding a preprocessor and splitter with a length boundary database to the watermark embedder at the sender side and a combiner to the watermark decoder at the receiver side. The steps of the programs are described below.

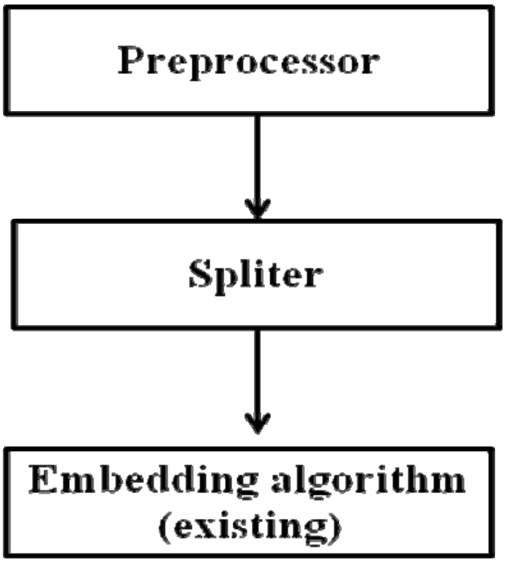
4.2. Sender Program

Firstly a length boundary database is constructed for the image which is used as the host image by running different kinds of the attacks predicted with their respective dimensions. It provides the lookup tables which have maximum length L_q values for various attack types predicted and the corresponding dimensions. Based on the senders attack type and the dimension predicted the length boundary database provides the L_q

value for the embedding data. In our implementation we have given the L_q value manually. Thus the L_q value, the embedding data and the host image in which we intend to embed the data are given as inputs to the preprocessor at the sender. The preprocessor calculates the number of bits of our embedding data L_M and compares with the L_q value. If it is less than or equal to L_q value it sends the embedding data to the watermark embedder or it sends the embedding data and the corresponding L_q to the splitter. Secondly the splitter divides the embedding data into 'n' number of pieces based on the L_M and L_q value given where $n = [L_M/L_q]$. Therefore we get 'n' numbers of pieces of our embedding data at the output of splitter. Finally the pieces of the embedding data if ($L_M > L_q$) or the single embedding data (if $L_M <= L_q$) with the host image are given as an input to the watermark embedder.

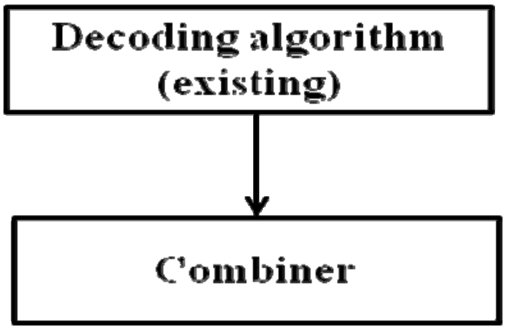
The embedding data is embedded by using the base algorithm given by the author, Fridrich by which we get the image with watermark embedded inside. The author for converts the image into zero mean and certain standard deviation which further undergoes into DCT transformation. Then the embedding data is embedded into both low frequency and middle frequency domain of the DCT transformed image. If the input is from the splitter the watermark algorithm is modified such as to take the copies of the host image to embed the pieces of the embedding data. The information of the number of copies which we get by splitting the embedding data are also sent to the receiver.

Then the embedded image is distorted according to the corresponding predicted attack type and dimension. For our experiments we have conducted four different kinds of distortions namely JPEG compression, mosaic filtering, permutation, noise addition. For JPEG compression we used the open source code as mentioned above where we get the compressed image. We performed the mosaic and noise addition attacks in Photo shop pro and implemented the permutation in Matlab. The figure below shows the flow of the program at the sender side. The value of ‘n’ which is the number of pieces of the embedding data is also sent to the receiver.



4.2. Receiver Program

At the receiver side we have taken the corresponding distorted image or copies of images and decoded the embedding data by using Fridrich decoding algorithm. The received image is again DCT transformed and the embedding data is decoded based on the secret key. Thus we get the decoded data or the pieces of it decoded which further goes to the combiner block. Here we have allocated a buffer to collect the pieces of the decoded data obtained at the receiver and also we get the value of the number of copies of the image. In the next step, based on the ‘n’ value the combiner combines the pieces of decoded embedding data and gives the output. Thus the framework which has the control on the length of the embedding data always assures full correlation. The figure below shows the flow of the program at the receiver side. The performance of the framework at various lengths of embedding data for the predicted attack types is verified in our case study described in chapter 6.



CHAPTER 5

Construction of Length Boundary Database

In this chapter we have given the examples for the procedure of finding L_q value for the four predicted attack types. The next section gives one example of finding L_q value for each predicted attack type at one of its dimension for the image Lena.pgm and thus formed lookup table is also given. In the section 5.2, the constructed example length boundary database for four standard images which have different standard deviation is given.

5.1 Lookup Tables for Different Attacks

The length boundary database comprises of the lookup tables with various L_q values for different attacks predicted. These lookup tables for the attacks predicted are formed by the experimental analysis of the watermarking algorithm as mentioned in chapter 4. Here we have shown an example of how we found the L_q value for the four attacks predicted. The lookup tables for image Lena.pgm are shown here by distorting it with the attacks JPEG compression, mosaic, noise addition and permutation which are described in detail in this section. And thus formed length boundary database for different standard images are given in the next section.

5.1.1 JPEG Compression Attack

Figure 4.1a shows the image Lena.pgm compressed at JPEG 50 quality factor. In our experiments we have calculated the recovery ratio for various lengths of embedding data at ten different compression ratios, namely Q10, Q15, Q20, Q25, Q30, Q35, Q40, Q45, Q50, Q55, Q60, Q65 and Q70 to distort the watermarked images. For each compression ratio, the maximum length of the embedding data, L_q which can be embedded are found. The results are given in lookup table 4.1. The graph below (figure 4.1b) shows the example of the recovery ratio of the embedding data with increasing its length at JPEG 50 compression for the image, Lena.pgm. We can see that for the quality factor 50, the error rate is increased gradually from the length 160bits. In this thesis for our LBD construction, we considered the length till there is no error in the recovery. So the length 160 bits is considered as the maximum length L_q for Q50. Similarly for all the other quality factors, maximum length is found to form the lookup table for JPEG compression attack which is given to the final length boundary database. The quality factor Q70 is considered as the best case of attack dimension since the perceptual quality of image is not degraded at this dimension and the Q10 is considered as the worst case of dimension since the perceptual quality is much degraded at this dimension.

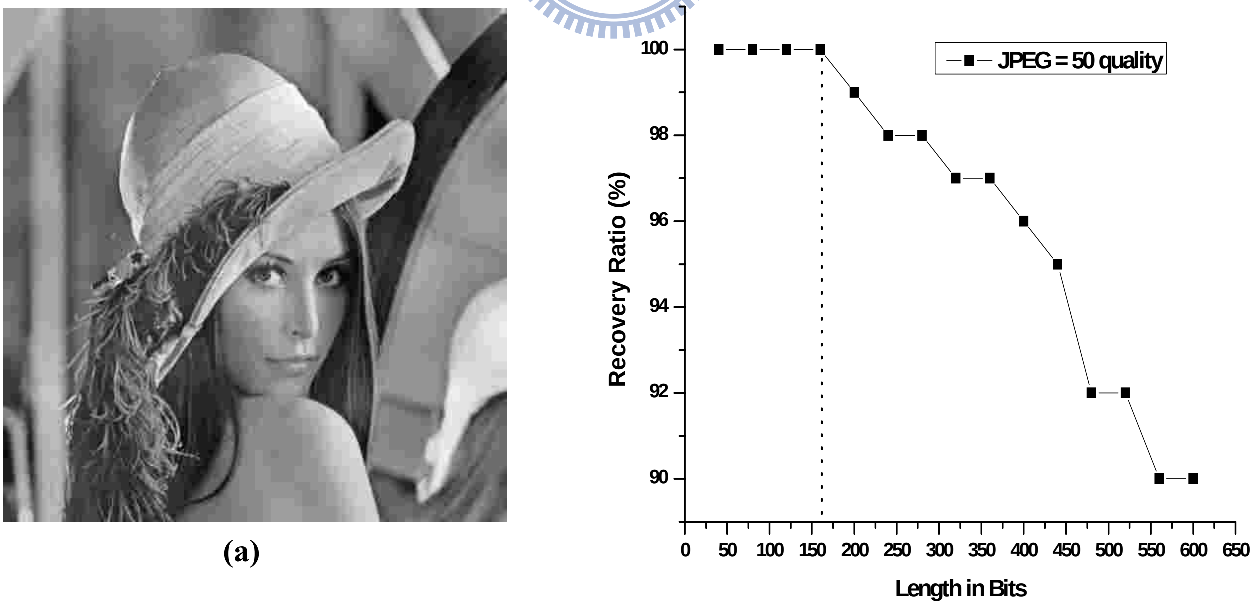


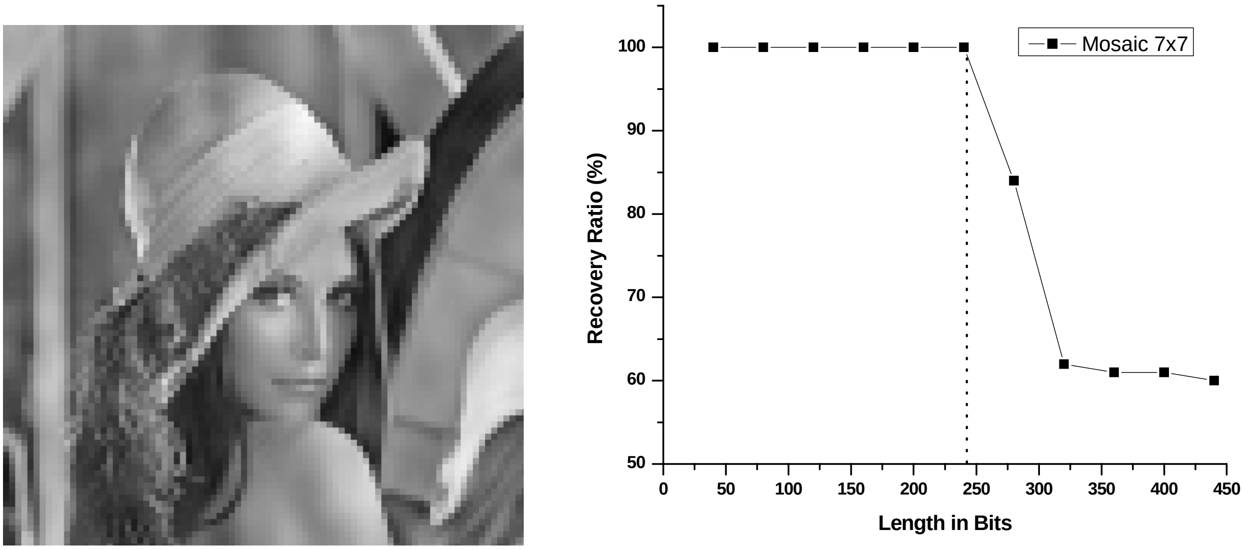
Figure 5.1. a) Watermarked image with JPEG compression quality factor 50. b) Length of embedding data vs. Recovery Ratio at quality factor 50.

Table 5.1: lookup table for JPEG compression

JPEG Quality Factor	**Maximum Length L_q (Bits)**
10	40
15	64
20	72
25	96
30	112
35	112
40	120
45	152
50	160
55	176
60	280
65	360
70	426

5.1.2 Mosaic Attack

The figure 4.1.a below is the image distorted by mosaic attack at the dimension of 7x7 and the graph below shows the recovery ratio vs length of the embedding data for the mosaic attack at the dimension 7x7. The input embedding data can be recovered completely till the length is 240 bits (see Figure 4.2.b). We performed experiments by varying the lengths at different dimensions of 10x10, 9x9, 8x8, 7x7, 6x6, 5x5 and came out with a corresponding lookup table suggesting the maximum length L_q for each dimension. 10x10 is considered as the worst dimension of mosaic filter and the dimension of 5x5 is considered as best dimension since the perceptual quality of the image is not much degraded.



(a) (b)

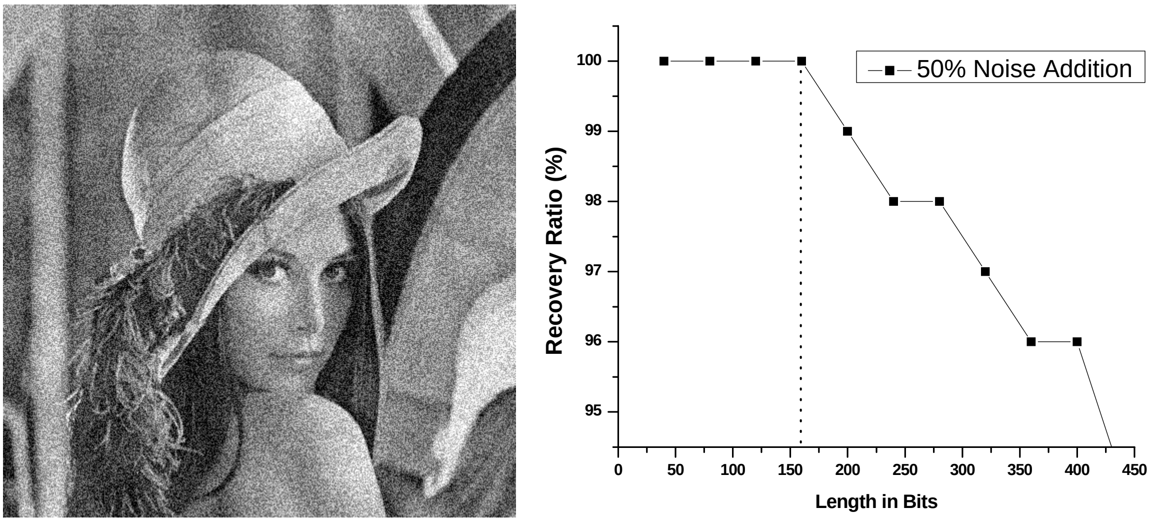
Figure 5.2 a) Watermarked image with mosaic 7 x 7. b) Length of embedding data vs. Recovery Ratio at mosaic 7 x 7.

Table 5.2: Lookup table for mosaic attack

Dimension of mosaic filter	**Maximum length L_q (Bits)**
10x10	72 bits
9 x 9	120 bits
8 x 8	160 bits
7 x 7	240 bits
6 x 6	256 bits
5 x 5	280 bits

5.1.3 Noise Addition Attack

We performed experiments of the noise addition. The figure 4.3 a shows the image distorted with noise 40. For the lookup table we performed the experiments to find the maximum length for standard deviation 40 to 100 gray levels. At the dimension of 50 the L_q value is 160 bits as shown in the graph 4.3 b.



(a) (b)

Figure 5.3. a) Watermarked image with 50% Noise Addition b) Length of embedding data vs. Recovery Ratio at 50% Noise Addition.

Table 5.3: Lookup table for Noise Addition attack

Noise Addition (%)	Maximum Length L_q(Bits)
100	40
90	48
80	56
70	80
60	88
50	160
40	264

5.1.4 Permutation Attack

We have written a simple program in which the image was again divided into squares and the pixels in those squares were randomly permuted. For 5x5 squares dimension our 'L_q' value in the lookup table is 120 bits. In the case of permutation attack, we did experiments on the dimensions 10 x10 to 4 x 4 squares. Similar to noise addition we could add very less length of embedding data for the worst case of dimension.

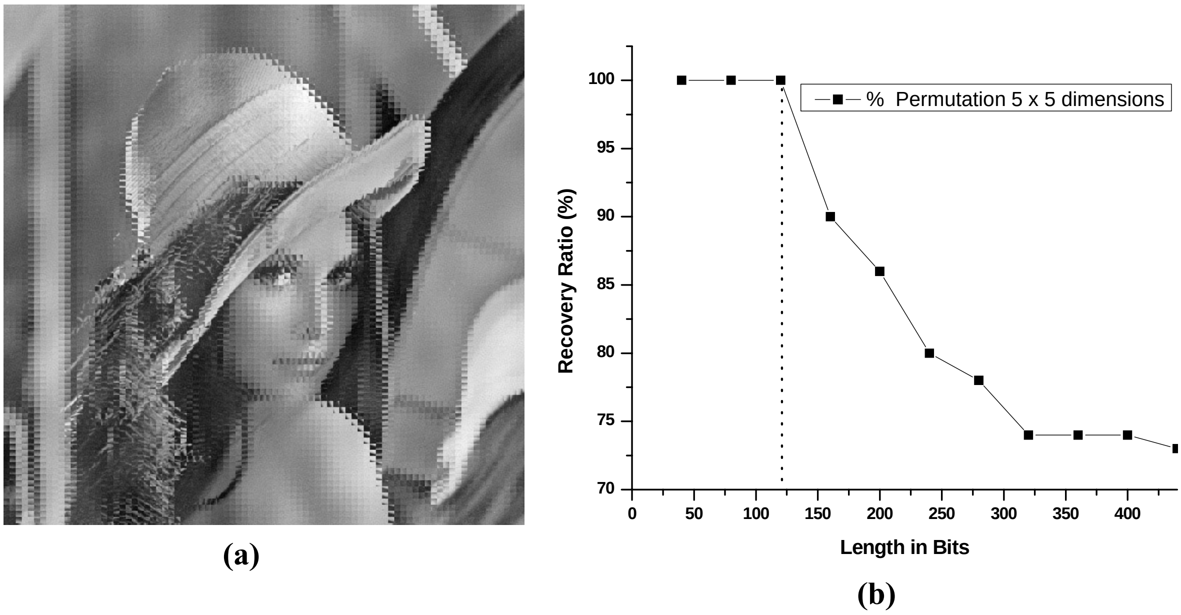


Figure 5.4. a) Watermarked image with permutation 5 x 5. b) Length of embedding data vs. Recovery Ratio at permutation 5 x 5

Table 5.4: lookup table for permutation attack

Dimension of permutation	**Maximum length L_q (Bits)**
10 x 10	64
7 x7	88
6 x 6	104
5 x 5	120
4 x 4	152

5.1.5 Generalized lookup table for Lena.pgm

Based on the above analysis of lengths for various attacks we propose the generalized lookup table for the image Lena.pgm which gives the maximum length boundary delta Δ in case where the dimension of attack is predicted as the minimum and maximum dimension. We give the delta values for best case as well as the worst case of attack dimensions represented as Δ and Δ'. Delta is actually the value of L_q for the minimum attack dimension .i.e. if we predict that the attack is occurred at least dimension and similarly we predict Δ' as L_q value for the worst case of attack dimension. . If length $L_q > \Delta$ we cannot guarantee the full correlation.

Table 5. 5: Generalized lookup table

Attack type	Maximum Length (Δ) (Best case)	Maximum Length (Δ') (Worst case)
JPEG Comp.	426 bits	40 bits
Mosaic	320 bits	88 bits
Noise Addition	264 bits	40 bits
Permutation	152 bits	64 bits

5.2 Example Length Boundary Database

Thus the lookup tables formed in the above section are congregated together to form the length boundary database of that image. In this thesis, we created some examples of length boundary database with lookup tables for different images, including lena.pgm, pirate.pgm, flight.pgm, and pepper.pgm as shown in Figure 5.5. The watermarking algorithm we used converts the image to zero mean and certain standard deviation before embedding. So the images with nearer standard deviation values are chosen for the purpose of observing the L_q values for images with similar standard deviation. The standard deviation values for the four chosen images are given in the Table 5.6 below. Before finding the L_q value for the distorted image, we have found the original capacity of the four images when there is no attack predicted given in the Table 5.7. The maximum length that can be embedded in the images is quite larger than that of the distorted image. The value decreases much more as the dimension of the attack increases.

The length boundary database of each image consists of lookup tables for four predicted attack types. A generalized lookup table is also given according to the best and worst cases of the attack dimension. The lookup tables should provide maximum length value to the preprocessor in our framework. Thus for the preprocessor we can form the length boundary database for various images with all the lookup tables for the respective predicted attack types. Similar length boundary database can be formed for the any images required with their lookup tables.

Table 5.6: Standard deviation values of different images

Image	**Standard deviation**
Lena	47.85
Pirate	47.64
Flight	45.12
Pepper	57.40

Table 5.7: Maximum length for different images without attack

Image	**Maximum Length L_q (Bits)**
Lena	2368
Pirate	1640
Flight	1984
Pepper	2112

Lena.pgm

Pirate.pgm

Flight.pgm

Pepper.pgm

Figure 5.5. Host images for which length boundary database is constructed.

Table 5.8: LBD with lookup tables of the image Lena.pgm

Single Attack Lookup Tables								**Generalized Lookup Table**		
JPEG Comp.		**Mosaic**		**Noise Addition**		**Permutation**				
Quality Factor	**L_q(bits)**	**Dimension**	**L_q(bits)**	**%**	**L_q(bits)**	**Dimension**	**L_q(bits)**	**Attack Type**	**Δ(bits)**	**Δ'(bits)**
10	40	10 x 10	88	100	40	10 x 10	64	JPEG Comp.	426	40
15	64	9 x 9	152	90	48	7 x7	88	Mosaic	320	88
20	72	8 x 8	160	80	56	6 x 6	104	Noise Addition	264	40
25	96	7 x 7	240	70	80	5 x 5	120	Permutation	152	64
30	112	6 x 6	256	60	88	4 x 4	152			
35	112	5 x 5	320	50	160					
40	120			40	264					
45	152									
50	160									
55	176									
60	280									
65	360									
70	426									

Table 5.9: LBD with lookup tables of the image Flight.pgm

Single Attack Lookup Tables								**Generalized Lookup Table**		
JPEG Comp.		**Mosaic**		**Noise Addition**		**Permutation**				
Quality Factor	**L_q(bits)**	**Dimension**	**L_q(bits)**	**%**	**L_q(bits)**	**Dimension**	**L_q(bits)**	**Attack Type**	**Δ(bits)**	**Δ'(bits)**
10	16	10 x 10	96	100	40	10 x 10	40	JPEG Comp.	312	16
15	32	9 x 9	104	90	48	9 x 9	72	Mosaic	280	96
20	64	8 x 8	128	80	56	8 x 8	80	Noise Addition	208	40
25	80	6 x 6	176	70	80	5 x 5	104	Permutation	104	40
30	88	5 x 5	280	60	104					
35	152			50	128					
40	168			40	208					
45	184									
50	216									
55	232									
70	312									

Table 5.10: LBD with lookup tables of the image Pirate.pgm

Single Attack Lookup Tables								**Generalized Lookup Table**		
JPEG Comp.		**Mosaic**		**Noise Addition**		**Permutation**				
Quality Factor	**L_q(bits)**	**Dimension**	**L_q(bits)**	**%**	**L_q(bits)**	**Dimension**	**L_q(bits)**	**Attack Type**	**Δ(bits)**	**Δ'(bits)**
10	24	10 x 10	88	100	40	10 x 10	24	JPEG Comp.	448	24
15	56	8 x 8	168	90	48	5 x 5	56	Mosaic	288	88
25	88	6 x 6	272	80	56	4 x 4	280	Noise Addition	192	40
35	104	5 x 5	288	70	72			Permutation	280	24
40	208			60	112					
45	272			40	192					
50	288									
55	312									
70	448									

Table 5.11: LBD with lookup tables of the image Pepper.pgm

Single Attack Lookup Tables								**Generalized Lookup Table**		
JPEG Comp.		**Mosaic**		**Noise Addition**		**Permutation**				
Quality Factor	**L_q(bits)**	**Dimension**	**L_q(bits)**	**%**	**L_q(bits)**	**Dimension**	**L_q(bits)**	**Attack Type**	**Δ(bits)**	**Δ'(bits)**
10	40	10 x 10	24	100	48	10 x10	24	JPEG Comp.	632	40
25	64	8 x 8	32	90	56	6 x 6	48	Mosaic	88	24
30	72	7 x 7	88	80	80	5 x 5	80	Noise Addition	232	48
35	96			70	104			Permutation	80	24
40	112			60	144					
45	152			50	216					
50	192			40	232					
55	224									
60	352									
65	536									
70	632									

The tables 5.8, 5.9, 5.10, 5.11 given above are the length boundary database for the images Lena.pgm, Flight.pgm, Pirate.pgm and Pepper.pgm respectively. Each LBD has four lookup tables for the four predicted attack types and its corresponding dimensions. The left most highlighted part of LBD is the generalized lookup table which has the maximum length value for the best case and worst case of the predicted attack type. The original capacity of the images is much higher whereas the maximum length that can be embedded starts decreasing to a maximum extent when an attack predicted. For example the capacity of the image Lena.pgm is 2368 bits original without any attack but its capacity decreases to 426 bits for JPEG compression attack at its best dimension and even to 40 bits at its worst dimension. So the study of maximum length value that can be embedded is quite crucial for the applications which need full correlation.

CHAPTER 6

Case Study

In our case study we checked the performance of our framework for similar images. We took the two images Lena.pgm and Pirate.pgm which have the nearer standard deviation. The LBD of Pirate.pgm was adopted for the host image Lena.pgm and the LBD of Lena.pgm was given for the host image Pirate.pgm. The recovery ratio for at various lengths of embedding data is calculated. We have taken the length from 40 bits to 160 bits and recovery ratios at these lengths for the four predicted attack types are calculated. The Table 6.1 below shows the Δ' value from the corresponding generalized lookup table of the two images for reference. The maximum length that can be embedded for the worst dimension of the attack type for the image Pirate.pgm is less than or equal to that of the image Lena.pgm.

Table 6.1: Generalized lookup table of Pirate.pgm and Lena.pgm.

Attack Type	**Pirate.pgm** **Δ'(bits)**	**Lena.pgm** **Δ'(bits)**
JPEG Comp.	24	40
Mosaic	88	88
Noise Addition	40	40
Permutation	24	64

The correct selection of LBD plays a key role in the recovery ratio of the embedding data. Tables 6.2, 6.3, 6.4 and 6.5 below show the recovery ratio of the embedding data with its length varying from 40 bits to 160 bits for the image Lena.pgm distorted by JPEG compression, mosaic filter, noise addition and permutation respectively. The embedding process for the image Lena.pgm is done by taking the

LBD of Pirate.pgm and vice versa for the image Pirate.pgm. The Table 6.2 and 6.5 shows the recovery ratio against the attacks JPEG compression and permutation respectively. The LBD of the image Pirate.pgm yielded 100% recovery ratio whereas the LBD of the Lena.pgm gave decreased recovery ratio because the capacity of the embedding data for Lena.pgm is more than that of the image Pirate.pgm. By observing the Tables 6.3 and 6.4 in the case of mosaic and noise addition attacks we could achieve full correlation in both the cases since the L_q value at that dimension is same for both the images. If the LBD of Pirate.pgm is chosen then full correlation is attained for all the four attacks. But the LBD of Lena.pgm does not give full correlation for some attack types. So for watermarking similar kind of images with similar standard deviation the LBD of the image whose capacity is lesser can be chosen which helps in obtaining full correlation.

Table 6.2: Recovery Ratio vs JPEG Compression attack

	I_H=Lena.pgm/LBD=Pirate.pgm			**I_H=Pirate.pgm/LBD=Lena.pgm**		
A_D **L_M**	**Q10**	**Q25**	**Q35**	**Q10**	**Q25**	**Q35**
40	100	100	100	98	100	100
80	100	100	100	98	100	100
120	100	100	100	98	98	99
160	100	100	100	98	98	99

Table 6.3: Recovery Ratio vs Mosiac attack

.	**I_H=Lena.pgm/LBD=Pirate.pgm**		**I_H=Pirate.pgm/LBD=Lena.pgm**	
A_D **L_M**	**10 x 10**	**8 x 8**	**10 x 10**	**8 x 8**
L_M = 40	100	100	100	100
L_M = 80	100	100	100	100
L_M =120	100	100	100	100
L_M =160	100	100	100	100

Table 6.4: Recovery Ratio vs Noise Addition attack

.	**I_H=Lena.pgm/LBD=Pirate.pgm**		**I_H=Pirate.pgm/LBD=Lena.pgm**	
A_D **L_M**	**100%**	**80%**	**100%**	**80%**
L_M = 40	100	100	100	100
L_M = 80	100	100	100	100
L_M =120	100	100	100	100
L_M =160	100	100	100	100

Table 6.5: Recovery Ratio vs Permutation attack

.	**I_H=Lena.pgm/LBD=Pirate.pgm**		**I_H=Pirate.pgm/LBD=Lena.pgm**	
A_D **L_M**	**10 x 10**	**5 x 5**	**10 x 10**	**5 x 5**
L_M = 40	100	100	98	100
L_M = 80	100	100	98	90
L_M =120	100	100	98	80
L_M =160	100	100	97	79

So, according to the tables, we can guarantee a recovery ratio of 100% if we choose proper LBD for similar kind images. In summary the LBD with smaller L_q values is more appropriate for obtaining full correlation.

CHAPTER 7

Conclusions and Future Work

In this thesis, we designed a framework which guarantees the full correlation for an existing digital watermarking system. We performed a detailed study on the maximum length conditions under which full correlation of the embedding data is assured. Based on the systematic analysis of the existing watermarking algorithm, we built various lookup tables suggesting maximum length for different kinds of attacks. A preprocessor and a splitter are added to the watermark embedder and a combiner is added to the watermark decoder. For each predicted attack and dimension, the embedding data of length chosen less than or equal to this value always assures full correlation.

Construction of the précised length boundary database is difficult because we need to consider the attack types predicted carefully. Moreover the lookup tables are provided predicting the single attack type and its dimension. A multi attack prediction based design of lookup tables are considered for future work. Similarly for different types of host images the lookup tables will be different. In our future, we intend to study the relation between the recovery ratios of different types of images.

References

[1] F. P. Gonzalez and J. R. Hern´andez, “A tutorial on digital watermarking”, Tech. Rep., University of Vigo, Spain. 1998.

[2] S. Decker, “Engineering Considerations in Commercial Watermarking”, IEEE Communications Magazine, Volume 39, Issue: 8, pp. 128 -133, 2001.

[3] S. P. Maity and M. K. Kundu, “Robust and blind spatial watermarking in digital image”, Tech. Rep., Dept. of Electronics and Telecomm., India, 2001.

[4] N. Nikolaidis and I. Pitas, “Copyright protection of images using robust digital signatures”, in Proc. IEEE ICASSP’96, pp. 2168–2171, 1996.

[5] R. B. Wolfgang, and E. J. Delp, “A watermark for digital images”, Tech. Rep., School of Electrical Engineering, Purdue University, USA, 1995.

[6] F. Alturki, R. Mersereau, “Secure blind image steganographic technique using discrete Fourier transformation”, Image Processing, Volume 2, pp. 542 -545, 2001.

[7] J. R. Hernandez, M. Amado, F. Perez-Gonzalez, “DCT-domain watermarking techniques for still images: detector performance analysis and a new structure”, Image Processing, IEEE Transactions on, Volume 9, Issue 1, pp. 55 -68, 2000.

[8] G. C. Langelaar, R. L. Lagendijk, “Optimal differential energy watermarking of DCT encoded images and video”, Image Processing, IEEE Transactions on, Volume 10, Issue 1, pp. 148 -158, 2001.

[9] S. Katzenbeisser, F. A. P. Petitcolas (eds.), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, 2000.

[10] X. Xia, C.G. Boncelet and G. R. Arce, Wavelet transform based watermark for digital images, Optics Express, Volume 3, No. 12, 1998.

[11] R. Dugad, K. Ratakonda, and N. Ahuja, A new wavelet-based scheme for watermarking images, Proceedings of the IEEE International Conference on Image Processing, ICIP 1998, Chicago, IL, USA, October 1998.

[12] H.-J. M. Wang, P.-C. Su, and C.-C. J. Kuo, "Wavelet-based digital image watermarking," Opt. Express, Volume 3, no. 12, pp. 491–496, 1998.

[13] H. Liu, H. Ma, M. El Zarki, and S. Gupta, "Error control schemes for networks: An overview," ACM Mobile Networking and applications (MONET), volume 2, no. 2, pp. 167-182, 1997.

[14] J. Lee and C. S. Won, "A watermarking sequence using parities of error control coding fro image authentication and correction," IEEE Transactions on Consumer Electronics, Volume 46, No.2, 313, 2000.

[15] N. Terzjia, M. Repges, K. Luck, W. Geisselhardt, Digital Image Watermarking Using Discrete Wavelet Transform: Performance Comparison of Error Correction Codes, published in Proceedings of IASTED 2002, September 2002.

[16] V. Darmstaedter, J.F. Delaigle, J.J. Quisquater, B. Macq, Low cost spatial watermarking, Comput. Graphics volume 22, 417-424, 1998.

[17] R. G. Van Schyndel, A. Z. Tirkel, C. F. Osborne, "A digital watermark", Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, and Volume 2, pp.86 -90, 1994.

[18] I. J. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for images, audio and video", Image Processing, Proceedings, International Conference on, Volume 3, pp.243 -246, 1996.

[19] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia", Image Processing, IEEE Transactions on, Volume 6, Issue 12, pp.1673-1687, 1997.

[20] C-T Hsu, J-LWu "Hidden digital watermarks in images," IEEE Trans. Image Processing, volume 8, pp. 58–68, 1999.

[21] J. J. K. Ó Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking", Proc. of the ICIP, Santa Barbara, California, volume 1, pp. 536–539, 1997.

[22] J. Fridrich, "Combining low-frequency and spread spectrum watermarking", Proc. SPIE Mathematics of Data/Image Coding, Compression, and Encryption, volume 3456, pp. 2−12, 1998.

[23] J. Fridrich, M. Goljan, Comparing Robustness of Watermarking Techniques, Proc. of SPIE (Security and Watermarking of Multimedia Content), Volume 3657, pp. 214-225, 1999.

[24] J. Fridrich, Methods for Tamper Detection in Digital Images, Proc. ACM Workshop on Multimedia and Security, Orlando, FL, pp. 19-23, 1999.

[25] J. Fridrich, M. Goljan, A.C. Baldoza, "New fragile authentication watermark for images," Image Processing, 2000. Proceedings. 2000 International Conference on, volume.1, pp.446-449, 2000.

[26] J. Fridrich, M. Miroslav, and N. Memon, "Cryptanalysis of the Yeung–Mintzer fragile watermarking technique", J. Electron. Imaging. Volume 11, pp. 262, 2002.

[27] J. Fridrich, "Security of Fragile Authentication Watermarks with Localization", Proc. SPIE Photonic West, Security and Watermarking of Multimedia Contents IV, Volume. 4675, pp. 691-700, 2002.

[28] A. G. Bors, I. Pitas, "Image watermarking using DCT domain constraints," Image Processing, 1996. Proceedings. International Conference on, volume 3, pp.231-234, 1996.

[29] M. Barni, F. Bartolini, V. Cappellini, A. Piva, “A DCT-domain system for robust image watermarking,” Signal Processing, Volume 66, pp. 357-372, 1998.

[30] M. D. Swanson, M. Kobayashi, A.H. Tewfik, “Multimedia data-embedding and watermarking technologies,” Proceedings of the IEEE, Volume 86, pp. 1064 –1087, 1998.

[31] P. Yu, and B. Liu, “Public Watermarking Algorithm Based on the Polarity of DCT Coefficients,” In Proceedings of the Sixth international Conference on intelligent Systems Design and Applications - Volume 02, pp. 277-282, 2006.

[32] Yong-Seok Seo; Min-Su Kim; Ha-Joong Park; Ho-Youl Jung; Hyun-Yeol Chung; Young Huh; Jae-Duck Lee, "A secure watermarking for JPEG-2000," Image Processing, 2001. Proceedings. 2001 International Conference,, volume 2, pp.530-533, 2001.

[33] S. Tsekeridou, I. Pitas, "Wavelet-based self-similar watermarking for still images," Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium, volume 1, pp.220-223, 2000.

[34] D. Kundur and D. Hatzinakos, “Digital watermarking using multi resolution wavelet decomposition,” Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing, Volume 5, pp. 2969-2972, 1998.

[35] Young-Sik Kim; O-Hyung Kwon; Rae-Hong Park, "Wavelet based watermarking method for digital images using the human visual system," Electronics Letters, volume 35, pp.466-468, 1999.