

# 國立交通大學

多媒體工程研究所

碩士論文

在 JPEG XR 影像上做資訊隱藏之新研究

A New Study on Information Hiding via JPEG XR Images

研究生：姚志憲

指導教授：蔡文祥 教授

中華民國九十九年六月

在 JPEG XR 影像上做資訊隱藏之新研究

A New Study on Information Hiding via JPEG XR Images


研究生：姚志憲

Student : Chih-Hsien Yao

指導教授：蔡文祥

Advisor : Wen-Hsiang Tsai

國立交通大學  
多媒體工程研究所  
碩士論文



A Thesis  
Submitted to Institute of Multimedia Engineering  
College of Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in  
Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

# **A New Study on Information Hiding via JPEG XR Images**

Student: Chih-Hsien Yao

Advisor: Wen-Hsiang Tsai

Institute of Multimedia Engineering  
National Chiao Tung University

## **ABSTRACT**

With the convenience of the Internet, information exchanges through the Internet become very frequent nowadays. The JPEG XR image is a new image file format which supports high compression ratios and becomes more and more popular in information communication. In this study, three methods for data hiding applications via JPEG XR images are proposed, including covert communication, image authentication, and copyright protection.

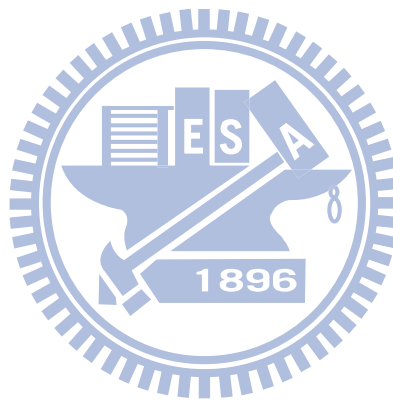
For covert communication, a data hiding method via JPEG XR images for use in covert communication is proposed, which encodes selected quantization parameters of highpass frequency bands to embed secret messages. In addition, a technique to generate a stego-image with the original cover image file size being kept for reducing the possibility of arousing notice from attackers is also proposed.

For image authentication, a block-based authentication method based on embedding DC\_LP coefficients as authentication signals into the highpass coefficients of the image blocks is proposed, which can be employed to verify the integrity and fidelity of JPEG XR images by comparing the extracted authentication signals with

those computed from the current image content.

For copyright protection, a removable visible watermarking method based on modifying the DC\_LP coefficients in JPEG XR images by certain number division operations is proposed. Specifically, the remainders of dividing the DC\_LP coefficients by a pre-selected integer number are embedded into the highpass coefficients of image blocks as watermark signals which can be losslessly removed if necessary.

Measures to enhance the security of each of the proposed methods are also suggested. Good experimental results show the feasibility of the proposed methods.





# CONTENTS

<b>ABSTRACT (in English)</b> .....	<b>i</b>
<b>CONTENTS</b> .....	<b>iii</b>
<b>LIST OF FIGURES</b> .....	<b>v</b>
<b>LIST OF TABLES</b> .....	<b>viii</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
1.1 Motivation .....	1
1.2 General Review of Related Works.....	2
1.3 Overview of Proposed Methods .....	3
1.3.1 Terminologies.....	3
1.3.2 Brief Descriptions of Proposed Methods.....	4
1.4 Contributions .....	7
1.5 Thesis Organization.....	8
<b>Chapter 2 Review of Related Works and JPEG XR Standard</b> .....	<b>9</b>
2.1 Review of Techniques for Image Data Hiding .....	9
2.2 Review of Techniques for Image Authentication.....	10
2.3 Review of Techniques for Visible Watermarking in Image.....	11
2.4 Review of JPEG XR Standard.....	12
2.4.1 Structure of JPEG XR standard.....	12
2.4.2 Process of encoding of JPEG XR file.....	16
2.4.3 Process of decoding of JPEG XR file.....	18
<b>Chapter 3 Covert Communication via JPEG XR Images by Variable     Macroblock Quantization</b> .....	<b>21</b>
3.1 Introduction .....	21
3.1.1 Problem definition.....	22
3.1.2 Major idea of proposed method .....	22
3.2 Proposed Method for Covert Communication.....	24
3.2.1 Information hiding by variable macroblock quantization .....	24
3.2.2 Algorithm for data embedding .....	29
3.2.3 Algorithm for data extraction.....	35
3.2.4 Resuming original file size .....	36
3.3 Security Consideration .....	38
3.3.1 Issues of security of proposed method .....	38
3.3.2 Proposed security enhancement measures.....	39

3.4	Experimental Results .....	39
3.5	Summary .....	40
<b>Chapter 4</b>	<b>JPEG XR Image Authentication by Comparison of DC and Low-pass Frequency Coefficients .....</b>	<b>46</b>
4.1	Introduction .....	46
4.1.1	Problem definition .....	46
4.1.2	Major idea of proposed method .....	47
4.2	Proposed Method for Image Authentication.....	48
4.2.1	Generation of authentication signals .....	48
4.2.2	Embedding of authentication signals.....	57
4.2.3	Extraction of authentication signals .....	58
4.3	Security Consideration .....	61
4.3.1	Issues of security of proposed method .....	61
4.3.2	Proposed security enhancement measures .....	61
4.4	Experimental Results .....	62
4.5	Summary .....	62
<b>Chapter 5</b>	<b>Copyright Protection by Removable Visible Watermarking Based on DC Parameter Manipulation in Frequency Domain .....</b>	<b>68</b>
5.1	Introduction .....	68
5.1.1	Problem definition .....	68
5.1.2	Major idea of proposed method .....	69
5.2	Proposed Method for Removable Visible Watermarking.....	70
5.2.1	Removable Visible Watermarking Based on DC Parameter Manipulation.....	70
5.2.2	Process of embedding visible watermark .....	73
5.2.3	Process of removing visible watermark.....	77
5.3	Security Consideration .....	78
5.3.1	Issues of security of proposed method .....	78
5.3.2	Proposed security enhancement measures .....	78
5.4	Experimental Results .....	79
5.5	Summary .....	79
<b>Chapter 6</b>	<b>Conclusions and Suggestions for Future Works .....</b>	<b>84</b>
6.1	Conclusions .....	84
6.2	Suggestions for Future Works.....	85
	<b>References</b>	<b>87</b>

# LIST OF FIGURES

Figure 2.1 The structure of image planes. (a) Primary image plane. (b) Alpha image plane.....	13
Figure 2.2 The hierarchy structure of JPEG XR images. The size of internal image is equal to that of the original image. The width and height are multiple of 16. ....	14
Figure 2.3 The components of a macroblock.....	14
Figure 2.4 The codestream modes of the JPEG XR standard: the spatial mode and the frequency mode. ....	15
Figure 2.5 Block diagram of JPEG XR format encoding process. ....	18
Figure 2.6 Block diagram of JPEG XR format decoding process. ....	20
Figure 3.1 The flowchart of data embedding and extraction via the un-decoded JPEG XR file format. ....	25
Figure 3.2 The original image used for compression distortion test, for which the applied quantization value is 15. In this study, we call this image “Two_dogs.” .....	27
Figure 3.3 The scatter plot with smooth lines of relationship between compression ratios and PSNRs. The quantization value of the original image is 15. The quantization parameter is used for compression. ....	28
Figure 3.4 The flowchart of the proposed data embedding process.....	33
Figure 3.5 The flowchart of the proposed data extraction process. ....	34
Figure 3.6 Inserting the padding data between the two tile packets. ....	36
Figure 3.7 The process of resuming to the original size of the cover image. ....	38
Figure 3.8 The stego-image which hides a message of 1534 bytes. All the macroblocks are used for hiding the message. And the PSNR is 51.4811. 41	
Figure 3.9 The stego-image which hides the information “I love dogs!! dogs love me~”.....	41
Figure 3.10 User interface for data extraction using the right secret key.....	42
Figure 3.11 User interface for data extraction using a wrong secret key. ....	42
Figure 3.12 The stego-image which hides the information “I love dogs!! dogs love me~”, and keeps the original file size of the Two_dogs image with PSNR 46.9041.....	43
Figure 3.13 A stego-image whose file size is kept to be the original one of the cover image. (a) The size of the “Two_dogs” image. (b) The size of Figure 3.6.43	
Figure 3.14 The stego-image in which all the macroblocks are used for hiding the message. And the PSNR is 52.324. ....	44
Figure 3.15 The stego-image in which all the macroblocks are used for hiding the	

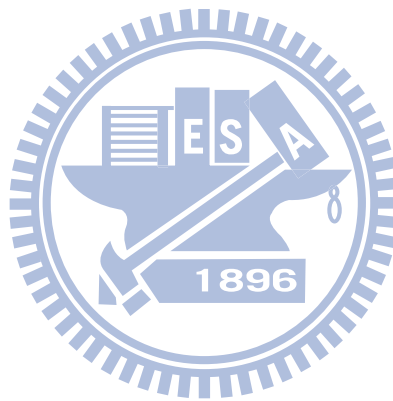
message. And the PSNR is 53.2166.....	44
Figure 4.1 The two level FCT transformation. ....	49
Figure 4.2 The two level IFCT transformation. ....	49
Figure 4.3 The flowchart of authentication signals embedding and extraction via the un-decoded JPEG XR file format. ....	49
Figure 4.4 The brief flowchart of authentication signals embedding process. ....	50
Figure 4.5 The brief flowchart of authentication signals extraction process.....	50
Figure 4.6 The overflowing pixels value truncated by re-encoding. ....	52
Figure 4.7 An example of overlap filtering straddling a 64-pixel block. ....	52
Figure 4.8 The tampered area will be expanded after the overlap filtering.....	53
Figure 4.9 When the overlap filtering is used, the tampered area will be map to the small one. ....	53
Figure 4.10 When a block has been tampered with, the tampered area will be expanded to other blocks of the macroblock after the re-encoding.....	54
Figure 4.11 The flowchart of the proposed authentication signals embedding process .....	55
Figure 4.12 The flowchart of the proposed authentication process. ....	56
Figure 4.13 A simple example of replacing the second LSB's. ....	62
Figure 4.14 A test to avoid the authentication signals destroying by overflowed pixel value after re-encoding the stego-image. (a) The cover image. (b) The authentication result without restricting the range of pixel value. (c) The authentication result by using the threshold $m$ to restrict the range of pixel value.....	63
Figure 4.15 The authentication signals is embedded into the image “Two_dogs” and different ways are used for authentication. (a) The cover image “Two_dogs.” (b) A stego-image into which the authentication signals are embedded with PSNR 39.3165. (c) A tampered image of (b). (d) The authenticated image without using the re-mapping algorithm when overlap filtering is used. (e) The authenticated image without using the extra authentication signals. (f) The final authenticated image. ....	64
Figure 4.16 A test of the authentication signals survival rate by hiding authentication signals into different last significant bits. (a) A stego-image by hiding authentication signals into LSB (289k bytes). (b) A stego-image by hiding authentication signals into the second LSB (347k bytes). (c) The compressed image of (a) with 124k bytes. (d) The compressed image of (b) with 220k bytes. (e) The authenticated image of (c). (f) The authenticated image of (d). ....	65
Figure 4.17 More authentication test for another image. (a) A cover image. (b) A	

stego-image into which the authentication signals are embedded with PSNR 38.6288. (c) A tampered images of (b). (d) The final authenticated image.....	66
Figure 4.18 More authentication test for a third image. (a) A cover image. (b) A stego-image into which the authentication signals are embedded with PSNR 39.0006. (c) A tampered images of (b). (d) The authenticated image. ....	67
Figure 5.1 A simple example for proposed data hiding method. ....	72
Figure 5.2 The flowchart of visible watermark insertion and removing via the un-decoded JPEG XR file format.....	72
Figure 5.3 The proposed method for watermarking and hiding data into the LSB of highpass coefficient. ....	73
Figure 5.4 The flowchart of the proposed visible watermark insertion process.....	74
Figure 5.5 The flowchart of the proposed visible watermark removing process.....	75
Figure 5.6 A watermark used in this study.....	80
Figure 5.7 The watermark inserted into “Two_dogs” and will be lossless removed later by a secret key. (a) The cover image “Two_dogs.” (b) A watermarked image. (c) A recovered image. (d) An image with the watermark removed incompletely with a wrong key.....	80
Figure 5.8 The watermarked images with different watermarking parameters. (a) The used threshold $m = 2$ . (b) The used threshold $m = 3$ . (c) The used threshold $m = 2$ and the cover image re-encoded without using the overlap filtering. (d) The threshold $m = 3$ and the cover image re-encoded without using the overlap filtering. (e) A part of (b). (f) A part of (d).....	81
Figure 5.9 The watermark was inserted into another image and losslessly removed later by a secret key. (a) The cover image. (b) A watermarked image. (c) A recovered image. (d) An image with the watermark removed incompletely with a wrong key.....	82
Figure 5.10 The watermark was inserted into a third image via V channel rather than Y channel and losslessly removed later by a secret key. (a) The cover image. (b) A watermarked image. (c) A recovered image.....	83

# LIST OF TABLES

Table 3.1 The relationship between compression ratios and PSNRs. The quantization value of the original image is 15. QP is the quantization parameter used for compression. QP = 0 means that the quantization parameter value 1 when encoded by the JPEG XR standard..... 28

Table 3.2 The PSNRs of the ten selected stego-images and their average. .... 45



# Chapter 1

## Introduction

### 1.1 Motivation

With the rapid development of the network technology, more and more people transmit multimedia through the Internet. Information hiding so becomes necessary for people to protect secret messages or multimedia from being illicitly tampered with when they are transmitted on the Internet. In the mean time, the size of compressed multimedia is also becoming smaller and smaller for faster transmission. Many new image formats have been proposed for specifying compressed images with higher qualities and lower data volumes. JPEG XR is one of the still-image standards, based on technologies originally developed and patented by Microsoft as part of the Windows Media family. On June 19, 2009, the JPEG XR format was passed by the ISO/IEC Final Draft International Standard (FDIS) ballot, resulting in the final approval of it as an international standard.

The JPEG XR file format supports higher compression ratios in comparison to the old JPEG format. The JPEG XR standard also supports lossless compression. Because of the efficiency and good quality yielded by the JPEG XR, it becomes more and more popular recently. However, studies on information hiding via the JPEG XR image are not found yet. It is desirable to develop new data hiding techniques via the JPEG XR image for various information hiding applications in this study.

Data hiding techniques can be used to hide *secret* data invisibly in a *cover image*, resulting in a *stego-image*. In this way, the stego-image with the secret data may be

transmitted through the network imperceptibly. Malicious people will be unable to know the secret data hidden in the stego-image and so will not try to extract the secret data. In this way, the secret data are protected.

Images communicated on the Internet might be tampered with by malicious people, as mentioned previously. When a receiver gets a tampered image, he/she will be misled into trusting wrong information. To distinguish whether the original image has been tampered with or not, one approach is to hide authentication signals into the original image without causing noticeable distortion. If the original image is modified, the authentication signals will be damaged, and a receiver then can determine that the fidelity or/and integrity of the received image has been destroyed. For these reasons, it is desired also in this study to design JPEG XR image authentication techniques which embed authentication signals into JPEG XR images for fidelity or/and integrity verification, thus achieving the aim of security protection.

With the concept of copyright protection discussed by more and more people, how to protect images from being illegally reproduced or utilized has become a hot topic nowadays. Lots of copyright protection techniques have been proposed. One of them is to embed a digital visible watermark into a cover image. Only the owner can extract the watermark to prove the copyright of the image in case the image is illicitly utilized. It is so desirable as well in this study to develop a reversible visible watermarking technique for the purpose of image copyright protection.

## **1.2 General Review of Related Works**

Since digital rights management has become more and more important recently, many works related to data hiding, authentication, and watermarking about images have been proposed. For data hiding via images, many techniques such as histogram



shifting, human visual model, etc. have been proposed. For image authentication, a lot of methods, such as authentication signal hiding, analysis of coefficient relationships, etc., have been proposed. And for image watermarking, techniques such as visible watermarking, invisible watermarking, and so on, have been proposed. A detailed review of all of these techniques will be given in Chapter 2. In addition, because the proposed techniques in this study are applied to JPEG XR images, we will also make a review of the JPEG XR standard there.

## 1.3 Overview of Proposed Methods

### 1.3.1 Terminologies

Before describing the proposed methods briefly, some definitions of terms used in this study are introduced first as follows.

1. *Secret*: a secret is a piece of important information that should be protected properly and not be revealed to unauthorized people.
2. *Stego-image*: a stego-image is an image into which some digital information is embedded.
3. *Cover image*: a cover image is one which is used for data hiding applications and steganography.
4. *Watermarked image*: a watermarked image is one into which a digital watermark is embedded.
5. *Recovered image*: a recovered image is one obtained by removing the embedded watermark from a watermarked image.
6. *Embedding process*: an embedding process is a process to embed data in a cover image.
7. *Extraction process*: an extraction process is a process to extract the

embedded data from a stego-image.

8. *Authentication process*: an authentication process is a process to decide whether a stego-image is tampered with or not.
9. *Authenticated image*: an authenticated image is one in which the tampered blocks are marked after checking whether the embedded authentication signals are destroyed or not.

## 1.3.2 Brief Descriptions of Proposed Methods

### A. Proposed Data Hiding Method for JPEG XR Images

A data hiding technique for covert communication by using the property of the JPEG XR standard is proposed in this study. The JPEG XR standard provides variable quantization parameters for image quantization, which permits the maximum amount of quantization parameters to be 16 in each macroblock. It is a good idea to use the index of the selected quantization parameter for data hiding.

In this study, we implement this idea via the use of the highpass bands of the frequency domain of a JPEG XR image, because an image changed by modifying the highpass coefficients slightly will not result in perceptible effects.

At first, we increase the number of quantization parameters of the highpass bands to be 16. Then, we convert the secret message into 4-bit segments as new quantization parameter indexes and select accordingly the corresponding quantization parameter in each macroblock. In this way, we can hide 4-bit secret data into every macroblock. After the above steps, we de-quantize the coefficients of the highpass bands by the original quantization value, and re-quantize them by a new quantization parameter which we select for reducing the perceptual difference. With these steps implemented, we can extract secret data more easily by decoding the quantization

parameter index in each macroblock. The details will be described in Chapter 3.

## **B. Proposed Authentication Method for JPEG XR Images**

An image authentication technique for verifying the integrity and fidelity of JPEG XR images by adding authentication signals into the cover image is proposed in this study. The method hides an authentication signal into every block for small-area authentication.

In the JPEG XR standard, every block has either a DC coefficient or a lowpass coefficient (i.e., has just one of the two; cannot have both --- a case quite deferent from that of the old JPEG standard). We take the DC coefficient or the lowpass coefficient of each block as an authentication signal for the block, which then is hidden into the highpass coefficients in each block. When the image is tampered with, the DC/lowpass coefficients of tampered blocks will be altered, and the authentication signals which have been embedded into the highpass coefficients of the tampered blocks will also change. We can compare the authentication signal with the DC/lowpass coefficient in each block to decide whether the blocks of the image have been tampered with or not.

The proposed method modifies the least significant bit (LSB) of a highpass coefficient to embed one bit of the authentication signal in each block. However, replacing the LSB of a highpass coefficient by a bit of the authentication signal directly is not a good idea, since a smart attacker may modify any bit other than the LSB of a highpass coefficient to avoid destroying the authentication signal. That is, if the attacker may change the look of the image by tampering appropriately with the highpass coefficients of the image, then the tampered image will undesirably pass the authentication!

To solve this problem, we select a bit  $h_1$  other than the LSB of a highpass

coefficient by a secret key and replace the LSB by the result of  $h_1 \oplus a_i$ , where  $a_i$  is the  $i$ -th bit of the authentication signal. As a result, even if an attacker tampers with the image by modifying those highpass coefficient bits other than the LSB, we still can authenticate the fidelity or/and integrity of the image.

The main idea behind this method is hiding low-frequency values into the high-frequency domain. In this way, every frequency bands are under protection; any change in each frequency band will be detected. The details will be described in Chapter 4.

### **C. Proposed Removable Visible Watermarking Method for JPEG XR Images**

In this study, a method for copyright protection of JPEG XR images using a new removable visible watermarking technique is proposed. Because modifying low frequency domain changes the luminance of an image, we can insert accordingly a visible watermark into an image by changing the luminance of the corresponding watermarked area to make the area look different from its neighborhood (or even from the entire image). The technique of removable visible watermarking we proposed is so based on modifying the DC coefficients and lowpass coefficients by certain division operations.

At first, we de-quantize each coefficient of the frequency domain by their corresponding quantization value, and reset all of the quantization values to 1 in each macroblock. After this step, all coefficients are de-quantized. Later, calling the block into which the watermark will be inserted *watermarking block*, we divide the DC/lowpass coefficient of the watermarking block by a constant value to insert the watermark, and hide the remainder into the highpass coefficients of the watermarking block for the purpose of removing the watermark when necessary. In order to

losslessly remove the watermark, we quantize the highpass coefficients by their corresponding quantization values, and left-shift the highpass coefficients by one bit, and then embed the remainder and the original highpass quantization values in the LSB's of the highpass coefficients.

When the owner of the image wants to remove the watermark, this can be easily achieved by extracting the remainder and the original highpass quantization value of each macroblock from the LSB's of the highpass coefficients, and then composing them to recover the original coefficient values. The details will be described in Chapter 6.

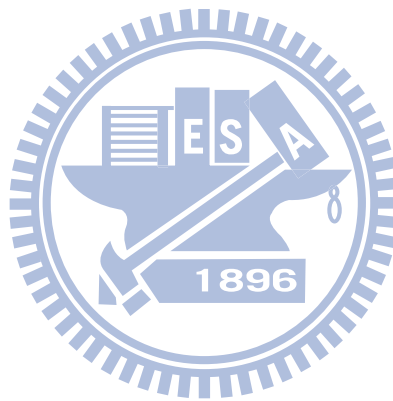
## 1.4 Contributions

The contributions made in this study are summarized in the following.

1. A data hiding method based on some properties of the JPEG XR standard is proposed for covert communication via JPEG XR images.
2. A data hiding method which maintains the original file size of the cover image is proposed, reducing the possibility of arousing notice from attackers of the communication.
3. An authentication method for verification of the integrity or/and fidelity of JPEG XR images by comparison of DC coefficients and low-pass coefficients is proposed.
4. A removable visible watermarking method based on modifying the DC coefficients and lowpass coefficients by the division operation is proposed to protect the copyright of JPEG XR images.

## 1.5 Thesis Organization

In the remainder of this thesis, a review of related works about techniques of data hiding via images, image authentication, and visible watermarking of images, as well as a brief introduction to the JPEG XR standard are given in Chapter 2. In Chapter 3, the proposed method for data hiding for covert communication via JPEG XR images is described. In Chapter 4, the proposed method for JPEG XR image authentication is described. In Chapter 5, the proposed method for removable visible watermarking of JPEG XR images is described. Finally, conclusions and some suggestions for future works are made in Chapter 6.



# Chapter 2

## Review of Related Works and JPEG XR Standard

Different types of multimedia are transmitted on the network. A lot of data hiding techniques have been proposed for multimedia copyright protection or covert communication based on the properties of distinct multimedia formats. They will be reviewed in this chapter. Because the data hiding techniques and applications we propose in this study are implemented on the JPEG XR format, we will also give a brief introduction to the JPEG XR standard in this chapter.

### 2.1 Review of Techniques for Image Data Hiding

Many data hiding methods have been proposed [2-13] in the last decade. Wu and Tsai [2] proposed a new steganographic method for images by pixel-value differencing. They changed the difference between two consecutive pixel-values to embed secret data. Two pixels with a small difference value mean the existence of a smooth area and those with a large difference value mean the existence of a sharp one. According to human vision's sensitivity, they embedded more secret data in the sharp area than in the smooth area for the purpose of yielding less distortion and higher quality. Lee and Tsai [3] proposed a lossless data hiding method by histogram shifting based on an adaptive block division scheme. As shown by experimental data, they could embed more secret data and have better PSNRs than other histogram shifting

methods. Huang and Tsai [4] proposed a new data hiding method via H.264/AVC videos based on the use of optional intra-prediction modes in the H.264/AVC format. Chang et al. [5] proposed a steganographic method via the JPEG image based upon quantization table modification. Jain and Gupta [6] proposed a JPEG compression resistant steganography scheme for raster graphics images. Barton [7] compresses the secret message before embedding them into the bit stream of digital data. Celik et al. [8] proposed a lossless data hiding method which quantizes each image pixel by into a number of scales, compresses the quantization residues, and embeds the secret bits as well as the compressed data into the quantified image by the least-significant-bit (LSB) substitution technique. Tian [9] proposed a technique of pixel-value difference expansion by performing fundamental arithmetic operations on pairs of pixels to discover hidable space. Ni, et al. [10] proposed a reversible data hiding method which shifts slightly the part of the histogram between the maximum point (also called the peak point) and the minimum point to the right side by one pixel value to create an empty bin besides the maximum point for hiding an input message. Fallahpour and Sedaaghi [11] proposed the idea of decomposing the entire cover image into blocks and using the peak point of the histogram of each block to hide data.

## **2.2 Review of Techniques for Image Authentication**

Authentication is a data hiding application for verifying the integrity and fidelity of an image. A lot of authentication methods have been proposed [14-16] in recent years. Yang and Tsai [14] proposed a block-based authentication method via PNG images by adjusting selected values in the spatial domain. They adjusted the sum of coefficients of  $3 \times 3$  blocks to a multiple of a previously-selected value as an



authentication signal, and checked the authentication signal by extracting the remainder from the sum divided by the previously-selected value. If the remainder does not equal zero, the image will be decided to have been tampered with. Huang and Tsai [15] proposed a block-based authentication method for grayscale images by embedding invisible authentication signals in them according to the human visual model. And the standard deviation values of  $3 \times 3$  blocks are used to classify each block into one of four quantization values, from smooth areas to edge ones. After classifying the quantization value of each block, the range of the grayscale is partitioned into multiple levels by the quantization value of each block. Let  $L$  be a level of the grayscale range which includes the value of the central pixel and has a lower bound value  $g_{\min}$ . Finally, the authentication signal is embedded by replacing the value of the central pixel of each  $3 \times 3$  block with  $g_{\min} + \gamma$ , where  $\gamma$  is a pre-selected constant. Lee and Tsai [16] proposed a new authentication method for software programs based on the use of invisible ASCII control codes as authentication signals.

## 2.3 Review of Techniques for Visible Watermarking in Image

Watermarking is widely used for copyright protection. A large number of watermarking methods have been proposed [17-19] in the past. Chiu and Tsai [17] proposed a method for copyright protection by watermarking for color images against print-and-scan operations using coding and synchronization of peak locations in the discrete Fourier transform domain. At first, the cover image is scaled to be a square one, and then a new coordinate system based on radiuses and angles is decided in the frequency domain. The positions  $P(R_i, \theta_j)$  and their symmetric positions  $Q(R_i, \theta_j)$  are used to embed the watermark, where  $R_1 \leq R_i \leq R_2$ ,  $0^\circ \leq \theta_j \leq 180^\circ$  ( $R_1$  and  $R_2$  are

pre-selected radiuses). Because  $P(R_i, \theta_j)$  and  $Q(R_i, \theta_j)$  are located in the middle frequency band, a threshold for watermark extraction can be selected. In other words, a pair of values of  $P(R_i, \theta_j)$  and  $P'(R_i, \theta_j)$  is replaced with a number large than the threshold for embedding the value 1. In addition, they also selected a synchronization peak  $P_{\text{sync}}(R_{\text{sync}}, \theta_{\text{sync}})$  for protection against rotation and scaling attacks, where  $R_2 \leq R_{\text{sync}}$ , and  $\theta_{\text{sync}}$  is a pre-selected angle value. Liu and Tsai [18] proposed a new method for generic lossless visible color watermarking based on reversible one-to-one compound mapping. Chen and Tsai [19] proposed a method for copyright protection of palette images by a robust lossless visible watermarking based on the use of color palette tables.

## 2.4 Review of JPEG XR Standard

In this study, all the data hiding techniques are implemented on JPEG XR images. The detailed JPEG XR specification is described in the ISO/IEC 29199-2 document [1]. We will give a brief review of the JPEG XR standard in this section. In Section 2.4.1, the structure of the JPEG XR standard will be described. The encoding and decoding process will be described later in Section 2.4.2 and Section 2.4.3, respectively.

### 2.4.1 Structure of JPEG XR standard

An image is composed of a primary image plane and an optional alpha image plane according to the JPEG XR standard, as seen in Figure 2.1. The primary image plane may have multiple image channels. The first channel is defined to be a luma component. Other channels are defined to be the chroma components. The alpha image plane contains exactly one channel which controls the weight of primary image plane that people can see. The structure of the image plane is shown in Figure 2.1.

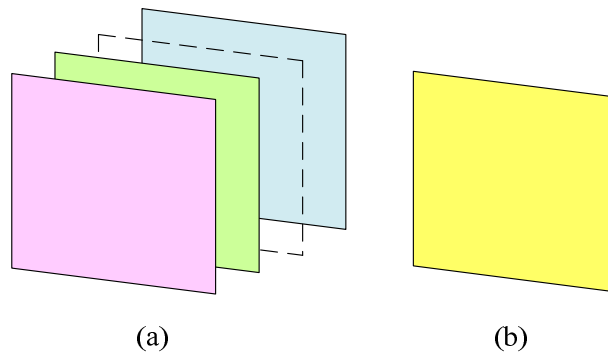


Figure 2.1 The structure of image planes. (a) Primary image plane. (b) Alpha image plane.

Every channel is composed of four bands in the frequency domain: the DC band, the lowpass band, as well as the highpass and flexbits bands. The DC and lowpass bands stand for information of a low frequency domain. The highpass band stand for information of a high frequency domain. The flexbits band carries information regarding the low order bits of the highpass coefficients.

The JPEG XR standard defines a hierarchy-level structure which includes the image, tile, macroblock, and block levels. The basic unit in the hierarchy structure is a block which is an area of  $4 \times 4$  pixels. A macroblock is the most important unit in the JPEG XR standard, which is a  $16 \times 16$  area consisting of 16 blocks. All the operations of coefficient conversion are implemented in macroblocks, such as the coefficient transformation of a space domain into a frequency domain. A tile is one of the result of a partition of the image into rectangular arrays of macroblocks. Every tile is an independent part in the JPEG XR standard. The tiles will not influence each other in the coding process. The hierarchy structure of the JPEG XR image is shown in Figure 2.2.

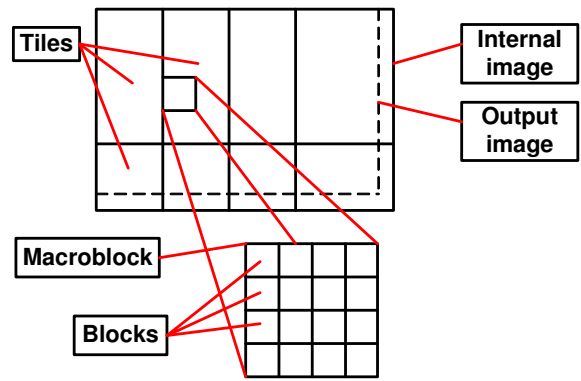


Figure 2.2 The hierarchy structure of JPEG XR images. The size of internal image is equal to that of the original image. The width and height are multiple of 16.

Because the macroblock is so important in the JPEG XR standard, we give a brief review here. In the frequency domain, a single macroblock contains 256 transform coefficients. The value of the left-top coefficient in a macroblock is called the DC value. The value of the left-top coefficient in each block, if not the DC value, is called the lowpass value. The remaining coefficients are called highpass coefficients. There are one DC coefficient, 15 lowpass coefficients, and 240 highpass coefficients in a macroblock. The components of a macroblock are shown in Figure 2.3.

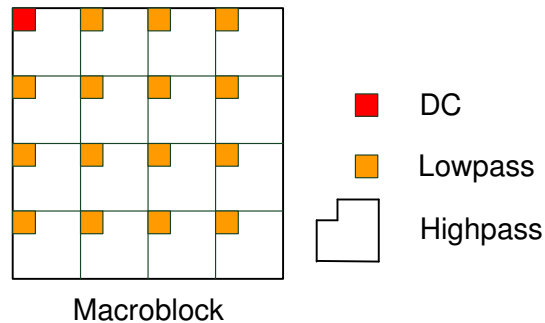


Figure 2.3 The components of a macroblock.

The JPEG XR standard defines two codestream modes: the spatial mode and the frequency mode. In both modes, the metadata and information of a JPEG XR image are coded and put in an image header. An index\_table behind the header indicates the start position of every tile packet, and the index\_table is followed by a sequence bit streams of tile packets. Every tile packet carries the information of a tile.

In the spatial mode, the coding order of the macroblocks in a tile is a raster-scan order from the left to the right and from the top to the bottom. The bitstreams of all macroblocks are combined together.

In the frequency mode, the codestream of each tile is composed of four packets: the DC, lowpass, highpass, and flexbits packets. Each packet carries the coefficients of one frequency band of that tile. The DC packet carries the information of the DC band of each macroblock, in a raster-scan order. The lowpass packet carries the information of the lowpass band of each macroblock. The highpass packet carries information of the highpass band of each macroblock. Finally, the flexbits packet carries the information of the flexbits band of each macroblock. The structure of the codestream according to the JPEG XR standard is shown in Figure 2.4.

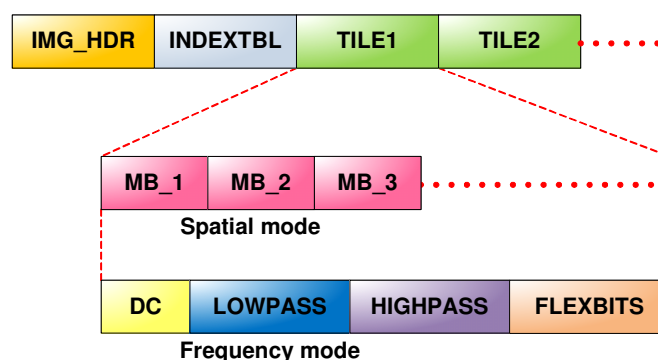


Figure 2.4 The codestream modes of the JPEG XR standard: the spatial mode and the frequency mode.

## 2.4.2 Process of encoding of JPEG XR file

In this section, we will give a sample description of the encoding process of JPEG XR image. The steps of encoding are as follows.

- i. Pre-scaling.
- ii. Color conversion.
- iii. Macroblock alignment and padding.
- iv. Transformation.
- v. Coefficient prediction.
- vi. Quantization.
- vii. Coefficient scanning.
- viii. Entropy coding.

The pre-scaling step is usually used when the input data item is greater than 27 or 24 bits. In this case, the input data will be right-shifted by  $m$  bits and reduced to 27 or 24 bits or below. When the input data item is unscaled, the maximum size of the input data is 24 bits. And the 27 bit limit is applied when the data item is scaled.

The color conversion step is used to convert OUTPUT\_COLOR\_FORMAT to INTERNAL\_COLOR\_FORMAT. In this study, we focus on the color conversion of the RGB model to the YUV. The function of converting the RGB model to the YUV is shown as follows:

$$\begin{aligned} V &= B - R; \\ t &= R - G + \left\lfloor \frac{V}{2} \right\rfloor; \\ Y &= G + \left\lfloor \frac{t}{2} \right\rfloor; \\ U &= -t. \end{aligned} \tag{2.1}$$

The size of the image is not always a multiple of 16. When an image width or height is not a multiple of 16, a macroblock alignment and padding step is conducted to extend the right column and bottom row of the image to the nearest higher multiple of 16. Then, the encoder pads the aligned width with horizontal samples, and pads the aligned height with vertical samples.

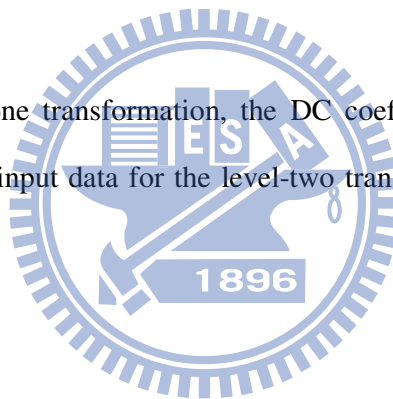
The encoder uses a two-level structure to transform the spatial domain into the frequency domain. The transformation process is shown as follows.

(1) The level-one transformation is applied to all coefficients of  $4 \times 4$  blocks of an image, which includes the following operations:

- outer pre-filtering;
- outer FCT.

(2) After the level-one transformation, the DC coefficients of  $4 \times 4$  blocks are grouped together as new input data for the level-two transformation, which includes the following operations:

- inner pre-filtering;
- inner FCT.



The pre-filtering operation is used to smooth coefficients, which is optionally applied to  $4 \times 4$  areas evenly straddling blocks in two dimensions. On the boundary, the pre-filtering operation is applied to  $4 \times 2$  or  $2 \times 4$  areas. And the pre-filtering operation does not work in the four  $2 \times 2$  corners.

The FCT operation is applied to  $4 \times 4$  blocks to transform the spatial domain into the frequency domain, whose function is like the DCT operation. The quantization step quantizes the transform coefficients to an integer value by division and rounding. Finally, the encoder converts the transform coefficients to a codestream. A flowchart of the JPEG XR encoding process is shown in Figure 2.5.

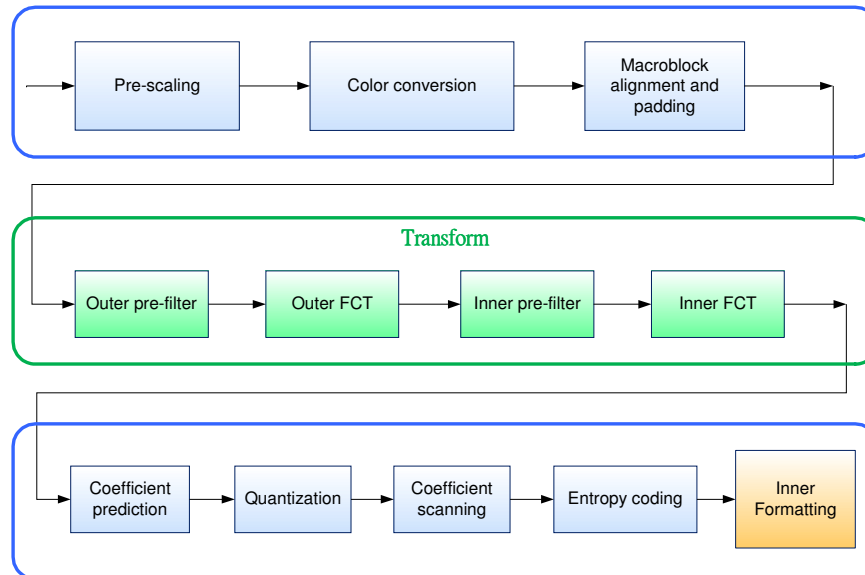


Figure 2.5 Block diagram of JPEG XR format encoding process.

### 2.4.3 Process of decoding of JPEG XR file

The JPEG XR decoder consists of two major parts: the parsing process and the decoding process, described as follows.

(1) The parsing process consists of steps described as follows:

- i. image layer and tile layer codestream parsing;
- ii. macroblock layer codestream parsing which includes parsing the transform coefficients and inverse scanning; and
- iii. adaptation of VLC table selection and context models.

(2) The decoding process consists of steps described as follows:

- i. coefficient remapping;
- ii. coefficient prediction;
- iii. de-quantization;
- iv. sample reconstruction; and
- v. output formatting.

In the parsing process, the decoder parses the information of codestream



structures such as image header and compressed data of the frequency domain. These data will be inversely transformed into the original coefficients of the frequency domain by inverse scanning and the VLC table selection.

After the parsing process, the decoder re-maps the original coefficients to correct positions in the image. And the transform coefficients may be predicted from the neighboring coefficients in the coefficients prediction process. Then, the transform coefficients are scaled by the quantization parameter in the de-quantization process.

The decoder takes a two-level inverse transform from the frequency domain to the spatial domain. The inverse transform process consists of the following steps.

The coefficients of DC and lowpass bands are grouped into a DC\_LP array as input data for the first-level transformation including the following steps:

- first-level inverse transform;
- when indicated, a first-level overlap filtering.

The resulting coefficients of the first-level transformation are combined with the highpass coefficients into the new input data for the second-level transformation including the following steps:

- second-level inverse transform;
- when indicated, a second-level overlap filtering.

An inverse core transform (ICT) is applied to 4×4 blocks, which is an inverse transform function of the FCT. The ICT operation transforms the frequency domain into the spatial domain, whose function is like the IDCT operation.

The overlap filtering is an inverse function of the pre-filtering, which is also optionally applied to 4×4 areas evenly straddling blocks in two dimensions. On the boundary, the overlap filtering operation is applied to 4×2 or 2×4 areas. And the overlap filtering operation does not work in four 2×2 corners.

Finally, the decoder converts the coefficients into OUTPUT\_COLOR\_FORMAT

for image display. The flowchart of JPEG XR decoding process is shown in Figure 2.5. In this study, we focus on the color conversion of the YUV model to the RGB.

The function of converting the YUV to the RGB is shown as follows:

$$\begin{aligned}
 t &= -U \\
 G &= Y - \left\lfloor \frac{t}{2} \right\rfloor \\
 R &= t + G - \left\lfloor \frac{V}{2} \right\rfloor \\
 B &= V + R
 \end{aligned}
 \tag{2.2}$$

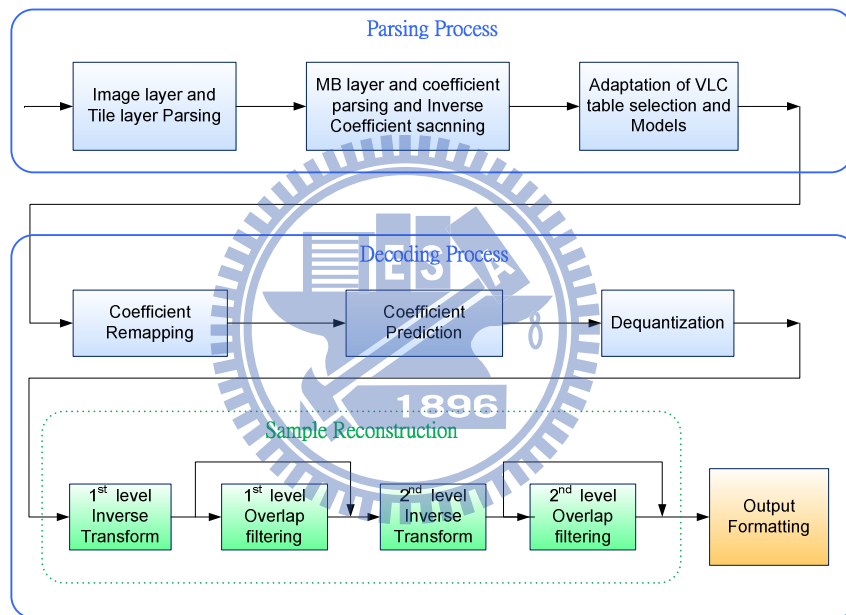


Figure 2.6 Block diagram of JPEG XR format decoding process.

# **Chapter 3**

## **Covert Communication via JPEG XR Images by Variable Macroblock Quantization**

### **3.1 Introduction**

Because of the rapid expansion of the Internet, the message transmitted through the Internet is not confidential nowadays. Private messages may be subject to sniffing attack by malicious people. To solve this problem, a common method is to use cryptography to protect the secret message. However, it is hard to conceal the action of secret communication via cryptography because the resulting stego-message usually is noise-like and often arouses suspicion from attackers. For the purpose of data confidentiality and covert communication, we propose a method to hide secret data via JPEG XR images.

In this chapter, we will describe the proposed data hiding technique via JPEG XR images. The major idea of the proposed method will be described in Section 3.1.2. The detailed data hiding and extraction processes will be given in Sections 3.2.2 and 3.2.3, respectively. In addition, some security enhancement measures for the proposed method will be proposed in Section 3.3.2. And some experimental results will be shown in Section 3.4. Finally, we will give a brief summary of this chapter in Section 3.5.

### 3.1.1 Problem definition

The JPEG XR images have become more and more popular nowadays, but researches of data hiding via them are not found yet according to a survey conducted in this study. Therefore, it is desired to develop a data hiding technique via JPEG XR images. The goal is to embed the secret message into a cover image without causing noticeable distortion. It is hoped that when the stego-image is transmitted to a receiver, other people will consider the behavior of transmission just as an activity of image sharing rather than secret communication. Only the receiver can extract the secret message from the stego-image with a secret key. Therefore, the aim of data hiding in JPEG XR images is how to design a method for embedding data imperceptibly and extracting the embedded data correctly. In addition, even a person knows the algorithms of the method, he/she still cannot extract the secret data without the secret key.

### 3.1.2 Major idea of proposed method

The proposed method is essentially based on the utilization of certain special characteristics of the JPEG XR format in the quantization of the FCT (forward core transform) coefficients in the frequency domain. In this aspect, the quantization process of the JPEG XR standard is different from that of the old JPEG standard. In the latter, a *fixed* 8×8 quantization table is used to quantize the DCT coefficients of every 8×8 blocks. Unlike this, the new JPEG XR standard supports *variable* quantization parameters for image quantization.

More specifically, before an image is compressed with tiles as units according to the JPEG XR standard, the following three sets of quantization parameters must be provided by the system for each color channel, called a *component*, of each tile for the

purpose of quantizing the FCT coefficients yielded by the compression process:

- (1) a DC quantization parameter;
- (2) a lowpass quantization parameter set  $S_L$ ;
- (3) a highpass quantization parameter set  $S_H$ .

After these parameter sets are used to quantize the FCT coefficients, four frequency bands are yielded, including:

- (1) a DC value;
- (2) a group of lowpass coefficients;
- (3) a group of highpass coefficients;
- (4) a group of flexbits which are also highpass coefficients but with very high frequencies and selected by a certain algorithm supplied by the JPEG XR standard.

This kind of quantization as conducted in JPEG XR image compression has the advantages of improving the image compression quality and rate. The reason is that images are not always smooth and usually have some sharp areas, and using a big quantization value for smooth areas and a small one for sharp areas will reduce data volumes and improve image qualities.

Additionally, each of the highpass and the lowpass parameter sets  $S_H$  and  $S_L$  is allowed to include at most 16 parameters, each of which is user-selectable for use in quantizing the FCT-coefficients for various application requirements. Use of these 16 parameters variably means that we may use them to encode 4 message bits for the purpose of data hiding. And this is just the idea of using variable quantization in JPEG XR image compression we propose for data hiding in this study.

Furthermore, for the purpose of reducing distortion, only the highpass band is used in the proposed data hiding method. This way provides an advantage which is not gained by the standard JPEG XR compression (such as by the compression

operation offered by the application software Adobe Photoshop) which quantize all bands of frequencies, not just the highpass band --- that is, the distortion coming from re-quantization using the proposed method is smaller than that produced by the standard compression provided by commercially-available JPEG XR software.

## 3.2 Proposed Method for Covert Communication

### 3.2.1 Information hiding by variable macroblock quantization

A configuration of the proposed system for covert communication is shown in Figure 3.1. As can be seen from the figure, the proposed method adopts a strategy of *after-encoding and before-decoding secret processing (including hiding and extraction)*. That is, the secret is embedded into a cover image after the image is encoded to become a JPEG XR file, and the hidden secret is extracted from the JPEG XR file of a stego-image before the file is decoded for display. Note that this is not the strategy adopted in some other methods for covert communication [11-12].

Owing to the above-mentioned strategy of secret processing, we have to design a special *JPEG XR file data extractor* for use in this study to extract the image information from the bitstream of the cover image and a special *data embedder* to insert secret data into the bitstream. Furthermore, the information of the cover image has been changed after data hiding. We therefore also need to design a special *JPEG XR file data encoder* to re-encode the bitstream into a stego-image. In short, to design a secret embedder, as the one shown in Figure 3.1, we have to include three software

“devices” in it: (1) a JPEG XR file data extractor; (2) a data embedder, and (3) a JPEG XR file data encoder.

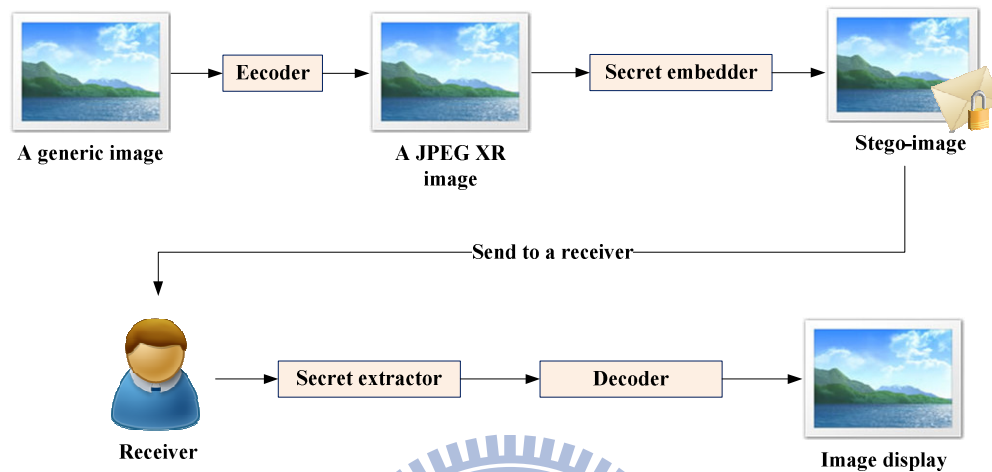


Figure 3.1 The flowchart of data embedding and extraction via the un-decoded JPEG XR file format.

After the devices are designed, we increase the number of quantization parameters of the highpass bands to be 16 for hiding 4 bits per macroblock, as mentioned previously. Because the distortion of the stego-image depends on the quantization parameters, how to select the new quantization parameter values to yield less distortion is an important issue. According to the experimental results of some compression distortion test conducted in this study as shown by Figure 3.2, Table 3.1, and Figure 3.3, we found four facts about the relationships between the distortion and the selected quantization parameter value, which are described in the following.

- i. Smaller quantization parameter values have a tendency to cause less distortion.
- ii. If a new quantization parameter value is one of the factors of the original

one, the re-compressed image will have a slight distortion.

- iii. The distortion will be rapidly raised when a quantization parameter value is larger than a multiple of the original quantization parameter value.
- iv. The trend line of distortion is smooth between the original quantization parameter value and its double.

Based on the above observations, candidate values for use as selected quantization parameters are classified into four groups according to the priority of being used as new quantization parameters. The first group has a top priority, being composed of the factors of the original quantization parameter value. The second group has the second highest priority, consisting of the candidate parameter values which are other than the candidate values in the first group and are smaller than the original quantization parameter value. The third group has the third highest priority, which contains candidate parameter values between the original quantization parameter value and its double. And the last group has the lowest priority, which includes candidate parameter values larger than a double of the original quantization parameter value.

It is observed in this study that using larger quantization values will cause smaller volumes of compressed image data. For this reason, the priority of the candidate parameter values in each group is sorted according to their magnitudes; a larger parameter value is given a higher priority. However, the distortion of the last group is rapidly raised in accordance with the growth of the magnitude of the parameter value. So in the last group, a smaller parameter value instead is given a higher priority.

After new quantization parameters are selected for use, we can start to embed secret data in the cover image. At first, the new quantization parameters are inserted into the correct places of the bitstream of the JPEG XR file according to the JPEG XR



standard. Next, the secret data are converted into 4-bit segments, and each segment is transformed into a decimal number ranging from 0 to 15. Then, in order to improve the security of the secret data, we use a one-to-one mapping function controlled by a secret key to transform the decimal numbers into new ones randomly. The result of the one-to-one mapping also ranges from 0 to 15 and forms a set of new quantization parameter values, called *parameter indexes*. Finally, these indexes are inserted into the bitstream to specify the corresponding quantization parameter in each macroblock. After these steps, the secret data are completely embedded into the cover image, and people who do not know the secret key cannot extract the secret data.



Figure 3.2 The original image used for compression distortion test, for which the applied quantization value is 15. In this study, we call this image “Two\_dogs.”

Table 3.1 The relationship between compression ratios and PSNRs. The quantization value of the original image is 15. QP is the quantization parameter used for compression. QP = 0 means that the quantization parameter value 1 when encoded by the JPEG XR standard.

QP	0	1	2	3	4	5	6	7
PSNR	68.524	68.490	56.450	68.396	52.798	68.221	52.799	54.299
QP	8	9	10	11	12	13	14	15
PSNR	47.473	48.971	50.105	49.130	49.844	51.136	54.049	66.430
QP	16	17	18	19	20	21	22	23
PSNR	41.101	41.363	41.578	41.623	41.819	41.734	41.875	41.069
QP	24	25	26	27	28	29	30	31
PSNR	41.192	41.344	41.325	41.427	41.525	41.661	41.745	38.476
QP	32	33	34	35	36	37	38	39
PSNR	38.476	38.541	38.645	38.541	38.639	38.685	38.751	37.039

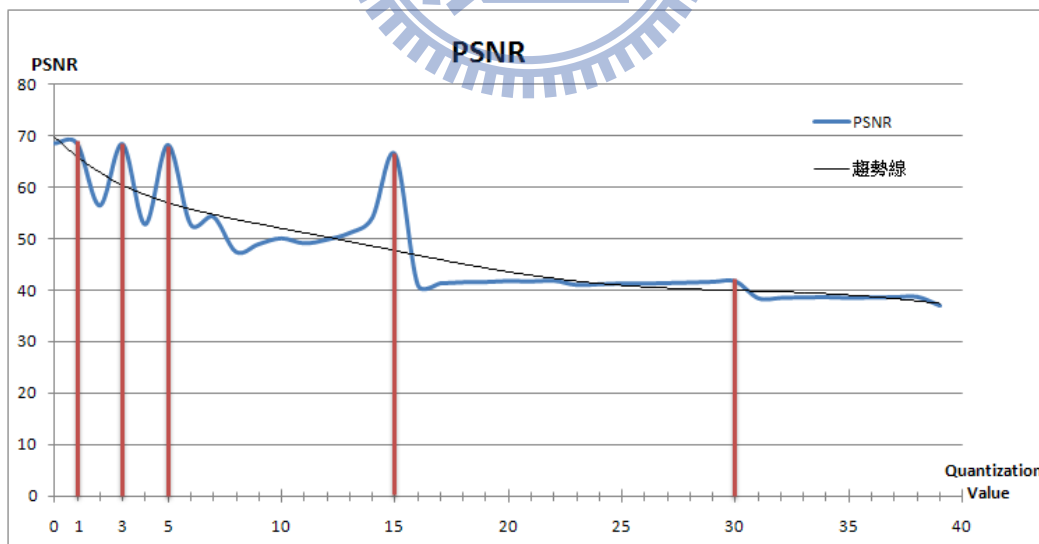


Figure 3.3 The scatter plot with smooth lines of relationship between compression ratios and PSNRs. The quantization value of the original image is 15. The quantization parameter is used for compression.

However, if we only change the quantization values of each macroblock without re-quantizing the highpass coefficients, it will cause large distortion in the resulting image. To solve this problem, we also quantize the highpass-band coefficients by the selected quantization parameter values. More specifically, we decode first the bitstream of each macroblock to extract the original quantization parameter values and the highpass coefficients. Then, the coefficients are de-quantized by the extracted original quantization parameter values and re-quantized by the newly-selected quantization parameter values. In this way, the distortion of stego-images is found to be under control. Finally, we encode the coefficients into bitstreams according to the JPEG XR standard and the result is transmitted to the receiver. The above-described secret data embedding process is illustrated by Figure 3.4.

In the data extraction process, we extract the quantization parameter indexes of each macroblock and transform the indexes into decimal numbers. Then, the secret key is used again in the previously-mentioned one-to-one mapping function to recover the original input numbers. Finally, all the numbers are transformed into 4-bits strings and combined together to get the embedded secret message. The extraction process is similar to the embedding process but conducted in a reverse order, as illustrated in Figure 3.5.

### **3.2.2 Algorithm for data embedding**

In this section, the detailed algorithm of data embedding will be introduced. We give a brief description of the algorithm and its input and output parameters here first. A secret message  $S$  which, with the data length of  $l$  bytes, will be embedded into a cover JPEG XR image  $I$ . The length  $l$  is transformed into a 16-bit unsigned integer, which will then be embedded into the first four macroblocks  $M_1, M_2, \dots, M_4$  of a tile. The message  $S$  will be separated into  $2 \times l$  4-bit segments and embedded into the

remaining macroblocks  $M_5, M_6, \dots, M_{2 \times l + 4}$ . And the method uses a secret key  $K$  and a random number generator  $f$  to conduct random 1-to-1 mappings from the range of 0, 1, ..., 15 onto the same range of 0, 1, ..., 15 in the algorithm. The detail is described in the following.

**Algorithm 3.1.** Data embedding for covert communication.

**Input:** a user key  $K$ , a random number generator  $f$ , a secret message  $S$  with character length  $l$ , and a cover JPRG XR image  $I$  with  $N$  macroblocks.

**Output:** a stego-JPEGX XR image  $I'$ .

**Steps:**

*Step 1.* Check if  $2 \times l - 4 \leq N$ ; if so, continue; otherwise, regard  $S$  as too big to be embedded into  $I$  and exit.

*Step 2.* Transform  $l$  and  $S$  into a series of decimal numbers  $l_1, l_2, \dots, l_4$  and  $i_1, i_2, \dots, i_{2 \times l}$  according to the following steps, where  $0 \leq l_1, l_2, \dots, l_4 \leq 15$ ,  $0 \leq i_1, i_2, \dots, i_{2 \times l} \leq 15$ .

2.1 Separate  $l$  and  $S$  into a series of 4-bit segments  $s'_1, s'_2, \dots, s'_4$  and  $s_1, s_2, \dots, s_{2 \times l}$ , respectively.

2.2 Transform  $s'_1, s'_2, \dots, s'_4$  and  $s_1, s_2, \dots, s_{2 \times l}$  into a series of decimal numbers  $d'_1, d'_2, \dots, d'_4$  and  $d_1, d_2, \dots, d_{2 \times l}$ , respectively, where  $0 \leq d'_1, d'_2, \dots, d'_4 \leq 15$  and  $0 \leq d_1, d_2, \dots, d_{2 \times l} \leq 15$ .

2.3 Use  $K$  and  $f$  to conduct random 1-to-1 mappings from  $d'_1, d'_2, \dots, d'_4$  and  $d_1, d_2, \dots, d_{2 \times l}$  to  $l_1, l_2, \dots, l_4$  and  $i_1, i_2, \dots, i_{2 \times l}$ , respectively.

*Step 3.* Extract the original highpass quantization parameter value  $Q$  from the bitstream of the JPEG XR file of  $I$ .

*Step 4.* Select new quantization parameter values  $q_0, q_1, \dots, q_{15}$  according to the selection priority described in Section 3.2.1, perform the following steps

with  $M$  used as a counter.

4.1 Get all the factors of  $Q$ , and perform the following steps:

- i. sort the factors;
- ii. select the largest 16 ones for use as  $q_0$  through  $q_{15}$  and set  $M = 16$ , if there are more than 16 factors; otherwise, select all of them, say  $m$  ones, as  $q_0$  through  $q_{m-1}$ ; select 0 as  $q_m$ , and set  $M = m + 1$  (where 0 means that the quantization parameter value 1 in the highpass band when encoded by the JPEG XR standard).

4.2 If  $M \neq 16$  (i.e., not all of the 16  $q_i$ 's are selected already), then perform the following steps; otherwise, go to Step 5:

- (a) sort all the integer numbers smaller than  $Q$  and other than the  $q_i$ 's selected in Step 4.1, and denote the number of these integers as  $N_Q$ ;
- (b) select the largest  $16 - M$  ones for use as  $q_{m+1}$  through  $q_{15}$  and set  $M = 16$  if  $N_Q - 1 > 16 - M$ ; otherwise, select all of the sorted numbers, say  $n$  ones, as  $q_{m+1}$  through  $q_{m+n}$  and set  $M = m + 1 + n$ .

4.3 If  $M \neq 16$  (i.e., not all of the 16  $q_i$ 's are selected already), then perform the following steps; otherwise, go to Step 5:

- (a) sort all the integer numbers which are smaller than  $2Q + 1$  and larger than  $Q$ , and denotes the number of these integers as  $N_{2Q}$ ;
- (b) select the largest  $16 - M$  ones for use as  $q_{m+n+1}$  through  $q_{15}$  and set  $M = 16$  if  $N_{2Q} - 1 > 16 - M$ ; otherwise, select all of the sorted numbers, say  $o$  ones, as  $q_{m+n+1}$  through  $q_{m+n+o}$  and set  $M = m + 1 + n + o$ .

4.4 If  $M \neq 16$  (i.e., not all of the 16  $q_i$ 's are selected already), then select the smallest  $16 - M$  integers larger than  $2Q$  for use as  $q_{m+n+o+1}$  through

$q_{15}$ ; otherwise, continue.

*Step 5.* Insert  $q_0, q_1, \dots, q_{15}$  into the correct places in the bitstreams of the JPEG XR file of  $I$  to be a new quantization parameter set of the highpass frequency band.

*Step 6.* For each macroblock  $M_i$  into which 4-bit secret data are to be embedded, perform the following steps, where  $1 \leq i \leq 2 \times l + 4$ .

6.1 Select the new quantization parameter value  $QP_i$  of each  $M_i$  by the following rule.

a. If  $i \leq 4$ , then

use  $l_1, l_2, \dots, l_4$  as indexes to decide the new quantization parameter values  $QP_1, QP_2, \dots, QP_4$  from  $q_0, q_1, \dots, q_{15}$  for  $M_1, M_2, \dots, M_4$ , respectively, in the following way, where  $1 \leq i \leq 4$ :

if  $l_i$  is the value  $k$ , then  $QP_i$  is the value  $q_k$ ;

b. else,

use  $i_j$  as an index to decide a new quantization parameter value  $QP_i$  from  $q_0, q_1, \dots, q_{15}$  for each  $M_i$ , respectively, in the following way, where  $j = i - 4, 5 \leq i \leq 2 \times l + 4, 1 \leq j \leq 2 \times l$ :

if  $i_j$  is the value  $k$ , then  $QP_i$  is the value  $q_k$ .

6.2 Extract all the coefficients  $h_1, h_2, \dots, h_{240}$  of each  $M_i$ , and de-quantize them into  $h'_1, h'_2, \dots, h'_{240}$  by  $Q$ , where  $5 \leq i \leq 2 \times l + 4$ .

6.3 Quantize  $h'_1, h'_2, \dots, h'_{240}$  into  $h''_1, h''_2, \dots, h''_{240}$  by  $QP_i$  where  $5 \leq i \leq 2 \times l + 4$ .

6.4 Repeat the above steps if  $2 \times l + 4 \geq i$ .

*Step 7.* Encode all new highpass coefficients into bitstreams according to the JPEG

XR standard to generate a stego-JPEG XR image  $I'$ .

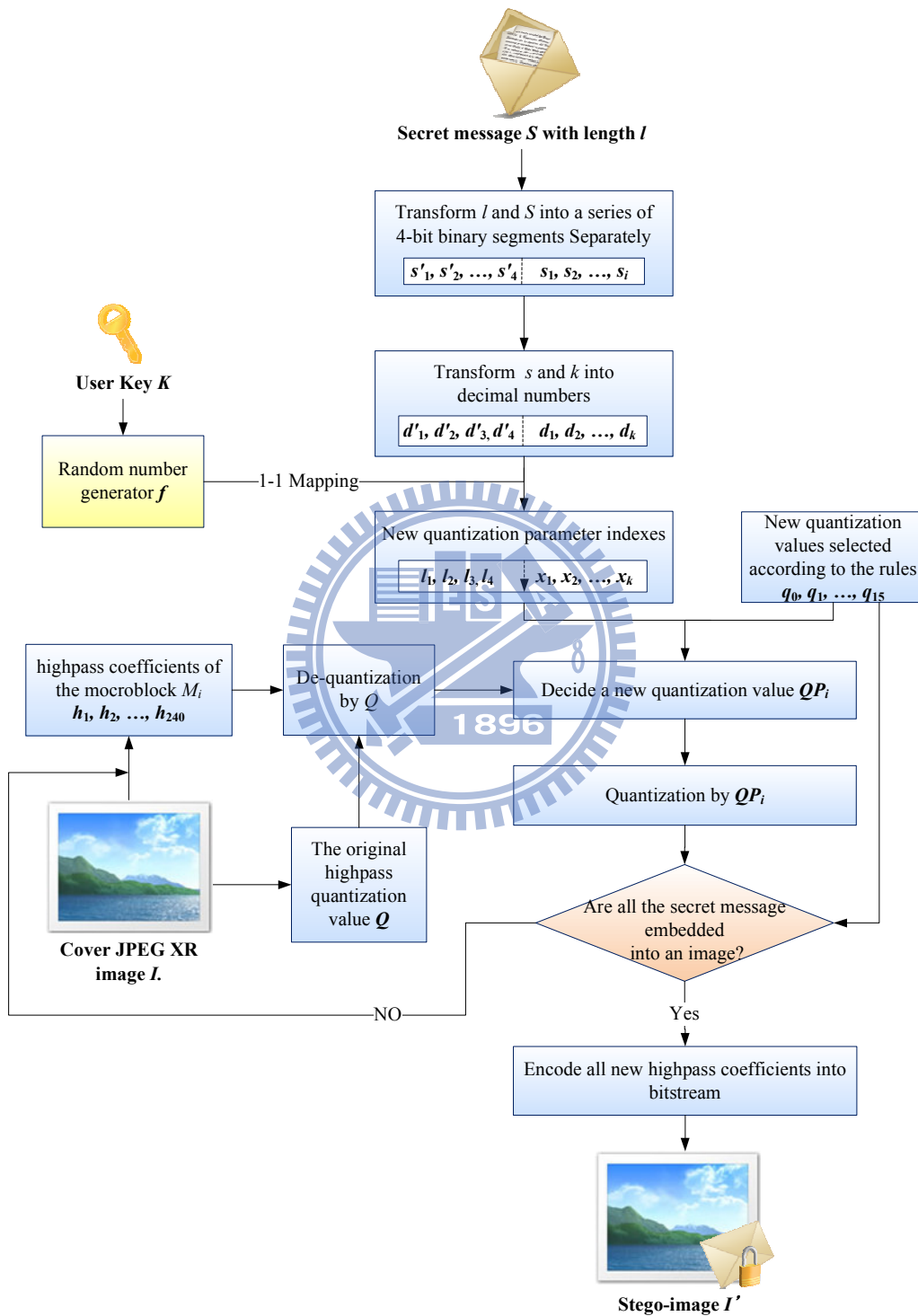


Figure 3.4 The flowchart of the proposed data embedding process.

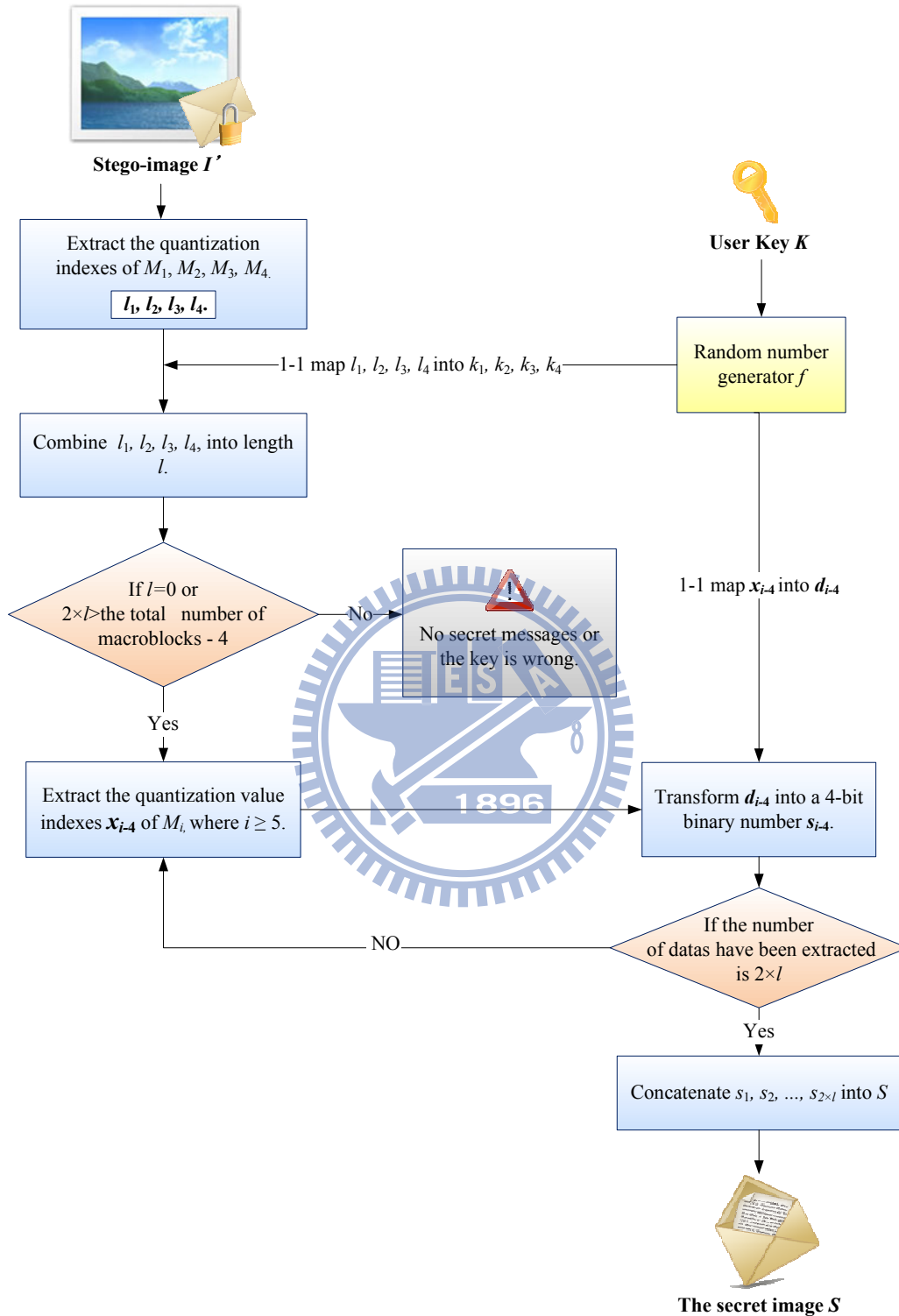


Figure 3.5 The flowchart of the proposed data extraction process.



### 3.2.3 Algorithm for data extraction

The detail of the proposed secret message extraction process is described as an algorithm in the following.

#### **Algorithm 3.2. Proposed data extraction process for covert communication.**

**Input:** A stego-JPEGX XR image  $I'$ , a user key  $K$ , and a random number generator  $f$  with  $K$  and  $f$  being identical to those used in Algorithm 3.1.

**Output:** A secret message  $S$ .

#### **Steps:**

*Step 1.* Extract the length  $l$  of the embedded data by performing the following steps.

- 1.1 Extract  $l_1, l_2, \dots, l_4$  by decoding the quantization parameter indexes of macroblocks  $M_1, M_2, \dots, M_4$ , respectively, from the JPEG XR file of  $I'$ .
- 1.2 Use the key  $K$  and  $f$  to conduct random one-to-one mappings from  $l_1, l_2, \dots, l_4$  to  $d'_1, d'_2, \dots, d'_4$ , and combine the mapping results to form  $l$ .

*Step 2.* If  $l = 0$  or  $2 \times l > N - 4$ , show an error message saying that “Wrong Key!!!”; otherwise, continue.

*Step 3.* For each macroblock  $M_i$ , perform the following steps to extract 4-bit secret data, where  $5 \leq i \leq 2 \times l + 4$ .

- 3.1 Extract the value  $x_{i-4}$  by decoding the quantization parameter index of  $M_i$  from the JPEG XR file of  $I'$ .
- 3.2 Use  $K$  and  $f$  to conduct random one-to-one mappings from all  $x_{i-4}$  to all  $x_{i-4}$  themselves to get  $d_{i-4}$ , where  $5 \leq i \leq 2 \times l + 4$ .
- 3.3 Transform  $d_{i-4}$  into a 4-bit value  $s_{i-4}$ .
- 3.4 Repeat the above steps if  $2 \times l + 4 \geq i$ .

*Step 4.* Concatenate  $s_1, s_2, \dots, s_{2 \times l}$  to form  $S$ .

*Step 5.* Output the desired secret message  $S$ .

### 3.2.4 Resuming original file size

The size of the stego-image will be changed after data hiding. For example, the file size of stego-image shown in Figure 3.8 was raised from 234572 bytes to 236623 bytes. As a result, it is necessary to resume the size to the original file in order to reduce the possibility of arousing notice from attackers during the secret communication process. In this study, we propose a method to generate a stego-image whose file size is kept to be the original one of the cover image based on the JPEG XR standard.

The JPEG XR standard uses an `index_table` in the JPEG XR format to indicate the start position of every tile packet. The data length  $L_1$  of each tile packet  $P_1$  is equal to the distance between its start position, denoted as  $T_1$ , and the start position  $T_2$  of the next tile packet  $P_2$ . If we modify  $T_2$  (i.e., increase the value of  $T_2$ ) of  $P_2$  in the `index_table` to give a larger space to  $P_1$  without changing its data length  $L_1$ , there will create some *undefined* space at the end of  $P_1$ , and the decoder will ignore the space in the decoding process. By this way, we can increase the file size of the stego-image without distortion.

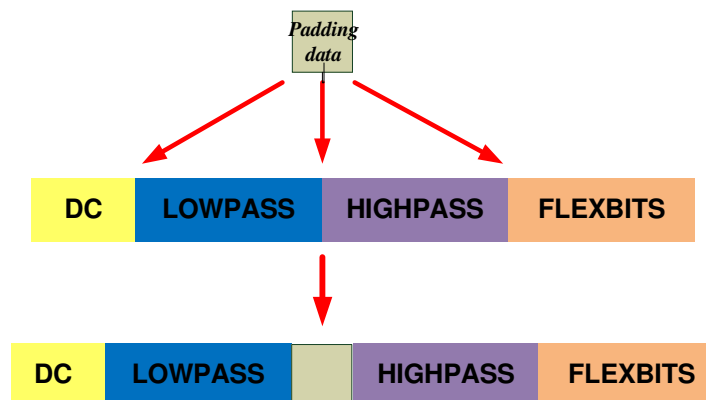


Figure 3.6 Inserting the padding data between the two tile packets.

On the other hand, the syntax element TRIM\_FLEXBITS is used to control the distortion of flexbits in the JPEG XR standard. When the TRIM\_FLEXBITS is  $m$ , the last  $m$  bits of each flexbits coefficient will be truncated (the *flexbits* specify a frequency band defined by the JPEG XR standard). Because the flexbits band stands for the highest frequency domain, truncating them will not cause a big distortion. By this property, we can reduce the size of the stego-image by modifying the TRIM\_FLEXBITS without degrading the stego-image quality. For example, the size of the flexbits packet of the stego-image in Figure 3.8 is 29233 bytes. After we modified TRIM\_FLEXBITS from 0 to 1 and re-encode the stego-image according to the JPEG XR standard, the size of the flexbits packet became 318 bytes.

To use the above two properties for keeping the original size  $Z_c$  of the cover image in the stego-image which, say, is of the size  $Z_s$ , we proceed in the following way:

- (a) if  $Z_c > Z_s$ , then increase the size of the stego-image by the above-mentioned way of creating undefined space after an arbitrarily selected tile packet;
- (b) otherwise,
  - i. reduce the size  $Z_s$  of the stego-image by the above-mentioned way of truncating a sufficient number of least significant bits (LSBs) of the flexbits coefficients so that the new size  $Z_s'$  of the stego-image is smaller than  $Z_c$ ; and
  - ii. increase the current size  $Z_s'$  of the stego-image by the above-mentioned way of increasing the size of the stego-image so that the new size  $Z_s'' = Z_c$ .

In short, since we can increase the image size and then reduce it in the above way, the size of the stego-image can be resumed to the original size of the cover image with no distortion (when no truncation operation is conducted) or with only a little

distortion (when truncation operations are conducted) , as proved by our experimental results. We give an example in Figure 3.7.

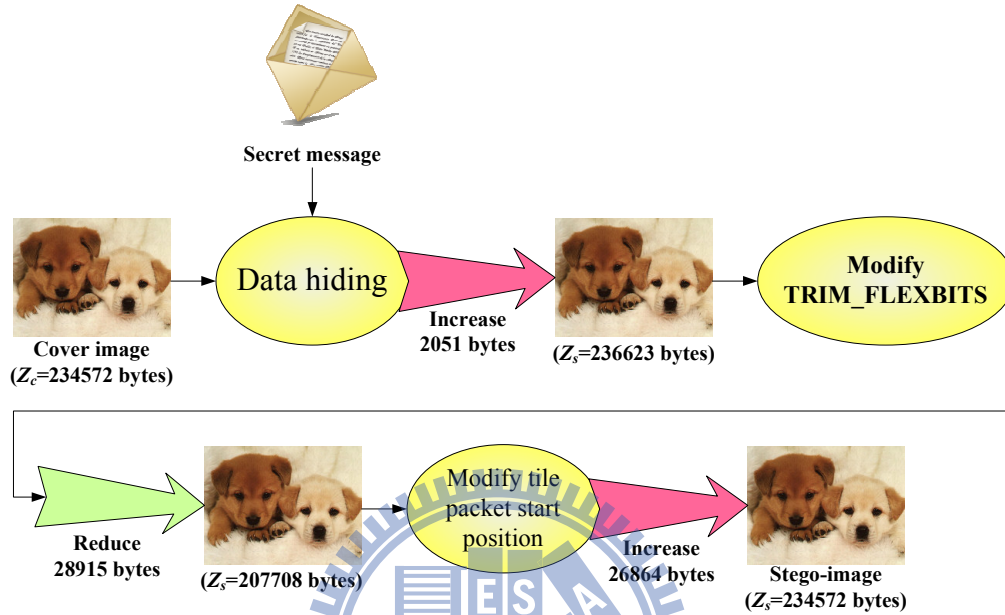


Figure 3.7 The process of resumming to the original size of the cover image.

## 3.3 Security Consideration

### 3.3.1 Issues of security of proposed method

In the proposed method, the secret key is used in the 16×16 one-to-one mapping function to map the secret data to new ones, and so there are 16! ( $\approx 20922$  billion) ways of such mapping. So, it is also impossible for a malicious person to extract the secret data without the correct secret key. But there is still a problem of security caused by the embedded data length. Because the data length is a number smaller than the total number of macroblocks, it will reduce the security of the one-to-one mapping function. Under the assumption that a malicious attacker knows the algorithms of the proposed method, he/she may try to recover the data length, perform the algorithms,

and observe the output data to guess the secret key by trial and error. We thus propose a method to solve this problem in the next section.

### 3.3.2 Proposed security enhancement measures

In order to improve the security of the proposed method in the aspect of the above-mentioned problem, another secret key is used to randomly select the macroblocks into which the data length and the secret data are embedded. In this way, the malicious people do not know where the data length and the secret data are embedded. Thus, the security of the embedded secret data thus is highly enhanced.

## 3.4 Experimental Results

In our experiments, the proposed embedding and extraction algorithms were implemented using Microsoft Visual C<sup>++</sup>. The JPEG XR images can be opened and displayed by a Windows Live Viewer. We create JPEG XR images by using Adobe Photoshop software.

One result of the experiments is illustrated as follows. A large message of 1534 bytes was embedded into a stego-image by the proposed method and the PSNR is 51.4811. The original cover image Two\_dogs is shown in Figure 3.2, and the stego-image is shown in Figure 3.8. Another sample message “I love dogs!! dogs love me~” was embedded into the cover image again. The result is shown in Figure 3.9.

Figure 3.10 shows that the message can be extracted by using the correct user key. If an erroneous user key was used, after the data extraction process was carried out, a series of noise-like random codes resulted, as shown in Figure 3.11.

In order to maintain the stego-image file size to be that of the cover image, the proposed size resuming technique was applied to the image shown in Figure 3.12.

And Figure 3.13 prove that the proposed method is feasible. The other results are shown in Figure 3.14 and Figure 3.15. The PSNRs of the ten selected stego-images and their average is shown in Table 3.2.

## 3.5 Summary

In this chapter, a new method of data hiding using variable macroblock quantization via JPEG XR images has been proposed for covert communication. The method hides the secret data into a cover image, and causes less distortion which is hard to notice. The principle of the proposed method is based on re-compressing the high-frequency band. And a random  $16 \times 16$  one-to-one mapping function controlled by a secret key is used to protect the secret data. If a person does not have the secret key, he/she cannot extract the secret data correctly. In addition, we have proposed a method to maintain the original file size of the cover image in producing a stego-image in order to reduce the possibility of arousing notice from attackers. An additional security measure based on random selections of macroblocks for data embedding has also been proposed. Experimental results have shown the feasibility of the proposed method.





Figure 3.8 The stego-image which hides a message of 1534 bytes. All the macroblocks are used for hiding the message. And the PSNR is 51.4811.



Figure 3.9 The stego-image which hides the information “I love dogs!! dogs love me~”.

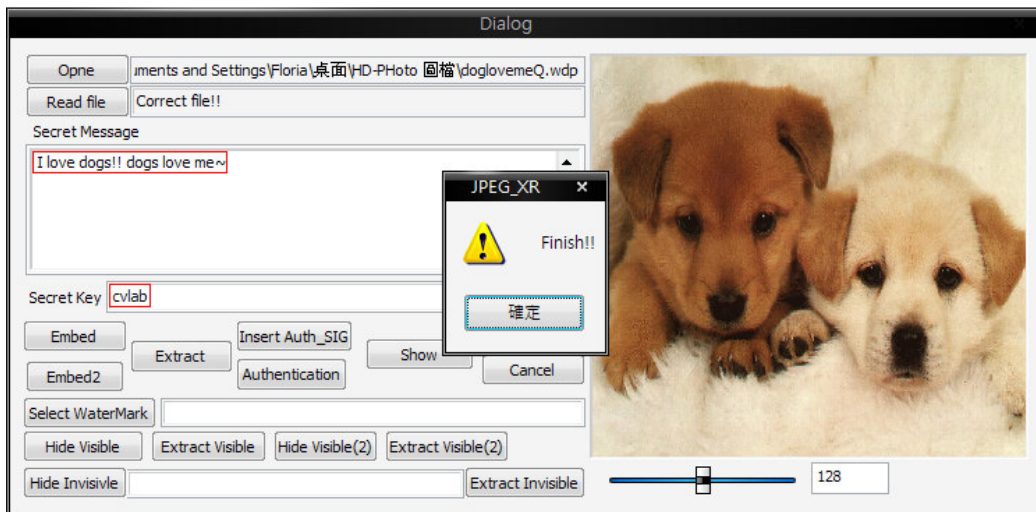


Figure 3.10 User interface for data extraction using the right secret key.

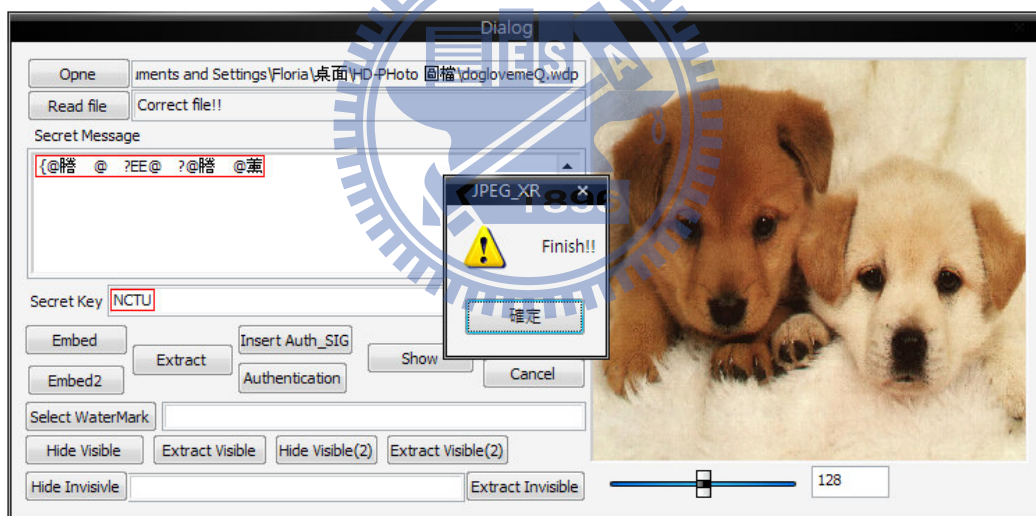


Figure 3.11 User interface for data extraction using a wrong secret key.





Figure 3.12 The stego-image which hides the information “I love dogs!! dogs love me~”, and keeps the original file size of the Two\_dogs image with PSNR 46.9041.

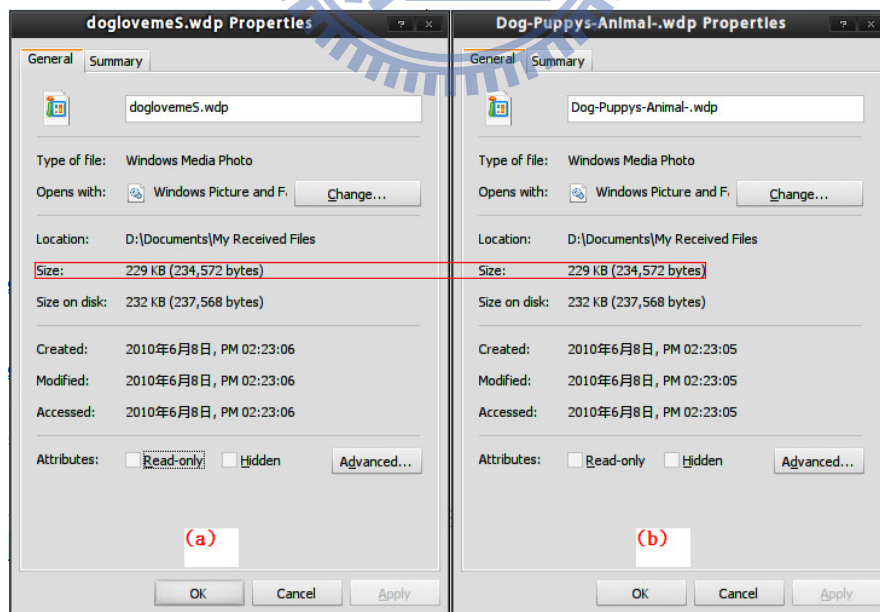


Figure 3.13 A stego-image whose file size is kept to be the original one of the cover image. (a) The size of the “Two\_dogs” image. (b) The size of Figure 3.6.



Figure 3.14 The stego-image in which all the macroblocks are used for hiding the message. And the PSNR is 52.324.

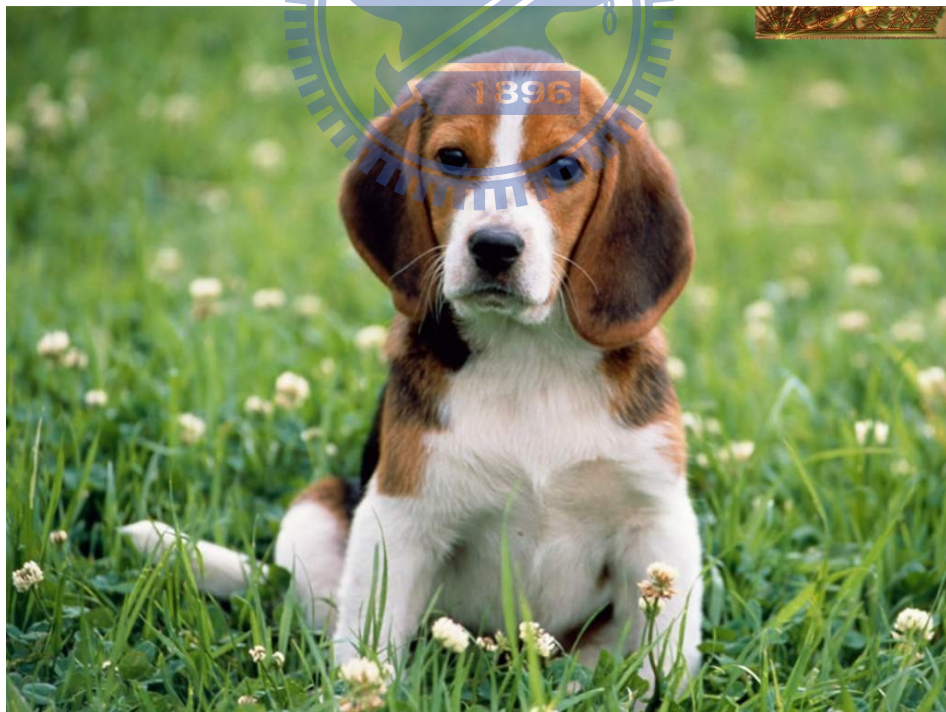


Figure 3.15 The stego-image in which all the macroblocks are used for hiding the message. And the PSNR is 53.2166.

Table 3.2 The PSNRs of the ten selected stego-images and their average.

Image	i	ii	iii	iv	v
PSNR	51.4811	51.6266	53.5597	51.123	52.2776
Image	vi	vii	viii	ix	x
PSNR	52.324	52.0849	53.2166	54.7805	53.2671
Average	52.57411				



# Chapter 4

## JPEG XR Image Authentication by Comparison of DC and Low-pass Frequency Coefficients

### 4.1 Introduction

Because of the convenience of the Internet, more and more people exchange images through the Internet in recent years. However, the images may be illicitly tampered with when they are transmitted on the Internet. To solve this problem, we propose an authentication method via JPEG XR images for fidelity or/and integrity verification.

In this chapter, we will describe the proposed authentication method via JPEG XR images. The major idea of the method will be described in Section 4.1.2. The detailed authentication signal generation and embedding process and the authentication process of JPEG XR images will be given in Sections 4.2.2 and 4.2.3, respectively. In addition, some security enhancement measures for the proposed method will be proposed in Section 4.3.2. And some experimental results will be shown in Section 4.4. Finally, we will give a brief summary of this chapter in Section 4.5.

#### 4.1.1 Problem definition

The JPEG XR images may be tampered with for some illegal purposes. After a

person receives a tampered image, he/she will be misled to believe the wrong content. The aim of the proposed authentication method is to hide authentication signals into each block of a cover image to protect the fidelity or/and integrity of the image blocks. When the authentication signals have been hidden, resulting in a stego-image, if the stego-image is tampered with, the authentication signals in the tampered areas will also be destroyed, too. We can authenticate the stego-image by extracting the authentication signals. If the extracted authentication signal of a block is incorrect, we decide this block have been tampered with. In addition, a secret key is used in the proposed authentication method to avoid the condition that somebody knows the proposed method and creates a fake stego-image.

#### 4.1.2 Major idea of proposed method

The JPEG XR standard defines a two-level macroblock-based FCT transformation to transform the image from the space domain into the frequency domain, as shown in Figure 4.1 and Figure 4.2. For each macroblock, a FCT is applied to all blocks. This completes the construction of the level-one frequency domain. The DC coefficients of 4×4 blocks are then grouped together into 4×4 DC\_LP arrays, and an FCT is applied to the DC\_LP arrays. This completes construction of the level-two frequency domain. By the transformation functions, the JPEG XR standard divides the frequency domain of each macroblock into three bands: a DC band, a lowpass band, and a highpass band.

The proposed method is also implemented in an *after-encoding and before-decoding* fashion. A simple flowchart of the authentication process is shown in Figure 4.3. A special *JPEG XR file data extractor*, a special *authentication signals embedder*, a special *authentication signals extractor* and a special *JPEG XR file data encoder* are also designed in this study for use here.



In the proposed method, we take the DC\_LP coefficient of each block as an authentication signal for the block, which then is hidden into the highpass coefficients in each block, where the DC\_LP coefficients are the DC coefficients after level-one FCT transformation. When a block is tampered with, the DC\_LP coefficient is altered, and the authentication signal which has been embedded into the highpass coefficients will also change. We can compare the authentication signal with the DC\_LP coefficient to decide whether this block have been tampered with or not. By this, we can authenticate the integrity of each block. The main idea behind this method is hiding low-frequency values into the high-frequency domain. In this way, every frequency bands are under protection; the changes in each frequency band will be detected. The authentication signals embedding and extraction processes are illustrated in Figures 4.4 and 4.5.

## **4.2 Proposed Method for Image Authentication**

### **4.2.1 Generation of authentication signals**

The proposed method takes the DC\_LP coefficients as the authentication signals of each block, and hides them into the LSBs of the highpass coefficients. For example, given a value of DC\_LP coefficient 405 ( $000000110010101_2$  in binary), we take the binary stream “000000110010101” as an authentication signal and hide them into the LSBs of 15 highpass coefficients of the block. After this, the authentication signals can be extracted from the LSBs and compared with the DC\_LP coefficients to authenticate the integrity of each block.

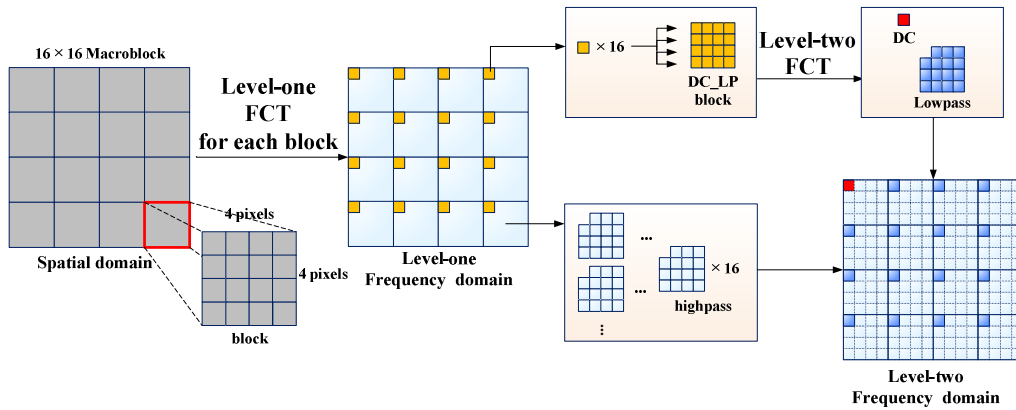


Figure 4.1 The two level FCT transformation.

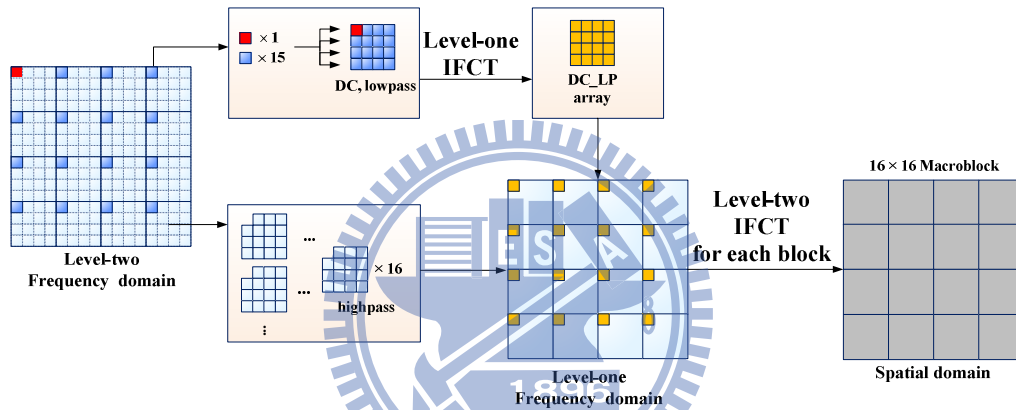


Figure 4.2 The two level IFCT transformation.

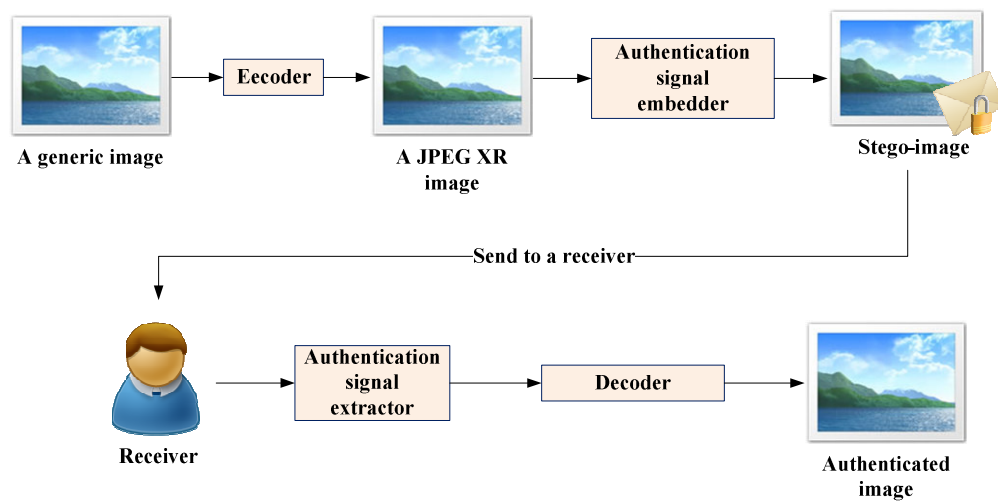


Figure 4.3 The flowchart of authentication signals embedding and extraction via the un-decoded JPEG XR file format.

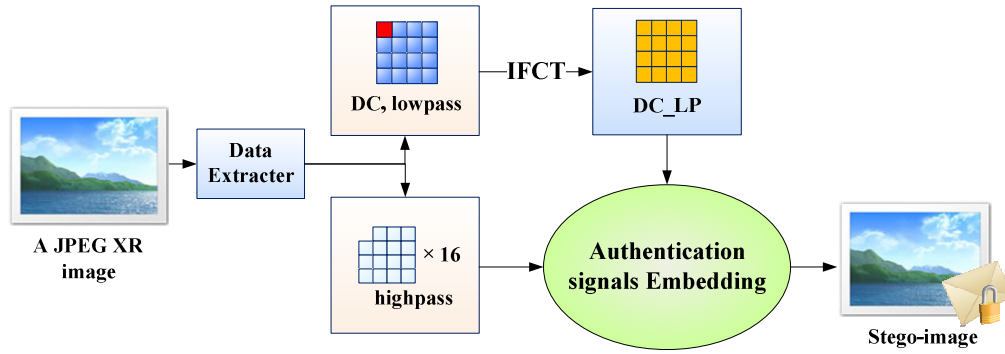


Figure 4.4 The brief flowchart of authentication signals embedding process.

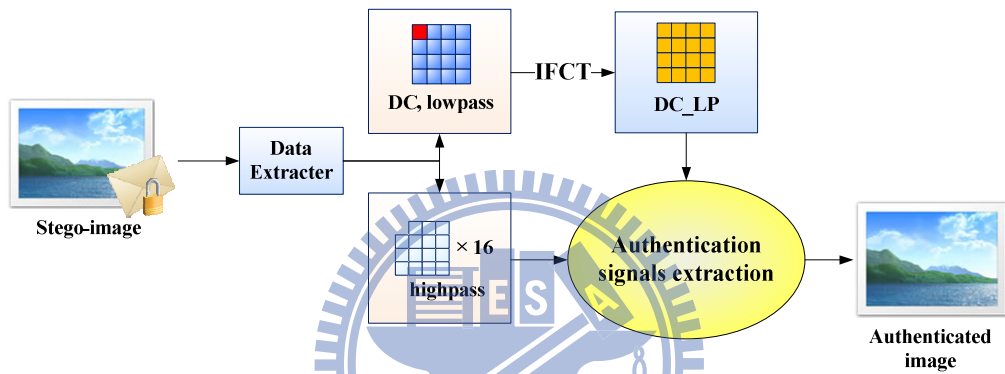


Figure 4.5 The brief flowchart of authentication signals extraction process

However, replacing the LSBs of highpass coefficients directly is not a good idea, because a hacker might make changes of highpass coefficient bits other than LSBs and such changes cannot be detected if only the LSBs are checked. To solve this problem, a special method of replacing the LSB is designed in this study. At first, we select the LSB  $l$  and another *random* bit  $r$  of each highpass coefficient by the secret key. When a bit  $a$  is to be embedded into a highpass coefficient, we replace the LSB  $l$  by the result of  $r \oplus a$ . As a result, changing the bits of the highpass coefficient other than the LSB may also destroy the authentication signal, and so will be detected.

In addition, in order to improve the security, we pair the  $l$ 's and  $r$ 's of all the highpass coefficients of a block randomly selected by the secret key. For example, the



LSB  $l_3$  of the 3rd highpass coefficient of the block may be paired with bit  $r_{14}$  of the 14-th highpass coefficient of the block.

Because of the specification of the JPEG XR standard, there still have some difficulties in the proposed method, which are described in the following.

- i The size of the image is not always a multiple of 16. When an image width or height is not a multiple of 16, the JPEG XR standard extends the right column and the bottom row of the image to the nearest higher multiple of 16. Therefore, the authentication signal will be destroyed in the right and bottom parts of the image by re-encoding because of the padding coefficients. By this, we should ignore the wrong results of the authentication in the right and bottom macroblocks when their sizes are smaller than 16 before extending.
- ii When a JPEG XR image is encoded by the encoder, it will be decoded into the spatial domain first, and then re-encoded by the encoder. In the JPEG XR standard, when a number overflow occurs, it will be taken as a legal number by the decoder. If re-encoding is conducted, the number will be truncated, as shown in Figure 4.6. Therefore, if a number after authentication signal embedding incurs an overflow, it will be reset to a legal number in the re-encoding process. As a result, the embedded authentication signals will be destroyed, too. In order to solve this problem, we should modify the pixel values first if they incur overflows after embedding authentication signals. The pre-selected threshold  $m$  is used to restrict the pixel value which is closed to the upper-bound or lower-bound of the pixel value. For example, let  $m$  be 2 and the pixel values range from 0 to 255. If a pixel value is larger than  $255 - m$  or smaller than  $0 + m$ , it will be modified into the range of  $m$  to  $255 - m$ .

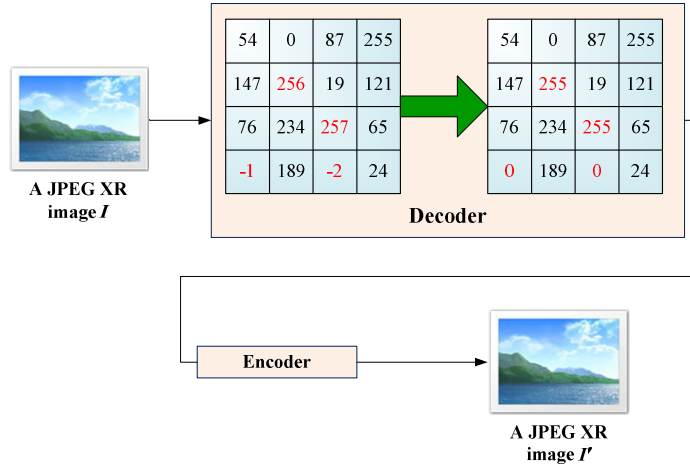


Figure 4.6 The overflowing pixels value truncated by re-encoding.

iii The JPEG XR standard provides an optional overlap filtering before the FCT to smooth coefficients, which is optionally applied to  $4 \times 4$  areas evenly straddling blocks in two dimensions. An example of overlap filtering is shown in Figure 4.7. If the overlap filtering is used, the changes of a block will also changes the neighboring blocks, as shown in Figure 4.8. So, we give an extra rule to decide the tampered areas we detect when the overlap filtering is used in the image. When a *64-pixel block, defined as a square area consisting of four neighboring  $4 \times 4$  blocks*, fail in the authentication signal comparison, we decide the  $4 \times 4$  pixel area evenly straddling the 64-pixel block has been tampered with. A simple example is shown in Figure 4.9.

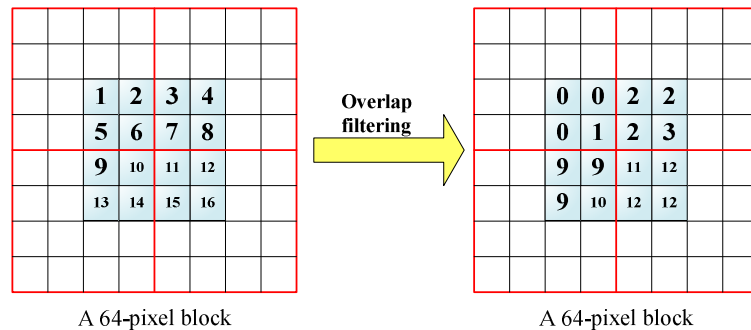


Figure 4.7 An example of overlap filtering straddling a 64-pixel block.

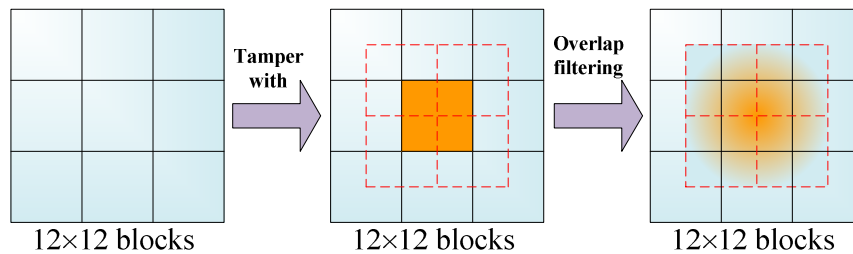


Figure 4.8 The tampered area will be expanded after the overlap filtering.

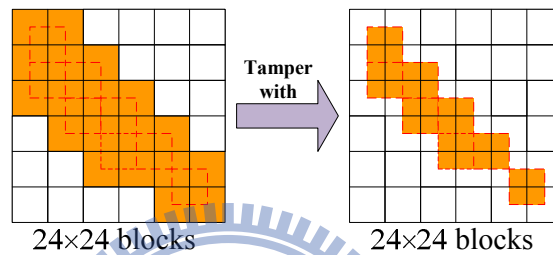


Figure 4.9 When the overlap filtering is used, the tampered area will be map to the small one.

- iv The two-level FCT and quantization will cause the detected area larger than real tampered area. When a block has been tampered with, the DC\_LP coefficient is also changed after the FCT. The DC\_LP coefficient will be used in the level-two FCT later. In this case, if the results of the level-two FCT are quantized, it will cause distortion of all the transformed coefficients by truncating the decimal fraction. So, they cannot recover to the original DC\_LP coefficients. The error is expanded from the block to the macroblock. A simple example is shown in Figure 4.10. To solve this problem, we use 9 LSBs to hide the authentication signal and 6 LSBs to hide an extra signal, e.g., “101010”. Because the highpass coefficients of the erroneously-decided blocks are not changed, we can remove them from the tampered block side by checking the extra signals.

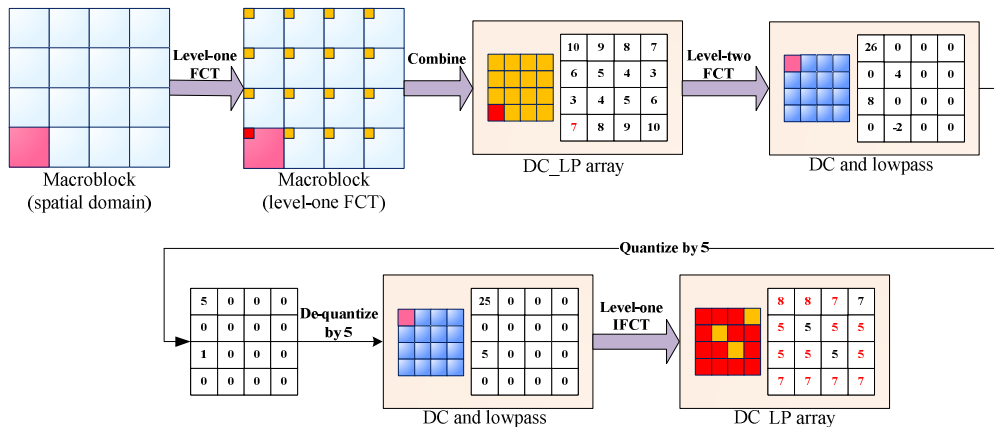


Figure 4.10 When a block has been tampered with, the tampered area will be expanded to other blocks of the macroblock after the re-encoding.

In the authentication signal extraction process, we transform the DC, and lowpass coefficients into DC\_LP coefficients first and extract the authentication signals from the highpass coefficients. Finally, tampered areas are decided by result of comparing the DC\_LP coefficients with the authentication signals.

Because it is still has a probability, though very low, for a block to be authenticated erroneously, we try in this study to “eliminate” such blocks if they appear to be different from their 8 neighbors in the authentication result, called *isolated blocks*. For this, a “filter” is proposed, which has the following filtering function: (1) if a block has 8 neighboring blocks all being decided as tampered blocks, it is also decided as a tampered block; and (2) if its 8 neighboring blocks are decided as un-changed block, the block is also decided as an un-changed block,. The above-described authentication signal embedding process is illustrated by Figure 4.11. And the authentication signal extraction and image authentication process is illustrated by Figure 4.12.

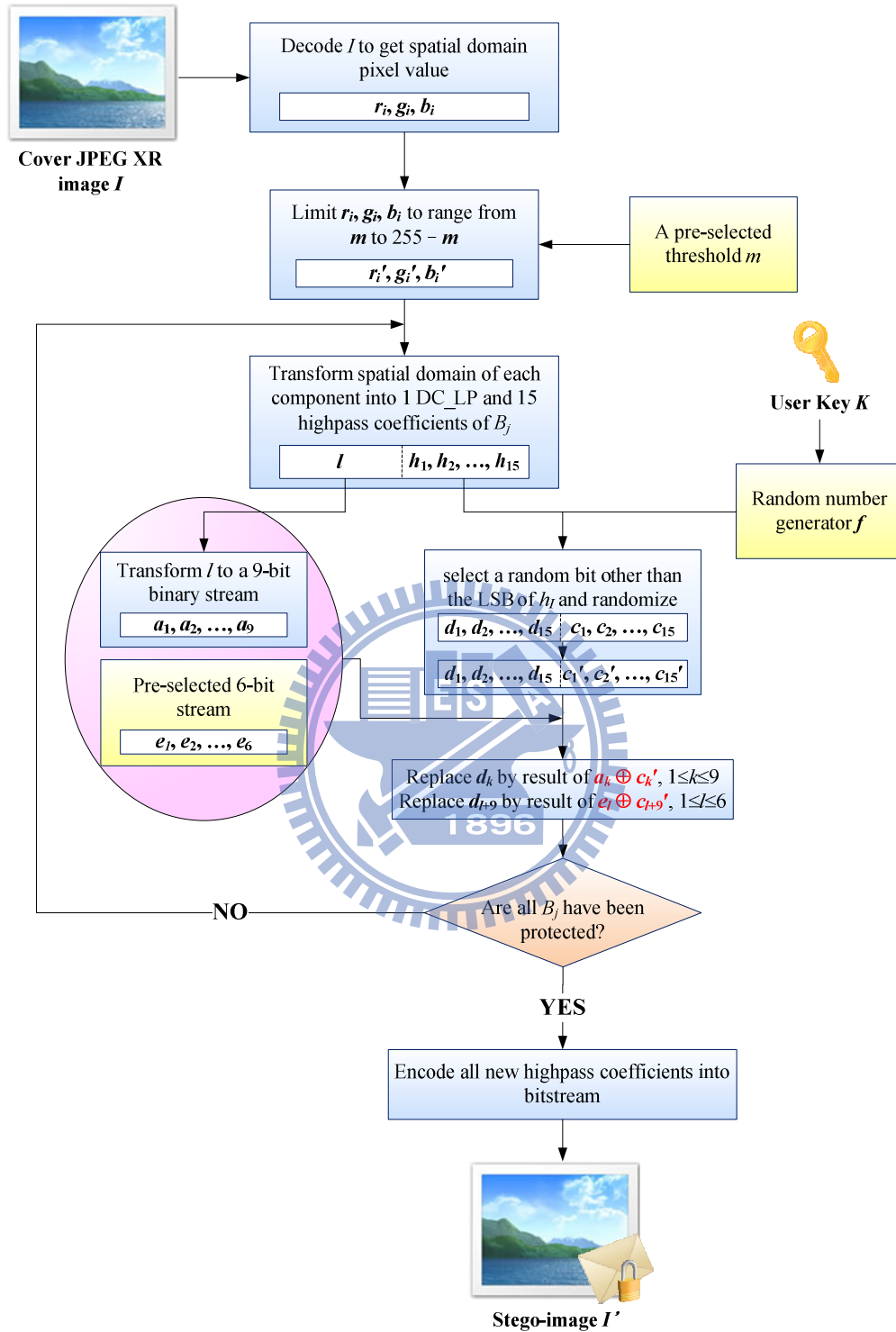


Figure 4.11 The flowchart of the proposed authentication signals embedding process.

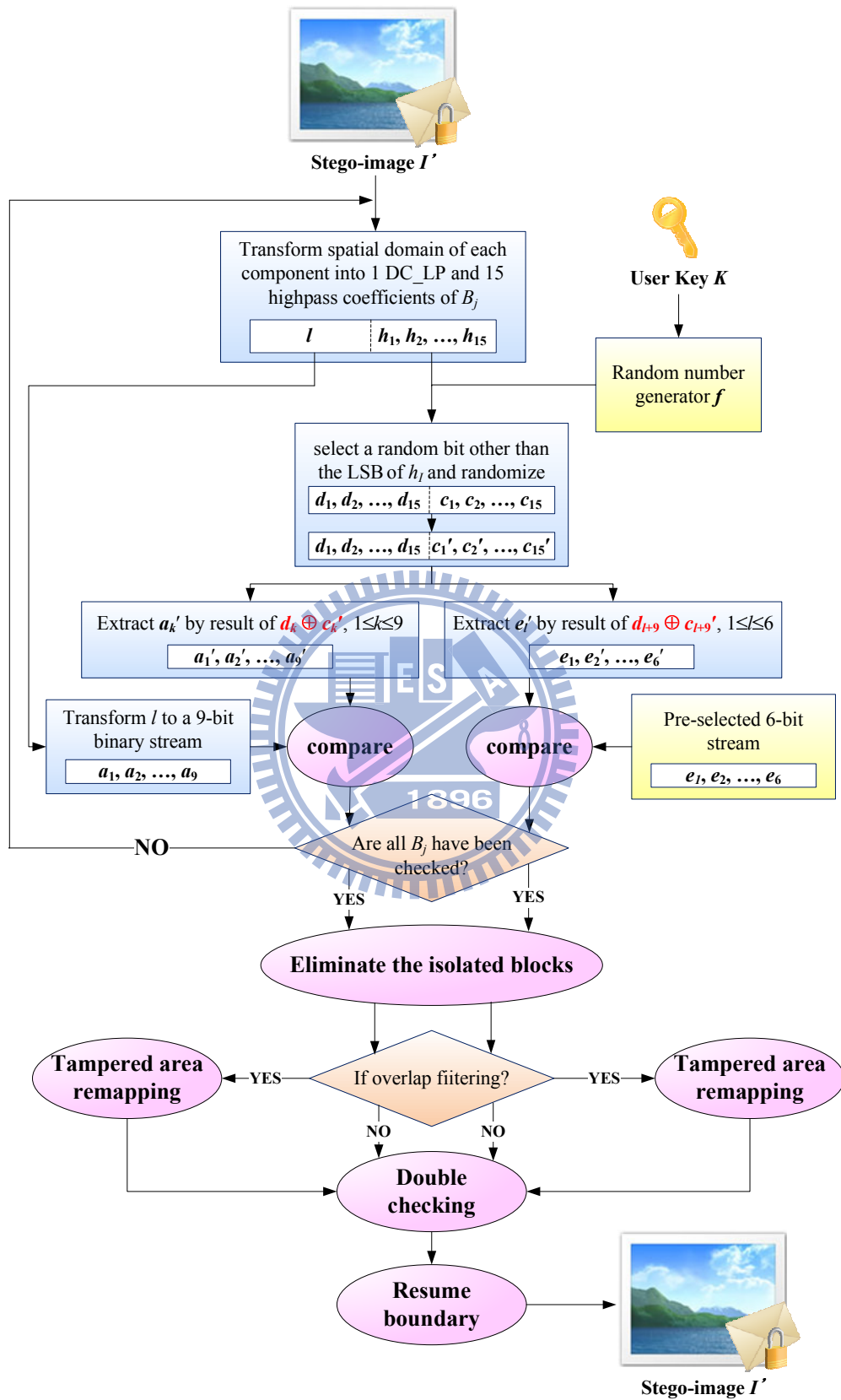


Figure 4.12 The flowchart of the proposed authentication process.

## 4.2.2 Embedding of authentication signals

The detail of the proposed authentication signal creation and embedding process is described as an algorithm in the following.

**Algorithm 4.1. Authentication signal creation and embedding for a JPEG XR image.**

**Input:** a user key  $K$ , a random number generator  $f$ , a threshold  $m$ , a pre-selected 6-bit binary stream  $e_1, e_2, \dots, e_6$ , and a cover JPRG XR image  $I$  with width  $W$  and height  $H$  and  $N$  macroblocks.

**Output:** a stego-JPEGX XR image  $I'$  with authentication signals embedded.

**Steps:**

*Step 1.* Decode  $I$  to get the colors, denoted by  $r_i, g_i,$  and  $b_i$ , of all the pixels in  $I$  in the spatial domain, where  $1 \leq i \leq W \times H$ .

*Step 2.* Limit the value  $v$  of each of the  $r_i, g_i$  and  $b_i$ , where  $1 \leq i \leq W \times H$ , by the following way.

- (a) If  $m > v$ , then set  $v = m$ .
- (b) If  $v > 255 - m$ , then set  $v = 255 - m$ .

*Step 3.* For each block  $B_j$  of macroblock  $M_i$  of each component into which authentication signals are to be embedded, perform the following steps, where  $1 \leq i \leq N, 1 \leq j \leq 16$ .

3.1 Transform the pixels' colors  $r_i, g_i,$  and  $b_i$  in the spatial domain into a DC\_LP coefficient  $l$  and 15 highpass coefficients  $h_1, h_2, \dots, h_{15}$  by the encoder of the JPEG XR standard.

3.2 Transform  $l$  into a 9-bit binary stream  $a_1, a_2, \dots, a_9$  as an authentication signal and let  $e_1, e_2, \dots, e_6$  (e.g., "101010") as an extra authentication signal.

3.3 Use  $K$  and  $f$  to select a random bit  $c_k$  other than the LSB  $d_k$  of each highpass coefficient  $h_k$  and randomize  $c_1, c_2, \dots, c_{15}$  to get a new sequence  $c_1', c_2', \dots, c_{15}'$ , where  $1 \leq k \leq 15$ .

3.4 Replace  $d_k$  by the value of  $a_k \oplus c_k'$ , where  $1 \leq k \leq 9$ .

3.5 Replace  $d_{k+9}$  by the value of  $e_k \oplus c_{k+9}'$ , where  $1 \leq k \leq 6$ .

*Step 4.* Encode all new highpass coefficients into bitstreams according to the JPEG XR standard to generate a stego-JPEG XR image  $I'$ .

### 4.2.3 Extraction of authentication signals

The detail of the proposed authentication process is described as an algorithm in the following.

#### Algorithm 4.2. Authentication of a JPEG XR image.

**Input:** A stego-JPEGX XR image  $I'$  with width  $W$  and height  $H$ , and  $N$  macroblocks, a user key  $K$ , a random number generator  $f$ , and a pre-selected 6-bit binary stream  $e_1, e_2, \dots, e_6$  with  $K, f$  and  $e_1, e_2, \dots, e_6$  all being identical to those used in Algorithm 4.1.

**Output:** An authenticated image  $A$ .

#### Steps:

*Step 1.* For each block  $B_j$  of macroblock  $M_i$  of each component, check the authentication signal and the extra authentication signal by performing the following steps, where  $1 \leq i \leq N, 1 \leq j \leq 16$ .

1.1 Extract the DC\_LP coefficient  $l$  and 15 highpass coefficients  $h_1, h_2, \dots, h_{15}$  of each  $B_j$ .

1.2 Use  $K$  and  $f$  to select a random bit  $c_k$  other than the LSB  $d_k$  of each highpass coefficient  $h_k$  and randomize  $c_1, c_2, \dots, c_{15}$  to get a new sequence  $c_1', c_2', \dots, c_{15}'$ , where  $1 \leq k \leq 15$ .



1.3 Authenticate the authentication signals of  $B_j$  by performing the following steps.

- (a) Set  $a_k$  to be the value  $d_k \oplus c_k'$ , where  $1 \leq k \leq 9$ .
- (b) Combine  $a_1, a_2, \dots, a_9$  and transform the result to get a decimal number  $a$ .
- (c) Compare  $a$  with the DC\_LP coefficient  $l$ ; if not equal, report  $B_j$  as failing to pass authentication signal checking.

1.4 Authenticate the extra authentication signal of  $B_j$  by performing the following steps.

- (a) Set  $e_k'$  to be the value  $d_{k+9} \oplus c_{k+9}'$ , where  $1 \leq k \leq 6$ .
- (b) Compare  $e_1', e_2', \dots, e_6'$  with  $e_1, e_2, \dots, e_6$ , respectively; if not all equal, report  $B_j$  as failing to pass extra authentication signal checking.

Step 2. Eliminate the *isolated* blocks of the result of Step 1 by performing the following steps.

- (a) Eliminate the isolated blocks which are authenticated to be different from their 8 neighboring blocks by the following way.
  - i If a block  $B$  has 8 neighboring blocks all failing to pass the authentication signal checking, decide  $B$  to fail as well to pass the authentication signal checking.
  - ii If a block  $B'$  has 8 neighboring blocks all passing the authentication signal checking, decide  $B'$  to pass as well the authentication signal checking.
- (b) Eliminate the isolated blocks which are authenticated to be different from their 8 neighboring blocks by the following way.
  - i If a block  $B''$  has 8 neighboring blocks all failing to pass the extra

authentication signal checking, decide  $B''$  to fail as well to pass the extra authentication signal checking.

- ii If a block  $B'''$  has 8 neighboring blocks all passing the extra authentication signal checking, decide  $B'''$  to pass as well the extra authentication signal checking.

*Step 3.* If no overlap filtering operation has been conducted, then go to Step 4; otherwise, revise the result of Step 2 by performing the following steps.

- (a) For each 64-pixel block  $B$  (consisting of four neighboring  $4 \times 4$  blocks) which fails in the authentication signal comparison, reduce the tampered area from  $B$  to a  $4 \times 4$  pixel area in the center of  $B$  and evenly straddling  $B$  in two dimensions.
- (b) For each 64-pixel block  $B'$  (consisting of four neighboring  $4 \times 4$  blocks) which fails in the extra authentication signal comparison, reduce the tampered area from  $B'$  to a  $4 \times 4$  pixel area in the center of  $B$  and evenly straddling  $B$  in two dimensions.

*Step 4.* Decide the tampered areas by double checking the authentication signals and the extra authentication signals. If an area fails in both the authentication signal and the extra authentication signal checking, then decide the area to have been tampered with.

*Step 5.* Resume the result of Step 4 on the image boundary, by performing the following steps.

- (a) If  $W$  is not a multiple of 16, then resume the rightmost-column macroblocks as un-tampered areas if any decided to be so.
- (b) If  $H$  is not a multiple of 16, then resume the bottom-row macroblocks as un-tampered areas if any decided to be so.

*Step 6.* Mark the tampered areas of  $I'$  and generate an authenticated image  $A$ .

## 4.3 Security Consideration

### 4.3.1 Issues of security of proposed method

When the stego-image is compressed to yield a great distortion, the authentication signals embedded into the stego-image may be destroyed. As a result, the stego-image will be seen as a tampered image. However, in some applications, they may hope that even though the stego-image has a great distortion after the compression, the authentication signals can still be extracted correctly for the authentication.

### 4.3.2 Proposed security enhancement measures

In this section, we describe a scheme we propose to hide authentication signals into the second LSB's of the highpass coefficients to improve the survival rate of the authentication signals. The proposed scheme for replacing the second LSB is described in the following.

Let  $s$  be the second LSB,  $b$  be the LSB of the highpass coefficient, and  $d$  be a bit which will be embedded into  $s$ .

- i If  $s$  is equal to  $d$ , then do nothing.
- ii If  $s$  is 0 and  $d$  is 1, then replace  $b$  by 0 and  $s$  by 1.
- iii If  $s$  is 1 and  $d$  is 0, then replace  $b$  by 1 and  $s$  by 0.

By the above way, the distortion will be less than just replacing only the second LSB directly, as can be figured out. An example is shown in Figure 4.13. The experimental result of the authentication signals survival rate testing will show in the next section.

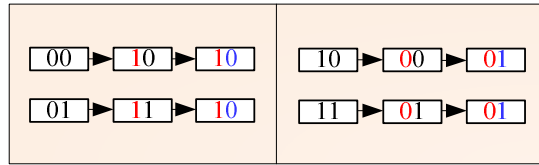


Figure 4.13 A simple example of replacing the second LSB's.

## 4.4 Experimental Results

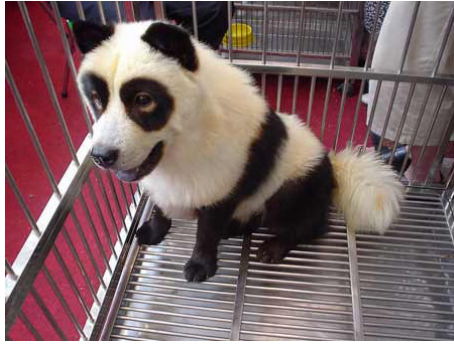
In our experiments, the proposed authentication signal embedding and authentication algorithms were implemented using Microsoft Visual C++. The JPEG XR images can be opened and displayed by a Windows Live Viewer. We created JPEG XR images by using the Adobe Photoshop software.

One result of the experiments is illustrated as follows. A test to avoid the authentication signal being destroyed by overflow pixel values after re-encoding the stego-image is shown in Figure 4.14. The authentication signals were embedded into the image “Two\_dogs” and different ways are used for authentication. The result is shown in Figure 4.15.

Figure 4.16 shows the result of the authentication signals survival rate testing by hiding authentication signals into different last significant bits. Some other results are shown in Figures 4.17 and 4.18.

## 4.5 Summary

In this chapter, a method of block-based authentication of JPEG XR images by comparison of DC and low-pass frequency coefficients has been proposed. The method takes the DC\_LP coefficients for use as the authentication signals, and hides the authentication signals into the highpass coefficients of each block. In this way, every frequency bands are under protection; the changes in each frequency band will be detected. Experimental results have shown the feasibility of the proposed method.



(a)



(b)



(c)

Figure 4.14 A test to avoid the authentication signals destroying by overflowed pixel value after re-encoding the stego-image. (a) The cover image. (b) The authentication result without restricting the range of pixel value. (c) The authentication result by using the threshold  $m$  to restrict the range of pixel value.

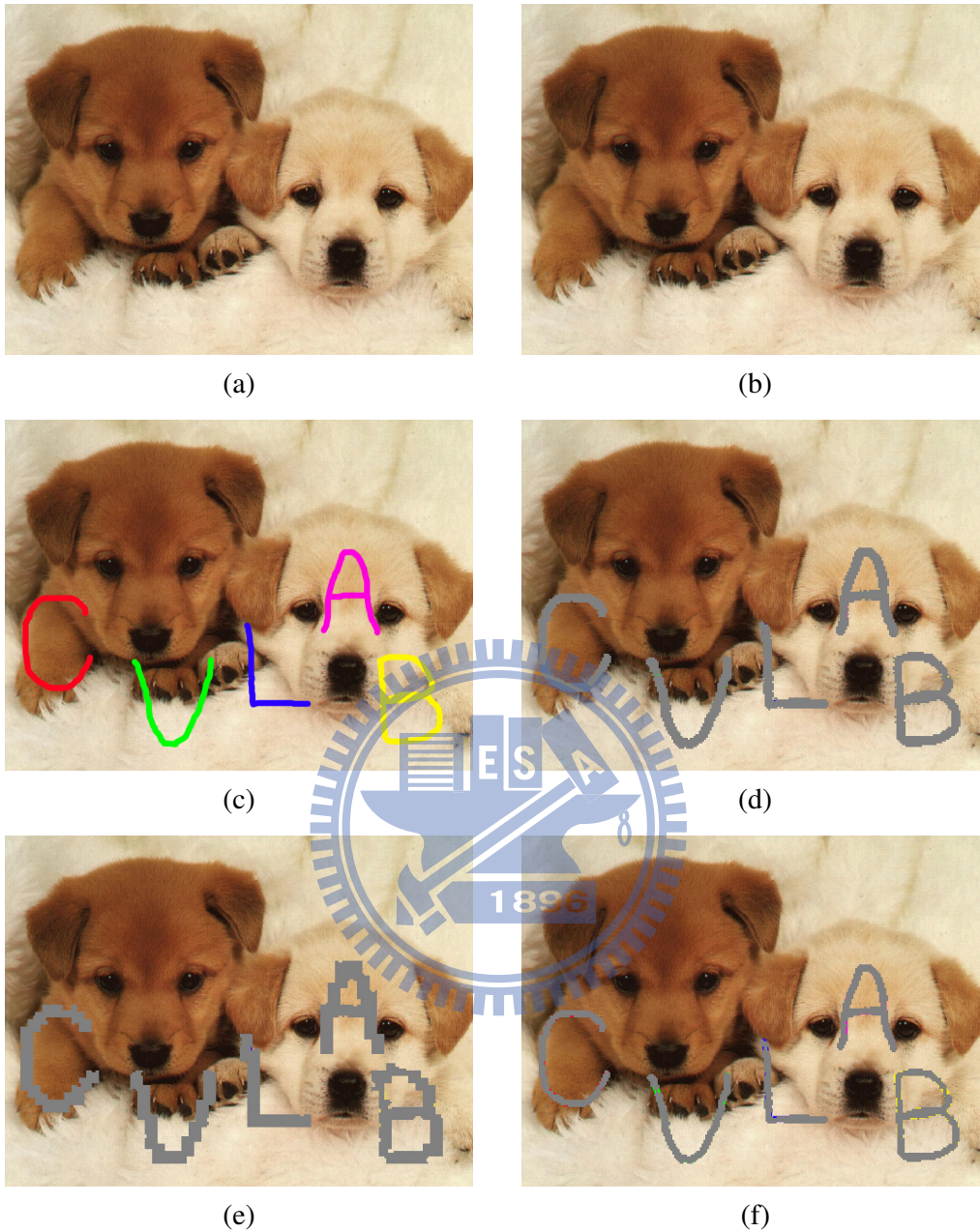


Figure 4.15 The authentication signals is embedded into the image “Two\_dogs” and different ways are used for authentication. (a) The cover image “Two\_dogs.” (b) A stego-image into which the authentication signals are embedded with PSNR 39.3165. (c) A tampered image of (b). (d) The authenticated image without using the re-mapping algorithm when overlap filtering is used. (e) The authenticated image without using the extra authentication signals. (f) The final authenticated image.



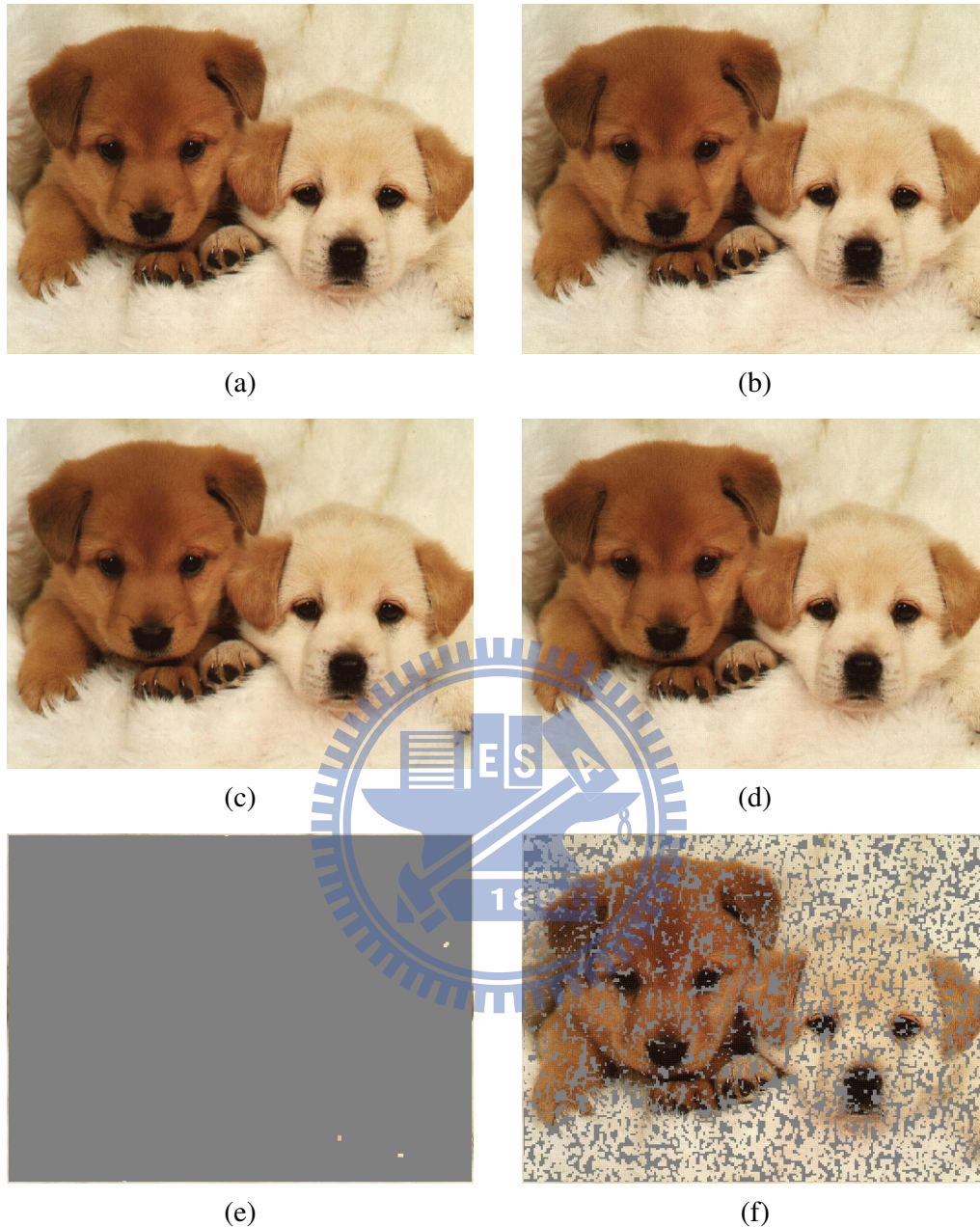


Figure 4.16 A test of the authentication signals survival rate by hiding authentication signals into different last significant bits. (a) A stego-image by hiding authentication signals into LSB (289k bytes). (b) A stego-image by hiding authentication signals into the second LSB (347k bytes). (c) The compressed image of (a) with 124k bytes. (d) The compressed image of (b) with 220k bytes. (e) The authenticated image of (c). (f) The authenticated image of (d).

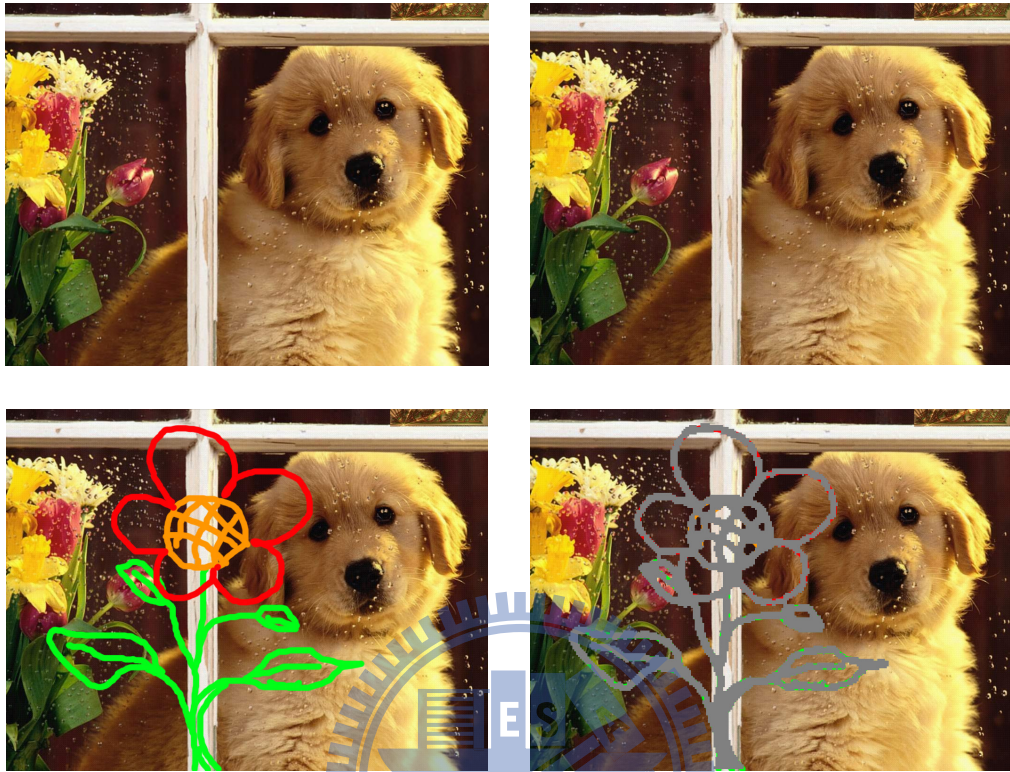


Figure 4.17 More authentication test for another image. (a) A cover image. (b) A stego-image into which the authentication signals are embedded with PSNR 38.6288. (c) A tampered images of (b). (d) The final authenticated image.

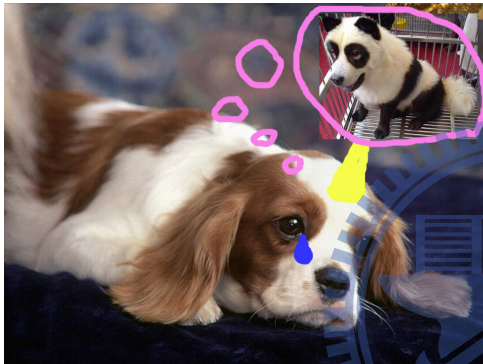




(a)



(b)



(c)



(d)

Figure 4.18 More authentication test for a third image. (a) A cover image. (b) A stego-image into which the authentication signals are embedded with PSNR 39.0006. (c) A tampered images of (b). (d) The authenticated image.

# **Chapter 5**

## **Copyright Protection by Removable Visible Watermarking Based on DC Parameter Manipulation in Frequency Domain**

### **5.1 Introduction**

With the growth of the Internet, exchanging images through the Internet become more frequent nowadays. However, images transmitted on the Internet may be illicitly intercepted, reproduced, or utilized. To solve this problem of copyright protection for the JPEG XR image, we propose a removable visible watermarking method based on DC parameter manipulation of each block of the JPEG XR image.

The major idea of the proposed method will be described in Section 5.1.2. The detailed visible watermark insertion and removing processes will be given in Sections 5.2.2 and 5.2.3, respectively. In addition, some security enhancement measures for the proposed method will be proposed in Section 5.3.2. And some experimental results will be shown in Section 5.4. Finally, we will give a brief summary of this chapter in Section 5.5.

#### **5.1.1 Problem definition**

The JPEG XR images may be illicitly reproduced or utilized by malicious people. They may claim that he/she has the copyright of the image. In this way, if the owner

cannot prove the copyright of the image, then it means that his/her copyright has been already violated. The aim of the proposed method is to insert a visible watermark into a cover image by using a secret key. And the watermark can be removed by the correct key later. By this way, the owner can prove the copyright of the image. Therefore, the main problem of the visible watermarking technique via JPEG XR images is how to design techniques for visible watermark insertion and removing.

## 5.1.2 Major idea of proposed method

The first component of a image stands for the luminance of an image. And the DC coefficient means the magnitude of the *overall color impression* of the block in the frequency domain. By these properties, modifying the first component DC coefficient of a block will also change the luminance of the block. We insert accordingly a visible watermark into an image by changing the luminance of the corresponding watermarked area to make the area look different from its neighborhood (or even from the entire image). This can be achieved by modifying the DC coefficient of each block, and for this we use certain division operations.

In this method, no matter whether the DC coefficient is a positive integer or a negative one, they will get closer to 0 after being divided by a constant integer. A block with high luminance will so be changed to one with a lower luminance, vice versa.

A method for hiding data in the LSB of the highpass coefficient is also proposed to achieve the goal of losslessly removing the visible watermark. The main idea of the method is left-shifting highpass coefficients before embedding secret data into the LSBs of them. The highpass coefficient will be left-shifted by one bit to a new one first. Then, the LSB of the new highpass coefficient will be replaced to by the secret data. The embedded data can be extracted easier from the LSB of the highpass

coefficient. And the highpass coefficient can be recovered to be the original one by one-bit right-shifting. A simple example is shown in Figure 5.1.

The proposed method is also implemented in the *after-encoding and before-decoding* fashion. A simple flowchart of the removable visible watermarking process is shown in Figure 5.2. A special *JPEG XR file data extractor*, a special *visible watermark inserter*, a special *visible watermark remover*, and a special *JPEG XR file data encoder* are also designed in this study.

## **5.2 Proposed Method for Removable Visible Watermarking**

### **5.2.1 Removable Visible Watermarking Based on DC Parameter Manipulation**

Because block is the unit for inserting the watermark in the proposed method, we need to scale the watermark into an appropriate size for insertion. The scaled width and height of the watermark are smaller than the width and height of the cover image divided by 4, which means that a pixel of the watermark corresponds to a block of the cover image.

After scaling the watermark, the proposed method de-quantizes all the coefficients of each frequency band by their corresponding quantization value, and resets all the quantization values of each frequency band to 1 in each macroblock. After this step, all the coefficients are de-quantized and the image has no change. Then, we transform the DC and the lowpass coefficients of each macroblock into the previously-mentioned DC\_LP arrays for block-based watermarking.

Later, calling a block into which the watermark will be inserted a *watermarking block*, we divide the DC\_LP coefficient of a watermarking block by a pre-selected constant value  $q$  and quantize the highpass coefficients by their corresponding quantization values for inserting the watermark. In order to losslessly remove the watermark, we left-shift the highpass coefficients by one bit, and then embed the remainder and the original highpass quantization values in the LSB's of the highpass coefficients. The embedding order of the highpass coefficients is randomized by a secret key. In addition, a pre-selected number  $m$  is used to left-shift the DC\_LP coefficients by  $m$  bits for controlling the luminance of the watermarking blocks. A simple diagram is shown in Figure 5.3. Finally, we encode the coefficients into bitstreams according to the JPEG XR standard as a watermarked image. The above-described visible watermark insertion process is illustrated by Figure 5.4.

In the watermark removing process, we transform the DC and lowpass coefficients of each macroblock into DC\_LP arrays first, and then extract the embedded remainder and the embedded highpass quantization value from the LSBs of the highpass coefficients of the watermarking block by using the watermark and the secret key. After extracting the remainder and the highpass quantization value, we right-shift the DC\_LP coefficient and multiply the result by  $q$ , and then add the remainder the new DC\_LP coefficient to get the original DC\_LP coefficient. Also, we right-shift all the coefficients of the block by 1 bit, and then multiply the result by the extracted highpass quantization value to get the original highpass coefficients. Finally, we encode the coefficients into bitstreams according to the JPEG XR standard as a recovered image. The above-described visible watermark insertion process is illustrated by Figure 5.5.

In the proposed method, when the overlap filtering is used in a cover image, the

watermark will become unclear in the edge area. That is because the watermarking block will be expanded after the overlap filtering. In order to get the clear watermark back in this case, we propose a method. When a cover image is used to insert a watermark, we check if the overlap filtering is conducted in the cover image first; if so, the cover image will be decoded to get the data in the spatial domain, which then are re-encoded to be a new image without using the overlap filtering before watermark insertion. Some experimental result of this way will be shown at the rear of this chapter.

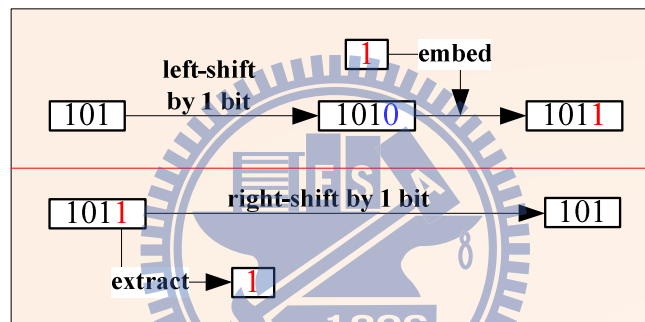


Figure 5.1 A simple example for proposed data hiding method.

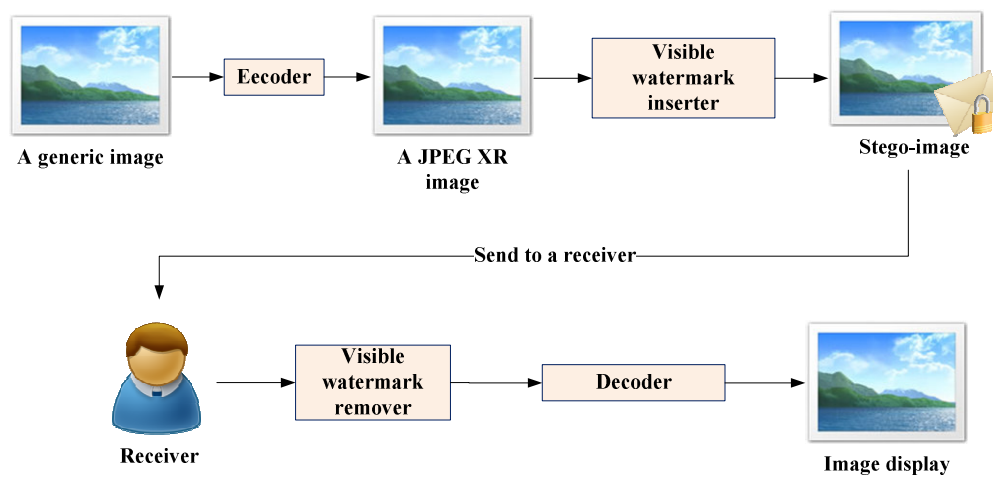


Figure 5.2 The flowchart of visible watermark insertion and removing via the un-decoded JPEG XR file format.

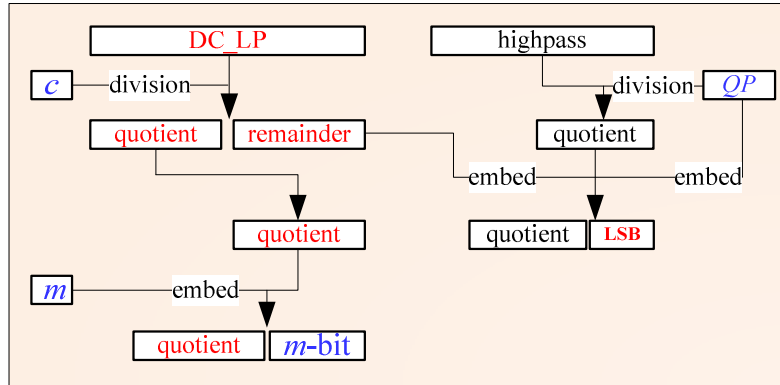


Figure 5.3 The proposed method for watermarking and hiding data into the LSB of highpass coefficient.

## 5.2.2 Process of embedding visible watermark

The detail of the proposed visible watermark insertion process is described as an algorithm in the following.

### Algorithm 5.1. Visible watermark insertion into a JPEG XR image.

**Input:** a user key  $K$ , a random number generator  $f$ , a selected watermark  $W$  with the width  $W2$  and the height  $H2$ , a selected constant number  $c$ , a threshold  $m$ , and a cover JPRG XR image  $I$  with the width  $W1$  and the height  $H1$  and  $N$  macroblocks.

**Output:** a stego-JPEGX XR image  $I'$  with a visible watermark inserted.

**Steps:**

*Step 1.* If  $W2 > \frac{W1}{4}$  or  $H2 > \frac{H1}{4}$ , then

scale  $W$  to a new size of  $W2'$  by  $H2'$ , where  $W2' \leq \frac{W1}{4}$  and  $H2' \leq \frac{H1}{4}$ ;

otherwise, continue.

*Step 2.* De-quantize all coefficients of each frequency band of the first component

by their corresponding quantization value.

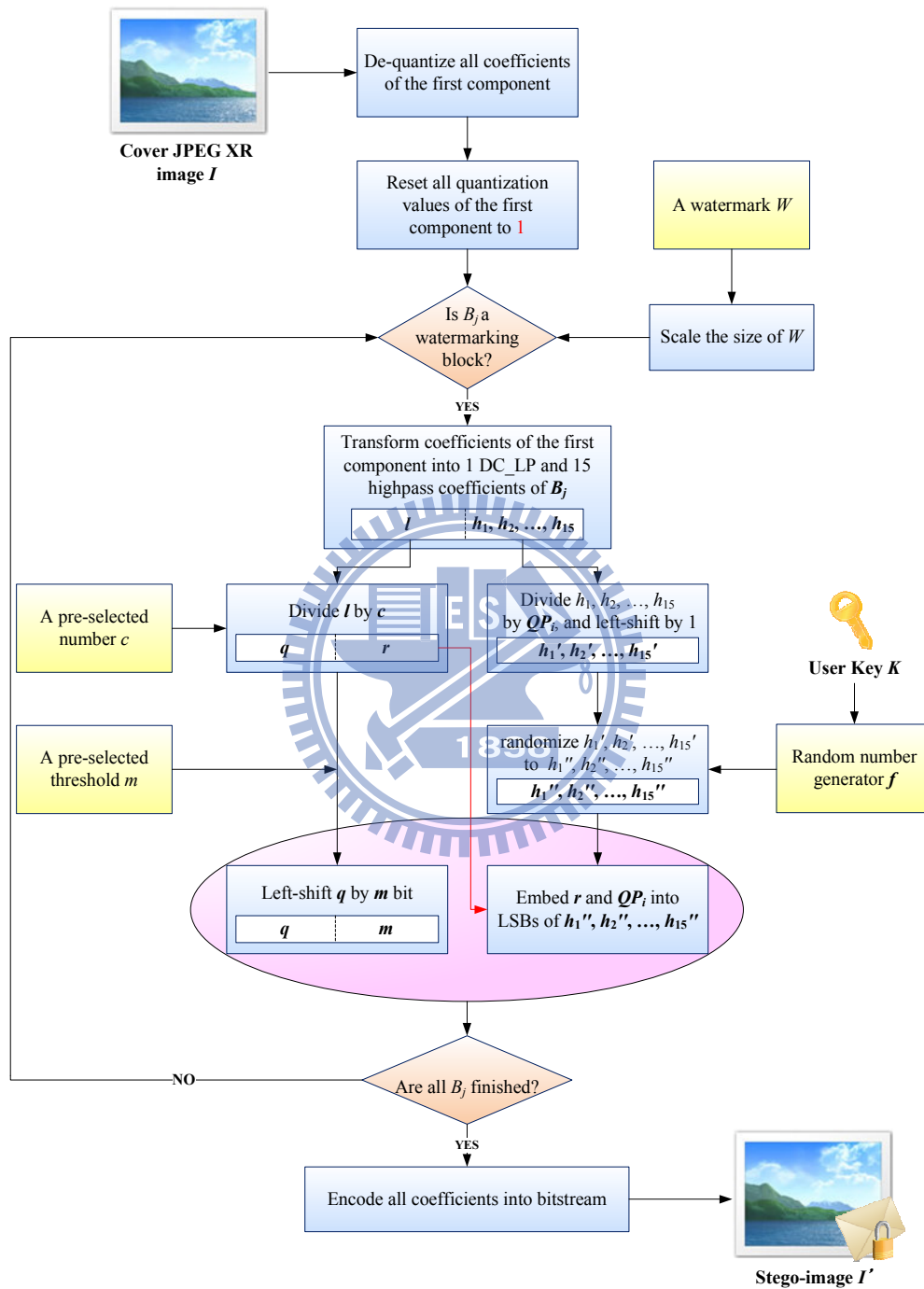


Figure 5.4 The flowchart of the proposed visible watermark insertion process.



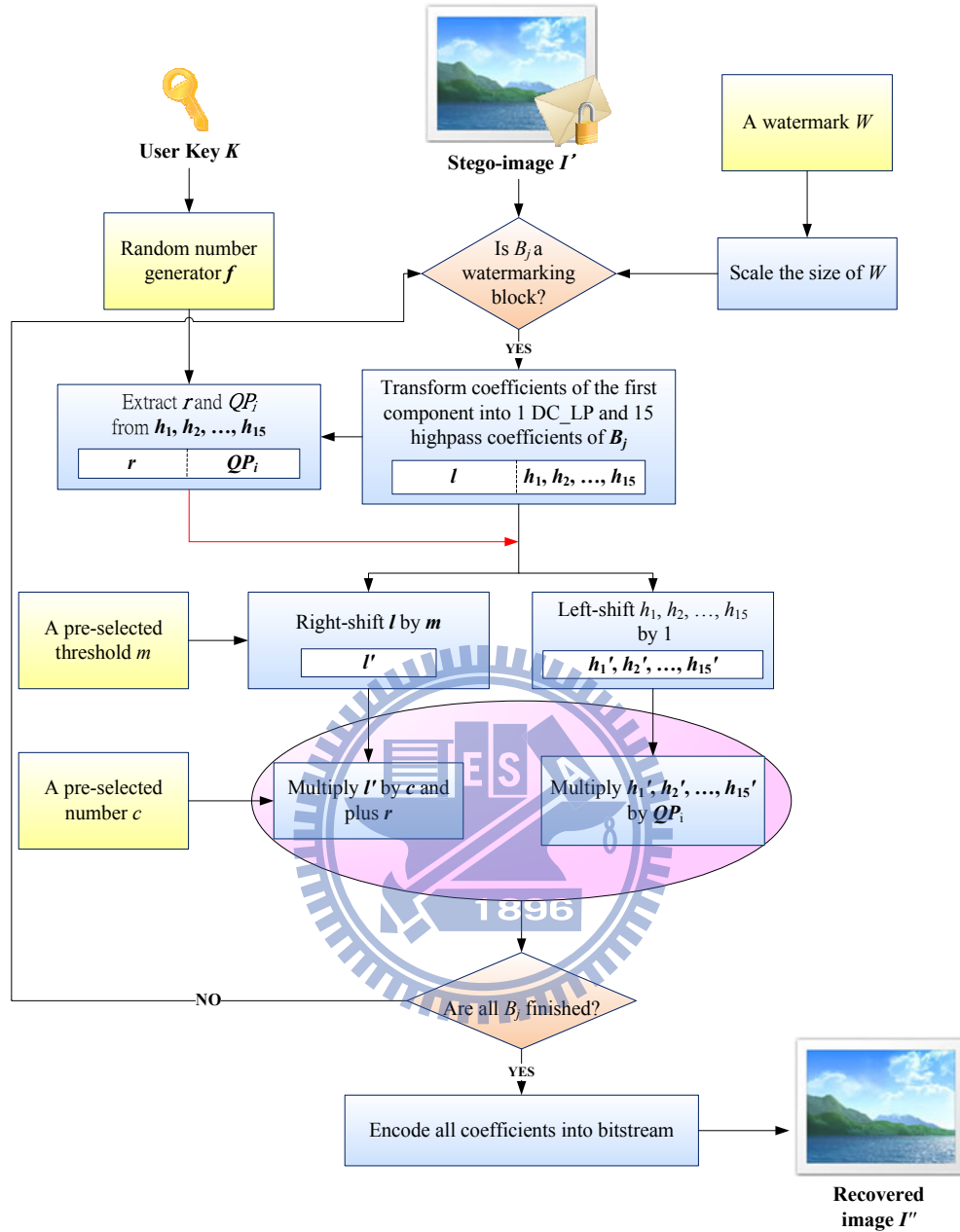


Figure 5.5 The flowchart of the proposed visible watermark removing process.

- Step 3.* Reset all quantization values of each frequency band of the first component to 1 in each macroblock.
- Step 4.* For each block  $B_j$  of macroblock  $M_i$  of the first component into which a visible watermark is to be inserted, perform the following steps, where  $1 \leq i$

$\leq N, 1 \leq j \leq 16$ .

4.1 Check if  $B_j$  is a watermarking block; if so, continue; otherwise, go to the next block.

4.2 Transform the result of Step 3 by the IFCT (the inverse transform of FCT) to get a DC\_LP coefficient  $l$  and 15 highpass coefficients  $h_1, h_2, \dots, h_{15}$  of each  $B_j$ .

4.3 Change the DC\_LP coefficient  $l$  to a new one  $l'$  for inserting the watermark into  $B_j$  by performing the following steps.

- i. Divide  $l$  by  $c$  to get a quotient  $q$  and a remainder  $r$ .
- ii. Left-shift  $q$  by  $m$  bits to be  $l'$  as a new DC\_LP coefficient.

4.4 Embed information into highpass coefficient of  $B_j$  for losslessly removing the watermark by performing the following steps.

- (a) Divide  $h_1, h_2, \dots, h_{15}$  by the quantization value  $QP_i$  of the highpass band of  $M_i$  and left-shift the result by 1 bit to be  $h_1', h_2', \dots, h_{15}'$ .
- (b) Transform  $r$  into a 9-bit binary stream  $r_1, r_2, \dots, r_9$ , and transform  $QP_i$  into a 6-bit binary stream  $q_1, q_2, \dots, q_6$ .
- (c) Use  $K$  and  $f$  to randomize  $h_1', h_2', \dots, h_{15}'$  to get a new sequence  $h_1'', h_2'', \dots, h_{15}''$ .
- (d) Embed  $r_1, r_2, \dots, r_9$  into the LSBs of  $h_1'', h_2'', \dots, h_9''$ , respectively, and embed  $q_1, q_2, \dots, q_6$  into the LSBs of  $h_{10}'', h_{11}'', \dots, h_{15}''$ , respectively.

*Step 5.* Encode all coefficients into bitstreams according to the JPEG XR standard to generate a stego-JPEG XR image  $I'$ .

### 5.2.3 Process of removing visible watermark

The detail of the proposed visible watermark removing process is described as an algorithm in the following.

**Algorithm 5.2. Proposed visible watermark removing process for a JPEG XR image.**

**Input:** A stego-JPEGX XR image  $I'$  with the width  $W1$  and the height  $H1$  and  $N$  macroblocks, a selected watermark  $W$  with the width  $W2$  and the height  $H2$ , a user key  $K$ , a random number generator  $f$ , a selected constant number  $c$ , and a threshold  $m$  with  $K$ ,  $f$ ,  $q$ ,  $m$  and  $W$  being identical to those used in Algorithm 5.1.

**Output:** a recovered image  $I''$

**Steps:**

*Step 1.* If  $W2 > \frac{W1}{4}$  or  $H2 > \frac{H1}{4}$ , then,  
 scale  $W$  to a new size of  $W2'$  by  $H2'$ , where  $W2' \leq \frac{W1}{4}$  and  $H2' \leq \frac{H1}{4}$ ;

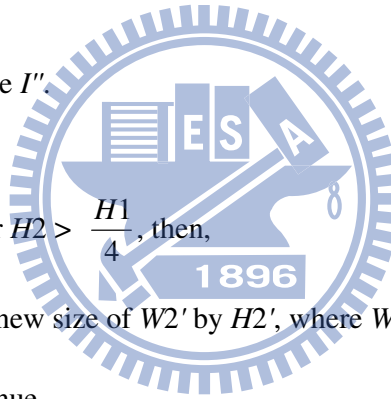
otherwise, continue.

*Step 2.* For each block  $B_j$  of macroblock  $M_i$  of the first component, remove the visible watermark losslessly by performing the following steps, where  $1 \leq i \leq N$ ,  $1 \leq j \leq 16$ .

2.1 Check if  $B_j$  is a watermarking block; if so, continue; otherwise, go to the next block.

2.2 Extract a DC\_LP coefficient  $l$  and 15 highpass coefficients  $h_1, h_2, \dots, h_{15}$  of each  $B_j$ .

2.3 Extract  $r$  and  $QP_i$  from the LSBs of  $h_1, h_2, \dots, h_{15}$  by using the key  $K$  and  $f$ .



2.4 Recover  $l$  to be the original DC\_LP coefficient by performing the following steps.

- (a) Right-shift  $l$  by  $m$  bit to be  $l'$ .
- (b) Multiply  $l'$  by  $q$ .
- (c) Add the result of (b) by  $r$  to get the original DC\_LP coefficient.

2.5 Recover  $h_1, h_2, \dots, h_{15}$  to be the original highpass coefficients by performing the following steps.

- (a) Right-shift  $h_1, h_2, \dots, h_{15}$  by 1 bit to be  $h_1', h_2', \dots, h_{15}'$ , respectively.
- (b) Multiply  $h_1', h_2', \dots, h_{15}'$  by the value  $QP_i$ , respectively, to get the original highpass coefficients.

*Step 3.* Encode all coefficients into bitstreams according to the JPEG XR standard to generate a recovered image  $I''$ .

## 5.3 Security Consideration

### 5.3.1 Issues of security of proposed method

In the proposed method, two user-selected parameter  $m$  and  $c$  is used to control the luminance of each watermarking block. If a person does not know  $m$  and  $c$ , he/she cannot remove the watermark completely. Here,  $m$  and  $c$  are both small integers. If a malicious attacker knows the algorithms of the proposed method, he/she may observe the result of the recovered image to guess  $m$  and  $c$  by trial and error. By this way, the security of the watermarked image is reduced.

### 5.3.2 Proposed security enhancement measures

In order to improve the security of the proposed method in the aspect of the

above-mentioned problem, the secret key  $K$  is used to randomize  $m$  and  $c$  to be  $m'$  and  $c'$  before they are embedded into highpass coefficients. In this way,  $m$  and  $c$  can only be extracted by a correct secret key. A person who does not know the secret key will have no idea to guess  $m$  and  $c$ . The security of the watermarked image is thus enhanced.

## 5.4 Experimental Results

In our experiments, the proposed visible watermark insertion and removing algorithms were implemented using Microsoft Visual C++. JPEG XR images can be opened and displayed by a Windows Live Viewer. We created JPEG XR images by using the Adobe Photoshop software.

One result of the experiments is illustrated as follows. The watermark used in this study is shown in Figure 5.6. The watermark was inserted into “Two\_dogs” and was losslessly removed later by a secret key, as shown in Figure 5.7. The watermarked images with different watermarking parameters are shown in Figure 5.8. Two other results are shown in Figure 5.9 and Figure 5.10.

## 5.5 Summary

In this chapter, a method for copyright protection of JPEG XR images by block-based removable visible watermarking based on DC\_LP coefficient manipulation has been proposed. The method divides the DC\_LP coefficients by a constant number for inserting a watermark, and hides the remainder into the highpass coefficients of each block. In this way, the visible watermark can be losslessly removed by a secret key. Experimental results have shown the feasibility of the proposed method.



Figure 5.6 A watermark used in this study.

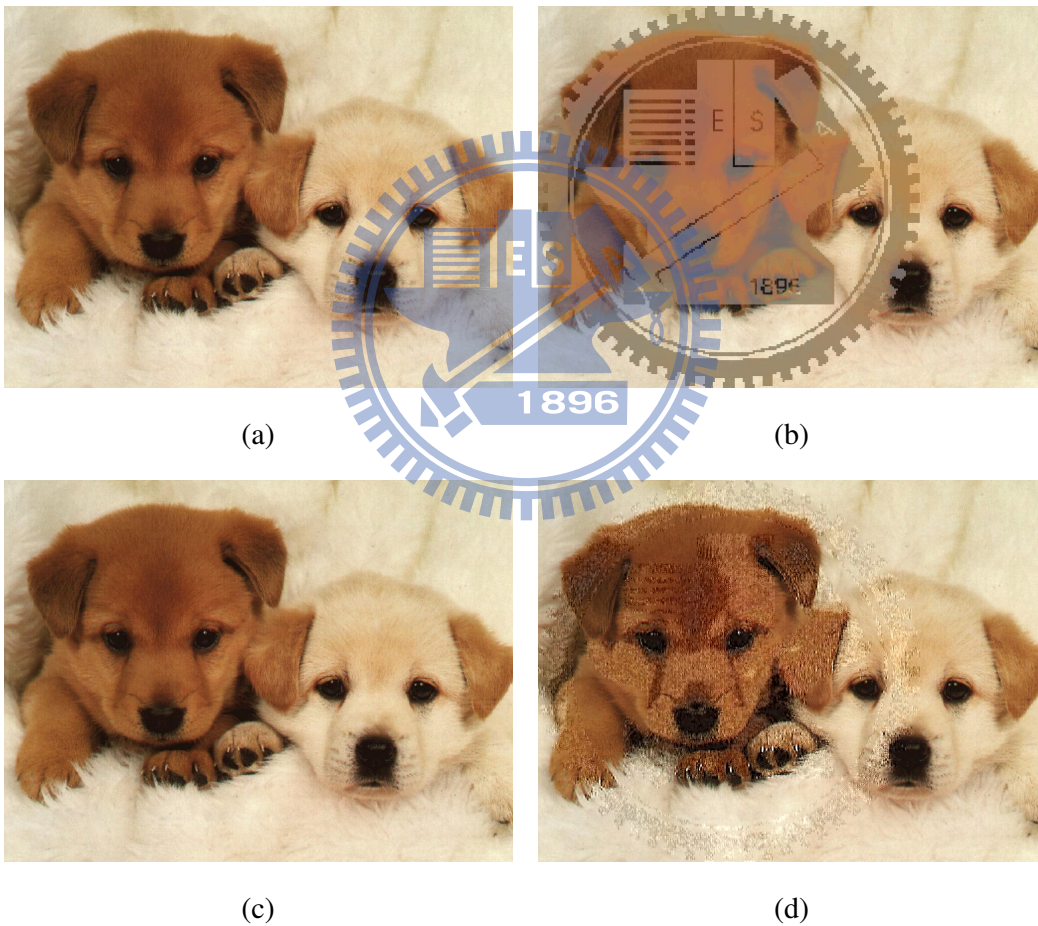


Figure 5.7 The watermark inserted into “Two\_dogs” and will be lossless removed later by a secret key. (a) The cover image “Two\_dogs.” (b) A watermarked image. (c) A recovered image. (d) An image with the watermark removed incompletely with a wrong key.

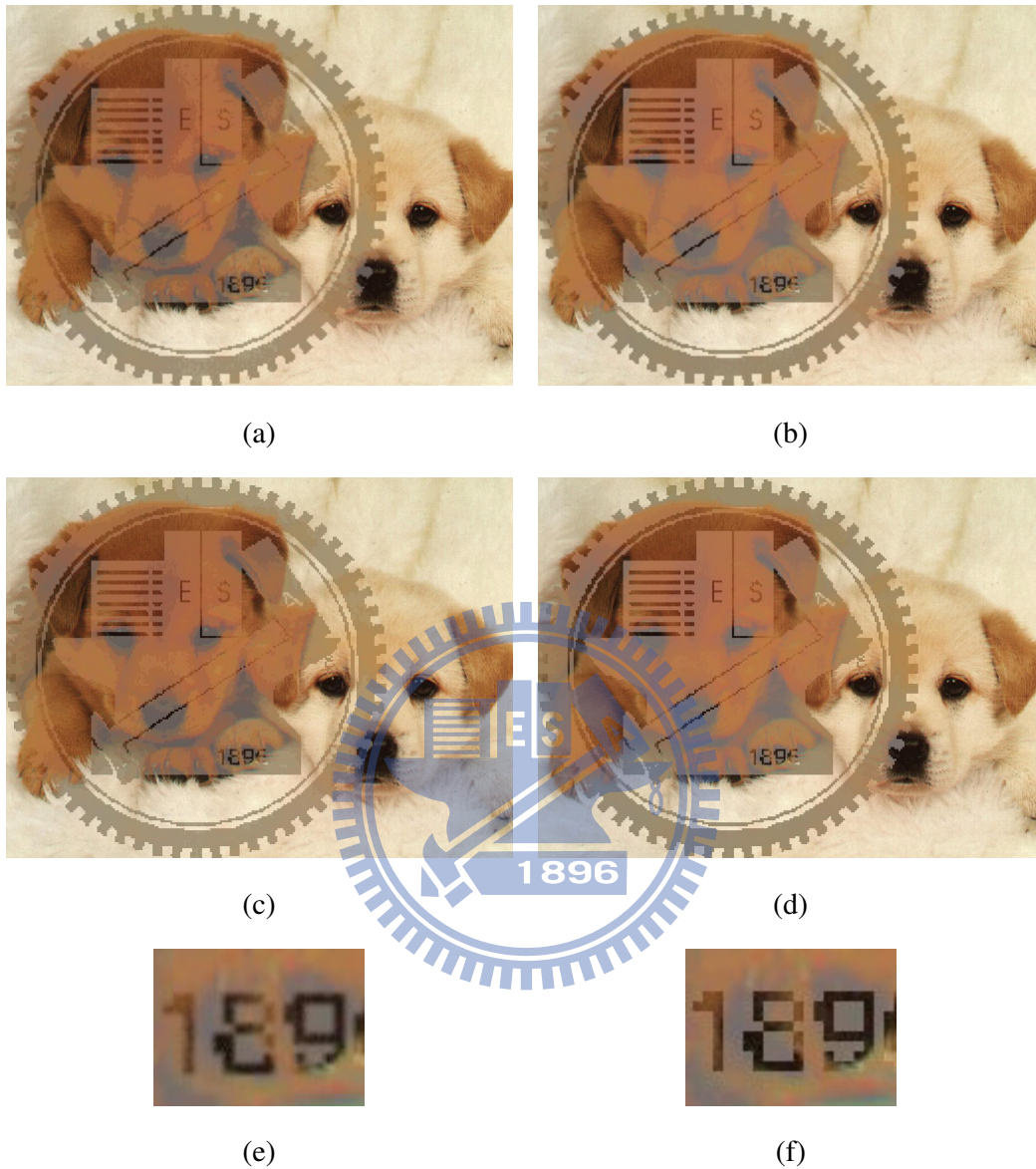
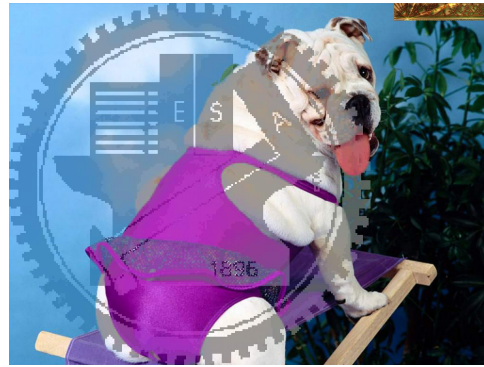


Figure 5.8 The watermarked images with different watermarking parameters. (a) The used threshold  $m = 2$ . (b) The used threshold  $m = 3$ . (c) The used threshold  $m = 2$  and the cover image re-encoded without using the overlap filtering. (d) The threshold  $m = 3$  and the cover image re-encoded without using the overlap filtering. (e) A part of (b). (f) A part of (d).





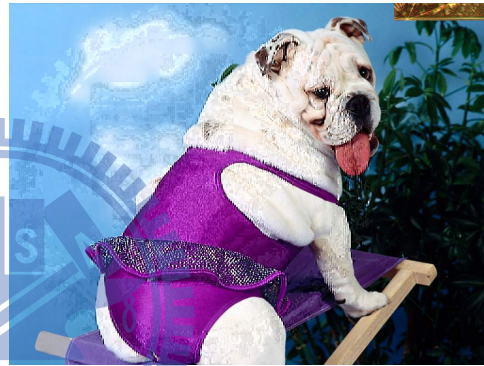
(a)



(b)



(c)



(d)

Figure 5.9 The watermark was inserted into another image and losslessly removed later by a secret key. (a) The cover image. (b) A watermarked image. (c) A recovered image. (d) An image with the watermark removed incompletely with a wrong key.





(a)



(b)

(c)

Figure 5.10 The watermark was inserted into a third image via V channel rather than Y channel and losslessly removed later by a secret key. (a) The cover image. (b) A watermarked image. (c) A recovered image.

# Chapter 6

## Conclusions and Suggestions for Future Works

### 6.1 Conclusions

In this study, we have proposed several methods for data hiding applications via JPEG XR images, such as image covert communication, authentication, and removable visible watermarking.

For cover communication, a method based on encoding the selected quantization parameters of the highpass frequency band to embed a secret message in a JPEG XR image has been proposed. According to the experimental result, embedding of the secret message into a JPEG XR image results in an imperceptible effect. To improve the security of the secret data, we used a one-to-one mapping function controlled by a secret key to transform the indexes of the selected quantization parameters into new ones randomly, so that malicious people cannot easily extract the secret message even though he/she knows the proposed algorithm. In addition, a method to generate a stego-image with the original cover image file size being kept for reducing the possibility of arousing notice from attackers has also been proposed. By the use of the above-mentioned techniques, the secret message can be delivered via a cover JPEG XR image safely on the Internet.

For fidelity or/and integrity verification of JPEG XR images, a block-based authentication method based on embedding DC\_LP coefficients as authentication signals into the highpass coefficients of blocks has been proposed. When a

stego-image is tampered with, the DC\_LP coefficients of the tampered blocks will be altered, and the authentication signals which have been embedded into the highpass coefficients of the tampered blocks will be also changed. Such attacks are authenticated by the proposed method by comparing the extracted authentication signals with those computed from the current JPEG XR image content. In this way, every frequency band is under protection; the changes in each frequency band will be detected.

For removable visible watermarking, a method based on modifying the DC\_LP coefficients in JPEG XR images by certain number division operations has been proposed. Specifically, the remainders of dividing the DC\_LP coefficients by a pre-selected integer number are embedded into the highpass coefficients of blocks for lossless removal of the watermark. By the use of the above-mentioned techniques, the watermark can be inserted into a cover JPEG XR image for copyright protection. And the inserted watermark can be losslessly removed when necessary.

## 6.2 Suggestions for Future Works

According to our experience obtained in this study, several suggestions for future works are listed in the following.

1. Other features of the JPEG XR standard could be found and used to design new data hiding techniques.
2. The multi-channels supported by the JPEG XR standard may be used to design new data hiding techniques.
3. The overflowing pixel values of the image will be adjusted into an available range when the image is displayed. This property may be used to design new lossless data hiding techniques.

4. More robust data hiding techniques via JPEG XR image can be developed for authentication and digital rights management applications.
5. New secret sharing techniques can be developed via JPEG XR images.



# References

- [1] ISO/IEC 29199-2, “JPEG XR image coding system -- Part 2: Image coding specification”.
- [2] D. C. Wu and W. H. Tsai, “A Steganographic Method for Images by Pixel-Value Differencing,” *Pattern Recognition Letters*, Vol. 24, No. 9-10, pp. 1613-1626, June 2003.
- [3] C. W. Lee and W. H. Tsai, “A lossless data hiding method by histogram shifting based on an adaptive block division scheme,” *Pattern Recognition and Machine Vision – in Honor and Memory of the Late Professor King-Sun Fu*, Patrick S. P. Wang (Ed.), River Publishers, Aalborg, Denmark, pp. 1–14., Mar 2010.
- [4] G. L. Huang and W. H. Tsai, “Optimal data hiding in H.264/AVC videos for covert communication,” *Proceedings of 2008 Conference on Computer Vision, Graphics and Image Processing*, Ilan, Taiwan, Aug 2008.
- [5] C.C. Chang, T.S. Chen and L.Z. Chung, “A steganographic method based upon JPEG and quantization table modification,” *Information Science*, Vol. 141, No. 1, pp. 123-138, 2002.
- [6] A. Jain and I. Sen Gupta, “A JPEG compression resistant steganography scheme for raster graphics image,” *TENCON 2007 - 2007 IEEE Region 10 Conference*, Taipei, Taiwan, Oct. 2007.
- [7] J. M. Barton, “Method and apparatus for embedding authentication information within digital Data,” U. S. Patent 5646997, 1997.
- [8] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Lossless generalized-LSB data embedding,” *IEEE Transactions on Image Processing*, Vol. 14, No. 2, 2005,

pp. 253-266.

- [9] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits Systems and Video Technology*, Vol. 13, No. 8, 2003, pp. 890-896.
- [10] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits Systems and Video Technology*, Vol. 16, No.3, 2006, pp. 354-362.
- [11] M. Fallahpour, M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification," *IEICE Electronics Express*, Vol. 4, No. 7, 2007, pp. 205-210.
- [12] C. T. Yang and W. H. Tsai, "Data hiding in PNG images for covert communication," *Technical Report*, Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, June 2009.
- [13] I. S. Lee and W. H. Tsai, "Data hiding in color images by color replacements with reduction of image distortion and change noticeability," *Proceedings of 2009 Conference on Computer Vision, Graphics and Image Processing*, Nantou, Taiwan, Republic of China, Aug 2009.
- [14] C. T. Yang and W. H. Tsai, "Authentication of PNG images by adjusting selected pixel values in spatial domain and its application to data hiding," *Technical Report*, Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, June 2009.
- [15] P. M. Huang and W. H. Tsai, "Copyright protection and authentication of grayscale images by removable visible watermarking and invisible signal embedding techniques: a new approach," *Proceedings of 2003 Conference on Computer Vision, Graphics and Image Processing*, Kinmen, Taiwan, Aug 2003.
- [16] I. S. Lee and W. H. Tsai, "Security protection of software programs by

information sharing and authentication techniques using invisible ASCII control codes,” *International Journal of Network Security*, Vol. 10, No. 1, pp. 1-10, Jan 2010.

[17] Y. C. Chiu and W. H. Tsai, “Copyright protection by watermarking for color images against print-and-scan operations using coding and synchronization of peak locations in discrete Fourier transform domain,” *Proceedings of 2004 Conference on Computer Vision, Graphics and Image Processing*, Hualien, Taiwan, Republic of China, Aug 2004.

[18] T. Y. Liu and W. H. Tsai, “Generic lossless visible watermarking -- a new approach,” *IEEE Transactions on Image Processing*, Vol. 19, No. 5, pp. 1224-1235, Jan 2010.

[19] P. P. Chen and W. H. Tsai, “Copyright protection of palette images by a robust lossless visible watermarking technique,” *Proceedings of 5th Workshop on Digital Archives Technologies*, Taipei, Taiwan, Aug 2006.

