

# 國立交通大學

資訊科學與工程研究所

## 碩士論文

在 H.264/AVC 視訊上做資訊隱藏之研究及  
其應用

A Study on Data Hiding in H.264/AVC Videos and Its  
Applications

研究生：黃冠霖

指導教授：蔡文祥 教授

中華民國 九十七 年六月

在 H. 264/AVC 視訊上做資訊隱藏之研究及其應用  
A Study on Data Hiding in H.264/AVC Videos  
and Its Applications

研究生：黃冠霖

Student : Guan-Lin Huang

指導教授：蔡文祥

Advisor : Wen-Hsiang Tsai



Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

# 在 H.264/AVC 視訊上做資訊隱藏之研究及其應用

研究生：黃冠霖 指導教授：蔡文祥 博士

國立交通大學資訊科學與工程研究所

## 摘要

隨著資訊科技的進步，越來越多的網路視訊應用也隨之發展，其中 H.264/AVC 的視訊檔案是現今應用最廣的視訊格式。本論文針對 H.264/AVC 的視訊檔案，利用資訊隱藏及數位浮水印之技術做秘密傳輸、版權保護及視訊分享之研究與應用。在秘密傳輸方面，我們提出一個大量以及一個最佳化的方法，這兩個方法皆是利用 H.264/AVC 視訊檔案的特性來隱藏資料。其中最佳化的方法可在隱藏資料量、視覺影響及壓縮後檔案大小間取得一個最佳結果。在版權保護方面，對於被客戶端下載的 H.264/AVC 視訊檔案，我們提出一個利用主動式可視浮水印技術及限制影片在特定電腦播放的方法來做版權保護。而在秘密分享的應用方面，我們利用邏輯運算從事秘密視訊資料的分享，並將分享出去的資料回藏至視訊當中，最後將分享後的視訊分發給各參與者保管。在所有參與者所擁有的視訊集合起來之後，即可回復成原本的秘密視訊。最後，我們以實驗結果證明了所提方法之可行性。

# **A Study on Data Hiding in H.264/AVC Videos and Its Applications**

Student: Guan-Lin Huang

Advisor: Wen-Hsiang Tsai

Institute of Computer Science and Engineering  
National Chiao Tung University

## **ABSTRACT**

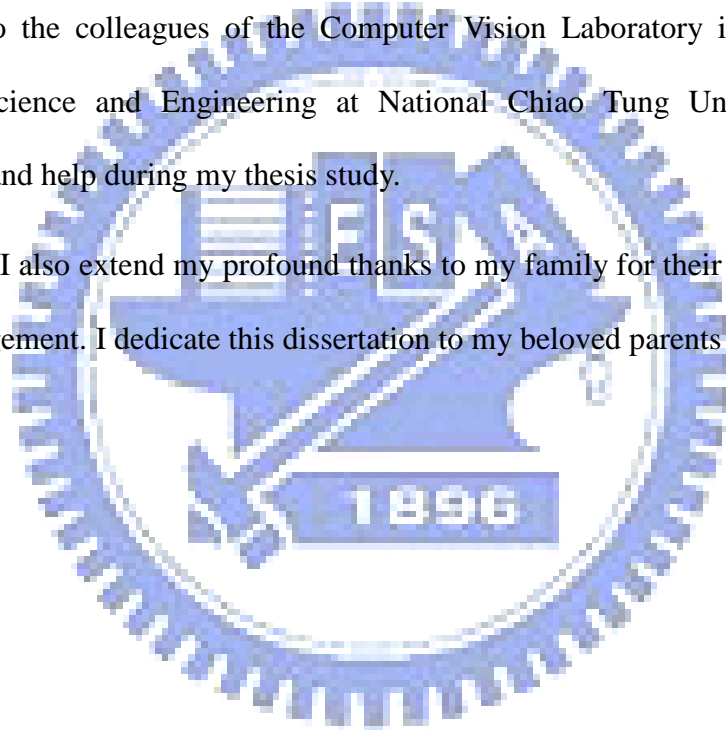
With the advance of information technologies, more and more digital video applications on the internet have been proposed. H.264/AVC videos are used in a wide variety of applications. In this study, several methods for data hiding applications, namely, covert communication, copyright protection, and video sharing, are proposed using H.264/AVC videos as cover media. For covert communication, we propose a large-volume method and an optimal method for hiding secret data in H.264/AVC videos, based on the use of prediction modes and tree structured motion compensation. The optimal method is a tradeoff between hiding capacity, imperceptibility, and low bit rating. For copyright protection, in order to protect the ownership of downloaded videos at the client site, a method using a removable visible watermarking technique with a scheme for display control on specified computers is proposed. For the application of secret sharing, we share the data of the prediction modes of a secret video based on exclusive-OR operations and hide the resulting share data into the prediction modes of cover videos. The resulting share videos are then distributed to participants to keep. By collecting the share videos owned by all participants, the secret video can be recovered. Good experimental results show the feasibility of the proposed methods.

# ACKNOWLEDGEMENTS

I am in hearty appreciation of the continuous guidance, discussions, support, and encouragement received from my advisor, Dr. Wen-Hsiang Tsai, not only in the development of this thesis, but also in every aspect of my personal growth.

Thanks are due to Miss Shung-Yung Tsai, Mr. Jiun-Tsung Wang, and Hsing-Chia Chen for their valuable discussions, suggestions, and encouragement. Appreciation is also given to the colleagues of the Computer Vision Laboratory in the Institute of Computer Science and Engineering at National Chiao Tung University for their suggestions and help during my thesis study.

Finally, I also extend my profound thanks to my family for their lasting love, care, and encouragement. I dedicate this dissertation to my beloved parents and friend.



# CONTENTS

<b>ABSTRACT (in Chinese)</b> .....	<b>i</b>
<b>ABSTRACT (in English)</b> .....	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b> .....	<b>iii</b>
<b>CONTENTS</b> .....	<b>iv</b>
<b>LIST OF FIGURES</b> .....	<b>vi</b>
<b>LIST OF TABLES</b> .....	<b>viii</b>
Chapter 1 Introduction.....	1
1.1 Motivation.....	1
1.2 General Review of Related Works.....	2
1.3 Overview of Proposed Methods.....	2
1.3.1 Terminologies .....	3
1.3.2 Brief Descriptions of Proposed Methods.....	3
1.4 Contributions.....	5
1.5 Thesis Organization.....	5
Chapter 2 Review of Related Works and H.264/AVC Standard.....	6
2.1 Review of Techniques for Video Data Hiding .....	6
2.2 Review of Techniques for Visible Watermarking in Videos .....	7
2.3 Review of Techniques for Secret Sharing.....	7
2.4 Review of H.264/AVC Standard.....	8
2.4.1 Structure of H.264/AVC Standard.....	8
2.4.2 Process of Encoding.....	11
2.4.3 Process of Decoding .....	12
Chapter 3 Data Hiding in H.264/AVC Videos for Covert Communication .....	13
3.1 Introduction.....	13
3.1.1 Problem Definition.....	13
3.1.2 Proposed Ideas .....	14
3.2 Review of Related Techniques.....	16
3.2.1 Intra-prediction .....	16
3.2.2 Tree Structured Motion Compensation.....	16
3.3 Hiding Secret Data into H.264/AVC Videos.....	18
3.3.1 Process for Hiding Large-Volume Data into I Macroblocks Based on Intra-Prediction Mode.....	19
3.3.2 Process for Hiding Data Optimally into I Macroblocks Based on Intra-Prediction Mode.....	21
3.3.3 Process for Hiding Data Optimally into P Macroblocks Based on	

Tree Structured Motion Compensation.....	23
3.4 Extraction of Secret Data from H.264/AVC Videos .....	28
3.4.1 Process for Extraction of Data from I Macroblocks by Proposed Large-Volume Method .....	29
3.4.2 Process for Extraction of Data from I Macroblocks by Proposed Optimal Method .....	30
3.4.3 Process for Extraction of Data from P Macroblocks .....	32
3.5 Experimental Results .....	34
3.5.1 Experimental Results of Large-Volume Method .....	34
3.5.2 Experimental Results of Optimal Method .....	36
3.6 Discussions and Summary .....	39
Chapter 4 Copyright Protection of H.264/AVC Videos by Watermarking and Display Control on Specified Computers.....	40
4.1 Introduction.....	40
4.1.1 Problem Definition.....	41
4.1.2 Proposed Ideas .....	41
4.2 Proposed Scheme for Display Control on Specified Computers and Embedding Visible Watermarks in H.264/AVC Videos.....	42
4.2.1 Process for Video Display Control on Specified Computers.....	42
4.2.2 Process for Embedding Visible Watermarks.....	44
4.3 Recovery of Original H.264/AVC Videos by Removing Visible Watermarks .....	48
4.4 Experimental Results .....	49
4.5 Discussions and Summary .....	54
Chapter 5 Secret H.264- AVC Video Sharing with Steganographic Effects .....	55
5.1 Introduction.....	55
5.1.1 Problem Definition.....	55
5.1.2 Proposed Ideas .....	56
5.2 Proposed Scheme for Secret Video Sharing with Steganographic Effects	56
5.2.1 Process for Share Data Creation .....	58
5.2.2 Process for Creating Steganographic Effects.....	60
5.3 Recovery of Secret Videos.....	65
5.4 Experimental Results .....	68
5.5 Discussions and Summary .....	68
Chapter 6 Conclusions and Suggestions for Future Works .....	73
6.1 Conclusions.....	73
6.2 Suggestions for Future Works.....	74
References.....	75

# LIST OF FIGURES

Figure 2.1 Relation between the Baseline, Main and Extended profiles. ....	9
Figure 2.2 Hierarchical structure of the H.264/AVC video. ....	10
Figure 2.3 Flow diagram of H.264/AVC encoding process. ....	12
Figure 2.4 Flow diagram of H.264/AVC decoding process. ....	12
Figure 3.1 Samples $a$ to $p$ of a luma $4 \times 4$ prediction block are calculated based on the sample values of A to M in neighboring prediction blocks. ....	17
Figure 3.2 Prediction modes for luma $4 \times 4$ prediction. ....	17
Figure 3.3 Macroblock partitions. ....	18
Figure 3.4 Sub-macroblock partitions. ....	18
Figure 3.5 Illustration of the proposed hiding method. ....	19
Figure 3.6 The quantized coefficient in the high-frequency. ....	21
Figure 3.7 Flowchart of the optimal data hiding process for I macroblocks. ....	24
Figure 3.8 Flowchart of the hiding process for P macroblocks. ....	27
Figure 3.9 Illustration of the proposed extraction method. ....	28
Figure 3.10 Flowchart of the extraction process for I macroblocks of optimal method. ....	31
Figure 3.11 Flowchart of the extraction process for P macroblocks. ....	33
Figure 3.12 The secret data file. ....	34
Figure 3.13 The 1 <sup>st</sup> to 4 <sup>th</sup> frames (III) of original video (left) and stego-video (right). ....	35
Figure 3.14 The extracted data file. ....	36
Figure 3.15 The 4 <sup>th</sup> to 7 <sup>th</sup> frames (PIIP) of original video (left) and stego-video (right). .....	37
Figure 3.16 The extracted data file. ....	38
Figure 4.1 Illustration of the proposed idea. ....	43
Figure 4.2 The checking process of video display control on specified computers. ....	44
Figure 4.3 Illustration of drift problem. ....	45
Figure 4.4 Types of macroblock to slice group maps (type 6 is “Explicit” which is user-defined). ....	46
Figure 4.5 Flowchart of the embedding process for I and P frames. ....	47
Figure 4.6 Flowchart of the recovery process for I and P frames. ....	50
Figure 4.7 A watermark binary image with size $16 \times 16$ . ....	51
Figure 4.8 Six frames of the original video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame). ....	51
Figure 4.9 Six frames of the watermarked video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame). ....	52



Figure 4.10 Six frames of the recovered video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame). ..... 53

Figure 5.1 An illustration of the proposed idea..... 57

Figure 5.2 Illustration of the secret sharing method. .... 60

Figure 5.3 Illustration of the prediction error problem. .... 61

Figure 5.4 Flowchart of the process of creating steganographic effects..... 64

Figure 5.5 Flowchart of the recovering process..... 67

Figure 5.6 The four frames of the original video (left) and randomized video (right).. 69

Figure 5.7 The four frames of the first cover video (left) and the share video (right)... 70

Figure 5.8 The four frames of the second cover video (left) and the share video (right).  
..... 71

Figure 5.9 The four frames of the recovered secret video. .... 72

Figure 5.10 Erroneous recovery result when the secret key is wrong. .... 72



# LIST OF TABLES

Table 2.1 Relationships between slice types and macroblock types.....	10
Table 3.1 Relations between hidden data and partition sizes.....	25
Table 3.2 Configuration parameters.....	34
Table 3.3 Configuration parameters.....	36
Table 3.4 NBH, PSNRI and BRI values for several video sequences ( $\gamma_b, \gamma_p = 250$ ).....	39
Table 4.1 Configuration parameters.....	51
Table 5.1 Relation between the data-to-be-hidden and the prediction mode.....	62



# Chapter 1

## Introduction

### 1.1 Motivation

With the advance of the Internet and multimedia technologies, more and more people transmit videos through the Internet. H.264/AVC is the latest video compression standard, and contains a lot of new features that allow it to compress videos more effectively than older standards. H.264/AVC also provides more flexibility for applications on a wide variety of networks, and so it is suitable for use as a kind of carrier for information hiding investigated in this study.

Data hiding techniques can be used to hide *secret data* in a video, resulting in a so-called *stego-video*. In this way, stego-videos instead of secret data may be transmitted through the network. Except the owner, other users usually do not know the existence of the hidden information and so will not try to get the information because the secret data hidden in a video are invisible. It is desired in this study to develop a data hiding method via H/264/AVC videos for covert communication.

Copyright protection of videos becomes more and more important nowadays because videos communicated on the Internet might be copied or misused. For video copyright protection, one approach is to utilize digital visible watermarking techniques. Using this approach to certify the copyright of a video has a main advantage, i.e., the watermark conveys a straightforward claim of the ownership of the video. But the video content might be partially occluded by an embedded visible

watermark. In order to solve this problem, it is desired to develop a removable visible watermark technique in this study, by which videos can be protected by the use of visible watermarks. And when displaying a video on a specified computer, it is also hoped that we can remove the watermark before watching the video.

Secret sharing is a technique for use to transform secret data into multiple shares, with each share kept by a participant. Each share may be created to have a meaningless content. By collecting a sufficient number of shares, we can recover the secret data. Because each meaningless share might be suspected easily, it is a good idea to hide each share into another meaningful media file. This effect can be accomplished by applying the technique of steganography to the meaningless shares.

Secret sharing techniques have been applied to various kinds of files, such as image and video. Though studies on the secret image sharing technique are getting intensive now, there are very few studies on secret video sharing yet. So, it is desired in this study to develop a secret video sharing method which creates steganographic effects.

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

Although all the above-mentioned goals of this study are related to the technique of hiding information within videos, different methods should be adopted for different applications. A detailed review of video data hiding, visible watermarking, and secret sharing techniques which have been developed in recent years will be introduced in Chapter 2. In addition, because the proposed data hiding and watermarking techniques are applied to H.264/AVC videos, we will also make a review of the H.264/AVC standard in Chapter 2.

## **1.3 Overview of Proposed Methods**

### 1.3.1 Terminologies

The definitions of some related terminologies used in this study are described as follows.

1. *Secret*: a secret is a piece of information that is important and should be preserved properly and not revealed to unauthorized people.
2. *Stego-video*: a stego-video is one in which some digital information is embedded.
3. *Watermarked video*: a watermarked video is one in which a visible watermark has been embedded.
4. *Recovered video*: a recovered video is one obtained by removing the embedded visible watermark from a watermarked video.
5. *Cover video*: a cover video is input one use for secret sharing and steganography.
6. *Share video*: a share video is one of the secret sharing results of a secret video.

### 1.3.2 Brief Descriptions of Proposed Methods

#### A. An Data Hiding Method for Covert Communication

A method using a data hiding technique for covert communication via H.264/AVC videos is proposed in this study. An H.264/AVC video consists of a series of I and P image frames. We propose proper hiding techniques for macroblocks in I and P frames by utilizing their properties, respectively. We hide data into the intra-prediction mode of the I macroblock, and into the tree structured motion compensation of the P macroblock. Briefly, we modify the prediction mode and the

variable partition size of the tree structured motion compensation to hide data and maintain imperceptibility of the hidden data. In addition, we also use the Lagrange optimization technique to optimize the changes yielded by the data hiding process.

## **B. A Visible Watermarking Method for Copyright Protection**

A method for copyright protection of videos using a removable visible watermarking technique and a scheme for video display control on specified computers is also proposed in this study. By performing encoding process on a given video, the quantized transform coefficients of all the  $16 \times 16$  luminance macroblocks of each frame of the video are obtained. We embed one pixel of a visible watermark into the direct current (*DC*) coefficient of the luminance (*luma*) macroblock.

On the other hand, a software player is designed to obtain certain hardware identification information (such as the number of the CPU and the volume serial number) from the user's computer, when a video is displayed. If the computer information is correct, the player will remove the visible watermark embedded in the watermarked-video; otherwise, a visible watermark will appear promptly to state copyright and prohibit the viewer from enjoying the video content.

## **C. A Video Sharing Method with Steganographic Effects**

A method using the secret sharing and steganography techniques for sharing secret videos is proposed in this study. The technique of sharing secret is based on the exclusive-OR operation, by which we encode the prediction mode of secret videos and the upper half of the prediction mode of a covert video into several pieces of share data. Then we hide the meaningful share data into the lower half of the prediction mode of the covert video. The share videos are distributed to the

participants for custody by them. After collecting all the share videos from the participants, we can extract the hidden data and recover the secret video.

## 1.4 Contributions

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

1. An data hiding method based on some properties of H.264/AVC is proposed for covert communication.
2. A removable visible watermarking method with a scheme for video display control on specified computers is proposed for protecting the copyright of an H.264/AVC video.
3. A method of video sharing with steganographic effects is proposed for protecting secret videos systematically and securely.

## 1.5 Thesis Organization

In the remainder of this thesis, a review of related works about techniques of video data hiding, visible watermarking, and information sharing, as well as the H.264/AVC standard is given in Chapter 2. In Chapter 3, the proposed method for video data hiding for covert communication is described. In Chapter 4, the proposed removable visual watermarking method is described. In Chapter 5, the proposed method for video sharing is described. Finally, conclusions and some suggestions for future researches are made in Chapter 6.

# Chapter 2

## Review of Related Works and H.264/AVC Standard

### 2.1 Review of Techniques for Video Data Hiding

Techniques of video data hiding are developed for hiding secret data into a video. By this way, secret data can be transmitted covertly. A lot of approaches related to hiding data into a video have been proposed [1-3]. Yang and Bourbakis [1] proposed a scheme for embedding data in the DCT coefficients by means of vector quantization. Hu et al. [2] proposed a method for hiding data in H.264/AVC videos based on intra-prediction modes. The basic idea is to modify 4×4 intra-prediction modes based on the mapping between 4×4 intra-modes and hidden bits. Their method uses only the intra-coded macroblock to hide data. Kapotas et al. [3] proposed a method for embedding data into encoded video sequences, in which the hiding technique is used to modulate the partition size to hide the secret data. This method can only be used for embedding information in inter-coded macroblock.



## 2.2 Review of Techniques for Visible Watermarking in Videos

Visible watermarking is a technique for copyright protection [4]. The owner of a video can embed a visible watermark representing copyright information into the video, and this embedded watermark can be removed when proving his ownership. Bhattacharya et al [5] surveyed different video watermarking techniques and used comparison analysis with reference to H.264/AVC. Mohanty et al. [6] proposed a DCT-domain visible watermarking technique for images. In their method, embedding visible watermarks in the DCT coefficients is based on a mathematical model developed by exploiting the texture sensitivity of the human visual system (HVS). Chien and Tsai [7] proposed an active watermarking method for the MPEG-4 videos with a scheme for video displays with limited counts. The basic idea is to use an active agent to check available play counts. If the play count of the video is not zero, the active agent will remove the visible watermark embedded in the video; otherwise, the visible watermark will appear promptly to state the copyright.

## 2.3 Review of Techniques for Secret Sharing

Secret sharing is a technique for use to transform secret data into multiple shares, with each shares kept by a participant. When a pre-defined group of shares is collected, the secret data can be recovered. Shamir [8] proposed the concept of secret sharing in his  $(k, n)$ -threshold method, in which  $n$  indicates the number of participants and  $k$  means a threshold as the minimum number of shares in the pre-defined group. Lin and Tsai [9] proposed an efficient  $(n, n)$ -threshold secret sharing method using

exclusive-OR operations. This method simply applies the exclusive-OR operation to a secret image and uses  $n-1$  images to generate the  $n$ th image. The  $n-1$  images and the  $n$ th image are taken as shares and are distributed to  $n$  participants separately. By exclusive-OR operations to  $n$  images held by the  $n$  participants, the secret image can be recovered quickly. Zou and Sun [10] proposed an approach which combines secret sharing and information hiding for covert communication.

## 2.4 Review of H.264/AVC Standard

In this study, all the proposed information hiding, watermarking and video sharing techniques employ H.264/AVC videos as carrier media for hiding information. Richardson described in detail the H.264/AVC standard in his book [11]. We will give a brief review of the H.264/AVC standard in this section. In Section 2.4.1, the structure of the H.264/AVC standard will be described. In Section 2.4.2 and Section 2.4.3, the encoding and decoding processes in the H.264/AVC standard will be described.

### 2.4.1 Structure of H.264/AVC Standard

The H.264/AVC standard defines a set of three *Profiles: Baseline, Main* and *Extended*, which support different functions and suit different environment. Figure 2.1 shows the relationship between the three profiles and the coding tools supported by the standard. The H.264/AVC video has a hierarchical structure as illustrated in Figure 2.2. A video sequence is composed of a series of pictures (frames). The picture is coded as one or more slices. In general, there are three main slice types for use in H.264/AVC standard, including intra-slice (I), predictive slice (P), and bi-predictive slice (B). The slice consists of a number of macroblocks. There are four different types of macroblocks, including I macroblock, P macroblock, B macroblock, skipped

macroblock. I macroblocks are predicted from previously coded data within the same slice. P macroblocks are predicted from one reference picture. B macroblocks are predicted from two reference pictures. Skipped macroblocks of the P slice are encoded with a motion vector and no transform coefficients. And skipped macroblocks of the B slice are encoded without motion vectors and no transform coefficients. Each slice type has its own macroblock types. The relationships between the slice types and the macroblock types are listed in Table 2.1.

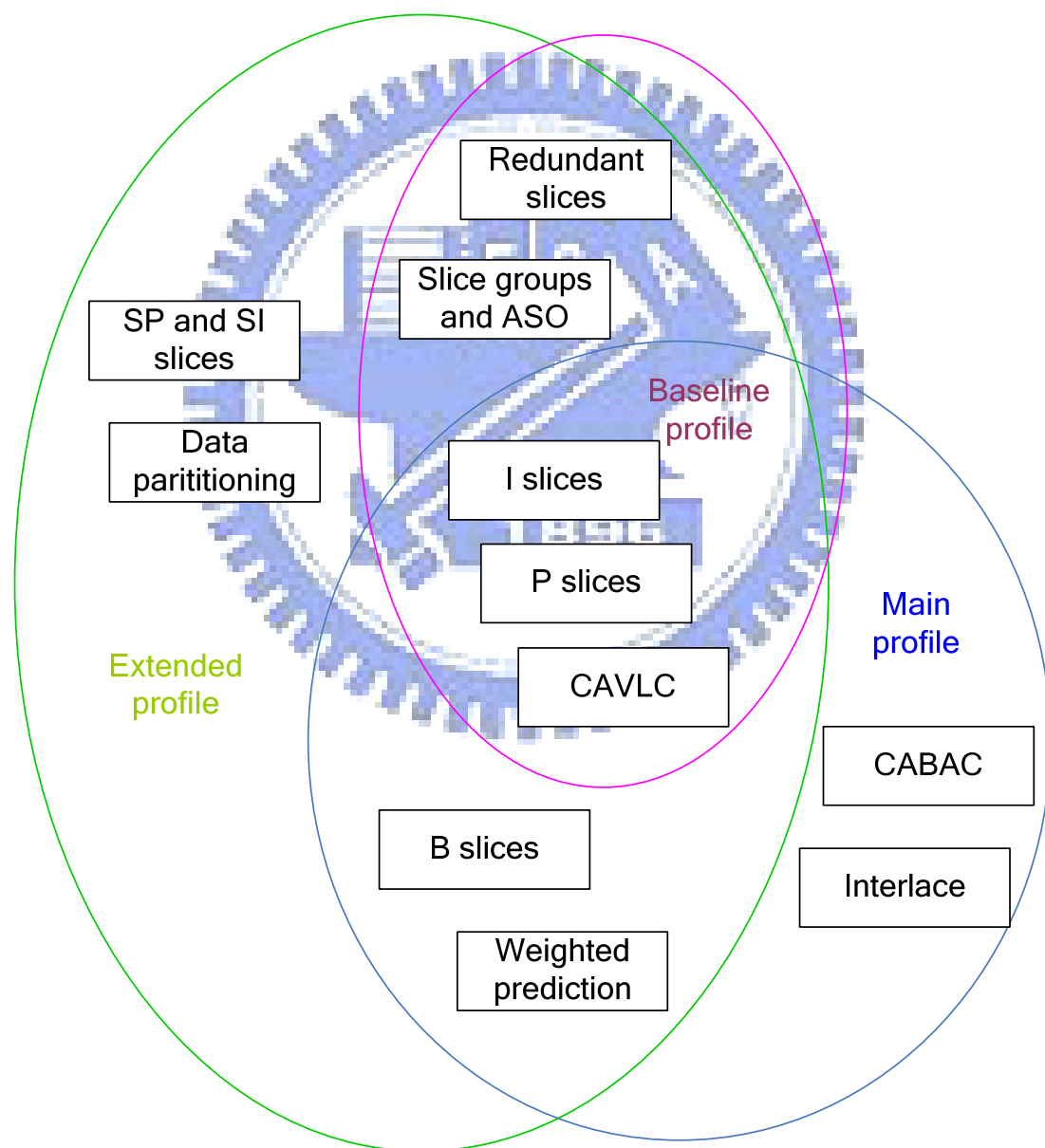


Figure 2.1 Relation between the Baseline, Main and Extended profiles.

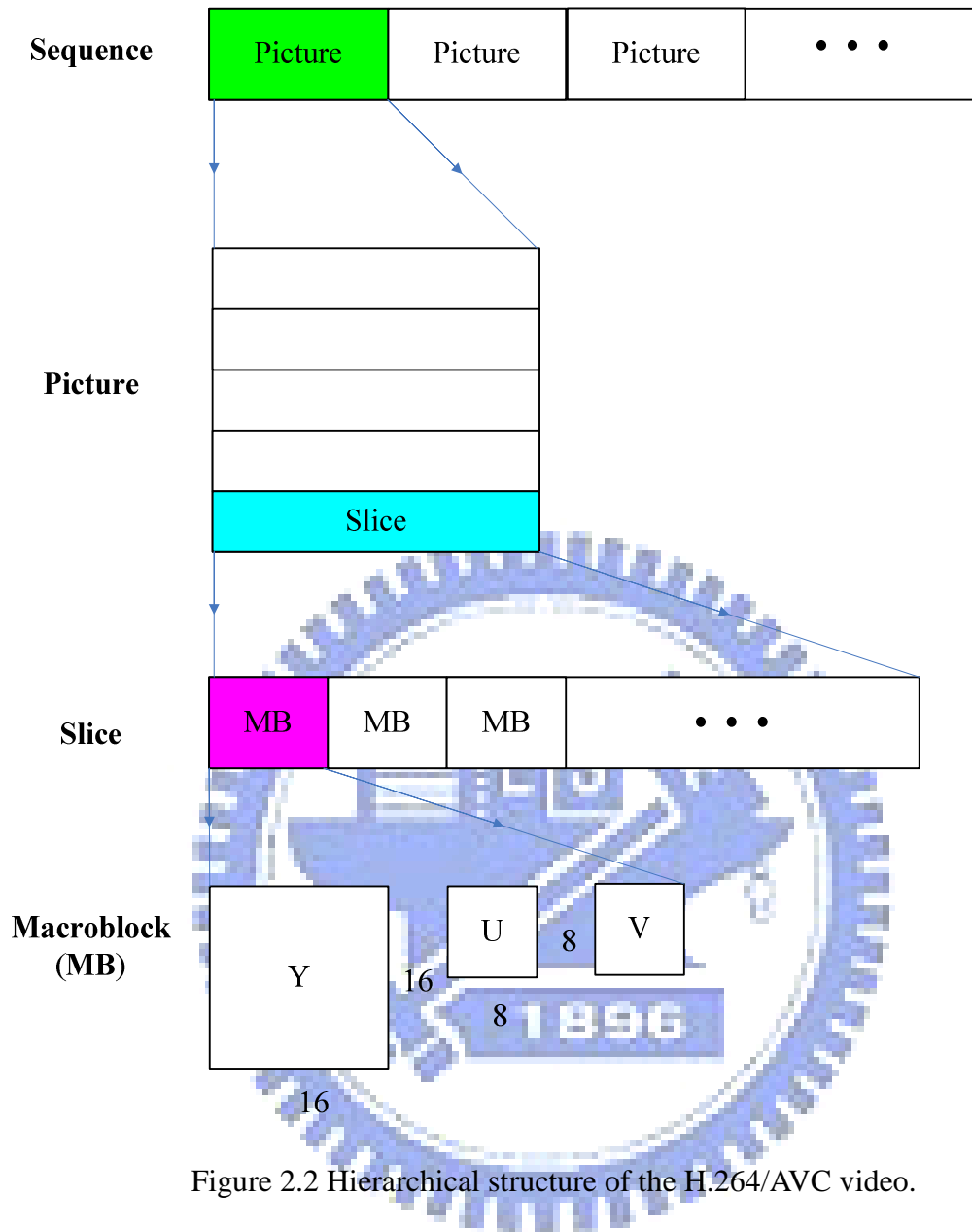


Figure 2.2 Hierarchical structure of the H.264/AVC video.

Table 2.1 Relationships between slice types and macroblock types.

	<b>I macroblock</b>	<b>P macroblock</b>	<b>B macroblock</b>	<b>Skipped macroblock</b>
<b>I slice</b>	●			
<b>P slice</b>	●	●		●
<b>B slice</b>	●		●	●

## 2.4.2 Process of Encoding

A flow diagram of the encoding process is shown in Figure 2.2. In the encoding process, there are two data flow paths, forward (left to right) and reconstruction (right to left). In forward paths, each  $16 \times 16$  macroblock is encoded in intra-mode or inter-mode, and a *prediction* (marked as  $P$  in Figure 2.2) is calculated by reconstructed data. In the intra-mode, the encoder calculates the best intra-prediction mode by reconstructed data in the current slice, and then computes the intra-prediction. In the inter-mode, the encoder calculates the best motion vector based on the reconstructed data in one or two reference picture(s), and then computes the motion-compensated prediction. The prediction is subtracted from the current block to produce a residual block (marked as  $D_n$  in Figure 2.2). A DCT-based transform is performed on each residual block. After that, each  $4 \times 4$  block of the transform coefficients is quantized. Each resulting block (marked as  $X$  in Figure 2.2) is scanned in a zig-zag order and entropy encoded. An entropy technique is used to compress the quantized coefficient data and other information required to decode each block within the macroblock and form the compressed bitstream. Finally, the compressed bitstream is passed to the network abstraction layer (*NAL*) for transmission or storage. In reconstruction paths, the encoder decodes (reconstructs) each block in a macroblock which is regarded as a reference for further prediction. The quantized coefficients are scaled and inverse-transformed to produce a difference block (marked as  $D'_n$  in Figure 2.2), and then the prediction is added to the difference block to produce a reconstructed block (marked as  $uF'_n$  in Figure 2.2). Finally, the filter is used to reduce the effects of blocking distortion and the reconstructed reference picture is created from a series of blocks.

## 2.4.3 Process of Decoding

A flow diagram of the decoding process is shown in Figure 2.3. The decoder receives a compressed bitstream from the NAL and entropy decodes the data to get the quantized coefficients. Through scale and inverse-transform, the decoder obtains a difference block. By the header information from the bitstream, the decoder creates a prediction, identical to the original prediction formed in the encoder. The prediction is added to the difference block to produce the reconstructed block which is then filtered to create a decoded block.

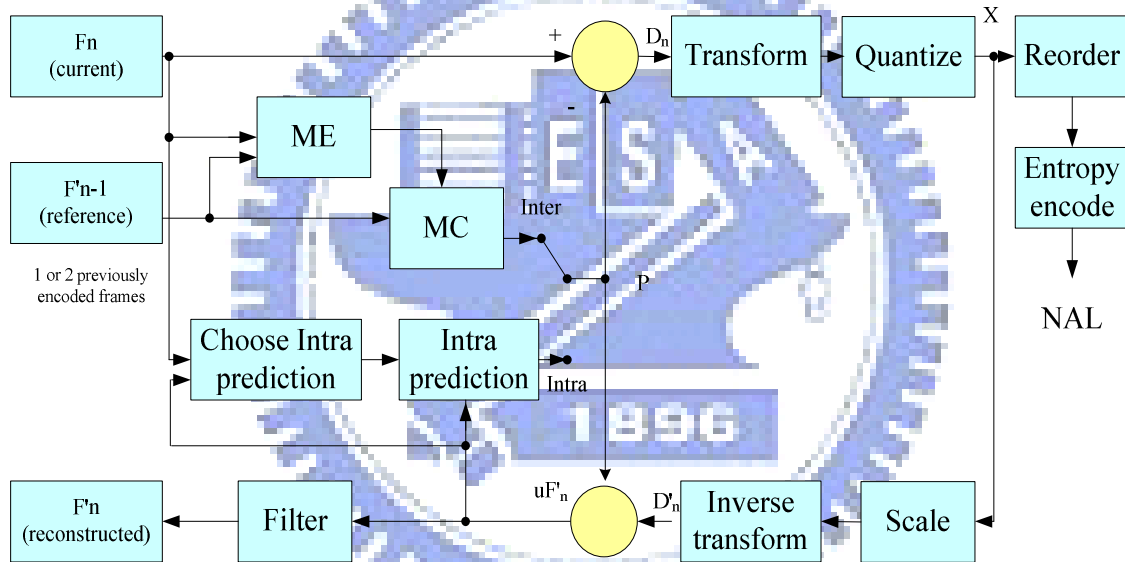


Figure 2.3 Flow diagram of H.264/AVC encoding process.

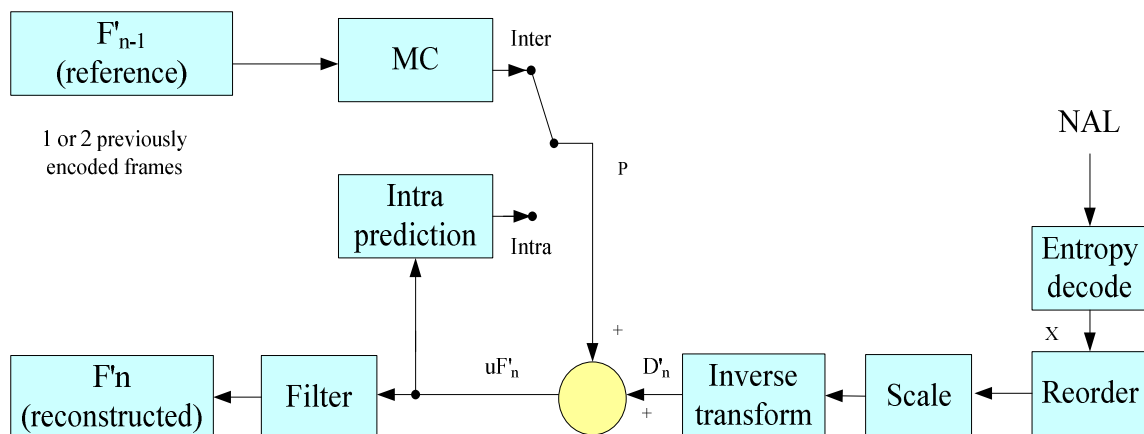
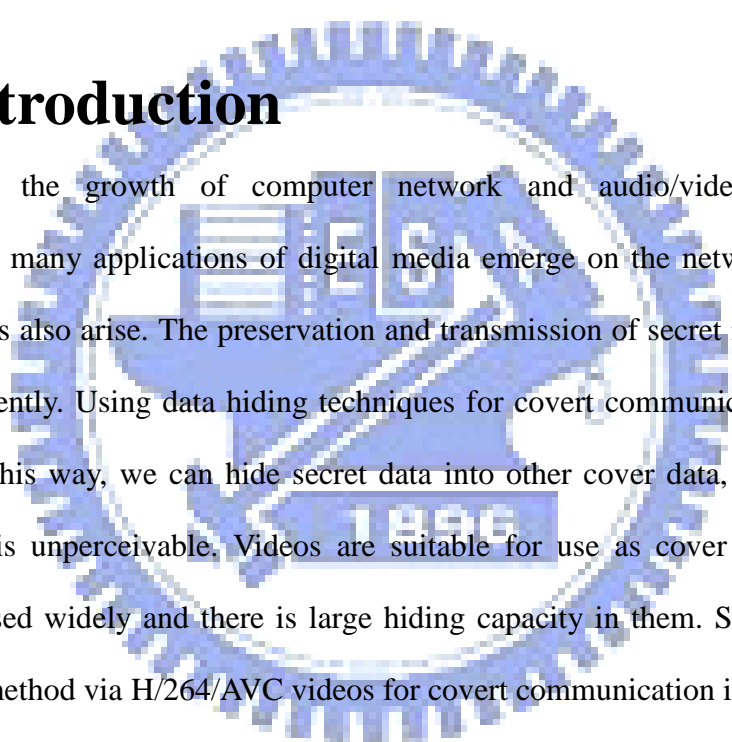


Figure 2.4 Flow diagram of H.264/AVC decoding process.

# Chapter 3

## Data Hiding in H.264/AVC Videos for Covert Communication

### 3.1 Introduction



Due to the growth of computer network and audio/video compression technologies, many applications of digital media emerge on the network. But many new problems also arise. The preservation and transmission of secret information is a hot topic recently. Using data hiding techniques for covert communication is a good solution. In this way, we can hide secret data into other cover data, and the hidden information is unperceivable. Videos are suitable for use as cover media because videos are used widely and there is large hiding capacity in them. So we propose a data hiding method via H/264/AVC videos for covert communication in this study.

In Section 3.1.1, some relevant definitions are given, and in Section 3.1.2 the basic ideas of the proposed method are presented. In Section 3.2, the proposed data hiding method is described, and the corresponding data extraction method is stated in Section 3.3. In Section 3.4, several experimental results are shown to prove the feasibility of the proposed method. Finally, some discussions and a summary of the proposed method are made in the last section of this chapter.

#### 3.1.1 Problem Definition

Traditionally, when applying video data hiding techniques for covert communication, the data hiding capacity and the imperceptibility of the hidden data are two of the major concerns. Therefore, the problem is how to hide data with large-volume capacity and imperceptibility.

In addition, with the popularity of web applications, people give more and more attention to low bit rate videos. Therefore, an additional problem is how to hide data into videos and get optimal results which take data hiding capacity, imperceptibility, and low bit rating into consideration.

### 3.1.2 Proposed Ideas

There are two macroblock types for use in the *baseline profile* of the H.264/AVC standard, which are I macroblock and P macroblock. We propose data hiding techniques for the two macroblock types, respectively, in this study.

Two methods are proposed for hiding data into I macroblocks based on the *intra-prediction mode*, which is a new coding method proposed in the H.264/AVC standard. In the first method, we transform the data to be hidden into *noventary* data and encode them by the use of the prediction modes.

In the second method, an encoder selects the *best* prediction mode for each block by a *Lagrangian cost function* [12] to minimize simultaneously the rate and distortion in the H.264/AVC standard, which is formulated as follows:

$$J = \arg \min_{M_k \in \tau} (D(S_k, M_k) + \lambda R(S_k, M_k)) \quad (3.1)$$

where

- (1)  $\tau$  denotes the set of all the nine prediction modes, i.e.,  $\tau = \{M_1, M_2, \dots, M_9\}$ ;
- (2)  $\lambda$  represents the Lagrange multiplier;
- (3)  $S_k$  denotes the block being processed;



- (4)  $D$  is a *distortion function* whose value is computed as the *sum of the squared differences (SSD)* between the reconstructed block  $S_k'$  and the original one  $S_k$ ;
- (5)  $R$  denotes the used bits for encoding the block  $S_k$  using the prediction mode  $M_k$ .

In our approach, the block  $S_k$  is fixed to be  $4 \times 4$  which yields higher data embedding rates. Furthermore, we add the hiding capacity as a new parameter to the Lagrangian cost function described by (3.1), resulting in:

$$J = \arg \min_{M_i \in \tau} (D(S, M_i) + \lambda R(S, M_i) - \gamma_1 \cdot N_i) \quad (3.2)$$

,where the new parameter  $\gamma_1$  is a multiplier for the hiding capacity  $N_i$  (in unit of bit) in the  $4 \times 4$  block. By this function, we can get the best result as a tradeoff among the data hiding capacity, the bit rate, and the resulting distortion.

The idea of hiding data in the P macroblocks proposed in this study is to modify the variable partition size of the tree structured motion compensation, which is a different feature of the H.264/AVC standard from earlier standards. Tree structured motion compensation is a method of partitioning macroblocks into motion compensated sub-blocks of varying sizes. The encoder selects the partition size for each macroblock by a Lagrangian cost function described as follows:

$$J = \arg \min_{P_k \in \omega} (D(S, M_k) + \lambda R(S, M_k)) \quad (3.3)$$

where  $\omega$  denotes the set of all alternative partition sizes, and  $P_k$  denotes the current partition size. Similarly, we add hiding capacity as a new parameter to the Lagrangian cost function, resulting in:

$$J = \arg \min_{P_i \in \omega} (D(S, M_i) + \lambda R(S, M_i) - \gamma_2 \cdot N_i) \quad (3.4)$$

,where the new parameter  $\gamma_2$  is the multiplier for hiding capacity (in unit of bit) in P macroblocks. By this formula, we can get the best result as a tradeoff among the data

hiding capacity, the bit rate, and the resulting distortion.

## 3.2 Review of Related Techniques

### 3.2.1 Intra-prediction

For each I macroblock of an H.264/AVC video, a  $4 \times 4$  *prediction block* as shown in Figure 3.1 includes 16 *samples*  $a, b, \dots, p$  whose values are computed from some samples of previously encoded and reconstructed blocks ( $A, B, C, D$  in the top row from the upper neighboring block;  $E, F, G, H$  from the upper right block;  $I, J, K, L$  in the leftmost column from the left neighboring block; and  $M$  from the upper left block, as shown in Figure 3.1). And the resulting prediction block is subtracted from the current block prior to encoding. On the other hand, to compute the values of the prediction block samples, it is noted first that there are nine possible prediction modes for a luminance  $4 \times 4$  block (abbreviated as a *luma block* in the sequel). The nine prediction modes are illustrated in Figure 3.2(a). Except prediction mode 2 with its samples all of the same value which is computed as the mean of  $A$  through  $D$  and  $I$  through  $L$ , the values of the samples of the remaining eight modes are computed from those values of  $A$  through  $M$  according to eight directions as illustrated in Figure 3.2(b). The H.264/AVC standard allows the selection of an encoder which adopts, among the nine modes, the best one with the lowest *rate-distortion cost* computed by the Lagrangian cost function described by Eq. (3.1).

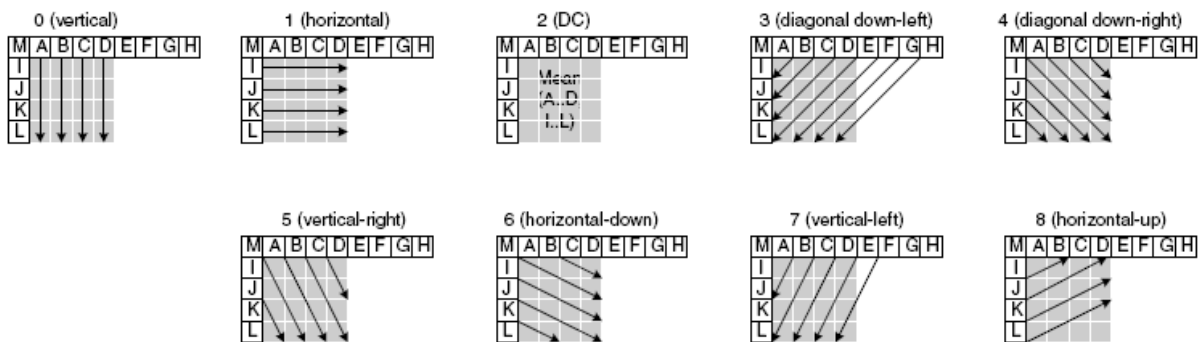
### 3.2.2 Tree Structured Motion Compensation

A P macroblock may be split and motion compensated by four ways as (1) one  $16 \times 16$  macroblock partition; (2) two  $16 \times 8$  partitions; (3) two  $8 \times 16$  partitions; or (4) four  $8 \times 8$  partitions, as shown in Figure 3.3. If the  $8 \times 8$  partitions are selected, each of

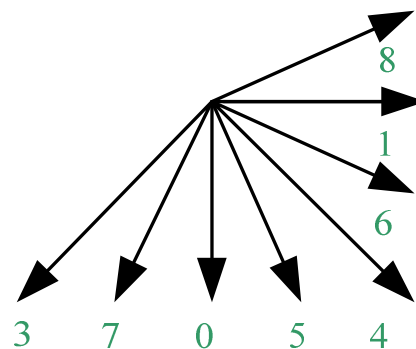
the four 8×8 sub-macroblocks may be split further by four ways as (1) one 8×8 sub-macroblock partition; (2) two 8×4 partitions; (3) two 4×8 partitions; or (4) four 4×4 partitions, as illustrated in Figure 3.4. An encoder selects the best partition size which has the lowest rate-distortion cost computed by the Lagrangian cost function (3.3).

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 3.1 Samples *a* to *p* of a luma 4×4 prediction block are calculated based on the sample values of A to M in neighboring prediction blocks.



(a) Nine prediction modes for 4×4 prediction blocks.



(b) Directions for computing samples of eight prediction modes.

Figure 3.2 Prediction modes for luma 4×4 prediction.

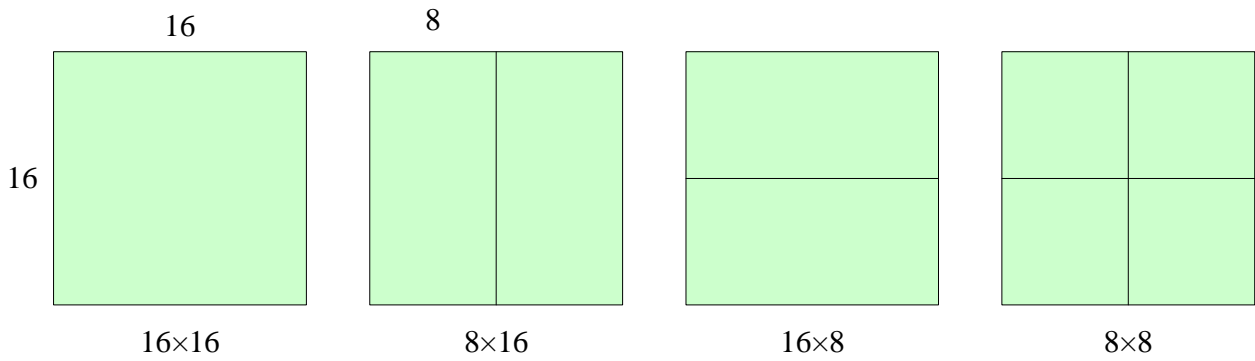


Figure 3.3 Macroblock partitions.

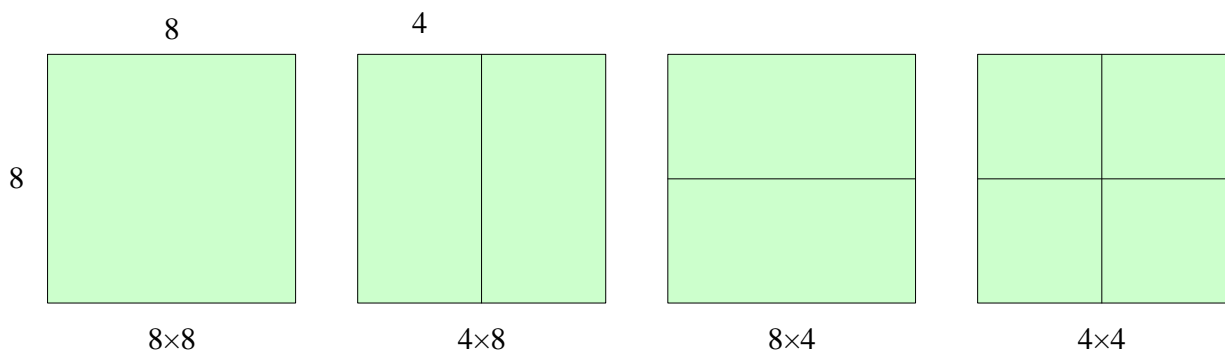


Figure 3.4 Sub-macroblock partitions.

## 3.3 Hiding Secret Data into H.264/AVC Videos

In this section, the proposed methods of hiding data into different types of macroblocks of H.264/AVC videos will be described. An illustration of the hiding method is shown in Figure 3.5. In Section 3.3.1, the proposed method for hiding large-volume data in I macroblocks based on the use of the nine intra-prediction modes is described. In Section 3.3.2, the proposed method for hiding data in I macroblocks based on optimal choice of an intra-prediction mode is described. Finally, the proposed method for hiding data in P macroblocks optimally based on tree

structured motion compensation is described in Section 3.3.3.

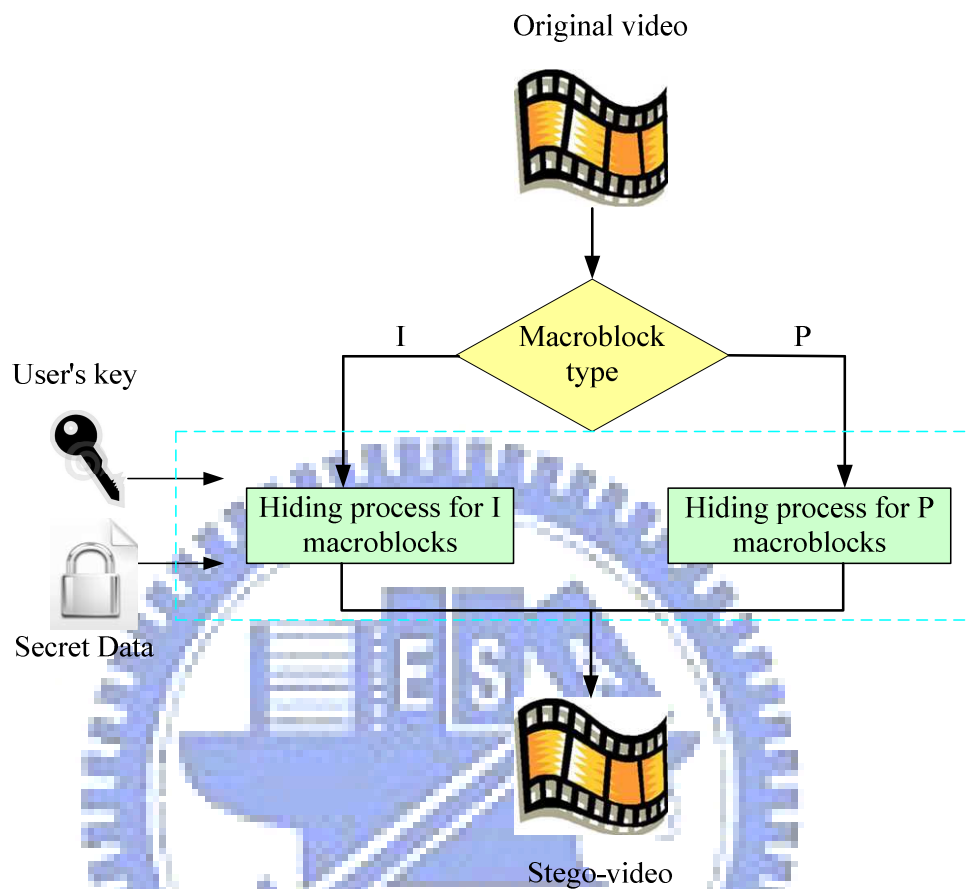


Figure 3.5 Illustration of the proposed hiding method.

### 3.3.1 Process for Hiding Large-Volume Data into I Macroblocks Based on Intra-Prediction Mode

In this section, we describe the proposed method for hiding secret data based on the direct use of the nine prediction modes. To take full advantage of the nine prediction modes, we transform the binary data to be hidden into novenary ones, and then encode the result by the prediction modes. In addition, we also combine the user's secret key and the secret data by exclusive-OR operations for the purpose of ensuring that the hidden data can be extracted only by a user who has the correct key.

A detailed algorithm of the process is described in the following.

**Algorithm 3.1:** large-volume data hiding process using I macroblocks.

**Input:** a user's key  $R$ , a secret data file  $D$ , and the  $4 \times 4$  luma prediction mode  $M$ .

**Output:** a stego-macroblock  $I'$ .

**Steps:**

1. For each character  $D_i$  of the secret data  $D$ , perform the following steps.
  - 1.1 Compute the remainder  $R'$  of dividing  $R$  by 256.
  - 1.2 Transform each character  $D_i$  of the secret data  $D$  in the following way to form encrypted data  $E$ :
 
$$E_i = D_i \oplus R' . \quad (3.5)$$
2. Transform  $E$  into a six novenary number  $N$  by converting every nineteen bits of  $E$  into a novenary digit. So each  $4 \times 4$  luma prediction mode in this method macroblock can be used to hide  $19/6$  bits of data.
3. Encode each digit  $N_i$  of  $N$  with magnitude  $i$  by the corresponding prediction mode  $M_i$ .

For example, if the user key is  $R = 3735$ , then  $R' = 3735/256 = 141_{10} = 10001101_2$ . Now, suppose that a secret message character  $D_1 = 'a'$  is to be embedded, whose corresponding binary form is  $01100001_2$ . Then, the encrypted form of  $D_1$  is  $E_1 = 01100001 \oplus 10001101 = 11101100_2$ . Similarly, if  $D_2 = 'b,' D_3 = 'c,'$  with binary forms being  $01100010_2$  and  $01100011_2$ , respectively, then  $E_2 = 01100010 \oplus 10001101 = 11101111_2$  and  $E_3 = 01100011 \oplus 10001101 = 11101110_2$ . Together, we get  $E = E_1 E_2 E_3 = \underline{111011001110111111101110}_2$  whose first 19 bits as underlined, when converted into novenary, becomes the novenary number  $818563_9$ , and so may be encoded by the prediction mode  $M_1 = 8, M_2 = 1, M_3 = 8, M_4 = 5, M_5 = 6, M_6 = 3$ .

### 3.3.2 Process for Hiding Data Optimally into I

#### Macroblocks Based on Intra-Prediction Mode

In this section, we describe how we hide secret data optimally in a sense mentioned previously, based on the use of the nine prediction modes. Each 4×4 luma prediction mode in the I macroblock can be used to hide zero to four bits of data by this method and the method does not influence the degree of the imperceptibility. In addition, we recode the number of bits so hidden in the highest-frequency quantized coefficients of the 4×4 block, as shown in Figure 3.6. We also use the user’s secret key to encrypt the secret data to enhance the security. A detailed algorithm of the process is described in the following.

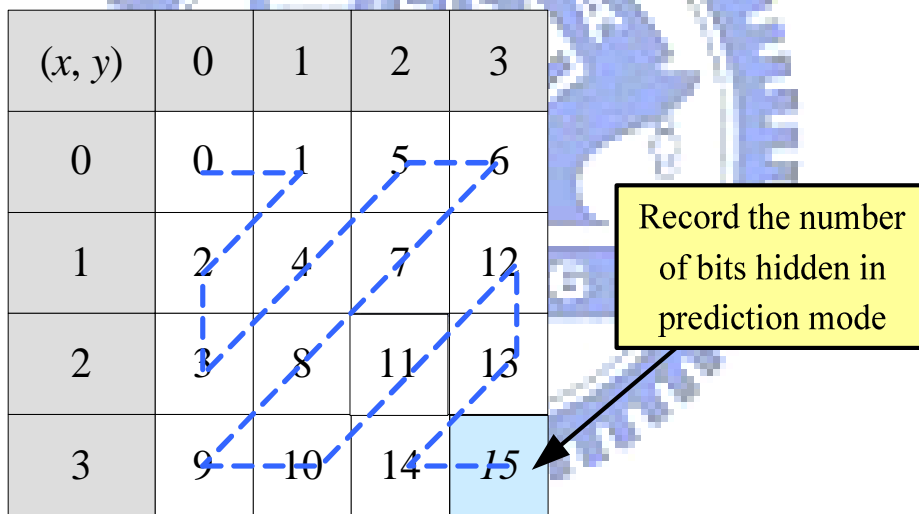


Figure 3.6 The quantized coefficient in the high-frequency.

**Algorithm 3.2:** optimal data hiding process for I macroblocks.

**Input:** an I macroblock in the spatial domain,  $I$ , a user’s key  $R$ , and a secret data file  $D$ .

**Output:** a stego-macroblock  $I'$ .

**Steps:**

1. For each character  $D_i$  of the secret data  $D$ , perform the following steps.
  - 1.1 Compute the remainder  $R'$  of dividing  $R$  by 256.
  - 1.2 Transform each character  $D_i$  of the secret data  $D$  according to Eq. (3.5) to form encrypted data  $E$ .
2. For each luma  $4 \times 4$  block  $B$  of  $I$ , perform the following operations.
  - 2.1 For each luma  $4 \times 4$  prediction mode  $M_i$ , perform intra-prediction, DCT-based transform, and quantization in the video coding process, and then match four bits of  $E$ ,  $E_3E_2E_1E_0$  with the 4-bit numeral value  $I_0I_1I_2I_3$  the index  $i$  of  $M_i$  to obtain the number of bits  $N_i$  which can be hidden in  $M_i$  in the following way:
    - if  $E_3 = I_3$ , then set  $N_i = 1$ ;
    - if  $E_3 = I_3$  and  $E_2 = I_2$ , then set  $N_i = 2$ ;
    - if  $E_3 = I_3$ ,  $E_2 = I_2$  and  $E_1 = I_1$ , then set  $N_i = 3$ ;
    - if all  $E_j$  are equal to  $I_j$ , then set  $N_i = 4$ ;
    - otherwise, set  $N_i = 0$ .
  - 2.2 Replace the highest-frequency quantized coefficients  $C$  as shown in Figure 3.4 by a new value according to the following mapping rules:
    - if  $N_i = 0$ , then set  $C = 0$ ;
    - if  $N_i = 1$ , then set  $C = 1$ ;
    - if  $N_i = 2$ , then set  $C = -1$ ;
    - if  $N_i = 3$ , then set  $C = 2$ ;
    - if  $N_i = 4$ , then set  $C = -2$ .
3. Select the best prediction mode according to Eq. (3.2), and so decide the number of bits which can be hidden in this block. Take away from  $E$  these bits.
4. Repeat the above steps to encode more bits in the remaining portion of  $E$  until no more is left.



For example, if the user key is  $R = 3735$ , then  $R' = 3735/256 = 141_{10} = 10001101_2$ . Now, suppose that a secret message character  $D_1 = 'a'$  is to be embedded, whose corresponding binary form is  $01100001_2$ . Then, the encrypted form of  $D_1$  is  $E_1 = 01100001 \oplus 10001101 = 11101100_2$ . Similarly, if  $D_2 = 'b'$  with binary form being  $01100010_2$ , then  $E_2 = 01100010 \oplus 10001101 = 11101111_2$ . Together, we get  $E = E_1E_2 = \underline{1110}110011101111_2$  whose first 4 bits are then matched to the binary equivalent of the index  $i$  of each prediction mode  $M_i$ . Suppose the best mode selected using the Lagrangian cost function is  $M_3$  whose corresponding binary index is  $3 = 0011_2$  (bits from right to left correspond to bits of  $E$  from left to right), we get two matching bits which can be hidden in  $M_3$ . And so we set  $C = -1$  and hide it in the highest-frequency quantized coefficient.

A flowchart of the optimal data hiding process for I macroblocks is shown in Figure 3.7.

### 3.3.3 Process for Hiding Data Optimally into P Macroblocks Based on Tree Structured Motion Compensation

In this section, we hide secret data based on variable partition sizes of  $16 \times 16$  macroblocks. Each  $16 \times 16$  P macroblock can be used to hide one or four bit(s) of data by modifying the partition size. In order to allow better choices of sizes to reduce rate-distortion, we encode hidden data by the partition size with multiple choices for 0 or 1 according to Table 3.1, in which two groups of sizes are used to encode 0 and 1, respectively. In addition, we use the user's secret key to encrypt secret data. A detailed algorithm of the process is described in the following.

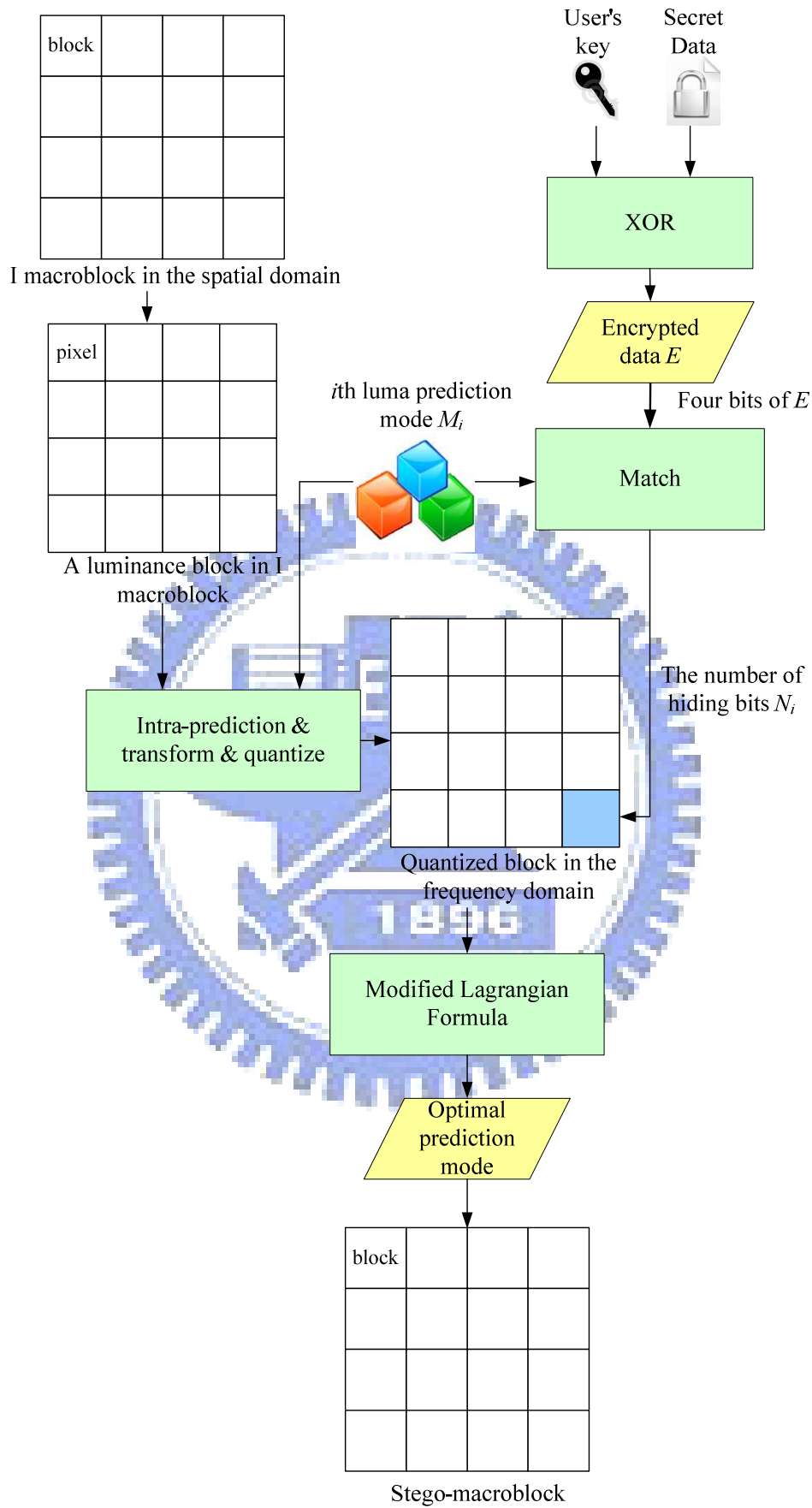


Figure 3.7 Flowchart of the optimal data hiding process for I macroblocks.

Table 3.1 Relations between hidden data and partition sizes.

<i>Partition size</i>	Hidden data
<b><i>16×16</i></b>	1
<b><i>8×16</i></b>	0
<b><i>16×8</i></b>	0
<b><i>8×8</i></b>	1
<b><i>4×8</i></b>	0
<b><i>8×4</i></b>	0
<b><i>4×4</i></b>	1

**Algorithm 3.3:** optimal data hiding process for P macroblocks.

**Input:** a P macroblock in the spatial domain  $P$ , a user's key  $R$ , a secret data file  $D$ , and the macroblock partition size  $K$ .

**Output:** a stego-macroblock  $P'$ .

**Steps:**

1. For each character  $D_i$  of the secret data  $D$ , perform the following steps.
  - 1.1 Compute the remainder  $R'$  of dividing  $R$  by 256.
  - 1.2 Transform each character  $D_i$  of the secret data  $D$  according to Eq. (3.5) to form encrypted data  $E$ .
2. According to Table 3.1, hide one bit  $e_1$  or four bits  $e_j$  of  $E$  into the macroblock partition according to the following rules for the macroblock partition size  $K$ .

2.1 When the partition size  $K$  is  $16 \times 16$ :

$$\begin{aligned} &\text{if } e_1 = 0, \text{ then perform next partition size;} \\ &\text{if } e_1 = 1, \text{ then perform Step 3.} \end{aligned} \quad (3.7)$$

2.2 When the partition size  $K$  is  $8 \times 16$  or  $16 \times 8$ :

$$\begin{aligned} &\text{if } e_1 = 0, \text{ then perform Step 3;} \\ &\text{if } e_1 = 1, \text{ then perform the next step.} \end{aligned} \quad (3.8)$$

2.3 When the partition size  $K$  is  $8 \times 8$ , split  $P$  into the four  $8 \times 8$  sub-macroblocks  $P_j$ .

For  $j = 1$  to 4,

i. when partition size is  $4 \times 8$  or  $8 \times 4$ :

if  $e_j = 0$ , then perform Step 3;  
if  $e_j = 1$ , then perform the next step (3.9)

ii. when partition size is  $8 \times 8$  or  $4 \times 4$ :

if  $e_j = 0$ , then perform the next step;  
if  $e_j = 1$ , then perform Step 3. (3.10)

3. According to Eq. (3.4), compute the best partition size, and decide the number of bits which can be hidden in this block. Take away from  $E$  these bits.
4. Repeat the above steps to encode more bits in the remaining portion of  $E$  until no more is left.

For example, if the user key is  $R = 3735$ , then  $R' = 3735/256 = 141_{10} = 10001101_2$ . Now, suppose that a secret message character  $D_1 = 'a'$  is to be embedded, whose corresponding binary form is  $01100001_2$ . Then, the encrypted form of  $D_1$  is  $E_1 = 01100001 \oplus 10001101 = 11101100_2$ . Similarly, if  $D_2 = 'b.'$  with binary form being  $01100010_2$ , then  $E_2 = 01100010 \oplus 10001101 = 11101111_2$ . Together, we get  $E = E_1E_2 = \underline{11101100}11101111_2$ . We may choose the macroblock partition size of  $16 \times 16$  to hide the first bit or choose four sub-macroblock partitioning  $8 \times 8$ ,  $4 \times 4$ ,  $8 \times 8$ ,  $4 \times 8$  further to hide the first four bits of  $E$ . We use the Lagrangian cost function to decide the best partition size. If the best partition is  $16 \times 16$ , we hide one bit in this block.

A flowchart of the optimally hiding process for  $P$  macroblocks is shown in Figure 3.8.

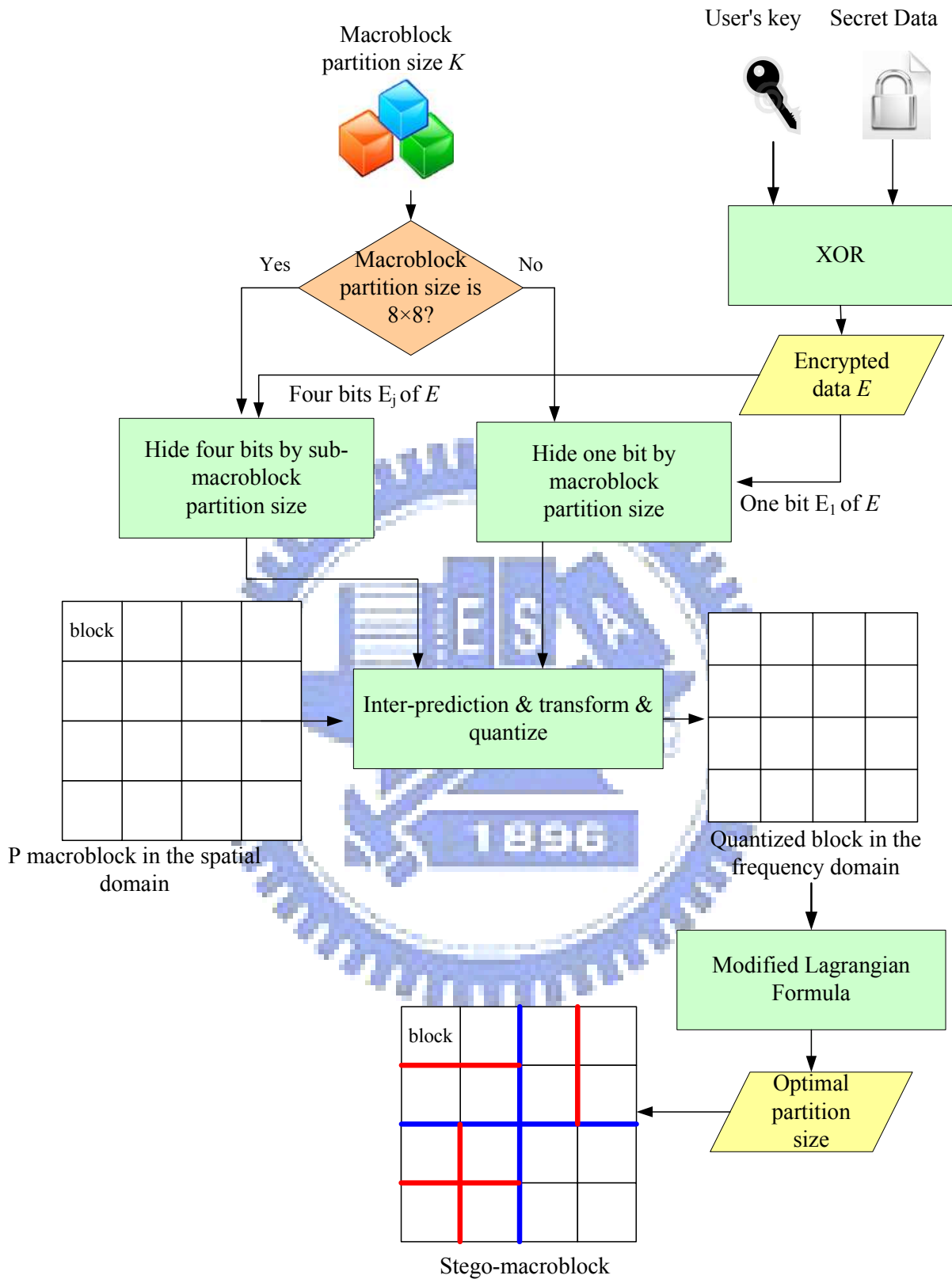


Figure 3.8 Flowchart of the hiding process for P macroblocks.

## 3.4 Extraction of Secret Data from H.264/AVC Videos

In this section, the proposed methods for extracting the hidden data from an input H.264/AVC stego-video will be described. An illustration of the proposed data extraction method is illustrated in Figure 3.9. In Section 3.3.1 and Section 3.3.2 the two processes for extracting data from I macroblocks will be described. Next, the process for extracting data from P macroblocks will be described in Section 3.3.3.

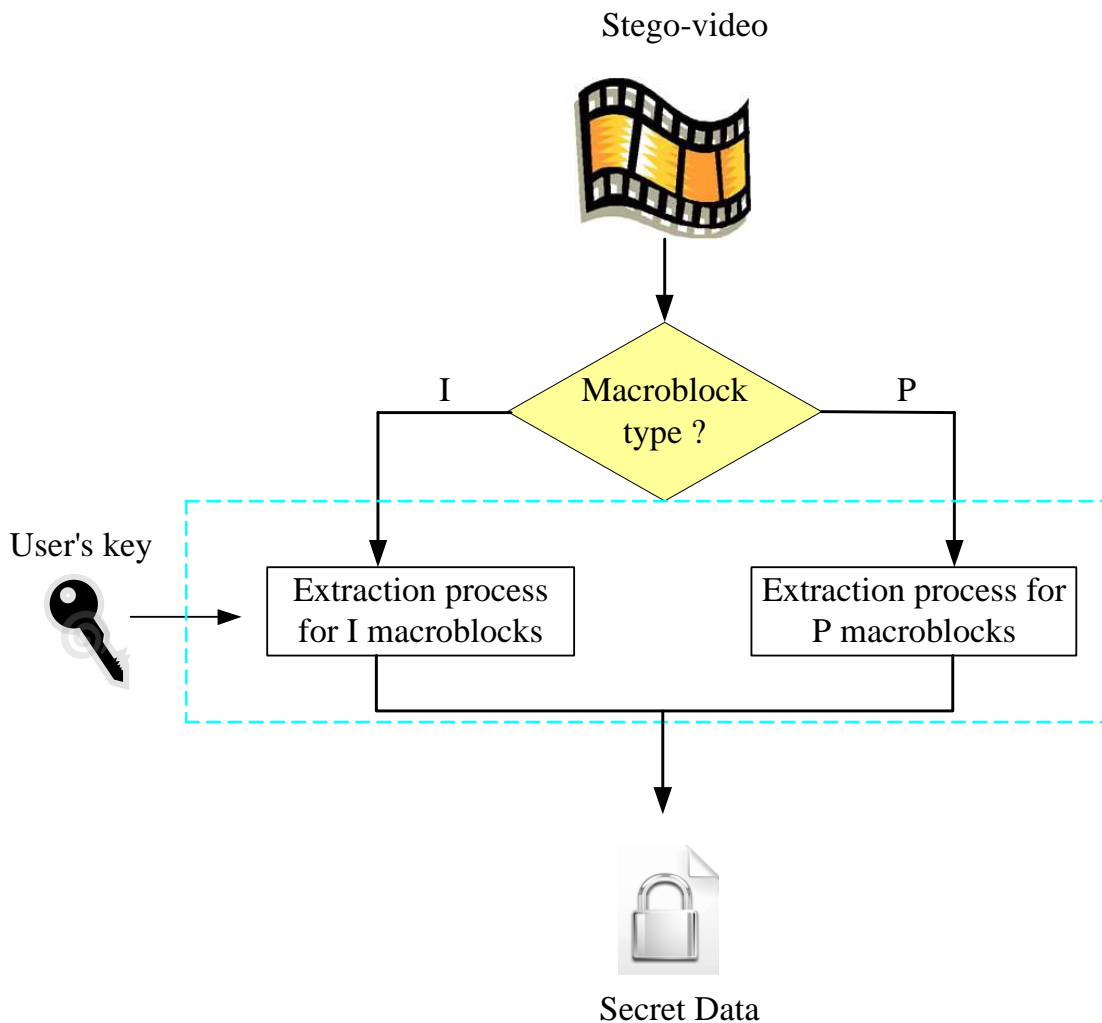


Figure 3.9 Illustration of the proposed extraction method.

### 3.4.1 Process for Extraction of Data from I Macroblocks by Proposed Large-Volume Method

The proposed data extraction process retrieves the adopted prediction modes for the I macroblocks from the bitstream of the stego-video. The prediction mode and the user's key then are taken as input to the extraction process for I macroblocks of the proposed large-volume data hiding method. The detailed algorithm is described in the following.

**Algorithm 3.4:** extraction process for I macroblocks of the large-volume method.

**Input:** the prediction mode  $M$  and a user's key  $R$ .

**Output:** an extracted data file  $D$ .

**Steps:**

1. Extract the prediction mode  $M$  from blocks.
2. Transform every six prediction modes  $M_6$  (novenary) into nineteen bits  $E_j$  of encrypted data  $E$ .
3. For every eight bits  $E_i$  of the  $E$ , perform the following steps.
  - 2.1 Compute the remainder  $R'$  by dividing  $R$  by 256.
  - 2.2 Compute the 8-bit code of each character  $D_i$  of the secret data  $D$  as follows and transform the obtained codes into original characters to get the embedded message:

$$D_i = E_i \oplus R' . \quad (3.11)$$

For example, if the user key is  $R = 3735$ , then  $R' = 3735/256 = 141_{10} = 10001101_2$ . Suppose the extracted six prediction modes are  $M_1 = 8$ ,  $M_2 = 1$ ,  $M_3 = 8$ ,  $M_4 = 5$ ,  $M_5 = 6$ ,  $M_6 = 3$  which, when combined together, becomes the novenary

number 818563<sub>9</sub>. After being converted into binary, it becomes the binary data  $\underline{111011001110111111}_2$  whose first eight bits  $E_1$  may be converted to get a secret message character 'a' after conducting the exclusive-OR operations  $E_1 \oplus R' = 11101100_2 \oplus 10001101_2 = 01100001_2$  to get the binary code 01100001<sub>2</sub> of 'a.'

### 3.4.2 Process for Extraction of Data from I

#### Macroblocks by Proposed Optimal Method

The proposed data extraction process retrieves the adopted prediction modes of I macroblocks from the bitstream of the stego-video first. After entropy-decoding the stego-video, the highest-frequency quantized coefficient of each luma 4×4 block of I macroblocks is retrieved. Then we take these coefficients, the prediction modes and the user's key as input to the data extraction process for I macroblocks. The detailed algorithm is described in the following.

**Algorithm 3.5:** extraction process for I macroblocks for the proposed optimal method.

**Input:** a quantized luma 4×4 block of the I macroblock in the frequency domain  $I$ , the prediction mode  $M$  and a user's key  $R$ .

**Output:** an extracted data file  $D$ .

**Steps:**

1. Obtain the number of bits  $N_i$  hidden in the mode  $M$  from the highest-frequency coefficient  $C$  of  $I$  according to the following rules:

$$\begin{aligned}
 &\text{if } C = 0, \text{ then set } N_i = 0; \\
 &\text{if } C = 1, \text{ then set } N_i = 1; \\
 &\text{if } C = -1, \text{ then set } N_i = 2; \\
 &\text{if } C = 2, \text{ then set } N_i = 3; \\
 &\text{if } C = -2, \text{ then set } N_i = 4.
 \end{aligned} \tag{3.12}$$

2. Extract the last  $N_i$  bit(s)  $e$  of  $M$  as part of encrypted data  $E$ .



3. For every eight bits  $E_i$  of  $E$ , perform the following steps.
  - 3.1 Compute the remainder  $R'$  of dividing  $R$  by 256.
  - 3.2 Set each character  $D_i$  of the secret data  $D$  according to Equation (3.11)

For example, if the prediction mode is  $M_7$  whose index is  $0111_2$  in binary and  $N_i = 4$ , then we can get the message data  $e = 1110_2$ .

A flowchart of the extraction process is shown in Figure 3.10

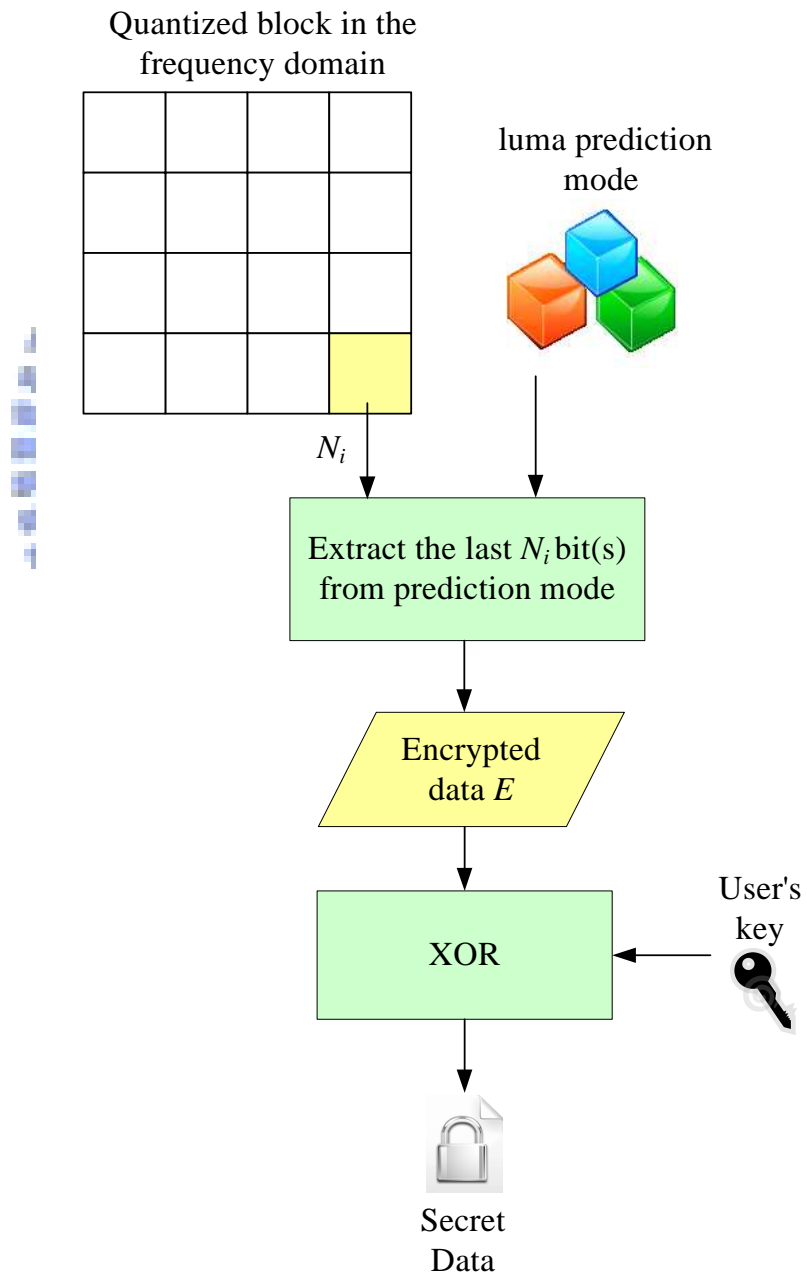


Figure 3.10 Flowchart of the extraction process for I macroblocks of optimal method.

### 3.4.3 Process for Extraction of Data from P

#### Macroblocks

The proposed data extraction process retrieves the macroblock partition sizes of the P macroblocks from the stego-video first; if the macroblock partition size is  $8 \times 8$ , we get the sub-macroblock partition size of P macroblocks further, and take the partition size and the user's key as input to the data extraction process for P macroblocks. A flowchart of the data extraction process for P macroblocks is shown in Figure 3.11 and the detailed algorithm is described in the following.

**Algorithm 3.6:** data extraction process for P macroblocks.

**Input:** a macroblock partition size  $P$  (and four sub-macroblock partition size  $P_j$ ) and a user's key  $R$ .

**Output:** an extracted data file  $D$ .

**Steps:**

1. Extract a bit  $e_1$  or four bits  $e_j$  as part of the encrypted data  $E$  from the  $P$  according to the following rules.
  - 1.1 When  $P$  is  $16 \times 16$ ,  $8 \times 16$ , or  $16 \times 8$ , extract a bit  $e_1$  from the macroblock partition size  $P_1$  in the following way:
$$\begin{aligned} &\text{if } P_1 \text{ is } 16 \times 16, \text{ then set } e_1 = 1; \\ &\text{if } P_1 \text{ is } 8 \times 16, 16 \times 8 \text{ then set } e_1 = 0. \end{aligned} \tag{3.13}$$
  - 1.2 When  $P$  is  $8 \times 8$ , extract 4 bits  $e_j$  from four sub-macroblocks partition size  $P_j$  in the following way:  
for  $j = 1$  to 4:
$$\begin{aligned} &\text{if } P_j \text{ is } 8 \times 8, 4 \times 4, \text{ then set } e_j = 1; \\ &\text{if } P_j \text{ is } 4 \times 8, 8 \times 4, \text{ then set } e_j = 0. \end{aligned} \tag{3.14}$$
2. For every 8 bits  $E_i$  of  $E$ , perform the following steps.
  - 2.1 Compute the remainder  $R'$  of dividing  $R$  by 256.

2.2 Set each character  $D_i$  of the secret data  $D$  according to Equation (3.11).

For example, if the extracted macroblock partition size is  $P = 8 \times 8$  and the four sub-macroblock partition sizes are  $P_1 = 8 \times 8$ ,  $P_2 = 4 \times 8$ ,  $P_3 = 4 \times 4$ ,  $P_4 = 8 \times 4$ , then we can get four message bits  $e_1 = 1$ ,  $e_2 = 0$ ,  $e_3 = 1$ ,  $e_4 = 0$ .

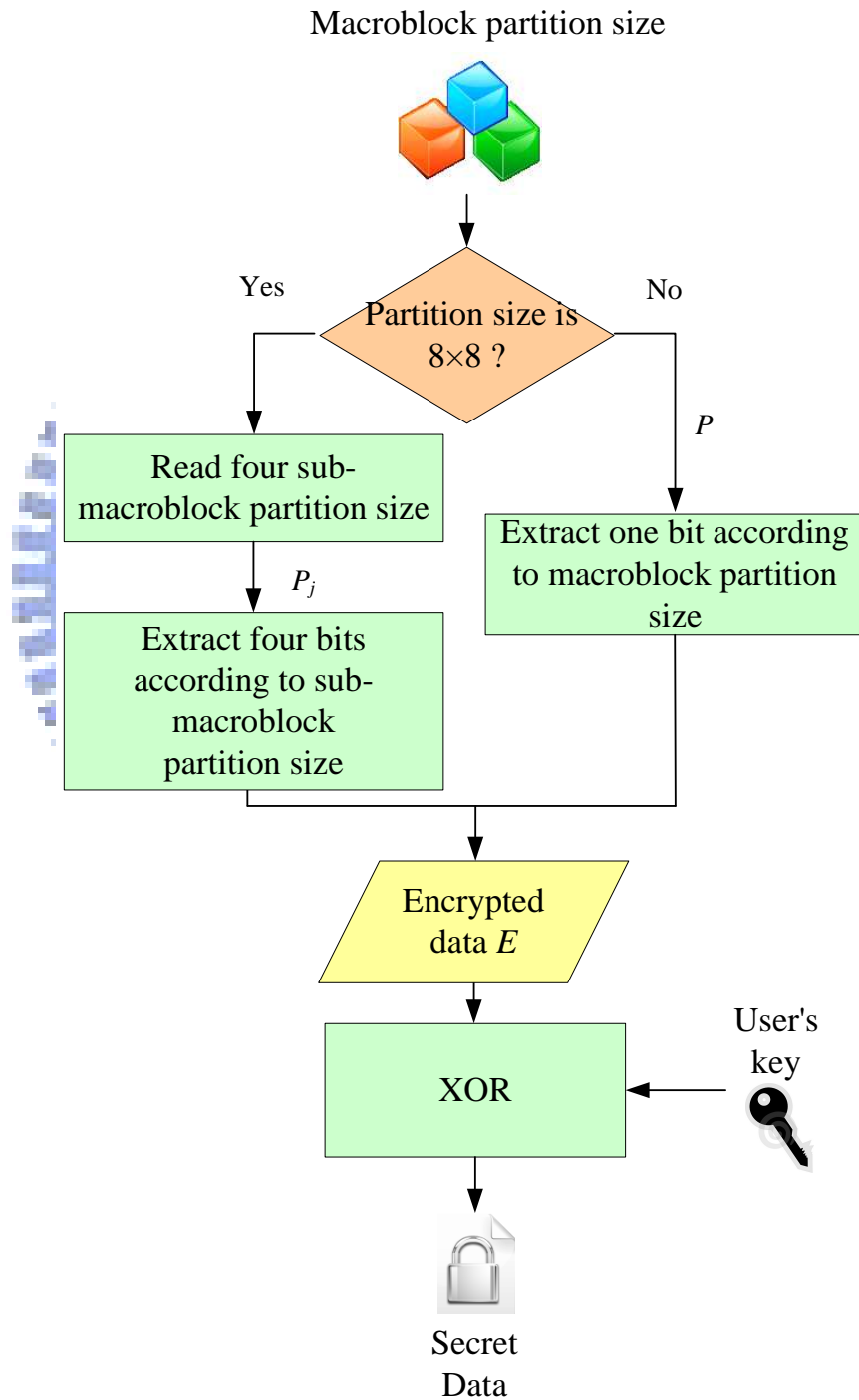


Figure 3.11 Flowchart of the extraction process for P macroblocks.

# 3.5 Experimental Results

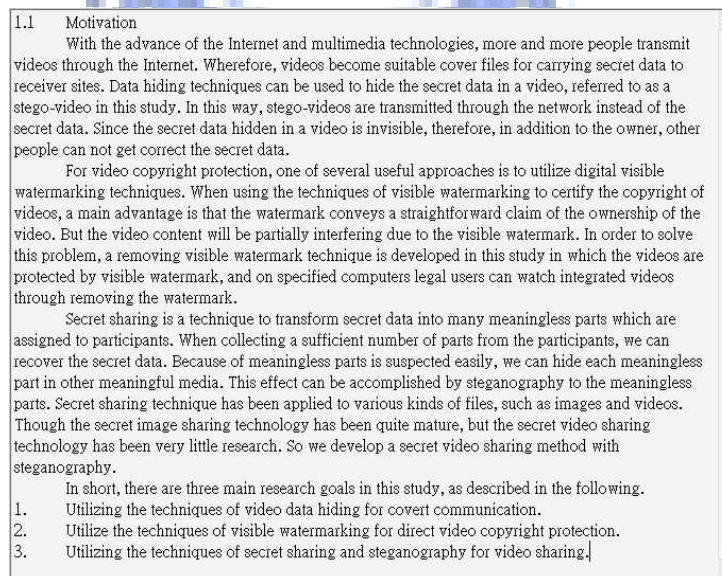
## 3.5.1 Experimental Results of Large-Volume Method

In our experiments, the proposed video sharing algorithm has been integrated into the H.264 reference software JM12.4 [13]. The most important configuration parameters of the JM12.4 are shown in Table 3.2; other parameters are kept to retain their default values. An H.264/AVC video in CIF (352×288 pixels) format was used in our experiments.

Table 3.2 Configuration parameters.

<b><i>Profile</i></b>	Baseline
<b><i>Number of frames to be coded</i></b>	5
<b><i>Period of I-pictures</i></b>	1

The secret data with the size of 2262 bytes used in the experiments are shown in Figure 3.12. Four of five frames of the input video and the resulting stego-video are shown in Figure 3.13. The extracted data are shown in Figure 3.14.



1.1 Motivation

With the advance of the Internet and multimedia technologies, more and more people transmit videos through the Internet. Wherefore, videos become suitable cover files for carrying secret data to receiver sites. Data hiding techniques can be used to hide the secret data in a video, referred to as a stego-video in this study. In this way, stego-videos are transmitted through the network instead of the secret data. Since the secret data hidden in a video is invisible, therefore, in addition to the owner, other people can not get correct the secret data.

For video copyright protection, one of several useful approaches is to utilize digital visible watermarking techniques. When using the techniques of visible watermarking to certify the copyright of videos, a main advantage is that the watermark conveys a straightforward claim of the ownership of the video. But the video content will be partially interfering due to the visible watermark. In order to solve this problem, a removing visible watermark technique is developed in this study in which the videos are protected by visible watermark, and on specified computers legal users can watch integrated videos through removing the watermark.

Secret sharing is a technique to transform secret data into many meaningless parts which are assigned to participants. When collecting a sufficient number of parts from the participants, we can recover the secret data. Because of meaningless parts is suspected easily, we can hide each meaningless part in other meaningful media. This effect can be accomplished by steganography to the meaningless parts. Secret sharing technique has been applied to various kinds of files, such as images and videos. Though the secret image sharing technology has been quite mature, but the secret video sharing technology has been very little research. So we develop a secret video sharing method with steganography.

In short, there are three main research goals in this study, as described in the following.

1. Utilizing the techniques of video data hiding for covert communication.
2. Utilize the techniques of visible watermarking for direct video copyright protection.
3. Utilizing the techniques of secret sharing and steganography for video sharing.

Figure 3.12 The secret data file.

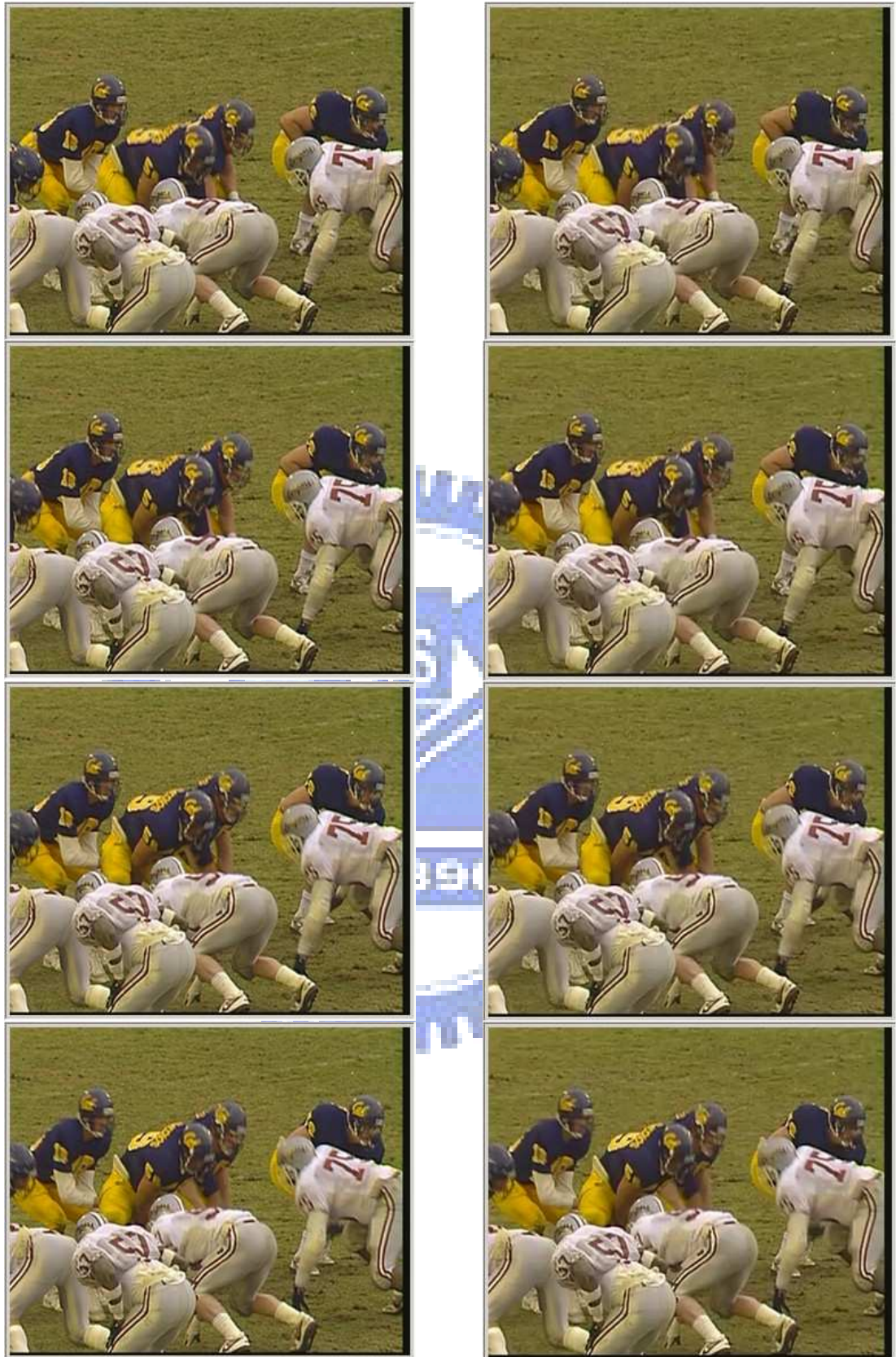


Figure 3.13 The 1<sup>st</sup> to 4<sup>th</sup> frames (III) of original video (left) and stego-video (right).

1.1 Motivation

With the advance of the Internet and multimedia technologies, more and more people transmit videos through the Internet. Wherefore, videos become suitable cover files for carrying secret data to receiver sites. Data hiding techniques can be used to hide the secret data in a video, referred to as a stego-video in this study. In this way, stego-videos are transmitted through the network instead of the secret data. Since the secret data hidden in a video is invisible, therefore, in addition to the owner, other people can not get correct the secret data.

For video copyright protection, one of several useful approaches is to utilize digital visible watermarking techniques. When using the techniques of visible watermarking to certify the copyright of videos, a main advantage is that the watermark conveys a straightforward claim of the ownership of the video. But the video content will be partially interfering due to the visible watermark. In order to solve this problem, a removing visible watermark technique is developed in this study in which the videos are protected by visible watermark, and on specified computers legal users can watch integrated videos through removing the watermark.

Secret sharing is a technique to transform secret data into many meaningless parts which are assigned to participants. When collecting a sufficient number of parts from the participants, we can recover the secret data. Because of meaningless parts is suspected easily, we can hide each meaningless part in other meaningful media. This effect can be accomplished by steganography to the meaningless parts. Secret sharing technique has been applied to various kinds of files, such as images and videos. Though the secret image sharing technology has been quite mature, but the secret video sharing technology has been very little research. So we develop a secret video sharing method with steganography.

In short, there are three main research goals in this study, as described in the following.

1. Utilizing the techniques of video data hiding for covert communication.
2. Utilize the techniques of visible watermarking for direct video copyright protection.
3. Utilizing the techniques of secret sharing and steganography for video sharing.

Figure 3.14 The extracted data file.

### 3.5.2 Experimental Results of Optimal Method

The proposed optimal data hiding algorithm was integrated into the H.264 reference software JM12.4. The most important configuration parameters of the JM12.4 are shown in Table 3.3; other parameters are kept to retain their default values. Several video sequences, Foreman, Football, Mobile and Tempete, in CIF (352×288 pixels) format were used in our experiments.

Table 3.3 Configuration parameters.

<b><i>Profile</i></b>	baseline
<b><i>Number of frames to be coded</i></b>	10
<b><i>Period of I-pictures</i></b>	5
<b><i>RD Optimization</i></b>	High complexity mode

The performance of the optimal data hiding algorithms was evaluated with the number of bits hidden (NBH), the Peak-Signal-to-Noise-Ratio increase in the Y color

space (PSNRI), the bit rate increase (BRI) and the subjective perception testing by comparing the original video and the stego video frames with hidden secret data. The secret data with the size of 2262 bytes used in the experiments are shown in Figure 3.12. Four of ten frames of the input video and the resulting stego-video are shown in Figure 3.15. The extracted data are shown in Figure 3.16. Finally the NBH, PSNRI and BRI values for several video sequences are shown in Table 3.4. From the above results and data, it can be observed that the proposed method can embed the secret data into H.264/AVC videos imperceptibly with light bit rate increasing.



Figure 3.15 The 4<sup>th</sup> to 7<sup>th</sup> frames (PIIP) of original video (left) and stego-video (right).

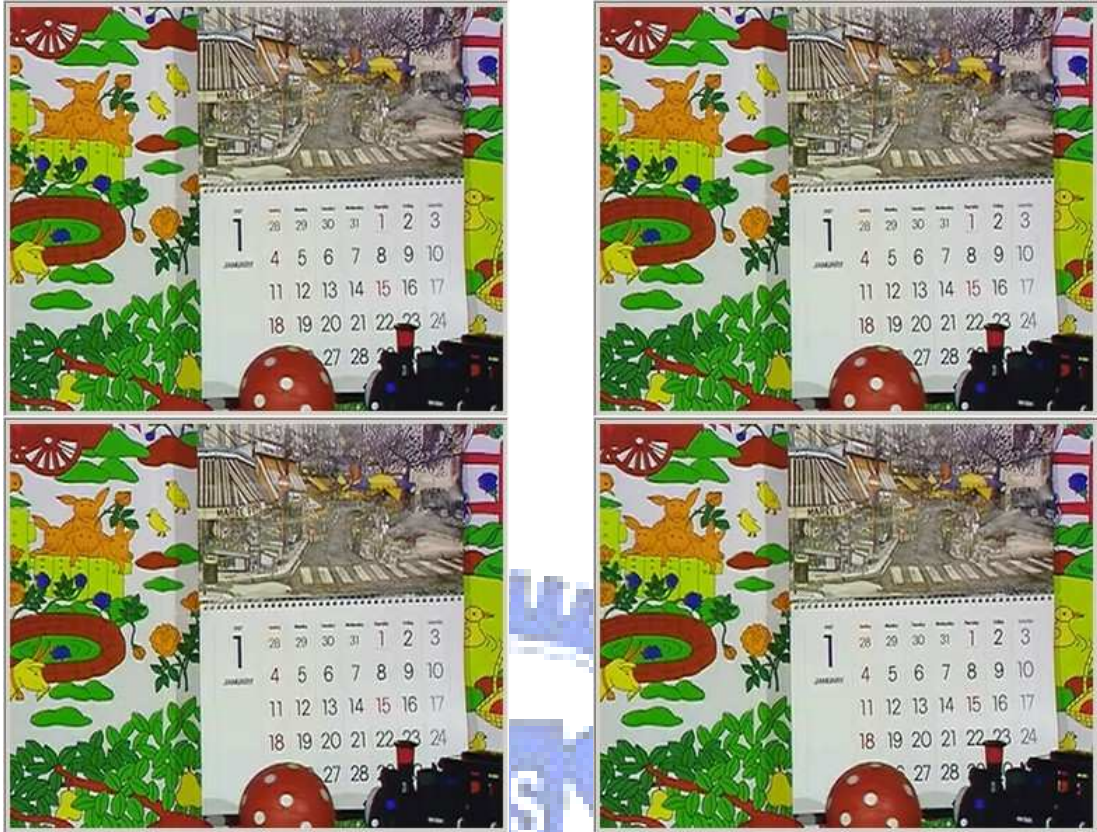


Figure 3.15 The 4th to 7th frames (PIIP) of original video (left) and stego-video (right).  
(continued)

1.1 Motivation

With the advance of the Internet and multimedia technologies, more and more people transmit videos through the Internet. Wherefore, videos become suitable cover files for carrying secret data to receiver sites. Data hiding techniques can be used to hide the secret data in a video, referred to as a stego-video in this study. In this way, stego-videos are transmitted through the network instead of the secret data. Since the secret data hidden in a video is invisible, therefore, in addition to the owner, other people can not get correct the secret data.

For video copyright protection, one of several useful approaches is to utilize digital visible watermarking techniques. When using the techniques of visible watermarking to certify the copyright of videos, a main advantage is that the watermark conveys a straightforward claim of the ownership of the video. But the video content will be partially interfering due to the visible watermark. In order to solve this problem, a removing visible watermark technique is developed in this study in which the videos are protected by visible watermark, and on specified computers legal users can watch integrated videos through removing the watermark.

Secret sharing is a technique to transform secret data into many meaningless parts which are assigned to participants. When collecting a sufficient number of parts from the participants, we can recover the secret data. Because of meaningless parts is suspected easily, we can hide each meaningless part in other meaningful media. This effect can be accomplished by steganography to the meaningless parts. Secret sharing technique has been applied to various kinds of files, such as images and videos. Though the secret image sharing technology has been quite mature, but the secret video sharing technology has been very little research. So we develop a secret video sharing method with steganography.

In short, there are three main research goals in this study, as described in the following.

1. Utilizing the techniques of video data hiding for covert communication.
2. Utilize the techniques of visible watermarking for direct video copyright protection.
3. Utilizing the techniques of secret sharing and steganography for video sharing.

Figure 3.16 The extracted data file.



Table 3.4 NBH, PSNRI and BRI values for several video sequences ( $\gamma_i, \gamma_p = 250$ ).

	<i>football</i>	<i>foreman</i>	<i>mobile</i>	<i>tempe</i>
<i>Average of PSNRI/NBH in I macroblocks (%)</i>	-0.031761	-0.052796	-0.024492	-0.030125
<i>Average of BRI/NBH in I macroblocks</i>	7.668156	13.382799	4.941370	6.188712
<i>Average of NBH in I macroblocks</i>	1567	593	3650	2321
<i>Average of PSNRI/NBH in P macroblocks (%)</i>	-0.189027	-0.162934	-0.235225	-0.250322
<i>Average of BRI/NBH in P macroblocks</i>	27.370086	11.071856	17.341488	13.365207
<i>Average of NBH in P macroblocks</i>	1043	835	981	868

### 3.6 Discussions and Summary

In this chapter, we proposed a large-volume data hiding method and an optimal data hiding method that can be used to hide data into I macroblocks and P macroblocks. The optimal data hiding method not only considers hiding capacity of secret data and imperceptibility, but also considers bit rates. Therefore, the method is suitable for covert communication applications, especially when we need to transmit an H.264/AVC video in a low bit rate network.

# Chapter 4

## Copyright Protection of H.264/AVC

### Videos by Watermarking and

### Display Control on Specified

### Computers



## 4.1 Introduction

With the fast development of network techniques, web 2.0 becomes 'hot' in recent years. Everyone can share videos on the Internet in various forms. However, these videos on the web can be easily downloaded and might be distributed to other people illegally. Hence, development of methods for protecting the copyright of videos is essential. In this study, a removable visible watermarking method with a scheme for video display control on specified computers is proposed for copyright protection of H.264/AVC videos, and is described in this chapter.

In Section 4.1.1, some related problem definitions are given and in Section 4.1.2, the basic idea of the proposed method are presented. In Section 4.2, the proposed visible watermark embedding method with a scheme for video display control on specified computers is described, and a corresponding visible watermark removal method is stated in Section 4.3. In Section 4.4, several experimental results of the

proposed method will be shown. Finally, some discussions and a summary will be made in last section of this chapter.

### 4.1.1 Problem Definition

To solve the problem of protecting the copyright of videos mentioned above, two important issues are how to prevent a video from being illegally distributed and how to implement a scheme for display control on specified computers. Another issue is how to embed a visible watermark in an H.264/AVC video with a secret key, with the visible watermark being sufficiently robust against attacks from unauthorized users. A related issue is how to remove the embedded visible watermark to recover the original video.

### 4.1.2 Proposed Ideas

To deal with all the above-mentioned issues, it is proposed in this study to proceed in the following way:

1. A user downloads an *active program* from a server site to read the identification information of the local computer (called *computer information* below) and uploads it together with a selected secret key to the server site.
2. The server embeds a visible watermark together with the received computer information into a video selected by the user, and sends the resulting stego-video to the user site.
3. The user downloads an *active player* to display the protected video after removing the embedded visible watermark using the key he/she provides.
4. If the user distributes the stego-video together with the key and the active player to a third-party user, the user cannot remove the embedded visible watermark to view the video clearly because the active player will check the correctness of not

only the key but also the local computer information to decide whether or not to remove the visible watermark.

An illustration of the proposed idea is shown in Figure 4.1.

## **4.2 Proposed Scheme for Display Control on Specified Computers and Embedding Visible Watermarks in H.264/AVC Videos**

In this section, the proposed scheme of video display control on specified computers and embedding visible watermarks into H.264/AVC videos will be described. In Section 4.2.1, the process for video display control on specified computers will be described. In Section 4.2.2, the process of embedding visible watermarks is stated.

### **4.2.1 Process for Video Display Control on Specified Computers**

In order to protect the copyright of an H.264/AVC video and avoid the video being distributed illegally, we propose the idea of controlling video displays only on pre-specified computers. While a user requests a download service from the video supplier, the active program sent to the user will access the CPU and disk information of the user's computer by *Windows API*, which is in the Microsoft core set of application programming interfaces (APIs), and then send the information to the server site for the server to embed a visual watermark together with such information into the video selected by the user. The video together with an active video player

then is sent to the user.

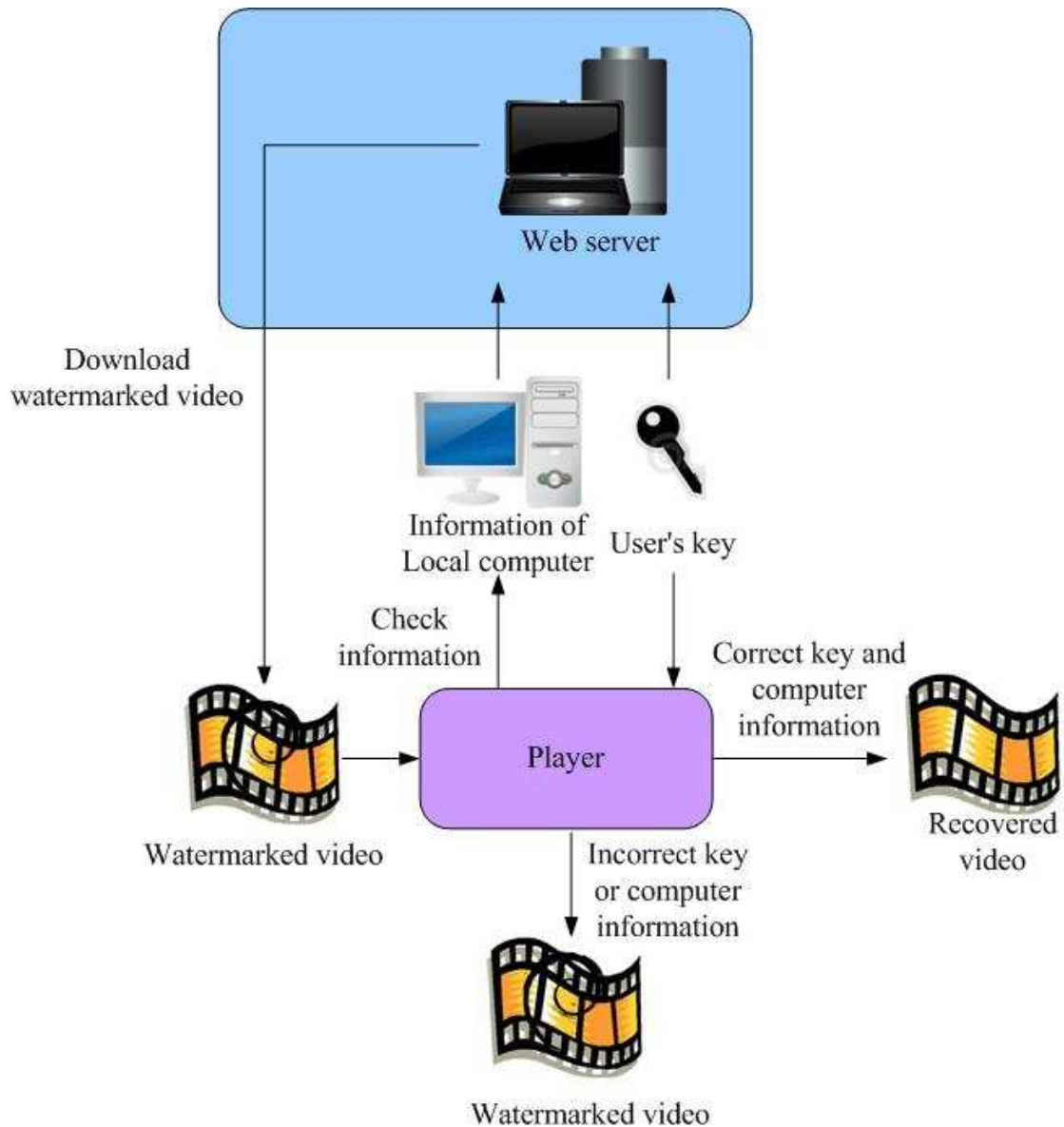


Figure 4.1 Illustration of the proposed idea.

When the user plays the video with the active player, the player will access the CPU, disk information of the *local* computer, and use this information to remove the visual watermark if this information is the same as the embedded computer information in the stego-video. Otherwise, the video will be covered with the previously-embedded visible watermark. Then, supposing that the user duplicates the stego-video and sends the copy to other users, the computer information checking

process will fail on other computers and so the watermark will not be removed, thus claiming the ownership of the video and in the mean time protecting the video from being viewed. An illustration of this checking process is shown in Figure 4.2.

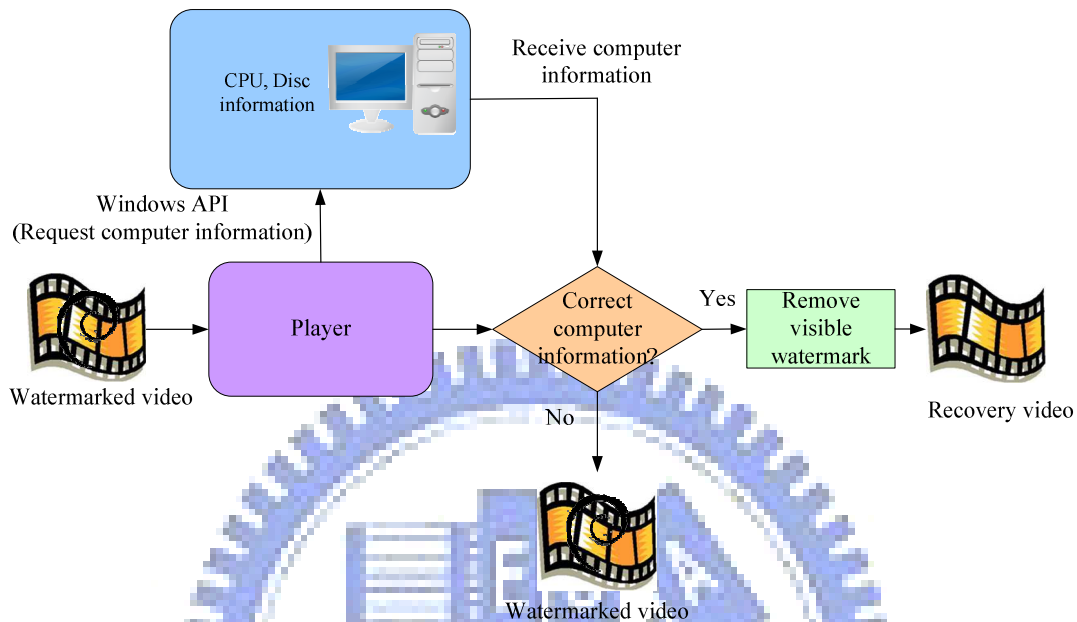


Figure 4.2 The checking process of video display control on specified computers.

## 4.2.2 Process for Embedding Visible Watermarks

A watermark to be embedded into a video is assumed to be a binary image which has only black and white pixels. The embedding process for I and P macroblocks in a given video utilizes a  $16 \times 16$  luminance macroblock of the video to embed a watermark pixel. A new technique of using intra-prediction modes is adopted in the H.264/AVC standard and results in higher compression efficiency than previous standards. A detailed introduction can be found in Section 3.2.1.

The sample values of a prediction block are computed by those of previously encoded and reconstructed blocks, as mentioned previously. Therefore, we cannot modify the transform coefficients directly for embedding a watermark pixel as the

traditional method does, because such modification will cause prediction errors, resulting in the so-called *drift problem*. That is, when a block which is not watermarked refers to a watermarked block, this block will also be watermarked. An illustration of the drift problem is shown in Figure 4.3. So we not only have to embed the visible watermark but also have to prevent the drift problem.

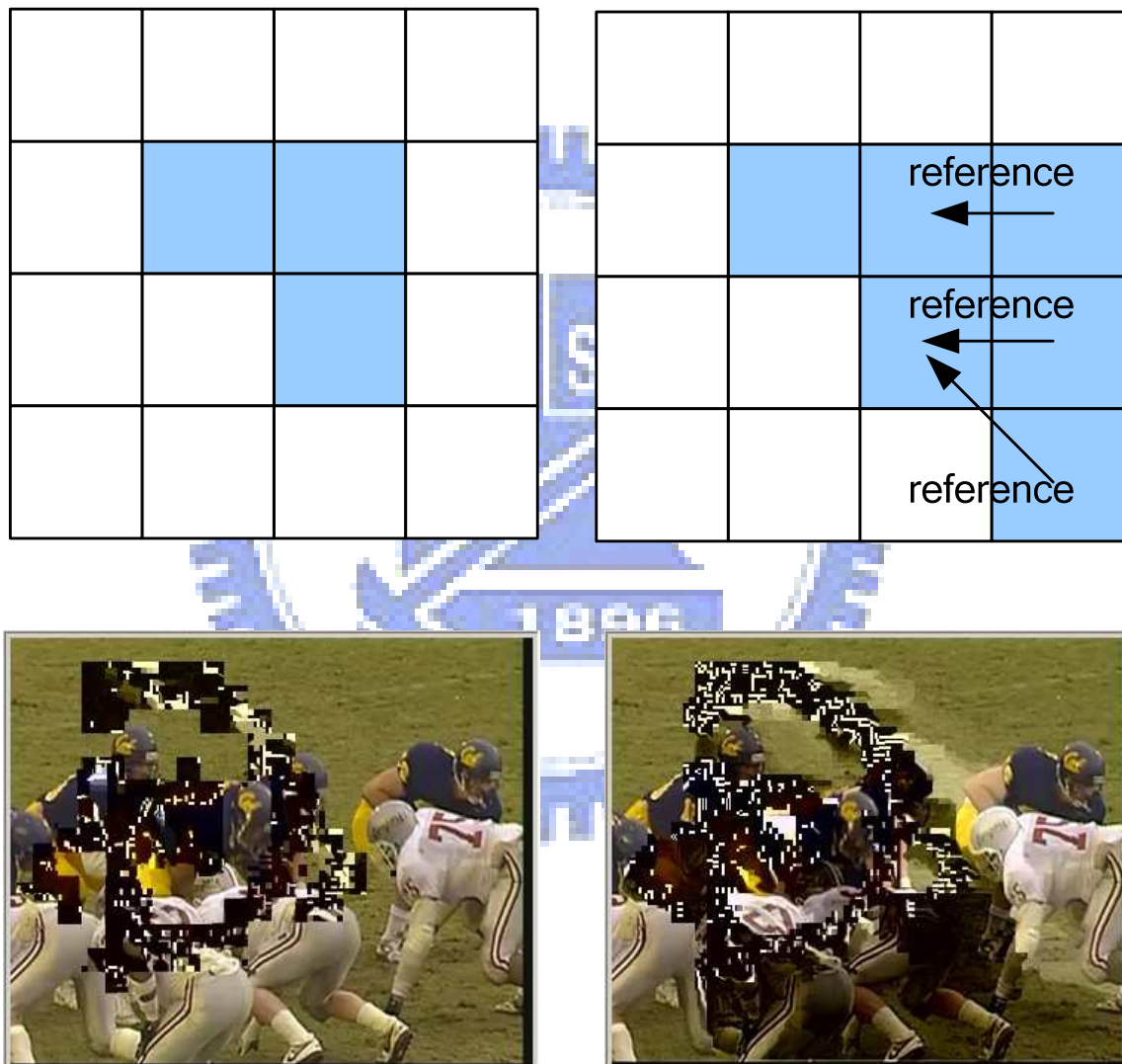


Figure 4.3 Illustration of drift problem.

We propose the use of *multiple slice groups* (described as *Flexible Macroblock Ordering* or FMO in the draft standard) to resolve the drift problem. Multiple slice

groups make map the slice group to coded macroblocks in a number of flexible ways. Figure 4.4 list the seven different types of *macroblock-to-slice-group mappings*. The macroblocks in different slice groups will not refer to one another. Therefore, we can use this feature to set watermarked and non-watermarked macroblocks in different slice groups to avoid the drift problem, and use the DC coefficient to embed the watermark pixel. A flowchart of the embedding process for I and P frames is shown in Figure 4.5 and the detailed algorithm of the proposed process is described in the following.

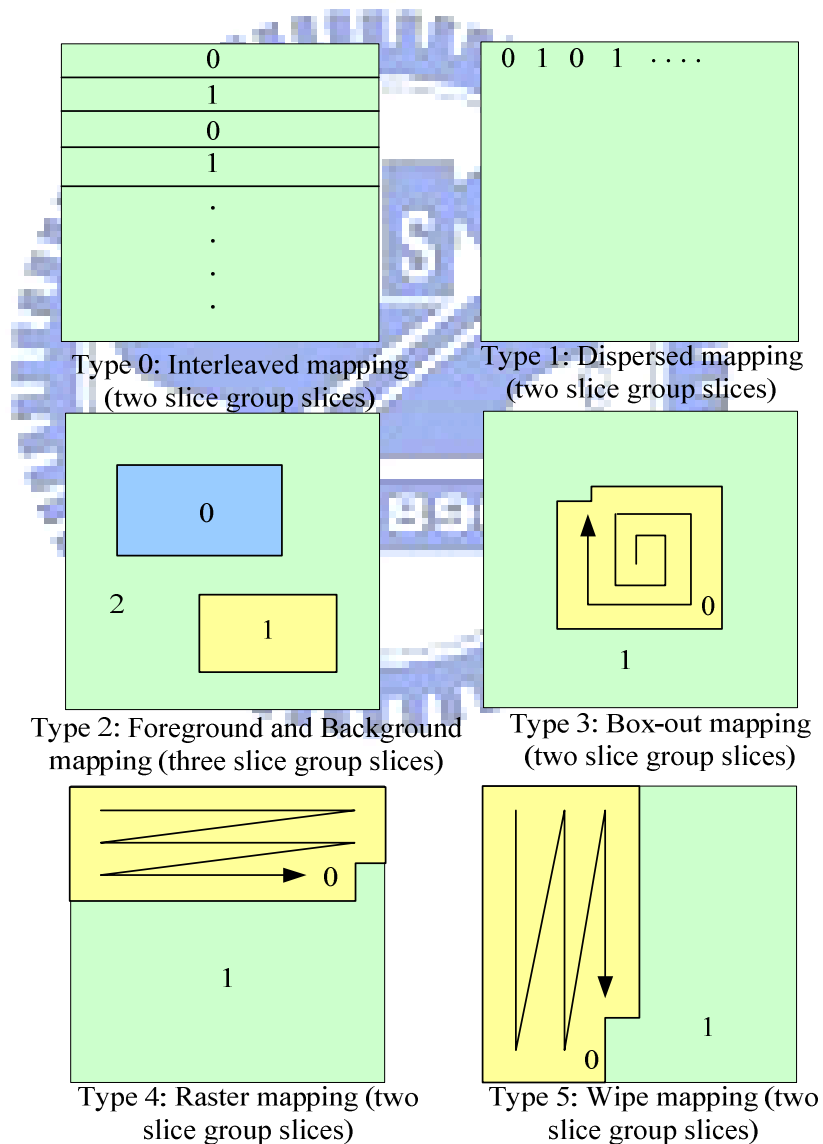


Figure 4.4 Types of macroblock to slice group maps (type 6 is “*Explicit*” which is user-defined).



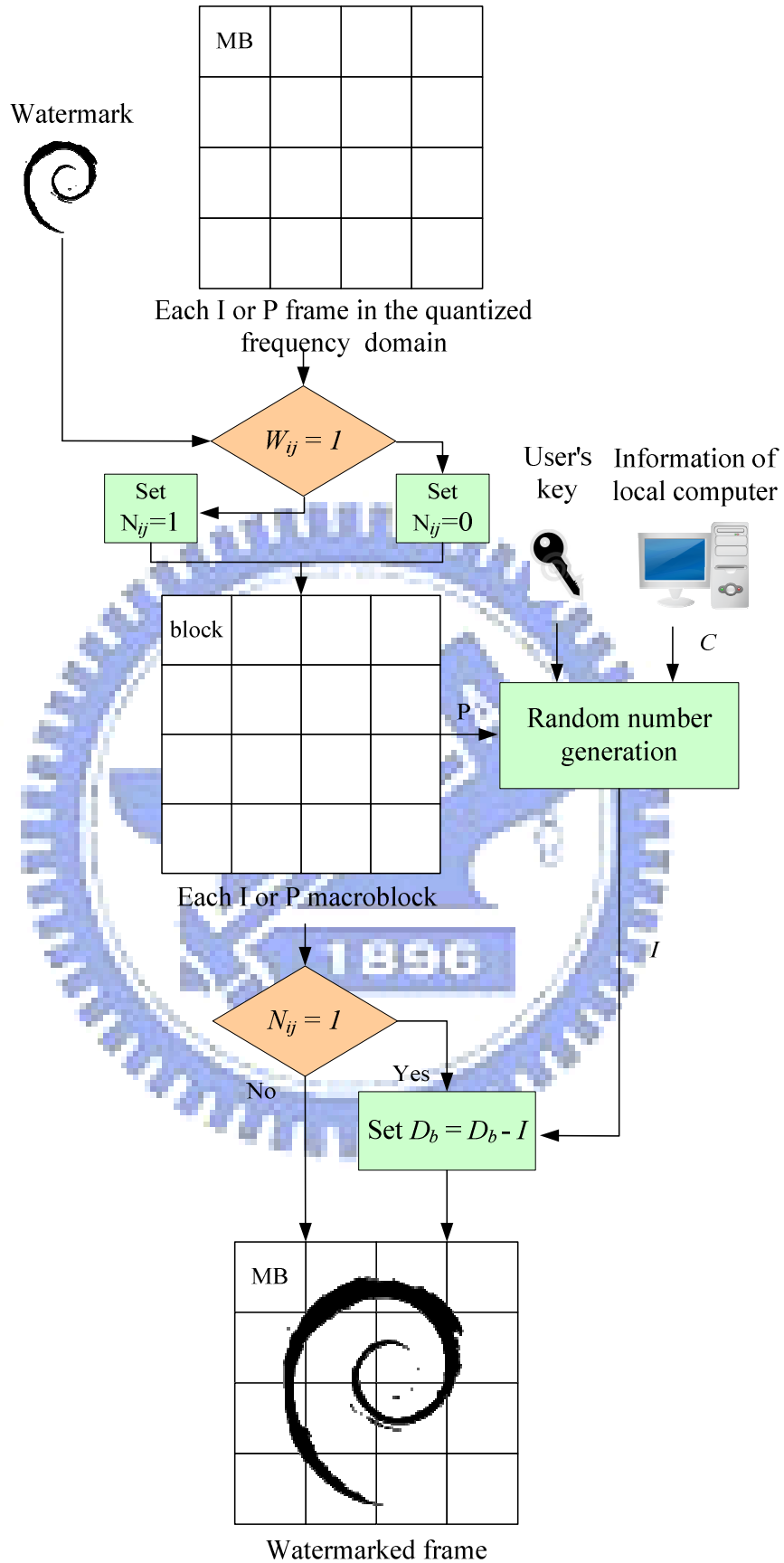


Figure 4.5 Flowchart of the embedding process for I and P frames.

**Algorithm 4.1:** embedding process for I and P frames.

**Input:** an I or P frame  $F$  in the quantized frequency domain, a secret key  $K$ , the information of the local computer  $C$ , a binary watermark image  $W$ , and a random number generator  $f$ .

**Output:** a watermarked frame  $F'$ .

**Steps:**

1. Map each watermark pixel  $w_{ij}$  to the corresponding  $16 \times 16$  macroblock  $M_{ij}$  of  $F$  and use the type-6 slice group mapping “Explicit,” to set the slice group numbers  $N_{ij}$  of all  $M_{ij}$  according to the following rule:

$$\begin{aligned} &\text{if } w_{ij} = 0, \text{ then set } N_{ij} = 0; \\ &\text{if } w_{ij} = 1, \text{ then set } N_{ij} = 1. \end{aligned} \quad (4.1)$$

2. Combine the computer information  $C$  and the secret key  $K$  to form a seed for the random number generator  $f$  to generate a random number  $I$ .
3. Utilize the DC coefficient  $D_b$  of each block in  $M_{ij}$  to embed a visible watermark pixel as follows:

$$\begin{aligned} &\text{if } N_{ij} = 0, \text{ then keep } D_b \text{ unchanged;} \\ &\text{if } N_{ij} = 1, \text{ then set } D_b = D_b - I. \end{aligned} \quad (4.2)$$

## 4.3 Recovery of Original H.264/AVC Videos by Removing Visible Watermarks

In this section, the process of recovery of watermarked I and P frames will be described. The activation of the recovery process depends on whether the player is able to get the right secret key and the right information of the local computer. The main task is to find out the watermarked blocks and to remove the watermark pixels

by using the secret key and received the information of the local computer. A flowchart of the recovery process is shown in Figure 4.6 and the detailed algorithm is described in the following.

**Algorithm 4.2:** the process for recovery of I and P frames.

**Input:** a watermarked I or P frame  $F'$ , a secret key  $R$ , the information of the local computer  $C$ , and a random number generator  $f$ .

**Output:** a recovered I or P frame  $F$ .

**Steps:**

1. For each  $16 \times 16$  macroblock  $M_{ij}$  of  $F'$ , combine the information of the local computer  $C$ , the secret key  $R$ , and the position  $P$  of  $M_{ij}$  to form a seed for the random number generator  $f$  to generate a random number  $I$ .
2. Check the slice group number  $N_{ij}$  of  $M_{ij}$  and modify the DC coefficient  $D_b$  of the blocks in  $M_{ij}$  to remove visible watermark pixels according to the following rule

$$\begin{aligned} &\text{If } N_{ij} = 0, \text{ then keep } D_b \text{ unchanged;} \\ &\text{If } N_{ij} = 1, \text{ then set } D_b = D_b + I. \end{aligned} \tag{4.3}$$

## 4.4 Experimental Results

The proposed watermarking algorithm has been integrated into the H.264 reference software JM12.4. The most important configuration parameters of the JM12.4 are shown in Table 4.1; other parameters are kept to retain their default values. An H.264/AVC video in the CIF (352×288 pixels) format was used to embed a watermark image with size 16×16 in our experiments.

The binary watermark image is shown in Figure 4.7. Six frames of the original video are shown in Figure 4.8. The corresponding six frames of the watermarked video are shown in Figure 4.9. The corresponding six frames of the recovered video

are shown in Figure 4.10. If the secret key or the information of the local computer is wrong, the watermarked video will be shown in Figure 4.11.

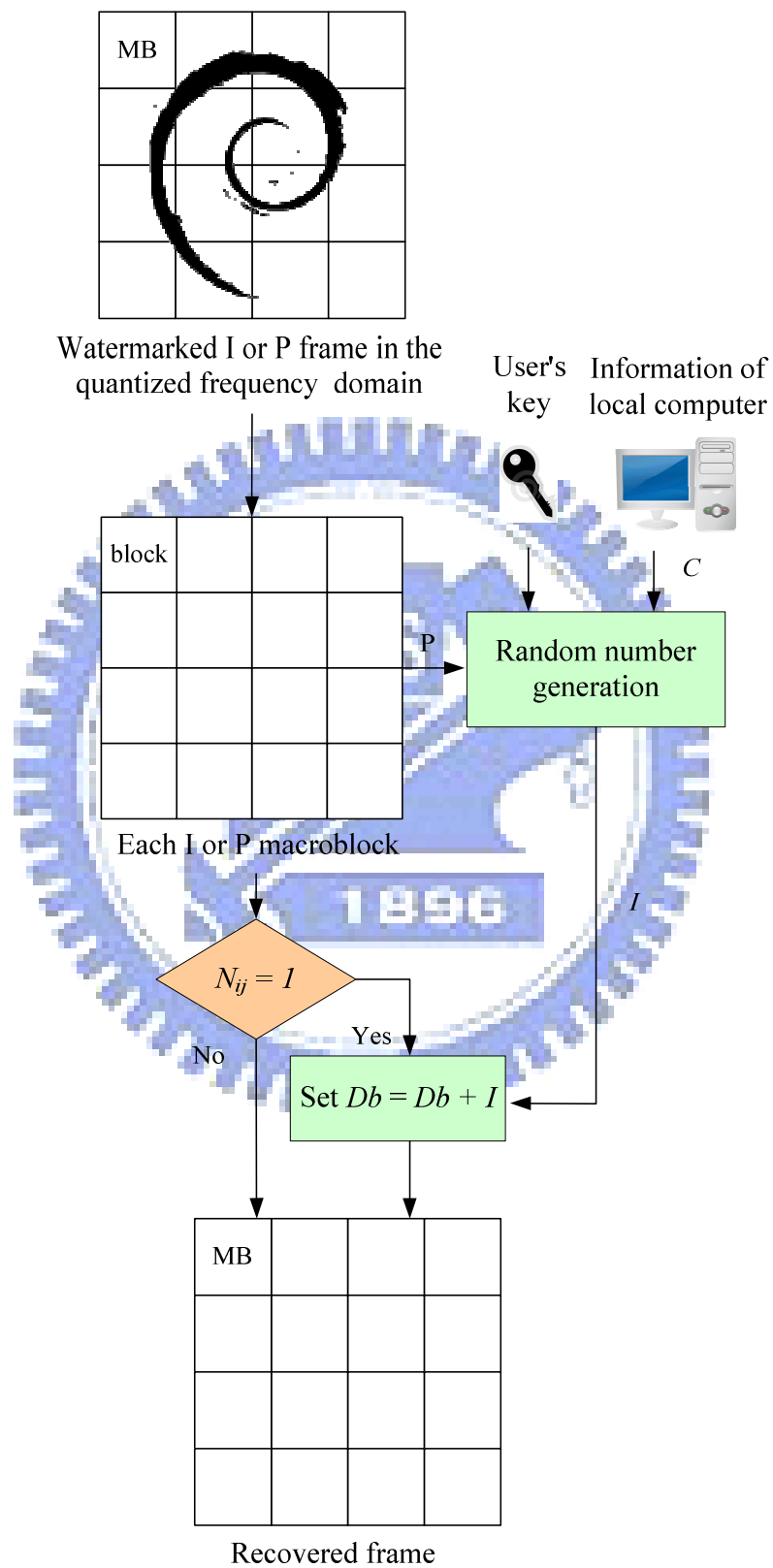


Figure 4.6 Flowchart of the recovery process for I and P frames.

Table 4.1 Configuration parameters

<i>Profile</i>	<i>baseline</i>
<i>Number of frames to be coded</i>	6
<i>num_slice_groups_minus1</i>	1
<i>slice_group_map_type</i>	6



Figure 4.7 A watermark binary image with size 16×16

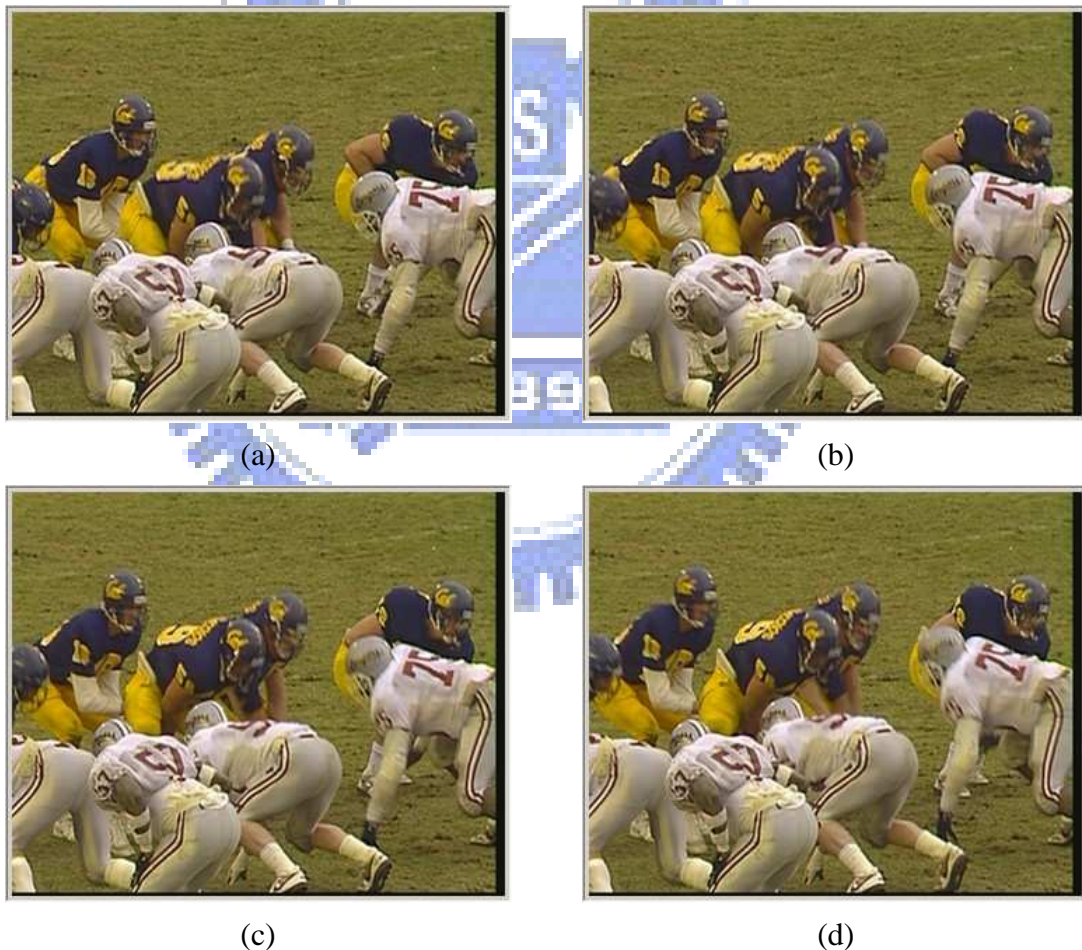


Figure 4.8 Six frames of the original video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame).



(e)



(f)

Figure 4.8 Six frames of the original video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame). (continued)



(a)



(b)



(c)



(d)

Figure 4.9 Six frames of the watermarked video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame).



(e)



(f)

Figure 4.9 Six frames of the watermarked video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame). (continued)



(a)



(b)



(c)



(d)

Figure 4.10 Six frames of the recovered video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame).



(e)

(f)

Figure 4.10 Six frames of the recovered video. (a) The first frame (I frame). (b) The second frame (P frame). (c) The third frame (P frame). (d) The 4th frame (P frame). (e) The 5th frame (P frame). (f) The 6th frame (P frame). (continued)



Figure 4.11 A watermarked frame played with wrong the secret key or the information of the local computer.

## 4.5 Discussions and Summary

In this chapter, we have proposed a scheme to protect the copyright of H.264/AVC videos by display control on specified computers through the information of the local computer. And we embed a visible watermark in a different way from the traditional method which not only claims the copyright but also prevents a video from being illegally distributed. The feasibility of the proposed method has been proved by our experimental results.



# Chapter 5

## Secret H.264-AVC Video Sharing with Steganographic Effects

### 5.1 Introduction

With the advance of computer and digital camera technologies, more and more videos are transformed into digital versions and transmitted on the Internet. Some of them are important secret or personal videos, such as video conferencing or surveillance videos of companies or governmental organizations. These videos should not be exposed to the public. So we need a method which is systematic and secure to keep them. The use of the secret sharing technique is a good solution. In this study, a technique of secret sharing with steganographic effects is proposed for H.264/AVC video sharing and is described in this chapter.

In Section 5.1.1, some relevant definitions are given, and in Section 5.1.2 the basic idea of the proposed scheme is presented. In Section 5.2, the proposed process of secret video sharing with steganographic effects is described, and the proposed corresponding process of recovery of secret videos is presented in Section 5.3. In Section 5.4, several experimental results of the proposed method will be shown. Finally, some discussions and a summary will be made in the last section of this chapter.

#### 5.1.1 Problem Definition

It often needs a systematic and secure method to protect secret videos, and a solution, as mentioned previously, is to use the secret sharing method to conduct systematic management of secret videos. For this aim, a problem is how to hide each meaningful part of a secret file in other cover media files, better with steganographic effects. And a corresponding problem is how to recover the secret video from the shares kept by the participants.

### 5.1.2 Proposed Ideas

Our idea is briefly described here. First, we extract prediction modes from given cover videos and the secret video. Then, we *share* the intra-prediction modes of the secret video based on the exclusive-OR (*XOR*) operation, and *hide* the resulting share data into the prediction modes of the cover videos, yielding some stego-videos. We allow the user to select a secret key to *randomize* of the content of the secret video before it is shared. Finally, the stego-videos and the randomized secret video are regarded as share videos and distributed to the participants for them to keep.

When recovery of the secret video is desired, after getting the secret key and collecting all the share videos from the participants, we can extract the hidden data from them. We then use the hidden data to recover the prediction modes, and construct accordingly the secret video finally. An illustration of the proposed idea is shown in Figure 5.1.

## 5.2 Proposed Scheme for Secret Video Sharing with Steganographic Effects

In this section, the proposed scheme of sharing the secret H.264/AVC video with steganographic effects will be described in detail. The process contains two parts:

creation of share data and creation of steganographic effects. In Section 5.2.1, the process for creating share data will be described. In Section 5.2.2, the process of creating steganographic effects is presented.

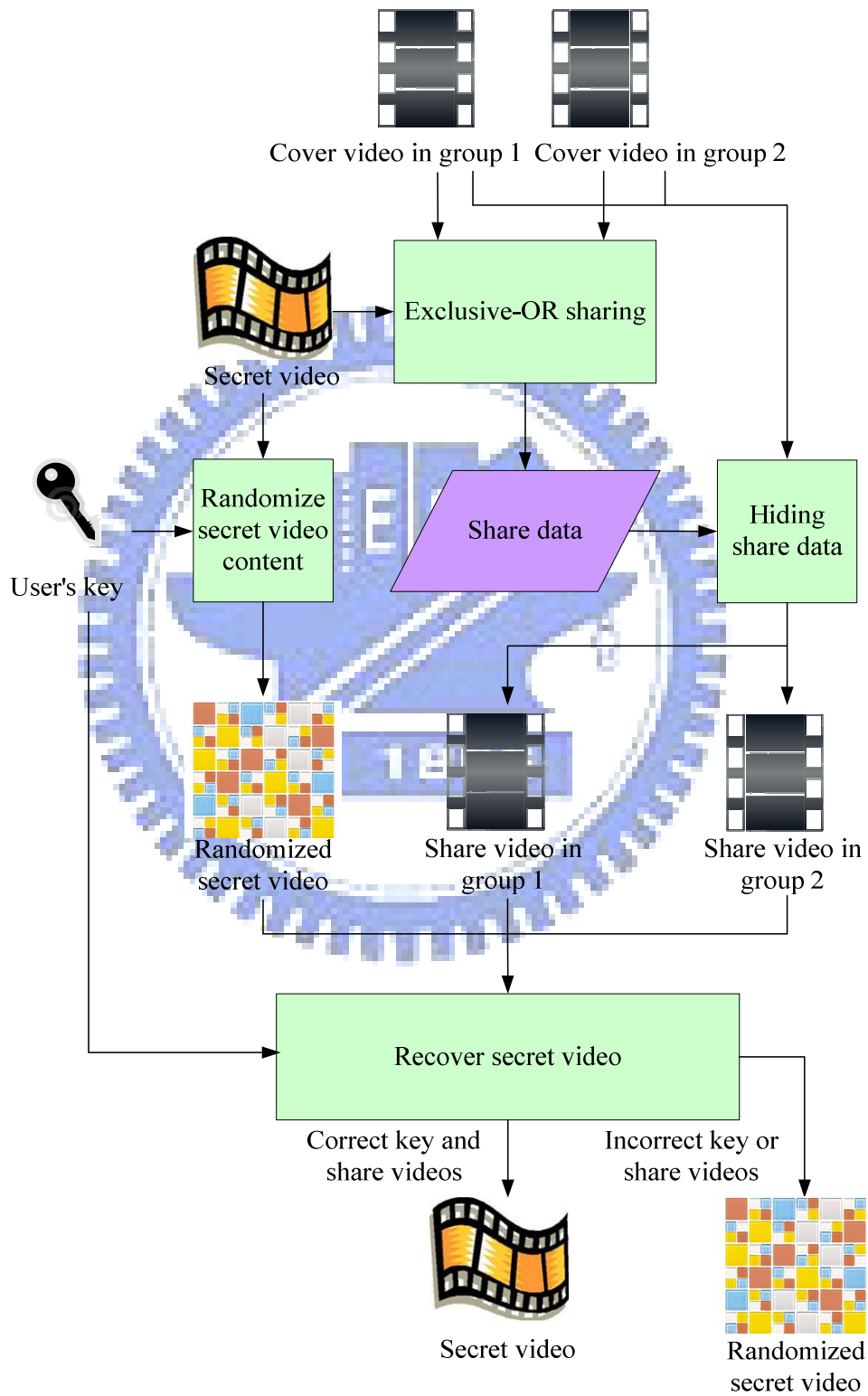


Figure 5.1 An illustration of the proposed idea.

## 5.2.1 Process for Share Data Creation

In the process of creating share data, we divide the cover videos into two groups  $G_1$  and  $G_2$ . We extract the prediction modes from the upper half parts of the frames of  $G_1$  and  $G_2$ , and extract the prediction modes from the entire frames of the secret video. To generate the share data, we apply the XOR operation to the prediction modes of the upper half part of the odd number frames of  $G_1$  and  $G_2$  as well as the prediction modes of the upper half part of each frame of the secret video. Symmetrically, we apply the XOR operation to the prediction modes of the upper half part of the even number frames of  $G_1$  and  $G_2$  as well as the prediction modes of the lower half part of each frame of the secret video. An illustration of the secret sharing method is shown in Figure 5.2. Finally, we use the prediction modes of the secret video and a user secret key to form a seed for a random number generator to generate random numbers. We subtract the random numbers from the quantized frequency coefficients of the frames of the secret video to randomize the content of the secret video. A detailed algorithm of the proposed secret video sharing method is described in the following.

**Algorithm 5.1:** the process of creating share data from the secret video.

**Input:** a secret video  $S$ , two cover video groups  $G_1 = \{V_{11}, V_{12}, \dots, V_{1n_1}\}$  and  $G_2 = \{V_{21}, V_{22}, \dots, V_{2n_2}\}$  including  $n_1$  and  $n_2$  videos, respectively, a secret key  $K$ , and a random number generator  $f$ .

**Output:** a randomized video  $S'$ , and a set of share data  $D$ .

**Steps:**

1. Extract the prediction modes  $M_{1i}$  and  $M_{2i}$  of the upper half part of each block of each frame of the  $n_1$  and  $n_2$  cover videos of  $G_1$  and  $G_2$ , respectively. Extract also the prediction modes  $M_s$  from each corresponding  $4 \times 4$  block of each frame of  $S$ . And then compute four bits  $D_{share} = d_0d_1d_2d_3$  of the share

data  $D$  by

$$D_{share} = M_{11} \oplus M_{12} \oplus \dots \oplus M_{1n_1} \oplus M_{21} \oplus M_{22} \oplus \dots \oplus M_{2n_2} \oplus M_s. \quad (5.1)$$

Note that for each luma block of  $G_1$ ,  $G_2$ , and  $S$ , four bits are generated as part of the share data  $D$ , whose embedding will be carried out by Algorithm 5.2 described next.

2. Combine  $M_s$  and the secret key  $K$  to form a seed for the random number generator  $f$  to generate three random numbers  $R_a$ ,  $R_c$ , and  $R_d$ .
3. Randomize the content of the secret video  $S$  by modifying each of the DC coefficient and the AC coefficients, denoted as  $C$ , of each block of the luma frame and the chroma frame of  $S$  by the following rules.

[1] *When  $C$  is the DC coefficient of the luma frame,*

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C + R_d; \\ &\text{if } C < -1, \text{ set } C = C - R_d. \end{aligned} \quad (5.2)$$

[2] *When  $C$  is the DC coefficient of the chroma frame,*

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C + R_c; \\ &\text{if } C < -1, \text{ set } C = C - R_c. \end{aligned} \quad (5.3)$$

[3] *When  $C$  is the AC coefficient of the luma frame,*

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C + R_a; \\ &\text{if } C < -1, \text{ set } C = C - R_a. \end{aligned} \quad (5.4)$$

[4] *When  $C$  is the AC coefficient of the chroma frame,*

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C + R_c; \\ &\text{if } C < -1, \text{ set } C = C - R_c. \end{aligned} \quad (5.5)$$

4. Set  $M_s$  as 0 to destroy the original value of  $M_s$  to enhance the security.

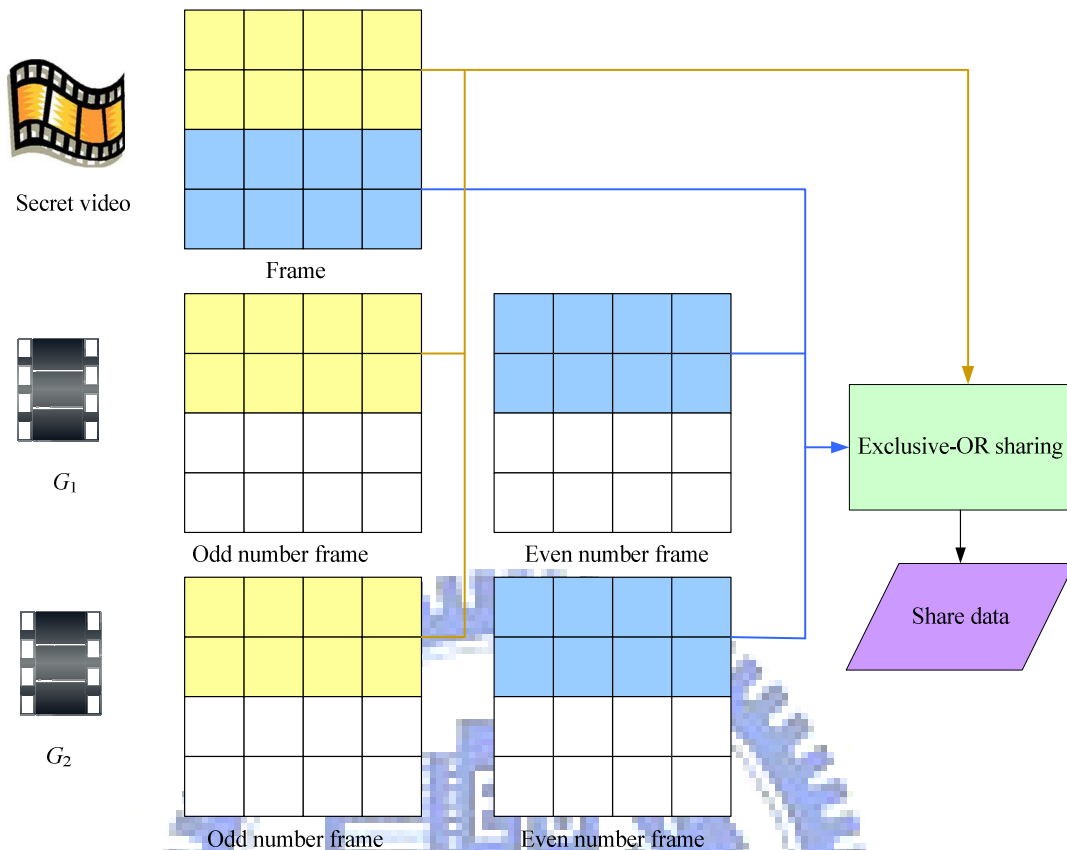


Figure 5.2 Illustration of the secret sharing method.

## 5.2.2 Process for Creating Steganographic Effects

The previous algorithm produces the share data which are yet to be embedded for later recovery. Embedding of the share data is again accomplished by the use of the prediction modes in the videos. It is desired also that after the data are embedded, the resulting stego-videos, called *share videos*, are still observable as normal videos. Due to this aim, it is undesirable for the data embedding process to cause the resulting prediction modes erroneous for video display. This *prediction error problem* might occur when the sample values in a luma block are computed in terms of the sample values of some preceding blocks which actually do not exist. A case of this occurs when the current block is the leftmost one in a frame so that its horizontally preceding block is nonexistent. An illustration of the prediction error problem is shown in Figure 5.3.

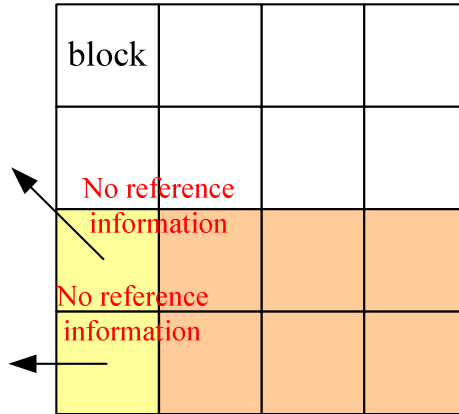


Figure 5.3 Illustration of the prediction error problem.

To avoid this problem, instead of skipping the leftmost blocks in the video, which is not applicable in our study here, we utilize only those prediction modes which do not refer to the sample values of its left neighboring block. Such prediction modes are  $M_0, M_2, M_3, M_7$ . There are totally four of them, which can be used to encode two bits of the share data. Note that we do not skip prediction modes which refer to the upper preceding block contents because, as will be clear in the sequel, we embed the share data into the lower half of each frame of the cover videos such that reference to upper neighboring blocks is no problem.

Recall that for each block, four bits of share data are to be embedded. But only two bits can be encoded using the four modes  $M_0, M_2, M_3$  and  $M_7$  in each block. So we divide the cover videos into two groups  $G_1 = \{V_{11}, V_{12}, \dots, V_{1n_1}\}$  and  $G_2 = \{V_{21}, V_{22}, \dots, V_{2n_2}\}$  including  $n_1$  and  $n_2$  videos, respectively, and embed the first two bits, denoted as *FTB*, of every 4-bit share data segment into  $G_1$  and the last two bits, denoted as *LTB*, of the segment into  $G_2$ , respectively. Furthermore, it is desired that the share data can be kept by all the cover videos of the group uniformly. Due to this aim, we embed the share data according to the following rules.

- (1) If  $n_1$  (or  $n_2$ ) is one, then we set *FTB* (or *LTB*) as the *data-to-be-hidden* in the corresponding block of the cover video  $V_{11}$  (or  $V_{21}$ ) directly.

(2) If  $n_1$  (or  $n_2$ ) is larger than one, then we set respectively a *random number*, which is from 0 to 3, as the data-to-be-hidden  $N_{1i}$  (or  $N_{2i}$ ) of the block of each of the  $n_1$  cover videos,  $V_{1i}$ , except the first one ( $V_{11}$  or  $V_{21}$ ); and use the XORing result of the *FTB* (or *LTB*) and all  $n_1 - 1$   $N_{1i}$ 's (or  $n_2 - 1$   $N_{2i}$ 's) as the data-to-be-hidden of the block of the first video  $V_{11}$  (or  $V_{21}$ ).

For example, suppose that  $G_2$  has three cover videos  $V_{21}, V_{22}, V_{23}$  and *LTB* is  $01_2$ , and suppose that we assign random number  $0 = 00_2$  as the data-to-be hidden  $N_{22}$  of  $V_{22}$  and assign random number  $2 = 10_2$  as the data-to-be-hidden  $N_{23}$  of  $V_{23}$ . After conducting the XOR operations  $LTB \oplus N_{22} \oplus N_{23} = 01_2 \oplus 00_2 \oplus 10_2 = 11_2$ , we get the data-to-be-hidden as  $11_2$  for the first cover video  $V_{21}$ .

Finally, we encode all the data-to-be-hidden by the corresponding prediction modes of the lower half part of each frame of the cover videos in  $G_1$  and  $G_2$  according to Table 5.1. A detailed algorithm of the steganographic effect creation process is described in the following.

Table 5.1 Relation between the data-to-be-hidden and the prediction mode

<i>data-to-be-hidden</i>	Prediction mode
00	0
01	7
10	2
11	3

**Algorithm 5.2:** the process of creating steganographic effects.

**Input:** two cover video groups  $G_1 = \{V_{11}, V_{12}, \dots, V_{1n_1}\}$  and  $G_2 = \{V_{21}, V_{22}, \dots, V_{2n_2}\}$  including  $n_1$  and  $n_2$  videos, respectively, and a set of share data,  $D$ .



**Output:** two share video groups  $G_1' = \{V_{11}', V_{12}', \dots, V_{1n_1}'\}$  and  $G_2' = \{V_{21}', V_{22}', \dots, V_{2n_2}'\}$ .

**Steps:** For each block of a frame of a video, perform the following steps.

1. Get in order four bits  $D_4$  of the share data  $D$  and denote the first two bits of  $D_4$  as  $FTB$  and the last two bits as  $LTB$ .
2. If  $n_1$  (or  $n_2$ ) is one, go to Step 3; if  $n_1$  (or  $n_2$ ) is larger than one, perform the following steps.

- 2.1 Set respectively a random number, which is from 0 to 3, as the data-to-be-hidden  $N_{1i}$  (or  $N_{2i}$ ) to the corresponding block of each of the  $n_1$  (or  $n_2$ ) cover videos  $V_{12}, V_{13}, \dots, V_{1n_1}$  (or  $V_{22}, V_{23}, \dots, V_{2n_2}$ ) except the first one  $V_{11}$  ( $V_{21}$ ).

- 2.2 Compute the data  $FHD$  by

$$FHD = FTB \oplus N_{12} \oplus N_{13} \oplus \dots \oplus N_{1n_1}.$$

$$(\text{or } FHD = LTB \oplus N_{22} \oplus N_{23} \oplus \dots \oplus N_{2n_2}). \quad (5.6)$$

- 2.3 Set  $FHD$  as the data-to-be-hidden  $N_{11}$  (or  $N_{21}$ ) of the block of  $V_{11}$  ( $V_{21}$ ).

3. Create a new data set  $E$  consisting in order of  $FTB$ ,  $LTB$ ,  $FHD$ , and  $N_{ij}$ 's in binary form, and encode every two bits  $S$  of  $E$  by the prediction modes of the corresponding block of the lower half part of each frame of  $G_1$  (or  $G_2$ ) according to the following rules:

$$\begin{aligned} &\text{if } S = 00, \text{ then encode } S \text{ by prediction mode} = 0; \\ &\text{if } S = 01, \text{ then encode } S \text{ by prediction mode} = 7; \\ &\text{if } S = 10, \text{ then encode } S \text{ by prediction mode} = 2; \\ &\text{if } S = 11, \text{ then encode } S \text{ by prediction mode} = 3. \end{aligned} \quad (5.7)$$

A flowchart of the steganography process is shown in Figure 5.4.

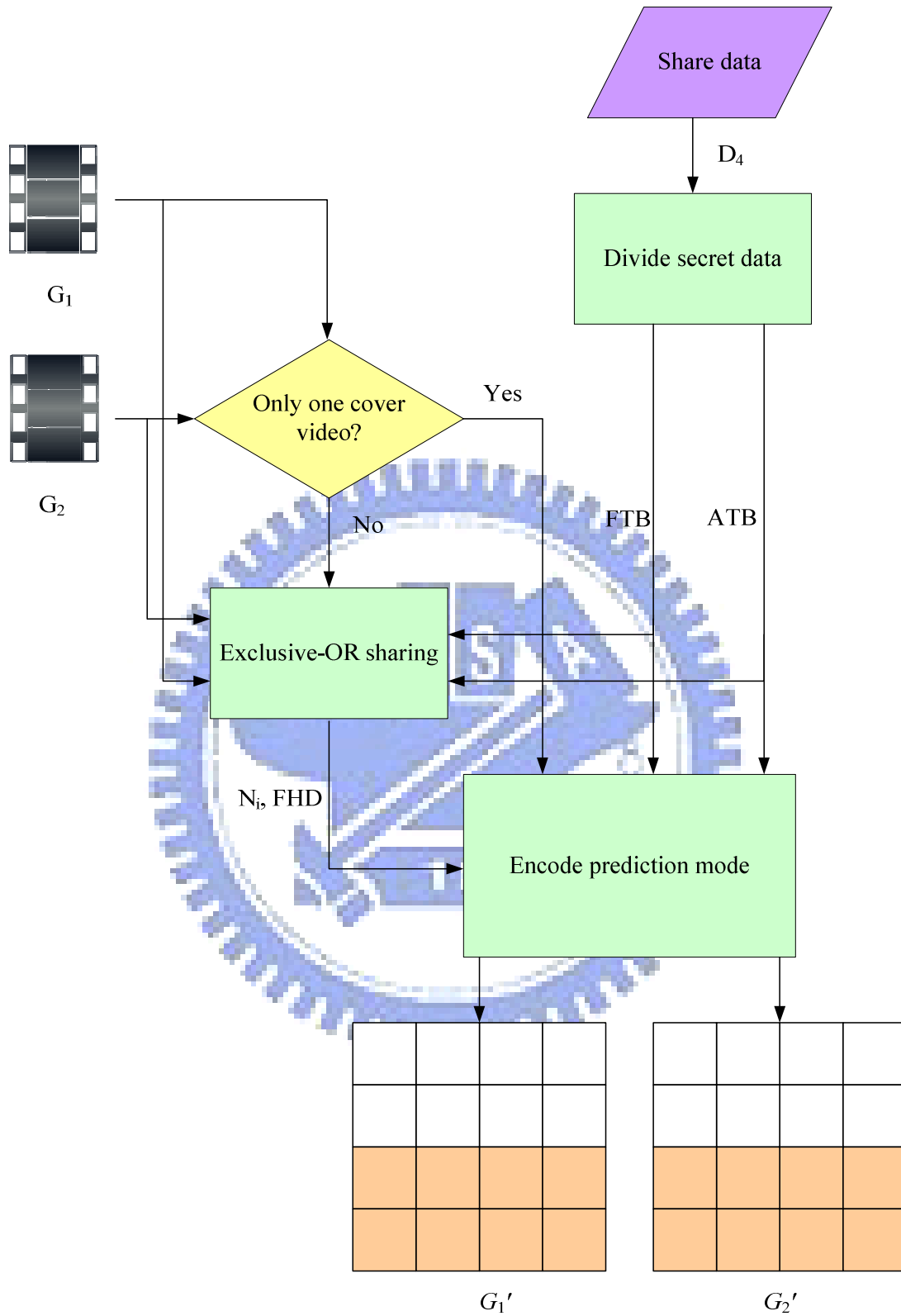


Figure 5.4 Flowchart of the process of creating steganographic effects.

## 5.3 Recovery of Secret Videos

In this section, the process of recovery of the secret video will be described. The success of the recovery process depends on whether we can get all the share videos and the right secret key. The tasks of the process are to extract the share data from the stego-videos, then to recover the prediction modes of the secret video based on the XOR operation, and finally to use the secret key and the prediction modes to recover the secret video. The detailed algorithm is described in the following.

**Algorithm 5.3:** the process of recovery of the secret video.

**Input:** two share video groups  $G_1' = \{V_{11}', V_{12}', \dots, V_{1n_1}'\}$  and  $G_2' = \{V_{21}', V_{22}', \dots, V_{2n_2}'\}$  including  $n_1$  and  $n_2$  videos, respectively, a randomized video  $S'$ , a secret key  $K$ , and a random number generator  $f$  as that used in algorithm 5.1.

**Output:** a secret video  $S$ .

**Steps:** For each block of a frame of a video, perform the following steps.

1. Extract every two bits of the hidden data  $N_{1i}$  and  $N_{2i}$  by the prediction modes  $M$  of the corresponding block of the lower half part of each frame of the  $n_1$  and  $n_2$  cover videos of  $G_1'$  and  $G_2'$  according to the following rules:

$$\begin{aligned}
 &\text{if } M = 0, \text{ then set } N_{1i}(N_{2i}) = 00; \\
 &\text{if } M = 7, \text{ then set } N_{1i}(N_{2i}) = 01; \\
 &\text{if } M = 2, \text{ then set } N_{1i}(N_{2i}) = 10; \\
 &\text{if } M = 3, \text{ then set } N_{1i}(N_{2i}) = 11.
 \end{aligned} \tag{5.8}$$

2. If  $n_1$  (or  $n_2$ ) is one, set the first two bits of every four bits  $D_4$  of the share data  $D$  as  $N_{1i}$  and the last two bits of  $D_4$  as  $N_{2i}$ . Go to Step 4.
3. If  $n_1$  (or  $n_2$ ) is larger than one, we perform the following steps.
  - 3.1 Compute the hidden data  $N_{1f}$  (or  $N_{2f}$ ) of the corresponding block of each video of the group  $G_1'$  (or  $G_2'$ ) by

$$N_{1f} = N_{11} \oplus N_{12} \oplus \dots \oplus N_{1n_1}$$

$$(\text{or } N_{2f} = N_{21} \oplus N_{22} \oplus \dots \oplus N_{2n_2}). \quad (5.9)$$

3.2 Set the first two bits of every four bits  $D_4$  of the share data  $D$  as  $N_{1f}$  and the last two bits of  $D_4$  as  $N_{2f}$ .

4. Extract the prediction modes  $M_{1i}$  and  $M_{2i}$  of each block of the upper half part each frame of the  $n_1$  and  $n_2$  cover videos of  $G_1$  and  $G_2$ . Then set the prediction mode  $P$  of the corresponding block of the secret video by

$$P = M_{11} \oplus M_{12} \oplus \dots \oplus M_{1n_1} \oplus M_{21} \oplus M_{22} \oplus \dots \oplus M_{2n_2} \oplus D_4. \quad (5.10)$$

5. Combine  $P$  and the secret key  $K$  to form a seed for the random number generator  $f$  to generate three random numbers  $R_a$ ,  $R_c$ , and  $R_d$ .
6. Recover of the secret video by modifying each of the DC coefficient and the AC coefficients, denoted as  $C$ , of the luma frame and the chroma frame of  $S'$  to by performing following rules.

[1] When  $C$  is the DC coefficient of the luma frame,

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C - R_d; \\ &\text{if } C < -1, \text{ set } C = C + R_d. \end{aligned} \quad (5.11)$$

[2] When  $C$  is the DC coefficient of the chroma frame,

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C - R_c; \\ &\text{if } C < -1, \text{ set } C = C + R_c. \end{aligned} \quad (5.12)$$

[3] When  $C$  is the AC coefficient of the luma frame,

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C - R_a; \\ &\text{if } C < -1, \text{ set } C = C + R_a. \end{aligned} \quad (5.13)$$

[4] When  $C$  is the AC coefficient of the chroma frame,

$$\begin{aligned} &\text{if } C > 1, \text{ set } C = C - R_c; \\ &\text{if } C < -1, \text{ set } C = C + R_c. \end{aligned} \quad (5.14)$$

A flowchart of the recovery process for the secret video is shown in Figure 5.5

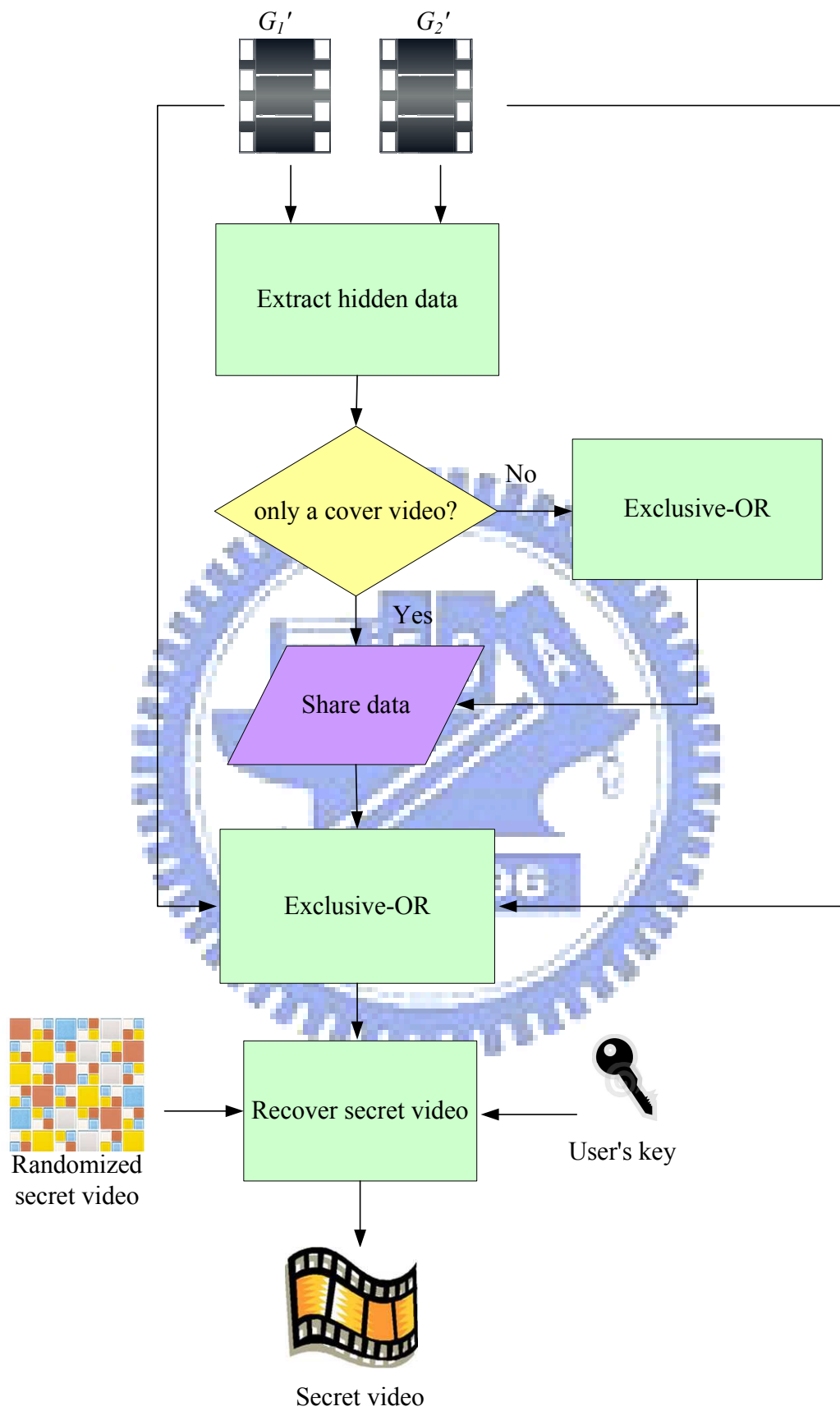


Figure 5.5 Flowchart of the recovering process.

## 5.4 Experimental Results

The proposed video sharing algorithm has been integrated into the H.264 reference software JM12.4 in our experiments. An H.264/AVC video in CIF (352×288 pixels) format was used in our experiments.

Four frames of the secret video and the resulting randomized video are shown in Figure 5.6. Four frames of two cover videos and the resulting share videos are shown in Figure 5.7 and Figure 5.8, respectively. And corresponding four frames of the recovered secret video are shown in Figure 5.9. If the secret key or share videos are wrong, we cannot recover the secret video. A result of such cases is shown in Figure 5.10.

## 5.5 Discussions and Summary

In this chapter, we have proposed a scheme to protect secret H.264/AVC videos systematically and securely by using secret sharing and steganography techniques. By this scheme, we can transform secret data into multiple share videos to be kept by participants of the secret sharing activity. Therefore, the method is suitable for use for multiple people or an organization to manage the secret video. The feasibility of the proposed method has been proved by our experimental results.

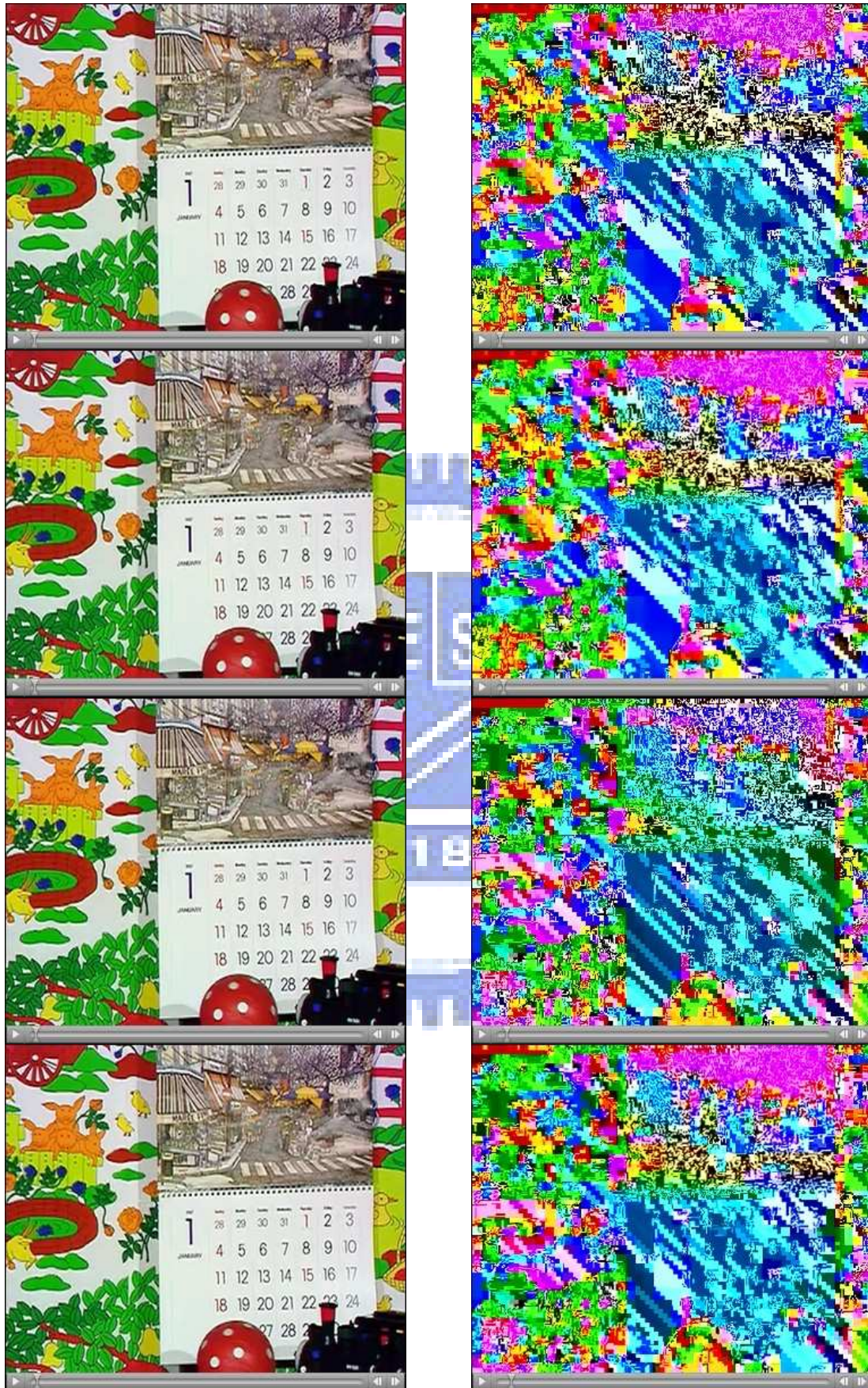


Figure 5.6 The four frames of the original video (left) and randomized video (right).



Figure 5.7 The four frames of the first cover video (left) and the share video (right).



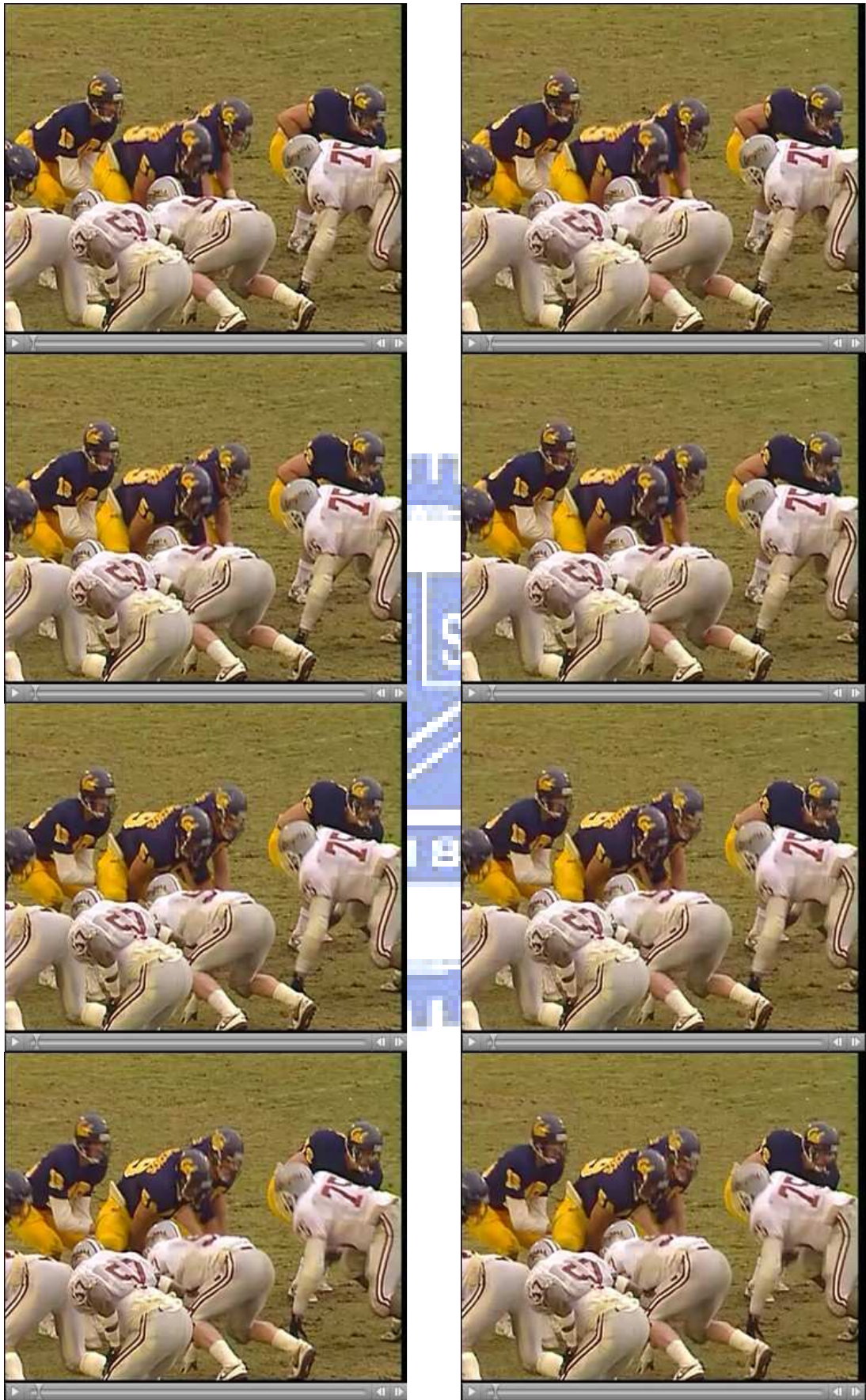


Figure 5.8 The four frames of the second cover video (left) and the share video (right).



(1)



(2)



(3)



(4)

Figure 5.9 The four frames of the recovered secret video.

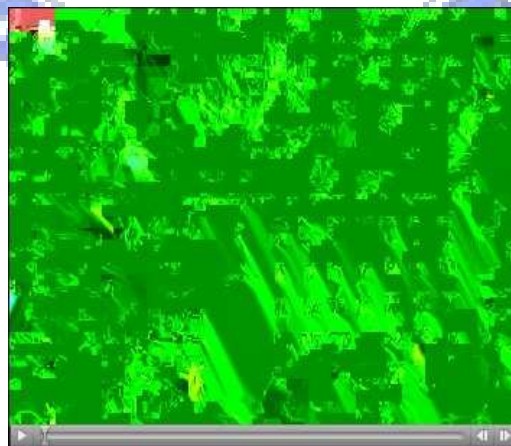


Figure 5.10 Erroneous recovery result when the secret key is wrong.

# Chapter 6

## Conclusions and Suggestions for Future Works

### 6.1 Conclusions

In this study, we have proposed several methods for a variety of data hiding application purposes, such as covert communication, copyright protection, and video sharing using H.264/AVC videos as cover data.

For covert communication, a large-volume data hiding method and two optimal data hiding methods based on some properties of H.264/AVC have been proposed. For I macroblocks, data are hidden based on the use of the intra-prediction modes. For P macroblocks, data are hidden based on the use of the tree structured motion compensation. Both the intra-prediction mode and the tree structured motion compensation are used properly to hide data in order to maintain the imperceptibility of the hidden data. Since not only I macroblocks but also P macroblocks are used to hide data in the proposed method, the data hiding capacity is increased substantially.

For copyright protection, a removable visible watermarking method with a scheme for video display control on specified computers has been proposed. Multimedia providers can protect, with the proposed method, their H.264/AVC videos downloaded to the client site by display control on the specified computer. When users sent the video to others who are not unauthorized, a visible watermark will appear actively to claim the ownership of this video by the video provider.

For video sharing, a method of video sharing with steganographic effects has been proposed for protecting secret videos systematically and securely. A secret video

is transformed into multiple share videos by distributing the prediction mode information of the secret video into the shares. Each share video contains part of the meaningful content of the secret video imperceptibly and is then kept by a participant of the secret sharing activity. By collecting all share videos, the content of the original secret video can be recovered.

Experimental results show the feasibility and practicality of the proposed methods for covert communication, copyright protection, and secret sharing.

## 6.2 Suggestions for Future Works

Several suggestions for future research works are listed as follows.

1. The proposed ideas in this study may be extended to handle digital videos of other profiles of the H.264/AVC standard.
2. It is interesting to add the user's information into the visible watermarking method proposed in this study.
3. The visible watermarking method proposed in this study may be extended to create semi-transparent effects.
4. It is interesting to extend the data hiding method proposed in this study to handle video authentication problems.
5. The video sharing method proposed in this study may be integrated with a video authentication method to provide the ability of verifying the integrity and fidelity of the share videos.
6. It is also interesting to extend the proposed ideas in this study to handle problems using audio data as the cover files. The protection of secret audio data is also an important topic.

# References

- [1] M. Yang and N. Bourbakis, "A High Bitrate Information Hiding Algorithm for Digital Video Content under H.264/AVC Compression," *Proceedings of IEEE International Conference on Image Processing Midwest Symposium on Circuits and Systems*, Cincinnati, OH, USA, vol. 2, pp. 935- 938, Aug., 2005.
- [2] Y. Hu, et al., "Information hiding based on intra prediction modes for H.264/AVC," *Proceedings of IEEE International Conference on Multimedia and Expo*, Beijing, China, pp. 1231-1234, Jul., 2007.
- [3] S. K. Kapotas, et al., "Data hiding in H.264 encoded video sequences," *Proceedings of International Workshop on Multimedia Signal Processing*, Chania, Crete, Greece, pp. 373-376, Oct., 2007.
- [4] J. Meng and S. F. Chang, "Embedding visible video watermarks in the compressed domain," *Proceedings of IEEE International Conference on Image Processing*, Chicago, IL, USA, vol. 1, pp. 474-477, Oct. 1998.
- [5] S. Bhattacharya et al. "A survey on different video watermarking techniques and comparative analysis with reference to H.264/AVC," *Proceedings of 2006 IEEE Tenth International Symposium on Consumer Electronics (ISCE 2006)*, St. Petersburg, Russia, June-July, 2006, pp. 1-6, July, 2001.
- [6] S. P. Mohanty et al., "A DCT domain visible watermarking technique for images," *Proceedings of IEEE International Conference on Multimedia and Expo*, New York, NY, USA, vol. 2, pp. 1029-1032, Aug. 2000.
- [7] K. F. Chien and W. H. Tsai, "A study on information hiding in MPEG4 videos and applications to copyright protection and security surveillance," *Master Thesis*, Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, June, 2006.

- [8] A. Shamir, "How to share a secret," *Communications of Association for Computing Machinery*, vol. 22, no. 11, pp. 612- 613, Nov., 1979.
- [9] C. C. Lin and W. H. Tsai, "Secret multimedia information sharing with data hiding capacity by simple logic operations," *Proceedings of 5th World Multiconference on Systemics, Cybernetics, and Informatics, Vol. I: Information Systems Development*, Orlando, Florida, U. S. A., pp. 50-55, Jul., 2001.
- [10] X. Zou and S. Sun, "Information hiding using secret sharing scheme," *Proceedings of International Conference on Innovative Computing, Information and Control*, Beijing, China, vol. 1, pp. 484-487, Aug., 2006.
- [11] I. Richardson, "H.264 and MPEG-4 video compression video coding for next-generation multimedia," John Wiley & Sons, Hoboken, USA, 2003.
- [12] T. Wiegand et al., "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, pp. 688- 703, July, 2003.
- [13] H.264/AVC JVT reference software JM12.4,  
<http://iphome.hhi.de/suehring/tml/download/>