VIDEO AUTHENTICATION IN HEVC COMPRESSED DOMAIN

TEW YIQI

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2016

VIDEO AUTHENTICATION IN HEVC COMPRESSED DOMAIN

TEW YIQI

THESIS SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2016

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: ***TEW YIQI***

Registration/Matrix No.: ***WHA 120010***

Name of Degree: ***DOCTOR OF PHILOSOPHY***

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

VIDEO AUTHENTICATION IN HEVC COMPRESSED DOMAIN

Field of Study: ***INFORMATION HIDING***

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date

Subscribed and solemnly declared before,

Witness's Signature Date

Name:
Designation:

ABSTRACT

High Efficiency Video Coding (HEVC) is the latest video compression standard finalized in year 2013. While H.264/Advance Video Coding (AVC) is still the mostly deployed video-coding standard, HEVC is gaining ground, especially for storage and transmission of high-resolution videos such as High Definition (HD), 4K, 8K and beyond. In this thesis, video authentication based on information hiding technique is studied. The concept of authentication, layout and implementation are presented under the latest HEVC video compression standard. One of the unique properties of HEVC standard, i.e., combination of coding unit size, which is sensitive to video manipulation, is utilized in the proposed information hiding technique. A video authentication scheme is then put forward by exploiting this unique property of HEVC to embed authentication code based on a predefined mapping rule. In addition, temporal dependency is enforced, where the authentication tag generated in one video slice is embedded into its subsequent slice. Furthermore, multiple layers of authentication are presented to detect and localize the tampered regions in a HEVC video, as well as verifying the source / sender of the video using a shared secret key. Moreover, several encryption techniques are presented to incorporate with the proposed scheme to achieve video authentication in encrypted domain without compromising on compression efficiency. Video sequences from various classes (i.e., resolutions) are considered to verify the performance of the proposed multi-layer authentication scheme. Results show that, at the expense of slight degradation in perceptual quality, the proposed scheme is robust against video tampering within and across video slices. Lastly, a functional comparison is performed between the proposed authentication scheme and the conventional schemes for both plaintext and ciphertext (encrypted) videos.

Abstrak

HEVC (Piawaian Pengekodan Video Bercekapan Tinggi) adalah piawaian pemampatan video terkini yang dimuktamadkan pada tahun 2013. Walaupun H.264 / AVC masih merupakan pemampatan video yang paling kerap digunakan sebagai piawaian pengekodan video, penggunaan HEVC tetap semakin meningkat, terutamanya dalam penyimpanan dan penghantaran video beresolusi tinggi seperti HD, 4K dan 8K. Dalam tesis ini, pengesahan video berdasarkan teknik penyembunyian maklumat diselidik dengan teliti. Konsep, susun atur dan pelaksanaan pengesahan dihuraikan di bawah HEVC. Salah satu daripada sifat-sifat unik HEVC, iaitu, gabungan saiz unit pengekodan, yang sensitif kepada manipulasi video, digunakan dalam teknik menyembunyikan maklumat yang dicadangkan. Satu skim pengesahan video telah dikemukakan dengan mengeksploitasi sifat unik HEVC ini untuk menyembunyi kod pengesahan berdasarkan peraturan pemetaan yang telah ditetapkan. Di samping itu, dengan pelaksanaan pengantungan temporal, tag pengesahan yang dihasilkan dalam satu keping video akan dibenam di kepingan video berikutnya. Tambahan pula, pelbagai lapisan pengesahan ditawarkan dalam skim yang dicadangkan untuk menyetempatkan bahagian video HEVC yang dimanipulasi, serta mengesahkan sumber / penghantar video dengan menggunakan kekunci rahsia. Selain itu, beberapa teknik penyulitan yang dibentangkan beroperasi dengan skim yang dicadangkan untuk mencapai pengesahan video dalam domain penyulitan tanpa menjejaskan kecekapan mampatan video. Urutan video dari pelbagai kelas (iaitu resolusi) digunakan untuk mengesahkan prestasi skim pengesahan. Keputusan menunjukkan bahawa skim pengesahan yang dicadangkan adalah teguh terhadap kepingan video yang dimanipulasikan, dengan sedikit degradasi dalam kualiti persepsi video. Akhir sekali, perbandingan fungsi antara skim pengesahan yang dicadangkan dan skim konvensional juga dibincangkan.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Dr. KokSheik Wong, for the generous support he provided to me from the first moment I started this research work. I highly appreciate his valuable and thorough comments and guidance, and in many ways, I have learnt much from him. I enjoyed the constructive discussions with him throughout the preparation of this study, which enriched my spirit of patience, discipline and hard-working.

Meanwhile, I would like to thank to all my research colleagues in Multimedia Signal Processing and Information Hiding (MSPIH) research group for the spiritual supports during my research period. Positive, constructive and valuable comments are provided by them in order to improve my work and guide me to the right track whenever I'm confused in my research work.

Not to forget my financial support, MyBrain program. It is a program introduced by Ministry of Education under the 10^{th} Malaysia Plan to produce a target of 20,000 PhD graduates in year 2020. I'm glad, and feel honored to be a part of this plan and I shall contribute to the country after the graduation. Not to forget another supporting scheme from University of Malaya, i.e., SBUM, who has continuously provided educational fund and allowances throughout my research period. I'm pleased and honored for getting these sponsorships to support my PhD study.

Last but not least, thousands of thanks to my family, my parents and my beloved wife, Suh Yenn, for being company with me all the time, support me to continue my research study for my self enrichment.

Thanks for being a part of my life to fulfill my achievement as a PhD graduate.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

F	:	Frequency of occurrences.
K	:	Secret key.
M	:	Total number of CTU in one video slice.
$P(i,j)$	:	Detected binary pixel in original video slice.
S_{n+1}	:	Next video slice.
S_{n-1}	:	Previous video slice.
S_n	:	Current video slice.
Δ	:	Difference between two highest count of δ.
Γ	:	Difference between two highest count of γ.
Π	:	Difference between CU count using intra or inter.
$\Re$	:	Edge Differential Ratio.
α_1	:	Bit(s) at location 1 in a tag.
α_2	:	Bit(s) at location 2 in a tag.
α_3	:	Bit(s) at location 3 in a tag.
α_4	:	Bit(s) at location 4 in a tag.
$\bar{P}(i,j)$	:	Detected binary pixel in encrypted video slice.
δ	:	Delta value for CU.
γ_m	:	Size of the CU in m-th CTU.
κ	:	A key with value $\in [0, 2^{32}-1]$.
$\mathbb{B}$	:	Number of embedded bits in a CU.
$\mathbb{B}_{10}$	:	Decimal representation of $\mathbb{B}$ bits.
$\mathbb{D}$	:	Threshold value.
$\mathbb{H}_i$	:	Addition value of i-th CU.
$\mathbb{I}_i$	:	Predictor index of i-th CU.
$\mathbb{L}$	:	Position of the last non-zero coefficient.
$\mathbb{M}$	:	Search points position, $\mathbb{M} \in \{0,1,3,6,8\}$.
$\mathbb{N}$	:	Search points position, $\mathbb{N} \in \{2,4,5,7\}$.
$\mathbb{S}_i$	:	Sum of DCT coefficient of i-th CU.
$\mathbb{V}$	:	New non-zero coefficient.
π_m	:	Type of the prediction mode in intra or inter in m-th CTU.
ρ	:	Type of prediction mode.
$\widetilde{Cb}$	:	Reference Cb.
$\widetilde{Cr}$	:	Reference Cr.
$\widetilde{T1s}$	:	Modified T1s codeword.
c_j	:	Coefficient at location j.
cnz	:	Count of non-zero DCT coefficients.
m	:	index of CTU in one video slice.

s	:	Difference between count of positive and negetive signs.
sav	:	Sum of absolute value of non-zero DCT coefficients.
v_1	:	First layer authentication status.
v_2	:	Second layer authentication status.
v_3	:	Third layer authentication status.
w	:	information bit.
AC	:	Alternating Current.
ACPO	:	Association of Chief Police Officers.
AES	:	Advanced Encryption System.
AI	:	All Intra.
AMP	:	Asymmetry Motion Partition.
AVC	:	Advance Video Coding.
B-	:	Bidirectional predicted-.
bps	:	bit per second.
CABAC	:	Context-Adaptive Binary Arithmetic Coding.
CAN	:	Canny Outline Detection.
CAVLC	:	Context-Adaptive Variable Length Coding.
Cb	:	Blue-difference Chrominance Component.
CCTV	:	closed-circuit television.
CD	:	Compact Disc.
Cr	:	Red-difference Chrominance Component.
CTU	:	Coding Tree Unit.
CU	:	Coding Unit.
DC	:	Direct Current.
DCT	:	Discrete Cosine Transformation.
DES	:	Data Encryption Standard.
dQP	:	Delta Quantization Parameter.
DVD	:	Digital Video Disc.
DVR	:	Digital Video Recorder.
DWT	:	Discrete Wavelet Transformation.
FMO	:	Flexible Macroblock Ordering.
fps	:	frames per second.
GOP	:	Group of Pictures.
HD	:	High Definition.
HDTV	:	High Definition Television.
HEVC	:	High Efficiency Video Coding.
I-	:	Intra-.
I4MB	:	4×4 in Intra Prediction Mode.
IEC	:	International Electro-technical Commission.
ISO	:	International Standard Organization.

ITU-T : International Telecommunication Union - Telecommunication Standardization Sector.
JND : Just Noticeable Difference.
JVT : Joint Video Team.
k : kilo.
LB : Low Delay B.
LP : Low Delay P.
LSB : Least Significant Bit.
M : Mega.
MD5 : Message-Digest Algorithm 5.
MPEG : Motion Picture Expert Group.
MPEG-1 : Motion Picture Expert Group Phase 1.
MPEG-2 : Motion Picture Expert Group Phase 2.
MPEG-3 : Motion Picture Expert Group Phase 3.
MPEG-4 : Motion Picture Expert Group Phase 4.
MV : Motion Vector.
MVComp : Motion Vector Competition Index.
MVD : Motion Vector Displacement.
NAL : Network Abstraction Layer.
P- : Predicted-.
PSNR : Peak Signal-to-Noise Ratio.
PU : Prediction Unit.
QP : Quantization Parameter.
RA : Random Access.
RDO : Rate Distortion Optimizer.
ROI : Region of Interest.
SHA : Secure Hash Algorithm.
SKID : Secret Key IDentification.
SOB : Sobel Outline Detection.
SSIM : Structural SIMilarity index.
SVC : Scalable Video Coding.
T1s : Trailing Ones.
TU : Transform Unit.
TV : Television.
UHD : Ultra High Definition.
VCEG : Video Coding Expert Group.
VLC : Variable Length Coding.
VQ : Vector Quantization.

LIST OF APPENDICES

CHAPTER 1 : THESIS OVERVIEW

An overview of the thesis is presented in this chapter. It includes the research problem statements, objectives, scopes, limitations and contribution, under the general topic of video authentication. Then, the thesis organization is briefly delineated for the beneficial of reader to understand the presentation flow.

1.1 Overview of Video Authentication

Digital video has become an important part of the modern daily life thanks to the widely accepted standardization of video coding formats and their successful deployments in various applications. People watch movies over the Internet, record video using car Digital Video Recorder (DVR), establish video conference across heterogeneous network environments, etc. However, these videos can be easily manipulated (e.g., trimmed, cropped, re-compressed) due to the availability of high performance personal computer at affordable prices and user-friendly yet powerful video editing software (Waddilove, 2015). As a result, the integrity of digital video and its origin become implausible. Hence, a video needs to be authenticated so that its source can be confirmed to be someone trustworthy and its content can be verified to be genuine prior to consumption or broadcasting (Atrey et al., 2009).

Unlike its success in providing entertainment (Maillard, 2009), the viability of digital video as evidence in the judicial process has been largely unprecedented. And yet there is an increasing number of videos from personal cameras or mobile phones being released in social media corresponding to incidents, e.g., road bullying. Digital evidence is often ruled inadmissible by courts because it is owned without authentication or its authenticity cannot be verified (Casey, 2011). According to the guideline stipulated in (Williams, 2012), it is necessary to demonstrate how evidence is authenticated and to show the in-

tegrity of each process through which the evidence was obtained. The evidence should be preserved from any third party who is able to repeat the same process and attain the same result as that presented to the court. Therefore, implementing a secure authentication scheme to confirm the authenticity of viable video evidence is imperative. It is also important to prevent any digital video tailored for causing hatred or benefiting a certain party.

1.2 Problem Statements

Currently, the recent advanced video coding technology serves minimum focus on video content protection, particularly to identify the genuineness of video content through authentication process. In general, several issues have been brought to the researchers' attention:

1. Recent advanced video editing software allows end-users to arbitrarily manipulate any video content while appearing innocuous without being noticed.

2. Existing security designs based on the previous video coding standards (e.g., H.264) may not be applicable to the recently released HEVC standard.

3. Existing video encoders require simple and fast authentication process to serve high-resolution video (e.g., 4K and 8K Ultra High Definition (UHD)).

1.3 Research Objectives

Based on the aforementioned problem statements, several objectives are prescribed as follows:

1. Seek and enable security applications based on information hiding in the current state-of-the-art video compression standards.

2. Evaluate the performance of various information hiding techniques in protecting

video integrity, particularly for authentication purpose.

3. Give recommendation on the design of video authentication based on information hiding technique.

4. Propose a new video authentication scheme in encrypted domain for HEVC compressed video.

1.4 Research Scopes and Limitations

There are several scopes and restrictions needed to be highlighted throughout the research duration, in order to conduct research efficiently and achieve research objectives. The scopes and restrictions are prescribed as follow:

1. To explore the video coding structure of the latest standard, i.e., HEVC with editable reference source codes (i.e., HM10.0) under C++ programming language. This requires good programming skills for understanding the video coding structure and manipulation on the original reference source codes.

2. Six classes of video resolution, i.e., Class A (2560×1600), Class B (1920×1080), Class C (832×480), Class D (416×240), Class E (1280×720) and Class F (1024×768) are considered for common encoding/decoding references, which requires large memory space to store and evaluate the processed videos.

3. Long computational time for HEVC video encoding process, i.e., Class A with 500 Mega (M)bit per second (bps) takes 12 hours to encode a video of 10 seconds at 30 frames of second (fps), i.e., 300 frames all together. This requires powerful machine to process the entire video sequences in various classes within a reasonable period of time.

1.5 Research Contributions

This research contributes in the following manners:

1. Advances the research in video authentication based on information hiding technique for achieving higher video quality and capacity while suppressing complexity.

2. Realizes invented video authentication with features for detecting forged video content and identifying the genuineness of the video.

3. Enables encryption based on invented video authentication scheme specifically in the field of video coding.

1.6 Thesis Organization

This thesis is compiled in seven chapters, namely Thesis Overview, Literature Study, Information Hiding Technique, Video Authentication Scheme, Video Encryption Scheme, Joint Authentication & Encryption Scheme, and Conclusion & Future Work of the research. After the thesis overview, Chapter 2 surveys the literature for four general scopes: Video Coding Standard, Information Hiding, Video Authentication and Video Encryption. Next, an information hiding technique for HEVC video is proposed in Chapter 3, followed by the proposal of a video authentication scheme in Chapter 4. Then, a video encryption scheme is put forward to form a joint video encryption and authentication as detailed in Chapter 5. Discussions on the proposed joint scheme are presented in Chapter 6 and finally conclusions and future work are presented in Chapter 7.

CHAPTER 2 : LITERATURE STUDY

2.1 Overview

In this chapter, a general overview and evolution of video compression standards are presented. It includes Motion Picture Expert Group Phase 1 (MPEG-1), Motion Picture Expert Group Phase 2 (MPEG-2), Motion Picture Expert Group Phase 4 (MPEG-4), i.e, H.264/AVC, and the recently released HEVC standards. Then, several information hiding techniques by utilizing the video structure are described. Next, a general overview of labeling and watermarking based video authentication are presented, followed by an overview of naïve and selective video encryption scheme. Lastly, a problem analysis on literature study in this research is discussed.

2.2 Introduction

In this study, a video authentication scheme based on information hiding technique is sought for under the recent HEVC video compression standard. Then, video encryption scheme is studied with the intention of forming a joint video encryption and authentication scheme (i.e., Objective 4).

2.3 Overview of Video Compression Standard

Motion picture, widely known as video, has become one of the most influential media in the entertainment industry. A working group of authorities, Motion Picture Expert Group (MPEG), was formed by International Standard Organization (ISO) and International Electro-technical Commission (IEC) in 1989 to establish the video compression standards (e.g., MPEG-1). These standards are published through ISO/IEC and recommended by International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) as H.26X (e.g., MPEG-1 as H.261).

2.3.1 MPEG-1 (H.261) and MPEG-2 (H.262)

The first MPEG compression standard (i.e., MPEG-1) was introduced in 1993. It was basically designed to enable moving pictures and sound to be encoded at the bitrate of a Compact Disc (CD) (ISO, 1993), i.e., 1.5 M bps. It was used in Video CD, cable Television (TV) services before MPEG-2 standard became widespread. In 1995, MPEG-2 standard was introduced. It supports interlacing, high definition and enables the Digital Video Disc (DVD) and digital satellite television technologies (ISO, 2000). Motion Picture Expert Group Phase 3 (MPEG-3) was intended for High Definition Television (HDTV) compression but was found to be redundant and merged with MPEG-2.

2.3.2 MPEG-4 (H.264 / AVC)

In the pursuit of higher efficiency in video coding, the Joint Video Team (JVT) is formed by Video Coding Expert Group (VCEG) and MPEG to propose MPEG-4 standard and it has become one of the most commonly practiced video compression formats since 2003. The design of MPEG-4 standard provides an enhanced compression performance on video representation and achieves a significant improvement in rate distortion trade-off by offering high video quality for relatively low bitrate. Various technologies lay on the MPEG-4 compression framework, such as Blu-ray videodisc, video streaming (e.g., YouTube, Dailymotion), surveillance camera, handy video recorder, etc. MPEG-4 standard Part 10 - AVC is one of the most commonly used formats. For rest of the discussion, H.264/AVC is referred to as H.264 unless specified otherwise.

Technically, in comparison to the previous standard (ISO, 2000), H.264 standard incorporates various new features to further improve video compression efficiency. Notably, these features include intra-prediction in intra-frame, multiple frames reference capability, quarter-pixel interpolation, de-blocking filtering post-processing, and flexible macroblock ordering (ISO, 2010; Yang et al., 2011; Wedi, 2002; List et al., 2003; Shan-

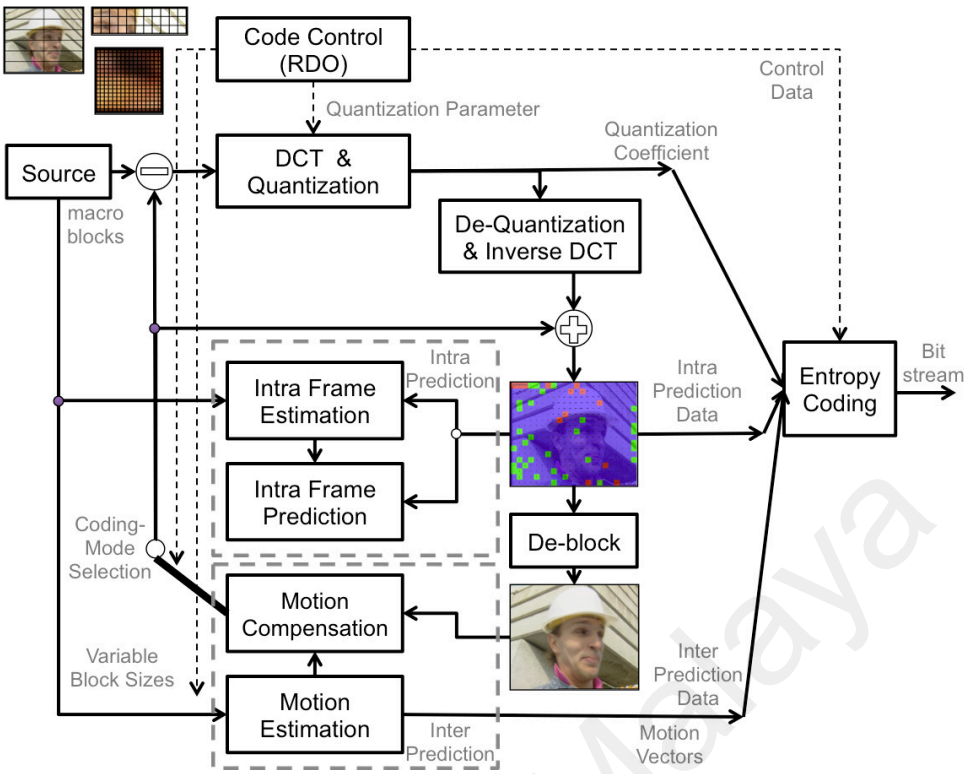


Figure 2.1: H.264 hybrid video encoder.

ableh, 2012a). In general, H.264 standard divides the sequences of images into several groups of pictures (GOP). These images are labeled as Intra- (I-), Predicted- (P-), and Bidirectional predicted- (B-)frames, depending on the order in which they appear.

The hybrid encoding process of the H.264 video compression standard is shown in Figure 2.1. At the source part, each frame is divided into non-overlapping blocks of uniform size (i.e., 16×16 pixels) called macroblocks, and these macroblocks are handled uniquely depending on their types. Each macroblock can be further divided into smaller blocks (i.e., 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) with 4×4 being the smallest possible block size, as shown in Fig. 2.2. The macroblock are subjected to Discrete Cosine Transformation (DCT), quantization and entropy coding. First, the pixel values in a macroblock are used in the DCT and quantization process. The outputs of the DCT and quantization processes, i.e., the quantized DCT coefficient, undergo the de-quantization and inverse DCT process for prediction and motion estimation purposes. In particular, the intra- and inter-prediction processes utilize these reconstructed pixel values to execute pixel value

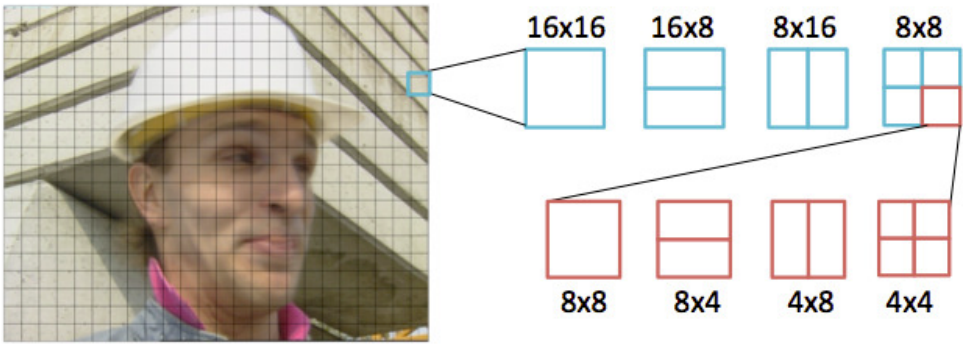


Figure 2.2: Coding block type

estimation and to make decision on coding-mode. Ordinarily, Rate Distortion Optimizer (RDO) is utilized to choose the best operational point between inter- and intra-mode for coding each macroblock. The code control block in Figure 2.1 represents an optimizer that regulates the selection of coding modes and block sizes (Sullivan & Wiegand, 1998). It requires high computational complexity in sequential processing to create data dependency of neighboring coding units. It also controls the Quantization Parameter (QP) to achieve the targeted video bitrate. Finally the result of the DCT and quantization process, prediction data, motion vectors, control data from RDO are sent for entropy coding. The output of entropy coding is a series of compressed video contents in the binary stream preceded and/or inter-leaved with various predefined markers. The combined bitstream is then transmitted and/or stored in various mediums.

Specifically in I-frame, the pixel values in a block are either coded directly by using coefficient in the transformed domain or predicted (i.e., intra-prediction) using neighboring blocks in the same frame to exploit the spatial redundancies within a frame. On the other hand, in P-frame, motion estimation (i.e., inter-prediction) between two frames can be implemented to take advantages of the temporal redundancies. For that, the previously encoded frame, which itself could be a motion compensated frame, is decoded and its prediction errors, if any, are decoded and added to the decoded frame for motion estimation purposes. In the case of B-frame, up to two frames (past and/or future) can be considered for motion estimation purposes. Outputs from the aforementioned processes, including

coefficient values, prediction errors, motion vectors, etc., are further entropy coded.

There are two entropy-coding methods in the H.264 standard to encode the quantized DCT coefficients, namely, Context-Adaptive Variable Length Coding (CAVLC) (Bjøntegaard & Lillevold, 2002) and Context-Adaptive Binary Arithmetic Coding (CABAC) (Marpe et al., 2003). CAVLC processes a macroblock in the form of run-level pairs, whereas CABAC binarizes all the entities for further processing. Both of them choose the best table or probability model depending on the local context to encode syntax including quantized DCT coefficients, motion vector information, etc. CABAC always offers higher compression gain because it allows the assignment of a non-integer number of bits to each symbol of an alphabet, and permits the adaptation to statistics of non-stationary symbol. However, CABAC is of higher computational complexity when compared to CAVLC. Output of the entropy coder is then preceded by and/or inter-leaved with various predefined markers to form the H.264 format compliant video for transmission and storage purposes.

2.3.3 HEVC (H.265)

HEVC is the latest video-coding standard published in 2013. The main achievement of HEVC standard is its significant improvement in compression performance when compared with the previous state-of-the-art standard (i.e., H.264), with at least 50% reduction in bitrate for producing video of similar perceptual quality (ISO, 2013). HEVC standard is designed to address essentially all existing application of H.264. It achieves two addition major achievements, namely: (a) handle higher video resolution by introducing larger coding unit size, and (b) capitalize on parallel processing architecture in the video encoder design to boost the encoding time.

HEVC also introduced several new features to achieve higher video compression, such as various coding unit sizes, more intra-prediction modes, residual quad tree, sample

adaptive offset, tiles and wave front processing, etc. (Sullivan et al., 2012). Among these features, implementation of the variable prediction and transform unit size are exploited in this research for information hiding purposes.

Similar to H.264, HEVC treats a video as a sequence of images, namely, video slices (i.e., video frames in H.264), where these images are labeled as I-, P- and B-slices, depending on the order in which they appear. Each slice consists of certain number of Coding Tree Unit (CTU), while each CTU consists of some number of Coding Unit (CU) with size of 64×64, 32×32, 16×16 or 8×8 pixels. Each 8×8 CU can be further split into 4×4 pixels in the prediction process. The availability of CU in various sizes allows the video encoder to encode each part of the video slice based on its local texture (i.e., spatial activity). The encoder decides the CU size and the quantization value in each CTU based on the desired bitrate. In particular, due to the quantization process, a region with high spatial activity (e.g., water waves) requires smaller CU sizes to precisely capture the variation in pixel intensity values. On the other hand, a smooth region (e.g., background or cloudless sky) can be encoded by using larger CU size. In the case of low bitrate (e.g., 10 kilo (k)bps), a large quantization value (e.g., QP $= 40$) is utilized to encode every CTU with larger CU size, which leads to quality degradation and smaller video file size. On the other hand, for high bitrate (e.g., 100 Mbps), small quantization value (QP $= 12$) is utilized and most CTUs are coded in smaller CU sizes for representing the region without compromising on perceptual video quality, but at the expense of larger video file size.

The prediction and transformation processes utilize the CU structure to perform intra/inter prediction, DCT and quantization. The CU utilized in the prediction and transformation processes are called Prediction Unit (PU) and Transform Unit (TU), respectively. Specifically, in I-slices, CU can only be coded by using squares, which include $64 \times 64, 32 \times 32, \cdots, 4 \times 4$ pixels. On the other hand, in P- and B-slices, CU can be encoded by using all possible arrangements, including $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$

for $N \in 4, 8, 16, 32$ and Asymmetry Motion Partition (AMP), which can assume the dimension of $2N \times nU$, $2N \times nD$, $2N \times nL$, or $2N \times nR$. The implementation of AMP in HEVC provides better prediction reference and less bitstream size overhead for PU that contains slight movement at either the upper, lower, left or right part of a CU in P- or B-slice. Here, each CU is encoded with a depth value δ, to indicate the N value in CU size definition. The parameter $\delta \in \{0, 1, 2, 3\}$ signifies that $N = 64/2^{\delta}$ (e.g., CU of size $(64/2^{\delta}) \times (64/2^{\delta}) = 32 \times 32$ are considered for $\delta = 1$), except for $\delta = 3$ where both 8×8 and 4×4 blocks are included.

Each PU defines a region in the slice that shares the same prediction mode (i.e., intra, inter, skip and merge) (Vanne et al., 2014). In HEVC, intra prediction allows 33 angular predictions (i.e., modes) and two non-angular modes, which are respectively denoted by Direct Current (DC) and planar. The current PU's intra prediction is obtained through the extrapolation of values derived from the reference pixels of the neighboring PU's, a process, which requires numerous arithmetic operations per predicted pixel value. On the other hand, inter prediction encodes PU by storing the motion vector, which points to the position of the matching PU in the reference slice, as well as the residual values, which are the differences (prediction errors) between the reference PU and current PU.

2.4 Overview of Information Hiding in Video Domain

Information hiding is a process of inserting information (e.g., internal information from video content or any external information) into a media (i.e., video file) by manipulating the video content to serve a specific purpose(s). Here, information hiding can be referred to data hiding, data embedding, and information embedding, interchangeably. In the compressed video domain, information hiding is commonly utilized for general embedding purpose (e.g., embed video headers into video content), security purpose (e.g., watermarking, copyright protection, and authentication), error concealment purpose (e.g.,

content recovery due to transmission loss) and compression purpose (e.g., hide part of the video content in the current frame into the subsequent frame to reduce the video bitrate). Information hiding can be carried out at various stages in the video encoder, including prediction stage, transform stage, quantization stage, and entropy coding stage. Note that the discussion focuses on H.264 because it is widely researched in the literature when compared to the HEVC standard.

2.4.1 Prediction Stage

In video coding, the prediction process can be executed at various levels of granularity to achieve the targeted bitrate or image quality. In particular, a coding block can be further decomposed into smaller blocks of various sizes prior to the prediction process. Figure 2.2 illustrates some of the possible ways to decompose a coding block into combination of smaller blocks. These block sizes will be determined through an exhaustive search approach based on the RDO process. It decides the types of prediction to be utilized in each coding block by executing pixel value estimation for intra and inter-prediction modes.

Several researchers proposed to manipulate the block prediction process in vector quantization based image compression to embed information. Different coding methods are applied on dedicated blocks, such as truncate coding (J.-M. Guo & Tsai, 2012), and side-match vector quantization (M.-N. Wu et al., 2008). In the compressed video domain, similar approaches are taken by exploiting mode, block size, entities, etc. that are related to the prediction process.

2.4.1.1 Intra-Frame Prediction

If a macroblock is encoded in intra-mode, the prediction is carried out by utilizing one of the ρ type of prediction mode, i.e., $\rho = 14$ in H.264 (9 for 4×4 blocks, 4 for 16×16 blocks, and the skip mode) while referring to the previously encoded and reconstructed blocks, where they themselves could be macroblocks predicted using the intra-prediction

mode. To exploit mode selection for information hiding, mapping rules are usually considered to improve the payload without causing significant bitrate overhead (Hu et al., 2007; Zhu, Wang, Xu, & Zhou, 2010). These methods categorize the selected prediction modes for 4×4 in Intra Prediction Mode (I4MB) into two groups so that the first group denotes '0' and the other denotes '1'. The prediction process is forced to assume the best mode among those belonging to the group that represents the information to be embedded. The embedded message can be readily decoded by referring to flags such as *pre_intra_4×4_pred_mode*. Kim et al. also exploit the intra-prediction mode (in combination with coefficients) to realize blind (i.e., the extraction process can be performed without referring to the original frame) and semi-blind watermarks (D.-W. Kim et al., 2010). Similar approach is proposed by Xu et al. where macroblocks are selectively chosen based on a chaotic sequence and the most probable prediction mode is manipulated to embed information (D. Xu et al., 2010).

Yang et al. restrict information hiding to 4×4 blocks in I-frame using matrix encoding (Yang et al., 2011). 4×4 blocks are chosen because they contain high number of non-zero DCT coefficients and modifying their prediction modes (for hiding information purposes) lead to less visible artifacts as compared to the case of 16×16 blocks. Two bits of information are encoded by three blocks through matrix encoding. Experiment results on several test sequences demonstrate that this technique can achieve blind extraction in real-time.

2.4.1.2 Inter-Frame Prediction

In order to increase the coding efficiency in inter-prediction mode, H.264 standard has adopted seven different block sizes (namely, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4) and the motion estimation algorithm is invoked for each block size. The block type that results in the minimum number of bits will be selected. Kapotas et al. pro-

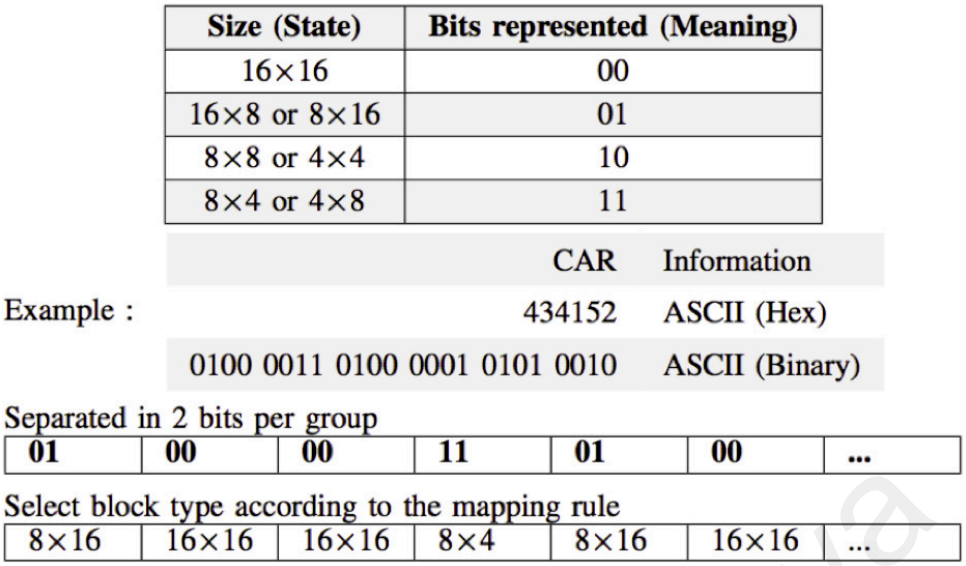


Figure 2.3: Mapping rules for prediction block type to embed information.

pose to force the encoder to choose a particular block type according to the information to be embedded (Kapotas & Skodras, 2008). In this technique, each block type is assigned to represent two bits. Then the information is divided into segments (i.e., each of length two bits) and each segment is encoded using block size as shown in Fig. 2.3. These macroblocks are then motion estimated using the forced block size. This technique only affects the visual quality of the video insignificantly. The payload is high and it is proportional to the size of host video.

2.4.1.3 Motion Vector Displacement

Information hiding can be achieved by using the motion vector attributes, including phase angle, horizontal and vertical magnitudes. Jordan et al. initiate this technique for video watermarking purpose (Jordan et al., 1997). Then, Zhang et al. and Dai et al. propose enhanced versions of Jordan et al.'s technique by restricting information hiding to specific types of inter-frame (J. Zhang et al., 2001; Dai et al., 2003). In particular, frames consisting of motion vectors with large magnitude and small in phase angle are considered. These three methods are studied by Su et al. and a steganalysis method is proposed (Y. Su et al., 2011). Similarly, Guo et al. propose a method to embed secret information in the motion vectors between two P-frames (Y. Guo & Pan, 2010). In particular, horizontal

and vertical offsets (i.e., odd or even) in motion vectors are modified to embed information. Experiment results show that this technique meets the requirement for real-time application in stream switching application.

Later, Xu et al. consider to embed information using DCT coefficients in I-frame and magnitude of motion vectors in P-frame to achieve higher payload (C. Xu et al., 2006). Aly extends Xu et al.'s technique by proposing a different information hiding approach aiming to achieve a minimum prediction error and bitstream size overhead (Aly, 2011). Instead of using magnitude and phase angle, Aly's technique exploits the prediction errors caused by the associated motion vector displacement to determine its suitability for information hiding. In particular, the prediction error is compared to an adaptive threshold. This technique causes low distortion in the video and suppresses bitstream size increment. Recently, Cao et al. design an adaptive and reversible information hiding technique based on motion vectors (Cao et al., 2012). Cao et al. implement calibration techniques to recover the inter-macroblocks whose motion vectors are modified for embedding purposes. Deng et al. compare the methods proposed by Su et al. and Cao et al., and propose an improved technique for higher detection accuracy (Deng et al., 2013).

2.4.1.4 Motion Vector Search Range

Hierarchical-based motion estimation is adopted in H.264 standard to support a range of block sizes and quarter-pixel precision for achieving high compression efficiency. For each macroblock, the motion estimation process starts by searching for the best macroblock in the integer-pixel level, then proceeds to the sub-pixel level around the best integer-pixel position, and finally continues searching at quarter-pixel level around the selected sub-pixel position to find the best matching point. The information can be embedded by modulating the search points of the motion estimation process according to the mapping rule. In particular, this technique utilizes two non-overlapping sets of search

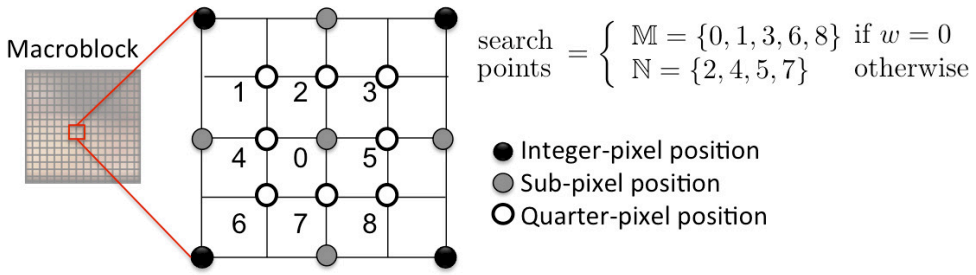


Figure 2.4: Quarter pixel search point position for information hiding.

points (i.e., $\mathbb{M}$ and $\mathbb{N}$) to embed information. A possible arrangement is shown in Fig. 2.4, where w denotes the bit to be embedded. Experiment results from Zhu et al. indicate that no obvious change is observed in terms of bitrate as well as quality of the video (Zhu, Wang, & Xu, 2010). Nonetheless, the change in direction of motion vector inevitably introduce larger prediction error. However, this error will be handled automatically (i.e., absorbed into the residual signal) and its effect to bitrate is insignificant.

2.4.2 Transform Stage

Similar to information hiding in still image, luminance DCT coefficients are commonly utilized as the venue to hide information by using bit plane replacement (i.e., odd-even) embedding technique. Odd-even indicates the embedding process by flipping binary number(s) between odd and even value. Ma et al. propose to embed information into the quantized DCT coefficients (luminance) in I-frame (Ma et al., 2010). Based on the analysis of the relationship between the DCT coefficients and the distortion incurred in pixel values, several coefficients are paired for information hiding and distortion adjustment purposes. Results show that this method is able to eliminate I-frame distortion drift, achieves higher payload, and causes lower visual distortion. As an extension of Ma's work, Lin et al. propose to embed two bits in the luminance channel of the selected macroblocks (Lin et al., 2013). Prediction mode (i.e., I4MB) and selected pixels in this macroblock are defined in their proposed mapping rule to achieve higher payload while maintaining video quality.

Earlier, Huang et al. embed message in the DC coefficient, followed by low-frequency Alternating Current (AC) coefficients (Huang & Shi, 2002). Similar technique is pro-

posed by Barni et al., who define the video content as a video object plane in the video object layer (Barni et al., 2005). Barni et al.'s technique computes the frequency mask to select a pair of DCT coefficients and divide them into two parts. For the non-zero DCT coefficients part, information is inserted into coefficients of magnitude greater than a pre-defined threshold level. For the zero DCT coefficients part, the QP is manipulated to represent embedded message. Chung et al.'s technique applies histogram shifting on DCT coefficients in I-frame and manipulates motion vectors in neighboring macroblocks in P-/B-frames to realize error concealment (Chung et al., 2010). Similarly, Shahid et al. propose to manipulate non-zero DCT coefficients in intra and inter-frame with different QP to embed information (Shahid et al., 2011). In (Chen et al., 2012), Chen et al. exert Watson's visual mask construction (Watson, 1993) and Lin et al.'s payload estimation method (Lin et al., 2013) to realize information hiding using the selected DCT coefficients in I-frame.

Thiesse et al. hide Motion Vector Competition Index (MVComp) in the chroma and luma DCT coefficients to reduce the total bitrate in the H.264 video stream (Thiesse et al., 2010a,b, 2011). A mapping rule is introduced based on the sum of the DCT coefficients $\mathbb{S}_i$ to control the bitrate change and minimize the distortion caused by motion prediction at reduced precision. The parity of $\mathbb{S}_i$ (coefficient sum) is utilized to represent MVComp by adding $\mathbb{H}_i$ to $\mathbb{S}_i$ (when necessary) to denote the predictor index $\mathbb{I}_i \in \{0,1\}$ as follows:

$$\mathbb{S}'_i = \begin{cases} \mathbb{S}_i & \text{if } |\mathbb{S}_i| \bmod 2 = \mathbb{I}_i, \\ \mathbb{S}_i + \mathbb{H}_i & \text{Otherwise.} \end{cases} \tag{2.1}$$

The results show good compromise among bit saving, prediction error propagation in luma texture, and visual quality in chroma aspect.

Meuel et al. work on a similar technique to hide Region of Interest (ROI) information

into the quantized DCT coefficients (Meuel et al., 2007). ROI information is utilized to represent significant object in still image and it is constructed based on skin pixel (boundary of object in still image):

$$\sqrt{(Cb-\widetilde{Cb})^2+(Cr-\widetilde{Cr})^2}<\mathbb{D}, \tag{2.2}$$

where $\widetilde{Cb}$ and $\widetilde{Cr}$ are the reference of Blue-difference Chrominance Component (Cb) and Red-difference Chrominance Component (Cr), respectively, and $\mathbb{D}$ is the threshold determining if the current pixel is marked as a skin pixel. Its position, width and height values are embedded into two Least Significant Bits (LSBs) of the non-zero DCT coefficients of the current frame. This technique achieves lossless reconstruction, but the results indicate that the frame payload is insufficient to host the entire ROI information.

Similarly, Yin et al. propose to hide information in edge pixels by using edge detection and multi-directional interpolation techniques on residual information (Yin et al., 2001). This technique is designed for error concealment application at the decoder in still image. Along the same direction, Yilmaz et al. propose to hide quantized edge information (deduced from neighboring macroblocks) for error concealment purposes (Yilmaz & Alatan, 2003). Based on (Yin et al., 2001) and (Yilmaz & Alatan, 2003), Kang et al. embed the important information of macroblocks including coding mode(s), reference frame(s), motion vector(s), etc. into the next frame using odd-even embedding method in DCT coefficients (Kang & Leou, 2005). Li et al. embed information in Discrete Wavelet Transformation (DWT) coefficients for video watermarking purposes (G. Li et al., 2009). The scaling coefficients in DWT are utilized to embed low resolution video frame while the watermark information is embedded using wavelet coefficients. Besides that, Wu et al. propose information hiding architecture, design and implementation in still image and video domains (M. Wu & Liu, 2003; M. Wu et al., 2003). They recursively embed infor-

mation in each video frame by using modulation and multiplexing techniques selectively in different regions for handling uneven payload.

Instead of modifying non-zero DCT coefficients, Nakajima et al. exploit the (zero) run component of non-zero coefficients to embed information in a compressed video (Nakajima et al., 2005). For each block, the position of the last non-zero coefficient (with respect to the zigzag scanning order), denoted by $\mathbb{L}$, is computed. The value $\mathbb{B} = \log_2(64 - \mathbb{L})$ then determines the number of bits that can be embedded in the current block. Information is embedded by introducing a non-zero DCT coefficient $\mathbb{V}$ at position $\mathbb{L} + \mathbb{B}_{10}$, where $\mathbb{B}_{10}$ is the decimal representation of $\mathbb{B}$ bits from the information to be embedded. The sign and magnitude of $\mathbb{V}$ can also be exploited for information hiding purposes.

2.4.3 Quantization Stage

In Wong et al.'s technique, quantization scale of each macroblock (if it is coded) is utilized for information hiding. This method is able to preserve the video bitstream size with low embedding complexity (K. Wong & Tanaka, 2007). In another work, Wong et al. maintain quality of the modified video exactly to that of the original host even after information hiding (K. S. Wong et al., 2009). If '0' is to be embedded, the macroblock is left as it is. Otherwise, the macroblock is manipulated by dividing the quantization scale by a prime number and multiplying each non-zero DCT coefficient by the same prime number.

Shanableh utilize matrix encoding technique to hide information in coded quantization scales and motion vectors of H.264/Scalable Video Coding (SVC) compressed video (Shanableh, 2012b). A video transcoding process is applied to allow information to be embedded in motion vectors using a non-iterative procedure regardless of the availability of the original raw video. Matrix encoding is utilized to minimize the number of modifications on quantization scale. Here, the coding structure of H.264/SVC is exploited

to increase payload. In particular, quantization scales in both the base and enhancement layer(s) are utilized to embed information. In another article by Shanableh, the Flexible Macroblock Ordering (FMO) feature and quantization scale are modulated to embed up to three bits of information per macroblock (Shanableh, 2012a).

Su et al. embed information in the non-zero DCT coefficients that are representing the prediction residuals (P.-C. Su et al., 2011). This technique manipulates the selected DCT coefficients by using quantization step based on the Just Noticeable Difference (JND) to determine the amount of information that is allowed to be embedded into each coefficient. Su et al. adopt Watson's perceptual model (Watson, 1993) and implement this technique as a video watermarking scheme.

2.4.4 Entropy Coding Stage

Two entropy coding methods, namely CAVLC and CABAC, are available in H.264 compression standard, and they are also exploited for information hiding purposes. In CAVLC, run-level coding is utilized to compactly represent strings of zeros by referring to the Trailing Ones (T1s) table to mark the last three ± 1 coefficients (Liao et al., 2010). Liao et al. utilize the T1s codeword (0-3) to carry information based on the following mapping rule:

$$\widetilde{T1s} = \begin{cases} 2, & \text{if } w = 0 \text{ and T1s} \geq 3, \\ 1, & \text{if } w = 1 \text{ and T1s = 2 or} \\ & \text{if } w = 1 \text{ and T1s = 0,} \\ 0, & \text{if } w = 0 \text{ and T1s = 1,} \\ \text{unchanged,} & \text{otherwise.} \end{cases} \tag{2.3}$$

$\widetilde{T1s}$ is the modified T1s codeword and w is the information bit to be embedded. This method is of low complexity and the quality degradation caused by information hiding is imperceptible in the resulting video. At the same time, this technique results in less

variation in bit length (i.e., bitstream size) and it is able to execute in real time. Similar approach is taken by Kim et al. where sign of the non-zero DCT coefficients and the number of non-zero DCT coefficients in I4MB are modified to embed information (S. Kim et al., 2007). Lu et al. consider the run-level pairs in macroblock for video watermarking purpose (Lu et al., 2005). In particular, the difference of average value of levels (from run-level pairs in each macroblock) from the original and filtered frames are utilized to encode the watermark information. On the other hand, Mobasseri et al. utilize the codeword of unused run-level pairs (i.e., those that never occurred in the video) in CAVLC to embed information (Mobasseri & Marcinak, 2005). They associate selected run-level pairs with unused ones to represent '0' and '1', respectively. This algorithm forces the selected pairs in intra-coded macroblock to be the associated pairs depending on the information to be embedded. However, side information is required to mark the originally unused codewords in the Variable Length Coding (VLC) table for detecting the embedded information.

Seo et al. apply LSB insertion on significant coefficient *sig_ctx* in context mapping during CABAC process (Seo et al., 2008). The LSB of each *sig_ctx* (absolute value) is manipulated by ± 1 to indicate the embedded bit. In year 2011, Wang et al. embed information in LSB of syntax elements (represented by values) during the binarization process in CABAC, which is a process to concatenate all the syntax elements in binary format (i.e., unary binarization) with delimiters (R. Wang et al., 2011). Xu et al. manipulate the K-th exponential Golomb code in the binarization scheme to embed information based on code mapping (D. Xu & Wang, 2011). Both researchers manipulate CABAC for watermarking purposes.

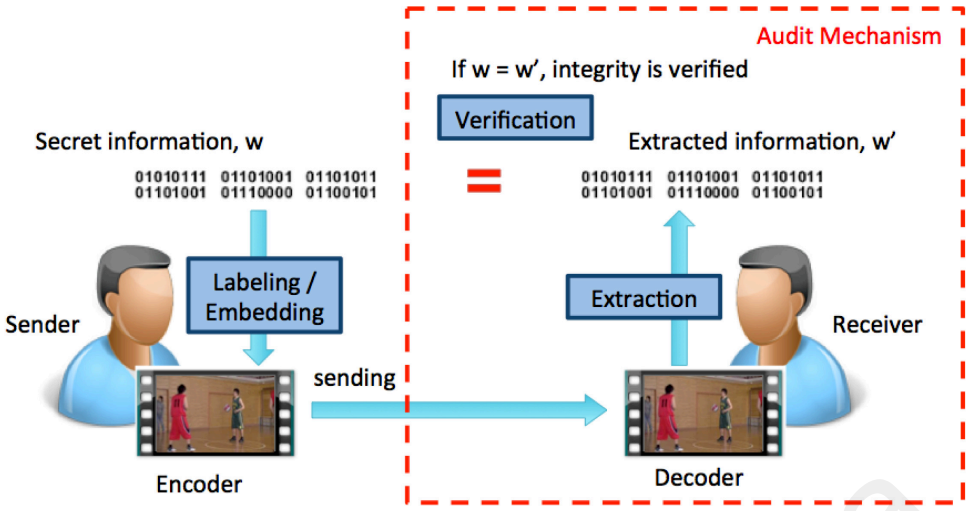


Figure 2.5: Overview of authentication scheme design.

2.5 Overview of Video Authentication Scheme

Video authentication is the act of confirming that the content of the video has integrity, viz., has not been tampered. Commonly, video authentication relies on two factors, namely the need for some secret information (e.g., password or key) and a mechanism designed to audit the authentication (e.g., video decoder) of content. Specifically, secret information (e.g., binary string) is a unique identifier selected by the sender (i.e., encoder). Here, the receiver (i.e., decoder) authenticates the genuineness of the video stream remains intact. Figure 2.5 shows an overview of authentication design to illustrate the relationship among secret information, sender, receiver and audit mechanism for verifying the integrity of video.

When handling compressed video, authentication is commonly achieved by four ways, namely, cryptography based, content based, labeling based and watermarking based. Authentication usually leads to additional processing overhead (e.g., cryptography and content based), bitrate increment (e.g., labeling based) or degradation in perception quality (e.g., watermarking based). Each authentication approach will be detailed in the following sub chapters.

2.5.1 Cryptography based Authentication

In cryptography based video authentication, cryptographic algorithm (e.g., hash functions) is utilized for protecting the confidentiality, integrity and availability of the authentication code in video. It encrypts the authentication code to prevent code imitation on authenticated video. This algorithm is a mathematical operation that takes an arbitrary block of data and returns a fixed-size bit string, which can be used as the authentication code. The fact that this function is one-way ensures that knowing the authentication code does not leak the value of the input data block.

In the literature, there are several well-designed cryptography based authentication schemes, include S/Key (i.e., one time password), Secret Key IDentification (SKID) (i.e., using symmetric cryptography and shared key between two parties) and public key authentication (Krzyzanowski, 1997). Each of the authentication scheme utilizes algorithms (e.g., Message-Digest Algorithm 5 (MD5) (Rivest, 1992), Secure Hash Algorithm (SHA) (NIST, 2002), Data Encryption Standard (DES) (NIST, 1999), Advanced Encryption System (AES) (NIST, 2001)) for hashing the authentication code. In the past, Tartary et al. proposed an cryptography based authentication scheme for any digital content by utilizing the Reed-Solomon code (Tartary et al., 2011). They utilized the list of recoverable codes in network stream distribution for designing their authentication protocol. A cryptographic function is introduced along with digital signature and hash function to ensure the robustness of the designed protocol. Later, Ren et al. introduced a cryptography based authentication scheme with loss-tolerant feature (Ren & O'Gorman, 2012; Ren et al., 2013). They combined a cryptographic fingerprint and video to achieve video authentication. However, this authentication scheme compromises on minor latency (i.e., access to the fingerprint) and video file size increment in the processed video.

In most of the authentication designs (e.g., content based, labeling based and water-

marking based), cryptography algorithm is applied for generating a unique authentication code, i.e., hashing the authentication code. This code verifies genuineness of video and authenticates the video source during the decoding process.

2.5.2 Content based Authentication

Content based video authentication extracts video characteristics (or features) to generate authentication code for verification of integrity. It also provides integrity protection by localizing tampered regions while allowing content-preserving changes (Lo et al., 2014). One of the earlier work by Queluz had relied on features such as edges and analyzed the problem of image/video integrity from a semantic, high-level point of view (Queluz, 1998). This work extracts essential content (e.g., edge pixels) from the video that remains intact after the video encoding and transmission process. The extracted content is encrypted and conveyed as an additional information for authenticating the integrity of the transmitted video. Wu studied the limitations of content based authentication by evaluating the resemblance of extracted features from two similar videos (C. W. Wu, 2002). Then, he proposed an authentication scheme to overcome these limitations by utilizing cryptographic digital signature scheme. With the same spirit, Atrey et al. (Atrey et al., 2006) utilized the differential energy between the video frames to verify the video integrity. In their proposed work, selected coefficients are hashed based on a cryptographic function and compared with the authenticated code to ensure the video integrity in three hierarchical levels, namely, key-frame level, shot level and video level. However, the sharing of authenticated code between two parties is not discussed, viz., it requires an additional channel to share the code secretly with the receiver to authenticate the video. Later, in Xu et al.'s work, video features (i.e., residues and predicted coefficients) is extracted from video frame blocks to generate authentication code (D. Xu et al., 2011). This code is embedded into the video stream for verifying the integrity of video and locating

tampered video frame during the decoding process.

One of the most significant advantages of the content based authentication is that the protection is introduced largely independent of the actual format of the content, which provides the greatest flexibility for subsequent content processing and adaptation (Zeng & Dong, 2008). Yet, in order to extract features, the video decoding process requires additional processing overhead. In addition, since similar videos generate similar features, it is possible for a forger to generate nonidentical video which has the same features (C. W. Wu, 2002). This shortcoming can be solved by combining the content based authentication scheme with cryptography based authentication scheme.

2.5.3 Labeling based Authentication

Video source and integrity authentication without referring to any available resources is possible by labeling (i.e., inserting) authentication code on the video content in a way that is transparent to a noncompliant decoder. Apparently, the absence of the labels implies uncertainty of the video integrity. To support the credibility of the labeling based authentication, video standards (e.g., MPEG-2, H.264) contain some common characteristics to allows the insertion of metadata into the header file (e.g., Digital Signature box). The term 'box' refers to a binary sequence that contains objects and has the general form of | size | type | contents |. Here, any unknown types of box is to be ignored by normal decoder, so there is the possibility of creating custom boxes without losing compatibility. Hence, the authentication can be realized by enabling one box to create a directive object specifically designed for labeling based authentication (Furht & Kirovski, 2006).

In addition to the utilization of header file, the authentication code can be appended to the video stream for integrity verification. This code can be generated by a hash function using a set of features, such as those extracted or derived from the video, as the input. Specifically, Baek et al. proposed a labeling based authentication framework through

public key infrastructure for video broadcasting network (Baek et al., 2013). They introduced an identity-based signature to authenticate online and offline broadcasted videos, and improved the performance of (Liu et al., 2010). With the same spirit, Song et al. put forward an interactive content-based authentication scheme using labeling for video streaming (Song et al., 2013). Their design generates levels of signature in the chosen video slices, multiple authentication paths as well as authenticating information on network packets. It is reported that their scheme is of high tolerance against packet loss. On the other hand, Wei et al. proposed an authentication scheme in the scalable video code stream, where the authentication codes are encapsulated (labeled) in the network abstraction layer unit (Wei et al., 2014). Their proposed scheme is efficient in detecting content-preserving manipulation attack (e.g., recompression), but vulnerable to content-changing manipulation (e.g., color or luminance) attack.

While offering attractive performances, video labeling based authentication can hardly provide the security feature to authenticate video due to the nature of the code appending process, which fails to prevent the code from being copied, manipulated or counterfeited.

2.5.4 Watermarking based Authentication

In watermarking based video authentication, the authentication code is imperceptibly embedded into the video stream rather than append to it. It overcomes the shortcoming of utilizing labeling based video authentication (i.e., fails to prevent from being copied, manipulated or counterfeited). Mobasseri et al. proposed a watermarking based authentication in any digital video by embedding the watermark matrix (i.e., authentication code) at bit plane level (Mobasseri et al., 2000). The proposed watermarking algorithm is capable in identifying cuts and splices both in length and duration of tampered video, but the authenticated video requires reversible process due to non-format compliant to the video standard. Later, Cross et al. extended Mobasseri's scheme by realizing the authentication

code embedding in variable length coding (refer to Chapter 2.4.3) under MPEG-2 standards (Cross & Mobasseri, 2002). The proposed embedding and recovery process does not require computationally expensive transforms and partial or full decompression, while the embedded watermark cannot survive after the re-encoding process. With the similar approach, Du et al. designed a watermarking authentication by manipulating the LSB of selected quantized DCT coefficients to embed authentication code in MPEG-2 (Du & Fridrich, 2002). In their design, to avoid the spread of distortion, authentication code is embedded into B-frames only because the distortion in B-frames due to code embedding will not spread to subsequent frames. However, it is only applicable with small GOP size (e.g., 5) and MPEG-2 due to the less reference dependency among frames in a sequences of video.

Later, Lang et al. (Lang et al., 2003) analyzed the risk of content authentication and presented a watermarking scheme that protects the scene description and content in MPEG-4. The authentication code is generated based on scene description and content in different frames and embed into some predefined pairs of quantized DCT coefficients. Lang et al.'s proposal is then realized by He et al. (He et al., 2004) in their proposed watermarking based authentication scheme. The watermark is generated by using error correction coding and cryptography hashing to increase robustness and security of the authentication system. It is embedded in a set of randomly selected discrete Fourier Transform coefficient groups before MPEG-4 encoding and the watermark is robust to scaling, rotation and inaccurate segmentation (He et al., 2003). In a similar direction, Roy et al. realized a video authentication scheme in hardware by using field programmable get arrays (Roy et al., 2013), where the authentication information is embedded to resist against cover-up and cropping segment attacks. An authenticated video under Roy et al.'s scheme can be easily adapted in common video standards with minor quality degradation, but may not be viable for video of higher resolution due to the high computational

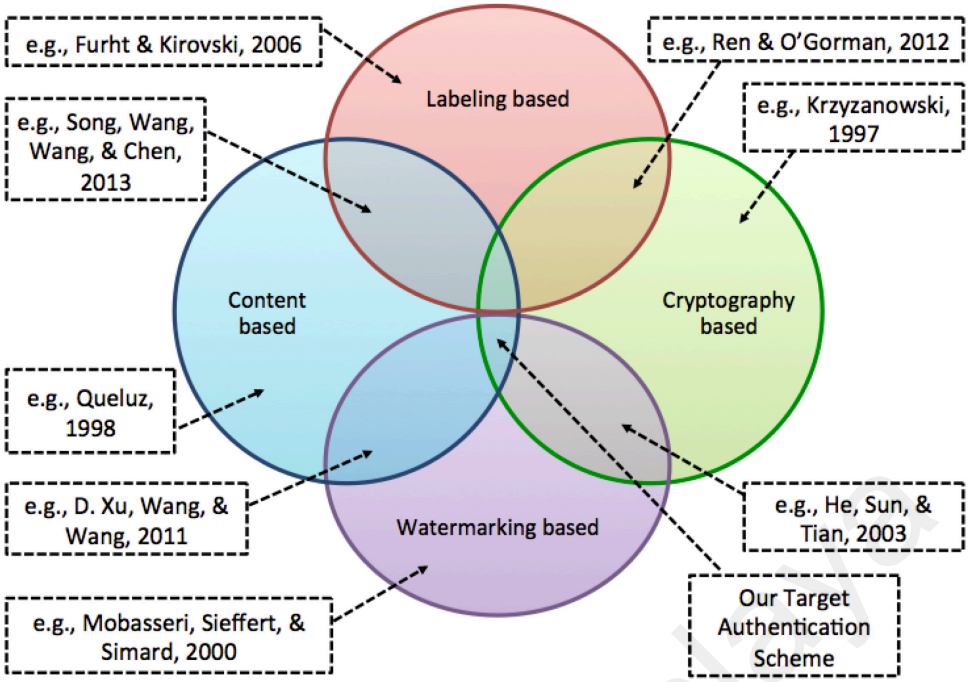


Figure 2.6: Classification of authentication methods.

complexity.

Figure 2.6 classifies video authentication into 4 classes as well as the hybrid classes. A representative example is presented in each class. For instance, Song et al. and Xu et al. label and watermark the authentication code respectively in their proposed content based authentication scheme. Ren et al. and He et al. hash the authentication code with cryptographic algorithm, label and watermark the code respectively in their proposed cryptography based authentication scheme. Noted that authentication code can be only either appended with the video content (i.e., labeling based) or embedded into video content (i.e., watermarking based). Here, the target authentication scheme design is clearly located between cryptography based, content based and watermarking based categories to achieve objectives 2, 3 and 4, as mentioned in Chapter 1.3.

2.6 Overview of Video Encryption Scheme

Security and confidentiality of multimedia contents (e.g., HEVC video) become a challenging research topic while it is gaining attention. The most straightforward method to secure a video content is to encrypt the entire bitstream by using naïve encryption algo-

rithm, e.g., AES, and selective encryption algorithm, e.g., sign bin encryption. These two algorithms are further detailed in the following sub chapters.

2.6.1 Naïve Encryption

Naïve encryption algorithm treats the video bitstream as binary data without considering the structure of the compressed video (Abomhara et al., 2010). It suffers from several drawbacks. First, the encryption/decryption process becomes computationally expensive for large-scale data, specifically for video of high resolution (e.g., 4K and 8K UHD) and high bitrate (Shah & Saxena, 2011). Therefore, it is not suitable for real time video transmission application, which have rigid restriction on delay and power consumption on mobile devices.

Second, naïve encryption prevents untrusted middle-box in the network to perform post-processing operations on the encrypted video bitstream such as transcoding and watermarking. In other words, it produces a non-format compliant encrypted video when it is applied directly to the compressed video.

2.6.2 Selective Encryption

Selective encryption algorithm emerges as an attractive alternative to Naïve Encryption algorithm (Hofbauer et al., 2014; Shahid & Puech, 2014). It considers the coding structure of the video compression standard in question and encrypts only the most sensitive information in the video bitstream. Massoudi et al. presented several selective encryption algorithms with the aim to reduce the amount of encrypted data while preserving a sufficient security level (Massoudi et al., 2008). The presented techniques preserve scalability function in image codec (e.g., JPEG2000), but infeasible for video due to the high complexity of video codec. Along the same direction, Wang et al. proposed a tunable selective encryption algorithm by modifying sign bits of non-zero coefficients, intra prediction modes and sign bits of motion vectors in H.264 video (Y. Wang et al., 2013). This

technique provides different encryption levels by adjusting three control factors with minimal impact of compression performance. However, the proposed technique is vulnerable to replacement attack (Martina Podesser, 2002), which improves the quality of the scrambled video, e.g., reveals some useful information by setting all the sign bits of non-zero coefficient to positive value.

For HEVC video, Shahid et al. proposed a selective encryption algorithm based on CABAC bin-strings in a format compliant manner by utilizing truncated rice code (Shahid & Puech, 2014). They put forward an algorithm to convert the encryption space from non-dyadic to dyadic, which can be concatenated to form the plaintext for AES-Cipher Feedback mode. Hofbauer et al. proposed another selective encryption algorithm for HEVC compressed video, which is applicable to a wide range of QP (Hofbauer et al., 2014). Their approach focuses on the AC Coefficient signs because the signs are not entropy coded and hence they can be altered directly in the bitstream. This approach enables fast encryption and decryption while maintaining full format-compliance and length preservation (i.e., identical bitstream size).

2.7 Problem Analysis

Some researchers utilized the statistical information from video content to detect forgery and tampering attacks (Upadhyay & Singh, 2011). The detection involves machine learning algorithms to classify video content, where neither the secret key nor the embedding process is needed. However, this class of authentication is not able to verify the source of the video. To overcome this shortcoming, the implementation of information hiding technique can be incorporated to achieve the video source identification feature.

With the same goal, several researchers then proposed information hiding based authentication by utilizing histogram shifting technique in the spatial domain (Caciula & Coltuc, 2014), or manipulating motion vector (Sharp et al., 2010), coefficients (Patra &

Patra, 2012) and macroblock (J. Zhang & Ho, 2006) in the compressed domain (e.g., MPEG-2, H.264). However, these schemes are not implemented in the latest video coding standard, i.e., HEVC, which is anticipated to replace H.264 standard especially when more high resolution (e.g., 4K) cameras, display devices and video contents are available.

Besides that, video security can be enhanced by applying encryption scheme on top of the authentication scheme, which has to be designed separately to form a joint video encryption and authentication scheme. However, most information hiding based techniques (e.g., authentication, watermarking) are unable to comply with encrypted domain. Although (X. Zhang, 2012) and (Hong et al., 2012) are able to extract embedded data in encrypted domain (e.g., image), none of them exploits the data extraction in decrypted domain, i.e., maintain embedded data after decryption. With this consideration, the research aims to provide alternative solution to secure video content (e.g., authentication) in both encrypted and decrypted video.

2.8 Summary

A overview of video compression standard was presented, including MPEG-1, MPEG-2, H.264 and HEVC standard. The fundamental of information hiding techniques based on video structure, transform domain and bitstream domain were presented to realize the authentication application. Next, several conventional labeling and watermarking based authentication schemes were reviewed. Then, naïve encryption and selective encryption algorithm were described to study the viability of authentication application in the encrypted domain. Finally, by analyzing the problems in existing authentication schemes, a solution was invented to secure video in encrypted and decrypted domain.

CHAPTER 3 : INFORMATION HIDING TECHNIQUE

3.1 Overview

In this chapter, the HEVC structure is exploited to realize information hiding. Technical steps to hide information into a video are described based on the coding structure of HEVC. Experiment results show that the perceptual quality is maintained and the embedded information can be extracted during the decoding process with minimum bitstream size overhead.

3.2 Introduction

Information hiding techniques are well researched for the previous state-of-the-art compression standard (i.e., H.264). As mentioned in Chapter 2.4, these techniques manipulate selected part of the video coding structure, including intra prediction (D.-W. Kim et al., 2010), motion vector (Y. Guo & Pan, 2010), DCT coefficient (Lin et al., 2013), syntax element (R. Wang et al., 2011), etc., to insert information. The application of information hiding includes authentication that embeds unique code for verifying integrity of media (Du & Fridrich, 2002), watermarking that inserts copyright information (D. Xu et al., 2011), steganography that camouflages secret information (Marvel et al., 1999), error concealment that aims at improving quality when transmission error occurs (Chung et al., 2010), etc.

Since HEVC is recently finalized, literature review shows that there is still no information hiding technique designed to specifically exploit its coding structure. Hence, an information hiding technique is put forward based on the CU structure in HEVC. This approach manipulates the size of CU decision on every coding tree unit to embed external information based on the pre-defined mapping rules. Particularly, each CU is forced to assume certain size for representing the information to be embedded. With this ap-

proach, the encoder decides the most appropriate size for every CU and the encoded video maintains format-compliance without compromising perceptual quality, at the expense of slight bitstream size expansion.

To improve payload, the odd-even based information hiding technique is further deployed by manipulating the non-zero DCT coefficients in certain ranges, in which case each range depends on the size of CU. Results suggest that by combining both approaches, improvement is achieved in the terms of payload for the higher bitrate scenario and insignificant degradation in perceptual video quality for the low bitrate scenario.

3.3 Design and Implementation

During encoding, the RDO calculates the cost function (i.e., a tradeoff between the distortion produced and the number of bits spent) of each possible block size for coding a given CU (Sullivan & Wiegand, 1998). RDO selects the size of CU with the lowest cost as the final decision to achieve the best compression ratio based on the desired bitrate. Instead of using the size suggested by RDO, the size of CU is forced to be the one representing the information to be embedded based on a predefined mapping rule. A possible implementation is shown in Fig. 3.1. In particular, different size of CU selection technique (i.e., CUSize) is applied when handling I-, P- and B-slices.

In each I-slice, all CU are forced to be encoded using 8×8 or 4×4 mode to attain higher payload. In particular, the CUs are forced to assume the respective sizes according to the mapping rules shown in Fig. 3.1. In this case, the size of CU in 8×8 and 4×4 pixels denote '1' and '0', respectively. In P- and B-slices, the size of CU is decided based on two categories, where one encodes '0' and the other encodes '1'. In particular, category '0' includes $2N \times N$, $2N \times nU$, $nL \times 2N$, and $N \times N$, while category '1' includes $N \times 2N$, $2N \times nD$, $nR \times 2N$, and $2N \times 2N$. These categorizations are summarized in Fig. 3.1. Note that in I-slice, the size of CU is forced to be 8×8 for $w = 1$, i.e., the same mapping rule

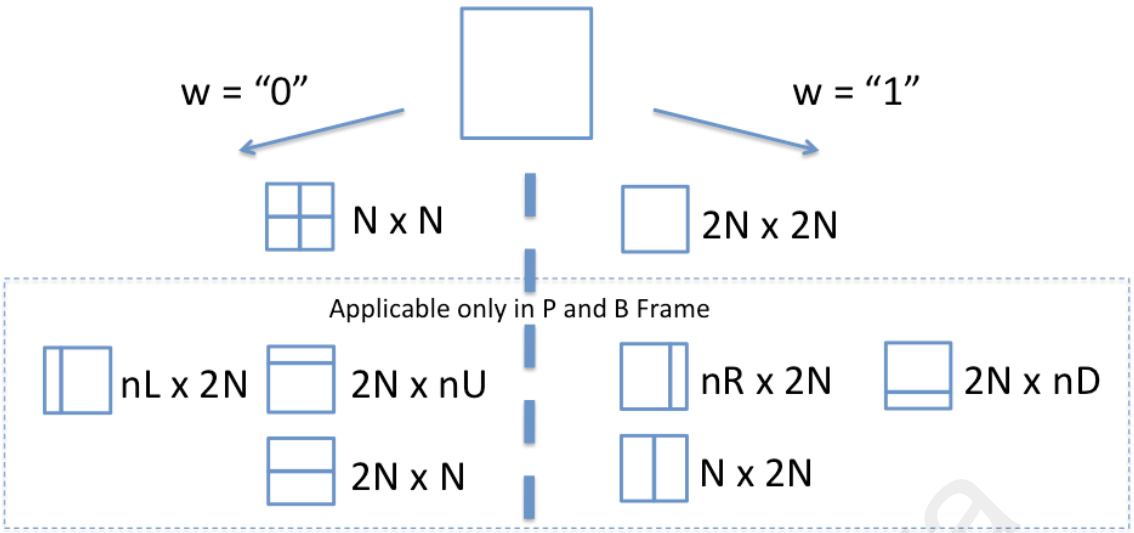


Figure 3.1: Two categories of CUSize utilized in the proposed information hiding technique.

for P- and B-slides is applied without the consideration for $2N \times nU$, $nL \times 2N$, $2N \times nD$ and $nR \times 2N$ depicted in Fig. 3.1. The notation of N, U, D, L and R are prescribed in Section 2.3.3. For instance, if the size of CU decided by RDO is 16×8 and $w = 1$, then the proposed technique will force the RDO to recalculate the cost of 8×16, 16×16, and two AMP's (i.e., $2N \times nD$, $nR \times 2N$), then choose the size with the smallest cost as the size of CU. For CU with larger block size (e.g., 32×32), it is reasonable to encode it by using some combination of smaller block sizes (e.g., two 32×16, four 16×16, etc.). It is because a larger CU size is utilized to encode a smooth region (e.g., background or cloudless sky) and a smaller CU size precisely captures a more complex region (e.g., water waves).

Figure 3.2 shows an example of information embedded in B-slices based on Fig. 3.1. The left figure shows the original CU structure and the right figure shows the modified CU structure with embedded information (i.e., yellow text). Here, the proposed technique modifies CU structure to embed information and achieves as close (similar) as possible to the original CU structure (i.e., refer to the CU structure of left and right figures). By utilizing the mapping rules in Fig. 3.1, RDO will choose the closest (i.e., smallest cost with the restriction by mapping rules) of CU structure to the original CU structure for

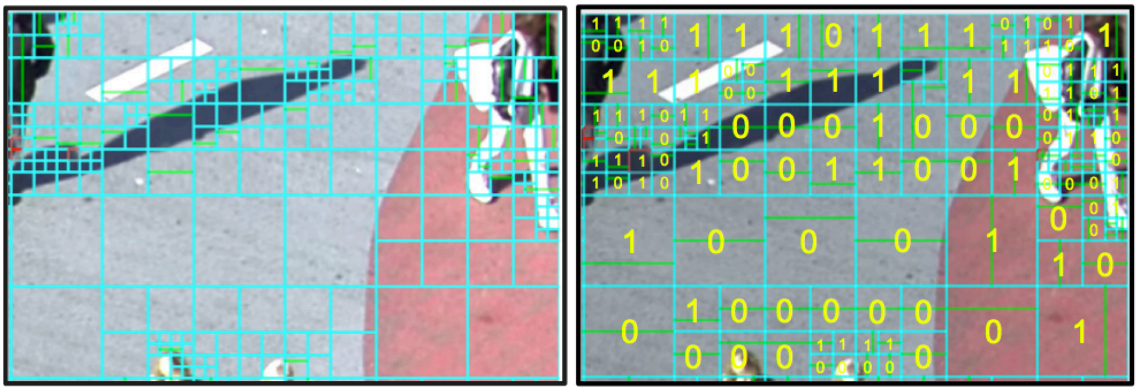


Figure 3.2: Example of original and modified coding unit in B-slice.

Table 3.1: Average coding unit count in all class of video

Video (Class)	Resolution	I-slice	P/B-slice
Traffic (A)	2560×1600	44305	39882
Kimono (B)	1920×1080	9744	9173
PartyScene (C)	832×480	6111	5564
BasketballPass (D)	416×240	1182	768
FourPeople (E)	1280×720	8829	4945
ChinaSpeed (F)	1024×768	7521	6852

representing the embedded information. However, compare to the original CU structure (with smallest cost among all possible CU structure), the modified CU structure contains higher bitstream size. Hence, the proposed technique maintains the video quality at the expense of slight bitstream size expansion. The embedded information can be extracted in the decoding process by examining the size of CU based on Fig. 3.1.

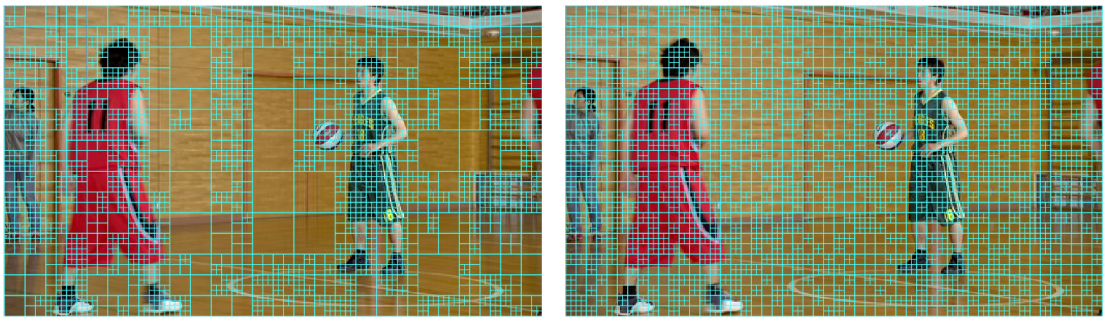
An average amount of information is computed that can be embedded in all video class, based on the encoded video with original CU structure. Table 3.1 shows the average number of coding unit in I-slices and P-/B-slices for all video classes. In fact, lower resolution video consists of lower number of CU, vice versa. Note that the computed average CU count is only an approximate amount of information that can be embedded, because of the modified CU count is based on external information and video slice contents with respect to the mapping rules in Fig. 3.1. For instance, in a complete Class A video sequence (e.g., 150 video slices), there are approximate 6 millions CU (e.g.,

150 slices $\times$ 40000 CU). In other words, it can embed approximately 750 kBytes external information by using the proposed information hiding technique.

Here, the effect of information hiding using size of CU is investigated based on the video bitrate and quality. The standard test video sequences for HEVC (i.e., *BasketballPass*, *BasketballDrill*, *FourPeople*, *Tennis*) from (*YUV sequences repository*, 2013) are considered to evaluate the basic performance of the proposed technique under various bitrates. The HEVC reference model video encoder version HM10.0 (*High Efficiency Video Coding: HEVC software repository*, 2013) is modified to encode the video sequences while hiding information into it. These video sequences are encoded by using a targeted bitrate ranging from 100 kbps to 50 Mbps. Here, CUSize (proposed in Chapter 3.3) can be considered as the improved version of (Kapotas & Skodras, 2008) in HEVC for comparison purposes. To combine both Coeff and CUSize techniques, the CUSize technique is first invoked, followed by the Coeff technique.

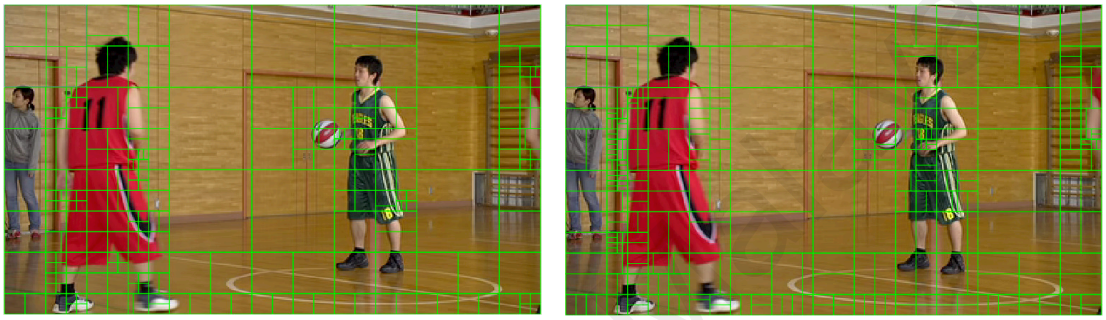
Table 3.2: Video quality and payload of the proposed techniques for various bitrates.

Video Sequences	Bitrate	Original		Coeff			CUSize			Coeff + CUSize		
		PSNR	SSIM	PSNR	SSIM	bits/sec	PSNR	SSIM	bits/sec	PSNR	SSIM	bits/sec
BasketballPass (416 × 240)	100k	28.50	0.789	28.17	0.780	5571	26.25	0.726	7752	26.10	0.722	11779
	500k	36.18	0.931	35.95	0.929	49992	35.23	0.959	13878	34.99	0.912	58150
	1M	39.87	0.965	39.58	0.963	116922	39.11	0.994	17126	38.82	0.956	127241
BasketballDrill (832 × 480)	100k	25.62	0.715	25.51	0.710	5452	23.08	0.624	25993	23.08	0.625	24637
	500k	32.43	0.912	32.24	0.908	35517	30.78	0.885	30869	30.78	0.881	54378
	1M	35.24	0.954	35.04	0.951	86693	34.15	0.940	38548	34.32	0.937	112633
	5M	42.48	0.994	42.21	0.993	594515	41.77	0.993	59230	42.06	0.992	656513
	10M	45.33	0.996	45.08	0.996	1327068	44.78	0.996	69906	45.05	0.996	1420820
FourPeople (1280 × 720)	100k	27.13	0.835	26.94	0.828	8559	22.47	0.730	61827	22.33	0.725	51712
	500k	36.41	0.983	36.18	0.981	48901	31.63	0.932	61883	31.34	0.927	79372
	1M	39.83	0.992	39.64	0.992	110236	38.57	0.989	66571	38.38	0.988	140880
	5M	44.18	0.997	44.07	0.997	608916	43.96	0.997	94786	43.85	0.997	683097
	10M	45.31	0.998	45.15	0.998	1297460	45.13	0.998	116141	44.98	0.998	1411533
Tennis (1920 × 1080)	100k	27.18	0.782	27.21	0.781	2347	21.84	0.566	144466	22.06	0.578	119117
	500k	32.23	0.907	31.89	0.902	34627	24.02	0.700	146326	24.06	0.698	142426
	1M	36.06	0.953	35.92	0.951	82913	30.62	0.879	147232	30.48	0.876	164993
	5M	41.25	0.992	41.14	0.991	543583	40.30	0.987	188189	40.19	0.987	600679
	10M	42.79	0.996	42.69	0.996	1183906	42.28	0.995	223226	42.18	0.994	1223809
	50M	46.15	0.999	45.81	0.998	6945862	45.84	0.998	425027	45.55	0.998	7190970



(a) Original I-slice. (b) I-slice with embedded info.

Figure 3.3: I-slice CU structure of compressed video at 1 Mbps.



(a) Original P-slice. (b) P-slice with embedded info.

Figure 3.4: P-slice CU structure of compressed video at 1 Mbps.

3.4 Experiment Result

Fig. 3.3(a) shows the first I-slice of the original compressed video of *BasketballPass*. The external information is embedded into the same I-slice and the output video is illustrated in Fig. 3.3(b). Note that almost all large blocks in Fig. 3.3(a) are decomposed into combinations of smaller blocks to embed information as suggested by Fig. 3.3(b). It is observed that the changes in block size between 8×8 and 4×4 are affecting the perceptual quality insignificantly. Results suggest that smaller CU can be implemented for all I-slices to achieve higher payload while maintaining video quality because smaller size of CU generally results in better video quality. Similar conclusions can be drawn for the P-slices. Fig. 3.4(a) and 3.4(b) show the original and modified P-slices, respectively. It is observed that some of the CU's are replaced by combinations of two (non-square) rectangles, including the AMP's which are not available in H.264.

To quantify the effect of information hiding on perceptual image quality, Peak Signal-

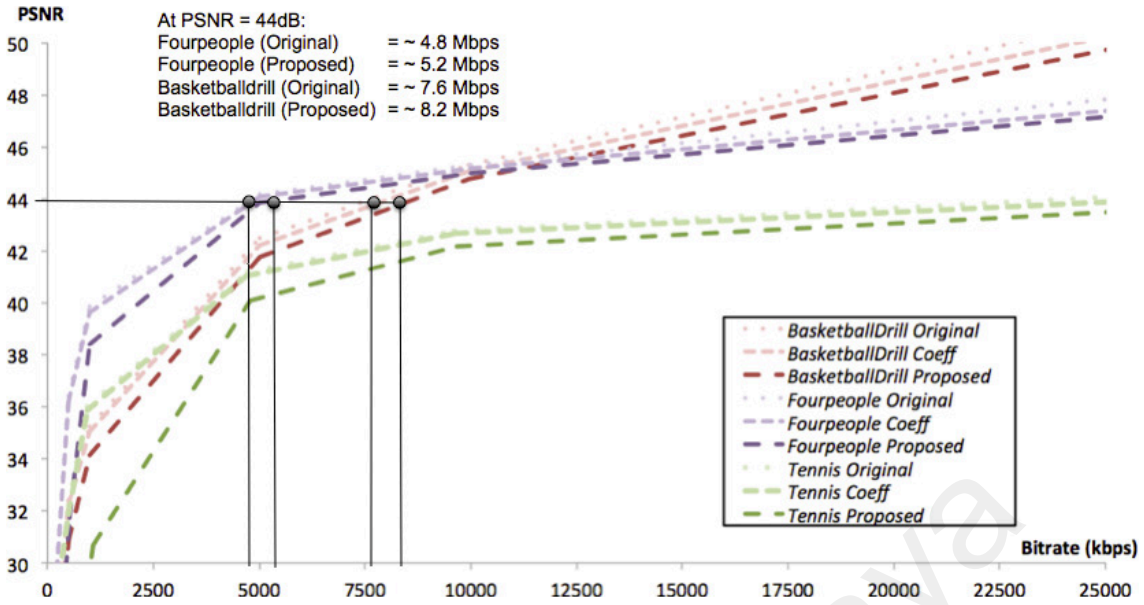


Figure 3.5: Rate distortion curve for the original compressed video, Coeff technique and the proposed combined technique.

to-Noise Ratio (PSNR) and Structural SIMilarity index (SSIM) (Z. Wang et al., 2004) are computed using the average value over the video sequence, and the results are recorded in Table 3.2. Quality of the original compressed video sequences are also recorded for reference purposes. Here, the results of the implemented techniques (i.e., Coeff, CUSize and the combination of both) are collected for the bitrate ranging from 100 kbps to 50 Mbps. To visualize the performance of the proposed combined technique, part of the results in Table 3.2 are translated into Fig. 3.5.

3.5 Discussion

Based on Table 3.2, it is observed that at low bitrate (i.e., 100 kbps), regardless of the video sequence (and hence the resolution), CUSize consistently offers higher payload when compared to Coeff. It is because at low bitrate, most coefficients are quantized to zero, while the numbers of CU are relatively consistent regardless of the bitrate. This trend is particularly obvious for video of high resolution (i.e., *Tennis*). On the other hand, as bitrate increases, the opposite trend is observed, i.e., Coeff offers significantly higher payload when compared to CUSize. This justifies the combination of both CUSize and

Coeff techniques to ensure the availability of payload for information hiding purposes.

Next, the perceptual video quality of the video manipulated by CUSize is, in general, lower than that of Coeff, especially at lower bitrates. As bitrate increases, the quality attained by both the CUSize and Coeff techniques are similar. These observations are also applicable to the combined technique, where the distortion is mainly caused by CUSize. These results also suggest the bitrate from which the performance of the HEVC encoder starts to saturate for purposes of information hiding for a given video / resolution. For example, in the case of *FourPeople*, when the bitrate is greater than 1 Mbps, both CUSize and Coeff (as well as the combined technique) are equally viable for information hiding. Similar, these results may also suggest the bitrate at which the performance of the HEVC encoder starts to saturate for encoding purposes, i.e., determining the maximum bitrate for a given video / resolution, and the research in this direction shall be carried out as the future work.

The graphs in Fig. 3.5 suggest that by implementing the combined technique at higher bitrate, PSNR decreases with a magnitude of $<$ 3dB, while the perceptual quality of all video sequences are maintained as suggested by the SSIM results. From the perspective of bitrate, to achieve the PSNR of 44dB, the proposed combined technique requires an additional 7.9% and 8.3% of bitrates in *BasketballDrill* and *FourPeople*, respectively, when compared to their original counterparts (i.e., compressed videos). These results suggest that the proposed technique has negligible impact on the bitrate when considering the amount of payload that can be embedded.

All in all, the video produced by the combined technique is of slightly lower quality than that by Coeff embedding itself. However, the quality improves when the bitrate increases. Naturally, the payload in the combined technique is higher than each individual technique. Therefore, the combined technique can be considered to achieve higher payload, with acceptable perceptual quality. For coding complexity, based on the infor-

mation to be embedded, the encoder evaluates only the selected size of CUs instead of every possible size of CU, which reduces the encoding time up to 20% in cases of higher bitrates (e.g., *BasketballDrill* at 1.25 Mbps).

3.6 Summary

An information hiding technique was proposed to insert external information in HEVC compressed video. This technique encoded information by manipulating the size of CU. In addition, the proposed technique was combined with odd-even embedding using non-zero coefficients belonging to selected ranges of value. The ranges, in turn, depended on the size of CU to achieve similar perceptual quality as the original video. Simulation results suggested that the proposed CUSize technique maintained the perceptual quality of the video for higher bitrate scenarios and improved the conventional odd-even embedding in terms of payload. As bitrate increases, the contribution by CUSize manipulation became negligible in terms of quality degradation and capacity. However, in the case of lower bitrates, CUSize offered minimal payloads at the expense of quality degradation.

CHAPTER 4 : VIDEO AUTHENTICATION SCHEME

4.1 Overview

In this chapter, an authentication scheme is presented by utilizing the proposed information hiding technique in Chapter 3. The architecture overview of the authentication scheme is first presented, and the individual processes are detailed in the following sub chapters. The processes include tag generation, tag implementation, tag alteration and tag verification. Then, results and analysis are reported in terms of video quality, robustness against forgery, sensitivity, computational cost and functional comparison with other authentication schemes. Lastly, a summary is given to conclude the proposed authentication scheme chapter.

4.2 Introduction

As a result of wide deployment of digital media streaming, various applications emerged for the purposes of video content viewing and recording. Nowadays, various user-friendly tools are available for video content manipulation (e.g., trimming, cropping, recompression) and powerful hardware at affordable prices ensure their viabilities. Hence, a video needs to be authenticated so that its source can be confirmed to be someone trustworthy and its content can be verified to be genuine prior to consumption or broadcasting (Atrey et al., 2009).

The viability of digital video as evidence in the judicial process has been largely unprecedented, but there is an increasing number of videos from closed-circuit television (CCTV) being released in social media corresponding to incidents, e.g., house breaking, shoplifting. Based on the study, digital evidence is often ruled inadmissible by courts because it is owned without authentication or its authenticity cannot be verified (Casey, 2011). According to the Association of Chief Police Officers (ACPO) guideline, it is

necessary to demonstrate how evidence is authenticated and to show the integrity of each process through which the evidence was obtained (Williams, 2012). The evidence should be preserved from any third party who is able to repeat the same process and attain the same result as that presented to the court. Consequently, it is crucial to implement a secure authentication scheme for confirming the authenticity of viable video evidence and preventing any digital video designed for causing benefiting or hatred a certain party.

Based on the study in Chapter 2.5, a video authentication scheme has to be designed to protect the confidentiality of authentication code (hereinafter referred to as tag) from being manipulated, verify the integrity of video against content tampering and localize the manipulated region if any content tampering is detected. These features can be achieved by combining the concept of cryptography based (i.e., Chapter 2.5.1), content based (i.e., Chapter 2.5.2) and watermarking based (i.e., Chapter 2.5.4) authentication.

A thorough literature survey shows that there is no authentication scheme specifically designed under the HEVC coding structure. Therefore, a multi-layer authentication scheme is put forward for HEVC compressed video. In this scheme, the combination of CU sizes, which is unique to HEVC and sensitive to video manipulation, is considered along with other elements in the HEVC coding standard to generate the tag. Temporal dependency was enforced, where the tag generated in one slice is embedded into its subsequent slice. By design, the tag is repeatedly but selectively embedded using various elements in a HEVC video, including non-zero DCTs coefficients, QPs parameter values, and prediction modes, depending on the bit segment in the generated tag.

The proposed scheme offers three layers of authentication to detect and localize the tampered regions in a HEVC video, as well as verifying the source / sender of the video using a shared secret key. In the experiment, video sequences from various classes (resolutions) are considered to verify the performance of the proposed multi-layer authentication scheme. Results show that, at the expense of slight degradation in perceptual quality,

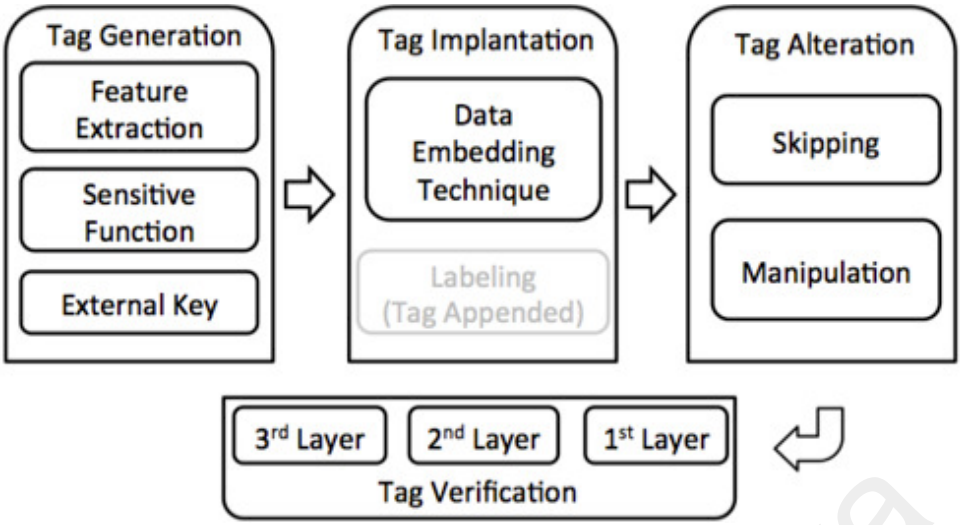


Figure 4.1: Architecture overview of multi-layer authentication scheme

the proposed scheme is robust against several common attacks. Moreover, a functional comparison is performed between the proposed multi-layer authentication scheme and the conventional schemes.

4.3 Authentication Scheme Design

The proposed scheme aims to detect and localize the tampered region(s) in a HEVC compressed video by means of information hiding and the dependency in the temporal axis. Specifically, the tag is generated and embedded into the video. Fig. 4.1 shows the architecture overview of the proposed multi-layer authentication scheme, which consists of the following four processes: Tag Generation, Tag Implantation, Tag Alteration and Tag Verification. In tag generation, the extracted features from the video and the secret key are combined then fed into a hash function as detailed in Chapter 4.3.1. The tag implantation process using information hiding technology is detailed in Chapter 4.3.2. Tag alteration by means of masking or skipping, as well as the embedding schemes is described in Chapter 4.3.3. To validate a video, the tag is verified in three different layers of authentication as detailed in Chapter 4.3.4.

Figure 4.2 shows the operation of proposed multi-layer authentication scheme during the encoding process. In the previous video slice (e.g., S_{n-1}), the video features are

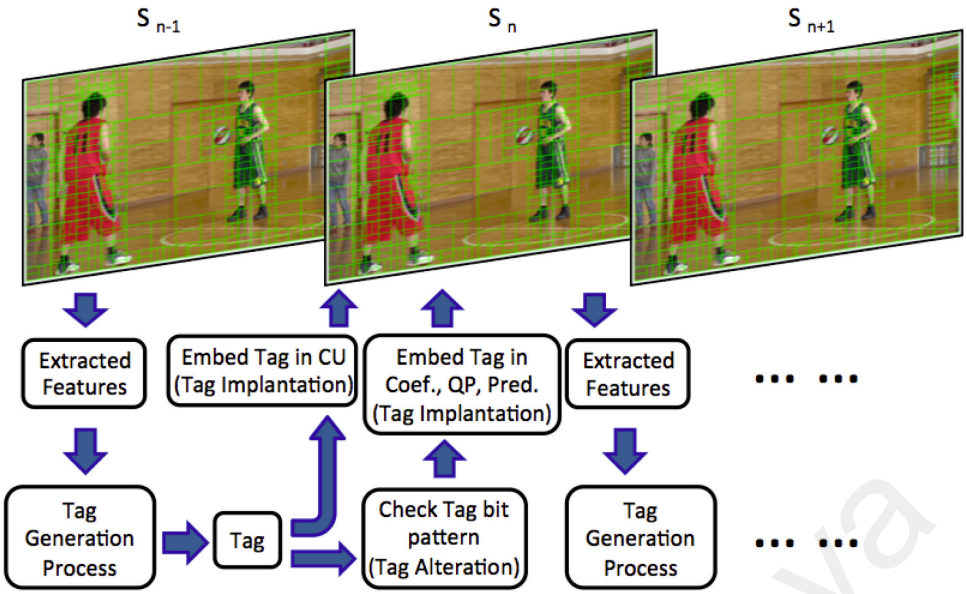


Figure 4.2: Operation of multi-layer authentication scheme in video encoder

extracted as input to the tag generation process. The generated tag is embedded in the current video slice (e.g., S_n) by utilizing the size of CU, as detailed in Chapter 4.3.2.1. The same generated tag is utilized for determining the location and value to be embedded in the selected coefficients, QPs and prediction type of CU in the current video slice based on the bit pattern of tag, as detailed in Chapter 4.3.3. After that, video features are extracted from the current slice, i.e., the modified video slices with embedded tag, and utilized as an input to the tag generation process. Then, the newly generated tag is embedded it into the next video slice (e.g., S_{n+1}) by modifying its video structure. This sequence of operations is repeated until the end of the video sequence which creates firm content dependency among the video slices in the temporal axis.

4.3.1 Tag Generation

In video authentication, a generated tag must be unique as well as sensitive to its input, and its genuineness must be verifiable by anyone who has the secret key. To fulfill these requirements, the unique statistical features of the video content and a hash function (e.g., SHA-2) are exploited to generate the tag, which is in turn embedded into the video.

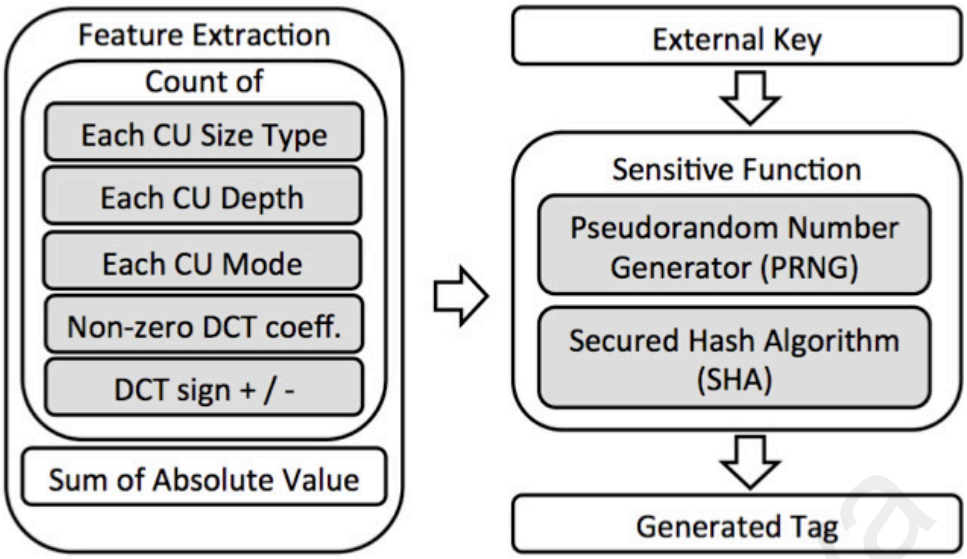


Figure 4.3: Tag generation

4.3.1.1 Feature Extraction

Several video features are considered to serve as the input for tag generation. These features, including the size types, depths and modes in every CU, as well as non-zero DCTs coefficient values, are extracted from each CTU in every video slice. Recall from Chapter 2.3.3 that in every video slice, HEVC divides each CTU into some combination of CUs in different sizes. To facilitate the discussion, let $\gamma_m \in \{2N \times 2N, 2N \times N, N \times 2N, N \times N, 2N \times nU, 2N \times nD, nL \times 2N, nR \times 2N\}$ refer to the category of CU size, $\delta_m \in \{0,1,2,3\}$ refer to the depth of quad tree decomposition, and $\pi_m \in \{\text{intra, inter}\}$ refer to the prediction mode in the m-th CTU, where $m \in \{1,2,\ldots M\}$ for

$$M = \lceil (width/64) \rceil * \lceil (height/64) \rceil. \tag{4.1}$$

The frequency of occurrences for γ_m, δ_m and π_m in the m-th CTU are computed and referred to as $F(\gamma_m)$, $F(\delta_m)$ and $F(\pi_m)$, respectively.

Here, the number of 4×4 pixel blocks is considered, i.e., the number of pixels is divided by 16. Suppose the 3-rd CTU is being processed (i.e., $m = 3$). Given one CU of size 32×32, when $\delta_3 = 1$, the corresponding frequency of occurrences is $F(\gamma_3 =$

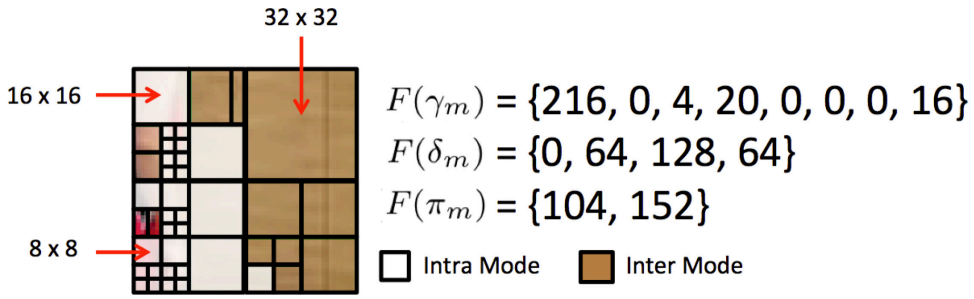


Figure 4.4: Illustration of feature extraction

$2N \times 2N) = F(\delta_3 = 1) = 64$ since there are exactly 64 units of 4×4 pixel block within it. Similarly, for a CU of size 16×16 with intra mode, the frequencies of occurrence $F(\pi_3 = \text{intra}) = 16$ since there are exactly 16 units of 4×4 pixels within it. For further illustration, frequencies of occurrences for $F(\gamma_m)$, $F(\delta_m)$ and $F(\pi_m)$ are calculated based on the example given in Fig. 4.4. Here, $F(\gamma_m = 2N \times 2N) = 216$ since there are 1 CU of size 32×32 (i.e., 64 units of 4×4), 7 CUs of size 16×16 (i.e., 112 units of 4×4) and 10 CUs of size 8×8 (i.e., 40 units of 4×4). On the other hand, $F(\pi_m = \text{intra}) = 104$ since there are 104 units of 4×4 block coded in intra mode while $F(\pi_m = \text{inter}) = 152$ because there are 152 units of 4×4 block coded in inter mode.

For features extraction, let $\gamma_m^{\max_1}$ and $\gamma_m^{\max_2}$ be the two most frequently occurring CU categories in the m-th CTU. The difference in frequency of occurrences between them, denoted by $\Gamma(m)$, is computed as $\Gamma(m) = F(\gamma_m^{\max_1}) - F(\gamma_m^{\max_2})$. Similarly, let $\delta_m^{\max_1}$ and $\delta_m^{\max_2}$ be the two most frequently occurring depths in the m-th CTU, and the difference in frequency, denoted by $\Delta(m)$, is computed as $\Delta(m) = F(\delta_m^{\max_1}) - F(\delta_m^{\max_1})$. Similarly, for prediction mode, the difference between frequency of using intra and inter in the m-th CTU, denoted by $\Pi(m)$, is computed as $\Pi(m) = |F(\pi = \text{intra}, m) - F(\pi = \text{inter}, m)|$. In addition, in each CTU, the count of non-zero DCTs coefficient $cnz(m)$, the sum of absolute value of non-zero DCTs coefficient $sav(m)$, and the difference between the frequency of occurrences for positive and negative signs $s(m)$ are computed. Note that these entities highly sensitive to re-compression and only available in the HEVC standard (i.e., γ_m and

δ_m), which will change drastically when encoded in different bitrate or when different content is encoded.

4.3.1.2 Secret Key

The (secret) key K with a specific length is required to verify the origin (i.e., sender) of a video. This key must be owned by both parties (i.e., sender and receiver) to generate the same tag for verification purpose. Note that the secret key is not revealed during verification because only the generated tag is compared against the embedded tag. In case the origin of the video need not be verified, the secret key can be conveniently replaced by any value such as DCTs coefficient values, motion vectors, etc.

4.3.1.3 Sensitive Function

A sensitive function (e.g., hash function, pseudo-random number generator) is required to generate a unique tag from the extracted features and shared secret key K. The tag generated by this function should differ significantly even when the inputs (e.g., statistics of video) are similar but not identical. It should be practically impossible to analyze this tag for inverting the mapping process, that is, to obtain the input value from the tag. Here, a cryptographic hash function H, namely, SHA-2 (NIST, 2002), is utilized to meet the aforementioned requirements. For each video slice, the extracted feature values, viz., $\Gamma(m)$, $\Delta(m)$, $\Pi(m)$, $cnz(m)$, $sav(m)$, $s(m)$ for all CTUs (M in total) as well as the shared secret key K are concatenated to form the input for the hash function H for generating the tag w. The hash function H, input and output w are related as expressed in Eq. (4.2), where $\theta||\zeta$ concatenates θ and ζ together. Note that the length of the tag x depends on the applied hash function and in the proposed scheme, the output tag is 32 bytes since SHA-256 (NIST, 2002) is considered.

$$w = H\left((\Gamma(m))_{m=1}^{M}||(\Delta(m))_{m=1}^{M}||(\Pi(m))_{m=1}^{M}||(cnz(m))_{m=1}^{M}||(sav(m))_{m=1}^{M}||(s(m))_{m=1}^{M}||K\right) \tag{4.2}$$

4.3.2 Tag Implantation

In this chapter, the generated tag in Chapter 4.3.1 is embedded into the HEVC compressed video by utilizing four information hiding techniques. These techniques are deployed to achieve high imperceptibility and reliability by utilizing four HEVC video elements (i.e., CU size, non-zero DCTs coefficient, QPs and prediction type). In the following sub chapters, each element is described in detail to selectively and repeatedly embed the generated tag. For CU size, the tag is directly embedded repeatedly within a video slice. On the other hand, for non-zero DCTs coefficient, QPs and prediction type, the tag is embedded based on the bit pattern of tag, which will be further described in Chapter 4.3.3.

4.3.2.1 Coding Unit Size

In HEVC encoder, the RDO decides the CU sizes to achieve the best compression ratio based on the desired bitrate. In the proposed multi-layer authentication scheme, instead of using the size determined by RDO, the size of CUs in slice S_{s+1} is forced to embed the tag w, which is computed from the previous slice S_s based on a predefined mapping rule. An example of the mapping rule is shown in Fig. 3.1.

The CU sizes are divided into two categories, where one encodes '0' and the other encodes '1'. In particular, category '0' includes $2N \times N$, $2N \times nU$, $nL \times 2N$, and $N \times N$ pixels, while category '1' includes $N \times 2N$, $2N \times nD$, $nR \times 2N$, and $2N \times 2N$. In other words, the CU size in S_{s+1} can be $N \times 2N$, $2N \times nD$, $nR \times 2N$, or $2N \times 2N$ for $w_l =$ '1' where $l = 1, 2, \cdots, 256$, and the rest of the cases are for $w_l =$ '0', as depicted in Fig. 3.1.

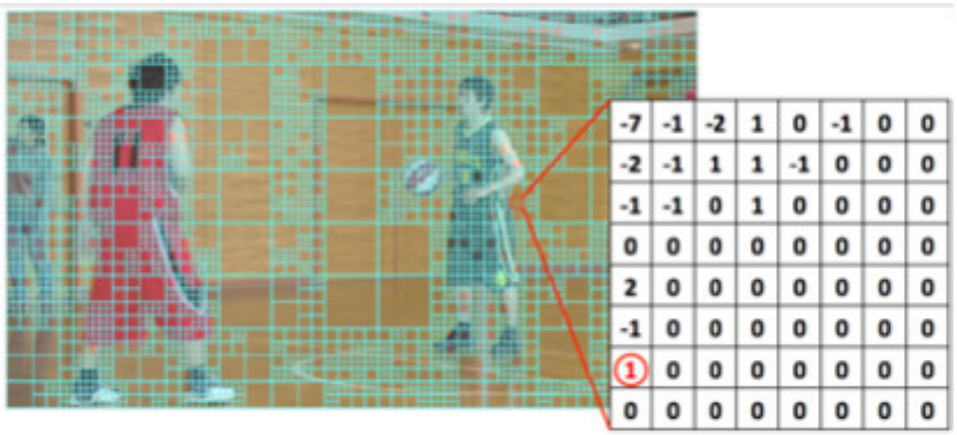


Figure 4.5: Modified LSB of the non-zero DCT coefficient

For instance, if the CU size decided by RDO is 16×8 and $w_l =$ '1', then the proposed scheme will force the RDO to recalculate the required bitrate (i.e., cost) for 8×16, 16×16, and two AMP's (i.e., $2N \times nD$, $nR \times 2N$), then choose the CU size that results in the lowest cost. For CU with larger size (e.g., 32×32), it is justifiable to encode it by using some combination of blocks with smaller sizes (e.g., two 32×16, four 16×16, etc.), because in this case, a smooth block is merely decomposed into combination of smaller blocks, which are conventionally considered for encoding region of higher spatial activity. The tag w is repeatedly and selectively embedded following the order from top-left to bottom right (i.e., Z-scanning) as in the HEVC structure (ISO, 2013). This approach maintains the video quality at the expense of slight increment in bitstream size. It should be noted that, smaller blocks are not combined into a larger block to maintain the video quality at the expense of slight file size increment.

4.3.2.2 Non-Zero DCTs Coefficient

Here, LSB of non-zero DCTs coefficients is utilized to embed the tag without causing significant quality degradation. To minimize distortion, the last non-zero DCTs coefficient (with respect to the scanning order in use) of each CU is chosen in every CTU, as shown in Fig. 4.5. The selected DCTs coefficient c is modified to an even integer for embedding $w_l = 0$, and vice versa.

Figure 4.6: Quantization value for each CTU in a slice

4.3.2.3 Quantization Parameter

During encoding, RDO utilizes the QPs value to achieve the desired bitrate. In order words, it determines the quality of video, where smaller QPs value leads to higher video quality, and vice versa. HEVC encodes each CTU with different QPs value based on the predefined QPs value range as stipulated in the configuration file. For instance, if QPs and MaxDeltaQP are defined as 28 and 2, respectively in the configuration file, the range of QPs value will be $[26,30]$. Here, the QPs of each CTU is forced to embed w_l by modifying the offset (i.e., MaxDeltaQP) range during the encoding process. The RDO calculates the cost of each CTU (i.e., total bit requires to code the CTU) based on the QPs value with the selected offset. Odd QPs values will be utilized in the calculation when embedding $w_l = 1$, and vice versa. Fig. 4.6 shows the possible QPs values for embedding $w = 11111000\cdots$.

4.3.2.4 Prediction Type

Video compression is closely tied with the implementation of various prediction methods, which can be coarsely divided into two approaches: prediction within the video slice itself (intra) and among few neighboring slices (inter). Two approaches of prediction

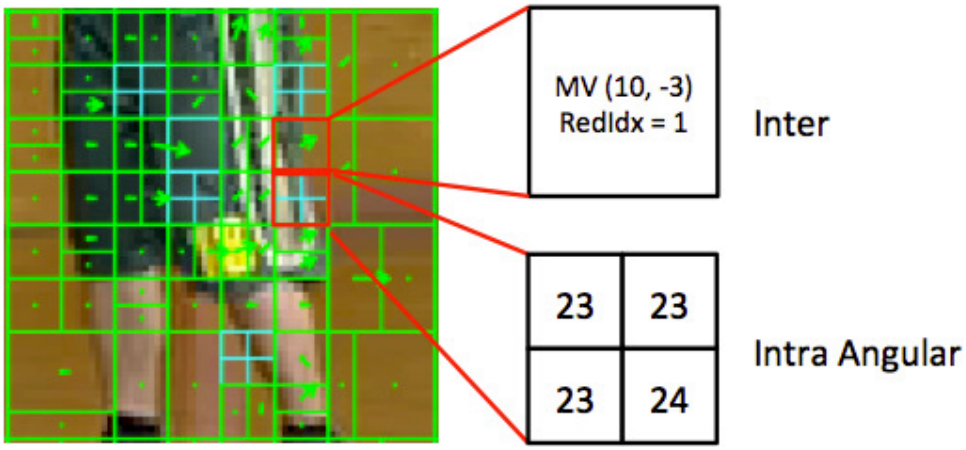


Figure 4.7: Intra and inter prediction mode decision in a slice

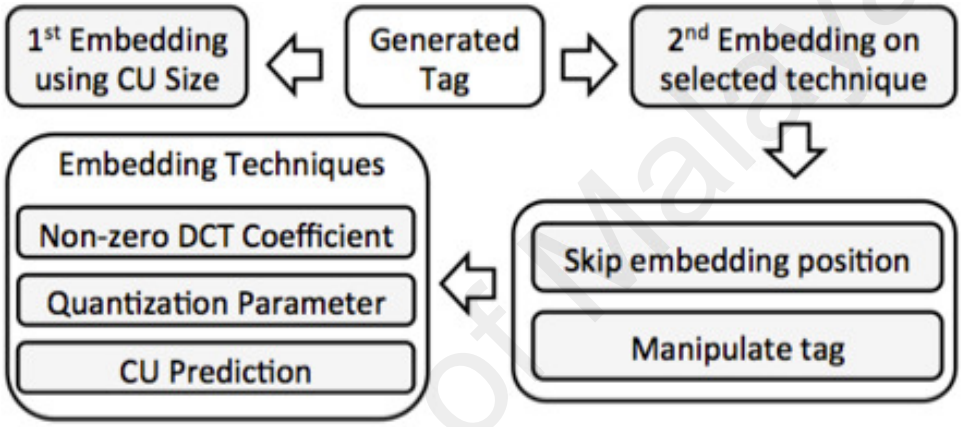


Figure 4.8: Tag implantation and alteration process

are exploited to represent the tag $w_l \in \{1,0\}$. Again, RDO is set to consider only the CTU cost for all 34 types of intra prediction (see Chapter 2.3.3) while ignoring those for inter prediction when $w_l = 0$. On the other hand, only the costs for inter prediction are considered when $w_l = 1$. Figure 4.7 shows the selected CU to embed $w_l \in \{0,1\}$ in inter prediction mode with Motion Vector (MV) $= (10,-3)$ using RedIdx (i.e., reference slice index) = 1, and intra prediction mode using mode 23 and 24.

4.3.3 Tag Alteration

In the proposed multi-layer authentication scheme, the first slice S_1 of the video is utilized for generating the tag w. This tag is conveyed to the next slice S_2 via two embedding steps. The first step utilizes the CU size embedding technique detailed in Chapter 3 to embed the tag by modifying the CU size in S_{n+1}. The second step embeds the tag by

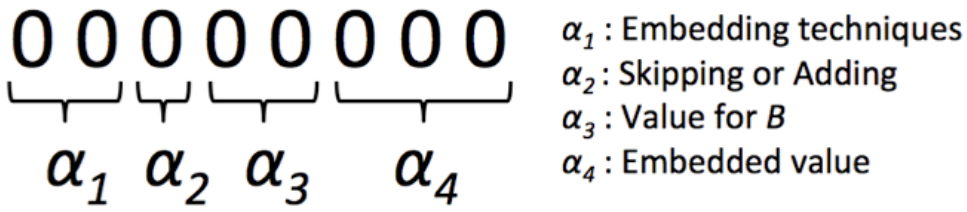


Figure 4.9: Bit segment of every byte in tag

using three other embedding techniques by considering the bit segment in each byte of the tag. Figure 4.8 shows the aforementioned embedding steps. The purposes of having two embedding processes are to: (a) enable a quick way to check the authenticity of a given video, and (b) localize the tampered regions, with precision up to the CU size. Note that these processes can be performed without using the secret key K.

The tag is divided into non-overlapping segments where each segment is processed and embedded one at a time. As an illustration, Fig. 4.9 shows an 8-bit segment of the tag, which will be processed by the second embedding process. Specifically, the second embedding process determines the technique to be applied (for embedding), the skipping of positions, and the manipulation on (i.e., masking) the tag itself. These processes are included to complicate the act of mimicking.

4.3.3.1 Selection

Given a segment, the first two bits (denoted by α_1) determine the embedding technique to deploy. All four possible combinations are listed in Table 4.1. Specifically, when $\alpha_1 = 01$, α_4 is embedded into the last non-zero DCTs coefficient of the next three CUs in the next CTU(s). In the case of $\alpha_1 = 10$, α_4 is embedded into the QPs of the next three CTUs. For $\alpha_1 = 11$, prediction mode for the next three CUs are utilized to encode α_4. For $\alpha_1 = 00$, no embedding takes place.

Table 4.1: Syntax of bit pattern in every byte of tag

Bits	α_1: Embedding Technique	α_2: Mode	α_3: Value
00	No embedding	Skipping	0
01	Coefficient	Adding	1
10	QPs	-	2
11	Prediction mode	-	3

4.3.3.2 Manipulation

The embedding process is complicated to discourage mimicking of the tag by skipping selected embedding locations (synchronization) or adding the value α_3 to the bit segment α_4 (masking) prior to actual information hiding. For instance, when $\alpha_2 = 0$, based on the decided embedding technique (signaled by α_1), position of the selected locations (e.g., non-zero DCTs coefficient when $\alpha_1 = 01$) is skipped for α_3 times, then α_4 is embedded into the $(\alpha_3 + 1)$-th position by using the selected technique α_1. For $\alpha_2 = 1$, α_4 is added to α_3 before being embedded into the selected location. Due to the problem of overflow, $\mod(\alpha_4 + \alpha_3, 2)$ will be embedded. The embedding process continues until all bit segments in the tag w are processed.

The pseudo-code of the proposed multi-layer authentication scheme is presented as Algorithm 1, which includes the tag generation, implantation and alteration processes in the HEVCs encoder. The tag is generated in the n-th slice, i.e., S_n, and embedded into S_{n+1}.

4.3.4 Tag Verification

The embedded tag is verified during video decoding. Three layers of authentication are achieved in the proposed multi-layer authentication scheme, namely: the conveniently applicable layer without the need of the secret key (first layer); the dedicated layer to localize tampered region (second layer), and; the sophisticated layer which extracts the video features for computation of the hash value to verify origin of video (third layer). Algorithm 2 shows the extraction and verification of tag during decoding. Here, v_1, v_2

Input: K
Output: w

initialization ; $m \leftarrow 0$; $n \leftarrow 0$; $i \leftarrow 0$; $j \leftarrow 0$;
repeat
  $\Gamma(m) \leftarrow 0$; $\Delta(m) \leftarrow 0$; $\Pi(m) \leftarrow 0$;
  if $n \neq 0$ **then**
    Embed w via CU size embedding technique ;
    foreach *byte* (w_i) *in* w **do**
      switch α_1 *of* w_i **do**
        case 1 **do** : set as prediction modes embedding ;
        case 2 **do** : set as CTU QPs value embedding ;
        case 3 **do** : set as non-zero DCTs coeff. embedding ;
        otherwise do : not embedding ;
      end
      if $\alpha_2 = 0$ **then**
        Skip α_3 time(s) on selected embedding technique ;
      else
        $\alpha_4 = \alpha_4 + \alpha_3$;
      end
      Embed α_4 using α_1 technique;
    end
  end
  foreach *CTU in* S_n **do**
    $m = m + 1$;
    foreach *coefficient*(c_j) *in m-th CTU* **do**
      if $c_j \neq 0$ **then**
        $cnz(m) \leftarrow cnz(m) + 1$;
        $sav(m) \leftarrow sav(m) + |c_j|$;
        if $c_j > 0$ **then**
          $s_{+,m} \leftarrow s_{+,m} + 1$;
        else
          $s_{-,m} \leftarrow s_{-,m} + 1$;
        end
      end
    end
    foreach 4×4 *pixels in m-th CTU* **do**
      check CU sizes and add count on $\{F(\gamma_m)\}$;
      check CU depths and add count on $\{F(\delta_m)\}$;
      check CU modes and add count on $\{F(\pi_m)\}$;
    end
    $\Gamma_m^{\max_1}, \Gamma_m^{\max_2} \leftarrow$ max and second max of $\{F(\gamma_m)\}$;
    $\Delta_m^{\max_1}, \Delta_m^{\max_2} \leftarrow$ max and second max of $\{F(\delta_m)\}$;
    $\Gamma(m) \leftarrow |\Gamma_m^{\max_1} - \Gamma_m^{\max_2}|$;
    $\Delta(m) \leftarrow |\Delta_m^{\max_1} - \Delta_m^{\max_2}|$;
    $\Pi(m) \leftarrow |\Pi_{intra,m} - \Pi_{inter,m}|$;
    $s(m) \leftarrow |s_{+,m} - s_{-,m}|$;
  end
  $w \leftarrow H((\Gamma(m))_{m=1}^{M} || (\Delta(m))_{m=1}^{M} || (\Pi(m))_{m=1}^{M} || (cnz(m))_{m=1}^{M} || (sav(m))_{m=1}^{M} || (s(m))_{m=1}^{M} || K)$;
  $m \leftarrow 0$;
  $n \leftarrow n + 1$;
until *end of slices*;

Algorithm 1: Pseudo-code for Tag Generation, w

and v_3 show the first, second and third layer authentication statuses, respectively. Status 1 indicates the video is authenticated, and status 0 indicates a failed authentication in that particular slice.

Input: K
Output: v_1, v_2, v_3
initialization ; $n \leftarrow 0$, $w \leftarrow 0$, $w'_i \leftarrow 0$;
repeat
  if $n \neq 0$ **then**
    $w'_0 \leftarrow 0$;
    foreach *x bytes of CU in* S_n **do**
      if $w'_0 = 0$ **then**
        $w'_0 \leftarrow x$ bytes of w based on in Fig. 3.1 ;
      else
        $w'_i \leftarrow x$ bytes of w based on in Fig. 3.1 ;
        if $w'_i = w'_0$ **then** $v_1 \leftarrow 1$;
        else $v_1 \leftarrow 0$;
      end
    end
    foreach w'_i **do**
      check α_1 in w'_i ;
      apply tag alteration based on α_2,α_3 in w'_i ;
      if $\alpha'_4 = \alpha_4$ **then**
        $v_2 \leftarrow 1$
      else $v_2 \leftarrow 0$;
    end
    if $w'_i = w$ **then** $v_3 \leftarrow 1$;
    else $v_3 \leftarrow 0$;
  end
  Algorithm 1 step 21 - 47 to obtain w;
  $n \leftarrow n+1$;
until *end of slices*;

Algorithm 2: Pseudo-code for Tag Verification, w'

4.3.4.1 First Layer of Authentication

Recall that the tag is selectively and repeatedly embedded using CUs sizes in each slice. Therefore, the first layer of authentication checks for uniformity of the embedded tag throughout the slice under consideration. The embedded tag can be extracted during the decoding process by examining the CU size based on Fig. 3.1. The tags are extracted following the Z-scanning order in every CTU in a slice (ISO, 2013). The first instance of the tag (32 bytes in length) is extracted and stored as w^0, while the following instances within the same slice are stored as w^i for $i = 1, 2, \cdots$. If $\exists i$ such that $w_l^i \neq w_l^{i+1}$ at any bit location l, then the video is termed *tampered*. Specifically, the group of CTUs encoding

the instance of the tag that differs from the majority are marked as the tampered group. On the other hand, when $w_l^i = w_l^{i+1}$ for all i and all l, the video is authenticated with respect to the first layer.

4.3.4.2 Second Layer Authentication

Since the first layer of authentication depends only on the CU sizes, it is possible that some elements such as coefficients and QPs are tampered, while maintaining the CU sizes. Therefore, the second layer is invoked to further verify the video at the byte-level of the extracted tag. Specifically, the last non-zero DCTs coefficient in a CU, the CU prediction mode or the QPs in the previous slice is considered, depending on the value α_1 extracted from the tag. Then, α_2, α_3 and α_4 are also obtained from the extracted byte segment. Next, the derived α_4 from the previous slice is compared with the embedded α_4' based on the embedding technique as stipulated by α_1. When $\alpha_4' \neq \alpha_4$, it implies that tampering occurs at the region(s) under investigation. Note that when $\alpha_1 = 0$, no verification is performed because the tag is not embedded into any coefficient, QPs or prediction mode, for that particular byte segment of the tag.

4.3.4.3 Third Layer Authentication

When a video passes the first and second layers of authentication, the video is merely verified to be neither modified nor tampered, but its source (i.e., sender) is not verified. To verify the video source, the same secret key K (supplied during the encoding process) is required (see Chapter 4.3.1.2). Specifically, the values $\Gamma(m)$, $\Delta(m)$, $\Pi(m)$, $cnz(m)$, $sav(m)$ and $sm(m)$ in S_n are computed to generate w (see Eq. 4.2). Next, the source of the video can be verified by comparing the generated tag w against the embedded tag w^i. In case the tags match (i.e., $w = w^i$), the video is authenticated to be originating from a known (reliable) source, otherwise the source cannot be verified and hence the video cannot be trusted.

Original (compressed) video Processed video

Figure 4.10: Illustration of the 8-th slice of the test video - *BasketballPass*

4.4 Experiment Result

The HM16.0 reference software model is modified to implement the proposed multi-layer authentication scheme. Video in Class A (2560×1600), B (1920×1080), C (832×480), D (416×240), E (1280×720) and F (1024×768) are utilized as the test video sequences. Three profiles, namely, Random Access (RA), Low Delay P (LP), and Low Delay B (LB), consisting of P-/B-slices are selected to collect results using QPs in the range of $[8, 48]$. RA profile is defined by a sequence of one I-slice followed by eight B-slices, LP profile consists of a sequence of one I-slice followed by four P-slices and LB profile is same with LP profile by replacing B-slices with P-slices. The results are recorded in Table 4.2.

4.4.1 Video Quality

The results in Table 4.2 indicate that both the original and processed videos exhibit similar and steady growth in image quality when QP decreases. For video encoded with small QPs (such as those in the range of $[8, 48]$), the degradation in quality with respect to SSIM index (Z. Wang et al., 2004) is hardly noticeable in all video classes. However, in terms of PSNR, the video quality drops, on average, < 1 dB for QPs in the range of $[8, 48]$ for all video classes considered.

Table 4.2: Results of embedding tags in various classes of standard test video

Class (Test Video)	QP	RA				LP				LB			
		Original		Authenticated		Original		Authenticated		Original		Authenticated	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Class A (*PeopleOnStreet*)	8	50.5264	0.9999	50.3907	0.9999	51.7031	0.9999	51.6017	0.9999	51.6246	0.9999	51.5039	0.9999
	16	44.0813	0.9997	44.0174	0.9997	45.0852	0.9998	44.9942	0.9998	45.1223	0.9998	45.0277	0.9998
	24	38.9427	0.9985	38.8355	0.9985	39.3133	0.9986	39.2417	0.9986	39.4622	0.9987	39.3945	0.9987
	32	34.2105	0.9941	34.0721	0.9940	32.2255	0.9936	34.1066	0.9936	34.3092	0.9936	34.2095	0.9935
	40	29.7969	0.9799	29.6221	0.9795	29.7152	0.9786	29.5799	0.9786	29.7694	0.9783	29.6609	0.9783
	48	25.3706	0.9399	25.2184	0.9388	25.0522	0.9334	24.9315	0.9345	25.0255	0.9317	24.8976	0.9318
Class B (*Tennis*)	8	49.8782	0.9996	49.7813	0.9996	51.2741	0.9997	51.1930	0.9997	51.2695	0.9997	51.1816	0.9997
	16	44.2705	0.9979	44.2204	0.9978	44.5771	0.9982	44.5325	0.9982	44.6120	0.9982	44.5604	0.9982
	24	41.7317	0.9939	41.6404	0.9937	41.8567	0.9942	41.7730	0.9940	41.9911	0.9944	41.9232	0.9943
	32	38.2451	0.9799	38.1230	0.9791	38.3561	0.9804	38.2316	0.9798	38.5466	0.9813	38.4612	0.9808
	40	34.5212	0.9454	34.4460	0.9438	34.6043	0.9671	36.3681	0.9660	34.8057	0.9489	34.7805	0.9484
	48	30.3875	0.8689	30.4880	0.8705	30.5435	0.8707	30.6879	0.8720	30.6313	0.8733	30.8077	0.8766
Class C (*PartyScene*)	8	50.1873	0.9977	50.0968	0.9977	51.7037	0.9983	51.6155	0.9983	51.6403	0.9983	51.5349	0.9983
	16	42.4209	0.9925	42.3561	0.9924	43.2778	0.9929	43.2236	0.9928	43.2807	0.9930	43.2236	0.9930
	24	36.8693	0.9799	36.7610	0.9795	36.6196	0.9781	36.5374	0.9778	36.7842	0.9785	36.7004	0.9782
	32	31.6608	0.9495	31.4896	0.9483	31.0041	0.9428	30.9262	0.9420	31.0513	0.9431	30.9694	0.9422
	40	27.0375	0.8898	26.8283	0.8860	26.0257	0.8728	25.9443	0.8703	26.0159	0.8714	25.9227	0.8691
	48	22.9707	0.7619	22.8111	0.7549	22.2403	0.7402	22.2189	0.7385	22.1968	0.7371	22.2086	0.7364
Class D (*BasketballPass*)	8	50.3374	0.9954	50.2384	0.9952	50.8117	0.9956	50.7380	0.9955	50.9240	0.9958	50.8318	0.9957
	16	44.9885	0.9876	44.8839	0.9873	45.0952	0.9868	45.0297	0.9866	45.2152	0.9872	45.1418	0.9870
	24	39.1638	0.9629	39.0220	0.9616	39.2302	0.9600	39.1576	0.9594	39.3147	0.9604	39.2333	0.9597
	32	33.5886	0.8986	33.4273	0.8954	33.4390	0.8894	33.3368	0.8874	33.4943	0.8899	33.3970	0.8881
	40	29.2736	0.7943	29.1272	0.7889	28.9012	0.7785	28.8067	0.7765	28.9669	0.7804	28.8853	0.7781
	48	25.5622	0.6816	25.5205	0.6789	25.0662	0.6642	25.0752	0.6639	25.0087	0.6666	25.1259	0.6683
Class E (*FourPeople*)	8	50.0288	0.9994	49.9057	0.9994	51.2102	0.9996	51.1023	0.9996	51.1498	0.9996	51.0191	0.9996
	16	44.6663	0.9974	44.5989	0.9974	45.0461	0.9977	44.9736	0.9977	45.1308	0.9977	45.0451	0.9977
	24	41.9733	0.9956	41.8807	0.9955	41.5085	0.9951	41.4503	0.9951	41.6532	0.9951	41.5962	0.9951
	32	38.3121	0.9896	38.1473	0.9890	37.5055	0.9873	37.4642	0.9873	37.5704	0.9873	37.5439	0.9873
	40	33.6861	0.9679	33.4551	0.9655	32.7283	0.9597	32.7010	0.9599	32.7436	0.9594	32.7497	0.9596
	48	28.8121	0.8989	28.5426	0.8914	27.9325	0.8772	27.9864	0.8788	27.9577	0.8774	28.0315	0.8801
Class F (*ChinaSpeed*)	8	52.5190	0.9996	52.3205	0.9996	53.0841	0.9996	52.9650	0.9996	53.1060	0.9996	52.9716	0.9996
	16	47.1133	0.9983	46.9471	0.9983	47.5657	0.9986	47.4487	0.9985	47.6250	0.9986	47.4983	0.9985
	24	41.1222	0.9925	40.8611	0.9922	41.4770	0.9938	41.3542	0.9936	41.5332	0.9938	41.4090	0.9937
	32	35.0748	0.9717	34.7306	0.9708	35.3295	0.9736	35.1068	0.9729	35.3495	0.9739	35.1260	0.9732
	40	29.9592	0.9362	29.6215	0.9339	29.8973	0.9322	29.7294	0.9306	29.9264	0.9335	29.7613	0.9328
	48	25.6314	0.8690	25.1712	0.8625	25.1290	0.8478	25.1002	0.8481	25.1377	0.8509	25.1218	0.8524

To further examine the results, Fig. 4.11 and 4.12 show the rate distortion curve for video sequences in Class A, B, C, D, E and F in RA, LP and LB. Each graph is featured with a magnified region to show the detailed PSNR vs Bitrate performance between the original and processed video. In Class A, quality of the processed video drops ~ 0.5 dB when considering the same bitrate (e.g., at 45 kbps, original and processed videos yield ~ 41.5 and ~ 41.0 dB, respectively). In other words, the processed video requires extra ~ 5 kbps (e.g., at 41.5 dB, the original and processed bitrates are ~ 40 and ~ 45 kbps, respectively) to achieve the same quality as the original (compressed) video. Similar performances are observed in Class B (drop by ~ 0.25 dB), Class C (drop by ~ 1.5 dB), Class D (drop by ~ 1.0 dB), Class E (drop by ~ 0.25 dB) and Class F (drop by ~ 1.2 dB).

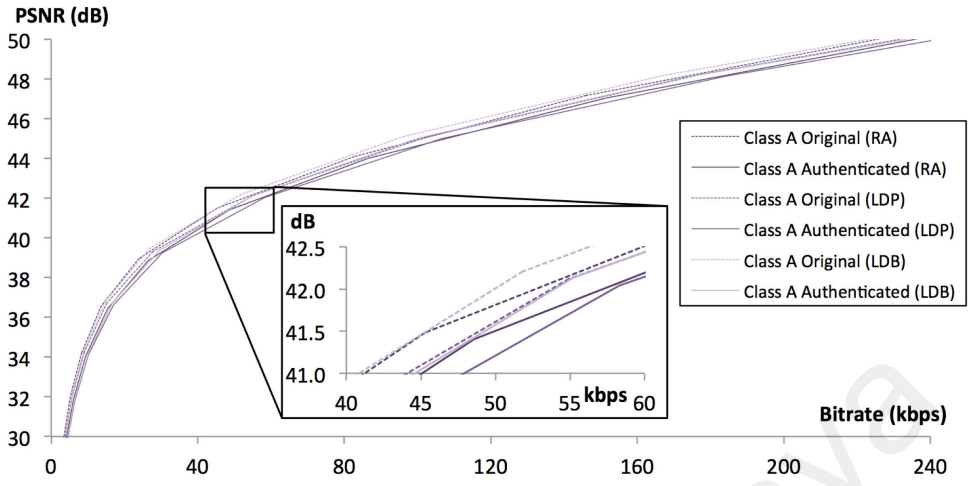
Overall, the quality of the processed videos degrade by $< 1\%$ in terms of both SSIM and PSNR when compared to their original compressed counterparts. Perceptually, both original and processed videos appear to be identical by visual inspection. As a representative example, the 8-th slice from the original and processed video of Class D are shown in Fig. 4.10, which appear to be identical.

4.5 Discussion

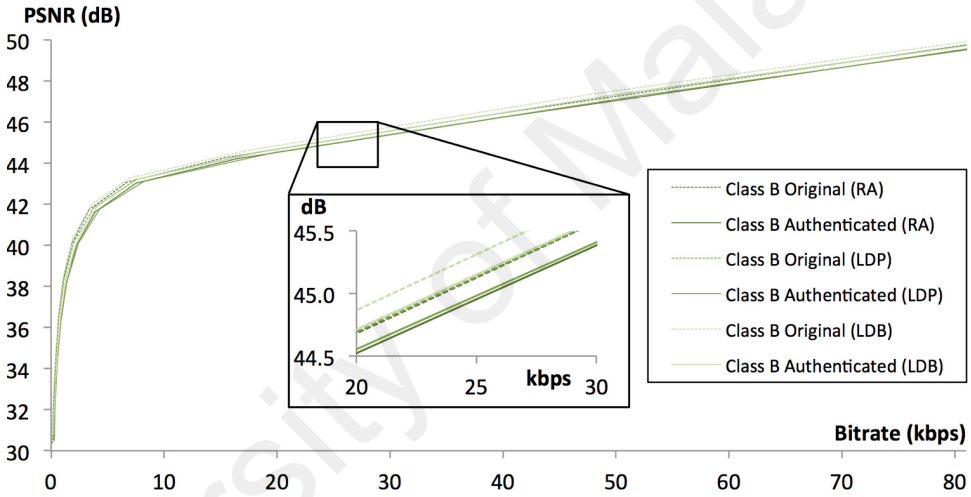
In this section, robustness and sensitivity of proposed scheme, computational cost of scheme and comparison with conventional schemes are discussed.

4.5.1 Robustness against forgery

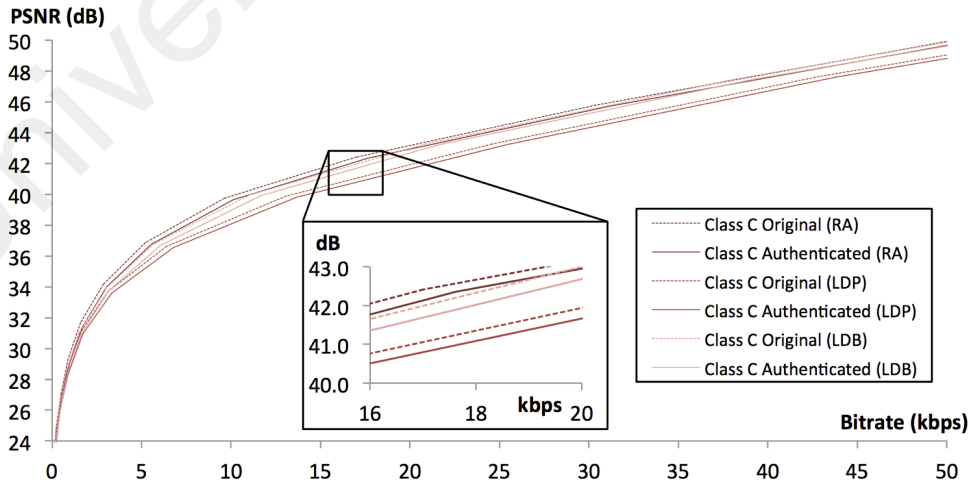
The robustness of the proposed multi-layer authentication scheme is verified by considering the following attacks: slice dropping, CU replacement, generic and Vector Quantization (VQ) attack.



(a) Class A

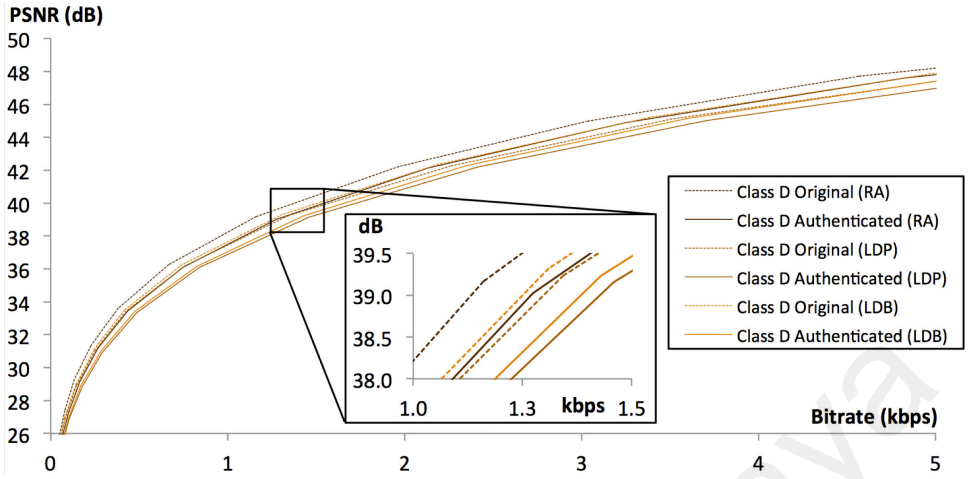


(b) Class B

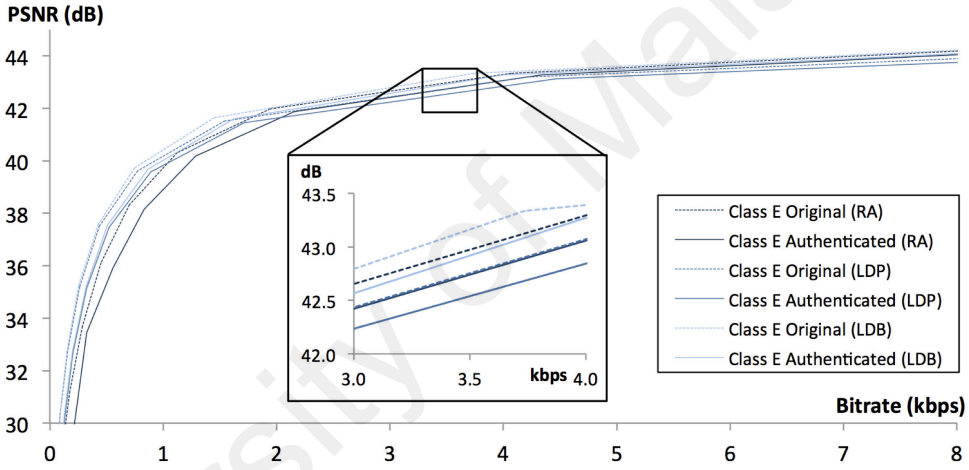


(c) Class C

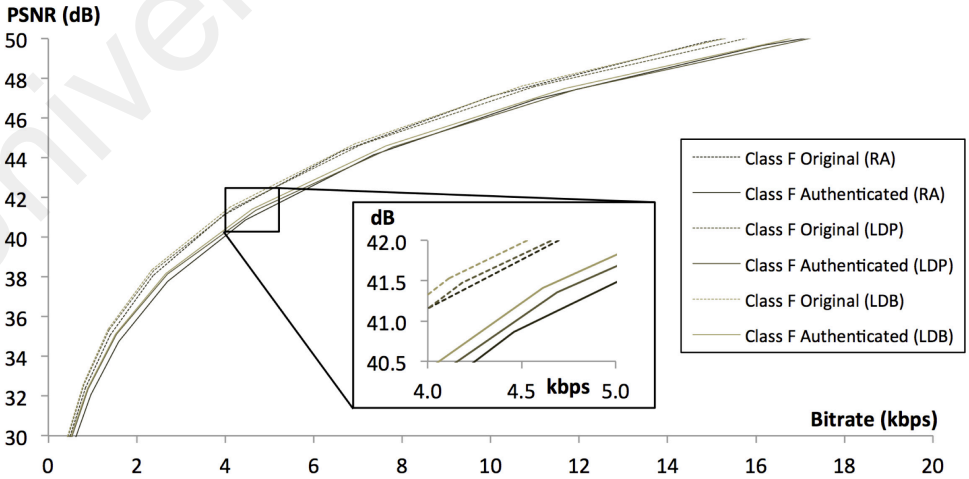
Figure 4.11: PSNR vs Bitrate performance for original and processed Class A, B and C video



(a) Class D



(b) Class E



(c) Class F

Figure 4.12: PSNR vs Bitrate performance for original and processed Class D, E and F video

4.5.1.1 Slice Dropping

During video transmission, video content are transmitted slice by slice. If any slice (e.g., S_n) is accidentally dropped or intentionally removed, then the following slice (i.e., S_{n+1}) will be authenticated in first and second layers, but not the third layer due to the dependency between two consecutive slices, where features from S_n are required to generate the tag for verification in S_{n+1}.

4.5.1.2 CU Replacement Attack

By replacing one of the CU contents with that of any other CU of the same size, the tampered slice will still be authenticated by the first layer. It is possible to change the content of a CU (e.g., coefficient, QPs) while maintaining its size, which represents one bit of the embedded tag. However, in the second layer of authentication, the replaced CU content will be examined by extracting the embedded α_4' in the last non-zero coefficient, QPs or prediction mode, depending on α_1. If $\alpha_4' \neq \alpha_4$, the mismatched CU can be identified and utilized to pinpoint the modified CU as well as the CTU involved.

4.5.1.3 Generic Attack

Lo et al. detailed a generic attack on tagged video stream by exploiting the coefficients of CU (Lo et al., 2014). According to Lo et al., information hiding in LSB of coefficients, sign of coefficients and count of zero/non-zero coefficients are potentially attacked by changing the coefficients that are not involved in the authentication process so that the modified/tampered video will be authenticated at the decoder. However, this modification is infeasible under the proposed authentication scheme due to the unpredictable location of coefficients utilized for tag embedding. Recall that the tag is repeatedly embedded into selected non-zero coefficients, where non-zero coefficients are skipped in a non-regular manner as illustrated in Fig. 4.9 and Table 4.1. Hence, any discrepancies among multiple copies of the embedded information will be detected by the second layer of authentication

in the proposed scheme (see Chapter 4.3.4.2).

In addition, due to the large number of possible combinations of CUs as well as other considered entities in HEVCs video (including non-zero coefficients count and sign), although in theory other video content may have the same coarse features, it will be unlikely that these visually different contents (but producing the same tag) would be perceptually meaningful or having unnoticeable visual distortion. In other words, the distortion caused by tampering would be obvious to the naked eyes and the distortion may further propagate to other future slices due to motion compensation. Therefore, generic attack (Lo et al., 2014) is infeasible in attacking the proposed authentication scheme.

4.5.1.4 Vector Quantization Attack

VQ is a technique designed to retrieve the embedded information based on a constructed codebook obtained from a learning process using a huge quantity of authenticated videos with the same embedded tag. When a VQ style attack (Holliman & Memon, 2000) is attempted, the proposed multi-layer authentication scheme is able to localize the attacked area. It is because the proposed scheme requires the exact sequence of CU sizes to generate and match the tag w'. Specifically, the tag w' of length 32 bytes may be copied from one part of the slice and pasted onto another part in the same slice (similar to copy-move attack in image forgery). Considering the HEVCs encoding structure, the large number of possible combinations of CU sizes in any CTU (i.e., $> 2^{256}$) suggests that this attack is practically infeasible. In other words, it may be possible to fabricate a perceptually meaningless (i.e., noise-like) video to deceive the proposed multi-layer authentication scheme, but the fabricated video can be easily identified by visual inspection or non-reference image quality assessment (Moorthy & Bovik, 2011). Hence, the proposed scheme is robust against VQ style attack.

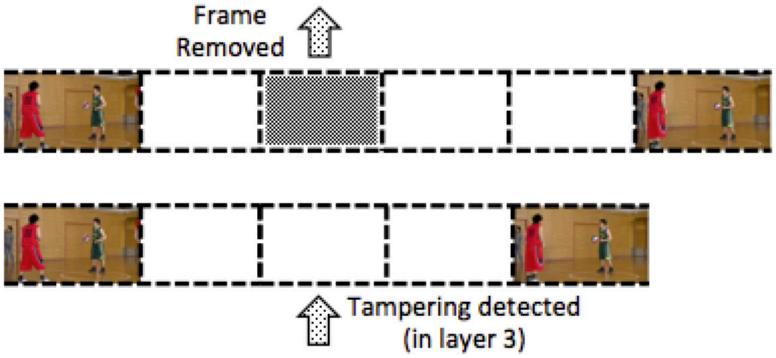


Figure 4.13: Tampering detected when a slice is removed

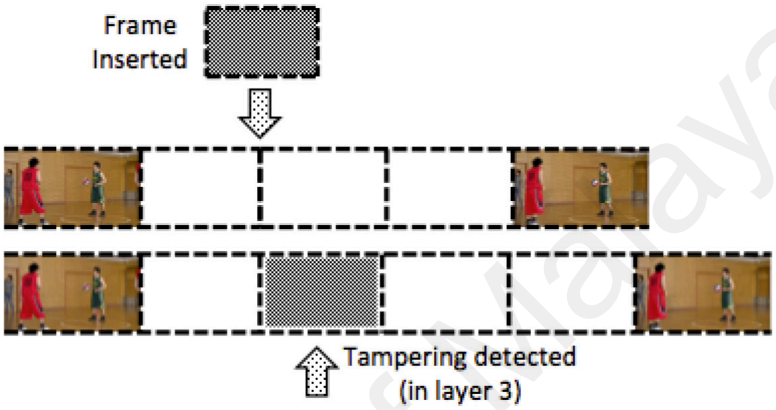


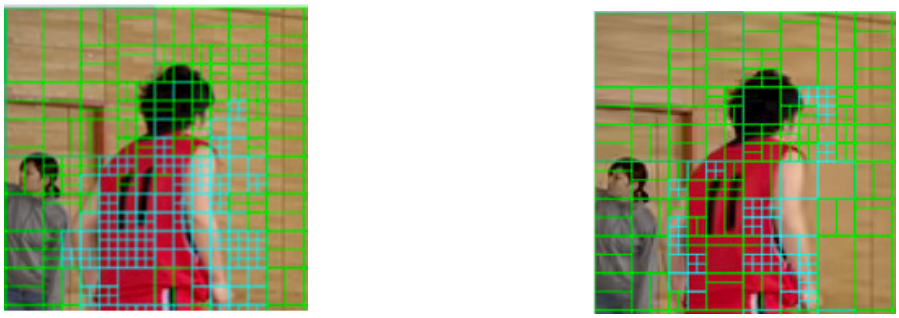
Figure 4.14: Tampering detected when a slice is inserted

4.5.2 Sensitivity

The proposed multi-layer authentication scheme is sensitive against modification. For instance, any modification in the video content (e.g., pixel values, DCTs coefficients) will lead to modified $cnz(m)$, $sav(m)$ and $s(m)$, where the third layer authentication will fail (see Chapter 4.3.4) since the correct tag w cannot be reproduced.

4.5.2.1 Slice Tampering

For tampering across slices such as slice shuffling (reordering), insertion (see Fig. 4.13) and removal (see Fig. 4.14), the tampered slice can be detected due to the dependency between adjacent slices as part of the proposed authentication design (see Chapter 4.3.4). For instance, if n slices are removed and inserted at any other position, the positions of the removed slice as well as the starting and ending of the inserted slices can be detected. More precisely, as detailed in Chapter 4.3.4, the tag generated by using the features of S_n is embedded in the subsequent slice S_{n+1}. Therefore, by checking the tag in the non-



QP=12, 145 CUs QP=32, 106 CUs

Figure 4.15: Re-compression result of 4 CTU with QP = 12 and QP = 32

tampered slice immediately after the attacked slice, the act of removal or insertion of slice can be detected. This verification process is applicable regardless of the number of slices being copied and moved, inserted, or removed.

4.5.2.2 Slice Re-compression

The embedded tags in the processed video are sensitive against re-compression (re-encoding) at different bit-rates or different QPs values. For example, Fig. 4.15(a) and (b) show the same slice compressed with QPs = 12 and 32, respectively. It is apparent that the CTU sizes are different, and hence the same tag cannot be regenerated for authentication purpose. For further illustration, Fig. 4.16 shows $\Gamma(m)$, which represents the difference in number of occurrences for two most frequently occurring CU categories in each CTU. Here, $\Gamma(m)$ for video compressed with QPs = 12 and 24 are shown. The x-axis is the CTU index throughout the entire test video sequence while the y-axis represents the value of $\Gamma(m)$. Results suggest that $\Gamma(m)$ for QPs $=$ 12 and 24 are significantly different, which confirm that the same tag cannot be regenerated. Furthermore, for slice re-sizing, cropping or rotation, the encoding process is required to generate the format compliant HEVCs video. These modifications will further eradicate the embedded tags, hence failing the authentication process and hence indicating the sign of tampering.

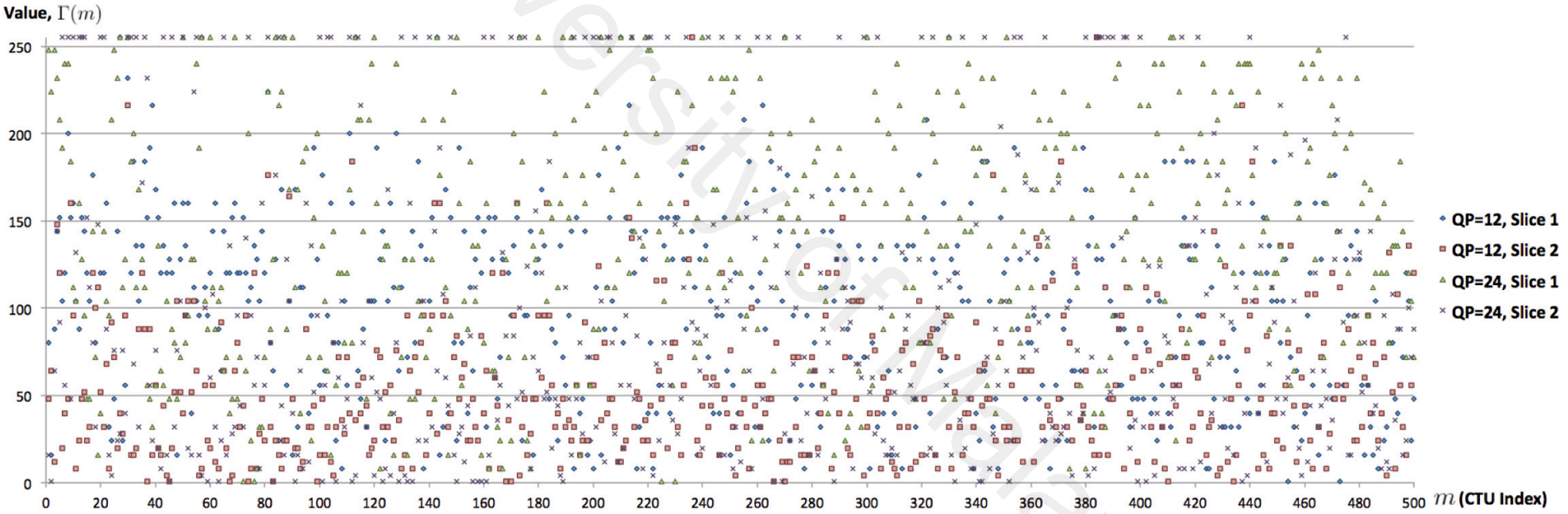


Figure 4.16: Graph of $\Gamma(m)$ vs m (CTU index) in slice 1 and 2 for Class B compressed with QP = 12 and 24

Table 4.3: Percentage of decoding time increment for original and processed video

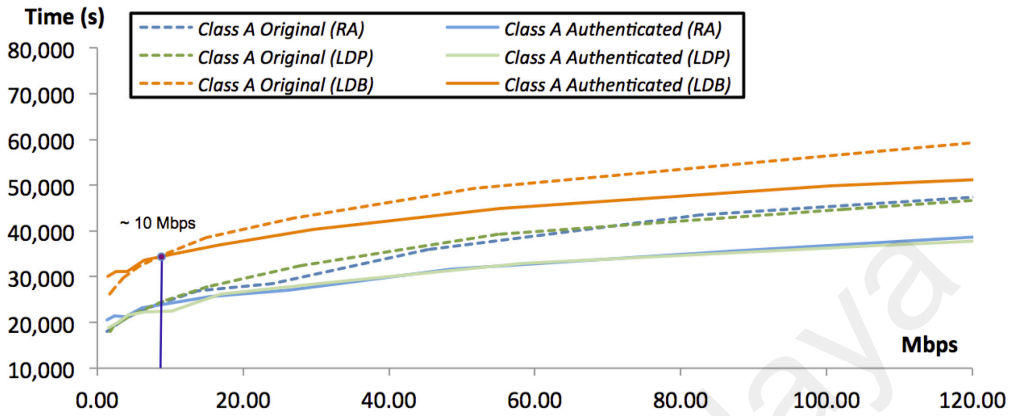
Video Class	OA vs. O				A vs. O			
	LDP		LDB		LDP		LDB	
	min.	max.	min.	max.	min.	max.	min.	max.
A	1.44	6.08	1.55	2.21	-4.27	4.82	-3.20	4.27
B	1.06	7.73	-0.60	4.51	0.03	10.77	-1.30	9,23
C	-1.30	3.44	-1.48	6.20	-6.72	2.65	-5.92	6.71
D	-2.39	9.27	-4.17	5.11	-6.23	6.74	-7.88	5.11
E	-6.18	4.24	-3.67	4.95	-9.08	5.05	-6.91	12.69
F	-4.65	-1.57	0.46	5.14	-8.04	3.18	-3.65	10.91

O = Original decoder on original video,
OA = Original decoder on processed video,
A = Modified decoder on processed video

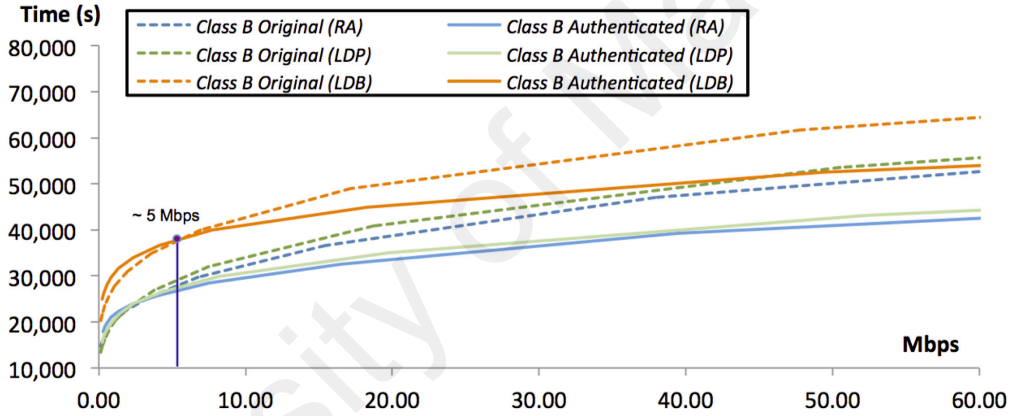
4.5.3 Computational Cost

Figure 4.17 and 4.18 show the graphs of total time needed for the original and modified encoders versus bitrate for various classes of test video sequence. It is observed that the modified encoder requires lower computational time when encoding at higher bit rates (e.g., for LB profile, $>$ 10Mbps in Class A, and $>$ 5Mbps in Class B). In particular, by utilizing the proposed tag embedding technique, RDO of the modified encoder is restricted to choose one of the 4 types of CU to embed the authentication tag as described in Chapter 4.3.2.1, in contrast to the original encoder that considers all 8 cases. Note that the time spent on computing the cryptographic function is less than the time saved by restricting the choices of CU type due to tag embedding. However, the opposite situation is observed at lower bit rates (e.g., for LB case, $<\sim$ 10Mbps in Class A, and $<\sim$ 5Mbps in Class B), where the modified encoder needs longer time when compared to the original video in all classes of video. It is because, at lower bit rate, the video sequence is encoded with CUs of larger sizes (e.g., mostly 32×32 and larger). In other words, the number of CU is reduced, and the chances to embed tag (i.e., time saving) are also reduced at lower bit rate.

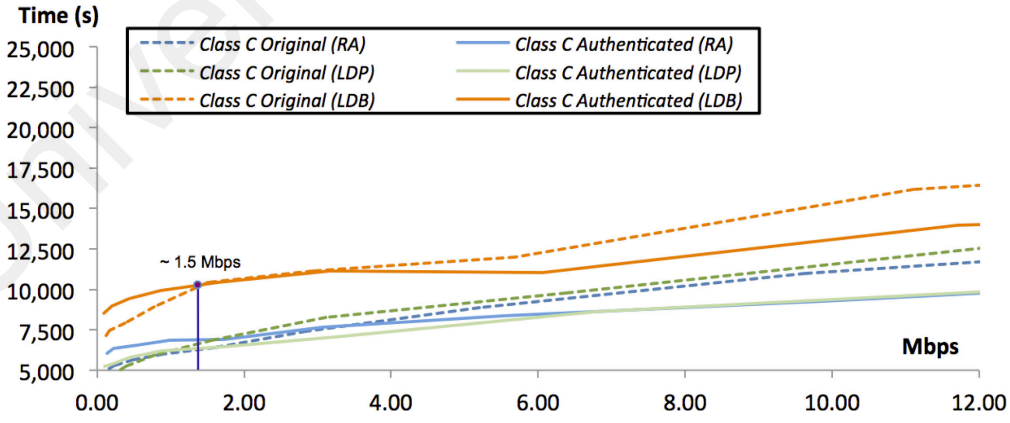
Table 4.3 shows the percentage of increment in computational time, where the maximum and minimum increment among all considered QPs values are recorded. To facili-



(a) Class A

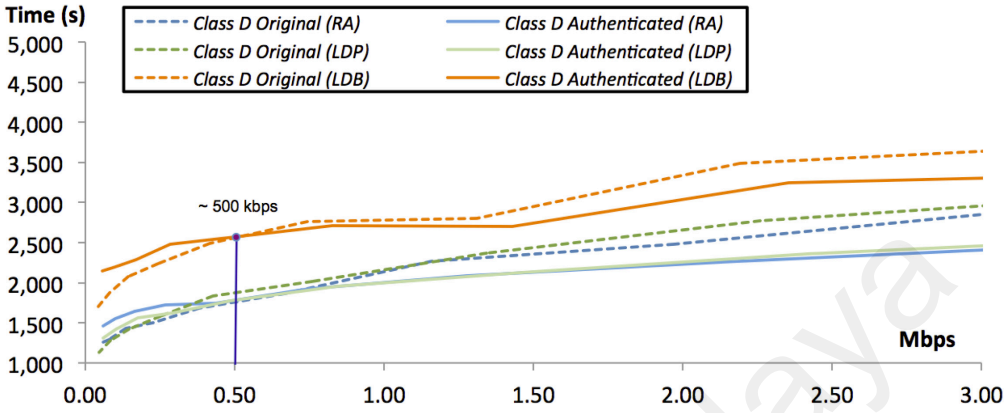


(b) Class B

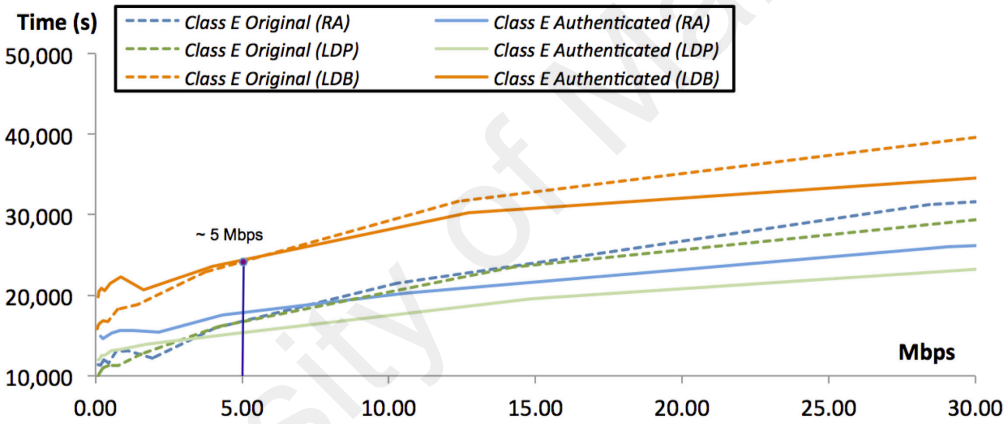


(c) Class C

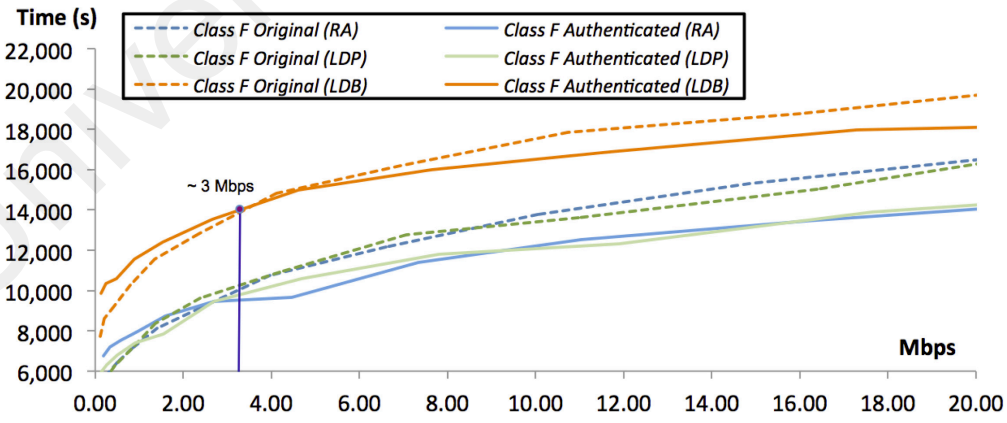
Figure 4.17: Encoding time vs bitrate for original and processed Class A, B and C video



(a) Class D



(b) Class E



(c) Class F

Figure 4.18: Encoding time vs bitrate for original and processed Class D, E, and F video

tate the presentation, let O denote the time needed to decode the original video using the original decoder, OA denote the time needed to decode the processed video (i.e., video with tag) using the original decoder, and A denote the time needed to decode and authenticate the processed video using the modified decoder. Positive percentage indicates an increment of computational time, and vice versa. Results for OA vs. O (i.e., Column 2 to 5) suggest that some of the test video sequences yield negative percentage of time increment. That is, the time needed to decode the processed video is shorter than that of the original video. Here, the decoding time is reduced due to the differences in the encoded CU structure in the original and processed videos. Specifically, this happens when a more complex CU structure is reduced to a simpler one due to tag embedding. One of the many possible scenarios is as follows: a CTU originally encoded with four 32×32 blocks is modified to be encoded by just one 64×64 block to embed the tag.

The negative percentages recorded in Column 6 to 9 of Table 4.3 suggest that the time needed for decoding and authenticating the processed video using the modified decoder is shorter than the time needed to decode the original video using the original decoder. In contrast, the opposite situations are captured by the positive percentages in Table 4.3. All in all, the proposed authentication scheme takes an additional computational time of -9.08% and $+12.69\%$ for decoding as well as authenticating the processed video in the best and worse scenarios, respectively.

4.5.4 Comparison

The conventional schemes may be applicable to all video coding standards (e.g., MPEG-2, H.264), but they are not specifically implemented on or experimented with the HEVCs coding standard, and hence it is not clear whether they are suitable for deployment on HEVCs. Based on the literature study, there is no authentication scheme specifically designed to exploit / adapt to the coding structure of HEVCs. As such, the proposed multi-

Table 4.4: Comparison Among Authentication Scheme

Function	✠ 1	✠ 2	✠ 3	✠ 4	✠ 0
Require feature extraction	2	1	1	2	2
Apply on all slices	1	1	2	1	2
Robust to slice dropping	1	1	2	2	2
Require key for verification	2	2	0	2	1
Localize tampered region	1	0	0	1	1
Source identification	2	2	2	2	2
Exploit temporal axis dependency	0	1	0	1	2

✠ 1 = (Ren & O'Gorman, 2012), ✠ 2 = (Roy et al., 2013), ✠ 3 = (Wei et al., 2014),
✠ 4 = (Upadhyay & Singh, 2011), ✠ 0 = Proposed authentication scheme,
2 = fully functional, 1 = partially functional, 0 = no function

layer authentication scheme is compared with four conventional schemes under different video standard. The first scheme is proposed by Roy et al. (Roy et al., 2013), where the authentication tags are embedded in the mid-frequency range of the non-zero DCTs coefficients through hardware implementation. However, this process is only performed for I-frame under the H.264 standard. The second scheme is proposed by Wei et al. (Wei et al., 2014), where tags are embedded into the Supplement Enhancement Information in Network Abstract Layer Unit for both the base- and enhancement-layers in H.264/SVC. The third scheme is proposed by Upadhyay et al. (Upadhyay & Singh, 2011). They utilize a non-linear classifier to compute the statistical local information (i.e., absolute difference) between every two consecutive slices and exploit this feature to determine whether a frame is tampered or genuine. The fourth scheme is proposed by Ren et al. (Ren & O'Gorman, 2012), where the digital signature architecture is considered. Local video features were calculated from slices to form a concise fingerprint sequence, which is in turns appended to the video signal for authentication purpose.

Table 4.4 functionally compares the proposed and conventional authentication schemes for compressed video using a scale from 0 to 2. Here, "2" implies that the scheme is completely in line with the function, "1" indicates that the scheme achieves part of the function and "0" signifies that the scheme does not have the function. All schemes are robust to

video frames/slices dropping but only the proposed scheme exploits the dependency of all video slices in the temporal axis, i.e., a tampered content in current slice will be detected by the following slice. Also, the proposed scheme extracts and utilizes the video features to verify the integrity of every slice without the need of the secret key K, which is only required to verify the origin of the video.

4.6 Summary

A multi-layer authentication scheme for HEVCs compressed video was put forward. The temporal dependency was enforced and exploited, where authentication tag generated based on the statistics of the current slice was embedded into the subsequent slice. The video slices were verified by three layers of authentication: first layer provided an surface verification without utilizing the shared secret key; second layer localized tampered region, if any, and; third layer verified the source / sender by comparing the hash value of the combination of the shared secret key as well as the statistics from the video against the extracted tag. Results suggested that proposed multi-layer authentication scheme generated output video with high perceptual quality. Robustness of the proposed scheme against common attacks (e.g., CU replacement, VQ attack) as well as its sensitivity against slice tampering and re-compression were analyzed and justified. The proposed scheme was also compared to the conventional video authentication schemes.

CHAPTER 5 : VIDEO ENCRYPTION SCHEME

5.1 Overview

In this chapter, several selective encryption techniques are proposed to mask the HEVC video by means of distorting its perceptual video quality. It is designed to combine with the information hiding technique presented in Chapter 3, where the embedded data is preserved before and after the decryption process.

5.2 Introduction

Video sharing has become a trend of multimedia communication, thanks to the widely available portable video recording devices such as smartphones, as well as the ease of connectivity to social media. A recent report reveals that the average time a person spent on watching online videos has increased by 12.2% in 2014 and further by 38.2% in 2015, with reference to the data in 2013 (Austin et al., 2015). Some factors leading to the growth include the increased use of smartphones due to price reduction and improved network infrastructure, particularly in developing countries.

Although video communication can be carried out conveniently nowadays, security and privacy are at risk under uncontrolled video streaming. Some of the issues include privacy infringement and illegal distribution. Video encryption can each serve as a feasible solution to secure the video stream from illegal viewing and combat piracy, respectively. Based on the study in Chapter 2.6, selective encryption technique is of lower computational cost and produces a format-compliant encrypted video. Compared to the naïve encryption technique, selective encryption technique exploits the coding structure of the video compression standard in question and encrypts only the most sensitive information.

Therefore, a video encryption scheme is proposed based on the manipulation of Sign Bins, Transform Skip Bins and Suffix Bins for the HEVC standard. These elements

are randomized to visually distort the video, at the same time, preserving the embedded information based on the information hiding technique detailed in Chapter 3. The basic performance of the proposed selective encryption techniques are evaluated in terms of perceptual inspection, outline detection and sketch attack using various classes of test video sequences. Experiment results show that the presented video encryption scheme successfully distorts the perceptual quality of the video, and maintains the perceptual quality after the decryption process. Functional comparison between the proposed video encryption scheme and the conventional video encryption scheme is then presented.

5.3 Encryption Technique

To apply the selective encryption techniques on HEVC video stream, several requirements should be fulfilled: (1) The encrypted video will not reveal any video content perceptually. (2) The embedded information should remain intact after the encryption process. Three selective encryption techniques (i.e., significant bins, transform skip bins and suffix bins) are proposed to fulfill the requirements, at the same time maintain the video quality after the decryption process.

5.3.1 Sign Bin Encryption

HEVC stores the sign of non-zero coefficients, Motion Vector Displacement (MVD) and Delta Quantization Parameter (dQP) as they are (i.e., raw and uncompressed) in the bit-stream. This makes the signs easily accessible and modifiable without impacting the format compliant requirement, while keeping the parsing overhead low.

For coefficient sign, a complete sign encryption (i.e., all signs are randomized) results in a fairly distorted video, albeit partial sign encryption can introduce sufficient distortions (Shahid & Puech, 2014). Therefore, the sign of non-zero coefficients (i.e., *coeffSigns*) of each block is randomized. Furthermore, the proposed technique only randomizes sign bits in the luminance channel since the distortion introduced by toggling

chrominance channels results in chromatic aberration, which makes the outline more noticeable. The sign of MVD (i.e., *m_iHor*, *m_iVer*) and dQP (i.e., *iDQp*) are also randomized. Thus, the proposed approach is faster as it minimizes parsing overhead and does not require any modification during the decoding process, i.e., the encrypted (ciphertext) video is format-compliant.

5.3.2 Transform Skip Bin Encryption

In HEVC encoder, the option to enable transform skip operation is configured in the picture parameter set configuration. If activated, a transform skip bin is signaled for each transform block of size 4×4 separately for each color component. The quantizer scaling operation for the coded transform coefficient levels is performed independently of transform skip application. If transform skip operation is indicated for a transform block, the inverse transform operations are omitted (Wien, 2015; Sze et al., 2014).

Practically, the transform skip flag array (i.e., *m_puhTransformSkip*) is randomized based on the hash values during the encoding process. The RDO in HEVC encoder determines the appropriate CU structure by considering the modified transform skip flags. When the transform skip flag is toggled, the RDO pursues a CU structure that differs from the originally encoded CU structure. This leads to a slight degradation in video quality, as discussed in Sec. 5.4.1.

5.3.3 Suffix Bin Encryption

The binary syntax elements with fixed-length codeword is exploited to maintain video compression efficiency and format compliance. During the entropy coding process under HEVC standard (refers to Chapter 2.3.3), coefficients and MVDs are binarized using a combination of Truncated Rice code (p bits) and Ex-Golumb code (k bits), where $p \in \{0,1,2,3,4\}$ and $k = p+1$. The suffix part of the coefficients and MVDs code can be safely encrypted without impacting the compression efficiency. In order to suppress

processing time, the last coefficient of each 8×8 CU is only considered.

Specifically, the horizontal and vertical absolute value of motion vector, i.e., *uiHorAbs* and *uiVerAbs*, is manipulated to encrypt the video content in the P- and B-slices. *uiHorAbs* and *uiVerAbs* values are selected based on an encryption key and the suffixes of these values are encrypted by changing the suffix LSB. In addition, the coefficient suffix, i.e., *escapeCodeValue* is also manipulated with the same approach to further distort the perceptual quality of every video slice. The proposed modification is simplified by manipulating only the last coefficient of the CU to suppress processing time, at the same time produce an encrypted (distorted) video stream.

Algorithm 3 shows the pseudo-code for applying the aforementioned encryption during CABAC process in HEVC encoder. A pseudorandom bit sequence, Ω is generated based on the secret key. Each of the encryption technique compares the elements (e.g., Sign Bin Encryption compares *coeffSigns*) to $\Omega(K)$, and decides to toggle the value by means of adding 1 or multiplying by -1. For decryption, receivers compare the elements with the generated $\Omega(K)$ and recover the plaintext video by manipulating the elements by invoking the same algorithm. By implementing these encryption techniques, the embedded information (i.e., CU type and size) can remain unchanged in encrypted video.

5.4 Experiment Result

The same reference software model utilized in Chapter 4 (i.e., HM16.0) is modified to implement the proposed selective encryption techniques in HEVC video. Video in Class A (2560×1600), B (1920×1080), C (832×480), D (416×240), E (1280×720) and F (1024×768) are utilized as the test video sequences. These video sequences are encoded in four profiles, namely, All Intra (AI), LB, LP and RA by using QP in the range of $[8, 48]$. AI profile is defined by a series of video slices with only encoded in intra-prediction mode. The video quality of encrypted video and decrypted video in all class are recorded

Input: K
Output: w

initialization ; $n \leftarrow 0$;
repeat
  Generate random bits, $\Omega(K)$;
  foreach *CTU in* S_n **do**
    Encode Skip Flag, Pred. Mode, Part. Size ;
    while *Encode Pred. Info.* **do**
      Encrypt MV Sign Bin ;
      if $(m\_iHor > 0) \neq \Omega(K)$ **then**
        $m\_iHor \leftarrow m\_iHor \times -1$;
      end
      if $(m\_iVer > 0) \neq \Omega(K)$ **then**
        $m\_iVer \leftarrow m\_iVer \times -1$;
      end
      Encrypt MV Suffix Bin;
      if $uiHorAbs \bmod 2 \neq \Omega(K)$ **then**
        $uiHorAbs \leftarrow uiHorAbs + 1$;
      end
      if $uiVerAbs \bmod 2 \neq \Omega(K)$ **then**
        $uiVerAbs \leftarrow uiVerAbs + 1$;
      end
    end
    EncodeIPCMInfo ;
    while *Encode Coeff.* **do**
      Encrypt Transform Skip Bin ;
      if $m\_puhTransformSkip \neq \Omega(K)$ **then**
        flip $m\_puhTransformSkip$ bit ;
      end
      Encrypt Coefficient Sign Bin ;
      if $coeffSigns \neq \Omega(K)$ **then**
        flip $coeffSigns$ bit ;
      end
      Encrypt Coefficient Suffix bin;
      if $escapeCodeValue \bmod 2 \neq \Omega(K)$ **then**
        $escapeCodeValue \leftarrow escapeCodeValue + 1$;
      end
      Encrypt dQP Sign Bin ;
      if $iDQp \neq \Omega(K)$ **then**
        flip $iDQp$ bit ;
      end
    end
  end
  $n \leftarrow n + 1$;
until *end of slices*;

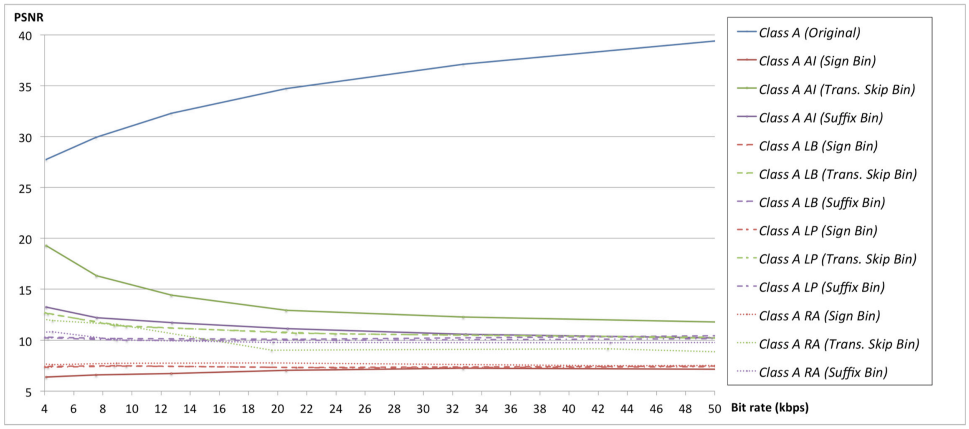
Algorithm 3: Pseudo-code for Encryption in BAC process

in Fig. 5.1, 5.2 and 5.3. In addition, for better visualizing purpose, the perceptual results are shown in Fig. 5.4 - 5.9.

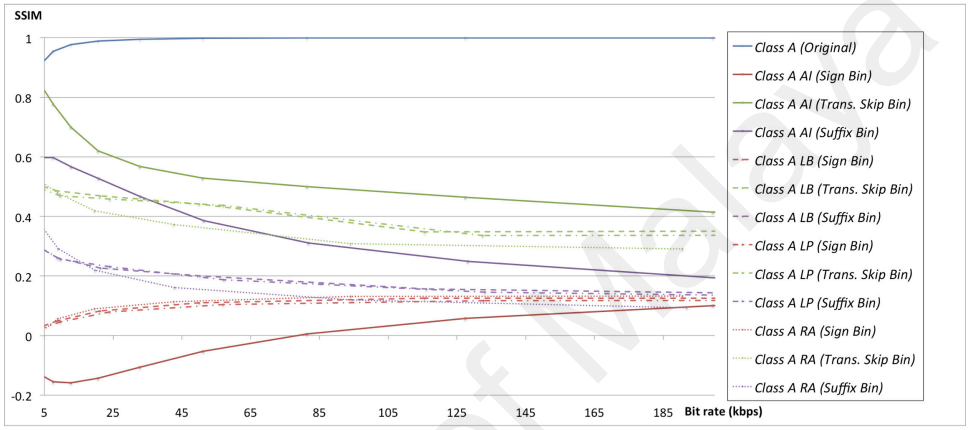
5.4.1 Video Quality

The results in Fig. 5.1- 5.3 show the rate distortion curve of PSNR and SSIM for Class A, B, C, D, E and F video sequences in AI, LB, LP and RA. Each figure includes results of original encoded video and encrypted video by utilizing three encryption techniques (i.e., Sign Bin, Transform Skip Bin and Suffix Bin). For results in PSNR (e.g., Fig. 5.1), videos (i.e., encrypted) exhibit similar quality degradation (i.e., with a range of $[10, 25]$ across all bitrates considered). However, in terms of SSIM, the encrypted video quality by utilizing Sign Bin technique always drop lower than other two techniques and the results by utilizing Transform Skip Bin indicate slower quality degradation towards higher bitrates when compared to other two techniques, except in Class F (i.e., Fig. 5.3). In other words, Transform Skip Bin technique is less effective in degrading sharp edges (i.e., Class F video comprises of scenes from video games) when compared to Sign Bin and Suffix Bin techniques.

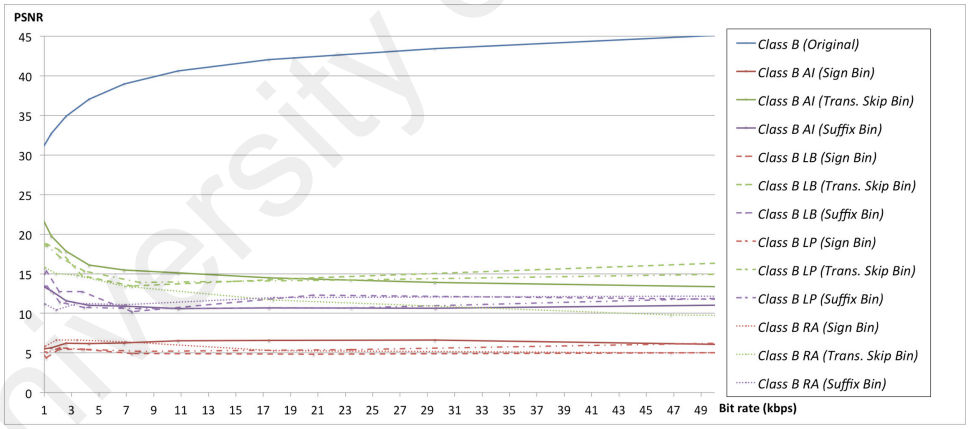
To further examine the results, Table 5.1 shows the BD-rate (Bjøntegaard, 2001) of decrypted video quality, i.e., the difference between original video and decrypted video in terms of PSNR and SSIM. Noted that encrypted video by utilizing Sign Bin technique is not considered in Table 5.1 because Sign Bin manipulation does not affect the original video quality. Results suggest that most average PSNR and SSIM differences are below zero. These values indicate that Transform Skip and Suffix Bin encryption slightly degrades the video quality. Particularly, most PSNR and SSIM rates fall in the range of $[0, 3]$. Note that for LB and RA (i.e., video profiles with B-slices), the PSNR and SSIM rates decrease slightly (e.g., PSNR rate $= -0.0291, -0.0379$ and SSIM rate $= -0.383, -0.0373$ in Class D LB and RA respectively), due to the insignificant difference between the orig-



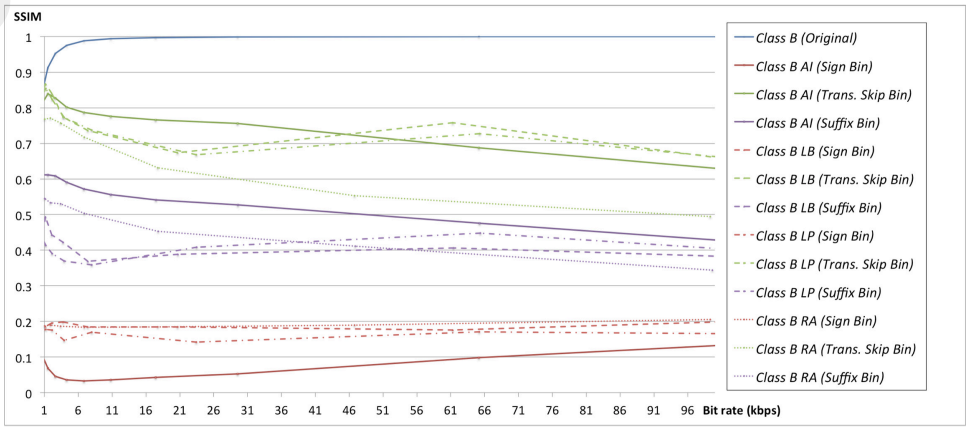
(a) Class A PSNR



(b) Class A SSIM

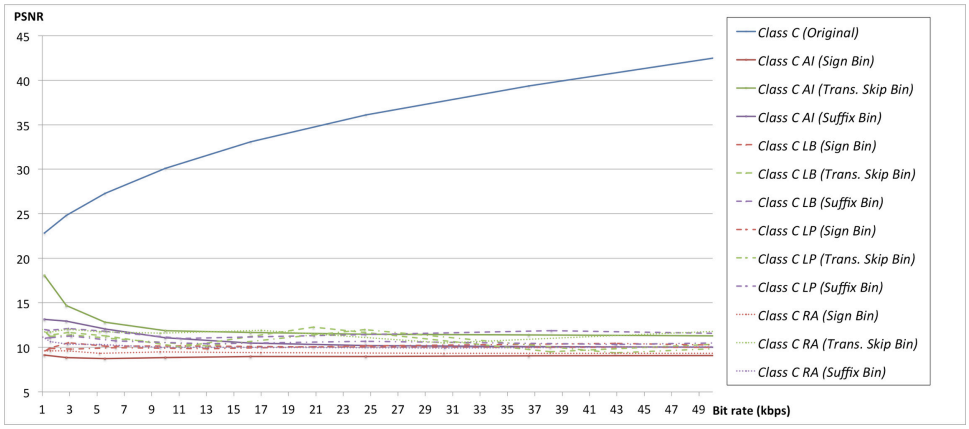


(c) Class B PSNR

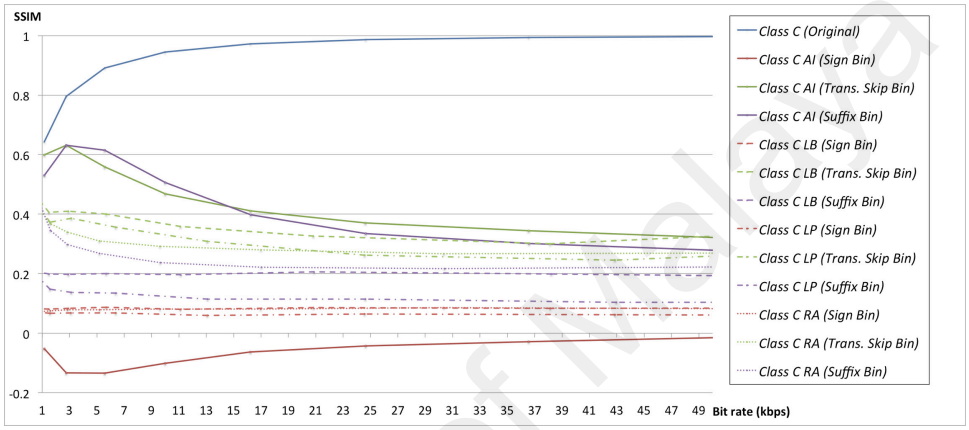


(d) Class B SSIM

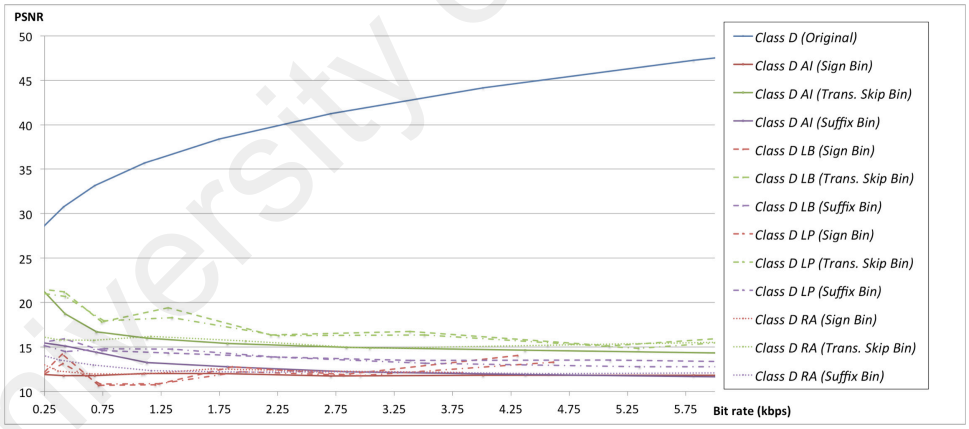
Figure 5.1: PSNR & SSIM of original encoded and encrypted Class A & B video



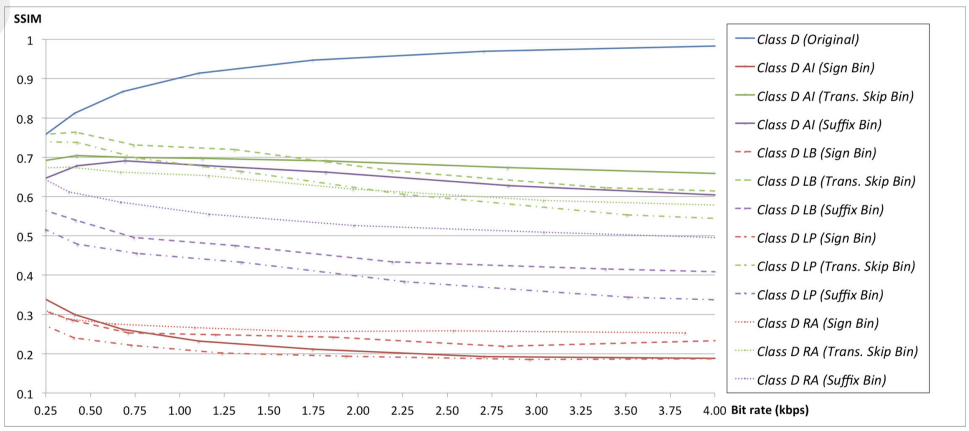
(a) Class C PSNR



(b) Class C SSIM

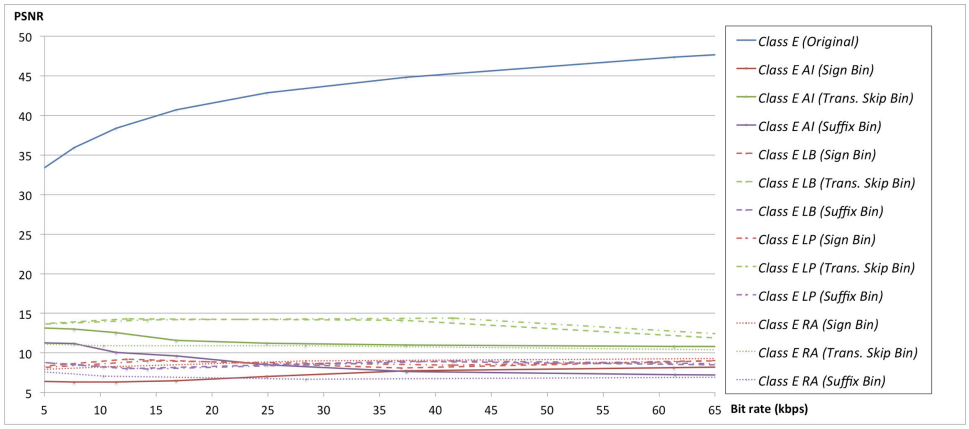


(c) Class D PSNR

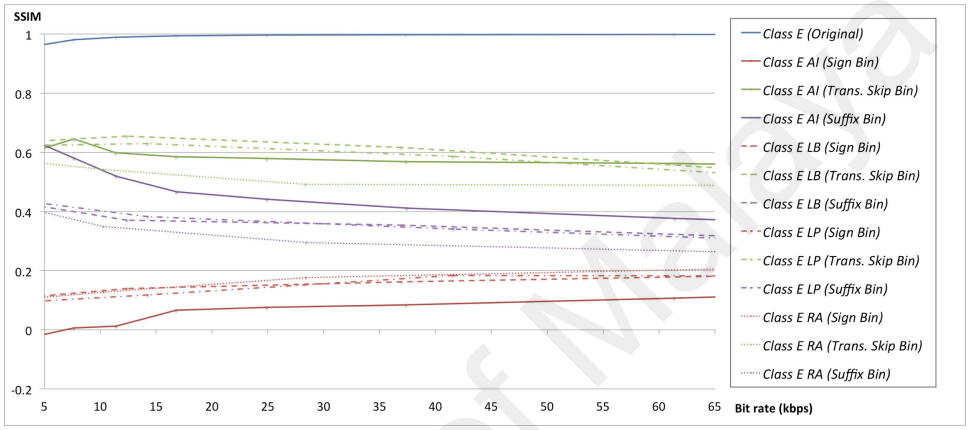


(d) Class D SSIM

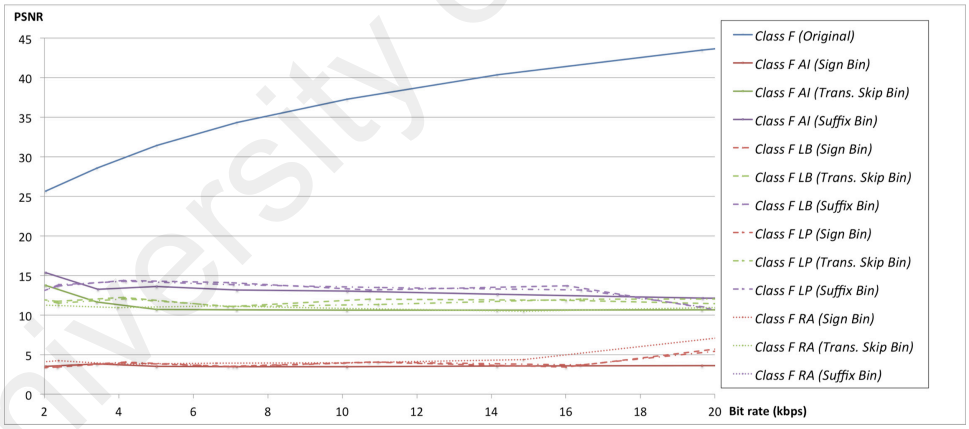
Figure 5.2: PSNR & SSIM of original encoded and encrypted Class C & D video



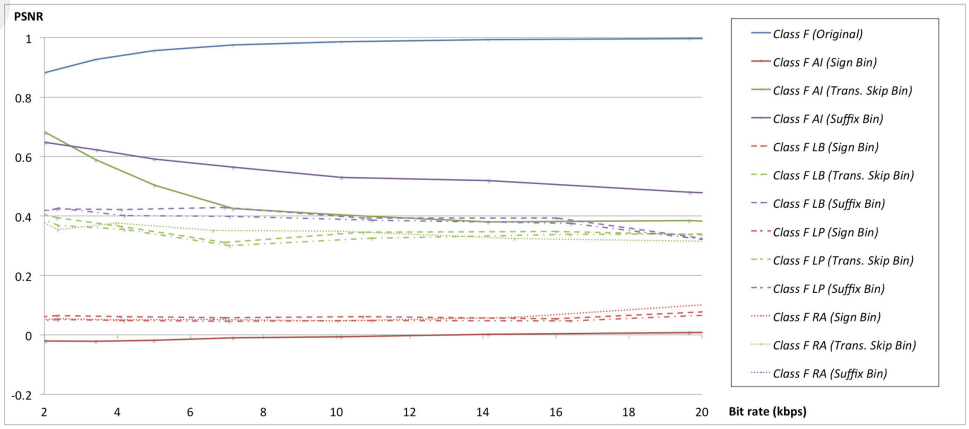
(a) Class E PSNR



(b) Class E SSIM



(c) Class F PSNR



(d) Class F SSIM

Figure 5.3: PSNR & SSIM of original encoded and encrypted Class E & F video

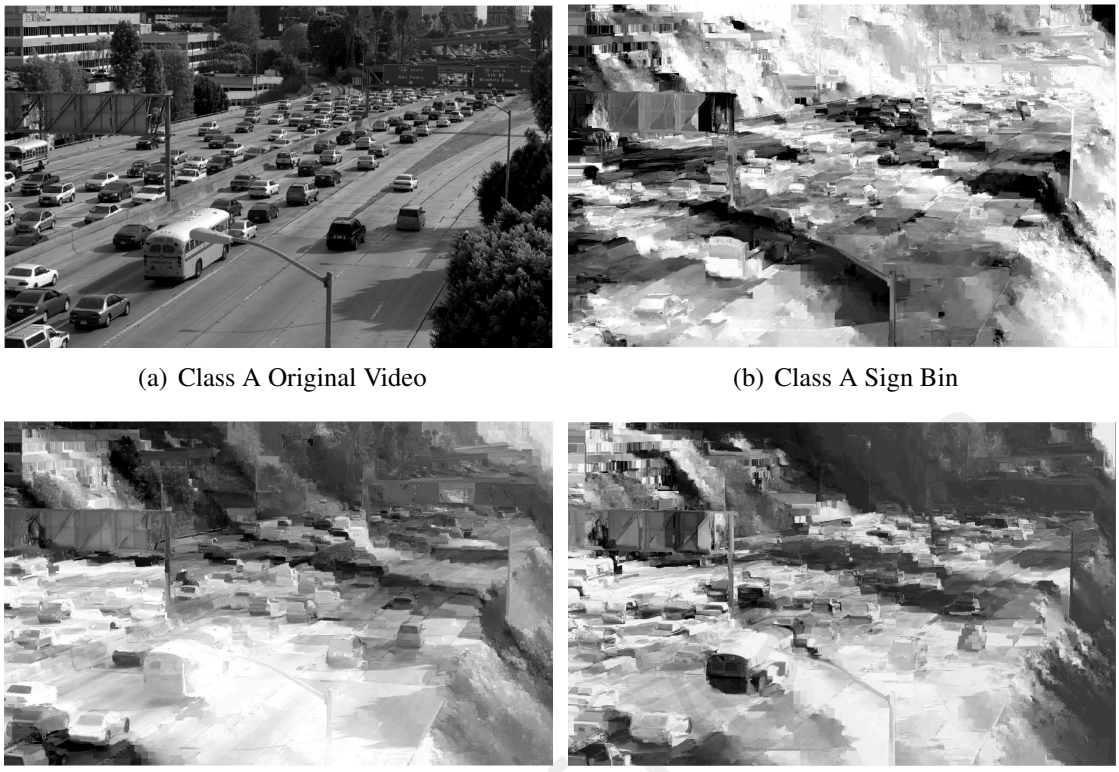
inal plaintext and the decrypted videos. On the other hand, the perceptual quality (quantified by SSIM score) is found to be consistently lower for all video classes and profiles.

Table 5.1: BD-Rate of original and decrypted video

Video	Transform Skip Bin							
	PSNR				SSIM			
	AI	LB	LP	RA	AI	LB	LP	RA
A	-0.0002	0.0000	-0.0009	0.0000	0.0000	0.0000	-0.0002	0.0000
B	-0.0002	0.0000	-0.0009	-0.0001	-0.0005	0.0000	- 0.0003	0.0000
C	-0.0004	0.0000	0.0039	-0.0008	-0.0014	-0.0001	-0.0008	0.0000
D	-2.8439	-0.0291	-2.9935	-0.0379	-2.9820	-0.0383	-2.9947	-0.0373
E	-0.0001	0.0000	-0.0042	0.0000	-0.0061	-0.0001	-0.0003	0.0000
F	-0.0005	0.0000	-0.0031	-0.0001	-0.0043	-0.0003	-0.0028	0.0000

Video	Suffix Bin							
	PSNR				SSIM			
	AI	LB	LP	RA	AI	LB	LP	RA
A	-0.0288	-0.0001	-0.0097	0.0000	-0.0087	0.0000	-0.0124	0.0000
B	-0.0066	0.0000	-0.0062	0.0000	-0.0063	-0.0001	-0.0047	0.0000
C	-0.0073	-0.0001	-0.0041	-0.0011	-0.0064	0.0003	-0.0072	0.0000
D	-2.8492	-0.0291	-2.9993	-0.0380	-2.9876	-0.0383	-3.0001	-0.0373
E	-0.0088	0.0000	-0.0015	0.0000	-0.0028	0.0000	-0.0040	0.0000
F	-0.0145	-0.0001	-0.0081	-0.0001	-0.0085	-0.0001	-0.0105	-0.0001

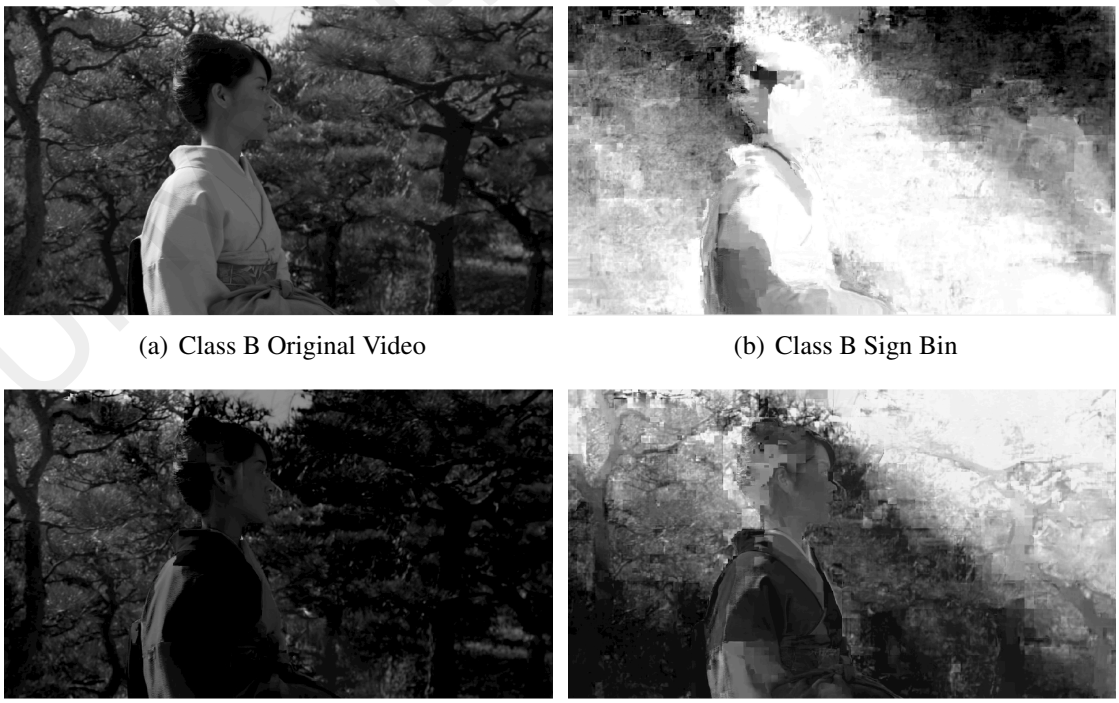
To further illustrate the results, Fig. 5.4 - 5.9 show the original and three encrypted video sequences in Class A, B, C, D, E and F by utilizing Sign Bin, Transform Skip Bin and Suffix Bin. Overall, the quality of the decrypted videos degrade by $\leq 3\%$ in terms of SSIM and PSNR when compared to their original compressed counterparts. Perceptually, both original and decrypted videos appear to be identical by visual inspection.



(a) Class A Original Video (b) Class A Sign Bin

(c) Class A Transform Skip Bin (d) Class A Suffix Bin

Figure 5.4: Original and encrypted Class A video by using three encryption techniques

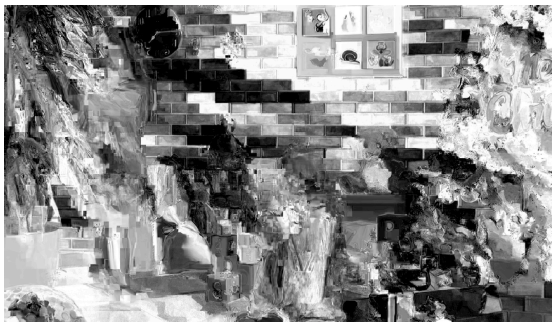


(a) Class B Original Video (b) Class B Sign Bin

(c) Class B Transform Skip Bin (d) Class B Suffix Bin

Figure 5.5: Original and encrypted Class B video by using three encryption techniques

(a) Class C Original Video



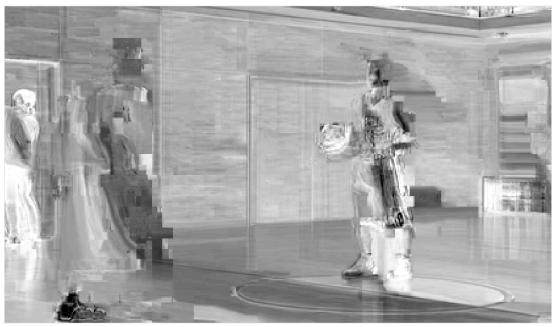
(b) Class C Sign Bin

(c) Class C Transform Skip Bin

(d) Class C Suffix Bin

Figure 5.6: Original and encrypted Class C video by using three encryption techniques

(a) Class D Original Video



(b) Class D Sign Bin

(c) Class D Transform Skip Bin

(d) Class D Suffix Bin

Figure 5.7: Original and encrypted Class D video by using three encryption techniques

(a) Class E Original Video

(b) Class E Sign Bin

(c) Class E Transform Skip Bin

(d) Class E Suffix Bin

Figure 5.8: Original and encrypted Class E video by using three encryption techniques

(a) Class F Original Video

(b) Class F Sign Bin

(c) Class F Transform Skip Bin

(d) Class F Suffix Bin

Figure 5.9: Original and encrypted Class F video by using three encryption techniques

5.5 Discussion

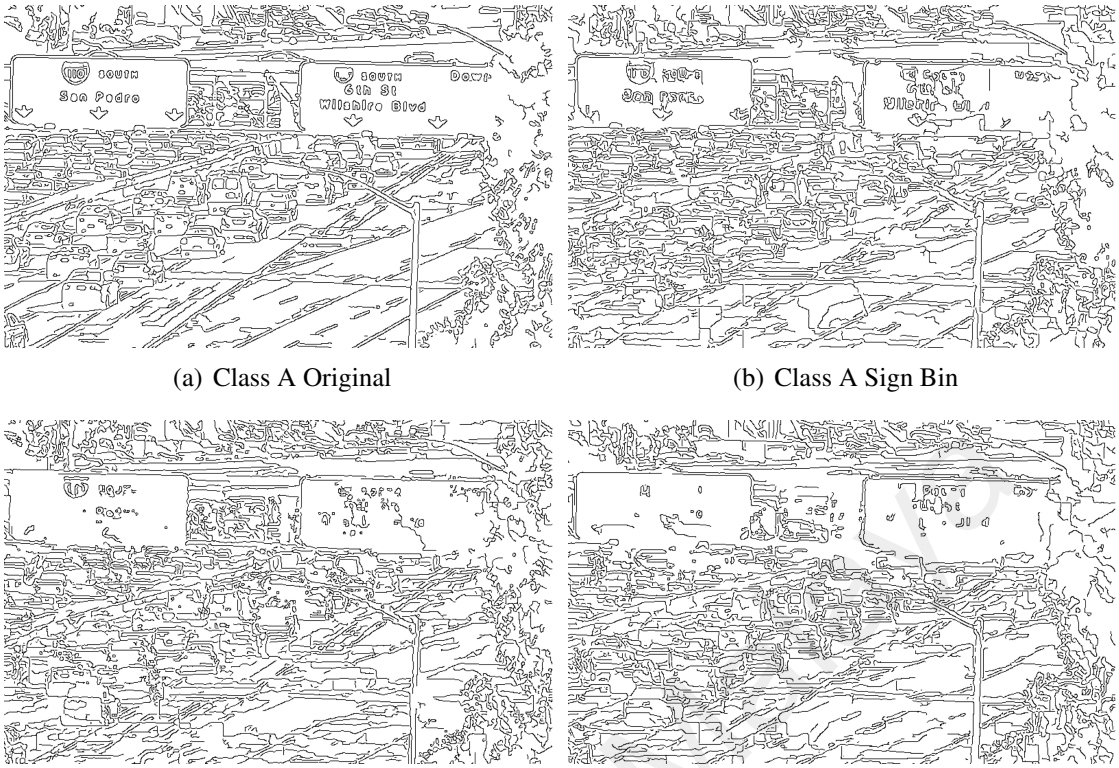
In this chapter, the robustness of the proposed selective encryption techniques are analyzed using outline detection attack.

5.5.1 Outline Detection on Encrypted Video

The proposed selective encryption techniques are analyzed by considering edges (i.e., outline) throughout the encrypted video sequences. Two commonly considered edge detection techniques, namely, Canny Outline Detection (CAN) (Canny, 1986) and Sobel Outline Detection (SOB) (Sobel & Feldman, 1968), are chosen to analyze the encrypted video sequences generated by the proposed technique. Figure 5.10 - 5.21 show the detected outline of Class A - F video sequences by using CAN and SOB edge detectors, respectively. These figures consist of detected (i.e., recognizable) edge from the original video (i.e., Fig. 5.4(a), 5.5(a), 5.6(a), 5.7(a), 5.8(a), 5.9(a),), which show a clear outline of object (e.g., basketball players and court lines in Fig. 5.13(a) and 5.19(a)). Noted that Figure 5.10 - 5.21 show only part of the Class A, B, C, D, E and F video slice for closer observation on edge detection. Based on the differences between (b), (c) and (d) in Fig. 5.4 - 5.9, the number of contour lines of the object (e.g., wall and basketball players in Fig. 5.13(b), 5.13(c), 5.19(b) and 5.19(c)) increases in the proposed technique. That is, the video encrypted by the proposed technique produces more complex outline when compared to the encrypted video generated by Sign Bin encryption technique.

The quality of the encrypted video sequences is further evaluated by measuring the edge differential ratio between the original and encrypted videos (Shahid & Puech, 2014). The edge differential ratio, denoted by $\Re$, is computed as follows:

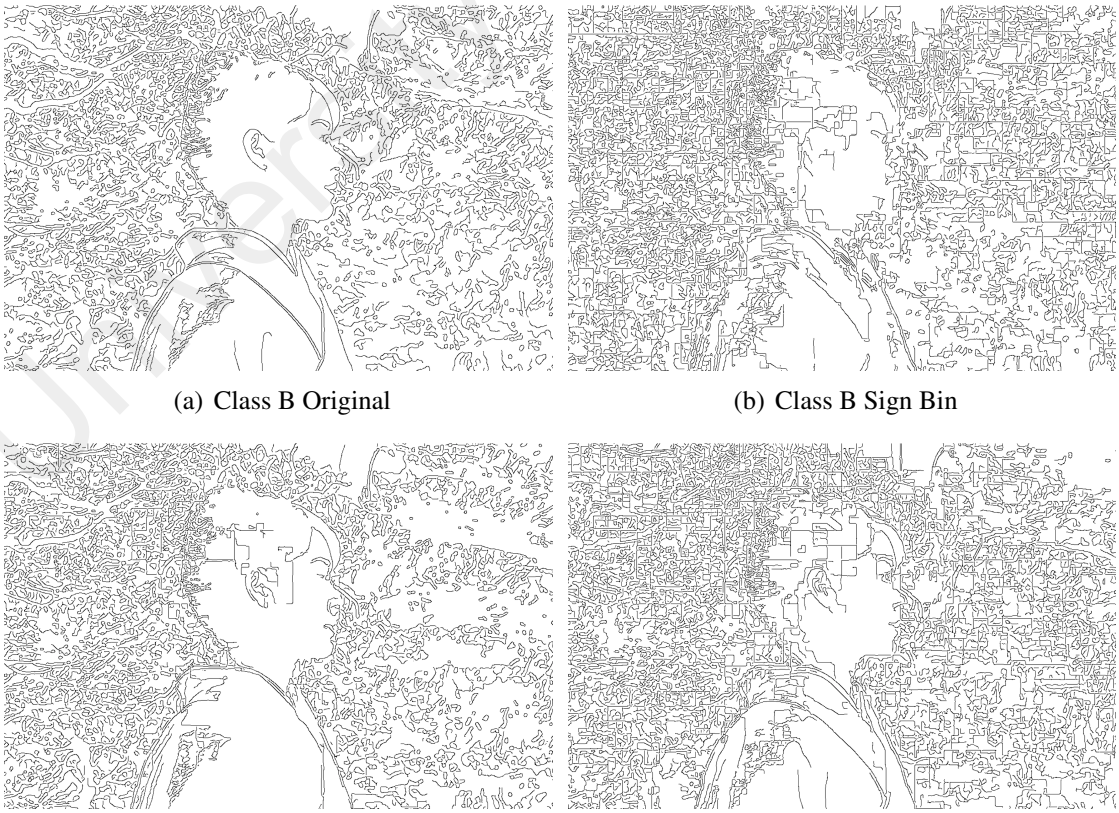
$$\Re = \frac{\sum_{i,j=1}^{N} |P(i,j) - \bar{P}(i,j)|}{\sum_{i,j=1}^{N} |P(i,j) + \bar{P}(i,j)|}, \tag{5.1}$$



(a) Class A Original (b) Class A Sign Bin

(c) Class A Trans. Skip Bin (d) Class A Suffix Bin

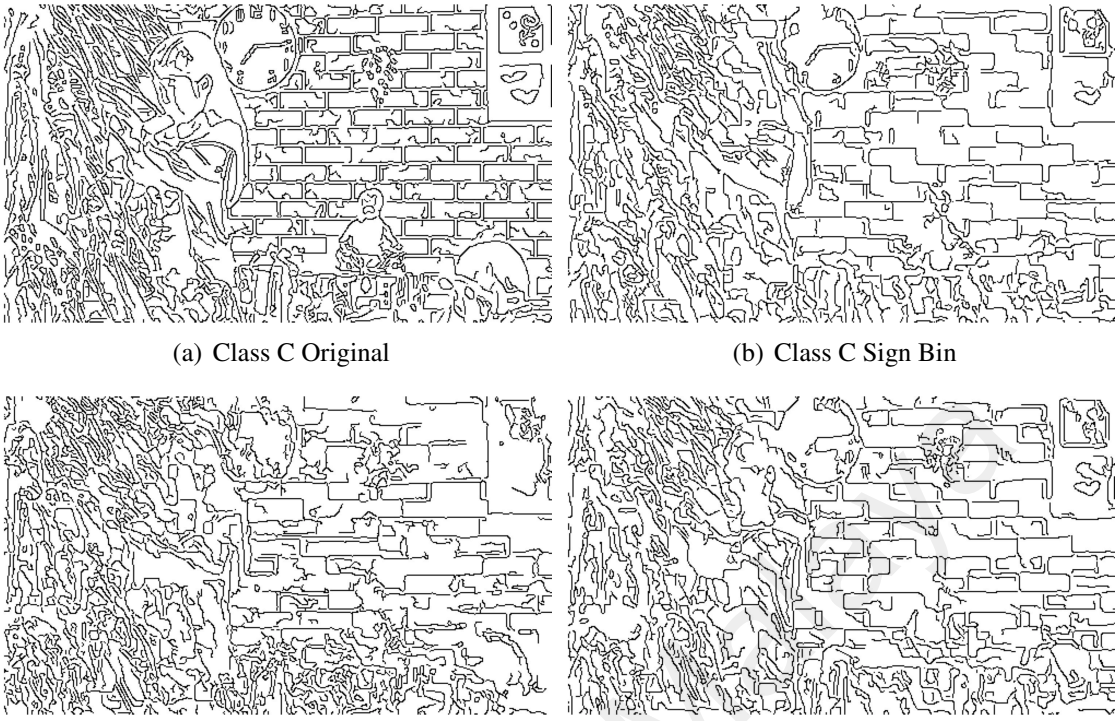
Figure 5.10: Canny outline detection on encrypted Class A video under RA profile



(a) Class B Original (b) Class B Sign Bin

(c) Class B Trans. Skip Bin (d) Class B Suffix Bin

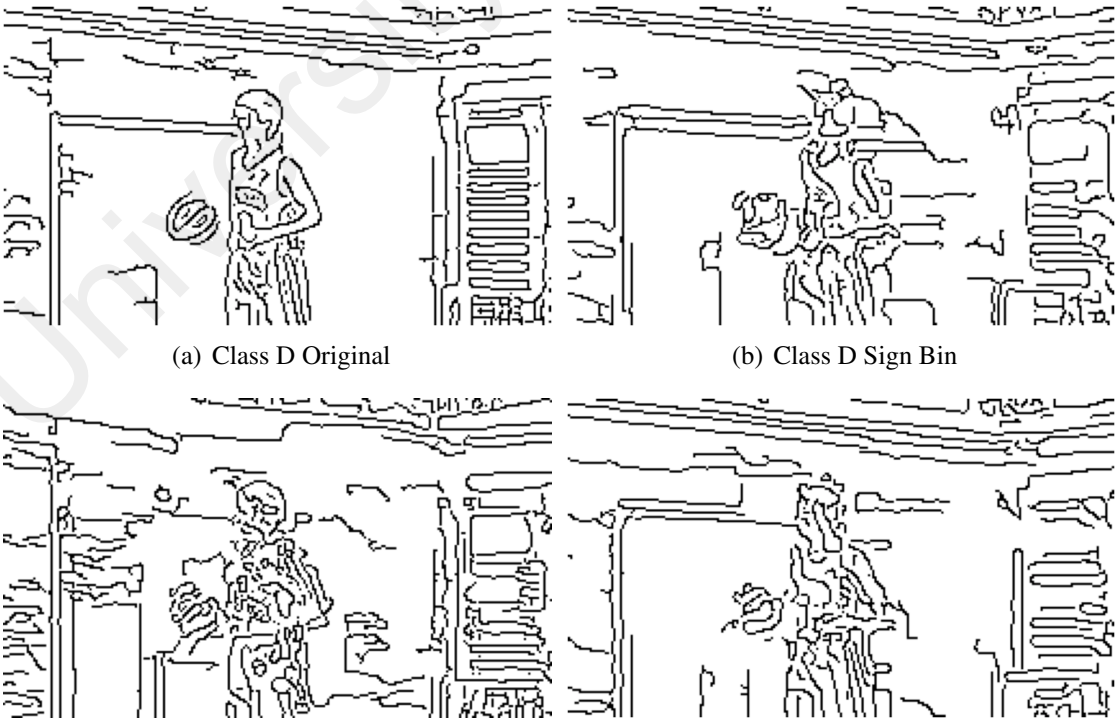
Figure 5.11: Canny outline detection on encrypted Class B video under RA profile



(a) Class C Original (b) Class C Sign Bin

(c) Class C Trans. Skip Bin (d) Class C Suffix Bin

Figure 5.12: Canny outline detection on encrypted Class C video under RA profile



(a) Class D Original (b) Class D Sign Bin

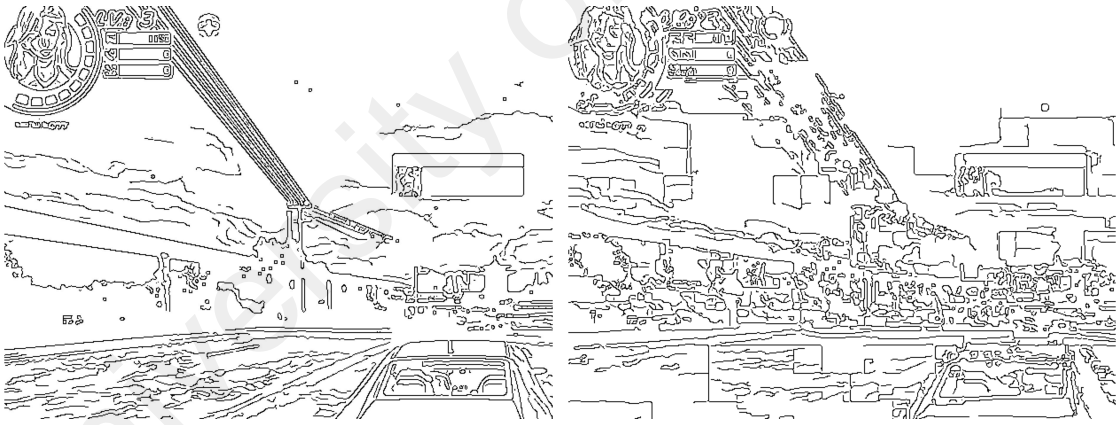
(c) Class D Trans. Skip Bin (d) Class D Suffix Bin

Figure 5.13: Canny outline detection on encrypted Class D video under RA profile

(a) Class E Original (b) Class E Sign Bin

(c) Class E Trans. Skip Bin (d) Class E Suffix Bin

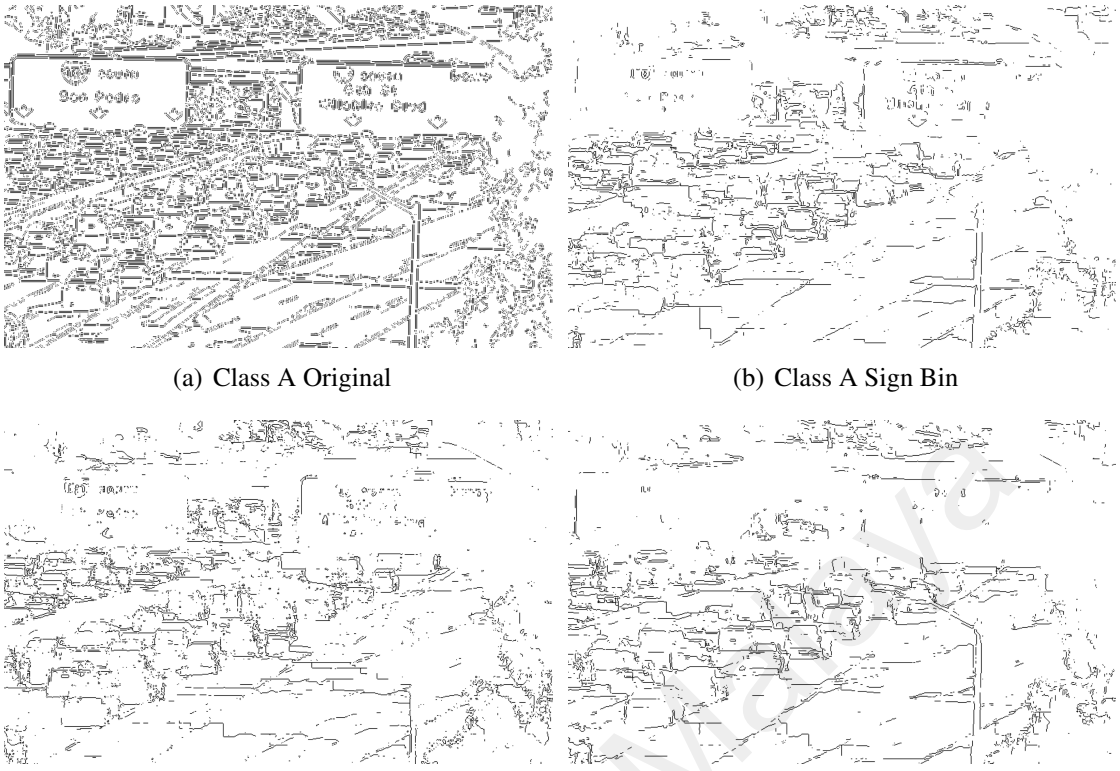
Figure 5.14: Canny outline detection on encrypted Class E video under RA profile



(a) Class F Original (b) Class F Sign Bin

(c) Class F Trans. Skip Bin (d) Class F Suffix Bin

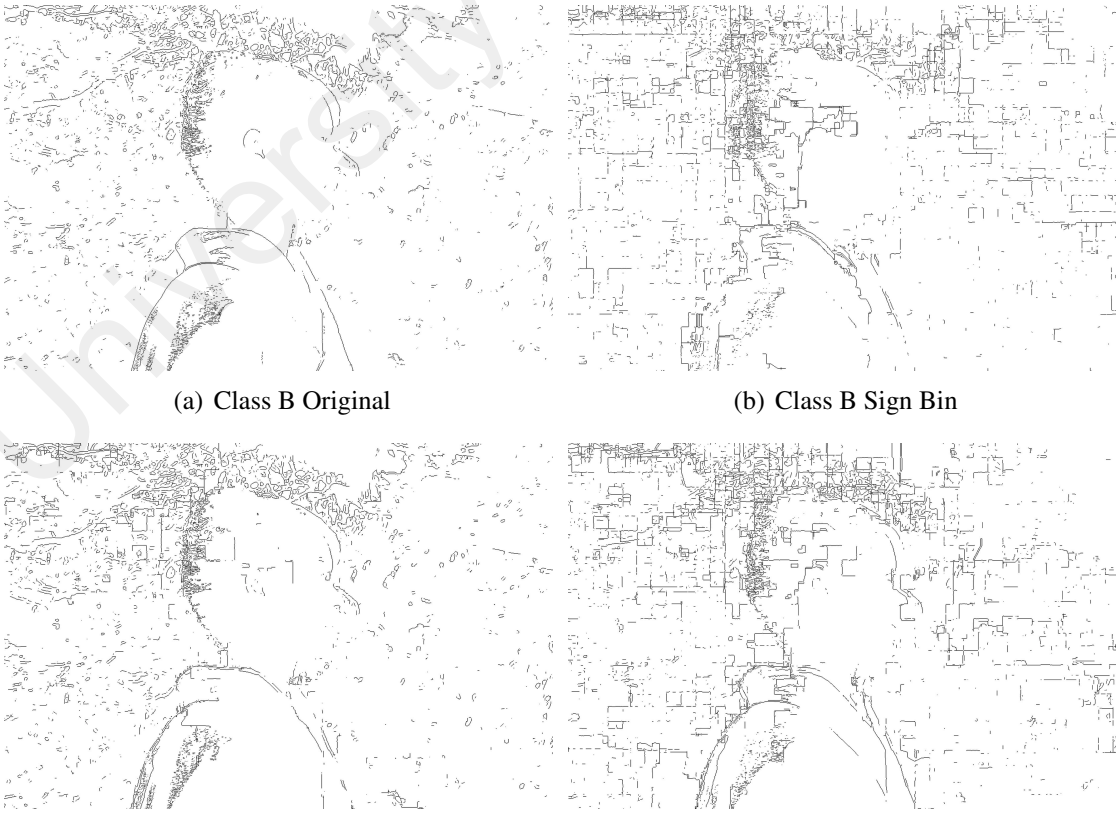
Figure 5.15: Canny outline detection on encrypted Class F video under RA profile



(a) Class A Original (b) Class A Sign Bin

(c) Class A Trans. Skip Bin (d) Class A Suffix Bin

Figure 5.16: Sobel outline detection on encrypted Class A video under RA profile

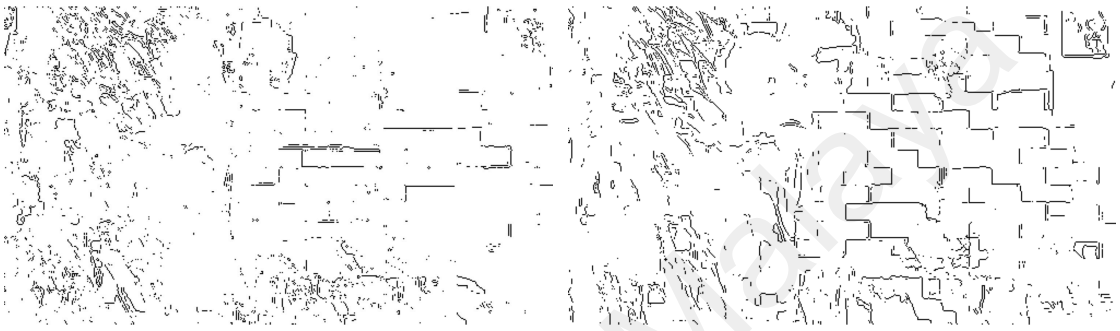


(a) Class B Original (b) Class B Sign Bin

(c) Class B Trans. Skip Bin (d) Class B Suffix Bin

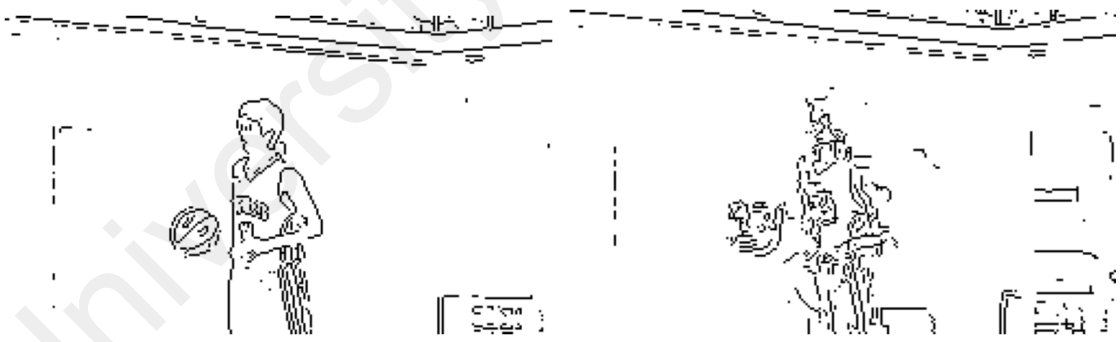
Figure 5.17: Sobel outline detection on encrypted Class B video under RA profile

(a) Class C Original (b) Class C Sign Bin



(c) Class C Trans. Skip Bin (d) Class C Suffix Bin

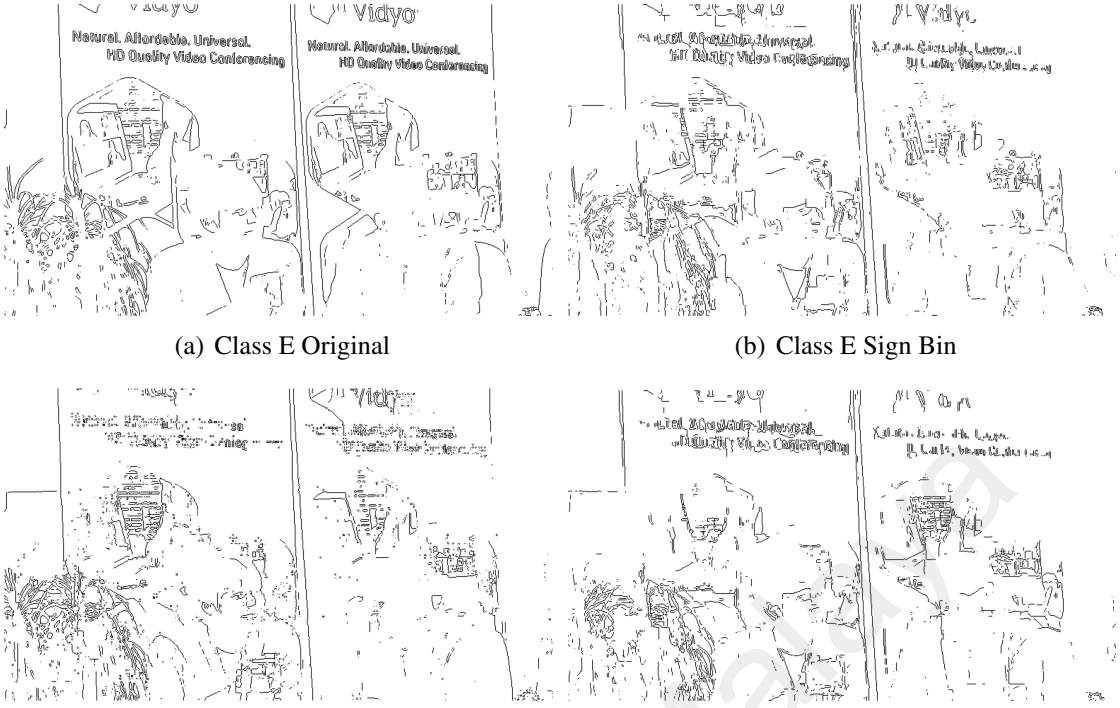
Figure 5.18: Sobel outline detection on encrypted Class C video under RA profile



(a) Class D Original (b) Class D Sign Bin

(c) Class D Trans. Skip Bin (d) Class D Suffix Bin

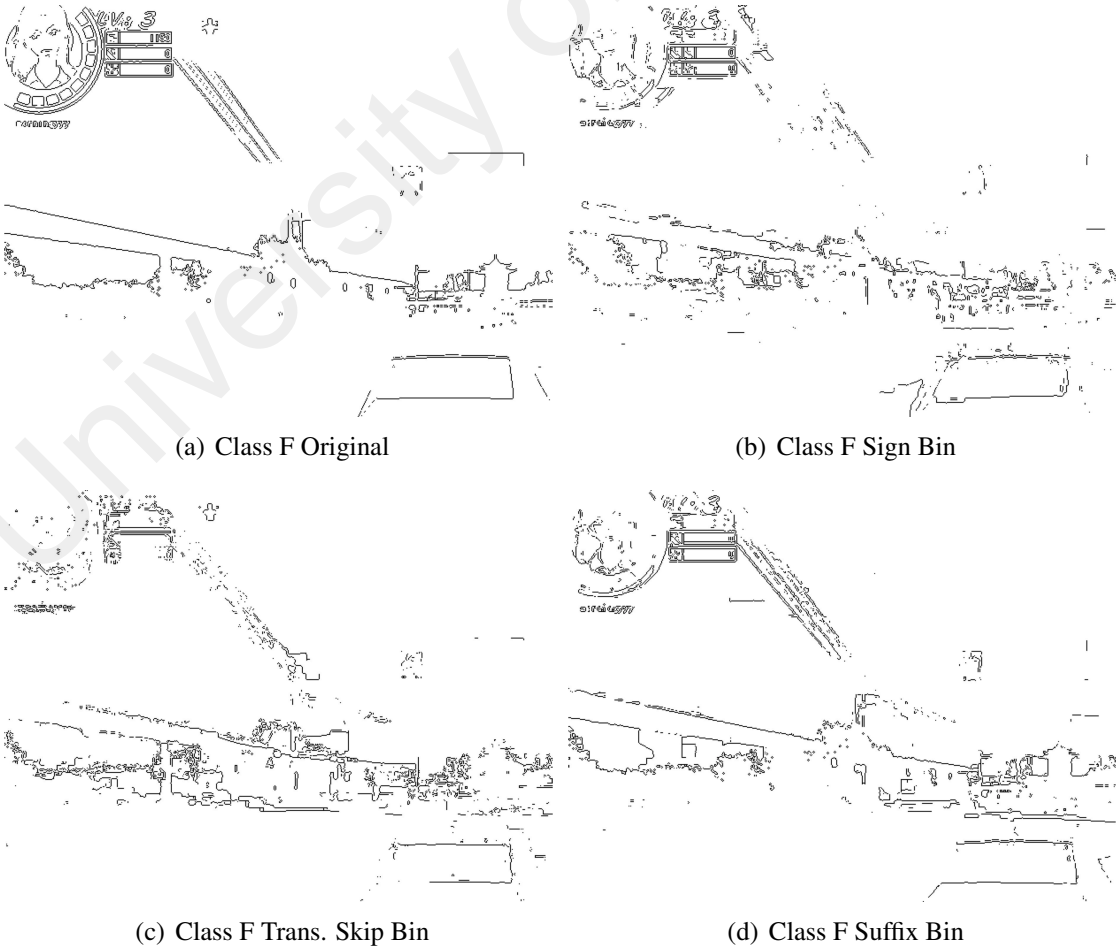
Figure 5.19: Sobel outline detection on encrypted Class D video under RA profile



(a) Class E Original (b) Class E Sign Bin

(c) Class E Trans. Skip Bin (d) Class E Suffix Bin

Figure 5.20: Sobel outline detection on encrypted Class E video under RA profile



(a) Class F Original (b) Class F Sign Bin

(c) Class F Trans. Skip Bin (d) Class F Suffix Bin

Figure 5.21: Sobel outline detection on encrypted Class F video under RA profile

Table 5.2: Edge Difference Ratio in between original and encrypted video.

Video Class	Sign Bin		Trans. Skip Bin		Suffix Bin	
	Canny	Sobel	Canny	Sobel	Canny	Sobel
A	0.2592	0.2071	0.1932	0.1357	0.2411	0.1880
B	0.6393	0.6192	0.4778	0.4535	0.4017	0.3716
C	0.5713	0.5381	0.2770	0.2538	0.3655	0.3351
D	0.5121	0.4744	0.3283	0.3019	0.4125	0.3739
E	0.3916	0.3261	0.3684	0.2932	0.3825	0.3124
F	0.4294	0.4185	0.3408	0.3393	0.3605	0.3514

where $P(i,j)$ and $\bar{P}(i,j)$ denote the detected binary pixel values in the original and encrypted video slices, respectively, (i,j) denotes the position of the binary pixel, and N denotes total number of pixels in a video slice. The value of $\Re$ ranges from 0 to 1, where higher value indicates better masking of the structural information of a video slice while lower value indicates higher similarity between the original plaintext and encrypted video slices. Table 5.2 shows the average $\Re$ for the encrypted video sequences from various classes generated with the proposed techniques. It is observed that the $\Re$ value for Sign Bin encryption is higher than Transform Skip and Suffix Bin encryption techniques (i.e., > 0.2071). In other words, Sign Bin encryption is able to mask the perceptual meaning of the video more effectively when compared to other two encryption techniques.

5.5.2 Error Concealment Attack

Encrypted video can possibly be recovered (i.e., decrypted) by utilizing error concealment techniques (Stütz & Uhl, 2009). In this case, attackers can decode and recover the encrypted video without considering the decryption process, i.e., the encrypted video element (e.g., modified coefficient) will be treated as an error in each coding pass during the decoding process. Here, several common actions can be taken by the decoder to conceal the errors during the decoding process: (a) truncate the encrypted file at the position where the error has occurred (stop decoding immediately after the error), (b) set the encrypted video elements (e.g., coefficients) to zero, or (c) reset the encrypted video element to the last value of non-encrypted video element before the detected error.

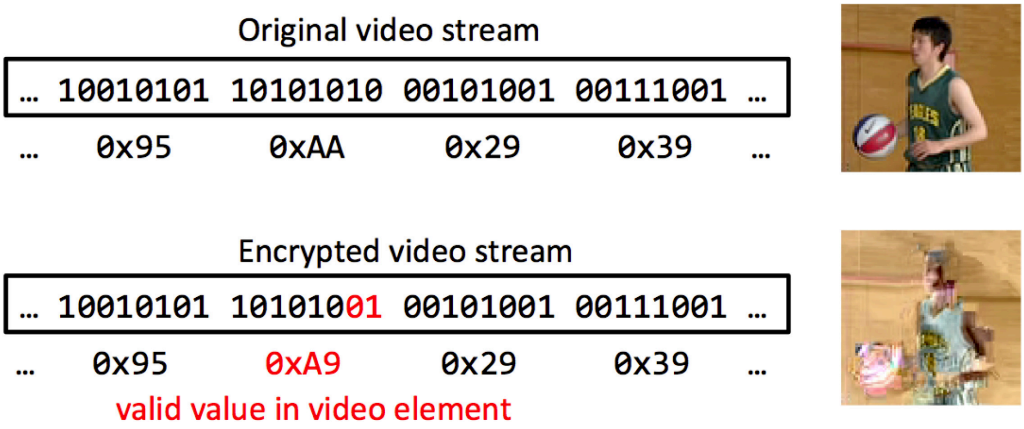


Figure 5.22: Example of original and encrypted video bitstreams

However, these error concealment attacks only work when errors are detected in an encrypted video. The proposed video encryption scheme encrypts video based on sign bin, transform skip bin and suffix bin manipulation and it achieves video format compliance. The manipulated bins (e.g., sign of coefficients) are valid values in coding pass during the decoding process. In other words, decoder cannot detect any error on encrypted video because the encrypted video elements exhibit normal coding pass in the decoder. Figure 5.22 shows part of the original and encrypted video stream by the proposed scheme. The value 0xAA is modified to 0xA9 in video bin to encrypt the video (i.e., distort the video quality). Noted that value 0xA9 is a valid value in coding pass, which can be decoded by original decoder to produce distorted video content. Therefore, the proposed scheme is robust to common error concealment attacks.

5.5.3 Functional Comparison

In this chapter, the proposed scheme is compared with five conventional encryption schemes (i.e., AES Encryption (Dumbere & Janwe, 2014), Network Abstraction Layer (NAL) unit encryption (C. Li et al., 2008), Coding Block Header data encryption (Lian et al., 2007), Syntax encryption (X. Wang et al., 2010) and Sign encryption (Hofbauer et al., 2014)). Table 5.3 summarizes the functional comparison among the video encryption schemes considered. An encryption scheme is indicated as format compliant if it is applicable to the latest HEVC video standard, and able to be decoded while being in the encrypted form

Table 5.3: Comparison with other encryption scheme

Encryption scheme	Functionality			
	Domain	F	C	T
AES Encryption (Dumbere & Janwe, 2014)	Bitstream		✓	
NAL unit encryption (C. Li et al., 2008)	Bitstream		✓	✓
Header data encryption (Lian et al., 2007)	Transform		✓	
Syntax encryption (X. Wang et al., 2010)	Bitstream	✓	✓	
Sign encryption (Hofbauer et al., 2014)	Bitstream	✓		✓
Proposed scheme	Trans. & Bits.	✓	✓	✓

F = Format compliant, C = Compression dependent, T = Low computational time

(i.e., without decryption prior to decoding). It is found that most schemes that manipulate the video content cannot be decoded by using the original decoder except (Hofbauer et al., 2014), (X. Wang et al., 2010) and the proposed scheme.

For those schemes that modify the video content with respect to the RDO decision, it is indicated as compression dependent. Sign encryption is the only technique which does not affect the RDO decision after the encryption process. Hence, the proposed scheme includes the sign encryption to exploit this advantage. Computational cost for applying encryption scheme depends on the complexity of the encryption algorithm. Schemes (Dumbere & Janwe, 2014), (Lian et al., 2007) and (X. Wang et al., 2010) involve high cost operations (e.g., permutation) and long execution time to perform the encryption operation(s) on the particular video components (e.g., coding block header, motion vector displacement). On the other hand, NAL unit encryption, sign encryption and the proposed scheme encrypt a video stream by manipulating particular syntax elements (e.g., *nalUnitType*, *coeffSigns*, *m_puhTransformSkip*) in the HM16.0 encoder during the encoding process. Therefore, these manipulation require minor computational cost, when compared to those that manipulate the video component(s).

5.6 Summary

A selective video encryption scheme was proposed by utilizing three encryption techniques to secure the confidentiality of video from unauthorized receiver. These techniques

exploited the HEVC video coding structure (i.e., Sign Bin, Transform Skip Bin and Suffix Bin) to selectively encrypt the video stream and preserve the embedded information before and after decryption. Results suggested that the output video (i.e., decrypted video) exhibited similar perceptual quality as the original encoded video. On the other hand, the encrypted video were analyzed and justified based on the edge differential ratio. Meanwhile, the proposed scheme also compared to the other encryption schemes and found to be format compliant, compression dependent and low computational time.

CHAPTER 6 : JOINT AUTHENTICATION & ENCRYPTION SCHEME

6.1 Overview

In this chapter, the integration work of Chapter 3, 4 and 5 is discussed to realize a joint video authentication and encryption scheme. By maintaining the authentication feature through the proposed video encryption scheme, the output video can be authenticated in encrypted or decrypted domain. Experiment results suggest that the proposed authentication scheme with selective encryption techniques achieves format compliance and maintains the quality of decrypted video. Functional comparison with conventional joint schemes and possible application of the proposed joint scheme are discussed.

6.2 Introduction

Multimedia communications and information security are two active areas in both academia and industry. The trend shows a fusion between them to allow a secure delivery of multimedia data. According to (Rivest, 1991; Shirey, 2000), the security of data is pursued by assuring, among others: authentication, to verify the identity claimed by or for any system entity; data confidentiality, to protect data against unauthorized disclosure; data integrity, to verify that data have not been changed, destroyed, or lost in an authorized or accidental manner. To satisfy these constraints, several works were put forward, such as information hiding (Chapter 3) and encryption (Chapter 5).

Based on the proposed scheme in Chapter 4, information hiding techniques are suitable for video authentication and copyright protection. Here, the video stream is the cover and the protection of its ownership is the goal of the information hiding technique. On the other hand, encryption scrambles the video contents so that they become unintelligible. It focuses on rendering information not intelligible to any unauthorized entity who might intercept them. In this case, the video content is kept secret. Actually, encrypted

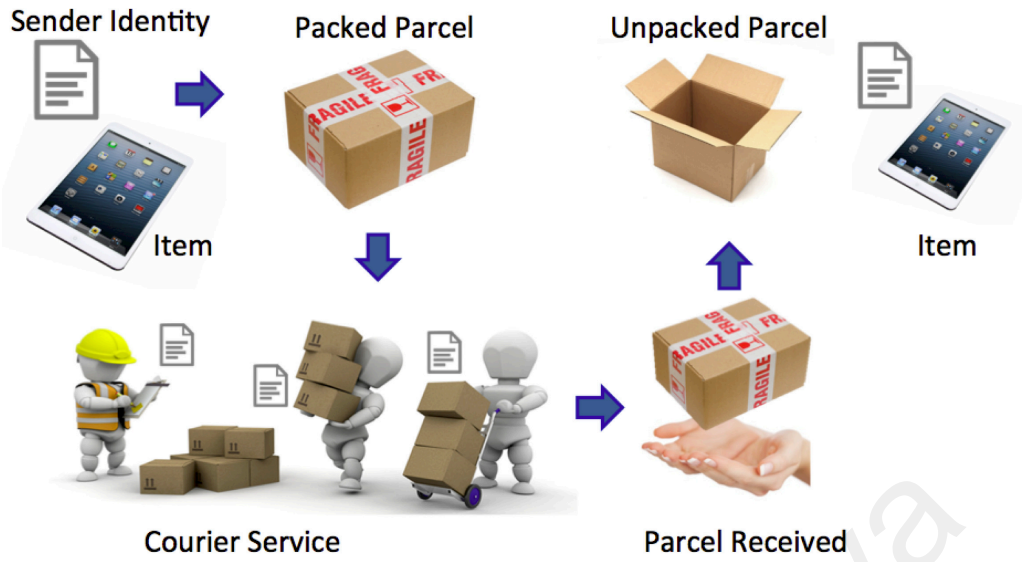


Figure 6.1: Proposed scheme for Parcel Delivery

videos need an additional level of protection in order to keep control on them after the decryption phase. In fact, when the encrypted video is decrypted by the authorized user, it is unprotected and it can be easily modified, tampered, or stolen. The scientific community started focusing on the possibility of providing both security services simultaneously and therefore to have the chance of embedding and detecting the tag (i.e., authentication code) before and after decryption. This allows the operability in the encrypted domain, dealing with encrypted (ciphered) video without giving access to the plain video as well as increasing the operation efficiency.

In most practical cases, the embedded tag can be replaced by any other information (e.g., watermark) to achieve specific application (e.g., content protection against unauthorized receiver). For instance, Fig. 6.1 illustrates an analogue scenario for the possible application of the proposed scheme. Here, a well-packed parcel (i.e., encrypted video) is sent to a receiver via a courier service. The courier service takes the responsibility to identify the source of the parcel by obtaining the sender identity (i.e., tag extraction) from the parcel. Then at the receiver side, the parcel is unpacked (i.e.,decrypted) and the same sender identity can be retrieved.

Another possible application is to achieve fingerprinting features in encrypted video. Fig. 6.2 shows the similar approach with Fig. 6.1, a well-packed parcel includes (embed-

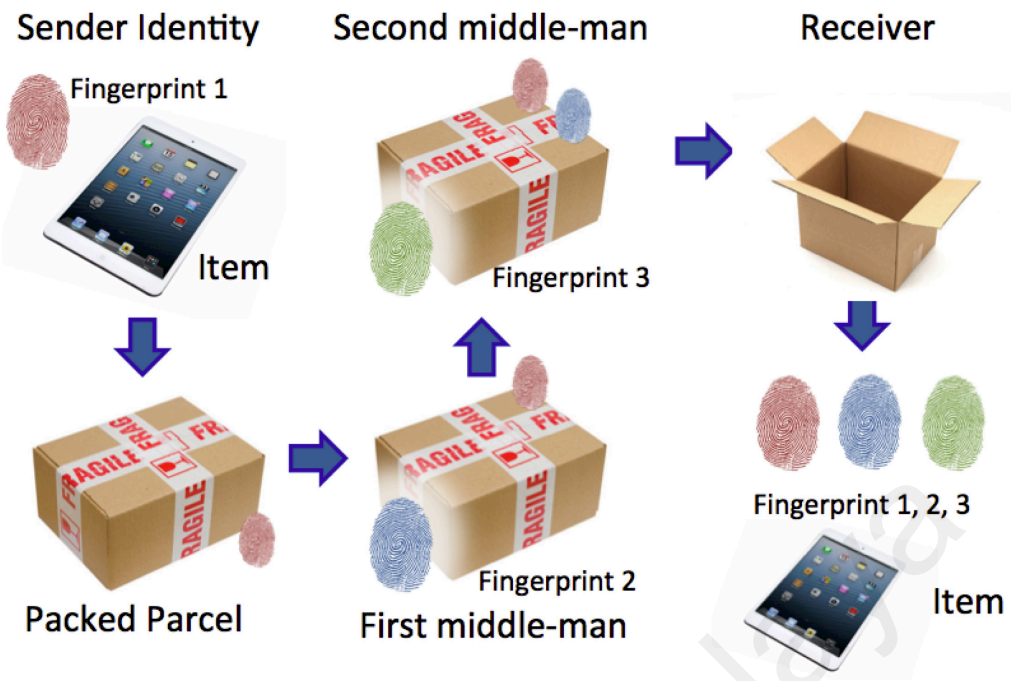


Figure 6.2: Proposed scheme for Fingerprinting

ded with) fingerprint 1 from the sender. Throughout the process of parcel delivery, each middle-man left his fingerprint (concatenated to embed middle-man identity) to the host (i.e., encrypted video) and send to the following receiver. At the end, the receiver can obtain the sender and middle-man identity in encrypted or decrypted video.

Therefore, an HEVC format-compliant joint authentication and encryption scheme is proposed. The joint scheme is able to secure video content and support authentication in both encrypted and decrypted forms. As described in Chapter 5, the joint scheme is separable, where the decryption and authentication (i.e., tag extraction and validation) processes are independent, with minimal parsing overhead. Specifically, elements in the HEVC coding structure are divided into two groups, where one group is manipulated to perceptually mask the video content, while another is modified to embed tag.

6.3 Experiment Result

The proposed joint scheme is implemented by utilizing the same reference software, as mentioned in Chapter 4 and 5 (i.e., HM16.0). Four profiles (i.e., AI (All Intra), LP (low delay P), LB (low delay B) and RA (random access)) are considered for performance evaluations. Here, the combination of all encryption techniques proposed in Chapter 5 is

performed using the authentication scheme proposed in Chapter 4. All video classes, i.e., *PeopleOnStreet* (Class A), *BasketballDrive* (Class B), *BQMall* (Class C), *RaceHorses* (Class D), *Night* (Class E) and *Kendo* (Class F) are utilized to evaluate the video quality performance of the proposed joint scheme.

6.3.1 Quality Evaluation

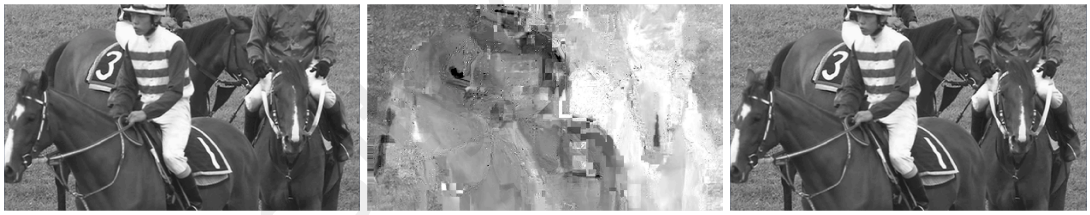
Figure 6.3(b), 6.3(e), 6.3(h), 6.3(k), 6.3(n) and 6.3(q) show all classes of video encrypted by the combination of SiB, TsB, SuB encryption techniques. Generally, it is observed that the video becomes blocky (e.g., outline of basketball players in 6.3(e)) due to the bin manipulation in CU, where the modification of each bin leads to the distortion in the corresponding square block. Next, Fig. 6.3(c), 6.3(f), 6.3(i), 6.3(l), 6.3(o) and 6.3(r) show the decrypted videos, which exhibit similar perceptual quality with the original video. Note that tags are presented in both encrypted and decrypted videos, in other words, the authentication feature is preserved in both videos.

To further illustrate the perceptual video quality, Fig. 6.4 and 6.5 show the rate distortion curves of the original and encrypted videos for all classes in RA profile. The solid line represents the original video, which yields the highest quality, while the dotted lines with PSNR value ≤ 20 dB represent the encrypted videos with embedded authentication code (i.e., tag). Note that lower PSNR value implies less similarity to the original video, where low PSNR value is sought for in the case of an encrypted video. Results of three individual encryption techniques are also presented (similar to the result in Fig. 5.1 - 5.3) as comparisons to the combination of all encryption technique (i.e., light blue dotted line). Results suggest that by applying the encryption techniques as well as their combination, the encrypted video achieves sufficient distortion in quality, indicating the achievement of secrecy. Furthermore, a magnified graph shows that the tag embedding process using information hiding technique in Chapter 3 only leads to insignificant quality degradation,

(a) *PeopleOnStreet* original (b) *PeopleOnStreet* encrypted (c) *PeopleOnStreet* decrypted

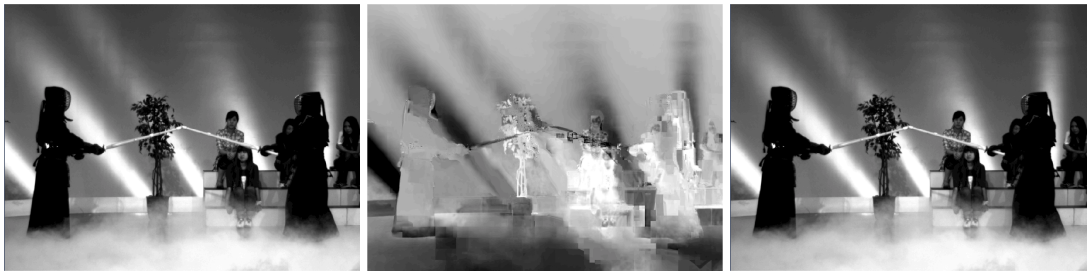
(d) *BasketballDrive* original (e) *BasketballDrive* encrypted (f) *BasketballDrive* decrypted

(g) *BQMall* original (h) *BQMall* encrypted (i) *BQMall* decrypted



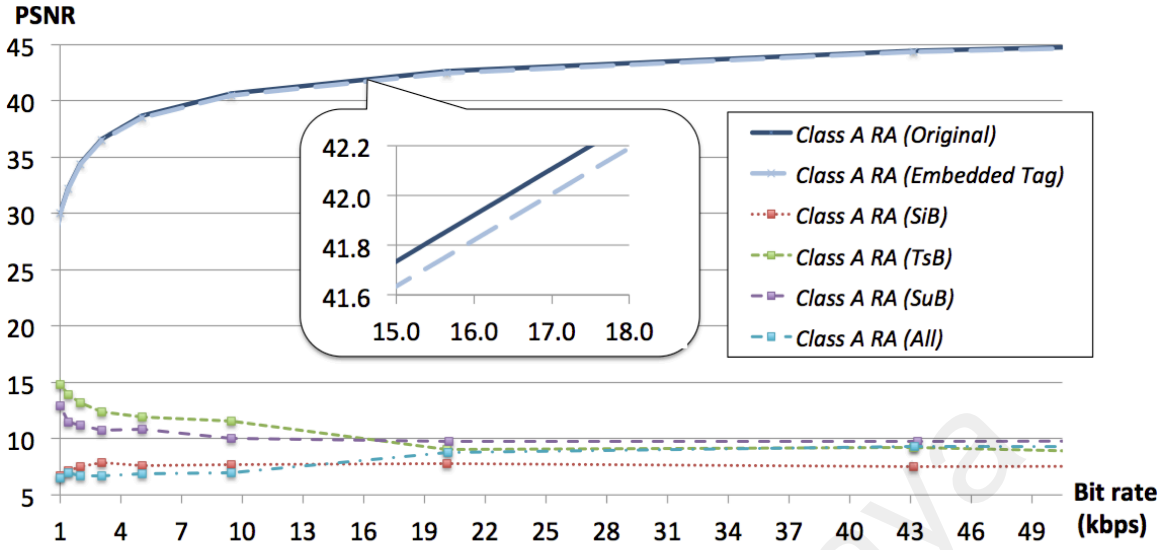
(j) *RaceHorses* original (k) *RaceHorses* encrypted (l) *RaceHorses* decrypted

(m) *Night* original (n) *Night* encrypted (o) *Night* decrypted

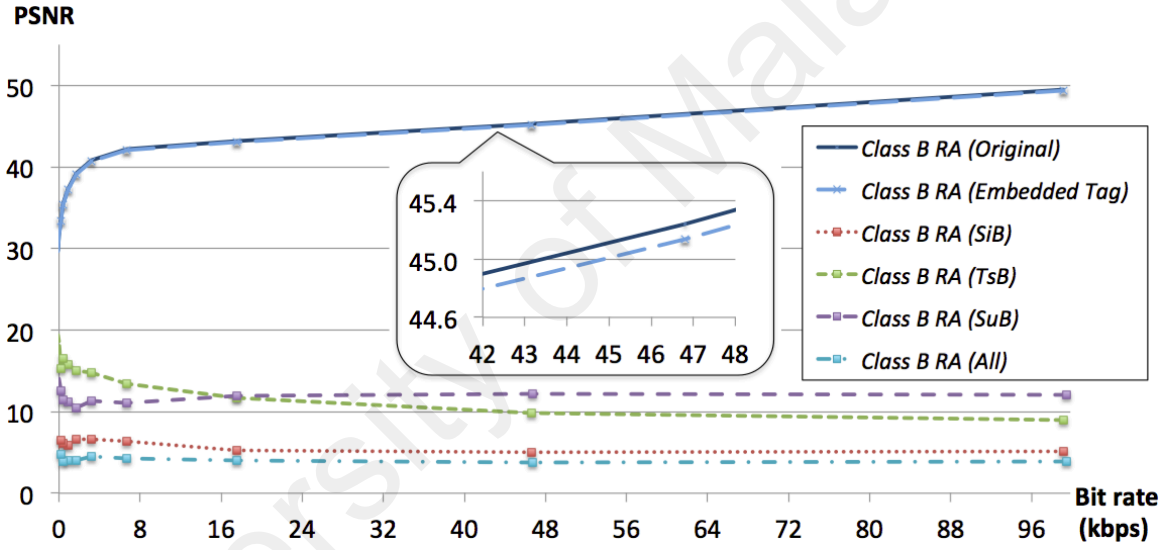


(p) *Kendo* original (q) *Kendo* encrypted (r) *Kendo* decrypted

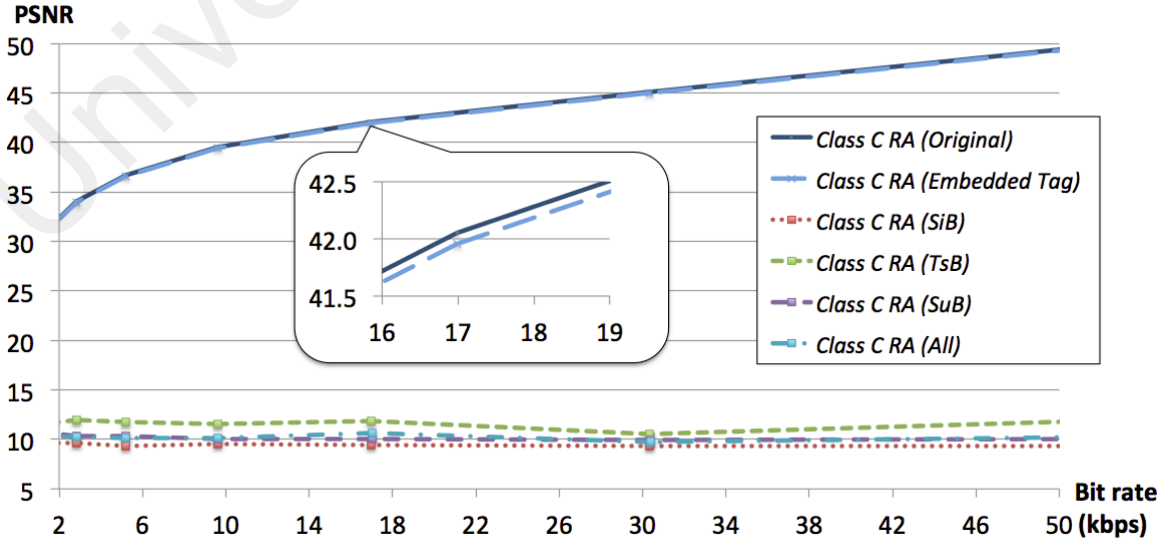
Figure 6.3: Original, encrypted and decrypted videos with authentication feature



(a) Class A PSNR

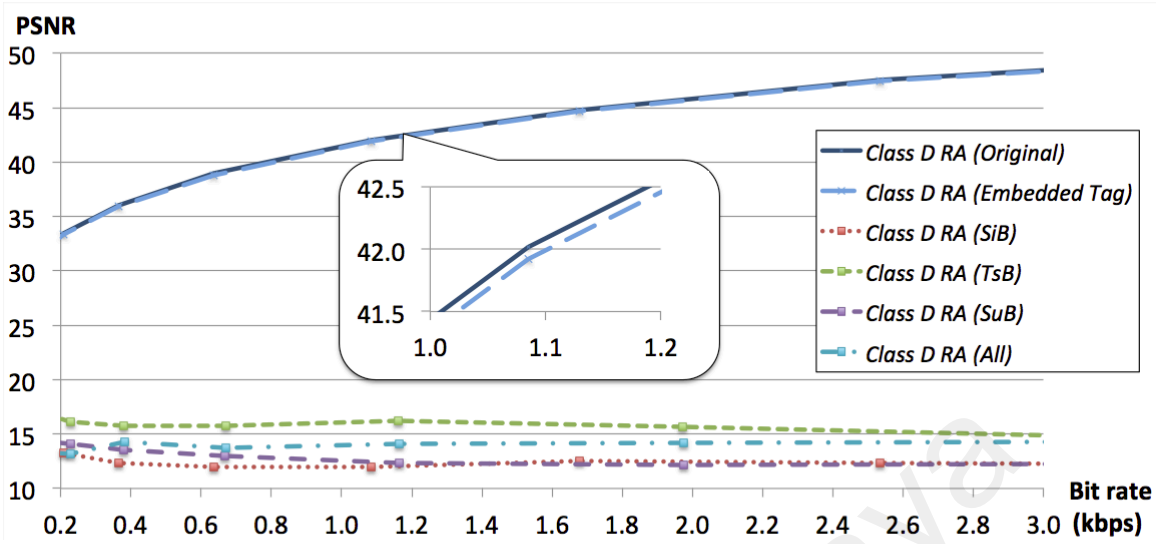


(b) Class B PSNR

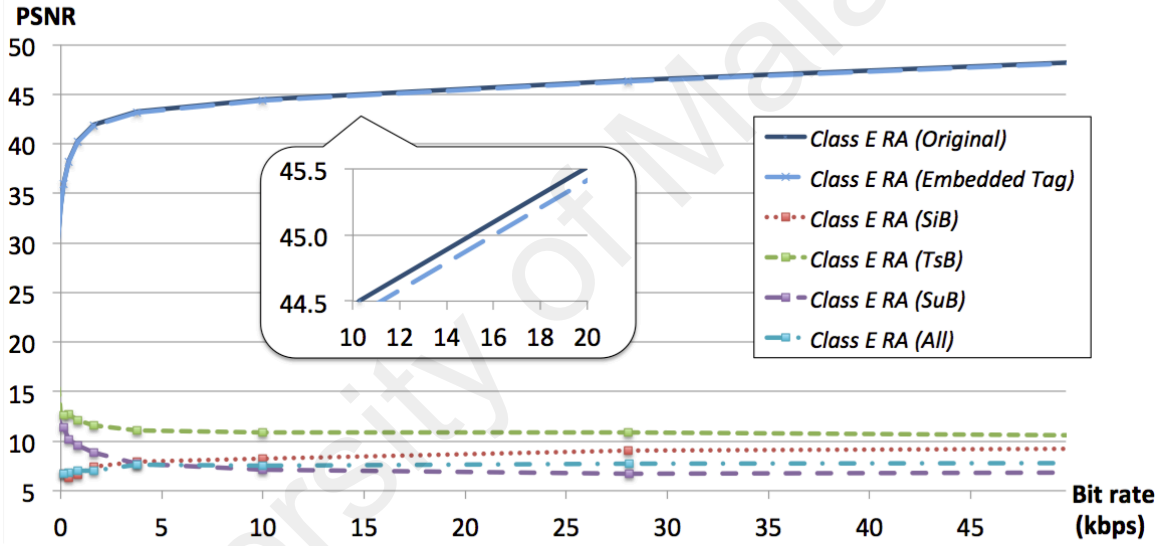


(c) Class C PSNR

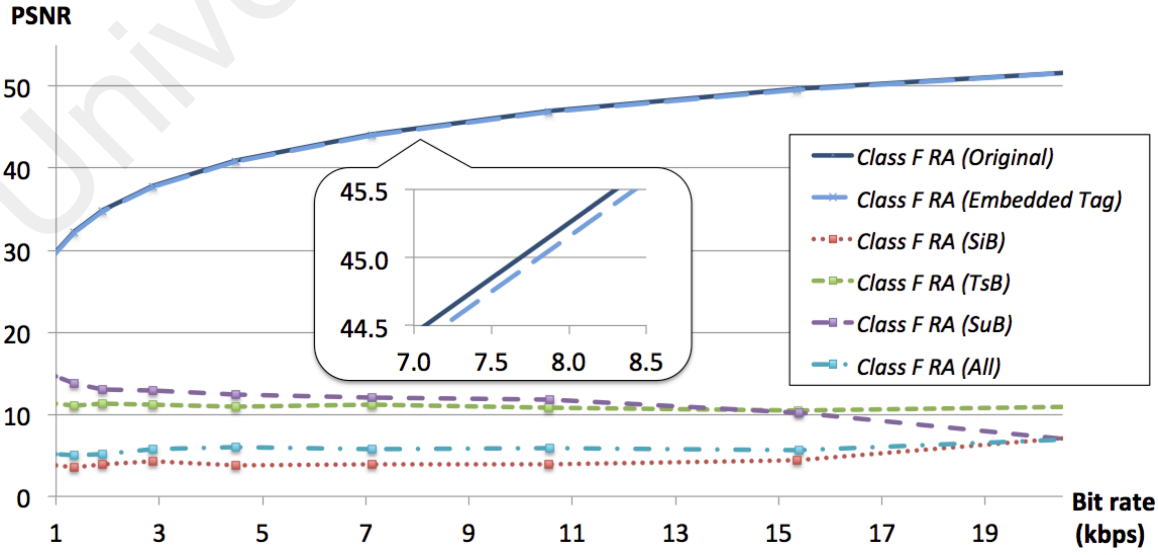
Figure 6.4: Rate Distortion Curve for video in Class A, B and C



(a) Class D PSNR



(b) Class E PSNR



(c) Class F PSNR

Figure 6.5: Rate Distortion Curve for video in Class D, E and F

Table 6.1: BD-Rate in PSNR for original and encrypted video

Video	All			
	AI	LP	LB	RA
A	-32.172	-30.979	-30.962	-31.608
B	-36.718	-39.746	-27.357	-27.826
C	-23.631	-23.679	-23.589	-23.816
D	-26.062	-25.380	-25.432	-23.628
E	-33.656	-33.139	-33.044	-33.703
F	-33.655	-32.214	-32.041	-31.717

where the distortion intensifies after applying the proposed encryption scheme. Moreover, as expected, the lowest video quality can be achieved by applying all encryption techniques altogether (see Fig. 6.4(b)).

Table 6.1 further records the BD-rate of the original and encrypted-and-authenticated videos by using the combined techniques in terms of PSNR (dB). In most video classes, the proposed video encryption scheme (i.e., combination of all three encryption techniques) achieves greater distortion (e.g., -32.172 in Class A) in AI when compared to other profiles. In summary, the proposed video encryption scheme distorts the encrypted video in the range of -23.63 to -39.75 in terms of BD rate.

6.3.2 Key Sensitivity, Decoding and Extraction Times

To examine the effectiveness of the proposed video encryption scheme, the encryption key space for bin selection, i.e., deciding which bin to manipulate is considered. Here, a 32-bit key, which yields a key space of 2^{32} combinations is utilized. By design, a key is fed into a hash function (i.e., SHA256), and its output is used in selecting the bins for modification. To carry out the test, a video is encrypted by using the proposed scheme with key $= \kappa \in [0, 2^{32} - 1]$. Then, 255 random numbers (each of length 32 bits) are considered as the keys to decrypt the video. Results in Fig. 6.6 show the graph of PSNR vs. key index, where only the exact 32-bit key (i.e., κ) can decrypt the video. In other words, the high quality video (i.e., 44.348 dB) is attained with the correct key while the rest of the keys result in low quality video. Recall that when SiB, TsB, and SuB are not in

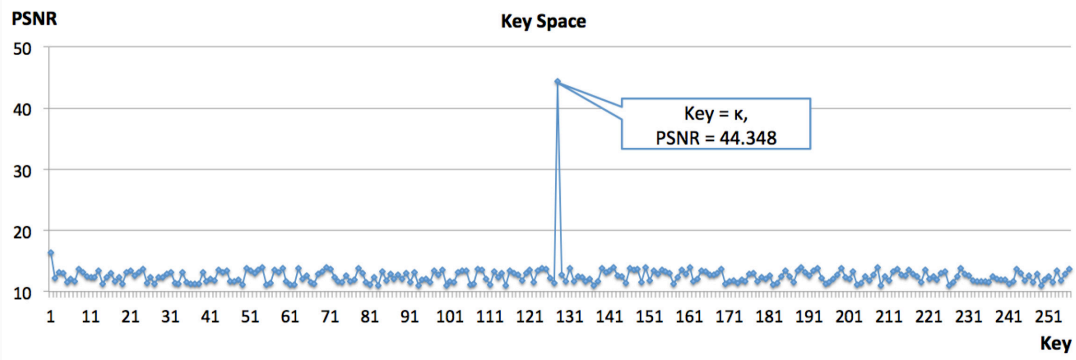


Figure 6.6: Encryption key space

their original forms, the video is completely imperceptible (i.e., effect of each encryption technique). For example, SiB determines the phase of the basis vector, and when toggled, the pixels are flipped from black to white, and vice versa. Therefore, the proposed scheme is sensitive to the decryption key.

Next, the time needed for decrypting the encrypted video, decoding it for display and verifying the video authenticity are evaluated. A full length video (i.e., 500 slices in 30 fps) from Class D is considered, where the original and encrypted videos are decoded for 10 times to compute the average decoding and verifying time (in unit of second) when using different bitrates. Fig. 6.7 shows the graph of decoding (decrypting and verifying) time vs bitrate for the proposed joint encryption scheme. In the worst case scenario (i.e., longest decoding time), it took 11s to decode and verify a 16s video, which is encoded at 18kbps, where this performance is acceptable for real-time video streaming application.

6.4 Discussion

In this chapter, a functional comparison among proposed joint scheme and conventional joint scheme is presented. The possible application of the proposed joint scheme is put forward to realize the practical usability in digital media, specifically for security and privacy protection purposes.

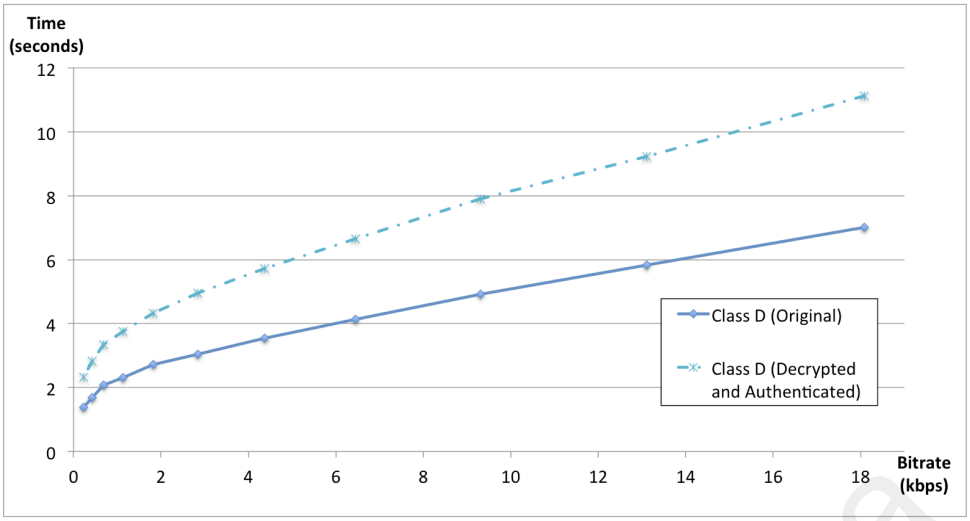


Figure 6.7: Time taken to decode the original video and decrypting-and-decoding the encrypted video - Class D

6.4.1 Functional Comparison among Schemes

Literature review reveals that there is no joint authentication and encryption scheme specifically designed to exploit / adapt to the coding structure of HEVC. As such, the proposed joint authentication and encryption scheme is compared with three similar conventional joint approaches designed for different standards / domains. Here, approaches that utilize information hiding technique to realize fingerprinting, watermarking and authentication are considered. The first approach was proposed by Kundur et al. (Kundur & Karthik, 2004), where the fingerprinting (embedded tag) and coefficient scrambling (encryption) processes are performed on H.264/AVC video. They maintain the decrypted video quality and preserve fingerprint imperceptibility after decryption, but the decryption key is susceptible to collusion attack. The second approach was proposed by Zhang. (X. Zhang, 2012), where the tag is embedded in an encrypted (raw) image by utilizing the sparse space vacated by the proposed LSB compression technique. However, this approach can only extract the embedded tag before the decryption process, i.e., the embedded tag is lost after decryption. The third approach was proposed by Rad et al. (Rad et al., 2014) where the predicted pixel values are replaced by the tag to be embedded. This approach requires additional bits to store the prediction errors and the embedded tag is

Table 6.2: Functional Comparison among Joint Schemes

Function	(Kundur & Karthik, 2004)	(X. Zhang, 2012)	(Rad et al., 2014)	✠
Extract info. in encrypted domain	●	●	●	●
Maintain quality after decryption	●	▲	●	●
Applicable in video domain	▲	▲	×	●
Maintain info. after decryption	▲	×	×	●

● : fully functional, ▲ : partially functional, × : not functional,
✠ : Proposed joint scheme

lost after decryption.

Table 6.2 compares the proposed and conventional joint authentication and encryption schemes. Here, "●", "▲" and "×" indicate that the scheme is completely, partially, or not in line with the function of interest, respectively. All schemes are able to extract the embedded tag in the encrypted domain (e.g., image) but only (Kundur & Karthik, 2004) and the proposed schemes offer the tag extraction functionality after decryption. Therefore, the proposed joint authentication and encryption scheme serves as an alternative solution to secure video content in both encrypted and decrypted video. Last but not least, the viability of the conventional image-based joint schemes in the video domain is also compared. It is concluded that the joint schemes (X. Zhang, 2012) and (Rad et al., 2014) could be ported directly to the video domain (i.e., dealing with coefficients instead of pixel values), but it is expected that the bitstream size of the encrypted-and-tag-embedded video to expand significantly, since the coding structure of HEVC is not considered.

6.5 Summary

A joint authentication and encryption scheme was proposed for HEVC compressed video. By considering different elements for tag embedding and video encryption purposes, the encrypted video maintained format compliance, at the same time, it could be authenticated before (i.e., in encrypted domain) and after the decryption process.

CHAPTER 7 : CONCLUSION

In this chapter, a conclusion of this study is summarized by recapturing the contribution of research outcome, together with the advantages and disadvantages of the proposed joint scheme. The future work is described to ensure the on-going research continues with the intention of carrying forward the research objectives by contributing to the research community and ultimately to the society.

7.1 Summary

An information hiding technique is proposed to embed information by exploiting CU structure in HEVC video stream. Then, an authentication scheme is put forwarded by utilizing this technique to verify the video authenticity based in two ways: with and without secret key. Next, a video encryption scheme is presented to support the proposed authentication scheme and formed a joint encryption and authentication scheme. In each design, their applicability and effectiveness are justified by utilizing several classes of test video sequences, with respect to the HEVC reference software. Finally, the contribution, advantages and disadvantages of the proposed joint scheme are presented.

7.2 Achievement and Contribution

This study has achieved its objectives:

1. The security application of information hiding is enabled in the current state-of-the-art video compression standards, i.e., an information hiding technique is proposed to realize authentication in HEVC standard in Chapter 3.

2. The performance of proposed information hiding techniques in protecting video integrity is evaluated, particularly for authentication purposes, i.e., the performance, video quality and computational time of proposed authentication scheme by using

various class of video test sequences are analyzed in Chapter 4.

3. Recommendation on designing video authentication by utilizing information hiding technique are presented based on the required properties, i.e., two layers of authentication are introduced to provide flexibility on detecting genuineness of received video, as mentioned in Chapter 4.3.4.

4. The designed authentication scheme is enhanced to operate in encrypted domain, i.e., selective encryption techniques are introduced to form a joint authentication and encryption scheme with the purpose of applying authentication features in encrypted and decrypted video in Chapter 5 and 6.

Meanwhile, the designed scheme has accomplished the following:

1. Advances the research in video authentication based on information hiding technique for achieving higher video quality and capacity while suppressing complexity.

   – The proposed joint authentication and encryption scheme utilizes the proposed information hiding technique in Chapter 3. Quality of authenticated video is observed to experience negligible perceptual quality degradation, which is unnoticeable by using naked eyes. The proposed technique provides sufficient embedding capacity to embed tag for verification purpose. Meanwhile, it produces minimal overhead in terms of time complexity for verifying the video authenticity.

2. Realizes invented video authentication with security features for detecting forged video content and identifying the source of the video.

   – In Chapter 4, multi-layer authentication scheme is designed to verified the video integrity in three layers: first layer provides an surface verification with

minimal parsing overhead; second layer detects tampering region in video, and; third layer identifies the video source by comparing the hash value of extracted features and authentication code in each video slice. With this design, the proposed scheme is capable to detect forged video content and verify the video authority.

3. Enables encryption based on invented video authentication scheme specifically in the field of video coding.
   - The presented authentication scheme in Chapter 4 is enhanced by implementing encryption on authenticated video to form a joint scheme, as mentioned in Chapter 6. The proposed encryption techniques on authenticated video which enables authenticity verification in encrypted and decrypted (i.e., original with authenticate code only) video is realized by the proposed joint authentication and encryption scheme.

7.3 Advantages and Disadvantages

The outcome of this research has following advantages:

1. The release of new standard is probably lack of architecture and design to provide sufficient security and privacy protection in HEVC video stream. Therefore, the designed scheme is proposed at the right time for protecting the integrity of video stream.

2. The designed authentication scheme provides two layers verification to serve independent receivers with different authority or access right to the received video.

However, there are some shortcomings as mentioned below:

1. Designer has to fully understand the complete architecture of the video encod-

ing/decoding process in order to implement the proposed scheme in video encoder/decoder.

2. Many sophisticated processes are involved to establish HEVC hardware implementation. Thus, the hardware encoder is rather rigid and it does not allow any modification for achieving video authentication or encryption scheme.

7.4 Future Works

A HEVC real-time encode-decode prototype is presented by Springer et al. based on Python programming under linux platform (Springer et al., 2014). By utilizing the similar implementation, the proposed joint authentication and encryption scheme can potentially be deployed for real-time applications. Next, along with the release of latest standard, more opportunities are yet to be discovered for realizing authentication, encryption as well as other applications that carry specific features (e.g., security, compression). Hence, the proposed joint scheme has the potential to be implemented in the latest HEVC standard extension (i.e., multi-layer video coding and 3D video coding).

Technically, to enhance the video integrity protection, the proposed joint scheme can be jointly deployed with other information hiding techniques in different domain (e.g., audio layer). For instance, tags can be embedded in audio and video layers to authenticate among the layers in temporal axis. Moreover, the proposed joint scheme design is closely related to the video coding standard, i.e., exploiting the coding structure to realize video authentication and encryption. Therefore, it can be recommended/proposed as a part of the video coding design during the video standardization process. Meanwhile, the proposed joint scheme can be endorsed by law enforcement agencies to enhance the existing video forensic policy, particularly for the utilization of authenticated video as an evidence in court cases (Ariffin & Ishak, 2008).

REFERENCES

Abomhara, M., Zakaria, O., & Khalifa, O. O. (2010, Feb.). An overview of video encryption techniques. In *International Journal of Computer Theory and Engineering* (pp. 103–110).

Aly, H. (2011, Mar.). Data hiding in motion vectors of compressed video based on their associated prediction error. *IEEE Transactions on Information Forensics and Security*, *6*(1), 14–18.

Ariffin, A. F. M., & Ishak, I. I. (2008, Aug.). Digital Forensic in Malaysia. In *Digital Evidence and Electronic Signature Law Review* (pp. 161–165).

Atrey, P. K., Saddik, A. E., & Kankanhalli, M. (2009). *Digital video authentication* (S. Lian & Y. Zhang, Eds.). Hershey, United State of America: Hershey: IGI Global.

Atrey, P. K., Yan, W.-Q., & Kankanhalli, M. S. (2006, May). A scalable signature scheme for video authentication. *Multimedia Tools and Applications*, *34*(1), 107–135.

Austin, A., Barnard, J., & Hutcheon, N. (2015, Jul.). *Online video forecast 2015* (Newcast No. 2059-6685). London, United Kingdom: ZenithOptimedia. Retrieved from `http://www.mumbrella.asia/content/uploads/2015/08/Online-Video-Forecasts-20154.pdf`

Baek, J., ji Byon, Y., Hableel, E., & Al-Qutayri, M. (2013, Oct.). An authentication framework for automatic dependent surveillance-broadcast based on online/offline identity-based signature. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 358–363).

Barni, M., Bartolini, F., & Checcacci, N. (2005, Feb.). Watermarking of MPEG-4 video objects. *IEEE Transaction on Multimedia*, *7*(1), 23–32.

Bjøntegaard, G. (2001, Apr.). *VCEG-M33: Calculation of average PSNR differences between RD curves* (Tech. Rep.). Austin, Texas, United State of America: Video Coding Experts Group (VCEG).

Bjøntegaard, G., & Lillevold, K. (2002, May). Context-adaptive VLC (CAVLC) coding of coefficients [ISO/IEC]. *JVT-C028, 3rd Meeting: Fairfax, Virginia, United State of America*.

Caciula, I., & Coltuc, D. (2014, May). Improved control for low bit-rate reversible watermarking. In *IEEE International Conference on Acoustic, Speech, and Signal Processing* (pp. 7425–7429).

Canny, J. (1986, Nov.). A computational approach to edge detection. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, *8*(6), 679–698.

Cao, Y., Zhao, X., & Feng, D. (2012, Jan.). Video steganalysis exploiting motion vector reversion-based features. *IEEE Signal Processing Letter*, *19*(1), 35–38.

Casey, E. (Ed.). (2011). *Digital evidence and computer crime: Forensic sciences, computers and the internet (3rd edition)*. United State of America: Elsevier Inc.

Chen, Q., Maitre, H., & Deng, Q.-P. (2012, Jan.). Reliable information embedding for image/video in the presence of lossy compression. *Signal Processing: Image Communication*, *27*(1), 66–74.

Chung, K.-L., Huang, Y.-H., Chang, P.-C., & Liao, H.-Y. (2010, Nov.). Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, *20*(11), 1643–1647.

Cross, D., & Mobasseri, B. G. (2002). Watermarking for self-authentication of compressed video. In *IEEE International Conference on Image Processing* (Vol. 2, pp. 913–916).

Dai, Y., Zhang, L., & Yang, Y. (2003, Apr.). A new method of MPEG video watermarking technology. In *International Conference on Communication Technology* (Vol. 2, pp. 1845–1847).

Deng, Y., Wu, Y., Duan, H., & Zhou, L. (2013, Jul.). Digital video steganalysis based on motion vector statistical characteristics. *Optik - International Journal for Light and Electron Optics*, *124*, 1705–1710.

Du, R., & Fridrich, J. (2002). Lossless authentication of mpeg-2 video. In *IEEE International Conference on Image Processing* (Vol. 2, pp. 893–896).

Dumbere, D. M., & Janwe, . N. J. (2014, Jul.). Video encryption using aes algorithm. In *International Conference on Current Trends in Engineering and Technology* (p. 332-337).

Furht, B., & Kirovski, D. (Eds.). (2006). *Multimedia encryption and authentication techniques and applications*. United State of America: Auerbach Publications.

Guo, J.-M., & Tsai, J.-J. (2012, Sep.). Reversible data hiding in low complexity and high quality compression scheme. *Digital Signal Processing*, *22*(5), 776–785.

Guo, Y., & Pan, F. (2010, Dec.). Information hiding for H.264 in video stream switching application. In *IEEE International Conference on Information Theory and Information Security* (pp. 419–421).

He, D., Sun, Q., & Tian, Q. (2003, Apr.). An object based watermarking solution for mpeg4 video authentication. In *IEEE International Conference on Acoustic, Speech, and Signal Processing* (Vol. 3, pp. 537–540).

He, D., Sun, Q., & Tian, Q. (2004). A secure and robust object-based video authentication system. *EURASIP Journal on Applied Signal Processing*, *14*, 2186–2200.

High Efficiency Video Coding: HEVC software repository. (2013). Fraunhofer Heinrich Hertz Institute. Retrieved from `https://hevc.hhi.fraunhofer.de`

Hofbauer, H., Uhl, A., & Unterweger, A. (2014, May). Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption. In *IEEE International Conference on Acoustic, Speech, and Signal Processing* (pp. 1986–1990).

Holliman, M., & Memon, N. (2000, Mar.). Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Transactions on Image Processing*, *9*(3), 432–441.

Hong, W., T-S, C., & Wu, H.-Y. (2012, Apr.). An improved reversible data hiding in encrypted image using side match. *IEEE Signal Processing Letter*, *19*(4), 199–202.

Hu, Y., Zhang, C., & Su, Y. (2007, Jul.). Information hiding based on intra prediction modes for H.264/AVC. In *IEEE International Conference on Multimedia and Expo* (pp. 1231–1234).

Huang, J., & Shi, Y. Q. (2002, Oct.). Reliable information bit hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *12*(10), 916–920.

ISO. (1993). *Information technology – Cdoing of moving pictures and associated audio for digital storage media at up to about 1, 5 mbit/s – part 2: Video* (ISO/IEC No. 11172-2:1993). Geneva, Switzerland: International Organization for Standardization.

ISO. (2000). *Information technology – Generic coding of moving pictures and associated audio information: Video* (ISO/IEC No. 13818-2:2000). Geneva, Switzerland: International Organization for Standardization.

ISO. (2010). *Information technology – Coding of audio-visual objects – Part 1: Systems* (ISO/IEC No. 14496-1:2010). Geneva, Switzerland: International Organization for Standardization.

ISO. (2013). *Information technology – High efficiency coding and media delivery in heterogeneous environments – part 2: High efficiency video coding* (ISO/IEC No. 23008-2:2013). Geneva, Switzerland: International Organization for Standardization.

Jordan, F., Kutter, M., & Ebrahimi, T. (1997). Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video. In *ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio.*

Kang, L.-W., & Leou, J.-J. (2005, Feb.). An error resilient coding scheme for H.264/AVC video transmission based on data embedding. *Journal of Visual Communication and Image Representation*, *16*(1), 93–114.

Kapotas, S. K., & Skodras, A. N. (2008, Apr.). A new data hiding scheme for scene change detection in H.264 encoded video sequences. In *IEEE International Conference on Multimedia and Expo Workshops* (pp. 277–280).

Kim, D.-W., Choi, Y.-G., Kim, H.-S., Yoo, J.-S., Choi, H.-J., & Seo, Y.-H. (2010, Aug.). The problems in digital watermarking into intra-frames of H.264/AVC. *Image and Vision Computing*, *28*(8), 1220–1228.

Kim, S., Kim, S., Hong, Y., & Won, C. (2007). Data hiding on H.264/AVC compressed video. In M. Kamel & A. Campilho (Eds.), *Image Analysis and Recognition* (Vol. 4633, pp. 698–707). Springer Berlin Heidelberg.

Krzyzanowski, P. (1997). Cryptographic communication and authentication. In *CS 417: Distributed Systems* (pp. 1–25). Rutgers University.

Kundur, D., & Karthik, K. (2004, Jun.). Video fingerprinting and encryption principles for digital right management. *Proceeding of IEEE*, *92*(6), 918–932.

Lang, A., Thiemert, S., Hauer, E., Liu, H., & Petitcolas, F. A. P. (2003, Jun.). Authentication of mpeg-4 data: risks and solutions. *Proceeding of Security and Watermarking of Multimedia Contents V*, *5020*.

Li, C., Zhou, X., & Zong, Y. (2008). Nal level encryption for scalable video coding. In *Proceeding on PCM* (pp. 496–505).

Li, G., Ito, Y., Yu, X., Nitta, N., & Babaguchi, N. (2009, Jan.). Recoverable privacy protection for video content distribution. *EURASIP Journal on Information Security*, 4:1–4:11.

Lian, S., Liu, Z., Ren, Z., & Wang, H. (2007, Jun.). Commutative encryption and watermarking in video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, *17*, 774–778.

Liao, K., Lian, S., Guo, Z., & Wang, J. (2010, Jun.). Efficient information hiding in H.264/AVC video coding. *Telecommunication Systems*, *49*(2), 261–269.

Lin, T.-J., Chung, K.-L., Chang, P.-C., Huang, Y.-H., Liao, H.-Y. M., & Fang, C.-Y. (2013, Mar.). An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames. *Journal of System and Software*, *86*, 604–614.

List, P., Joch, A., Lainema, J., Bjøntegaard, G., & Karczewicz, M. (2003, Jul.). Adaptive deblocking filter. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(7), 614–619.

Liu, J. K., Baek, J., Zhou, J., & Yang, Y. (2010, Aug.). Efficient online/offline identity-based signature for wireless sensor network. *Iinternational Journal of Iinformation Security*, *9*(4), 287–296.

Lo, S.-W., Wei, Z., Ding, X., & Deng, R. H. (2014, Jul.). Generic attacks on content-based video stream authentication. In *IEEE International Conference on Multimedia and Expo Workshops* (pp. 1–6).

Lu, C.-S., Chen, J.-R., & Fan, K.-C. (2005, Aug.). Real-time frame-dependent video watermarking in VLC domain. *Signal Processing: Image Communication*, *20*(7), 624–642.

Ma, X., Li, Z., Tu, H., & Zhang, B. (2010, Oct.). A data hiding algorithm for H.264/AVC video streams without intra-frame distortion drift. *IEEE Transactions on Circuits and Systems for Video Technology*, *20*(10), 1320–1330.

Maillard, J. (2009, Mar.). *Second emmy award for iso/iec mpeg-4 avc standard.* International Organization for Standardization (ISO). Retrieved from `http://www.iso.org/iso/home/news_index/news_archive/news.htm?refid=Ref1213`

Marpe, D., Schwarz, H., & Wiegand, T. (2003, Jul.). Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(7), 620–636.

Martina Podesser, A. U., Hans-peter Schmidt. (2002). Selective bitplane encryption for secure transmission of image data in mobile environments. In *5th Nordic Signal Processing Symposium* (Vol. 10, pp. 4–6).

Marvel, L. M., Jr., C. G. B., & Retter, C. T. (1999, Aug.). Spread spectrum image steganography. *IEEE Transactions on Image Processing*, *8*, 1075–1083.

Massoudi, a., Lefebvre, F., De Vleeschouwer, C., Macq, B., & Quisquater, J. J. (2008, Nov.). Overview on selective encryption of image and video: Challenges and perspectives. *EURASIP Journal on Information Security*, *2008*(179290), 1–18.

Meuel, P., Chaumont, M., & Puech, W. (2007, Sep.). Data hiding in H.264 video for lossless reconstruction of region of interest. In *EUSIPCO 07: 15th European Signal Processing Conference* (pp. 2301–2305). Poznan, Poland: EURASIP.

Mobasseri, B. G., & Marcinak, M. P. (2005, May). Watermarking of MPEG-2 video in compressed domain using VLC mapping. In *Proceedings of the 7th Workshop on Multimedia and Security* (pp. 91–94). New York, NY, USA: ACM.

Mobasseri, B. G., Sieffert, M. J., & Simard, R. J. (2000, Sept.). Content authentication and tamper detection in digital video. In *IEEE International Conference on Image Processing* (Vol. 1, pp. 458–461).

Moorthy, A. K., & Bovik, A. C. (2011, Dec.). Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Transactions on Image Processing*, *20*(12), 3350–3364.

Nakajima, K., Tanaka, K., Matsuoka, T., & Nakajima, Y. (2005, Jul.). Rewritable data embedding on mpeg coded data domain. In *IEEE International Conference on Multimedia and Expo* (pp. 682–685).

NIST. (1999). *Data Encryption Standard (DES)* (FIPS PUB No. 46-3). Gaithersburg, Maryland, United State of America: National Institute of Standards and Technology.

NIST. (2001). *Advanced Encryption Standard (AES)* (FIPS PUB No. 197). Gaithersburg, Maryland, United State of America: National Institute of Standards and Technology.

NIST. (2002). *Secure Hash Standard* (FIPS PUB No. 180-2). Gaithersburg, Maryland, United State of America: National Institute of Standards and Technology.

Patra, B., & Patra, J. C. (2012, Mar.). CRT-based self-recovery watermarking technique for multimedia applications. In *IEEE International Conference on Acoustic, Speech, and Signal Processing* (pp. 1761–1764).

Queluz, M. P. (1998, Dec.). Towards robust, content based techniques for image authentication. In *IEEE 2nd Workshop on Multimedia Signal Processing* (Vol. 1, pp. 297–302).

Rad, R. M., Wong, K., & Guo, J.-M. (2014, Apr.). A unified data embedding and scrambling method. *IEEE Transaction on Image Processing*, *23*(4), 1463–1475.

Ren, Y. J., & O'Gorman, L. (2012, Nov.). Accuracy of a high-level, loss-tolerant video fingerprint for surveillance authentication. In *International Conference on Pattern Recognition* (pp. 1088–1091).

Ren, Y. J., O'Gorman, L., Wu, L. J., Chang, F., Wood, T. L., & Zhang, J. R. (2013, Oct.). Authenticating lossy surveillance video. *IEEE Transactions on Information Forensics and Security*, *8*(10), 1678–1687.

Rivest, R. L. (1991). *Ssecurity Architecture for Open Systems Interconnection (OSI); Security, Structure and Applications* (ITU/T No. X.800). Geneva, Switzerland: The International Telergraph and Telephone Consultative Committee (CCITT).

Rivest, R. L. (1992). *The MD5 Message-Digest Algorithm* (RSA No. RFC 1321). Cambridge, Massachusetts, United State of America: Massachusetts Institute of Technology, Laboratory for Computer Science.

Roy, S. D., Li, X., Shoshan, Y., Fish, A., & Yadid-Pecht, O. (2013, Feb.). Hardware implementation of a digital watermarking system for video authentication. *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(2), 289–301.

Seo, Y.-H., Choi, H.-J., Lee, C.-Y., & Kim, D.-W. (2008, Aug.). Low-complexity watermarking based on entropy coding in H.264/AVC. *IEICE Transaction on Fundamentals of Electronics, Communication and Computer Science*, *E91-A*(8), 2130–2137.

Shah, J., & Saxena, V. (2011, Mar.). Video encryption: A survey. In *International Journal of Computer Science Issues* (Vol. 8, pp. 525–534).

Shahid, Z., Chaumont, M., & Puech, W. (2011, Apr.). Considering the reconstruction loop for data hiding of intra- and inter-frames of H.264/AVC. *Signal, Image and Video Processing*, 1–19.

Shahid, Z., & Puech, W. (2014, Jan.). Visual protection of HEVC video by selective encryption of CABAC binstrings. *IEEE Transactions on Multimedia*, *16*(1), 24–36.

Shanableh, T. (2012a, Apr.). Data hiding in MPEG video files using multivariate regression and flexible macroblock ordering. *IEEE Transactions on Information Forensics and Security*, *7*(2), 455–464.

Shanableh, T. (2012b, Oct.). Matrix encoding for data hiding using multilayer video coding and transcoding solutions. *Signal Processing: Image Communication*, *27*, 1025–1034.

Sharp, A. T., Devaney, J., Steiner, A. E., & Peng, D. (2010, Dec.). Digital video authentication with motion vector watermarking. In *4th International Conference on Signal Processing and Communication System* (pp. 1–4).

Shirey, R. W. (2000, May). *Internet Security Glossary* (ISD No. RFC 2828). Virginia, United State of America: GTE / BBN Technologies.

Sobel, I., & Feldman, G. (1968). *A 3x3 isotropic gradient operator for image processing.* Stanford Artificial Intelligence Project (SAIP). Retrieved from http://www.researchgate.net/publication/239398674_An_Isotropic_3_3_Image_Gradient_Operator

Song, L., Wang, J., Wang, L., & Chen, J. (2013, May.). A low-cost authenticating mechanism for interactive video streaming. In *International Conference on Software Engineering and Service Science* (pp. 422–425).

Springer, D., Schnurrer, W., Weinlich, A., Heindel, A., Seiler, J., & Kaup, A. (2014, Oct.). Open source HEVC analyzer for rapid prototyping (HARP). In *21st IEEE International Conference on Image Processing* (pp. 2189–2191).

Stütz, T., & Uhl, A. (2009). On JPEG2000 Error Concealment Attack. In T. Wada, F. Huang, & S. Lin (Eds.), *Proceedings of the 3rd pacific rim symposium on advances in image and video technology* (Vol. 5414, pp. 851–861). Springer Berlin Heidelberg.

Su, P.-C., Wu, C.-S., Chen, I.-F., Wu, C.-Y., & Wu, Y.-C. (2011, Oct). A practical design of digital video watermarking in H.264/AVC for content authentication. *Signal Processing: Image Communication*, *26*, 413–426.

Su, Y., Zhang, C., & Zhang, C. (2011, Aug.). A video steganalytic algorithm against motion-vector-based steganography. *Signal Processing*, *91*(8), 1901–1909.

Sullivan, G. J., Ohm, J.-R., Han, W.-J., & Wiegand, T. (2012, Dec.). Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(12), 1649–1668.

Sullivan, G. J., & Wiegand, T. (1998, Nov.). Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, *15*, 74–90.

Sze, V., Budagavi, M., & Sullivan, G. J. (Eds.). (2014). *High efficieny video coding (hevc): Algorithms and architectures*. Springer International Publishing.

Tartary, C., Wang, H., & Ling, S. (2011, Sep.). Authentication of digital streams. *IEEE Transactions on Information Theory*, *57*(9), 6285–6303.

Thiesse, J.-M., Jung, J., & Antonini, M. (2010a, Sep.). Data hiding of intra prediction information in chroma samples for video compression. In *17th IEEE International Conference on Image Processing* (pp. 2861–2864).

Thiesse, J.-M., Jung, J., & Antonini, M. (2010b, Oct.). Data hiding of motion information in chroma and luma samples for video compression. In *IEEE International Workshop on Multimedia Signal Processing* (pp. 217–221).

Thiesse, J.-M., Jung, J., & Antonini, M. (2011, Jun.). Rate distortion data hiding of motion vector competition information in chroma and luma samples for video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(6), 729–741.

Upadhyay, S., & Singh, S. K. (2011, Nov.). Learning based video authentication using statistical local information. In *International Conference on Image Information Processing* (pp. 1–6).

Vanne, J., Viitanen, M., & Hämäläinen, T. D. (2014, Sep.). Efficient mode decision schemes for HEVC inter prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(9), 1579–1593.

Waddilove, R. (2015, May). *Best free video editing software 2015 uk.* PC Advisor. Retrieved from http://www.pcadvisor.co.uk/test-centre/photo-video/14-best-free-video-editing-software-2015-uk-3512375/

Wang, R., Hu, L., & Wu, D. (2011, Jun.). A watermarking algorithm based on the CABAC entropy coding for H.264/AVC. *Journal of Computational Information System*, *7(6)*, 2132–2141.

Wang, X., Zheng, N., & Tian, L. (2010, Mar.). Hash key-based video encryption scheme for h.264/avc. In *Signal Processing : Image Communication* (pp. 427–437).

Wang, Y., O'Neill, M., & Kurugollu, F. (2013, Sep.). A tunable encryption scheme and analysis of fast selective encryption for cavlc and cabac in h.264/avc. *IEEE Transactions on Circuits and Systems for Video Technology*, *23*(9), 1476–1490.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004, Apr.). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612.

Watson, A. B. (1993, Sep.). DCT quantization matrices visually optimized for individual images. In *Storage and retrieval for image and video databases.*

Wedi, T. (2002). Adaptive interpolation filter for motion compensated prediction. In *IEEE International Conference on Image Processing* (Vol. 2, pp. 509–512).

Wei, Z., Wu, Y., Deng, R. H., & Ding, X. (2014, Apr.). A hybrid scheme for authenticating scalable video codestreams. *IEEE Transactions on Information Forensics and Security*, *9*(4), 543–553.

Wien, M. (2015). *High efficieny video coding: Coding tools and specification*. Springer-Verlag Berlin Heidelberg.

Williams, J. (2012). *ACPO good practice guide for digital evidence* (Tech. Rep.). England, Wales and Northern Ireland: Association of Chief Police Officers.

Wong, K., & Tanaka, K. (2007). A data hiding method using Mquant in MPEG domain. *IIEEJ Proceeding of Image Electronics and Visual Computing Workshop*.

Wong, K. S., Tanaka, K., Takagi, K., & Nakajima, Y. (2009, Oct.). Complete video quality - preserving data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *19*(10), 1499–1512.

Wu, C. W. (2002, Sep.). On the design of content-based multimedia authentication systems. *IEEE Transactions on Multimedia*, *4*(3), 385–393.

Wu, M., & Liu, B. (2003, Jun.). Data hiding in image and video I - Fundamental issues and solutions. *IEEE Transactions on Image Processing*, *12*(6), 685–695.

Wu, M., Yu, H., & Liu, B. (2003, Jun.). Data Hiding in Image and Video II - Designs and Applications. *IEEE Transactions on Image Processing*, *12*(6), 696–705.

Wu, M.-N., Lin, C.-C., & Chang, C.-C. (2008, Sep.). An embedding technique based upon block prediction. *Journal of System and Software*, *81*(9), 1505–1516.

Xu, C., Ping, X., & Zhang, T. (2006, Sep.). Steganography in compressed video stream. In *1st International Conference on Innovative Computing, Information and Control* (Vol. 1, pp. 269–272).

Xu, D., & Wang, R. (2011, Sep.). Watermarking in H.264/AVC compressed domain using Exp-Golomb code words mapping. *Optical Engineering*, *50*(9).

Xu, D., Wang, R., & Wang, J. (2010, Aug.). Prediction mode modulated data-hiding algorithm for H.264/AVC. *Journal of Real-Time Image Processing*, 1–10.

Xu, D., Wang, R., & Wang, J. (2011, Jul.). A novel watermarking scheme for H.264/AVC video authentication. *Signal Processing: Image Communication*, *26*(6), 267–279.

Yang, G., Li, J., He, Y., & Kang, Z. (2011, Apr.). An information hiding algorithm based on intra-prediction modes and matrix coding for H.264/AVC video stream. *International Journal of Electronics and Communication (AE U*, *65*(4), 331–337.

Yilmaz, A., & Alatan, A. (2003, Sep.). Error concealment of video sequences by data hiding. In *IEEE International Conference on Image Processing* (Vol. 2, pp. 679–682).

Yin, P., Liu, B., & Yu, H. (2001, May). Error concealment using data hiding. In *IEEE International Conference on Acoustic, Speech, and Signal Processing* (Vol. 3, pp. 1453–1456).

YUV sequences repository. (2013). Institut für Informationsverarbeitung. Retrieved from `ftp://hvc:US88Hula@ftp.tnt.uni-hannover.de/testsequences`

Zeng, W., & Dong, L. (2008). End-to-end security for multimedia adaptation. In B. Furht (Ed.), *Encryption and Authentication of H.264 Video* (pp. 206–211). Boston, MA: Springer US.

Zhang, J., & Ho, A. T. S. (2006, Aug.). Efficient video authentication for H.264/AVC. In *1st International Conference on Innovative Computing, Information and Control* (Vol. 3, pp. 46–49).

Zhang, J., Li, J., & Zhang, L. (2001, Oct.). Video watermark technique in motion vector. In *Proceeding of XIV Brazilian Symposium on Computer Graphics and Image Processing* (pp. 179–182).

Zhang, X. (2012, Apr.). Separable reversible data hiding in encrypted image. *IEEE Transactions on Information Forensics and Security*, *7*(2), 826–832.

Zhu, H., Wang, R., & Xu, D. (2010, May). Information hiding algorithm for H.264 based on the motion estimation of quarter-pixel. In *2nd International Conference on Future Computer and Communication* (Vol. 1, pp. 423–427).

Zhu, H., Wang, R., Xu, D., & Zhou, X. (2010, Oct.). Information hiding algorithm for H.264 based on the predition difference of intra 4X4. In *3rd International Congress on Image and Signal Processing* (Vol. 1, pp. 487–490).

Appendices

APPENDIX A : LIST OF PUBLICATIONS AND PAPERS PRESENTED

Journal Papers

- Tew, Y., & Wong, K. (2014a, Feb.). An overview of information hiding in H.264AVC compressed video. IEEE Transactions on Circuits and Systems for Video Technology, 24(2), 305–319.
- Tew, Y., Wong, K., Phan, R. C.-W., & Ngan, K. N. (2016, Jul.). Multi-layer Authentication Scheme for HEVC Video based on Embedded Statistics. Journal of Visual Communication and Image Representation, IN PRESS.
- Tew, Y., Wong, K., & Phan, R. C.-W. (2016, Apr.). Separable Authentication in Encrypted HEVC Video. Signal Processing : Image Communication (under review).

Conference Papers

- Tew, Y., & Wong, K. (2012, Nov.). A Survey of Information Hiding in H.264/AVC. IIEEJ 3rd Image Electronics and Visual Computing Workshop.
- Tew, Y., & Wong, K. (2014b, Oct.). Information hiding in HEVC standard using adaptive coding block size decision. In IEEE International Conference on Image Processing (pp. 5502–5506).
- Tew, Y., Wong, K., & Baskaran, V. M. (2015, Jun.). Dual layer video stream in HEVC through information hiding. In IEEE International Conference on Consumer Electronics - Taiwan (pp. 15–16).
- Tew, Y., & Wong, K. (2015, Sep.). HEVC Video Authentication using Data Embedding Technique. In IEEE International Conference on Image Processing (pp. 1265–1269).

– Tew, Y., Minemura, K., & Wong, K. (2015, Dec.). HEVC selective encryption using transform skip signal and sign bin. In Asia-pacific Signal and Information Processing Association Annual Summit and Conference (pp. 963–970).

– Tew, Y., & Wong, K. (2016, Jun.). Region-of-Interest Encryption in HEVC Compressed Video. In IEEE International Conference on Consumer Electronics - Taiwan (pp. 140–141).

– Tew, Y., Wong, K., & Phan, R. C.-W. (2016, Oct.). Joint Selective Encryption and Data Embedding Technique in HEVC Video. In Asia-pacific Signal and Information Processing Association Annual Summit and Conference (under review).