漫談 H.264/AVC 的選擇加密法

李琪霖、黃偉倫、郭育麟、王炳兼、陳宗和*
國立嘉義大學資訊工程學系
E-mail: thchen@mail.ncyu.edu.tw

摘要

隨著網路的普及以及多媒體資訊的發展，視訊內容的安全性逐漸受到重視，許多視訊加密技術也紛紛被提出。從早期的完全性加密到近年來提出的選擇性加密，足可得知視訊加密對效率與即時性的要求越來越高，然而卻鮮少有文獻內容統整分析這些加密方法。因此本文針對 H.264/AVC 這個最普及的視訊壓縮標準所設計的加密方案，試著將這些方法分類並指出常見的加密位置有哪些、何處最適合加密，並以安全性、高效率、不破壞壓縮率、保持語意格式這四個基本需求來分析及比較這些加密方案。以期能夠對希望接觸這領域或是剛接觸這領域的讀者提供導讀效果。

關鍵字：H.264/AVC、視訊加密、選擇性加密、機密性

Selective Encryption for H.264/AVC

Ci-Lin Li, Wei-Lun Huang, Yu-Lin Kuo, Bing-Jian Wang, Tzung-Her Chen*
Department of Computer Science and Information Engineering, National Chiayi University
E-mail: thchen@mail.ncyu.edu.tw

Abstract

With wide population of networking and rapid development of multimedia, security of multimedia content has drawn more and more attention such that a couple of video encryption schemes have been proposed in the literature. Selective encryption instead of full encryption to multimedia benefits more from efficiency in the real applications. In this article, the survey of H.264/AVC encryption is presented to demonstrate which process in H.264/AVC coding is suitable to encrypt and why it benefits more.

Keywords: H.264/AVC, Video encryption, Selective encryption, Confidentiality

*通訊作者：thchen@mail.ncyu.edu.tw

壹、 前言

隨著科技的日新月異，數位多媒體越來越普及化，加上網路世界急速的蓬勃發展，因此 Moving Picture Experts Group(MPEG)及 Video Coding Experts Group(VCEG)聯合開發了 H.264/AVC 視訊編碼標準[1]。H.264/AVC 視訊編碼標準是目前最有效率的視訊編碼標準，其壓縮效能比先前的視訊壓縮標準要高將近一倍[23]。所以目前有許多的應用使用 H.264/AVC 視訊編碼標準，例如視訊會議、行動視訊應用、標準高畫質數位電視…等。但由於網路普及化的關係，數位多媒體內容的取得變得相當得容易，所以內容機密性(Confidentiality)成為一個相當重要且熱門的議題，因為數位多媒體內容容易被竊取或是被非法的複製並且散播，如此數位內容的授權者及創造者的權益將會受損，因此需要設計安全的機制來保護數位內容的安全。數位權限管理系統(Digital Rights Managements, DRM)是近年來最常被用來保護數位內容不被惡意的散播或複製的方法，而數位權限管理系統可以分為以下四類：數位內容加密(encryption)、數位內容驗證(authentication)、數位浮水印技術(digital watermarking)及金鑰管理(key management)。本篇論文主要探討的議題為數位內容加密，數位內容加密可以有效的保護數位內容達到數位內容的機密性。

視訊編碼標準通常是將一個相當大資料量的視訊原始內容(raw)透過去除冗餘技術來減少檔案大小，以便於減少在網路上的傳輸量或儲存所需的空間。目前普遍所為人所知的傳統加密法(像是 DES, AES[17], RSA[22])是比較不適合用於視訊加密上，因為傳統的加密法原於加密數據文件，因為視訊內容的資料量相較之下顯得相當龐大，因此傳統的加密法無法達到即時性(Real-time)的處理加密或解密的程序。一個理想的數位視訊加密方法必須要達到以下四個需求:

(1)安全性

所謂的安全性不但是要達到人類視覺上的不可視覺(即為混淆數位內容)，而且從密碼學的角度來看，安全性同時也是代表惡意的攻擊者不能在沒有授權(沒有金錢)的情況下將加密後的數位視訊還原為原本的內容。

(2)高效率

因為視訊內容的資料量相當的龐大，所以加密的過程可能會伴隨著高運算複雜度，而無法達到即時性，因此為了達到即時性，加密的演算法必須具有高效率、低運算量的特性。

(3)不破壞壓縮率

因為視訊編碼標準通常是一個相當複雜的演算法，所以加密的過程有可能會影響到視訊編碼的演算法使得壓縮率變差，因此加密的程序不能破壞壓縮效果，使得檔案變大。

(4)保持語意(Element)格式[12]

保持語意格式為加密後的數位內容可以被視訊壓縮標準所解碼，並且可以正常播放，但是其播放結果大多為無意義之雜訊。

所以如何設計一個符合以上四個需求的視訊加密法是目前相當重要的議題。

目前的數位內容(含影像、視訊)的加密方法分為兩類，分別為完全性加密法(complete

encryption method)及選擇性加密法(selective encryption method)。完全性加密法是利用常見的密碼學加密技術，如 AES 或 DES，直接對數位內容進行加密。完全性加密法分為兩種，一種是針對壓縮前對原始的數位內容進行加密[15]，另一種是針對壓縮後的位元串流進行加密[4]，這其中也包括了將位元串流分為兩個部分並且一部份利用傳統加密法來加密，而另一部分則是利用秘密金鑰來進行 XOR 運算的方法[21]。加密未壓縮的數位內容容易破壞內容之間的關連性，因此此類的加密方法具有較高的視覺安全及保持檔案格式的完整性，但是因為打亂掉內容之間的關連性，因此畫面(Frame)及畫面(Frame)間的冗餘也會被破壞，所以會使得壓縮的效率大為降低，在雲端環境下顯得不可行；而加密壓縮後的位元串流同樣也是具有相當高的視覺安全，但是也造成了檔案格式的破壞[9]無法保持語意格式。不論是壓縮前還是壓縮後的完全性的加密法因為需要加密的資料量是相當大的，因此造成加密法的運算複雜度是較高的。選擇性加密法顧名思義就是指加密被選擇出來的係數及在視訊壓縮標準中的重要語意，並且在進行視訊壓縮的過程中也進行加密的動作，此類型的加密演算法的安全性和所選擇加密的語意及係數是息息相關的，不只如此，選擇加密的資料及語意會決定加密演算法是否會影響到效率及保持語意格式。

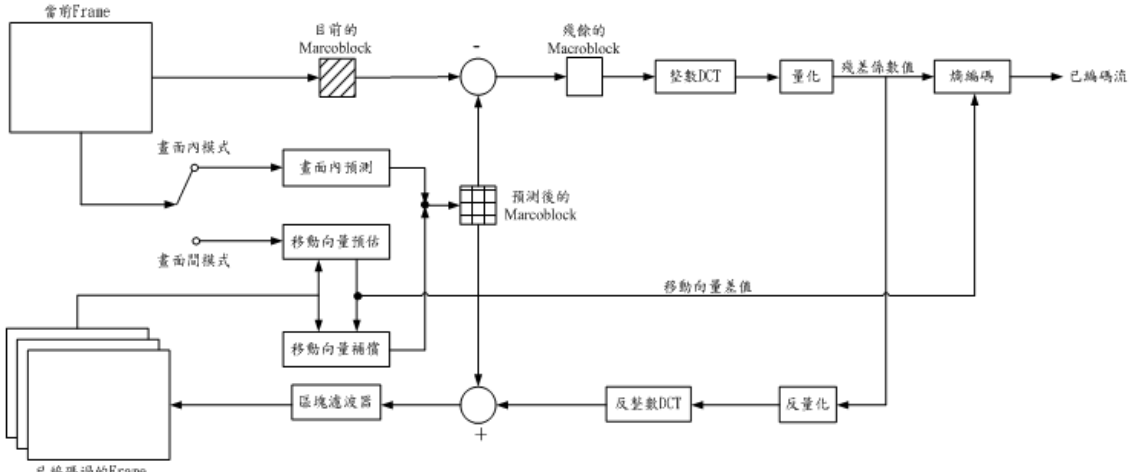
基於以上的分析我們可以知道如何選擇合適的語意與係數來進行加密是相當重要的，但是目前較少有學者完整的分析在 H.264/AVC 中的語意與係數有哪些是敏感且適合拿來做加密的，並且是如何加密的。因此本篇論文主要是以安全性、高效率、不破壞壓縮率、保持語意格式這四個基本需求來分析及比較目前已提出的 H.264/AVC 選擇性加密法中被加密的語意與係數，因此可以讓讀者對於 H.264/AVC 選擇性加密法與視訊加密法有基本的概念。

貳、H.264/AVC 視訊壓縮編碼標準

H.264/AVC 視訊壓縮編碼技術主要分為主要三種技術，分別為去除空間冗餘技術(Spatial redundancy)、去除時間冗餘技術(Temporal redundancy)以及去除編碼冗餘技術(Coding redundancy)，其編碼流程及解碼如圖一[3]。

在編碼流程中，如果目前要進行編碼的是 I-Frame，則必須要先經過畫面內預測(Intra prediction)去除空間冗餘再經過整數 DCT 與量化為殘差係數值，然後再經由熵編碼(Entropy coding)來去除編碼冗餘為編碼流。在編每一個 Frame 時，必須將殘差係數值經過反量化及反 DCT 在和預測後的結果相加，事實上就是經過解碼的動作將目前編碼的 Frame 存進記憶體中已編下一張 Frame 的編碼。因此，如果目前要進行編碼的是 B/P-Frame，則必須從記憶體中取出前一張 Frame 來進行畫面間預測(Inter-prediction)來進行移動相量預估(Motion vector evaluation)與移動相量補償(Motion vector compensated)來得到移動向量差值(Motion vector difference)，然後利用熵編碼將移動向量差值編為編

碼流，另外，經過預測後的 Frame 同樣需要進行整數 DCT 與量化為殘差係數值然後再經由熵編碼來去除編碼冗餘為編碼流，之後同樣要做解碼的動作將 Frame 存入記憶體中。



圖一：H.264/AVC 視訊壓縮編碼流程圖

編碼流程中最重要的去除空間冗餘技術、去除時間冗餘技術以及去除編碼冗餘技術的敘述如下:

(1)去除空間冗餘技術: 在 H.264/AVC 視訊編碼標準中,去除空間的冗餘主要是利用畫面內預測,所謂的畫面內預測是指利用畫面內預測模式去找出與目前編碼區塊最為相似的已編碼區塊，然後被預測後的區塊會與目前編碼區塊進行相減的動作以得到殘餘區塊，然後再經過整數 DCT 轉換編碼、量化的方式以減少在殘餘區塊裡的殘餘係數，當然預測後的區塊與目前編碼區塊越相似其殘餘係數的值也就越小，其殘餘係數的大小也就越趨近於 0，所以編碼資料量也就越小，以達到高壓縮率的效果。

(2)去除時間冗餘技術:去除時間上的冗餘是利用畫面間預測即移動向量預估對目前編碼的畫面與上一次編碼的畫面，利用 7 種不同大小區塊的畫面間預測模式來對不同的畫面複雜度進行預測找出移動向量，然後透過移動向量補償的方式，利用參考的畫面來與目前要編的畫面進行相減的動作求得殘餘區塊來減少目前畫面的編碼資料量，之後再經過整數 DCT 轉換編碼、量化的方式以減少在殘餘區塊裡的殘餘係數，而在 H.264/AVC 中可以選擇前後最多到 31 張的畫面來選擇要參考的畫面，因此可以選擇到與目前預編碼的畫面最相似的參考畫面，因此可以減少殘餘值，減少編碼資料量，以達到高壓縮率的效果。

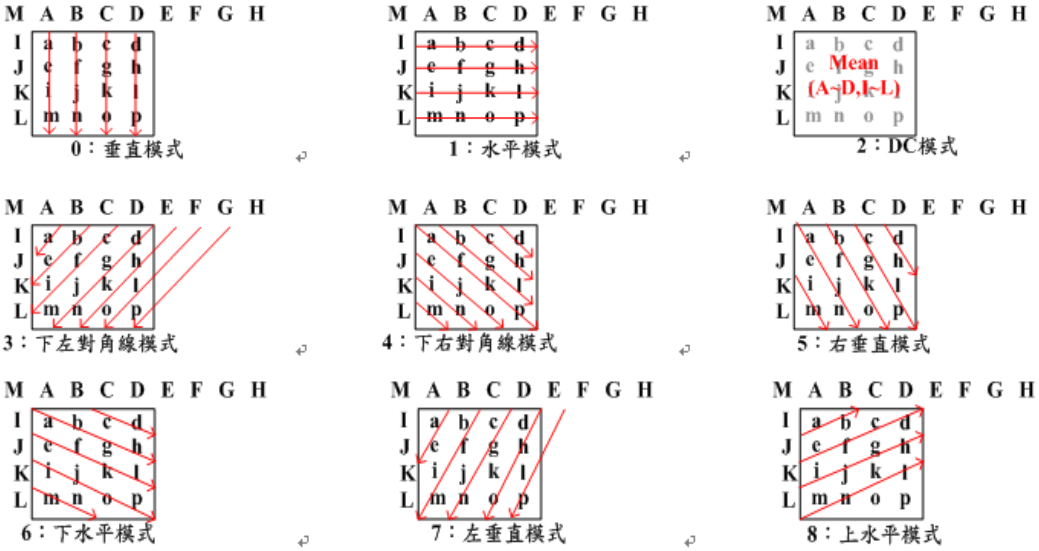
(3)去除編碼冗餘技術:去除編碼冗餘是利用熵編碼(Entropy coding)對殘餘係數與特定的語意進行編碼，編碼後即為編碼流輸出，在這裡所謂的熵編碼是指內文適應性可變動長度編碼 (Context-based Adaptive Variable Length Coding; CAVLC)或內文適應性算術編碼(Context-based Adaptive Binary Arithmetic Coding; CABAC)，而熵編碼會對已量化後的殘

餘係數的特性進行編碼，以達到去除編碼上的冗餘量的目的，並且 H.264/AVC 的熵編碼是可以利用已編碼的資料來動態調整可變動長度的編碼表，以達到高壓縮率的目的。當決定使用 CAVLC 對殘餘係數進行熵編碼時，其語意格式是利用指數哥倫布編碼(Exp-Golomb Codes) [24]與固定長度編碼(Fixed length code)來進行編碼的動作。指數哥倫布編碼比霍夫曼編碼更具有規律性，因此使用指數哥倫布編碼來編碼特定的語意可以減少編碼及解碼的複雜度。固定長度編碼 e 是直接對語意格式進行二元化的轉換。若選擇 CABAC 對殘餘係數進行熵編碼時，其語意格式同樣也是用 CABAC 來進行編碼。

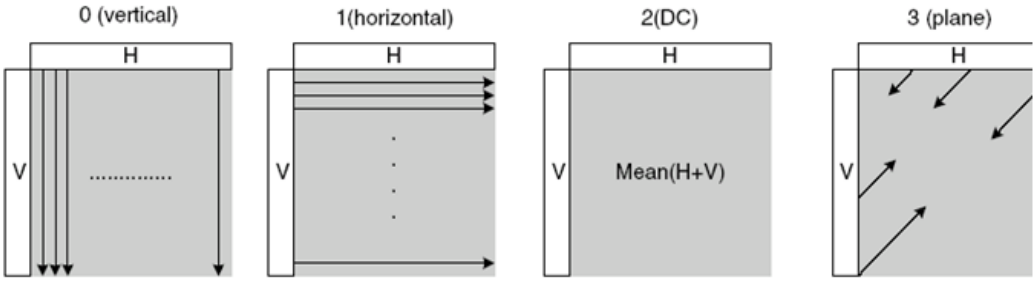
參、目前現有的 H.264/AVC 視訊壓縮編碼標準之選擇性加密法

A. 畫面內預測模式加密法

在 H.264/AVC 中提供 9 種 Intra 4×4 亮度區塊預測模式，4 種 8×8 彩度區塊預測模式與 4 種 16×16 亮度區塊預測模式，如圖二以及圖三，其中 8×8 彩度區塊預測模式與 16×16 亮度區塊預測模式是用相同的預測方向，都是 1)垂直模式 2)水平模式 3)DC 模式 4)平面模式，就只是區塊大小不一樣。



圖二：Intra 4×4 亮度區塊預測模式



圖三：8×8 彩度區塊預測模式與 16×16 亮度區塊預測模式

一般而言,在論文中加密 intra prediction mode(IPM)的方式大致上分成兩種 :用 3 bits 的 Key 將 IPM 的 9 種順序打亂，以及 Golomb code 編碼後半部字碼加密。

(1) 用 3 bits 打亂順序

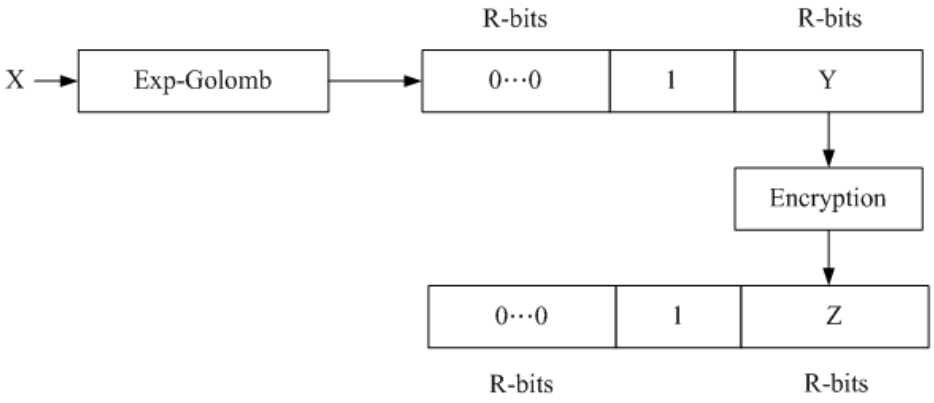
Wang 等學者[28]與 Ahn 等學者[1]利用 3 bits 長度的 Key 來打亂原本的 IPM 值，這種做法也可讓位元率不會增加。其加密的式子如下:

$$IPM_E = IPM \oplus K$$

K 為 3 bits 的 Key，而 IPM_E 為加密後的結果。

(2) Golomb code 編碼後半部字碼加密

在 H.264/AVC 中，IPM 的值最後會在熵編碼(Entropy coding)時用 Exp-Golomb code 來編碼。所以與前述的方法不同的是，加密的位置移到 Exp-Golomb code 時再做處理。然而，為了保持語意格式，必須使用 Length-kept encryption 的加密方式來處理。如下圖四所示，IPM 經過 Exp-Golomb code 編碼後的結果會是 R 個 bits 的 0 加上一個 bit 的 1 與 R 個 bits 的變動數值。其中前面 R+1 個 bits 的值是固定不變的格式，因此在加密時必須避開。因此 Li 等學者[7]與 Lian 等學者[10]的作法是將圖四中 Y 的部份利用 Key 加密為 Z。



圖四：Golomb code 編碼後半部字碼加密

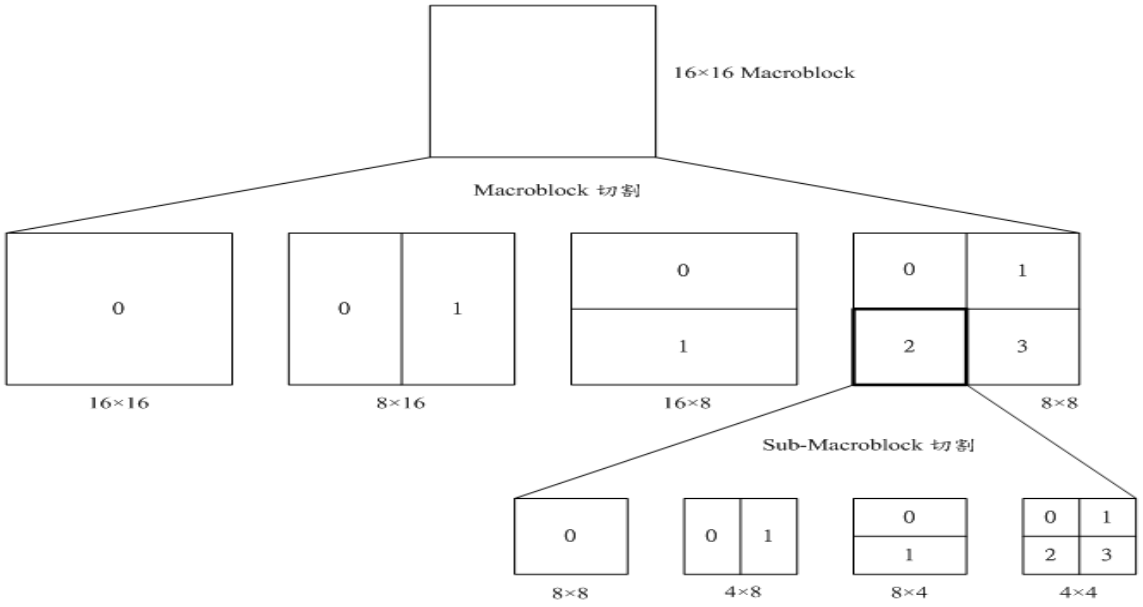
在 Lian 等學者[8][10]的論文中，探討如果只有純粹加密 IPM 的話，其安全性仍然不足，會遭受 Replacement attack [11][20]。意即，不管加密後的 IPM 為何，在解碼端將這些 IPM 皆設為某一定值，如此一來，會能夠看出視訊的一些輪廓，也代表視覺安全性降低。所以，在加密 IPM 的論文中，除了 IPM 外通常還會需要加密其他重要的語意，來達到適當安全性。

B. 畫面間預測模式與移動向量差值加密法

所謂的畫面間預測模式，同樣也是去找出與目前編碼區塊最為相似的已編碼區塊。但是與畫面內預測不同的是，畫面間預測是針對不同張畫面之間去尋找相似區塊，而畫面內預測是在同一張畫面當中去尋找相似區塊。在 H.264/AVC 裡利用七種不同區塊大小來進行移動向量預估與補償，以消除時間上的冗餘如圖五，

在進行畫面間預測時，是將目前編碼中的畫面與前一張畫面進行比較，找出相似的區塊。而對兩張畫面之間相似區塊的座標位置進行差值計算的結果，就稱之為移動向量，但是因為 Inter 區塊相當多，因次 H.264/AVC 就利用 DPCM 算出移動向量差值(Motion vector difference, MVD)以減少編碼量。在 H.264/AVC 視訊編碼標準中，根據移動向量差值，就能推算出畫面中某一編碼區塊在下一張畫面的位置。

利用畫面間預測與移動向量差值的技術，對於重複出現在多張畫面中的相似區塊，只需儲存其移動向量差值，不需重複儲存區塊的詳細資訊，因此能減少編碼後的資料量。在這裡會分別敘述畫面間預測加密法及移動向量差值加密法。



圖五：7 個畫面間預測模式

(1) 畫面間預測加密法

在 Li 等學者[6]的加密方法中，其中的一種加密法是對畫面間預測進行置亂，其置亂的方式為將 4×8 與 8×4 這兩種預測區塊進行調換，因此會改變預測的大小及方向，但是因為有相同數目的移動向量，因此不會破壞語意格式。

(2) 移動向量差值加密法

針對移動向量差值加密的處理方式通常有三種，分別為加密移動向量差值的 Sign、移動向量差值的 Level 以及加密編碼完的移動向量差值的 Bitstream 產生亂數加密。利用函數產生出擬亂的字串，並且以此字串將移動向量差值打亂。

- 加密移動向量差值的 Sign

利用金鑰將移動向量差值的正負號打亂包括 Wang 等學者[26]、Algin 等學者[2]以及 Guo 等學者[5]，也就是加密移動向量差值的 Sign，這種加密方式其實是一種相當有效率的加密方法，因為具有相當程度的視覺安全性，且不會破壞壓縮率。

- 加密移動向量差值的數值

劉等人[13]提出一個基於 H.264/AVC 的加密技術來加密移動向量使得產生錯誤的移動向量差值，其做法如下：

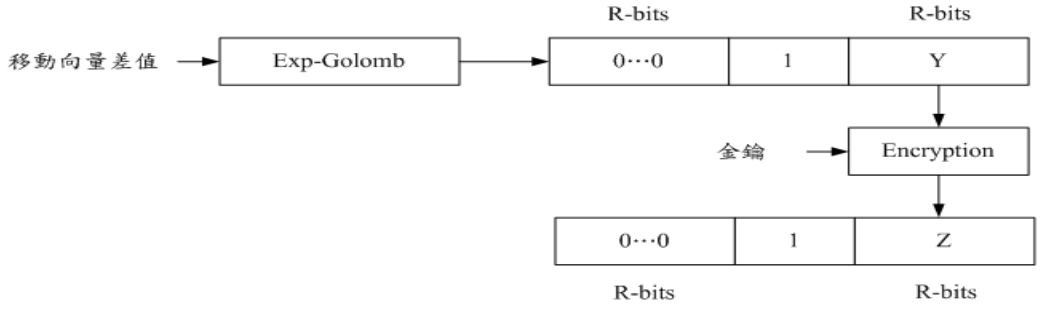
根據金鑰對參數表查找出運算子與隨機數後，把移動向量跟隨機數做運算子的運算，而運算子只包含四則運算，所以讓移動向量值置亂成另一個值以達到的加密效果，因此移動向量的水平值 *MVx* 與垂直值 *MVy* 分別跟隨機數做四則運算，其加密為以下的運算式。

$$\begin{aligned} MV_x^{'} &= MV_x(op)\ Rondown\ number \\ MV_y^{'} &= MV_y(op)\ Rondown\ number \end{aligned}$$ ，(*op* 為四則運算)

而 Magli 等學者[14]是直接利用金鑰對加密移動向量差值的 Level 進行 XOR 運算，因此這種方式的加密具有相當好的安全性，但是是和加密移動向量差值的 Sign 的安全性是差不多的，但是會強烈的破壞壓縮率。

- 加密移動向量差值的 Bitstream

Park 和 Shin[18]、Wang 等學者[28]以及 Wang 等學者[29]是將已經過 Exp-Golomb Codes 編碼的移動向量差值的 Bistream 加密，其加密方法為圖六。



圖六：移動向量差值 Exp-Golomb Codes 加密概念圖

這種加密法，其視覺安全性是和前面所敘述的移動向量差值加密法差不多，但是加密移動向量差值的 Bitstream 不會破壞壓縮率且保持語意格式兼容。

C. 殘餘係數加密法

在視訊區塊經過 DCT、量化之後，所殘餘的係數值會分為 DC 值與 AC 值。在一般常見的殘餘係數加密方案中，加密的位置是殘餘係數的值(Level)或者是正負號(Sign)，以下便介紹一些此類的加密方案。

(1) Wang 等人[27]提出一個 Luminance transform coefficients encryption (LTCE)，作者認為加密亮度區塊的效率比加密彩度區塊高，因為人類視覺對亮度的改變比較敏感，因此透過加密 intra 16×16 亮度區塊的 DC (標記為 T1)、AC (標記為 T2)值與 inter 及 intra 4×4 亮度區塊的 AC 值(標記為 T3)來達到視訊加密的目的。做法如下：

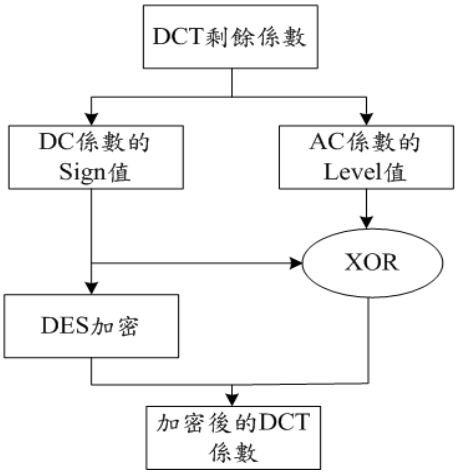
1) T1、T2 及 T3 透過 $C = E_{k1}(\mathrm{T1}(\alpha)) + E_{k2}(\mathrm{T2}(\beta)) + E_{k3}(\mathrm{T3}(\gamma))$這式子來加密。用三把不同的 Key 來加密，加密的運算為 XOR。此外，用三個變數 α、β、γ 來控制加密強度(限制 α 介於 0~16，β 介於 0~15，γ 介於 0~15)。
2) 為了保持壓縮率，0 的地方不加密。
3) 若加密後的值為 0，會與原本值就等於 0 的地方混淆，導致解密發生問題，因此需做而外的處理。加密後若值為 0，則 $C_i = C_i \oplus k_i$，k_i 為 key。

此方法是加密殘餘係數的 Level，一般而言，在視訊中 intra 4×4 的使用率會比 intra 16×16 來的高，但是 intra 4×4 的 DC 值卻不加密，可能導致加密效果不佳，但相對的對壓縮率的衝擊也較小，另外，此方法是可以保持語意格式兼容的需求。

(2) Guo 等人[5]提出的視訊加密中，加密的位置是 DC 的 Sign，AC 的 Level，以及 MV 的 Sign 及 Level，加密運算則是利用 DES encryption 來做加密。以下僅就 DC 與 AC 的部分介紹，其做法如下：

1) DC 的 Sign 透過 DES 來加密。
2) 將 4 個 bits 的未加密過的 DC Sign 做成一個 16 進位的值，再與 AC 的絕對值做 XOR。
3) 最後即可得到加密過後的 DC 及 AC。其流程圖如圖七

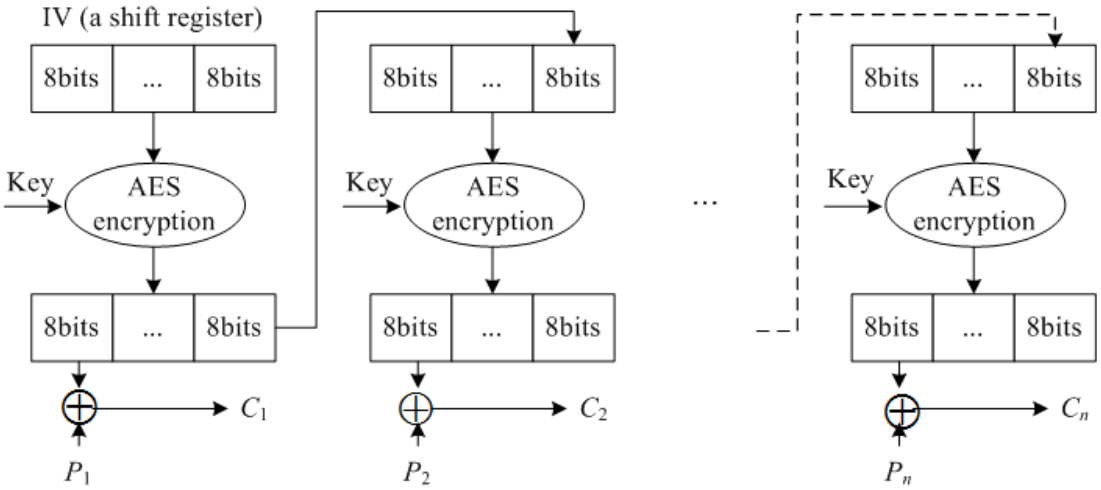
依照其實驗結果，可以發現有色塊的改變，所以猜測其應該連彩度區塊也有加密。若是如此，彩度區塊的加密效率不佳。一般而言，DC 擁有的視訊資料會比 AC 來的多，因此加密 DC 所產生的加密效果會比 AC 來的高，然而此方法 DC 僅加密 Sign，而 AC 卻加密了 value。再者，因為 AC 比 DC 的資料量來的大的多，加密 AC Level 所造成的壓縮率影響會比單純加密 DC value 更大，此外，此方法是可以保持語意格式兼容的需求。



圖七：Guo 等人視訊加密法流程圖

(3) Wang 等人[25]提出的加密演算法，其核心概念為用 AES 來加密亮度以及彩度區塊的 DC 值，並且加密值非 0 且非 1 的 AC 值，最後值為 1 的 AC 值則僅作 Sign encryption。做法如下：

1) 用 AES 的 OFB(Output Feedback Block)機制來對殘餘係數做加密，如圖八所示。*P* 為明文，IV 為 initial vector， key 金鑰，*C* 為密文。



圖八：Output Feedback Block(OFB)加密機制

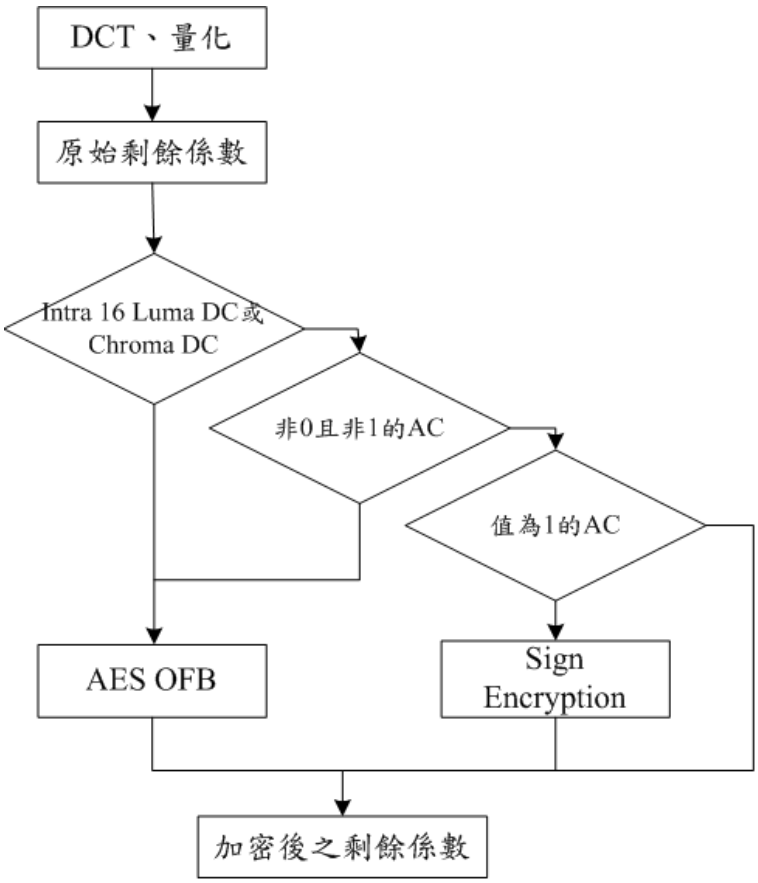
2) 用 OFB 對 Intra 16×16 的亮度及彩度區塊之 DC 值做加密。
3) 用 OFB 對非 0 且非 1 的 AC 值做加密。
4) 對值為 1 的 AC 做 Sign encryption。其流程圖如圖九：

在此方法中利用 OFB 對 DC 的 Level 進行加密，因此會造成壓縮率上會造成一定程度衝擊，而加密 AC 的 Sign 對壓縮率上的衝擊是微乎其微的，此外，此方法是可以保持語意格式兼容的需求。

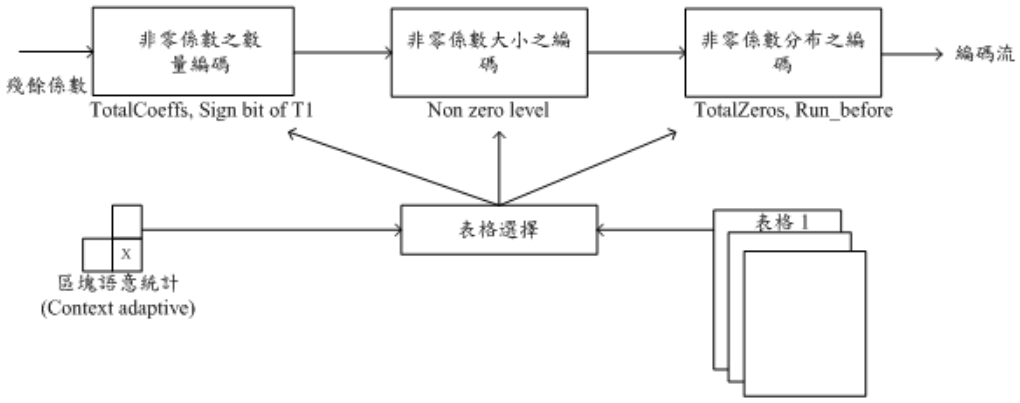
D. 熵編碼加密法

熵編碼加密較為常見的是 CAVLC 加密，而圖十為熵編碼 CAVLC 的編碼過程。

由圖十中可以知道殘餘係數經過 CAVLC 後，其輸出的編碼流是由五種語意元素所組成的，分別為 TotalCoeffs、Sign bit of T1、Non-zero Level、TotalZeros 以及 Run_before，因此在 CAVLC 的加密方法中就是對這五種編碼元素進行加密，其中 TotalCoeffs 是針對目前區塊與相鄰區塊的非零係數的個數進行編碼，Sign bit of T1 是對目前區塊的非零係數中最後有幾個 1，最多為 3 個，Non-zero Level 是針對非零係數的數值進行編碼，TotalZeros 以及 Run_before 是針對非零係數與零之間的關係。



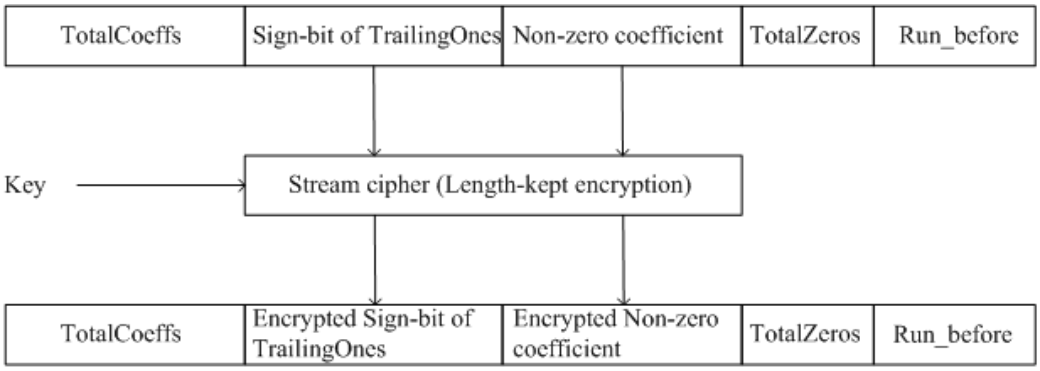
圖九：Wang 等人加密方案之流程圖



圖十：熵編碼 CAVLC 的編碼過程

(1) Lian 等人[7]提出的加密演算法中就加密了 Sign bit of T1 和 Non-zero Level 的 Codeword，並且其加密的過程中不會使得檔案格式被破壞已以及壓縮率變差，圖十一為加密的過程。做法如下：

1) Sign bit of T1 加密:利用 Stream cipher 對 Sign bit of T1 進行 Length-kept encryption
2) Non-zero Level 加密: Non-zero Level 的字碼結構分為 *LevelPrefix* 和 *LevelSuffix*，其中 *LevelPrefix* 是利用查表來得知其字碼，而加密 *LevelPrefix* 會破壞檔案格式，因為因為 *LevelPrefix* 等於代表要取後面多少 bits 當作 *LevelSuffix*，所以置亂後可能會發生多取或是少取 *LevelSuffix* 的 bit 數，而導致格式被破壞，因此只有 *LevelSuffix* 可以加密，所以此加密演算法就利用 Stream cipher 對 *LevelSuffix* 進行 Length-kept encryption。



圖十一：Sign bit of T1 和 Non-zero Level 的加密過程

由於這個方法是對 Sign bit of T1 和 Non-zero Level 的 Codeword 利用 Stream cipher 進行加密，因此其壓縮率不會被破壞而導致檔案擴大，此外因為這兩個編碼元素都只是

單純編其數值而不會影響到其他編碼元素的編碼，因此其檔案格式不會被破壞。

(1) Cia 等人[10]提出了置亂 TotalCoeffs、TotalZeros 以及 Run_before 這三個編碼元素的 Codeword 的加密演算法。做法如下：

1) TotalCoeff 加密 :加密表格中的 Index，以置亂到不同的 codeword，並考慮是否有超出範圍。
2) TotalZeros 加密 :加密表格中的 Index，以置亂到不同的 codeword，並考慮是否有超出範圍。
3) Runbefore 加密 :加密表格中的 Index，以置亂到不同的 codeword，並考慮是否有超出範圍。

其方法雖然具有高安全性，但是因為其 Codeword 被置亂，可能會導致其編碼區塊的係數會多於 16 或是小於 16 而導致檔案格式被破壞，會稍微影響到壓縮率。

肆、討論與分析

A. 安全性

(1) 在畫面內預測模式加密

在做 Intra Prediction 的步驟時，其單位大小分成亮度區塊 4×4 和 16×16，以及彩度區塊 8×8。對於人類的視覺來說，我們對於亮度的變化比較敏感，因此，針對亮度區塊來加密效果會比較好。而亮度區塊中又分成 4×4 和 16×16 兩種，通常視訊畫面中，4×4 的比例會比 16×16 高的多。所以針對 4×4 來加密，其效率會比加密 16×16 來的好。然而單純加密 IPM 在安全性上是堪虞的，只要將畫面中所有的模式都用同一種來取代，就會被看出一些端倪，也就是 Replacement Attack。因此，在加密的時候，必須伴隨其他的重要語意來做加密，這樣可以提高其安全性。最後，單論針對 4×4 和 16×16 的兩種常見加密方法。因為 4×4 有 9 種模式，大部分是利用 3 bits 來將其打亂，但是 3 bits 只能打亂其中 8 種模式，還是有機會沒有被打亂到。而 16×16 為了維持其格式，所以只在經過 Exp-Golomb 編碼的後半部加密。雖然加密 4×4 就能達到不錯的效果，為了提高其安全性，有些方案會將 16×16 的也一起加密。

(2) 畫面間預測模式與移動向量差值加密法

加密畫面間預測模式的視覺安全性是相當低的，因為只也採取 4×8 或是 8×4 的區塊被加密，而且被影響的移動向量的數目很少的，而加密移動向量差值的三種方法的安全性差不多，事實上具有比較好的安全性，但是如果被加密的視訊是屬於比較不會變動的畫面，例如天空的場景，就幾乎失去了加密的效果。

(3) 在殘餘係數加密

單論 DC 與 AC 的部份，加密 DC 值會比加密 AC 值的效率來的好，因為 DC 值含

有較高的視訊資訊。事實上，越接近低頻的係數只要稍加改變就能產生很好的加密效果。此外，若亮度區塊與彩度區塊的殘餘係數相比的話，則加密亮度區塊的殘餘係數效果較好，因為人類的視覺系統對亮度的變化較敏感，而對彩度的變化較不敏感。因此加密亮度區塊之殘餘係數對人類的視覺系統而言會有較佳的加密效果，也因此能有較高的視覺安全性。最後，從加密機制來探討，Sign encryption 的安全性會比 Level encryption 低，因為 Sign encryption 僅僅改變 Sign 值，與改變整個 Level 值的加密法相比，不僅視覺加密效果較差，在安全性上也較低。但是 Sign encryption 的加密速度較快，因此若真要使用 Sign encryption 這種加密機制，最好是要再加上其他資料或語意的加密。

(4) 熵編碼加密法

加密語意單元 Sign bit of T1，視覺上的安全性是相當差的，因為 Sign bit of T1 是編在區塊中最後的最多連續三個正負 1 的符號，因此加密 Sign bit of T1 只是改變 1 的正負號，因此在視覺的改變是微乎其微的，而在於加密機制來看，因為加密 Sign bit of T1 大多是使用 Length kept encryption，因此其加密機制的安全性是基於 Steam cipher 的安全性；加密語意單元 Non-zero Level，其視覺上的安全性是高於加密 Sign bit of T1，因為加密 Non-zero Level 實際上是在加密殘餘係數中的 DC 與 AC 的數值，但是相較於加密 DC 與 AC，Non-zero Level 加密是由其值域的拘限，因此加密後的視訊是還可以看見視訊內容的輪廓，但是其優點是不會破壞壓縮率，而加密機制的安全性也是基於 Steam cipher 的安全性；加密 TotalCoeffs、TotalZeros 以及 Run_before，其視覺安全性是相當的高，因為加密 TotalCoeffs、TotalZeros 以及 Run_before 是不具備保持語意格式兼容。

B. 運算效率

(1) 畫面內預測模式加密

不管是用加密 4×4 亮度區塊所採用的 3 bits 的加密方式或是加密 16×16 亮度區塊所採用的 Exp-Golomb code 後半段加密，加密機制都僅是打亂這些 bits 值，因此所花費的計算量不大。

(2) 畫面間預測模式與移動向量差值加密法

在加密畫面間預測模式時，其運算速度相當快，因為只是依金鑰來交換兩種預測區塊，而在加密移動向量差值時，主要都是利用 XOR 運算，因此其運算速度也是相當快，但是加密 Sign 的速度會比加密 Level 的速度來的快，因為只需要做正負號的改變。

(3) 殘餘係數加密

若以相同的加密機制來說，加密 DC 的運算效率會比 AC 好，因為 DC 值的資料量遠低於 AC 值。再以加密機制來探討，目前常見的加密機制就是 Sign encryption 及 Level encryption，加密 Sign 的速度會比加密 Level 的速度來的快，因為只需要做正負號的改變。

(4) 熵編碼加密法

由於目前加密 Sign bit of T1 和 Non-zero Level 的方式都是用 Stream cipher 來進行加密，如果不計算產生金鑰的時間，加密 Sign bit of T1 所花費的時間相當低，因為每一個區塊最多是做 3bit 的 XOR 運算，而加密 Non-zero Level 所花費的時間是比較多的，因為所需要執行 XOR 運算的 Bit 數是比較多，但是因為 XOR 運算所花費的時間很少，因此 Non-zero Level 加密真正所花費的時間也是相當低。依上面所敘述的加密演算法得知加密 TotalCoeffs、TotalZeros 以及 Run_before 所花費的時間是相當的少，因為只需要做簡單的 Table 置亂就能達到加密效果。

C. 壓縮效率

(1) 畫面內預測模式加密

不管是對 4×4 或是 16×16 作加密，其 bit stream 皆不會改變，所以在壓縮方面都不會影響其壓縮率。

(2) 畫面間預測模式與移動向量差值加密法

在畫面間預測加密法中，是將在編碼之前產生移動向量之後的兩個具有相同移動向量數目的畫面間預測區塊依金鑰來進行置亂，因此不會影響其壓縮率，而加密移動向量差值的 Sign 及加密移動向量差值的 Bittream 是不會影響壓縮率的，但是加密移動向量差值的 Level 則是會強烈的影響壓縮率。

(3) 殘餘係數加密

一般而言，加密 Sign 值的加密機制是不會影響壓縮率的。但是加密 Level 值若希望保持壓縮率，則需要相當注意演算法的設計、加密後的值域等等，因為在殘餘係數裡，越高頻的係數往往都是 0，或者是非常接近 0，而 H.264/AVC 的高壓縮率就是建立在這些 0 之上。若是因為加密而使得這些原本很小的值變大，那勢必影響壓縮效率。因此，在殘餘係數加密裡，我們會建議 DC 值的部份可以利用 Level 或是 Sign encryption 來加密，但是 AC 值比較適合直接用 Sign encryption，因為 AC 值資料量遠大於 DC 值，加密這部份很容易會影響壓縮率，所以用速度較快的 Sign encryption 會比較合適。

(4) 熵編碼加密法

由於加密 Sign bit of T1 和 Non-zero Level 都是使用 Stream cipher 加密的方式，因此不會破壞壓縮率，而加密 TotalCoeffs、TotalZeros 以及 Run_before，因為是利用置亂 Table 的方式，因此會影響到壓縮率，但是影響不大。

D. 保持語意格式兼容

(1) 畫面內預測模式加密

在加密亮度 4×4 區塊的 IPM 時，常見的方式是去加密亮度 4×4 的九種模式，我們

會直接去打亂代表其九種模式的 3 個 bits，這動作不會影響語意格式。然而，在加密亮度 16×16 的 IPM 時，就必須考慮可能會影響語意格式，因為亮度 16×16 的 IPM 是在 Entropy coding 時由 Exp-Golomb code 來編碼，其編碼的前半部是有一定格式的，因此為了保持語意格式，僅能加密後半部的部份。

(2) 畫面間預測模式與移動向量差值加密法

在加密畫面間預測模式與移動向量差值比較不需要擔心會影響語意格式，因為加密機制只是單純改變殘餘係數的 Sign 或是 Level 的大小或是在不改變長度的情況下改變 Exp-Golomb 的值，因此是不會去變動其語意格式。

(3) 殘餘係數加密

在殘餘係數加密的部份比較不需要擔心會影響語意格式，因為加密機制只是單純改變殘餘係數的 Sign 或是 Level 的大小，不會去變動其語意格式。

(4) 熵編碼加密法

在 CAVLC 中有 5 個語意單元，而這裡把這 5 種語意單元分為兩種類型來進行討論，第一種類型為單純的對某個語意或資料的符號或大小來進行編碼，這種類型的編碼不會去編此語意與其他語意的關連性，因此加密此類型的語意，在解壓縮時不會影響到其他語意的解碼，此種類型的語意單元有 Sign bit of T1 和 Non-zero Level；另一種為會對語意與語意之間或是區塊之間的關聯性進行編碼，因此加密此類型的語意，在解碼時會影響到其他的語意的解碼，此種類型的語意單元有 TotalCoeffs、TotalZeros 以及 Run_before。因此利用舊有的方式，例如 MPEG-4 中的 VLC scrambling 以及上述所敘述的方法是不能保持語意格式兼容的，因為 H.264/AVC 中 CAVLC 的 Table 通常表示上下文之間關連性或是特性，舉例來說加密 TotalCoeffs=5 為 TotalCoeffs=7，但是實際上 Non-zero Level 中只編 5 個係數，因此解壓縮的時候 Codec 會找 7 個係數來解壓縮，所以會出現解碼失敗而破壞格式，因此置亂 Table 的方式會使得檔案格式被破壞。所以要保持檔案格式只能加密不會影響上下文之間關連性或是特性的 Bitstream，例如 Sign bit of T1 或 Non-zero Level 的 Bitstream 只是單純的表達單一數值的特性而不會影響其他數值。因此在 CAVLC 的高複雜度使得在 CAVLC 的加密會有高度的侷限性的情況下，為了保持語意格式兼容，所以適合加密的地方只有 Sign bit of T1 及 Non-zero Level。

表一為適合加密的語意或是資料的視覺安全性與效能的比較表，而運算效率是將產生金鑰不列入考量。

(1)從表中可知道在去除空間冗餘中加密亮度 4×4 預測模式是最好的選擇，但是其視覺安全性還是不夠。

(2)在殘餘係數 DC、AC 值中加密亮度的 Sign 是最好的選擇，但是最好還是將亮度與彩度一起加密來增加安全性。

(3)對於加密移動向量差值，加密 Sign 及加密指數哥倫布編碼之後的 Bitstream 比較好，因為在相同的安全性下加密移動向量差值的 Sign 及加密指數哥倫布編碼的 Bitstream 並不會影響到壓縮率。

(4)在去除編碼冗餘中，對於CAVLC來說加密Non-zero Level是比較好的選擇，因為Signs of T1加密效果並不明顯，而其餘三種語意元素會無法保持語意格式。

表一：適合加密的語意或是資料的視覺安全性與效能的比較表

選擇的資料 \ 特性				安全性	壓縮比	運算效率	保持語意格式
去除空間冗餘	預測模式	亮度	4×4	中	不變	Higher	保持
			16×16	低	不變	Higher	保持
		彩度		低	不變	Higher	保持
殘餘係數 DC、AC 值	Sign	亮度		高	不變	Higher	保持
		彩度		低	不變	Higher	保持
	Level	亮度		高	大量增加	High	保持
		彩度		低	大量增加	High	保持
去除時間冗餘	移動向量差值	Sign		高	不變	Higher	保持
		Level		高	大量增加	High	保持
	預測模式			低	少量增加	Higher	保持
去除編碼冗餘	內文適應性可變動長度編碼(CAVLC)	Coeff_token		高	少量增加	Higher	不保持
		Signs of T1		低	不變	Higher	保持
		Non-zero Level		高	不變	High	保持
		Total zeros		高	少量增加	Higher	不保持
		Run_before		高	少量增加	Higher	不保持
	指數哥倫布編碼(Exp-Golomb Coding)	移動向量差		高	不變	High	保持
		預測模式	Intra	低	不變	Higher	保持

但單獨加密以上的語意或是資料不論視覺安全性或是加密機制都是不足的，事實上不論選擇哪一個來加密，雖然視訊內容可能會被混淆，但都還是可以看到視訊內容的輪廓。

(1)亮度4×4預測模式加密，雖說P/B畫面會參考到I畫面而被加密，但是加密亮度4×4預測模式本身其視覺安全性就不足的，而且因為移動向量差值並沒有被加密，因此還是可以被有心人士來得到移動向量差值。

(2)移動向量差值雖然可以嚴重破壞視訊內容，但只是破壞P/B畫面的資訊，而I畫面並沒有被加密。

(3)加密殘餘係數 DC、AC 值的Sign或是加密CAVLC中的Non-zero Level，雖說具有一定的視覺安全性，但還是可以被得知輪廓，如果要有足以保護視訊內容的話就必須加

密殘餘係數 DC、AC 值的 Level 或是加密 CAVLC 中的語意單元 TotalCoeffs、TotalZeros 以及 Run_before，但是前者會嚴重的影響壓縮率，而後者會破壞語意格式。

因此最好的方法是將加密後具有高視覺安全性，但是不影響壓縮率且保持語意格式兼容的語意或資料一起加密以增加其安全性。經本文的分析後，我們可以依循以上的歸結，可釐清最新的相關研究文獻中所採用的技術。例如: Meenpal [16]只加密 I 畫面的殘餘係數的量化後 DCT 值，Peng 等[19]則是加密 T1 及 MVD 兩者的 sign，以及 IPM，其選擇性加密的特色可由表一得知。

伍、結論

本研究探討了現有的選擇性視訊加密方案，研究在這些基於 H.264 而設計的視訊加密方案中，學者們會選擇加密的地方。並分別以安全性、運算效率、壓縮效率、保持語意格式兼容四項視訊加密方案的基本需求來分析比較這些加密方案的優缺點，以期能夠對剛接觸視訊加密這領域的讀者提供有效的導讀。另外，經過分析之後，我們可以知道在選擇性加密法中，是不能夠單獨的選擇一個語意單元或是資料來進行加密，因為只選擇一種來加密其視覺上的安全性是不夠的，如果是為了不可視覺的應用，其加密方案應該要同時加密去除空間冗餘的語意單元、去除時間冗餘的語意單元以及殘餘係數或去除編碼冗餘的語意單元，才能保證有最好的視覺安全性。

參考文獻

[1] J. Ahn, H. J. Shim, B. Jeon and I. Choi, “Digital video scrambling method using intra prediction mode,” *Lecture Notes in Computer Science*, Vol. 3333, pp. 386-393, 2005.

[2] G. B. Algin and E.T. Tunali, “Scalable video encryption of H.264 SVC Codec,” *Journal of Visual Communication and Image Representation*, Vol. 22, Issue 4, pp. 353-364, 2011.

[3] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC),” *in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050*, 2003.

[4] Y. Fan, J. Wang, T. Ikenaga, Y. Tsunoo and S. Goto, “An unequal secure encryption scheme for H.264/AVC video compression standard,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E91-A, No.1, pp. 12-21, 2008.

[5] S. S. Guo, Y. X. Meng and W. Zhang, “A new algorithm for using dynamic keys encryption video information,” *2009 WCSE '09. Second International Workshop on*

Computer Science and Engineering, Vol. 2, pp. 542-545, 2009.

[6] Y. Li, L.W. Liang, Z.P. Su and J.G. Jiang, "A new video encryption algorithm for H.264," *2005 Fifth International Conference on Information, Communications and Signal Processing*, pp. 1121-1124, 2005.

[7] C. Li, X. Zhou and Y. Zhong, "NAL level encryption for scalable video coding," *Lecture Notes in Computer Science*, Vol. 5353, pp. 496-505, 2008.

[8] S. Lian, Z. Liu, Z. Ren and Z. Wang, "Selective video encryption based on advanced video coding," *Advances in Multimedia Information Processing-PCM*, Vol. 3768, pp. 281-290, 2005.

[9] S. Lian, J. Sun and G. Liu, "Efficient video encryption scheme based on advanced video coding," *Multimedia Tools and Applications*, Vol. 38, pp. 75-89, 2008.

[10] S. Lian, J. Sun, G. Liu and Z. Wang, "Efficient video encryption scheme based on advanced video coding," *Multimedia Tools and Applications*, Vol. 38, pp. 75-89, 2008.

[11] S. Lian, J. Sun, D. Zhang and Z. Wang, "A selective image encryption scheme based on JPEG2000 codec," In *Pacific-Rim Conference on Multimedia*, Springer, Berlin, Heidelberg, Vol. 3332, pp. 65-72, 2004.

[12] F. Liu and H. Koenig, "A survey of video encryption algorithms," *Computers & Security*, Vol. 29, pp. 3-15, 2010.

[13] Y. Liu, C. Yuan and Y.Z. Zhong, "A new digital rights management system in mobile applications using H.264 encryption," *The 9th International Conference on Advanced Communication Technology*, pp. 583-586, 2007.

[14] E. Magli, M. Grangetto and G. Olmo, "Transparent encryption techniques for H.264/AVC and H.264/SVC compressed video," *Signal Processing*, Vol. 91, Issue 5, pp. 1103-1114, 2011.

[15] S. S. Maniccam and N. G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognition*, Vol. 37, No. 4, pp. 725–737, 2004.

[16] T. Meenpal, "A light weight and secure video conferencing scheme utilizing public network," *Multimedia Tools and Applications*, Vol. 76, No. 3, pp. 3699-3714, 2017.

[17] National Institute of Standards and Technology (U.S.), "Advanced Encryption Standards (AES)," *FIPS Publication 197*, 2001.

[18] S. W. Park and S.U. Shin, "Efficient selective encryption scheme for the H.264/Scalable Video Coding (SVC)," *2008, NCM '08. Fourth International Conference on Networked Computing and Advanced Information Management*, pp. 371-376, 2008.

[19] F. Peng, X. Q. Gong, M. Long and X. M. Sun, "A selective encryption scheme for

protecting H. 264/AVC video in multimedia social network," *Multimedia Tools and Applications*, Vol. 76, No. 3, pp. 3235-3253, 2017.

[20] M. Podesser, H. P. Schmidt and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," *5th Nordic Signal Processing Symposium, on board Hurtigruten, Norway*, 2002.

[21] L. Qiao and K. Nahrsedt, "Comparison of MPEG Video Encryption Algorithm," Computer and Graphics, Vol. 22, No. 4, pp. 437-448, 1998.

[22] R. L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.

[23] S. Sakazawa, Y. Takishima and Y. Nakajima, "H.264 native video watermarking method," *2006 IEEE International Symposium on Circuits and Systems, 2006, ISCAS 2006. Proceedings*, pp. 1439-1442, 2006.

[24] J. Teuhola, "A compression method for clustered bit-vectors," *Information Processing Letters*, Vol. 7, No. 6, pp. 308–311, 1978.

[25] Y. Wang, M. Cai and F. Tang, "Design of a new selective video encryption scheme based on H.264," *2007 International Conference on Computational Intelligence and Security*, pp. 883-887, 2007.

[26] J. Wang, Y. Fan, T. Ikenaga and S. Goto, "A partial scramble scheme for H.264 video," *2007 7th International Conference on ASIC*, pp. 802-805, 2007.

[27] L. F. Wang, J. W. Niu, J. Ma, W. D. Wang and X. Chen, "A lightweight video encryption algorithm for wireless application," *2008 Fifth IEEE International Symposium on Embedded Computing*, pp. 94-97, 2008.

[28] X. Wang, N. Zheng and L. Tian, "Hash key-based video encryption scheme for H.264/AVC," *Image Communication*, Vol. 25, Issue 6, pp. 427-437, 2010.

[29] D. Y. Wang, Y.J. Zhou, D.D. Zhao and J.F. Mao, "A partial video encryption scheme for mobile handheld devices with low power consideration," *2009 International Conference on Multimedia Information Networking and Security*, pp. 99-104, 2009.