國立交通大學

電信工程學系碩士班

碩 士 論 文

結合硬體與軟體的光碟防盜拷技術

A Hybrid Hardware/Software Copy Protection Technique for Optical Disks.

研 究 生：錢大源

指導教授：高銘盛　教授

中 華 民 國 九 十六 年 八 月

結合硬體與軟體的光碟防盜拷技術

A Hybrid Hardware/Software Copy Protection Technique for Optical Disks.

研究生　：錢大源　　Student：　Ta Yuen Chien

指導教授：高銘盛 博士　　Advisor：Prof. Ming-Seng Kao

國 立 交 通 大 學

電信工程學系碩士班

碩 士 論 文

A Thesis Submitted to
The Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
In partial Fulfillment of the Requirements
For the Degree of Master of Science
In
Communication Engineering
June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十六年八月

結合硬體與軟體的光碟防盜拷技術

A Hybrid Hardware/Software Copy Protection Technique for Optical Disks.

研究生　：錢大源　　Student： Ta Yuen Chien

指導教授：高銘盛 博士　　Advisor：Prof. Ming-Seng Kao

國立交通大學電信工程學系碩士班

摘要

我們提出同時使用硬體與軟體的光碟防盜拷技術。在我們的方法中，半準位訊號 x-bit 提供人為錯誤置於光碟片中，再利用多階的 MWE 程序來編碼原始鑰匙。由於多階的編碼程序導致一個非常特殊的解碼效能，會產生一個近似方波的原始鑰匙解碼正確機率，而這個特性可以用來防制盜拷。更進一步，我們提出新的編碼方法來增加盜拷光碟的困難度，並有效降低編碼所需的容量。

Abstract

We propose a hybrid hardware/software copy protection technique based on the X-bit coding and a multi-level word-extension /error-coding (MWE) scheme to encode an encryption key. The X-bits with half-level signaling behave as potential errors in optical disk. The multi-level coding leads to a very special decoding performance that a sharp cutoff exists in the detection probability of the encryption key. This cutoff behavior sets a tight bound on the signal level of X-bits, being the desired characteristic for us to prevent piracy. Finally，we propose a modified approach to decrease the tight bound on the signal level of X-bits as well as volume of the encryption key.

誌謝

本論文得以順利完成，首先要感謝我的指導教授高銘盛博士，在兩年的研究生活中，無論在電機專業領域或生活上的待人處事，都使我有很大的收穫。

感謝 909 實驗室的所有成員，謝謝你們平時生活的照顧與協助，研究生的生活因你們而多了許多歡笑與回憶，風城的求學日子因你們而充滿愉快的回憶。

僅以此論文獻給所有關心我、愛我的人。

目錄

表目錄

圖目錄

第 一 章 緒 論

有鑑於現今盜版光碟的普遍，各國如美國、日本、台灣 …… 等，皆致力於防止盜拷。但日益猖獗的盜拷行為，已經重創各國的市場。以往台灣唱片業尚能傳出銷售百萬張的佳績，如今能賣出一 、二十萬張就已經不錯了。這是盜拷行為氾濫所造成的結果，盜版商為了貪圖利益，罔顧他人的智慧財產權 ，而購買盜版光碟的民眾，為節省小錢，採買盜拷的產品，苦了他人的心血。版權擁有者與盜版商正展開一場角力戰，設計一些防止盜拷的方法，希望能有效的抑制盜版行為。子曰 ：『 勿以善小而不為，勿以惡小而為之。』，現今社會上對於智慧財產權的漠視，導致抄襲行為的漫延，創意是經過無數個難熬的夜晚所產生的結晶，但可能你的一個按鈕，就被剝奪。由衷希望大家能重視創作，之如愛惜自己。

長期以來，盜拷行為已經嚴重打擊光碟市場，防盜拷技術更是一個重要的議題[1-5]。雖然有很多機構紛紛投入防盜拷的研究，但至今所提出的方法只能防止一般民眾，卻無法制止專業的盜版商，結果造成盜版光碟大量流通，使版權擁有者損失慘重[6-7]。以下簡單介紹幾種常見的光碟防盜拷技術 ：

數位浮水印(digital watermarking ）:我們通常先將數位浮水印植入數位內容中，再交予使用者流通，一旦發現雷同的數位內容，則取出數位浮水印，比較兩個浮水印相似的程度作為是否侵權的依據[2]。我們知道身份證的主要功能在於辨識身份，而數位浮水印如同身份證，具有識別的功能。數位浮水印有兩種類型：一種是可見型，另一種是隱藏型。一般而言，可見型的浮水印顯露在數位內容中，而隱藏型的浮水印並不能直接觀察，所以隱藏型的浮水印安全性較高。

假如今天版權的擁有者持有一份數位內容I，為了保護它，將數位浮水印W加至I中，則生成標記後的數位內容I'：

$$I' = I + W \tag{1.1}$$

版權的擁有者再將I'流通，一但發現雷同的數位內容I''時，則取出I''的浮水印W'：

$$W' = I'' - I \tag{1.2}$$

如果I''的確是抄襲數位內容I'，則取出來的浮水印W'必定與W相似，我們可以藉由比較W'與W來判斷是否抄襲。

數位簽章(digital signature)： 所謂的公開鑰匙加密法就是將我們想保護的資料分別以兩把不同的鑰匙作加密、解密。例如甲、乙兩個人想要互相傳送資料，但是又怕資料外洩，就可以使用公開鑰匙加密法。方法是想要聯繫的兩個人各自擁有兩把鑰匙，一把稱為公鑰 ，另一把稱為私鑰。一般而言，公鑰是可以讓大家知道的，但是私鑰必須保密隱藏。今天甲想要傳送資料給乙，甲必須先使用乙的公鑰加密，再將加密後的資料傳送給乙，而乙收到後，則用自己的私鑰解密 。我們知道公開鑰匙加密法的另一個用途是數位簽章 [4]，數位簽章的概念與公開鑰匙加密法剛好相反，主要是確認原始訊息的傳送者是否是我們指定的對象。我們先將原始訊息代入 hash function 產生 hash value，再將 hash value 與私鑰作二位元加法產生數位簽章。我們知道 hash function 的特性是反函數很難找到，所以我們並不能從函數值回推原始訊息。接著，檢驗者只要將數位簽章與公鑰相加，如果加總後的結果等於 hash value，則該人士就是我們指定的對象。

Serial Copy Management System (SCMS) : 一般而言，SCMS 技術 [6]主要是在光碟中加入兩種位元，一種是 Copyright bit，而另一種是 Copy Status bit。我們知道光碟在播放時，光碟機會先去偵測 Copyright bit，如果能順利偵測出 Copyright bit，就代表正在播放的這張光碟是經過授權的。而 Copy Status bit 能顯示出光碟被拷貝的次數，如果拷貝的次數達到版權擁有者規定的最大次數時，我們就不能再拷貝光碟。

以往所提出的軟體防盜拷技術主要是隱藏光碟內容，但並不能有效阻止位元對位元的拷貝(bit-by-bit copy)。因此，光碟機無法分辨正版光碟和盜版光碟，故盜版光碟得以順利播放。一般而言，拷貝光碟需要三個要件：光碟、光碟內容和燒錄器 ； 如果缺少任一個就無法拷貝光碟。專業的盜版商要取得正版光碟並不困難，只要去商家購買就可以。接著，將一般的燒錄器作位元對位元的對拷可以得到光碟內容，甚至經過加密後的光碟也可以，因此最後的防線只剩下燒錄器。早期類比唱片的年代，拷貝是很困難的，因為盜拷的唱片與正版的唱片無法完全相同，故能自動地防止盜拷。這個概念便是我們所提出光碟防盜拷技術的主要想法，目的是使盜拷光碟很難與正版光碟完全相同，以有效防止盜拷。

我們所提出的方法主要是提高燒錄的困難度，所以拷貝光碟必須以極高精密度的機器完成，而製造這種機器需要極高的成本，專業的盜版商根本無法負荷，因此得以有效的阻止盜拷光碟。

我們提出結合硬體與軟體的防盜拷技術是利用半準位訊號 x-bit 和多階層錯誤更正碼[9]來防止盜拷。一般而言，光碟以兩種訊號準位代表二位元 0 與 1，現在我們引進第三種訊號 x-bit 加入光碟中，目的是提升拷貝的困難度。此方法有別於過去的防盜拷技術，主要以硬體的角度出發，而以往的防盜拷技術主要以軟體的角度做考量。我們採取兩次讀取程序偵測 x-bit，由於正確偵測 x-bit 的機率並不高，故可以將 x-bit 當作人為的錯誤置於光碟中。此錯誤機制再加上多階層錯誤更正碼便構成所提防盜拷技術的主體。

在所提出的方法中，我們利用一鑰匙將原始資料打亂(scramble)，接著使用 multi-level word-extension error coding(簡稱 MWE)與 x-bit encoder 將原始鑰匙編為加密鑰匙放入光碟中。MWE 程序會導致一個特殊的解碼效果，產生一個近似方波的原始鑰匙解碼正確機率。若光碟中的 x-bit 準確度不高時，解碼後的原始鑰匙會發生錯誤，故無法讀出正確的內容。由於製作高準確度的 x-bit 需要極精密的機器，一般的盜版商根本購買不起，故能有效的防止盜拷。

接著，我們希望提升拷貝光碟的困難度，所以提出變型系統 I。這個系統的確能提升光碟盜拷的難度，但是它所面臨的問題是加密鑰匙位元數過於龐大，實際應用上有問題。因此我們提出變型系統 II，將加密鑰匙位元數壓縮至一個合理的容量，以有效解決加密鑰匙過於龐大的問題。

本論文共分為五章，第一章為簡介。在第二章中，我們討論半準位訊號 x-bit，它可以當作人為的錯誤置於光碟中。接著，我們分析 x-bit 偵測正確的機率。在第三章中，我們描述 MWE 程序與解碼程序。在第四章，我們希望提高拷貝光碟的困難度，所以提出變形系統 I，但伴隨的問題是加密鑰匙位元數太大，所以提出改良的變形系統 II 以有效解決這個問題。最後在第五章中，我們作一個簡單的結論。

第二章 X-bit 之介紹與討論

2.1 光碟片的加密程序

過去所提出的加密方法，通常是設計一把鑰匙(key)至對應的光碟片，再將鑰匙與所要儲存的資料(source)做相加(二位元加法)，相加後的結果編碼到光碟中 。接著對這把鑰匙進行一連串的加密動作，再把加密後的鑰匙寫入光碟。光碟播放時，播放機(player)再把原始的鑰匙從加密鑰匙中解調回復，接著與光碟中的內容相加，以解出所儲存的資料。加密及解密的工作原理簡述如下：

ki ： 光碟所對應的鑰匙

K ： 加密後的鑰匙

ke ： 播放端解調出錯誤的鑰匙

di ： 我們想儲存的資料

Ci ： 光碟所存放的內容

在製作光碟時，先將光碟資料(di)加密；播放時再將它解密：

$$ki + di = Ci \tag{2.1.1}$$

$$Ci + ki = ki + di + ki = di \tag{2.1.2}$$

若播放機無法回復正確的鑰匙，而解調出錯誤的鑰匙ke，則

$$Ci + ke \neq di \tag{2.1.3}$$

由上式可知，若無法獲得正確的鑰匙，就無法得到正確的資料，故可以用來防止盜拷光碟。

我們所提出的方法是先將原始鑰匙(key)以 multi-level word-extension error-coding (簡稱 MWE)的方式編碼，再安插 x-bit 至編碼後的結果，以達成防盜拷功能(MWE 的編碼方式與 x-bit 的加入，這些細節我們稍後會提) 。

2.2 x-bit 之分析

一般光碟中的訊號有兩種，一為二位元訊號 0，另一為二位元訊號 1。我們稱之為正規位元(normal bit) 。現在，我們加入第三種訊號，稱為 x-bit。假設 A_0 代表二位元訊號 0 的反射率，A_1 代表二位元訊號 1 的反射率，而 A_x 表示 x-bit 之反射率，則

$$A_x = \mu + q \tag{2.2.1}$$

$$\mu = \frac{A_0 + A_1}{2} \tag{2.2.2}$$

上式中 μ 為 A_0 和 A_1 的平均值，而 q 是一個隨機變數[8](random variable)。理想上，我們希望 x-bit 的反射率為 μ，對應 q=0。但由於實際製程的不完美，或燒錄的準確率不高，使 x-bit 產生一個誤差值，這個誤差值就由 q 表示。實際應用上，x-bit 反射率之大小可以藉由改變光碟中的「平面」(land)與「微坑」(pits)的反射率，使 x-bit 的反射率更接近 A_0 和 A_1 的平均值 μ。

x-bit 可視為一種人為的錯誤機制加諸於光碟片(硬體)中，等同於在硬體中先放入錯誤機制，如果光碟拷貝不精確，q 的值域會加大，則 x-bit 的反射率與 μ 差距很大，而其目的為防止未經授權的盜拷。此外，由於不知 q 的真正機率分佈，故我們模擬成均勻分佈(uniform distribution)，假定 q 值域中的每一點出現機率相同。

2.3 讀取程序

我們知道光碟是屬於高記錄密度的材料，而光碟主要由多階層的金屬薄膜所組成。當讀取光碟時，不同材質的光碟會產生不同強度的雜訊。在 1967 年時，IBM 發現以 CoCrPtM（M= B，Ni，Ta，W）合金薄膜為材料的光碟產生的雜訊最小，所以現今的光碟都是以 CoCrPtM 合金薄膜為材料。

一般而言，源自於光碟材質的雜訊稱為媒體雜訊(media noise) 。當光碟播放時，最主要的雜訊來自於媒體雜訊和電雜訊(electronic noise)[11]。由於金屬薄膜中的結晶顆粒缺乏足夠的間隔，所以結晶顆粒尺寸會下降，使得結晶顆粒之間的磁交換耦合過強，導致媒體雜訊產生。當讀取光碟時，會將鐳射光轉換成電流，此時會產生電雜訊。我們可以將媒體雜訊和電雜訊模擬成高斯隨機程序[10] (Gaussian random process)。我們讓 $n_m(t)$ 為媒體雜訊，而 $n_e(t)$ 為電雜訊，且 $n_m(t)$ 和 $n_e(t)$ 互相獨立， $h(t)$ 為光碟偵測系統通道脈衝響應，則

$$r(t) = (\sum_k d_k P_T(t - kT_c) + n_m(t)) * h(t) + n_e(t) \qquad (2.3.1)$$

在(2.4.1)中，$r(t)$ 代表接收訊號，d_k 代表 NRZI 訊號，T_c 為通道位元的週期，P_T 為寫入脈波。

$$P_T = 1 \quad , \quad 0 \le t \le T_c \qquad (2.3.2)$$

$$= 0 \quad , \quad \text{else}$$

我們讓 $y(t)$ 代表無雜訊的接收訊號，則(2.4.1)可以改寫成下式 ：

$$r(t) = y(t) + n_m(t) * h(t) + n_e(t) \qquad (2.3.3)$$

$$= y(t) + n(t)$$

在(2.3.3)中，$n(t) = n_m(t) * h(t) + n_e(t)$。我們已知 $n_m(t)$ 、$n_e(t)$ 都是高斯程序，而 $h(t)$

為線性非時變的通道，故 $n(t)$ 亦為高斯程序。

當光碟播放時，可以經由兩次讀取程序(double read process)來偵測 x-bit。注意，此處並不是將整張光碟的內容讀取兩次，而是只讀 x-bit 部份兩次，所以額外花費的讀取時間，只占讀取整張光碟的一小部分，並不會浪費太多時間。

首先，我們必需先考慮讀取一次的錯誤機率。假設 $n(t)$ 為讀取時的雜訊， 我們將此雜訊模擬成高斯隨機程序，其平均值為零(zero mean)，並且假設其變異數為 σ_n^2。如果在特定時間下觀察，此高斯隨機程序可以視為高斯隨機變數(Gaussian random variable)。我們更進一步假設，$n(t)$ 與二位元訊號 0 、二位元訊號 1 和 x-bit 之反射率互為獨立(independence)。

已知 Maximum Likelihood(ML)法則：

$\vec{x}$ ：接收端所得到的觀察向量

$p(m_i)$ ： m_i 事件發生的事前機率

$f(\vec{x}/m_i)$： m_i 事件已發生且接收到 $\vec{x}$ 的機率密度函數(pdf)

$\hat{h}$ ：接收端判定發生的事件

$\forall\, m_i$ ， $f(\vec{x}/m_i)$ 為最大值，則 $\hat{h}=m_i$ 。

若 ξ 代表讀取頭所偵測的反射率，當所偵測的訊號為二位元 0 時

$$\xi = n(t) + A_0 \qquad (2.3.4)$$

同理，偵測的訊號實際上為二位元 1 時，則

$$\xi = n(t) + A_1 \tag{2.3.5}$$

假如 A_h 為最佳判定界限(optimum decision threshold)，由法則可以得知

$$A_h = \frac{A_0 + A_1}{2} \quad = \quad \mu \tag{2.3.6}$$

若 ξ 大於 μ ，讀取頭會認為此位元為 1，如圖 2.3.1 所示。

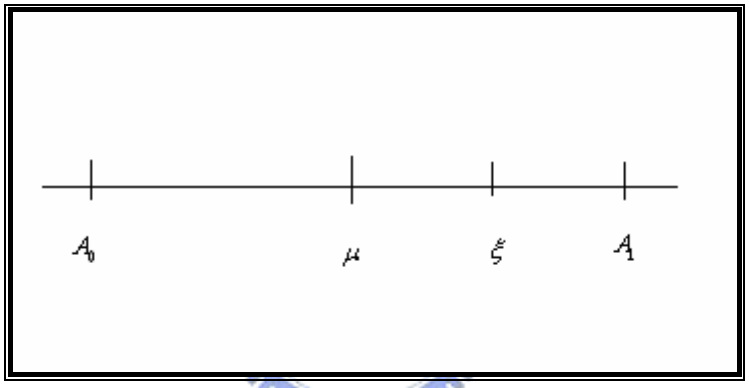

圖 2.3.1 :上圖顯示 $\xi > \mu$ ，故判定為二位元 1。

若 $\xi \leq \mu$ ，讀取頭會認為此位元為 0，如圖 2.3.2 所示。

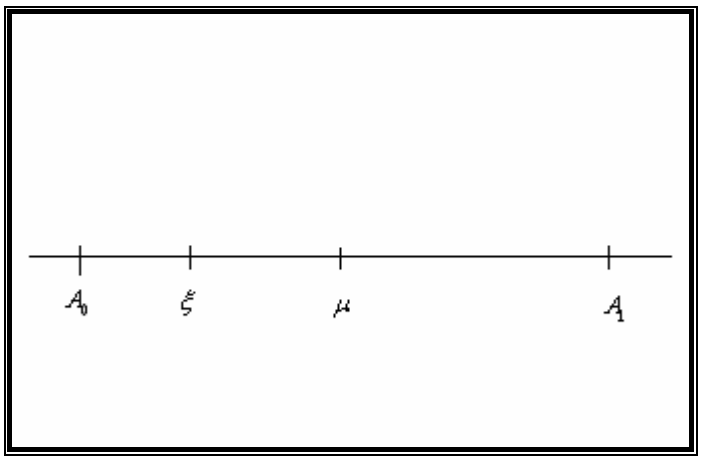

圖 2.3.2 :上圖顯示$\xi < \mu$，故判定為二位元 0。

我們令 $p_{e,0}$ 為偵測二位元訊號 0，但誤判為二位元訊號 1 的機率。由(2.3.7)式，$p_{e,0}$ 為ξ大於μ的機率密度函數(pdf)積分所得之結果，如圖 2.3.3 所示。

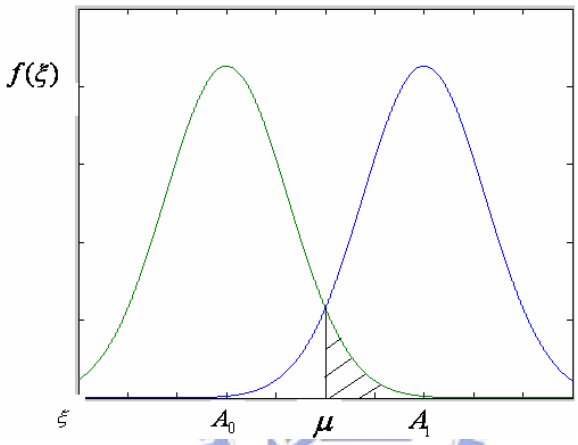

圖 2.3.3 :水平座標為ξ的值域，垂直座標為值域所對應的機率密度函數。左側的高斯曲線為 $f(\xi|A_0)$，右側的高斯曲線為 $f(\xi|A_1)$。

$$P_{e,0}=\int_{\mu}^{\infty}\frac{1}{\sigma_n\sqrt{2\pi}}\exp[-\frac{(x-A_0)^2}{2\sigma_n^{\,2}}]dx \qquad (2.3.7)$$

同理，假設$P_{e,1}$為偵測二位元訊號 1，但誤判為二位元訊號 0 的機率。由(2.3.8)式，$P_{e,1}$為小於μ積分所得之結果，如圖 2.3.4 所示。

$$P_{e,1}=\int_{-\infty}^{\mu}\frac{1}{\sigma_n\sqrt{2\pi}}\exp[-\frac{(x-A_1)^2}{2\sigma_n^{\,2}}]dx \qquad (2.3.8)$$

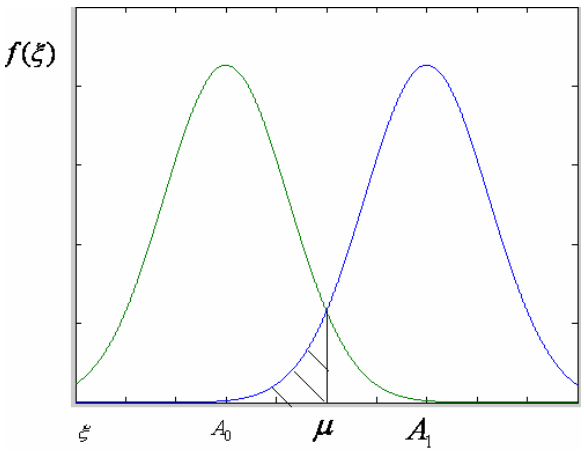

圖 2.3.4 :水平座標為ξ的值域，垂直座標為值域所對應的機率密度函數。左側的高斯曲線為 $f(\xi|A_0)$，右側的高斯曲線為 $f(\xi|A_1)$

我們可將(2.3.7) 和(2.3.8)表示為以下的關係式

$$P_{e,1}=p_{e,0}=\frac{1}{\sqrt{\pi}}\int_{\frac{d}{\sqrt{2}\sigma_n}}^{\infty}e^{-y^2}dy=\frac{1}{2}erfc(\frac{d}{\sqrt{2}\sigma_n}) \tag{2.3.9}$$

$$d=\frac{A_1-A_0}{2} \tag{2.3.10}$$

在這裡 erfc(.)稱為補誤差函數(complementary error function)，表示如下：

$$erfc(z)=\frac{2}{\sqrt{\pi}}\int_z^{\infty}e^{-y^2}dy \tag{2.3.11}$$

若讀寫頭所偵測到的訊號為 x-bit 時，則

$$\xi = n(t) + A_x = n(t) + \mu + q \qquad (2.3.12)$$

若ξ大於μ，讀取頭會認為此位元為 1；若$\xi \le \mu$，讀取頭會認為此位元為 0。換言之，若 x-bit 只讀取一次，所判定的結果不是二位元訊號 0，就是二位元訊號 1，其機率決定於反射率A_x。假定P_{X0}為讀寫頭偵測 x-bit，但被判定成二位元訊號 0 的機率。如下圖所示，當$\xi \le \mu$時，讀取頭會判定此位元為 0，因此

$$P_{X0} = \int_{-\infty}^{\mu} \frac{1}{\sigma_n \sqrt{2\pi}} \exp[-\frac{(x - A_x)^2}{2\sigma_n^2}]dx$$

$$= \int_{-\infty}^{\mu} \frac{1}{\sigma_n \sqrt{2\pi}} \exp[-\frac{(x - \mu - q)^2}{2\sigma_n^2}]dx \qquad (2.3.13)$$

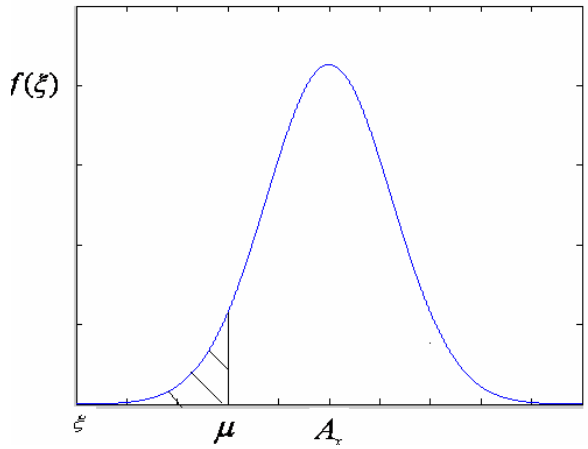

圖 2.3.5：水平座標為ξ的值域，垂直座標為值域所對應的機率密度函數。此圖中$A_x > \mu$，故q>0，對斜線上方的曲線積分可得P_{X0}。

同理，若P_{X1}代表讀寫頭偵測 x-bit，但被判決成二位元訊號 1 的機率，則

$$P_{X1} = \int_{\mu}^{\infty} \frac{1}{\sigma_n \sqrt{2\pi}} \exp[-\frac{(x-\mu-q)^2}{2\sigma_n^2}]dx = 1 - P_{X0} \qquad (2.3.14)$$

藉由變數變換，以上結果可以改寫如下：

$$P_{X0} = \frac{1}{\sqrt{\pi}} \int_{\frac{q}{\sqrt{2}\sigma_n}}^{\infty} e^{-y^2} dy = \frac{1}{2} erfc(\frac{q}{\sqrt{2}\sigma_n}) \qquad (2.3.15)$$

$$P_{X1} = 1 - \frac{1}{2} erfc(\frac{q}{\sqrt{2}\sigma_n}) \qquad (2.3.16)$$

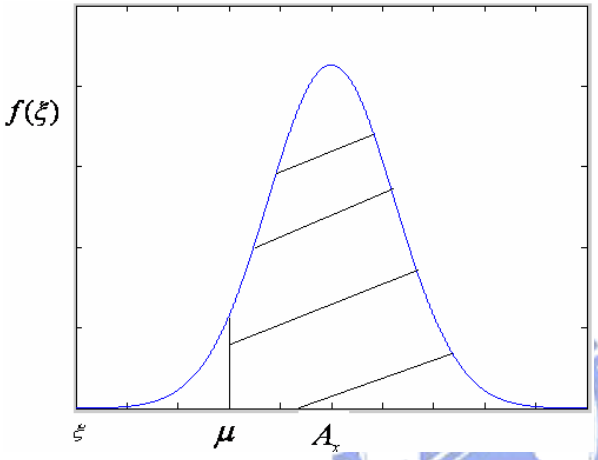

圖 2.3.6：水平座標為ξ的值域，垂直座標為值域所對應的機率密度函數。此圖中$A_x > \mu$，故q>0，對斜線上方的曲線積分可得P_{X1}。

2.4 兩次讀取程序(double read process)

在兩次讀取程序中，我們對 x-bit 讀取兩次，若兩次的結果不一樣，判定它為 x-bit。若兩次的結果一樣，不是判定為 0，就是判定為 1，所以對市售的光碟機而言，偵測 x-bit 並不困難。

第一次偵測	第二次偵測	判定結果
1	0	x-bit
0	1	x-bit
1	1	1
0	0	0

圖 2.4.1:說明 x-bit 的判別法則。

考慮 x-bit 的反射率 $A_x = \mu+q$，經由兩次讀取程序，正確判定 x-bit 的機率為 $P_{c,X}$。$P_{c,X}$ 為 x-bit 第一次判成 0，第二次判成 1，或 x-bit 第一次判成 1，第二次判成 0 之機率和，故

$$P_{c,X} = P_{X0}P_{X1} + P_{X1}P_{X0} \tag{2.4.1}$$

由(2.3.15) 、(2.3.16)與(2.4.1)，可獲得 x-bit 誤判的機率 $P_{e,X}$ 為

$$\begin{aligned} P_{e,X}(q) &= 1 - P_{c,X} \\ &= 1 - erfc(\frac{q}{\sigma_n\sqrt{2}})[1 - \frac{1}{2}erfc(\frac{q}{\sigma_n\sqrt{2}})] \end{aligned} \tag{2.4.2}$$

令 $q' = \frac{q}{\sigma_n}$，而 q' 為正規化後的 q 參數(normalized q parameter) 。圖 2.4.2 為 $P_{e,x}$ 與 q' 的關係圖。我們發現 $P_{e,X}$ 對稱於 q' 且 $P_{e,X}$ 的最小值為 0.5，發生在 $q' = 0$ 時。這代表透過兩次讀取程序，至少有 50% 的 x-bit 無法正確偵測。

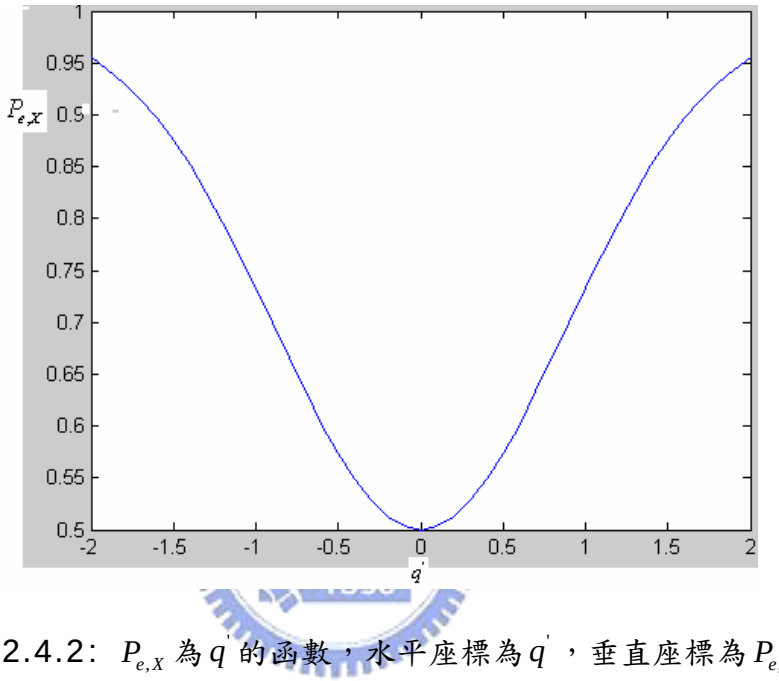

圖 2.4.2: $P_{e,X}$ 為 q' 的函數，水平座標為 q' ，垂直座標為 $P_{e,X}$ 。

實際上，由於製程的不完美，我們必需去考慮 x-bit 的波動。假設 A_x 是均勻分佈在 $[\mu-\alpha,\mu+\alpha]$ 這個區間，在這裡 α 定義為 A_x 對 μ 的最大偏差量，而 q 則均勻分佈在 $[-\alpha,\alpha]$ 之間，其機率密度函數可以表示如下：

$$p_q(q)=\frac{1}{2\alpha} \quad , \quad -\alpha \le q \le \alpha \qquad (2.4.3)$$

$$= \quad 0 \quad , \quad \text{otherwise}$$

由貝氏定理

$$\sum P(A/B)P(B) = P(A) \qquad (2.4.4)$$

利用上式可以求得偵測 x-bit 時之平均錯誤率為

$$P_{e,X}(\alpha) = \int_{-\alpha}^{\alpha} P_q(q)P_{e,X}(q)dq = \frac{1}{2\alpha}\int_{-\alpha}^{\alpha} P_{e,X}(q)dq \qquad (2.4.5)$$

令 $\alpha' = \frac{\alpha}{\sigma_n}$ ，稱 α' 為正規化後的 α 參數(normalized α parameter)。圖 2.4.3 為 $P_{e,x}$ 與 α' 的關係圖。如我們所預期的， $P_{e,X}$ 會隨著 α' 的增加而變大，而 $P_{e,X}$ 的最小值為 0.5，發生在 α'=0。α' 是一個重要因子(critical factor) ，我們稍後會討論到。

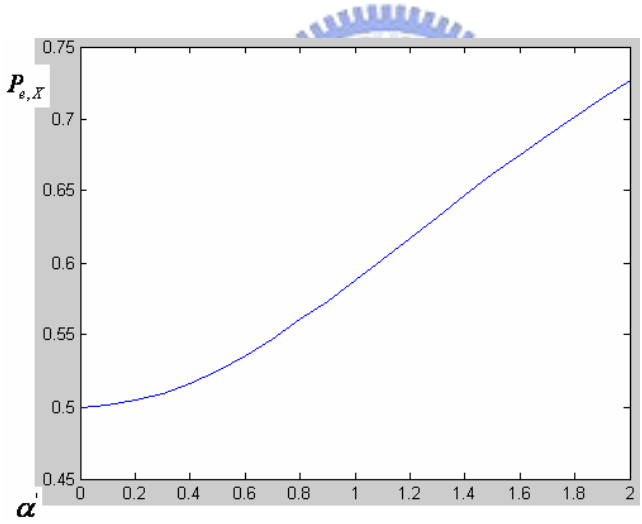

圖 2.4.3: $P_{e,X}$ 為 α' 的函數，水平座標 α' ，垂直座標為 $P_{e,X}$ 。

第三章 編碼與解碼

3.1 The Multi-level Word-extension

首先，我們介紹 word-extension 程序。一個二位元字由 v 個位元組成，我們可以將這個二位元字視為一個基本單位，故稱它為單位字(unit-word) 。令 y 為一

個 unit-word，

$$y = (b_1, b_2, ...b_v) \quad ， b_j \in \{0,1\} \tag{3.1.1}$$

我們利用 word-extension process 將 y 展開為一個碼字 D = $(d_1, d_2, ...d_m)$，在這裡 d_i 是一個 unit-word，即

$$d_i = (d_{i1}, d_{i2}, ...d_{iv}) \quad ， d_{ij} \in \{0,1\} \tag{3.1.2}$$

我們欲得到 y 與 D 的關係式如下：

$$\begin{pmatrix} d_{11} \oplus d_{21} \oplus ..... \oplus d_{m1} \\ d_{12} \oplus d_{22} \oplus ..... \oplus d_{m2} \\ \vdots \\ d_{1v} \oplus d_{2v} \oplus ..... \oplus d_{mv} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_v \end{pmatrix} \tag{3.1.3}$$

在這裡，符號⊕表示為二位元加法(mod-2 addition)。等式(3.1.3)說明了在 y 中的任一個位元，可以由 D 中所對應的位元做二位元加法求得。對於任意給定的 y，word-extension 可以很容易實現。首先隨機產生 $(d_1, d_2, ..., d_{m-1})$，接著 d_m 可以如下獲得：

$$\begin{pmatrix} d_{11} \oplus d_{21} \oplus ..... \oplus d_{(m-1)1} \oplus b_1 \\ d_{12} \oplus d_{22} \oplus ..... \oplus d_{(m-1)2} \oplus b_2 \\ \vdots \\ d_{1v} \oplus d_{2v} \oplus ..... \oplus d_{(m-1)v} \oplus b_v \end{pmatrix} = \begin{pmatrix} d_{m1} \\ d_{m2} \\ \vdots \\ d_{mv} \end{pmatrix} \tag{3.1.4}$$

由(3.1.4)，我們可以得到以下等式：

$$d_{1j} \oplus \cdots\cdots \oplus d_{(m-1)j} \oplus d_{mj}$$

$$= [d_{1j} \oplus \cdots\cdots \oplus d_{(m-1)j}] \oplus [d_{1j} \oplus \cdots\cdots \oplus d_{(m-1)j}] \oplus b_j$$

$$= b_j \qquad (3.1.5)$$

因此，假如 d_m 被給定如等式(3.1.4) ，則等式(3.1.3)可以被滿足。我們命名碼字 D 為延展碼字(word-extended codeword)，簡稱 EC。

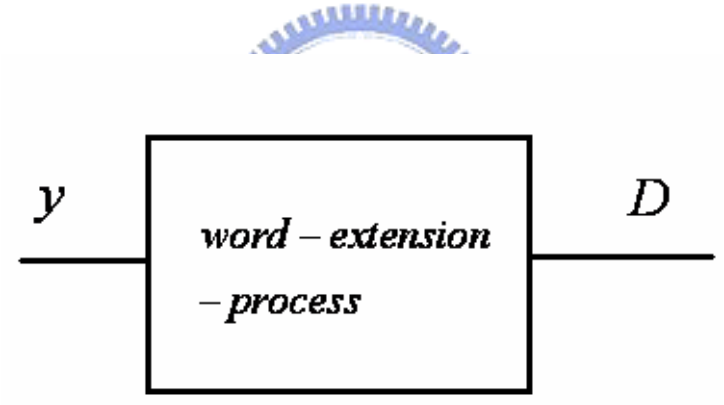

圖 3.1.1 :一個 unit-word y 輸入，含 m 個 unit-word 的碼字 D 輸出。

3.2 Error-coding process

在 word-extension 後，error-coding process 接著被執行。我們採用前向錯誤更正碼(forward error-correcting code)，簡稱 FEC，例如一個(s，m) Reed-Solomon code(RS-code) ，其每個符號(symbol)由 v 個位元組成，可以用來將延展碼字 D 作 error coding，所生成的碼字 E 由 s 個 unit-word 組成，稱作 FEC-coded codeword(FC) 。如圖 3.2.1 所示，word-extension/error-coding process 是將一個 unit-word y ，先 extend 為碼字 D，之後經過 FC 的編碼，所產生的碼字 E 則具

有錯誤更正的能力。

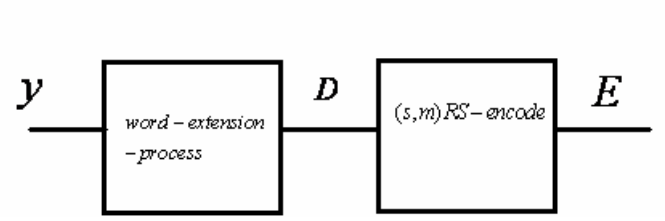

圖 3.2.1 :word-extension/error-coding process 示意圖

在所提的防盜拷程序中，原始鑰匙(key)由 k 個 unit-word 組成，此鑰匙先經過 multi-level word-extension error-coding (簡稱 MWE)的方式編碼，接著再安插 x-bit 至編碼後的結果，以有效預防未經授權的拷貝。

第一步，我們將 k 個 unit-word 組成的原始鑰匙先經過 (n,k) RS-code 編碼成一個碼字 C，$C=(c_1,c_2,.....,c_n)$，c_i 為一個 unit-word。例如 v=6 和 k=13，原始鑰匙由 13 個 unit-word 組成，每個 unit-word 由 6 個位元組成。我們利用一個(63，13)的 RS-code 可得到碼字 C，而 C 由 63 個 unit-word 組合而成。

第二步，碼字 C 再以 MWE 程序編碼。考慮 C 中的任意 unit-word c_i，在第一階($1^{st}-level$)的 word-extension 裡，先將 c_i 展開為長度 m 個 unit-word 的 EC，簡稱第一階 EC($1^{st}-level$ EC)，接著再使用(s，m) RS-code 將此 EC 編碼成 $1^{st}-level$ FC。因為 C 由 n 個 unit-word 組合而成，故 $1^{st}-level$ FC 的總個數 $N_1=n$，且每個 FC 由 s 個 unit-word 組成。

假設 E 為任一 $1^{st}-level$ FC。在第二階編碼中，E 中的每一個 unit-word 再經由 word-extension / error-coding 程序產生一個 $2nd-level$ FC，因為 E 由 s 個 unit-word 組成，所以 E 產生的 $2^{nd}-level$ FC 共有 s 個，故 $2^{nd}-level$ FC 的總個數 $N_2=N_1\cdot s=\ n\cdot s$。藉著 MWE 編碼，相同的過程被應用在任意階 FC。若共有 L 階的 FC，我們可以由歸納法則，輕鬆的求得 L 階 FC 的總個數 N_L 為

$$N_L = N_{L-1} \cdot s = n \cdot s^{L-1} \quad (3.2.1)$$

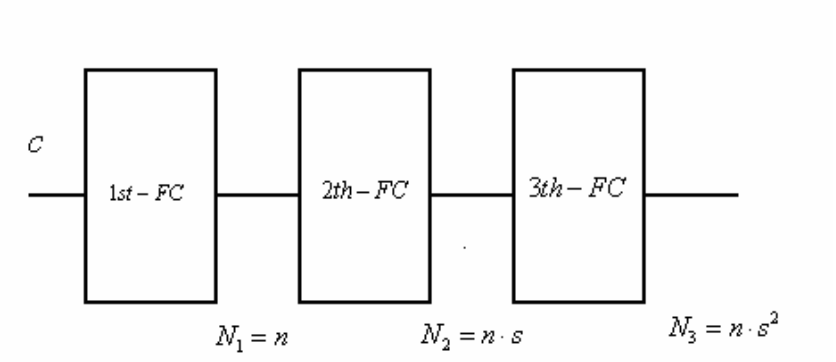

圖 3.2.2：MWE 編碼流程

3.3 X-bit 編碼

如前所述，第 L 階 FC 可以經由 MWE 程序產生。令 t 為(s，m)RS-code 之更錯能力(error-correcting capability) ，則

$$t = \frac{s-m}{2} \quad (3.3.1)$$

若 $Y = (y_0, y_1, ..., y_{s-1})$ 為第 L 階 FC 之一，在這裡，y_i 為 unit-word，故共有 s 個 unit-word 在 Y 中。在 s 個 unit-word 中，我們只挑出 u 個 unit-word 加入 x-bit，而

$$u = t + b \quad (3.3.2)$$

其中 b 是一個整數且 $b \geq 0$。這些編入 x-bit 的 unit-word 稱之為 X-word，其位置是固定且已知的。考慮一個特別的 unit-word $y = (b_1, b_2, ..., b_v)$，在這裡 $b_i \in \{0,1\}$。我們將 y 對應至 θ，θ 為一個 X-word，且 $\theta = (\rho_1, \rho_2, ..., \rho_v)$，又 $\rho_i \in \{0,1,X\}$，

其中 X 意指 x-bit。假定 β 為 θ 中 x-bit 的個數，θ 由 v 個位元組成，又 y 也是由 v 個位元組成，故有 2^v 種可能。所以 θ 的種類數目要大於或等於 y 的種類。換言之，就是要大於 2^v 個，即

$$C_{\beta}^{v} \cdot 2^{v-\beta} \geq 2^{v} \tag{3.3.3}$$

假如(3.3.3)式能滿足，則我們必定可以設計一個對照表，使 y 與 θ 有著一對一的對應關係。下表為 y 與 θ 的轉換表，其中 v=4，$\beta = 2$.

b1	b2	b3	b4	ρ1	ρ2	ρ3	ρ4
0	0	0	0	X	0	1	0
0	0	0	1	X	0	1	1
0	0	1	0	X	1	0	1
0	0	1	1	X	1	1	0
0	1	0	0	0	X	1	0
0	1	0	1	0	X	1	1
0	1	1	0	1	X	0	1
0	1	1	1	1	X	1	0
1	0	0	0	0	1	X	0
1	0	0	1	0	1	X	1
1	0	1	0	1	0	X	1
1	0	1	1	1	1	X	0
1	1	0	0	0	1	0	X
1	1	0	1	0	1	1	X
1	1	1	0	1	0	1	X
1	1	1	1	1	1	0	X

表 3.1：x-bit 編碼的對應關係，是一個 v=4 的例子。

圖 3.3.1 顯示整個編碼過程，包含 MWE 程序與 x-bit 編碼。在圖中，原始鑰匙先經過(n,k)RS-code 編碼成 C，再將 C 以第一階的 MWE 程序編碼，故可以得到 N_1 個第一階 FC。重複同樣的過程直到第 L 階，接著在每個第 L 階 FC 插入 u 個 X-word，這些加入 X-word 的 L 階 FC 被存入光碟的開端部分(lead-in sector) 。播放光碟時，會強制讀取這個部份，以有效的防止盜拷光碟。

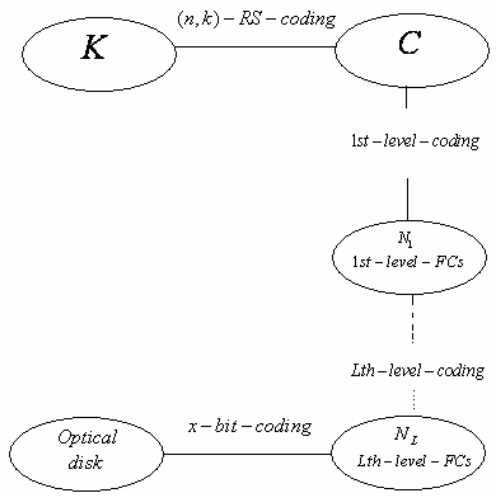

圖 3.3.1:原始鑰匙轉變為加密鑰匙的過程。

3.4 解碼過程

如之前所述，共有 N_L 個第 L 階 FC 被存入光碟的開端部分。我們設計解碼程序安裝在播放機，當播放光碟時，此解碼程序將讀取開端部分兩次以偵測加密鑰匙。令 $W=(w_0,w_1,.....,w_{s-1})$ 為一個第 L 階 FC，則 W 中 X-word 的數目有 u 個，其餘 $s-u$ 個 unit-word，我們通稱為正規字(normal-word) 。

若 $w_j=(\rho_1,\rho_2,.....,\rho_v)$ 為 W 中的任意一個 X-word，它含有 β 個 x-bit。假使 w_j 中的 x-bit 能經由兩次讀取程序正確偵察出，則 w_j 必能憑藉著對照表，正確無誤地還原成原來的 unit-word。假使一個或是一個以上的 x-bit 無法偵測出，這時播放端的解碼程序會隨機指定一個 unit-word 來代表 w_j 所對應的內容。若 x-bit 無法順利偵測出，則隨機指定的 unit-word 碰巧猜中 w_j 所對應的內容之機率為 $\frac{1}{2^v}$ 。

w_j所代表的內容	1	0	0	1
w_j	X	X	1	1
第一次偵測結果	0	1	1	1
第二次偵測結果	1	1	1	1
判定結果	X	1	1	1
給定的隨機字	0	0	0	0

圖 3.4.1:為一個 $v=4$ 的例子，說明 x-bit
若無法順利偵測出，會隨機給定一個 unit-word。

如下圖所示，解碼程序為編碼程序的反向動作，先將編入 X-word 的第 L 階 FC 經過 x-bit decoding，解出第 L 階的 FC 且不含 X-word，再將 N_L 個第 L 階 FC 通過 L 次的 FC-decoder，使碼字 C 順利解調出。最後再將 C 以(n，k) RS-decoder 解碼，則原始鑰匙(key)應運而生。

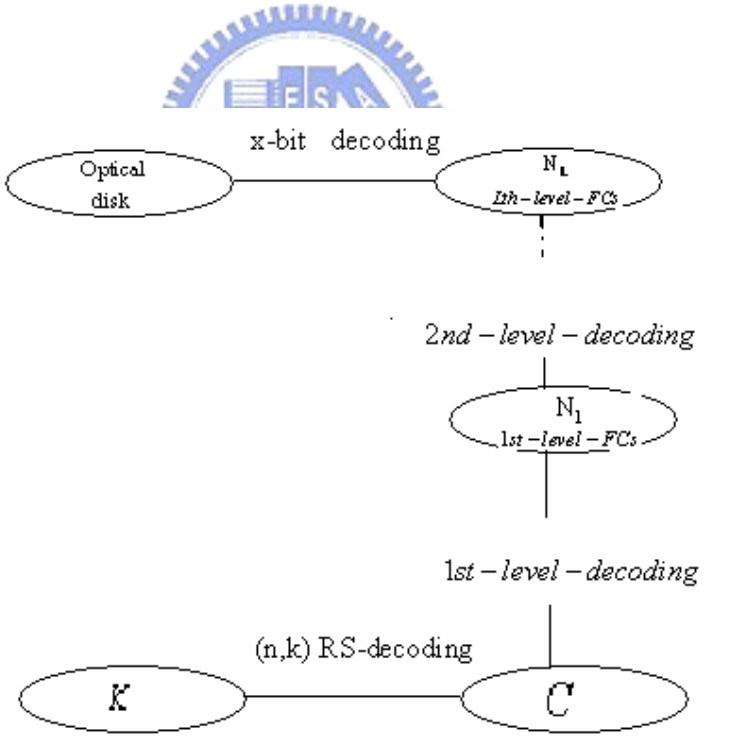

圖 3.4.2:整個解碼過程。

3.5 系統效能分析

假設W為一個第L階FC，且W中含有u個X-word。W可以表示為W=($w_0, w_1, ..., w_{s-1}$)，而X-word的位置為固定且已知。假定P_e表示為一個正規位元(normal-bit)偵測錯誤的機率，而正確偵測一個正規字(normal-word)之機率以P_n表示之。因為一個正規字由v個位元組合，因此

$$P_n = (1-P_e)^v \tag{3.5.1}$$

由先前所說，經由兩次讀取程序偵察x-bit錯誤之機率$P_{e,X}$，可以由等式(2.5.5)求出，又$P_{e,X}$為正規化參數α'的函數，且$P_{e,X}$的最小值等於0.5，發生在$\alpha' = 0$時。令P_d為正確偵測一個X-word的機率，在這裡P_d可以用條件機率(conditional probability)表示為

$$P_d = (1-P_{e,X})^{\beta}(1-P_e)^{v-\beta} + [1-(1-P_{e,X})^{\beta}]\cdot\frac{1}{2^v} \tag{3.5.2}$$

在(3.5.2)式裡，等式右邊的第一項表示偵測一個X-word時，β個x-bit與$v-\beta$個normal bit同時正確的機率，而第二項表示至少有一個x-bit無法正確偵測，但播放機指定的隨機字碰巧與X-word所對應的內容相同的機率。

因為$P_{e,X}$為α'的函數，由(3.5.2)可知P_d亦為α'的函數，故P_d會隨著α'改變。我們發現當α'增加時，P_d會隨之下降。由於 $\alpha' = \frac{\alpha}{\sigma_n}$，故$\alpha'$增加時，$\alpha$也跟著

增加，已知α為A_x對μ最大偏差量，則α越大，代表 x-bit 越不精準。

當α'等於零時，P_d產生最大值，但P_d的最大值還是很小，表示經由兩次讀取程序偵測 X-word 正確的機率不高。圖 3.5.1 為一個$v=6$，$\beta=2$和$P_e=10^{-5}$的例子，圖中當α'增加時，P_d的遞減量很小，故我們並不能從P_d準確的區分出 x-bit 的精準度，這是我們希望改善的，其方法稍後會提出。

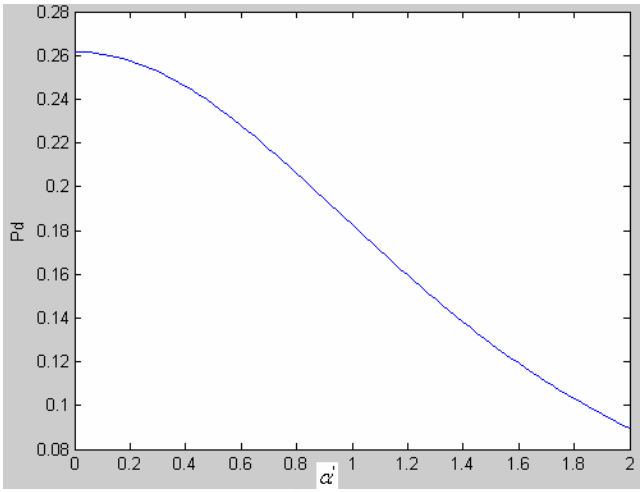

圖 3.5.1:水平座標為α'，垂直座標為P_d。

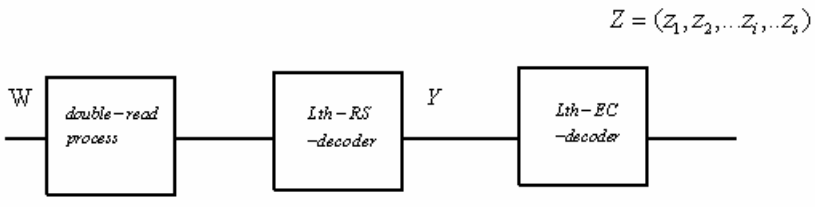

圖 3.5.2:顯示整體系統的解碼流程圖

已知有 u 個 X-word 與 s-u 個正規字在 W 中，當 W 經過兩次讀取程序後，剛好有 r 個 unit-word 發生錯誤的機率以 $P_L(r)$ 表示。假設 r 個錯誤中，有 i 個錯發生在 X-word 中，而剩下的 $r-i$ 個錯源自於正規字中，則

$$P_L(r) = \sum_{i=0}^{r} prob(i\ X-words\ erroneous) \cdot prob[(r-i)\ normal\ words\ erroneous]$$

$$= P(r) = \sum_{i=0}^{r} [C_i^u (1-P_d)^i P_d^{u-i}] \cdot [C_{r-i}^{s-u} (1-P_n)^{r-i} P_n^{(s-u)-(r-i)}] \qquad (3.5.3)$$

如圖 3.5.2，假設 Y 是第 L 階 EC，且 Y 來自於 W 經過 $Lth-RS-decoder$。假定 t 為 $RS-decoder$ 的最大更錯能力(max number of correctable errors)，若 $r \le t$，則 Y 會正確解碼。故 Y 正確的機率以 P_L 表示為

$$P_L = \sum_{r=0}^{t} P_L(r) \qquad (3.5.4)$$

我們繼續分析第 $(L-1)$ 階的效能。令 $Z = (z_1, z_2, ..., z_i, ..., z_s)$ 為一個第 $(L-1)$ 階 FC，Z 由 s 個 unit-word 所組成。又在 Z 中剛好有 r 個 unit-word 錯誤的機率以 $P_{L-1}(r)$ 表示為

$$P_{L-1}(r) = C_r^s (1-P_L)^r P_L^{s-r} \qquad (3.5.5)$$

假如 $r \le t$，代表 Z 中錯誤個數 r 小於等於 $(L-1)th-RS-decoder$ 的更錯能力 t，則 $(L-1)th-EC$ 會正確，故第 $L-1$ 階的 EC 正確的機率以 P_{L-1} 表示為

$$P_{L-1} = \sum_{r=0}^{t} P_{L-1}(r) = \sum_{r=0}^{t} C_r^s (1-P_L)^r P_L^{s-r} \qquad (3.5.6)$$

以上的程序可依此類推直到獲得第一階 EC 正確機率 P_1。由數學歸納法，我們可以得到前後兩階 EC 正確的機率為

$$P_{j-1} = \sum_{r=0}^{t} C_r^s (1-P_j)^r P_j^{s-r} \qquad , j = 2, 3, ..., L \qquad (3.5.7)$$

在這裡，P_j 代表第 j 階 EC 正確的機率，而 P_{j-1} 代表第 $j-1$ 階 EC 正確的機率，由(3.5.7)式可以知道不同階 EC 正確的機率為一疊代關係(recursive relationship)。

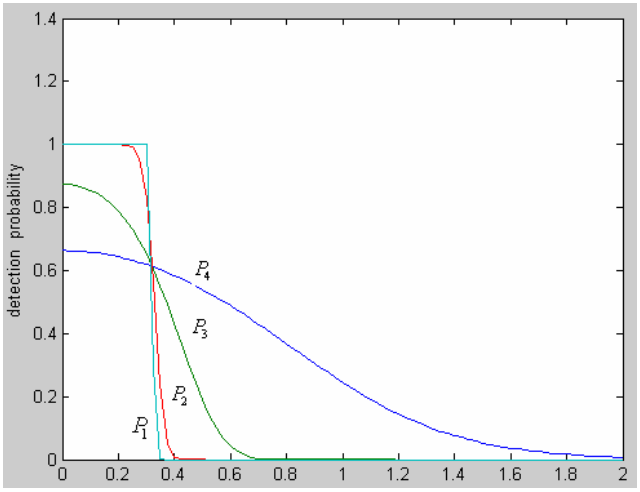

圖 3.5.3: P_j 在不同階數下對 α' 的圖形，在此，$L=4$，s=63，m=13，b=8, v=6，$\beta=2$ u =33。

圖 3.5.3 顯示 P_j 在不同階數下對 α' 的圖形。由圖中可以得到一個很重要的結果，即各階 EC 正確的機率會相交在同一點，我們令這點的水平座標為 α'_c，垂直座標為 P_{cutoff}，則 (α'_c, P_{cutoff}) 為此交點的座標表示式，又 P_{cutoff} 稱為截止機率 (cutoff- probability) 。

我們還可以發現階數越小的 P_j，其對 α' 的圖形越接近方波，所以圖 3.5.3 中以 P_1 最接近方波，因為 P_1 的階數為最小。當 $\alpha' = \alpha'_c$ 時，$P_1 = P_{cutoff}$，在 $\alpha' < \alpha'_c$ 的區域中，所對應的 P_1 接近 1，而在 $\alpha' > \alpha'_c$ 的區域中，所對應的 P_1 接近 0。

以上的結果可以說明如下，考慮 P_4 與 P_3 在圖 3.5.3 所對應的曲線，當 P_4 與 P_3

相交在點(α'_c, P_{cutoff})，則$\alpha' = \alpha'_c$，$P_3 = P_4 = P_{cutoff}$，這告訴我們 FC-decoder 程序並不影響 EC 正確的機率。而在$\alpha' < \alpha'_c$的區域中，$P_3 > P_4$，代表越低階的 EC 正確的機率越高；相反地，在$\alpha' > \alpha'_c$之區域中，$P_3 < P_4$，則越低階的 EC 正確的機率越低，如此的結果發生在各階之中。當 j 減少時，在$\alpha' < \alpha'_c$的區域中，P_j會持續的增加；當 j 減少時，在$\alpha' > \alpha'_c$之區域中，P_j會持續的減少。最後我們可以得到在$\alpha' < \alpha'_c$的區域中P_1非常接近 1，而在$\alpha' > \alpha'_c$之區域中，P_1非常接近 0。

在多次的 FC-decoder 後，我們可以得到第一階 EC 正確的機率P_1，而碼字 C 也可以得到，之後將C通過$(n,k)-RS-decoder$可以還原鑰匙(key)。在這裡，$C = (c_1, c_2, ..., c_n)$，C 由 n 個 unit-word 所組成，如果C中錯誤 unit-word 個數r小於等於$(n,k)-RS-decoder$之更錯能力時，原始鑰匙可以順利獲得。若C中錯誤 unit-word 個數r大於$(n,k)-RS-decoder$之更錯能力時，會得到錯誤的鑰匙，則播放出來的內容與原本的不一樣。因為$RS-code$更錯能力為$\frac{n-k}{2}$，令原始鑰匙順利回復的機率以P_{key}表示，則

$$P_{key} = \sum_{r=0}^{\frac{(n-k)}{2}} C_r^n (1-P_1)^r P_1^{n-r} \qquad (3.5.9)$$

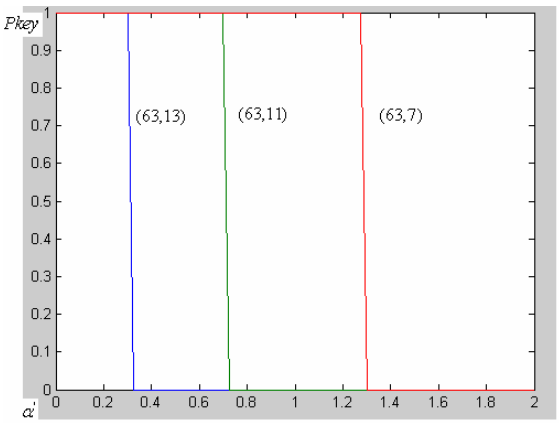

圖 3.5.4: P_{key} 對 α' 在不同(s，m) $RS-code$ 下正確的機率，

最左邊的方波圖形為(63,13)的 $RS-code$，由左而右依次為(63,11)、$(63,7)$ $RS-code$ ，在這裡，$L=4$，b=8，$n=63$，$k=11$。

圖 3.5.4 中，P_{key} 截止行為對於防盜拷是很重要的發現。我們回想 α 是 A_X 對 μ 的最大偏差量，P_{key} 截止行為顯示當 $\alpha' < \alpha_c'$ ，原始鑰匙可以被順利回復。由於 $\alpha' = \frac{\alpha}{\sigma_n}$，可以推得 $\alpha < \alpha_c'\sigma_n$ 時，鑰匙會順利回復。因為 x-bit 的反射率 $A_X = \mu + q$，又 q 介於 $[-\alpha, \alpha]$，故 A_X 落在 $[\mu-\alpha, \mu+\alpha]$ 區間中。將 α 的上限 $\alpha_c'\sigma_n$ 代入，若原始鑰匙要順利復原，A_X 必須落在 $[\mu-\alpha_c'\sigma_n, \mu+\alpha_c'\sigma_n]$ 區間中。

在圖 3.5.4 中，對於(63，13) RS-code 而言，$\alpha_c' \approx 0.325$，若鑰匙要順利復原，須滿足 $\alpha < (0.325)\cdot\sigma_n$。這如同一個很緊的邊界，以 α_c' 限制 x-bit 的反射率 A_X，而讀取時會產生雜訊，其標準差 σ_n 不同於其它的通訊系統之標準差，在讀取的過程中產生的 σ_n 通常很小。

對於正版光碟的製造商，他們可以設計很好的設備，藉由這些設備將 x-bit 的反射率 A_X 控制在 $[\mu-\alpha_c'\sigma_n, \mu+\alpha_c'\sigma]$ 區間內。相對地，要將 x-bit 的反射率 A_X 控制在 $[\mu-\alpha_c'\sigma_n, \mu+\alpha_c'\sigma]$ 區間內，對於盜版商而言是非常困難的。基於人性，會去盜版光碟的人都不是真正的有錢人，他們不會願意花費鉅額的資金來生產品質好的產品。故把握這樣的想法，真正有錢的人不會願意當盜版商，而盜版商不會願意花大錢，在成本考量下，可以有效的抑制盜拷行為。

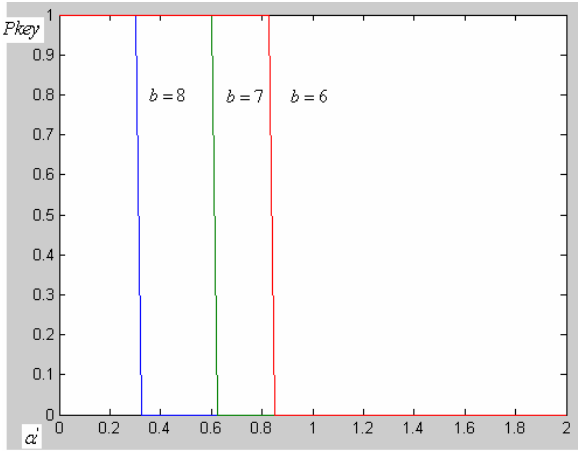

圖 3.5.5: P_{key} 對 $\alpha_c^{'}$ 在不同的 b 下，使用相同的(s，m) $RS-code$ ，在這裡，$L=4$，s=63，m=13，n=63，k=13。

圖 3.5.5 顯示參數 b 對 P_{key} 的影響，所以 b 是一個重要的參數，我們能夠利用它得到我們想要的截止曲線。b 增加時，代表第 L 階 FC 中 X-word 的個數增加，相對地，正規字的個數減少。

3.6 系統評比

過去所提出的防盜拷技術，不外乎是對光碟的內容做加密，而播放端的部份鮮少有人去討論。我們的做法為利用硬體與軟體加密光碟，藉此打擊專業的盜版商。一般而言，是沒有辦法可以絕對杜絕盜拷行為，所以我們能做的，只是提高盜拷的困難度。我們的系統，可以提高拷貝光碟的困難度，藉著加入 x-bit 與使用 MWE 程序，如此在拷貝光碟時，必需使用極精密的燒錄設施。如果高精密的燒錄設備被管控的很好，無法任意外流，則一般大眾無法使用低成本的燒錄器拷貝光碟，同時專業的盜版商也能被預防。

MWE 程序實際上包含三種機制，第一 ： 錯誤安插(error insertion)，我們安插 x-bit 至光碟中，可以想成主動放入潛在性錯誤至光碟中，用來防止位元對位元的盜拷。第二 ： 錯誤更正(error correction)，雖然錯誤被加入到光碟中，但由於 $RS-code$ 具有更錯能力，能夠回復鑰匙。第三 ： 錯誤散佈(error propagation)，在多次的 FC-decoder 之中，第 j 階 EC 正確的機率以 P_j 表示。如果 P_j 小於 P_{cutoff} ，錯誤將會影響 $j-1$ 階，所以 P_{j-1} 將會低於 P_j 。當 P_j 大於 P_{cutoff} 時，錯誤會越來越多，造成錯誤散佈的現象。根據這些機制，P_{key} 對 $\alpha_c^{'}$ 的圖形將以方波的樣子呈現，這是我們所樂見的結果。我們主動放入潛在性的錯誤至光碟中，但如果安插的太多，超過系統的更錯能力時，P_{key} 對 $\alpha_c^{'}$ 的圖形將不再以方波的樣子呈現，而是降至 0，故 x-bit 並不能無限的加入。

在我們提出的系統中，假如專業的盜版商設法去偵測 $x-bit$ ，透過多重的讀取程序(multiple read-detect process)，光碟裡全部 $x-bit$ 的位置是可以獲得的，故原始鑰匙的內容是可以得知的。但是知道鑰匙的內容並沒有用處，因為解碼程序被安置在播放機，這是無法修改的，盜版商無法去修改每一台光碟機的播放程式。當拷貝的光碟被播放時，解碼程序會偵測光碟中的 $x-bit$ ，若光碟中根本沒有 $x-bit$ ，或 $x-bit$ 的反射率 A_X 沒有落在 $[\mu-\alpha_c^{'}\sigma_n, \mu+\alpha_c^{'}\sigma_n]$ 區間內，則播放機會隨機給一把鑰匙，此錯誤的鑰匙與光碟中的內容做二位元加法，會產生不正確的資料。

實際上，隱藏鑰匙並不是我們的目標，雖然很多防盜拷程序採用此方法，我們傾向於提高拷貝光碟的困難度，在我們提出的結論當中，$\alpha_c^{'}$ 扮演一個很重要的角色，又 $\alpha^{'} < \alpha_c^{'}$ 時，原始鑰匙可以復原，我們希望能降低 $\alpha_c^{'}$ ，藉以提升拷貝光碟的困難度，在下一個章節都會以此為主要目標。

第四章 變形的系統

4.1 一對多的編碼與解碼

有鑑於圖 3.5.1，X-word 正確機率 P_d 對於不同 α' 下，其差距並不大，我們希望 P_d 對 α' 的斜率能提升，這表示隨著 x-bit 製造的準確度稍加改變，X-word 正確偵測的機率卻有著大幅度改變，故能提高盜拷光碟的困難度。

我們利用一個例子來說明我們的想法。考慮一 unit-word 由三個位元組成，讓此 unit-word 經過 word-extension process，產生長度為三個 unit-word 的 EC。換言之，此 EC 長度為九個位元，再加入 x-bit 成為 X-word。例如 101 是一個 unit-word 先將它展成 010　111　000 的 EC (由(3.1.4)得到)，接著將此 EC 編入 x-bit 成為 010　1xx　0xx。 x-bit 並不是任意加至 EC，EC 中的每一 unit-word，其開頭第一個位元不能加入 x-bit，這是以解碼端做考量，我們稍後會論述。

假設一個 unit-word 由 v 個位元組成，故有 2^v 個可能組合，則所對應的 X-word 之種類也要大於或等於 2^v 個可能。基於此點考量，EC 中的第一個 unit-word 我們不編入 x-bit，而 x-bit 只加入在第二個 unit-word 的第二或三個位元，依此類推。如上例，X-word 之種類等於 $2^5 \cdot 3^4$ 個，遠大於原始 unit-word 種類數目 2^3 個。

解碼時，我們放寬要求，不再要求所有的 x-bit 都要能正確偵測出，若 X-word 中有 β 個 x-bit，經兩次讀取程序，至少能偵測出 g 個以上的 x-bit，就認為此

X-word 正確($g < \beta$)，再經由對照表(mapping table)恢復成原始 unit-word。

0	1	0	0	x	x	1	x	x
0	1	0	0	1	x	1	x	x
0	1	0	0	0	x	1	x	x
0	1	0	0	x	1	1	x	x
0	1	0	0	x	0	1	x	x
0	1	0	0	x	x	1	1	x
0	1	0	0	x	x	1	0	x
0	1	0	0	x	x	1	x	1
0	1	0	0	x	x	1	x	0

圖 4.1.1：在此$\beta = 4$，g=3，經過兩次讀取後

若產生如上九種結果，都會對應回 EC ＝

010　111　000。

如上圖 4.1.1 所示，假設 X-word 含有$\beta = 4$ 個 x-bit，且 g=3，只要能順利偵測 3 個以上的 x-bit，則認為此 X-word 正確，故經由兩次讀取程序所認定正確的訊號都能恢復成 010　111　000 的 EC，再經過 EC-decoder 返回原始 unit-word　101。

若 X-word 在兩次讀取程序中無法讀出 g 個以上的 x-bit，我們就認為此 X-word 不正確。例如 X-word 在兩次讀取程序中判定為 010　11x　01x，只偵測出兩個 x-bit。我們讓兩次讀取判定的 X-word，第一個 unit-word 中的第一個正規位元變為其補數，故第一個位元由 0 變成 1，重新給定的結果為 110　11x　01x，如果此 X-word 在對照表中無法發現，則在 x-bit decoder 中，對裡面的 x-bit，指定成第一次讀取結果，則經 x-bit　decoder 成 110　110　010，再經過 EC-decoder(由(3.1.5)式)，可以返回成 unit-word　010。按照這樣的程序編碼與解碼，我們可以保證若無法偵測出 g 個以上的 x-bit 之 X-word，經解碼後的 unit-word　010，其第一個位元 0，與原始 unit-word　101 的第一個位元 1，一定不會相同，故解碼後的 unit-word 必定不正確。

將原始 unit-word 編碼為 EC，再以上述的程序加入 x-bit，這種編碼程序我們稱為一對多編碼(one to many encoder)，而相對應的解碼程序稱為一對多解碼(one to many decoder) 。由圖 4.1.1 得知，X-word 有可能讀出九種不同的結果，但都對應回 unit-word 101，故命名為一對多編碼。對一個 X-word 而言，β 個 x-bit 能正確偵測 g 個以上 x-bit 的機率以 Pd_g 表示，故

$$Pd_g = [\sum_{i=0}^{\beta-g} C_i^{\beta} P_{e,X}{}^{i} (1-P_{e,X})^{\beta-i}] \cdot (1-P_{e,n})^{jv-\beta} \qquad (4.1.1)$$

在(4.1.1)式中 j 為 EC 中 unit-word 的個數，而一個 unit-word 由 v 個位元組成，$P_{e,n}$ 是正規位元錯誤的機率。

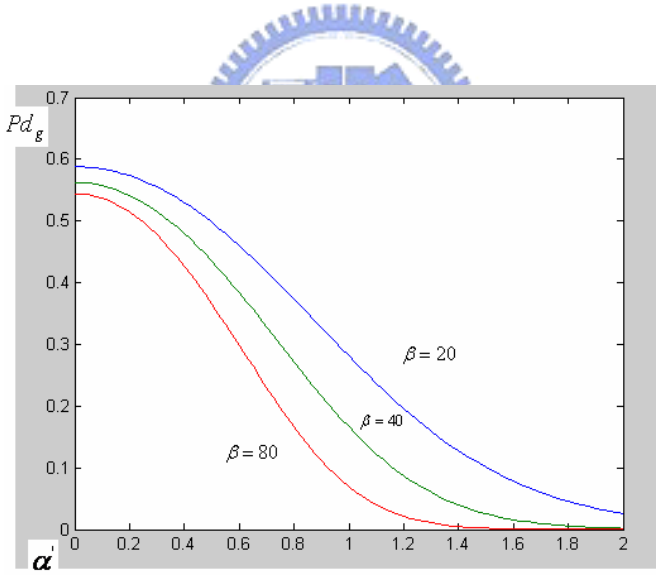

圖 4.1.2 :圖中曲線由上而下分別為
β=20，g=10， j=5， v=6
β=40，g=20， j=9， v=6
β=80，g=40， j=17， v=6
且水平座標為 α' ，垂直座標為 Pd_g ，g= $\frac{\beta}{2}$ 。

由圖 4.1.2 得知，隨著 β 增加，Pd_g 對 α' 圖形的斜率越大。隨著 α' 增加，Pd_g 越小，故在不同 α' 下，X-word 正確的機率，差距越加擴大。

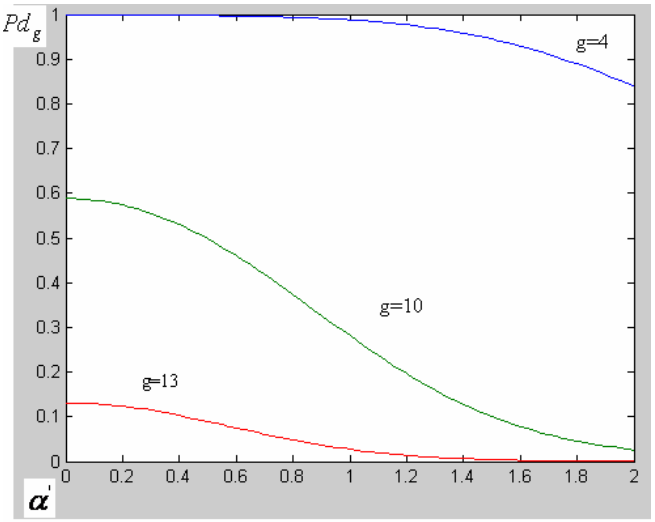

圖 4.1.3 :圖中曲線由上而下分別為
β=20，g=4， j=5， v=6
β=20，g=10， j=5， v=6
β=20，g=13， j=5， v=6

且水平座標為 α'，垂直座標為 Pd_g。

由圖 4.1.3 可知，我們可以藉由控制 X-word 中正確偵測 x-bit 的個數 g 來得到我們想要的 X-word 之機率。若我們希望 X-word 正確偵測機率高，就讓 g 小，希望 X-word 正確偵測機率低，就讓 g 大，故 X-word 正確偵測機率變得具有選擇性。

4.2 變形的系統I之編碼

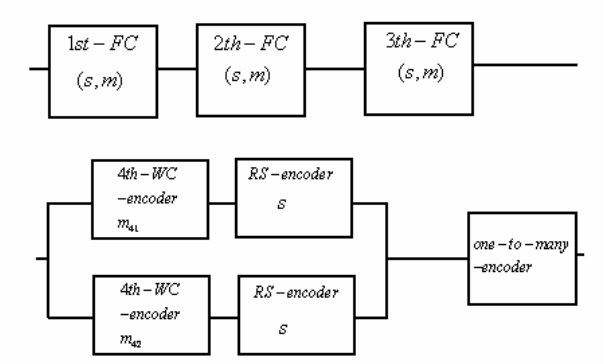

圖 4.2.1： 變形的系統I之編碼流程圖

基於降低$\alpha_c^{'}$的目標，我們提出變形系統I。我們的方法是先將原始鑰匙中，任意一個長度為v個位元的 unit-word 以第一階 FC 編碼，產生一個長度為s個 unit-word 的第一階 FC。再將此第一階 FC 依序用第二階、第三階的 FC 編碼，共生成s^2個第三階 FC，且每個第三階 FC 由s個 unit-word 組成。

現在考慮任意一個第三階 FC，此 FC 由s個 unit-word 組成。我們先從s個 unit-word 挑出前f_1個 unit-word，再編成第四階的 EC，故生成f_1個第四階 EC，此f_1個 EC 長度都由m_{41}個 unit-word 組成。

接著，再從剩下$s-f_1$個 unit-word，挑出前f_2個 unit-word，再編成第四階 EC，故生成f_2個第四階 EC，此f_2個 EC 長度都是由m_{42}個 unit-word 組成。最後剩下$s-f_1-f_2$個 unit-word，我們不對它們做任何處理，上述的過程顯示在圖 4.2.1 中。

第二步，將 f_1 個長度 m_{41} 個 unit-word 的 EC 作 RS-coding，生成 f_1 個長度 s 個 unit-word 的第四階 FC，其更錯能力為 $\frac{s-m_{41}}{2}$ 個 unit-word。同理，再將 f_2 個長度 m_{42} 個 unit-word 的 EC 作 RS-coding，生成 f_2 個長度 s 個 unit-word 的第四階 FC，其更錯能力為 $\frac{s-m_{42}}{2}$ 個 unit-word。

由此一個第三階 FC 產生的第四階 FC 共有 f_1+f_2 個，具有兩種更錯能力，分別為 $\frac{s-m_{41}}{2}$ 與 $\frac{s-m_{42}}{2}$。考慮任意一個更錯能力為 $\frac{s-m_{41}}{2}$ 的第四階 FC，將此 FC 挑出前 u_1 個 unit-word，以一對多的編碼，生成的 X-word 中共有 β_1 個 x-bit，而每個 X-word 由 j_1 個 unit-word 組成。同理，讓更錯能力 $\frac{s-m_{42}}{2}$ 的任意一個第四階 FC，挑出前 u_2 個 unit-word，以一對多的編碼，生成的 X-word 中共有 β_2 個 x-bit，而每個 X-word 由 j_2 個 unit-word 組成。

因此，任意一個第三階 FC 產生 X-word 的個數共有 $f_1u_1+f_2u_2$ 個，X-word 正確的機率也有兩種，而先前的系統，由一個第三階 FC 產生 X-word 的個數共有 $s\cdot u$ 個。更進一步，我們多加入調整 X-word 個數的機制，不再是之前的 $s\cdot u$ 個，而是 $f_1u_1+f_2u_2$，藉由調整 f_1， f_2，u_1，u_2，得到我們需要的 $\alpha_c^{'}$。

4.3 變形的系統I之解碼

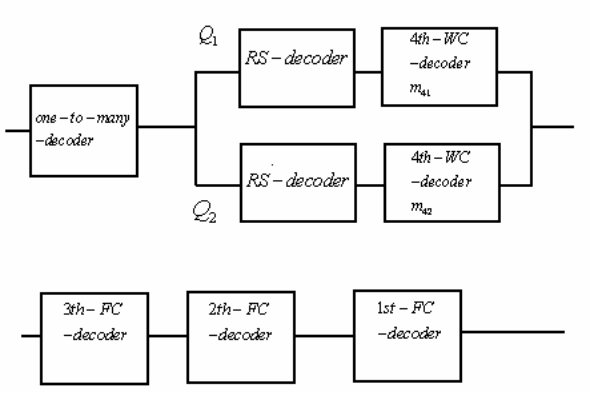

圖 4.3.1： 變形的系統I解碼流程圖

在這裡我們考慮播放端的動作分析，讓含有 β_1 個 x-bit 的 X-word，又長度為 $j_1 v$ 個位元。假設 Pd_{g1} 表示 X-word 中，能正確偵測出 g_1 以上個 x-bit 的機率，則

$$Pd_{g_1} = [\sum_{i=0}^{\beta_1 - g_1} C_i^{\beta_1} {P_{e,X}}^i (1-P_{e,X})^{\beta_1 - i}] \cdot (1-P_{e,n})^{j_1 v - \beta_1} \tag{4.3.1}$$

考慮含有 β_2 個 x-bit 的 X-word，其長度為 $j_2 v$ 個位元。若 Pd_{g2} 表示 X-word 中，能正確偵測出 g_2 以上個 x-bit 的機率，則

$$Pd_{g2} = [\sum_{i=0}^{\beta_2 - g_2} C_i^{\beta_2} {P_{e,X}}^i (1-P_{e,X})^{\beta_2 - i}] \cdot (1-P_{e,n})^{j_2 v - \beta_2} \tag{4.3.2}$$

在這裡 Q_1 為經一對多解碼後，所解調出的第四階 FC，其更錯能力為 $\frac{s-m_{41}}{2}$ ，且 Q_1 由 s 個 unit-word 組成。令 $P_{41}(r)$ 代表 Q_1 中剛好有 r 個 unit-word 發生錯誤的機率，則

$$P_{41}(r)=\sum_{i=0}^{r} C_i^{s-u_1}(1-P_n)^i P_n^{(s-u_1)-i} C_{r-i}^{u_1}(1-Pd_{g_1})^{r-i} Pd_{g_1}{}^{u_1-(r-i)} \qquad (4.3.3)$$

由(4.3.3)可知在編碼過程，Q_1 中前 u_1 個 unit-word 經過一對多編碼，又剩下的 $(s-u_1)$ 個 unit-word 為正規字，故經一對多解碼後，假設有 i 個錯發生在 $(s-u_1)$ 個正規字中， $r-i$ 個錯發生在 u_1 個 unit-word 中，正規字正確的機率以 P_n 示之，unit-word 正確的機率等於 Pd_{g_1} 。

同理 Q_2 為經一對多解碼後所解調出的第四階 FC，其更錯能力為 $\frac{s-m_{42}}{2}$ ，Q_2 由 s 個 unit-word 組成。若 $P_{42}(r)$ 代表 Q_2 中剛好有 r 個 unit-word 發生錯誤的機率，則

$$P_{42}(r)=\sum_{i=0}^{r} C_i^{s-u_2}(1-P_n)^i P_n^{(s-u_2)-i} C_{r-i}^{u_2}(1-Pd_{g_2})^{r-i} Pd_{g_2}{}^{u_2-(r-i)} \qquad (4.3.4)$$

當 Q_1 中錯誤 unit-word 個數小於等於第四階 FC 更錯能力時，其對應的第四階 EC 會完全正確。若 P_{4f_1} 代表第四階 EC 中，任意一個 unit-word 正確之機率，則

$$P_{4f_1}=\sum_{r=0}^{\frac{s-m_{41}}{2}} P_{41}(r) \qquad (4.3.5)$$

同理，若 P_{4f_2} 代表第四階 EC 中任意一個 unit-word 正確之機率，則

$$P_{4f_2}=\sum_{r=0}^{\frac{s-m_{42}}{2}} P_{42}(r) \qquad (4.3.6)$$

假設ψ為任意一個第三階 FC，其更錯能力為$\frac{s-m}{2}$，ψ剛好發生r個 unit-word 錯的機率以$P_3(r)$表示，則

$$P_3(r)=\sum_{i_1=0}^{r}\sum_{i_2=0}^{r-i_1} C_{i_1}^{s-(f_1+f_2)}(1-P_n)^{i_1}P_n^{\,s-(f_1+f_2)-i_1}C_{i_2}^{f_2}(1-P_{4f_2})^{i_2}P_{4f_2}^{\,f_2-i_2}$$
$$C_{r-(i_1+i_2)}^{f_1}(1-P_{4f_1})^{r-(i_1+i_2)}P_{4f_1}^{\,f_1-[r-(i_1+i_2)]} \qquad (4.3.7)$$

由(4.3.7)可知在編碼過程，ψ由s個 unit-word 組成，ψ中前f_1個 unit-word 編成第四階 EC，且每個 EC 由m_{41}個 unit-word 組成，再從剩下的 unit-word 挑出f_2個 unit-word，編成第四階 EC，且每個 EC 由m_{42}個 unit-word 組成，最後所剩的皆為正規字。在此假設經解調後，ψ中的前f_1個 unit-word 發生$r-(i_1+i_2)$個錯，而後f_2個 unit-word 發生i_2個錯，最後$s-(f_1+f_2)$個正規字中發生i_1個錯，而前f_1個 unit-word 中，任意一個 unit-word 正確的機率等於P_{4f_1}，而後f_2個 unit-word 中，任意一個 unit-word 正確的機率等於P_{4f_2}。

當ψ中錯誤個數r，小於等於其更錯能力時，第三階 EC 會完全正確。若P_3代表第三階 EC 中任意一個 unit-word 正確之機率，則

$$P_3=\sum_{r=0}^{\frac{s-m}{2}}P_3(r) \qquad (4.3.8)$$

令φ為任意一個第二階 FC，它由s個 unit-word 組成。假設$P_2(r)$代表φ中剛好有r個 unit-word 錯誤的機率，若φ中錯誤個數r小於等於其更錯能力$\frac{s-m}{2}$時，則第二階 EC 完全會正確。以P_2表示第二階 EC 中任意一個 unit-word 正確之機率，則

$$P_2=\sum_{r=0}^{\frac{s-m}{2}}P_2(r)=\sum_{r=0}^{\frac{s-m}{2}}C_r^s(1-P_3)^rP_3^{\,s-r} \qquad (4.3.9)$$

依此類推，若 P_1 表示第一階 EC 中任意一個 unit-word 正確之機率，則

$$P_1 = \sum_{r=0}^{\frac{s-m}{2}} P_1(r) = \sum_{r=0}^{\frac{s-m}{2}} C_r^s (1-P_2)^r P_2^{s-r} \qquad (4.3.10)$$

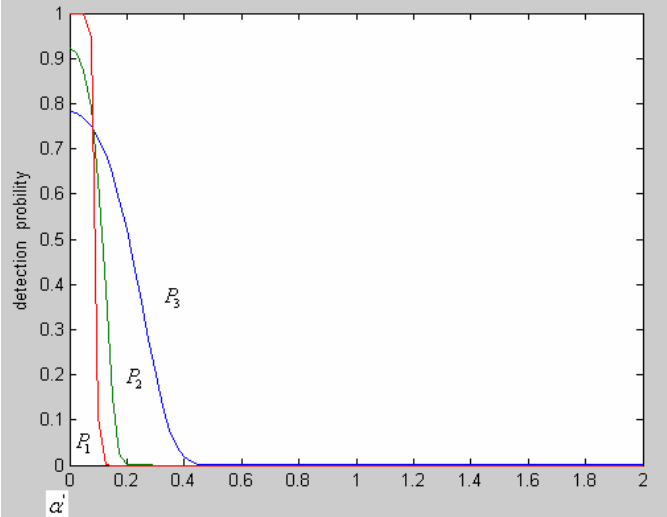

圖 4.3.2：$v=6$，$(s=63, m=27)$，$j_1=j_2=4$，$\beta_1=\beta_2=12$，$g_1=4$，$g_2=6$，$m_{41}=59$，$m_{42}=33$，$u_1=21$，$u_2=37$，$f_1=15$，$f_2=38$，共有四階 EC，圖中只顯示三階，水平座標為 α'，垂直座標為各階 EC 偵測正確的機率。

由圖 4.3.2 可知利用上述改良系統所得到的 α_c' 約為 0.1，與先前系統的 α_c'=0.325 下降很多，變形系統 I 的主要概念為藉由產生兩種偵測 X-word 的機率，且這兩種偵測 X-word 正確的機率對 α' 之斜率，比先前所提出系統來的大。

我們設計兩種 X-word，其中一種 X-word 正確偵測機率高，因為 g 小，而另一種 X-word 正確偵測的機率小，因為 g 大。這如同玩二十一點，當玩家手中的牌很大時，等效於加入很多偵測正確機率小的 X-word，這時再要牌，玩家一定希望拿到點數小，且加總後不超過二十一點的牌，此時可以藉由調整 f_1、f_2、u_1、u_2，來增加正確偵測機率高的 X-word。換言之，正確偵測機率小的 X-word 如同大牌，正確偵測機率高的 X-word 類似小牌，而製造精細的 X-word 點數一定比製造粗糙的 X-word 來的小，故更可以逼近系統的臨界錯誤，使 α_c' 變小。

若第一階 EC 完全正確，則原始鑰匙中任意一 unit-word 也會正確。令 P_{key} 代表鑰匙正確解調的機率，在圖 4.3.3 中考慮鑰匙由五個 unit-word 組成，且每個 unit-word 有 6 個位元，則 $P_{key} = P_1^5$。圖 4.3.3 為 P_{key} 對 α' 的變化，圖中 $\alpha_c' \approx 0.1$，比圖 3.5.5 下降許多。

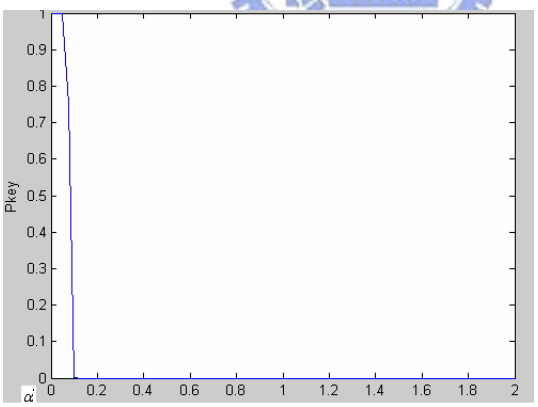

圖 4.3.3 :水平座標為 α'，垂直座標為鑰匙偵測正確機率， α_c' 約 0.1。

變形系統I達成讓$\alpha_c^{'}$下降的目標，但我們所付出的代價是加密鑰匙容量相當龐大。若*Co*代表加密鑰匙的容量，又原始鑰匙由*N*個unit-word組成，*Co*的單位為bit，則

$$Co = N[s^2(s-f_1-f_2)v + s^2 f_1(s-u_1)v + s^2 f_2(s-u_2)v + s^2 f_1 u_1 j_1 v + s^2 f_2 u_2 j_2 v] \quad (4.3.11)$$

在(4.3.11)中，等號右邊的第一項表示剩餘第三階FC位元數，第二項表示更錯能力為$\frac{s-m_{41}}{2}$的第四階FC位元數，第三項表示更錯能力為$\frac{s-m_{42}}{2}$之第四階FC位元數，而最後兩項分別表示兩種不同機率X-word位元數。

考慮圖4.3.2中的例子，N=5，v=6，$j_1=j_2=4$，$f_1=15$，$f_2=38$，$u_1=21$，$u_2=37$，代入(4.3.11)式中，可以得到$Co=1.0135\times10^9$ bit=1013.5 Mbits，如此龐大的容量，遠超出光碟所能負荷，是這個設計的主要缺點。接下來我們的目標不但希望$\alpha_c^{'}$下降，而且加密鑰匙的容量也要大幅降低。

4.4 變形系統II之編碼

變形的系統I雖然讓$\alpha_c^{'}$下降，但是所付出的代價是加密鑰匙容量太大，而主要原因是連續使用四階的FC編碼。我們希望降低FC的階數，但階數減少EC正確的機率未必能爬升至1，且圖型較不像方波。由圖4.3.2可以觀察P_2在$\alpha^{'}<\alpha_c^{'}$時未能攀爬至1，故無法以P_2代替P_1解調出加密鑰匙。

在此我們考慮用重複碼(Repetition Code)，簡稱*REC*，代替FC中的EC，而重複碼也是錯誤更正碼的一種，本身具有更錯能力。考慮由n個位元組成的

REC，又 n 為奇數，則可以更正 $(\frac{n-1}{2})$ 個位元，當錯誤的位元個數超過重複碼的更錯能力時，錯誤更正後的重複碼保留了錯誤，所以還是具有錯誤散佈的特性，故可以代替沒有更錯能力的 EC。

若一個由長度 v 個位元的 unit-word 編為重複碼，而產生的重複碼就是將 unit-word 重複 n 次，故此重複碼由 n 個 unit-word 組成，又重複碼中的每個 unit-word 也都是 v 個位元。接下來，再將此重複碼以(s，n) RS-code 編碼，產生長度 s 個 unit-word 的 code word，我們稱此種程序為前向錯誤更正重複碼 (forward error-correcting repetition code)，簡稱 FREC，而任意一個 FREC，其更錯能力為 $\frac{s-n}{2}$ 。

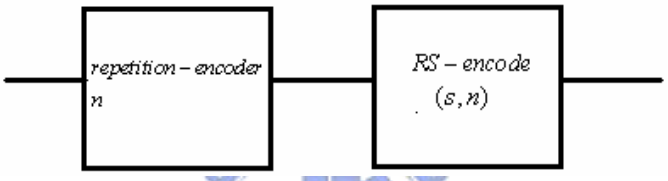

圖 4.4.1：上圖顯示一次 FREC 的過程。

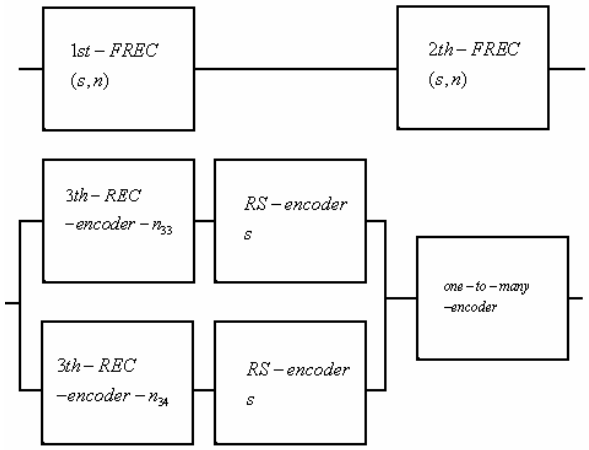

圖 4.4.2： 變形的系統II之編碼流程圖。

由圖 4.4.2 所示，先將原始鑰匙中任意一個 unit-word 以第一階 FREC 編碼，產生一個長度為 s 個 unit-word 的碼字。接著再做第二階 FREC 編碼，共生成 s 個第二階 FREC，且每個第二階的 FREC 由 s 個 unit-word 組成。第一階 FREC 與二階 FREC 的更錯能力皆為 $\frac{s-n}{2}$ 。

現在考慮任意一個第二階 FREC。此 FREC 由 s 個 unit-word 組成，我們先從 s 個 unit-word 挑出前 f_3 個 unit-word，再編成第三階的 REC，故生成 f_3 個第三階 REC，此 f_3 個 REC 長度都是由 n_{33} 個 unit-word 組成。

接著，再從剩下 $s-f_3$ 個 unit-word，挑出前 f_4 個 unit-word，再編成第三階的 REC，故生成 f_4 個第三階 REC，且此 f_4 個 REC 長度都是由 n_{34} 個 unit-word 組成。最後剩下 $s-f_3-f_4$ 個 unit-word，我們不對它們做任何處理。

第二步，將 f_3 個長度 n_{33} 個 unit-word 的 REC 作 RS-coding，生成 f_3 個長度 s 個 unit-word 的碼字，其更錯能力為 $\frac{s-n_{33}}{2}$ 個 unit-word。同理，再將 f_4 個長度 n_{34} 個 unit-word 的 REC 作 RS-coding，生成 f_4 個長度 s 個 unit-word 的碼字，其更錯能力為 $\frac{s-n_{34}}{2}$ 個 unit-word。故一個第二階 FREC 產生的第三階 FREC 共有 f_3+f_4 個分別具有兩種更錯能力的碼字。

考慮更錯能力為 $\frac{s-n_{33}}{2}$ 的第三階 FREC，將此 FREC 挑出前 u_3 個 unit-word，以一對多的編碼，生成的 X-word 中共有 β_3 個 x-bit，而每個 X-word 由 j_3 個 unit-word 組成。同理，令更錯能力 $\frac{s-n_{34}}{2}$ 的任意一個第三階 FREC，挑出前 u_4 個

unit-word，以一對多的編碼，生成的 X-word 中共有 β_4 個 x-bit， 而每個 X-word 由 j_4 個 unit-word 組成。

4.5 變形系統II之解碼

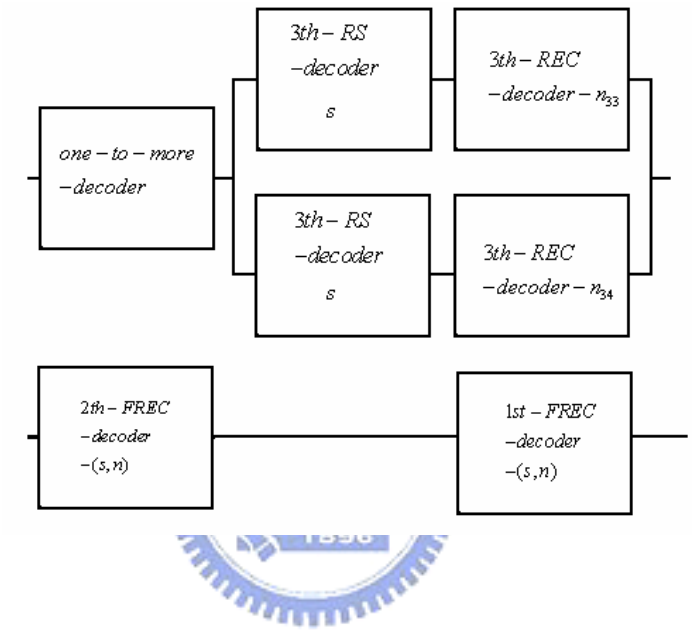

圖 4.5.1： 變形的系統II之解碼流程圖。

圖 4.5.1 是變形的系統II之解碼流程圖。為了得到第三階的 FREC，我們先將含有 β_3 個 x-bit 的 X-word，與含有 β_4 個 x-bit 的 X-word 分別通過一對多的解碼，如果能順利偵測出 g_3 、 g_4 以上個 x-bit，則 X-word 判定為正確，又兩種 X-word 正確偵測的機率分別以 Pd_{g3} 、 Pd_{g4} 表示。

在這裡 Q_3 為經一對多解碼後，所解調出的第三階 FREC，其更錯能力為 $\frac{s-n_{33}}{2}$ ，且 Q_3 由 s 個 unit-word 組成。令 $P_{33}(r)$ 代表 Q_3 中剛好有 r 個 unit-word 發生錯誤的機率，則

$$P_{33}(r)=\sum_{i=0}^{r} C_i^{s-u_3}(1-P_n)^i P_n^{(s-u_3)-i} C_{r-i}^{u_3}(1-Pd_{g_3})^{r-i} Pd_{g_3}{}^{u_3-(r-i)} \qquad (4.5.1)$$

由(4.5.1)可知在編碼過程，Q_3 中前 u_3 個 unit-word 經過一對多編碼，剩下的 $(s-u_3)$ 個 unit-word 為正規字，故經一對多解碼後，假設有 i 個錯發生在 $(s-u_3)$ 個正規字中， $r-i$ 個錯發生在 u_3 個 unit-word 中，正規字正確的機率以 P_n 示之，而其餘 unit-word 正確的機率等於 Pd_{g3}。

同理，Q_4 為經一對多解碼後，所解調出的第三階 FREC，其更錯能力為 $\frac{s-n_{34}}{2}$，且 Q_4 由 s 個 unit-word 組成。若 $P_{34}(r)$ 代表 Q_4 中剛好有 r 個 unit-word 發生錯誤的機率，正規字正確的機率以 P_n 示之，而其餘 unit-word 正確的機率等於 Pd_{g4}，則

$$P_{34}(r)=\sum_{i=0}^{r} C_i^{s-u_4}(1-P_n)^i P_n^{(s-u_4)-i} C_{r-i}^{u_4}(1-Pd_{g_4})^{r-i} Pd_{g_4}{}^{u_4-(r-i)} \qquad (4.5.2)$$

當 Q_3 中錯誤 unit-word 的個數，小於等於第三階 FREC 更錯能力 $\frac{s-n_{33}}{2}$ 時，第三階的 REC 會完全正確。若 $P_{n33,b}$ 代表第三階的 REC 中任意一個位元正確之機率，且此第三階的 REC 由 n_{33} 個 unit-word 組成，則

$$P_{n33,b}=\sum_{r=0}^{\frac{s-n_{33}}{2}} P_{33}(r) \qquad (4.5.3)$$

同理，當 Q_4 中錯誤 unit-word 的個數，小於等於第三階 FREC 更錯能力 $\frac{s-n_{34}}{2}$ 時，第三階的 REC 會完全正確。假設 $P_{n34,b}$ 代表第三階的 REC 中任意一個位元正確之機率，則

$$P_{n34,b}=\sum_{r=0}^{\frac{s-n_{34}}{2}} P_{34}(r) \qquad (4.5.4)$$

經由第三階的 RS-decoder 後，我們可以得到兩種第三階的 REC，長度分別為 $v \cdot n_{33}$ 個位元與 $v \cdot n_{34}$ 個位元。假設 Q_5 代表任意一個第三階 REC，且 Q_5 的長度為 $v \cdot n_{33}$ 個位元。我們知道此重複碼 Q_5 是將 unit-word 重複 n_{33} 次。換言之，unit-word 中每一個位元都被重複 n_{33} 次，且 n_{33} 為奇數。令 r 代表 Q_5 中每 n_{33} 個位元經解調後錯誤的位元個數，如果 r 小於等於 $\frac{n_{33}-1}{2}$ 時，經第三階的 REC-decoder 後，第二階 FREC 中，前 f_3 個 unit-word 中的任意一個位元可以復原，且 Q_5 中任意一個位元正確的機率為 $P_{n33,b}$。若 Q_7 代表任意一個第二階 FREC，且 Q_7 由 s 個 unit-word 組成，又 $P_{23,b}$ 代表 Q_7 中前 f_3 個 unit-word 中任意一個位元正確的機率，則

$$P_{23,b} = \sum_{r=0}^{\frac{n_{33}-1}{2}} C_r^{n_{33}} (1-P_{n_{33,b}})^r P_{n_{33,b}}^{n_{33}-r} \qquad (4.5.5)$$

同理，若 Q_6 代表任意一個第三階 REC，且 Q_6 的長度為 $v \cdot n_{34}$ 個位元，且 Q_6 中任意一個位元正確的機率為 $P_{n34,b}$。令 $P_{24,b}$ 代表 Q_7 中，剩下的 unit-word 中($s-f_3$ 個)，前 f_4 個 unit-word 中任意一個位元正確的機率，則

$$P_{24,b} = \sum_{r=0}^{\frac{n_{34}-1}{2}} C_r^{n_{34}} (1-P_{n_{34,b}})^r P_{n_{34,b}}^{n_{34}-r} \qquad (4.5.6)$$

若 P_{23} 表示 Q_7 中，前 f_3 個 unit-word 中任意一個 unit-word 正確的機率，而 P_{24} 表示 Q_7 剩下的 unit-word 中(剩下 $s-f_3$ 個) ，前 f_4 個 unit-word 中任意一個 unit-word 正確的機率，而 Q_7 最後剩下 $s-f_3-f_4$ 個 unit-word 皆為正規字，則

$$P_{23} = P_{23,b}{}^{v} \tag{4.5.7}$$

$$P_{24} = P_{24,b}{}^{v} \tag{4.5.8}$$

因為Q_7代表任一個第二階 FREC，若r代表Q_7中錯誤 unit-word 的個數，如果r小於等於$\frac{s-n}{2}$，則第二階的 REC 會完全正確。令$P_2(r)$代表Q_7中，剛好有r個 unit-word 發生錯誤之機率，(4.5.9)式中假設Q_7中前f_3個 unit-word 中發生$r-(i_1+i_2)$個錯，剩下的f_4個 unit-word 發生i_2個錯，而在$s-f_3-f_4$個正規字中產生i_1個錯，因此，若$P_{n2,b}$代表第二階的 REC 中，任意一個位元正確之機率，則

$$P_2(r) = \sum_{i_1=0}^{r}\sum_{i_2=0}^{r-i_1} C_{i_1}^{s-(f_3+f_4)}(1-P_n)^{i_1} P_n{}^{s-(f_3+f_4)-i_1} C_{i_2}^{f_4}(1-P_{24})^{i_2} P_{24}{}^{f_4-i_2}$$
$$C_{r-(i_1+i_2)}^{f_3}(1-P_{23})^{r-(i_1+i_2)} P_{23}{}^{f_3-[r-(i_1+i_2)]} \tag{4.5.9}$$

$$P_{n2,b} = \sum_{r=0}^{\frac{s-n}{2}} P_2(r) \tag{4.5.10}$$

依循上述邏輯，若$P_{1,b}$代表第一階 FREC 中，任意一個位元正確的機率，則

$$P_{1,b} = \sum_{r=0}^{\frac{n-1}{2}} C_r^n (1-P_{n2,b})^r P_{n2,b}{}^{n-r} \tag{4.5.11}$$

若P_1代表第一階 FREC 中，任一個 unit-word 正確的機率，則

$$P_1 = P_{1,b}{}^{v} \tag{4.5.12}$$

若 $P_1(r)$ 代表第一階 FREC 中剛好有 r 個 unit-word 錯誤的機率，如果 r 小於等於 $\frac{s-n}{2}$ 時，第一階 REC 會完全正確。換言之，第一階 REC 中的每個位元會同時正確。令 $P_{n1,b}$ 代表第一階 REC 中，任一個位元正確的機率，則

$$P_{n1,b} = \sum_{r=0}^{\frac{s-n}{2}} P_1(r) = \sum_{r=0}^{\frac{s-n}{2}} C_r^n \, (1-P_1)^r \, P_1^{n-r} \tag{4.5.13}$$

最後令 $P_{key,b}$ 代表原始鑰匙中任意一個位元正確的機率，則

$$P_{key,b} = \sum_{r=0}^{\frac{n-1}{2}} C_r^n \, (1-P_{n1,b})^r \, P_{n1,b}^{n-r} \tag{4.5.14}$$

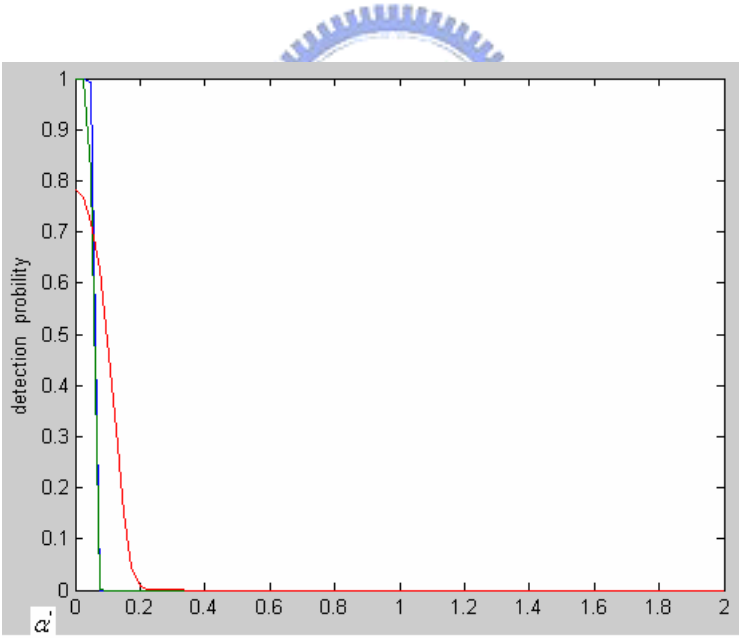

圖 4.5.2：圖中曲線由上而下分別為 $P_{key,b}$ ， $P_{n1,b}$ ， $P_{n2,b}$ ，水平座標為 α' ，$(s=31, n=11)$ ，$v=5$ ，$n_{33}=19$ ，$n_{34}=27$ ，$\beta_3=50$ ，$g_3=24$ ，$j_3=14$ ，$\beta_4=6$ ，$g_4=2$ ，$j_4=3$ ，$f_3=14$ ，$f_4=11$ ，$u_3=18$ ，$u_4=11$ 。

在圖 4.5.2 中，$P_{n2,b}$ 已經很陡峭，而 $P_{n1,b}$ 與 $P_{key,b}$ 快要重疊。在 $\alpha' < \alpha_c'$ 的區域中，所對應的 $P_{key,b}$ 接近 1，而在 $\alpha' > \alpha_c'$ 的區域中，所對應的 $P_{key,b}$ 接近 0，而 $P_{key,b}$ 對 α' 的圖型如同一方波。

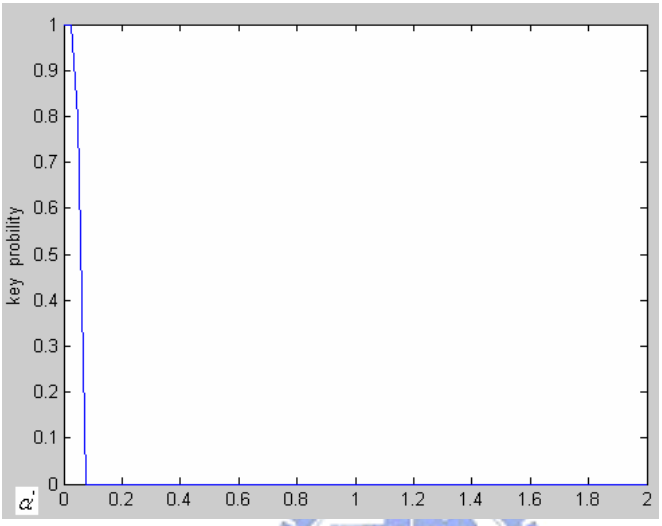

圖 4.5.3：水平座標為 α'，垂直座標為解碼後鑰匙正確的機率，α_c' 約為 0.075。

因為原始鑰匙由三十個位元組成，若 P_{key} 代表鑰匙正確的機率，則 $P_{key} = P_{key,b}^{30}$。

圖 4.5.3 顯示出 α_c' 約為 0.075，若 x-bit 的反射率 A_X 沒有落在 $[\mu - \alpha_c'\sigma_n, \mu + \alpha_c'\sigma_n]$ 區間內，則無法恢復正確的鑰匙。所以 $x-bit$ 必然要用很高的成本製造，一般的盜版商根本無法負荷，故可以有效的預防盜拷。

若 Bo 代表加密鑰匙的位元數，又鑰匙由 N 個 unit-word 組成，且個 unit-word 由 v 個位元組成，Bo 的單位為 bit，則

$$Bo = N[s(s - f_3 - f_4)v + sf_3(s - u_3)v + sf_4(s - u_4)v + sf_3u_3j_3v + sf_4u_4j_4v] \quad (4.5.15)$$

在(4.5.15)中，等號右邊第一項表示剩餘第二階 FREC 位元數，第二項表示剩餘第三階 FREC 位元數，其更錯能力為 $\frac{s-n_{33}}{2}$，第三項表示剩餘第三階 FREC 位元數，其更錯能力為 $\frac{s-n_{34}}{2}$，而最後兩項分別表示兩種不同機率 X-word 位元數。

考慮圖 4.5.2 中的例子，U =6，v=5，$s=31$，$j_3=14$，$j_4=3$，$f_3=18$，$f_4=11$，$u_3=18$，$u_4=11$，代入(4.5.15)式中，可以得到 Bo =4.98Mbit。結果顯示我們不只將 $\alpha_c^{'}$ 變小，更將加密鑰匙的位元數降為約 5 Mbits，而變形系統 II 成功解決加密鑰匙位元數過於龐大的問題，主要因素以重複碼代替 FC 中的 EC。單就錯誤更正能力討論，變形系統 II 勝於變形系統 I，可以有效減少 FC 階數，降低加密鑰匙的位元數。

第五章 結論

一般光碟中的訊號有兩種，一為二位元 0，另一為二位元 1，在我們提出的方法中，加入第三種訊號 x-bit 至光碟。我們使用兩次讀取程序偵測 x-bit，如果兩次讀出的結果不一樣就認定是 x-bit，由於正確偵測 x-bit 的機率與 x-bit 準位的精確度密切相關，我們可以利用這個特性防止盜拷光碟。在我們所提出的方法中，如果 x-bit 的訊號準位製造的很精準，落在某一個很窄小的區間，則光碟可以正確播出，否則會播出不正確的內容。又因為拷貝 x-bit 需要極精密的機器，專業的盜版商根本無法負擔，所以可以有效防止盜拷。

在光碟的加密程序中，通常會設計一把原始鑰匙。接著，對原始鑰匙進行一連串的加密動作，變成加密鑰匙存放入光碟中。當光碟播放時，先將加密鑰匙回復成原始鑰匙，若無法獲得正確的原始鑰匙，就無法得到正確的資料，故可以用來防止盜拷光碟。我們使用 MWE 程序與 x-bit encoder 將原始鑰匙編作加密鑰匙。我們發現在解碼過程中各階 EC 正確機率會相交在同一點，並且產生一個近似方波的原始鑰匙解碼正確之機率。在 $\alpha' < \alpha_c'$ 區域中，原始鑰匙可正確回復，而在 $\alpha' > \alpha_c'$ 區域中，會產生錯誤的原始鑰匙。由於這個解碼特性，若拷貝的光碟無法精確製作 x-bit 使得 $\alpha' < \alpha_c'$，則所製作的光碟將無法成功播放，故能有效防止盜拷。

接著，我們希望提升拷貝光碟的困難度，所以提出變形系統I。變形系統I提出新方法去設計 X-word，使 X-word 正確偵測的機率對α'的斜率變大，故更能區分出光碟中 x-bit 製作的精確度，有效的反應出光碟的生產成本。我們發現在變形系統I 中，各階 *EC* 正確的機率仍然會交至同一點。變形系統I主要的想法是設計兩種不同偵測機率的 X-word，一種正確機率高，而另一種正確機率低。我們知道 X-word 不能無限制的安插在光碟中，一旦加入的個數太多，則原始鑰匙無法回復。當系統接近臨界錯誤時，我們增加偵測機率高 X-word 的數目，使其更能逼近系統的臨界錯誤，目的在使α_c'變小，以提升拷貝的困難度。

變形系統I有效的降低α_c'，但所面臨的問題是加密鑰匙位元數過於龐大，加密鑰匙位元數大約 1Gbits，無法應用於光碟中，其主要原因是連續使用四次 FC 作編碼。我們知道每使用一次 FC 作編碼，產生碼字個數是原本的 *s* 倍，我們希望降低 FC 的階數，但是 FC 階數減少 EC 正確的機率未必能爬升至 1，原始鑰匙正確之機率也較不像方波，我們希望能改善這個缺點，所以提出變形系統II。

變形系統II主要是引入重複碼(repetition code)。我們知道重複碼也是更錯碼的一種，當錯誤的位元數大於重複碼的更錯能力時，解調出來的碼仍然保有錯誤，故可以代替沒有更錯能力的 EC。由於重複碼的加入，我們可以降低 FC 使用的次數，在變形系統II只連續使用三次的 FREC 作編碼，故有效的降低 FC 使用的次數。變形系統II將加密鑰匙位元數降至約 5Mbits，成功的改善變形系統I的問題。

我們知道絕大部份的唱片公司使用防盜拷光碟片來防止盜拷，但並不能有效的阻止盜拷行為，而主要的原因是防盜拷光碟片的數量太多，不容易管控。在早期的地下工廠經過加工後就可以生產防盜拷光碟片，再流入市面，所以要取得並不困難。專業的盜版商只要使用防盜拷光碟片燒錄正版光碟就可以製造出一模一樣的光碟。在我們的方法中，我們主要是提高拷貝光碟的困難度，故拷貝光碟必需以極昂貴的機器來完成。對專業的盜版商而言，要去購買如此昂貴的燒錄器是不可能的，因為真正的有錢人不會想當盜版商，而盜版商不會願意花大錢生產好的產品。我們知道昂貴的燒錄器能夠管控的很好，因為只有大型的公司有能力購買，所以數量不會太多，故更能夠有效的管控避免流入非法人士的手中。

在結合硬體與軟體的防盜拷技術中，所使用的光碟機是一般的機種，並不需要特製的光碟機來讀取光碟。我們只需要上網下載新的偵測程序，因為執行兩次讀取程序對目前的光碟機並不困難，而讀加密鑰匙兩次並不會等待太多時間，所以現有的光碟機都適用於我們所提出的方法。以往所提出的防盜拷技術主要是隱藏光碟內容，並未站在硬體的角度考量。在所提的方法中，我們並未刻意隱藏原始鑰匙，著重在硬體拷貝的困難度，這讓盜拷光碟更為困難，盜版商必須花費非常高的成本才能拷貝光碟，故能有效的阻止盜拷。未來我們將繼續研究更簡單有效的方法以進一步提升拷貝的難度，以有效防制光碟盜拷。

參考文獻

[1] Bloom, J.A.; Cox, I.J.; Kalker, T.; Linnartz, J.-P.M.G.; Miller, M.L.; Traw, C.B.S., "**Copy protection for DVD video**," Proceedings of the IEEE, vol. 87, pp. 1267 – 1276, July 1999

[2] Maes, M.; Kalker, T.; Linnartz, J.-P.M.G.; Talstra, J.; Depovere, F.G.; Haitsma, J., "**Digital watermarking for DVD video copy protection**," IEEE Signal Processing Magazine, vol. 17, pp. 47–57, Sep. 2000.

[3] Morito, H.; Roe, M.; Stewart Lee, E., "**Digital copy protection scheme using recording medium identifier**," in Proc.. 1999 IEEE International . Workshops on Parallel Processing, pp. 174–178.

[4] Potlapally, N.R., "**Optical fingerprinting to protect data: a proposal**," Computer, vol. 35, pp.23-28, Mar. 2002

[5] N. R. Potlapally, "**Optical fingerprinting to protect data: a proposal**," Computer, vol. 35, pp.23-28, Mar. 2002

[6] Wallach, D.S., "**Copy protection technology is doomed,**" Computer, vol. 34, pp.48-49, Oct. 2001.

[7] Geier, M.J, "**Lights, camera, controls!** " Computer, vol. 40, pp.28-31, May. 2003.

[8] S. Simon, Communication Systems, John Wiley & Sons, 2001, ch. 4
S. Lin and D. J. Costello, Error Control Coding, Pearson Prentice Hall,

[9] S. Lin and D. J. Costello, Error Control Coding, Pearson Prentice Hall,2003, ch. 7.

[10] Chubing Peng and M.Mansuripur, "**Source of Noise in Phase-change Disk Data Storage** "

[11] Min –Goo Kim , Kwang Man Ok ,and Jae Hong Lee , "**Performance Comparsion of Detection Methods in Magneto-optical Disc System with (1,7) RLL Code** "

簡　　歷

姓　　名：錢大源

居 住 地：台灣省台北市

出生年月：民國六十八年八月二日

學 經 歷：

輔仁大學電子工程學系　　(95 年 9 月~ 95 年 6 月)

交通大學電信工程學系碩士班　　(95 年 9 月~ 97 年 8 月)