

結合隨機正負邏輯與雜步跳躍之聲訊浮水印方法

古鴻炎

台灣科大資工系副教授
guhy@mail.ntust.edu.tw

王識揚

台灣科大資工所碩士
M9515903@mail.ntust.edu.tw

摘要

本研究提出一個在實數倒頻譜(cepstrum)域上，結合正負邏輯亂數序列與雜步跳躍之聲訊(audio)浮水印的藏入方法，不僅可提高藏入的位元量，並且可杜絕他人在未經授權的情況下作浮水印(watermark)的取出或竄改。此外我們以藏入同步位元的作法，來偵測出浮水印的藏入時間點，因而增加了同步攻擊的抵抗能力。為了驗證所提出方法的可行性，我們實際建造了一個聲訊浮水印的藏入及解出系統，然後在進行幾種常見的攻擊之後，作浮水印的解出，攻擊的方式包括 MP3 壓縮和低通濾波。實驗結果顯示，本研究方法均可在不察覺出音質破壞的門檻值設定下，幾乎完整地將浮水印解出。

關鍵詞：聲訊浮水印、倒頻譜、正負邏輯。

1. 前言

隨著個人電腦的普及、網際網路與多媒體技術之蓬勃發展，大家得以廣泛的使用數位資料作為資訊保存與交流之媒介。但是我們也面臨了數位資料被非法盜版及惡意篡改等嚴峻的問題。因此，將數位資料加入浮水印可說是保護版權的一個可行方法。

本論文研究了聲訊浮水印的藏入架構與方法，並且實作出一個可供實際操作之系統。我們的目標是在聲音品質不被察覺變化的情況下，將浮水印藏入聲訊中，並且在幾種常見的攻擊下，仍然可把所藏入的浮水印擷取出來。

聲訊藏入浮水印之方法，可以粗略地分為時域法、及轉換域法兩類，時域法為將浮水印直接嵌入聲音的時域訊號中[1]。

至於轉換域法，則是先對時域信號作離散傅利葉轉換(discrete Fourier transform, DFT)[2]、離散餘弦(cosine)轉換[3]、離散小波(wavelet)轉換[4]、或倒頻譜轉換(cepstrum transform)[5]，以轉換至另一領域(domain)，然後加入浮水印，之後再逆轉成時域信號。一般來說，使用時域法所需計算量小，但是所藏入的浮水印很容易遭壓縮及濾波處理的破壞。使用轉換域法所需之計算量比較高，但所藏入的浮水印也比較強健而不易遭到破壞。此外，最近有一種稱為“零浮水印”的藏入方式被提出[6]，但是該方式需要花費不少的記憶空間來保存所計算出的key值，這種需要另外保存key值的作法，和本研究的理念(不要另外去保存資料)不合。

過去，已有一些學者研究以修改倒頻譜係數的方式來藏入浮水印[5, 7, 8, 9, 10]，本研究的方法雖然也是藉由修改倒頻譜係數來藏入浮水印，但是我們提出了一種新的浮水印位元之藏入機制，這個機制應用了亂數序列來決定何時使用正負邏輯之正邏輯或負邏輯。如果把亂數序列改成位元值全都是 1 的序列，那麼該藏入機制就退化成前人所提出的，因此前人的機制可說是我們機制的一個特例。

另外，我們也在藏入方法的架構上作了改進，就是採取雜步跳躍方式(類似 interleaving 的觀念)，將一個浮水印位元作藏入所使用的倒頻譜係數分散於多個連續的音框之中，以解決信號因局部遭破壞而無法解出浮水印之問題。此外，先前研究者修改倒頻譜係數來藏入浮水印時，一個音框只藏一個位元，可藏入之資料量較少，而當運用本研究提出的方法架構時，則可提高藏入的資料量。

2. 複數倒頻譜與實數倒頻譜

當只提到“倒頻譜”這個術語時，一般指的是複數倒頻譜。複數倒頻譜係數的轉換步驟如圖1所示，對於一個音框的聲音訊號 $x(n)$, $n=0, 1, \dots, N-1$ ，首先作離散傅利葉轉換(DFT)以得到頻域的係數 $X(\omega)$, $\omega=0, 1, \dots, N-1$, N 為DFT的點數；因為 $X(\omega)$ 是複數，所以接著要取複數的自然對數，也就是 $\ln(X(\omega)) = \ln(|X(\omega)|) + i \cdot \text{Arg}(X(\omega))$ ；然後對 $\ln(X(\omega))$ 作離散傅利葉逆轉換(inverse DFT)即可得到一組倒頻譜係數 $c(n)$, $n=0, 1, \dots, N-1$ 。

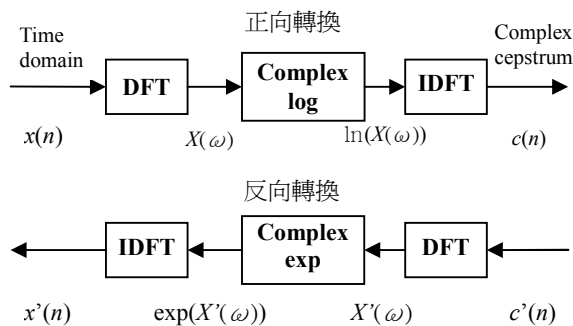


圖 1 複數倒頻譜轉換的步驟

當我們對一個音框的複數倒頻譜係數，作統計均值的調整(statistical mean manipulation)[7]，以藏入浮水印位元，之後把該音框拿去作解出處理，的確可以解出正確的浮水印位元；但是，如果在計算一個音框之複數倒頻譜係數的統計均值時，導入正負邏之輯觀念來藏入浮水印位元，然後把該音框拿去作解出處理，則發現不能夠解出正確的浮水印位元。因此，我們就改變去研究、使用實數倒頻譜係數[11]，以便解決前述的問題。

實數倒頻譜的轉換步驟如圖2所示，對於一個音框的聲音訊號 $x(n)$, $n=0, 1, \dots, N-1$ ，首先作離散傅利葉轉換(DFT)以得到頻域的係數 $X(\omega)$, $\omega=0, 1, \dots, N-1$, N 為DFT的點數；頻譜係數 $X(\omega)$ 為複數，可以分成絕對值(magnitude)及相位(phase)兩個部分，在此我們只將絕對值部分取自然對數得到 $\ln(|X(\omega)|)$ ，但是保留相位的部分 $\text{Arg}(X(\omega))$ 於暫存區，以供作反向轉換時使用；接著，以 $\ln(|X(\omega)|)$ 為實部，虛部設為0，

再進行離散傅利葉逆轉換(IDFT)，即可得到一組實數倒頻譜係數 $c(n)$, $n = 0, 1, \dots, N-1$ ，並且具有 $c(k) = c(N-k)$, $k = 1, 2, \dots, N/2 - 1$ ，之對稱特性。

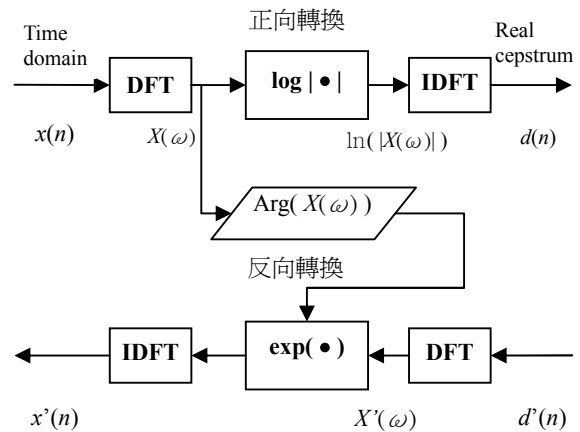


圖 2 實數倒頻譜轉換的步驟

3. 浮水印藏入與解出

我們系統的處理流程可分為浮水印藏入、和浮水印解出兩個階段，浮水印藏入階段的主要處理流程如圖3所示，圖中主要方塊的細節，將於下列子節中說明。

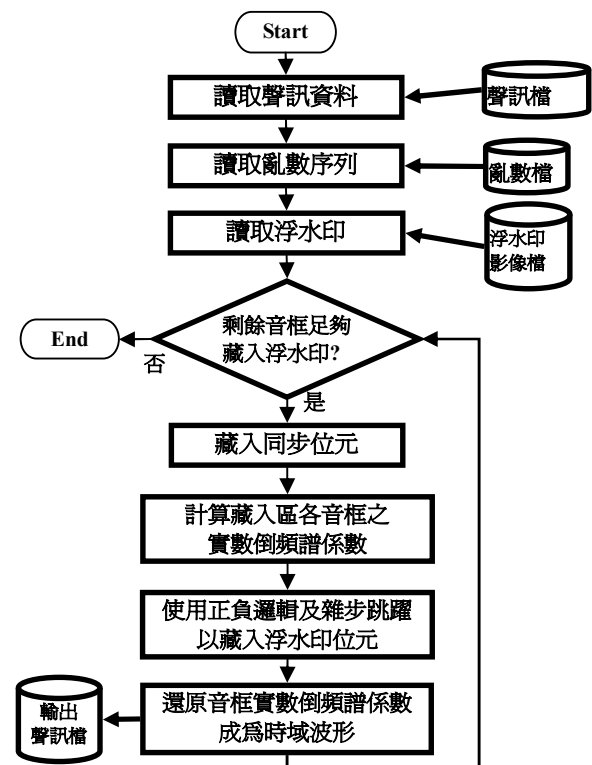


圖 3 浮水印藏入的主流程

3.1 藏入同步位元

在本研究裡，聲訊檔的取樣率是44,100Hz，音框的長度設為1,024個樣本點，並且相鄰的音框之間不重疊。對於同步位元，我們採取一個音框藏入一個位元的方式進行藏入，由於實數倒頻譜係數具有偶對稱的特性，因此要將一個同步位元藏於一個音框中時，我們只能考慮半數(即512個)的實數倒頻譜係數，此外要再考慮最前的幾個係數也是不能被使用(易造成訊號之破壞)。我們藏入同步位元的步驟詳述如下：

(a) 一個音框的訊號樣本，轉換至實數倒頻譜後，設得到的實數倒頻譜係數序列為： $d_0, d_1, d_2, \dots, d_{1023}$ 。

(b) 計算步驟(a)的係數經正負邏輯作用後之平均值，正負邏輯則使用一個值為1或-1的亂數序列 RS_j ，公式為：

$$SM = \frac{1}{482} \sum_{i=30}^{511} d_i \times RS_{i-30} \quad (1)$$

(c) 當藏入的同步位元為1時，調整 d_i 值，使得 SM 的值達到門檻參數 TH 的值。細部作法是，當 SM 值已大於等於門檻 TH 的值時，就不調整任何係數；反之則要調整係數 d_i 的值， $i=30, 31, \dots, 511$ ，首先計算出差額 δ 的值，即 $\delta = TH - SM$ ，再據以作調整，公式為：

$$\bar{d}_i = \begin{cases} d_i + \delta, & \text{if } RS_{i-30} > 0 \text{ (正邏輯)} \\ d_i - \delta, & \text{if } RS_{i-30} < 0 \text{ (負邏輯)} \end{cases} \quad (2)$$

(d) 當藏入的同步位元為0時，調整 d_i 值，使得 SM 達到負的門檻參數值(即 $-TH$)。細部作法是，當 SM 值已小於等於負的門檻值($-TH$)時，就不調整任何係數；反之則要調整係數 d_i 的值， $i=30, 31, \dots, 511$ ，首先計算出差額 δ 的值，即 $\delta = SM - (-TH)$ ，再據以作調整，公式為：

$$\bar{d}_i = \begin{cases} d_i - \delta, & \text{if } RS_{i-30} > 0 \text{ (正邏輯)} \\ d_i + \delta, & \text{if } RS_{i-30} < 0 \text{ (負邏輯)} \end{cases} \quad (3)$$

(e) 為使實數倒頻譜係數維持偶對稱的特性，修改另一半的係數值，即令 $\bar{d}_{1024-i} = \bar{d}_i$ ， $i=30, 31, \dots, 511$ 。

(f) 依據修改過的實數倒頻譜係數序列，作逆向實數倒頻譜轉換，以還原成時域訊號。

(g) 重複步驟(a)至(f)八次，以藏入連續的八個同步位元"10101010"。

上列作同步位元藏入的步驟裡，我們已經藉由亂數序列 RS_i 來導入正負邏輯的觀念。另外，當要解出一個音框被藏入的位元時，前述步驟只需做到步驟(b)，就可依據 SM 的正負情況來決定藏入的位元值，若 SM 為正，則代表解出位元1，否則就代表解出位元0。

3.2 隨機正負邏輯之優點

過去Xin Li等人提出，以修改一個音框的複數倒頻譜係數的平均值[7]，來藏入一個浮水印位元，這樣的藏入方法可說是，對各個複數倒頻譜係數都採取正邏輯的思考方式。然而從公式(1)、(2)和(3)可知，我們是依據一個數值為1或-1的亂數序列 RS_j ，來決定各個實數倒頻譜係數 d_i 要以正或負邏輯來作處理，所以亂數序列 RS_j 在此也稱為隨機正負邏輯之序列。

使用本文提出的隨機正負邏輯時，究竟可比前人的固定式正邏輯得到什麼好處？在此我們進行實驗，逐次把位元藏入的門檻值 TH 提高，當依據正負邏輯之亂數序列來藏入位元後，作逆轉換回到時域的訊號波形，會與原始的訊號波形相當接近。但是，當使用Xin Li等人的方法[7]來藏入位元後，作逆轉換回到時域的訊號波形，會和原始訊號波形之間變得有明顯的差距存在。

一種測量信號差距的客觀作法是，計算訊噪比(signal-to-noise ratio, SNR)，其公式為：

$$10 \cdot \log_{10} \left(\frac{\sum_{n=0}^N s[n]^2}{\sum_{n=0}^N (s[n] - s'[n])^2} \right) \quad (4)$$

其中 $s[n]$ 表示原始信號， $s'[n]$ 表示藏入浮水印後的信號。在此我們使用5首不同型態的樂曲，分別以上述的兩種藏入方法，在不同的門檻值設定下，去作浮水印位元的藏入，然後逆轉換回到時域，再作訊噪比的

計算及取平均值。結果我們得到如表1所示的平均訊噪比值，從表1的數值可知，本文提出的使用隨機正負邏輯的藏入方法，在藏入浮水印位元後、逆轉回時域的訊號，無論位元藏入之門檻值設定為大或小，其訊噪比平均值都比Xin Li等人的方法[7]的還高許多，所以本研究的方法，可以接受設定較高的門檻值，來增強抵抗攻擊的能力(強健性)，而仍然可維持訊號的品質。一般來說，訊噪比在十幾dB時，雖然在視覺上可以觀察出波形的改變，但是在聽覺上卻還不會察覺訊號品質的改變。

表 1 兩種藏入方法之訊噪比平均值

門檻值 (TH)	使用正負邏輯亂數 序列及實數倒頻譜 訊噪比(dB)	使用 Xin Li 等人的方 法[8] 訊噪比(dB)
0.0005	28.79	19.45
0.001	26.80	16.83
0.002	23.70	9.65
0.004	18.91	-0.82

3.3 以雜步跳躍方式藏入浮水印

在前人的研究中，一個音框只藏入一個位元，此種作法所藏入的浮水印位元，會因為聲訊信號遭受局部的破壞而無法正確解出。並且一個音框只藏入一個位元，則所能藏入的位元個數就比較有限，而使得浮水印被藏入的次數也比較少。

為了解決上述問題，我們把藏入一個位元所使用的實數倒頻譜係數，分散在多個連續的音框中，並且使用比較少個係數(如200個)來藏入一個位元。詳細的處理步驟如下：

(a) 依據浮水印的位元總數 WS ，藏入一個位元所需之實數倒頻譜係數的個數 DS ，以及一個音框轉換出的可用之實數倒頻譜係數的數量 US ，計算整張浮水印藏入所需之音框數量 FS ，計算公式為 $FS = WS * DS / US$ 。以浮水印大小為4,096個位元為例，設藏入一個位元需使用200個實數倒頻譜係數，再者一個音框算出的1024個實數倒頻譜係數中，只有482個(如公式(1))是可使用的，因此依據前述公式可計算出， $FS = 4,096 * 200$

/ 482 = 1700，一共需要1700個音框。

(b) 經由步驟(a)之計算，設需使用 N 個音框來作浮水印藏入。當各個音框作實數倒頻譜轉換後，分別可得到1024個係數，接著把各音框的實數倒頻譜係數依音框次序串成一個序列 Z_i^k ，其中 k 表示音框編號， i 表示係數編號。

$$z_0^1, z_1^1, z_2^1, \dots, z_{1023}^1, z_0^2, z_1^2, \dots, z_{1023}^2, \dots, z_0^N, z_1^N, \dots, z_{1023}^N$$

(c) 從步驟(b)的序列中取出可以用來藏入浮水印位元的係數，而得到另一個序列：

$$z_{30}^1, z_{31}^1, z_{32}^1, \dots, z_{511}^1, z_{30}^2, z_{31}^2, \dots, z_{511}^2, \dots, z_{30}^N, z_{31}^N, \dots, z_{511}^N$$

(d) 將步驟(c)的序列重編索引，而得到另一序列：

$$y_0, y_1, y_2, \dots, y_{M-1}, \text{ where } M = 482 \times N$$

其中

$$y_0 = z_{30}^1, y_1 = z_{31}^1, y_2 = z_{32}^1, \dots, y_{M-1} = z_{511}^N$$

(e) 設 $b_0, b_1, \dots, b_{WS-1}$ 為浮水印位元之序列，則針對第 j 個位元 b_j ，依如下之雜步跳躍公式：

$$g(k) = (k \times WS) + (j + 1021 \times k) \bmod WS, \quad (5)$$

$$\text{for } k = 0, 1, \dots, DS - 1$$

由重編索引之 y_i 序列中取出 DS 個係數，即依序取出下標 i 為 $g(0), g(1), \dots, g(DS-1)$ 之 y_i 係數，在此 DS 表示藏入一個位元要使用 DS 個實數倒頻譜係數。

一個 $WS=4096$ 且 $DS=200$ 之例子，當使用公式(5)來為浮水印位元 b_3 選取係數時，相當於是從圖4的匯集陣列中，取出各列裡的一個隨機位置上的係數。

(f) 應用正負邏輯之亂數序列 RS_k ，依下列公式

$$SA_j = \frac{1}{DS} \sum_{k=0}^{DS-1} y_{g(k)} \cdot RS_{h(k)}, \quad (6)$$

$$h(k) = g(k) \bmod 482$$

求得正負邏輯作用後之平均值 SA_j 。

(g) 依平均值 SA_j 、門檻值 TH 、和3.1節藏入同步位元之步驟(c)和(d)的方法，調整隨選係數 $y_{g(k)}$ ， $k=0, 1, \dots, DS$ ，的數值，以藏入浮水印之 b_j 位元。

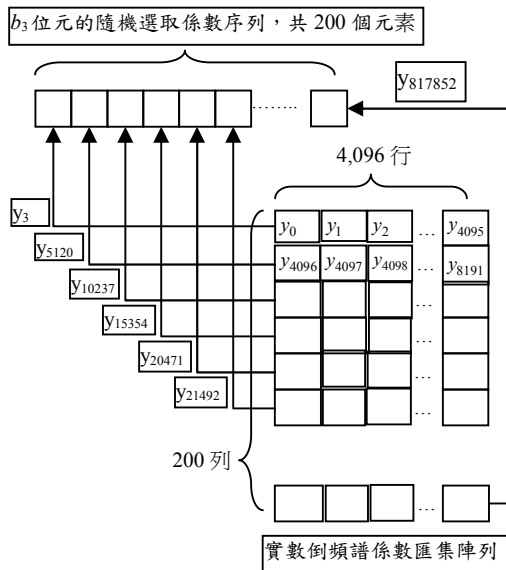


圖 4 取出隨選係數之圖形說明

(h) 對於 $j=0, 1, \dots, WS-1$, 反復執行步驟 (e)、(f)、(g), 以將所有浮水印位元 b_j 作藏入。

(i) 藏入所有浮水印位元後, 令 y_i 序列經調整後變成

$$\bar{y}_0, \bar{y}_1, \bar{y}_2, \dots, \bar{y}_{M-1}, \text{ where } M = 482 \times N$$

接著作如下取代 z_i^k 係數值的處理,

$$\bar{z}_{30}^1 = \bar{y}_0, \bar{z}_{31}^1 = \bar{y}_1, \bar{z}_{32}^1 = \bar{y}_2, \dots, \bar{z}_{511}^N = \bar{y}_{M-1}$$

(j) 為了保持一個音框的實數倒頻譜係數的偶對稱特性, 修改每一個音框另外一半(513點到994點, 共482點)的係數值。

(k) 將各音框調整後的實數倒頻譜係數序列作反向轉換, 以還原至時域的聲訊信號波形。

3.4 浮水印解出

在浮水印解出階段, 主要的處理流程如圖5所示, 其實許多的計算(如實數倒頻譜轉換、雜步跳躍方式、使用隨機正負邏輯之平均值計算)和浮水印藏入時所作的一樣的。

在尋找同步位元之方塊裡, 對於一個信號樣本點, 使用公式(1)來計算連續8個音框各自的平均值, 若一個音框的平均值大於等於0, 就判定解出同步位元1, 否則就判定解出同步位元0, 當8個音框解出的同

步位元形成10101010之序列時, 才判定是找到同步之樣本點, 否則就前進一個信號樣本點, 再繼續尋找同步位元。

當找到同步之樣本點後, 就進入下一個方塊, 將藏入區各個音框轉換出實數倒頻譜係數, 再使用雜步跳躍及隨機正負邏輯之方式(如公式(6)), 以計算出各個浮水印位元之平均值 SA_j , 若 SA_j 大於等於0, 就判定解出之浮水印位元為1, 否則就判定解出之浮水印位元為0。

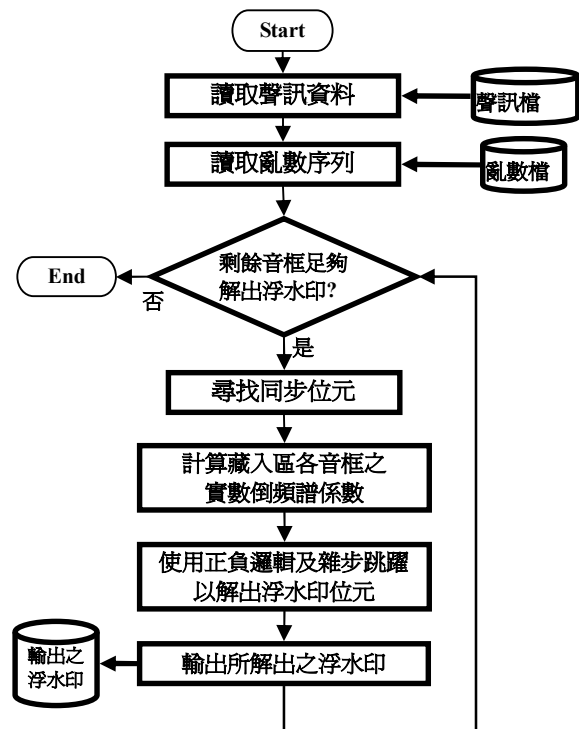


圖 5 浮水印解出之主流程

4. 測試實驗

為了方便作展示與測試, 我們依據圖3和圖5的流程, 製作出一個可供線上操作之聲訊浮水印系統, 該系統的程式畫面, 如圖6所示。

系統測試時, 我們使用5首不同樂曲型態的聲訊檔作為代表, 其中一首是人聲歌唱為主的樂曲; 一首為進行曲; 一首為鋼琴獨奏曲; 一首交響樂曲以及一首搖滾樂曲。樂曲聲訊檔的格式為: 單聲道, 取樣率44.1kHz, 每個樣本點以16個位元表示。至於藏入的浮水印, 我們使用一張64×64像素(pixels)的黑白影像。

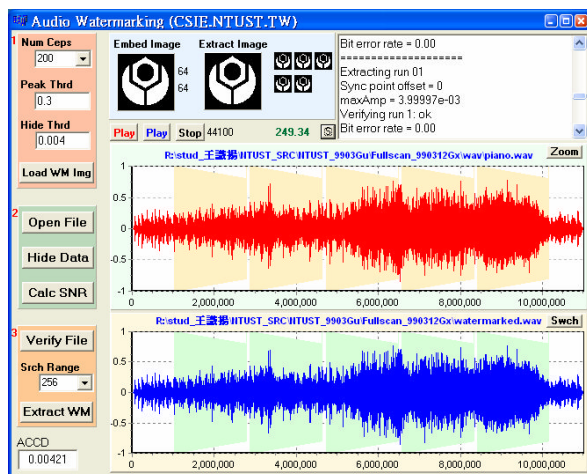


圖 6 聲訊浮水印系統之程式畫面

一般來說，藏入浮水印位元所用的門檻值 TH 愈大，則聲音品質會衰退得愈多。藉由人耳作聽測，我們發現如果門檻值大於0.015時，則藏入浮水印的樂曲，其音質衰退的程度，就會被聽者察覺出與原始樂曲不同。因此，當加大門檻值來增進對各類攻擊的抵抗力時，門檻值也不要大於0.015。以下我們就討論MP3壓縮、低通濾波等攻擊實驗。

4.1 MP3 壓縮攻擊

首先我們依照第3節的方法，在不同次的實驗裡，使用不同的門檻值及不同的 DS 值(係數個數)，將浮水印藏入5首樂曲的聲訊檔案中。然後使用CoolEdit軟體，在位元率為128kbps的情況下，對藏入浮水印的聲訊檔案作MP3壓縮。之後再把MP3儲存格式的檔案轉成wav格式，並進行浮水印的解出處理。

將解出的浮水印，和原始的浮水印作比較，我們可得到5首樂曲各自的bit error rate (BER，即錯誤的位元數除以浮水印的位元總數)，然後再對5首樂曲的BER值取平均，結果在不同的門檻值及 DS 值的組合情況下，我們得到BER的平均百分比值如表2所列。

從表2可知，本研究的浮水印藏入方法，只要門檻值 TH 大於等於0.004，藏入一個浮水印位元無論使用多少個實數倒頻譜係數(即 DS 值)，其解出浮水印之BER平均百分比值都會很靠近0，並且 DS 值愈大時，

BER平均百分比值也會降低，所以我們提出的藏入方法，對於MP3攻擊具有不錯的抵抗能力。另外我們也發現，當門檻值小於等於0.001時，解出浮水印的BER平均百分比值就會大幅上升，高到30%以上。關於原始浮水印的圖形、和幾個具有不同BER值的解出浮水印圖形，請參考圖7。

表 2 MP3 攻擊之 BER 平均百分比值

DS TH	100 BER(%)	150 BER(%)	200 BER(%)	250 BER(%)	300 BER(%)
0.001	36.95	32.43	27.93	34.00	28.41
0.002	10.38	6.38	2.49	1.39	0.41
0.004	0.76	0.23	0.06	0.00	0.01
0.006	0.03	0.00	0.00	0.00	0.00
0.008	0.00	0.00	0.00	0.00	0.00

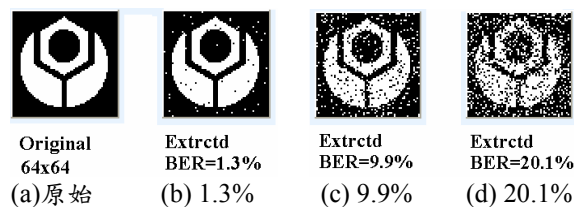


圖 7 原始和解出之浮水印圖形

4.2 低通濾波攻擊

我們依照第3節的方法，在不同次的實驗裡，使用不同的門檻值及不同的 DS 值(係數個數)，將浮水印藏入5首樂曲的聲訊檔案中。然後使用CoolEdit軟體提供的Butterworth低通濾波功能，設定其截止(cutoff)頻率為8kHz，來對藏入浮水印的聲訊檔案作低通濾波之攻擊，之後進行浮水印的解出處理。

將解出的浮水印，和原始的浮水印作比較，可得到5首樂曲各自的BER值，然後再對5首樂曲的BER取平均值，結果在不同的門檻值及 DS 值的組合情況下，我們得到BER的平均百分比值如表3所列。

表 3 低通濾波攻擊之 BER 平均百分比值

DS TH	100 BER(%)	150 BER(%)	200 BER(%)	250 BER(%)	300 BER(%)
0.002	36.91	38.76	34.02	37.83	31.41
0.004	14.40	9.81	5.72	4.04	2.52
0.006	4.79	2.50	1.18	0.53	0.13
0.008	1.37	0.37	0.17	0.07	0.01
0.010	0.44	0.09	0.04	0.00	0.00

從表3可知，當 DS 值設為250或300時，門檻值 TH 必須調至0.006，才能讓解出浮水印之BER平均百分比值靠近0；當 DS 值設為150或200時，則必須調門檻值 TH 至0.008，才能讓BER平均百分比值靠近0；而當 DS 值設為100時，則門檻值 TH 必須調至0.010，才能讓BER平均百分比值靠近0。所以，把 DS 值設得愈大，對於低通濾波攻擊，會愈有抵抗的能力。另一方面，當把門檻值 TH 調至0.002或更小時，則不管 DS 值設為多少，解出浮水印的BER平均百分比值都會大幅上升，高到30%以上。

另外，比較表2和表3可發現，對於MP3壓縮攻擊只需把門檻值 TH 調至0.004即可，然而對於低通濾波攻擊，則至少需把門檻值 TH 調至0.006，所以低通濾波攻擊是比較難抵抗的。我們希望使用較小的 TH 值，因為較大的 TH 值，會造成較大的音質破壞。

5. 結論

本研究在實數倒頻譜域上，提出以正負邏輯亂數序列與雜步跳躍之作法來藏入浮水印位元到聲訊資料，此一藏入方法我們尚未在其它文獻上看過。我們藉由使用雜步跳躍之位元藏入方式，來提高藏入的位元量，以增加浮水印藏入的次數，如此當藏入浮水印的聲訊檔案遭受局部破壞時，就有較多機會解出正確的浮水印。另外，所提出的正負邏輯之藏入觀念，可說是前人之正向邏輯藏入觀念的推廣 (generalization)，至於把正負邏輯觀念和亂數序列作搭配，除了可用以提升聲訊浮水印抵抗外在攻擊的能力，也可用以確保浮水印的安全性，以杜絕他人在未經授權的情況下取出或是竄改浮水印。

我們經由MP3壓縮攻擊、低通濾波攻擊之測試實驗，來評估本研究方法抵抗攻擊的能力。實驗結果顯示，我們可藉由調整藏入之門檻值 TH 、與藏入一個位元所使用的實數倒頻譜係數個數(DS 值)，來讓解出浮水印的位元錯誤率(BER值)很靠近0。所以本研究提出的聲訊浮水印方法，的確

可在不讓人感知音質改變的前提下，提供相當強健的抵抗MP3壓縮攻擊、低通濾波攻擊之能力。未來我們可再考慮對這個藏入方法作更多的加強，並且以更多的外在攻擊型式來測試它的防禦能力。

參考文獻

- [1] P. Bassia, I. Pitas, and N. Nikolaidis, "Robust audio watermarking in the time domain," *IEEE trans. on Multimedia*, Vol. 3, No. 2, pp. 232-241, 2001.
- [2] Q. S. Wang and S. H. Sun, "Watermark embedding algorithms based on quantizing frequency domain parameters of digital audio signal," *Acta Acustica.*, Vol. 27, No. 4, pp. 379-385, 2002.
- [3] Y. Wang, J. W. Huang, and Q. S. Yun, "Meaningful watermarking for audio with fast resynchronization," *Journal of Computer Research and Development*, Vol. 40, No. 2, pp. 215-220, 2003.
- [4] X. X. Niu and Y. X. Yang, "A new algorithm for digital watermarking based on the wavelet transform," *Journal of Computers*, Vol. 23, No. 1, pp. 21-27, 2000.
- [5] S. K. Lee and Y. S. Ho, "Digital audio watermarking in the cepstrum domain," *IEEE trans. on Consumer Electronics*, Vol. 46, No. 3, pp. 744-750, 2000.
- [6] B. L. Kuo, C. C. Lo, C. H. Liu, B. Y. Liao, and J. S. Pan, "Audio watermarking with HOS-based cepstrum feature", *Int. Conf. on Computational Collective intelligence*, Kaohsiung, Taiwan, pp. 316-323, 2010.
- [7] X. Li and H. H. Yu, "Transparent and robust audio data hiding in cepstrum domain," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, New York, USA, Vol. 1, pp. 397-400, 2000.
- [8] C. T. Hsieh and P. Y. Tsou, "Blind cepstrum domain audio watermarking based on time energy features", *IEEE Int. Conf. on Digital Signal Processing*, Santorini, Greece, 2002.
- [9] S. C. Liu and S. D. LIN, "BCH code-based robust audio watermarking in the

cepstrum domain", *Journal of Information Science and Engineering*, Vol. 22, pp. 535-543, 2006.

[10]B. K. Vivekananda, S. Indranil, and D. Abhijit, "Audio watermarking based on mean quantization in cepstrum domain,"

16th Int. Conf. on Advanced Computing & Communication, Chennai, India, pp. 73-77, 2008.

[11]O'Shaughnessy, D., *Speech Communications: Human and Machine*, IEEE Press, Piscataway, NJ, 2000.