中英文秘密文件的分享與隱藏

侯永昌
淡江大學資訊管理學系

杜淑芬
朝陽科技大學資訊管理學系

摘要

1998 年 Lin et al.提出了一個新的文件保護機制，利用有具體內容的文件來傳送秘密訊息，其後亦陸續有學者針對此方法加以延伸與改進，但是這套文件保護的方法仍有許多待改進之處。本研究的目的便是針對此套文件保護的機制，提出一個改進的方法，將秘密文件位置索引表改成差值表，並且增加機密分享的機制。本研究與其它研究比較，具有以下幾項優點：(1)本研究可以取用任何文字做為欺敵訊息；(2)不同的秘密訊息可以共用相同的欺敵訊息；(3)本研究可以處理中英文夾雜的訊息；(4)本研究可以防止部份差值表遺失以致無法還原秘密的情況；(5)本研究可以達到機密控管，降低洩密的風險；(6)最後所傳送出去的檔案皆是有具體內容的文件，不容易引人懷疑。

關鍵字： 文件保護機制、秘密分享、資訊隱藏

A Bilingual Secret Document Sharing and Hiding Scheme

Young-Chang Hou
Department of Information Management, Tamkang University

Shu-Fen Tu
Department of Information Management, Chaoyang University of Technology

Abstract

In 1998, Lin & Lee proposed a new document protection scheme which utilized a meaningful cheating document to protect the secret document from hackers. Afterward some researchers extended and improved their scheme. However, there are still lots of spaces for improvement. The purpose of our study is aimed at reforming their method via recording the code differences instead of recording the code indices, coupled with a secret sharing scheme. Compared with other studies, ours have the following advantages: 1. we can utilize any text as a cheating message; 2. different secrets can use the same cheating message in common; 3. we can handle a bilingual secret document easily without any special treatment; 4. we can recover the code difference table even if one of the share messages is lost; 5. we can decrease the risk of blabbing out the secret; 6. all the files delivered to the receivers are meaningful without giving rise to attentions of hackers.

Keywords: Document Protection Scheme, Secret Sharing, Information Hiding

壹、簡介

如何確保秘密文件能安全無虞地送達對方手中，一直是從古至今的研究課題。在尚未數位化的時代，便有人將秘密文件以雙方事先約好的協定，做某種順序的排列，或是參雜在其它文件的特定位置中(Neil & Sushil 1998) 。如今進入數位化的時代，資訊的交流更為便利，但相對地，秘密文件傳送的安全性卻降低了。因此資訊安全變成網際網路興起後的一個熱門的研究。

密碼學便是資訊安全中的一門學問，其方法主要是將秘密文件加密，將加密後的文件還原是有一定困難度的，通常是需要花費非常大量的時間和資源來完成。但是由於科技日新月異，過去一些保證在一定時間內無法破解的密碼，也紛紛傳出被破解的消息(黃景彰 2001)。由於密碼的破解並非不可能，只是時間早晚的問題，而且加密後的文件通常是沒有意義的，這種沒有意義的文件特別容易引起駭客的高度興趣。因此便有「資訊隱藏」的研究被提出，也就是把秘密隱藏在不會引起別人懷疑的文件或影像中，只要不露出蛛絲馬跡，就可以達到欺敵的效果。因為網路上流傳的資訊量非常大，駭客無法一一加以攔截監看，因此使用一般的文件或影像來隱藏秘密，較不致引起駭客懷疑。

通常使用一份欺敵的文件來掩護秘密訊息會受到文件大小的限制，也就是欺敵的文件大小不得小於秘密訊息，Lin & Lee (1998) 便提出一種秘密文件的保護機制，可以使用一份較小的欺敵文件，來傳送大量的秘密訊息。Lin et al.的方法需找尋一份欺敵文件，其中的字元能包含秘密文件中所有的字元，利用字元之間的對映關係建立一個秘密文件索引檔。但是 Lin et al.只處理了單位元碼的文件，Yeh & Hwang (2001a, 2001b) 則利用十六進位碼的表示方式，將字元集合縮小至只剩'0'～'9'、'A'～'F'十六種字元，使得此套理論可以應用在雙位元碼的文件上。由於 Lin & Lee 和 Yeh & Hwang 都是將秘密文件位置索引表以傳統密碼學方式加密後再傳送出去，其無意義的內容容易引起駭客的懷疑。因此 Wang & Lu (2003) 便將加密後的資料藏入一張灰階圖中，使得最後傳送出去的資訊是不致引人懷疑的文件和影像。除此之外，他們以影像的雙語說明文件來分別傳送中文和英文的秘密訊息。

然而，這套以欺敵訊息來保護秘密文件傳送的方法仍有許多待改進之處。首先，也是最嚴重的問題，便是無法任意取用任何文字做為欺敵訊息，當然也無法達到以相同的欺敵訊息，來掩護不同的秘密訊息的傳送。其次便是無法處理中英文夾雜的文件，雖然在 Wang & Lu 的研究中提出了一個傳送雙語文件的方法，但是他們所提出的方法只能解決當中文和英文是分屬於兩則獨立訊息的情況，而不是在相同的訊息中同時夾雜有中文和英文的情形。第三，由於此套機制是記錄字元的位置索引值，若欺敵訊息的字元不夠亂不夠多，便有露出蛛絲馬跡的可能。最後，秘密文件索引檔是還原機密訊息的重要關鍵，如果遺失了便無法還原機密訊息。從另一方面來說，機密掌握在一個人的手上，也存在著機密控管的問題。

本研究便提出一套傳送夾雜中英文秘密訊息的方法，透過記錄文件內碼差值的方法，便可以取用任何圖文做為欺敵文件，而不需要檢查欺敵文件中是否有包含秘密文件的字元。同時為了增加差值表傳送的安全性，我們將差值表利用機密分享的機制分解成多份，而且所分解出的每一份都有絕對的安全性。我們將所分解的差值表分別藏入不同的明圖中，因此最後所傳送出去的也是不致引人懷疑的明圖。這幾張圖可分別傳送給不同的人保管，必須要集合其中幾份才能還原原來的差值表，因而可以達到控管的目的，防止機密外洩。

貳、文獻探討

一、秘密文件的傳送

由於 Yeh & Hwang 和 Wang & Lu 大致上都是承襲 Lin & Lee 的方法，因此我們先介紹 Lin & Lee 的作法。首先選擇一份有具體內容的欺敵文件，將每個字元依照出現的順序給予一個位置索引值，然後將秘密訊息中的每個字元 P_i 依序轉換成 P_i 出現在欺敵文件中的位置，如果 P_i 在欺敵文件中可能出現不只一次，則隨機選取一個位置索引值來做記錄。藉由此種方式，即將秘密訊息轉換成一個文件索引檔(Plain-text Index File, PIF)，最後將欺敵文件及加密後的索引檔傳送出去，即可完成秘密文件的傳送工作。

但是 Lin & Lee 的做法只適合字元集比較小的英文系統，在英文系統中可能出現的字元總共只有 128 個，可是以中文字而言，至少有 5401 個常用的字。因此 Yeh & Hwang 提出利用內碼系統將中文字元轉成十六進位碼的方式，使得文件中可能出現的字元縮小至'0'～'9', 'A'～'F'共十六個元素。將欺敵文件和秘密文件都分別轉成十六進位碼，接下來便依照 Lin & Lee 的作法建立秘密訊息內碼的文件索引檔(Chinese Document Index File, CDIF)。在還原秘密訊息時，須先將欺敵文件轉成十六進位碼，再配合秘密訊息內碼的文件索引檔，還原出秘密訊息的內碼，再由此內碼組合成原來的中文秘密訊息。

由於 Lin & Lee 和 Yeh & Hwang 兩者的方法最後都是傳出加密的索引檔，而經過加密的檔案呈現的是無意義的內容，容易引起駭客的懷疑，因此 Wang & Lu 將加密過的文件索引檔轉成二進位碼，並且選擇一張影像來隱藏加密文件的位元。隱藏的方法則是隨機選取影像上的一個像素，根據所設定一個門檻值 T 及兩個模組數 m_u 和 m_l，如果像素值大於 T，則可以隱藏$\lfloor \log_2 m_u \rfloor$個加密文件的位元；如果像素值小於 T，則可以隱藏$\lfloor \log_2 m_l \rfloor$個加密文件的位元(Wang & Yang 2001)。若要同時傳送中文秘密訊息和英文秘密訊息，則欺敵文件除了包含要用來隱藏索引檔的圖片外，還要包含兩則獨立的中文和英文的欺敵訊息，英文的部份依照 Lin et al.的方法建立索引檔，中文的部份依照 Yeh & Hwang 的方法建立索引檔，這兩個索引檔分別加密後轉成二進位碼藏入一張圖片中，所以最後傳送出去的是一張圖片和兩段當做欺敵文件的雙語說明文字。

在這一系列的方法中我們可以看出來，欺敵文件一定要包含秘密文件的所有字元，否則會出現無字元可對映的情況。當發生這種情況時，就必須再另外尋找其它的欺敵文件來代替。因為每次都必須做這樣的檢查，因此無法真正達到任意一份文件皆可做為欺敵文件的要求。我們以 Yeh & Hwang (2001b)所舉的例子來說明：

秘密訊息：二八四師向林口集結。

欺敵訊息：【職籃消息】羅興樑於職籃五年將暫披達欣戰袍。

這兩段文字轉換成十六進位的 Big-5 碼時，表示如下：

秘密訊息：A4 47 A4 4B A5 7C AE 76 A6 56 AA 4C A4 66 B6 B0 B5 B2 A1 43

欺敵訊息：A1 69 C2 BE C4 78 AE F8 AE A7 A1 6A C3 B9 BF B3 BC D9 A9 F3 C2 BE C4 78 A4 AD A6 7E B1 4E BC C8 A9 DC B9 46 AA 59 BE D4 B3 54 A1 43

我們可以發現，在秘密訊息中的內碼'0'沒有出現在欺敵訊息中，也就是說，欺敵訊息沒有包含秘密訊息中所有的字元。唯一的解決辦法就是再增加欺敵訊息的文字，或者另取其它的欺敵訊息。

由於索引表中是記錄著秘密訊息的字元在欺敵訊息中出現的位置索引值，如果有的字元在欺敵訊息中只出現一次，那麼在秘密訊息中每次遇到此字元時，就必須記錄相同的數值。若這個字元在秘密訊息中出現非常多次，則相同的數值便會出現許多次，如此一來便留下蛛絲馬跡；而有的字元在秘密訊息中從未出現過，但是在欺敵訊息中卻出現多次，那麼欺敵訊息中便有許多字元是無用的。我們觀察 Wang & Lu (2003)在其論文中所舉的例子，便出現很多這種情況。因此，一方面為了讓索引表保有一定的亂度，另一方面在欺敵訊息中可能有很多字元是秘密訊息中用不到的，通常欺敵訊息的字元數都會比秘密訊息的字元數大的多。

如果想要傳送一份中英文夾雜的秘密訊息，在上述的研究中都無法達到。雖然 Wang & Lu 提出一個可以傳送中英文秘密訊息的方法，但是他們其實是將中文和英文的秘密訊息分開傳送，而非中英文夾雜的訊息。所以他們必須針對這兩則獨立的訊息，分別找尋一段欺敵文字，並且分別建立兩個不同的索引檔，所以不能算是真正的雙語文件傳送的方法。

利用欺敵文件加秘密文件索引檔的這套方法中，秘密文件索引檔是還原秘密的重要關鍵，如果遺失的話便無法還原秘密訊息。另一方面，關鍵秘密掌握在接收者一人手中，一旦此人洩露秘密，秘密便完全被他人得知。因此本研究希望透過機密分享的機制，來增加文件索引檔的安全性。

二、視覺式機密分享機制探討

Noar & Shamir (1994) 提出一個(k, n) 視覺式的機密分享機制((k, n)-threshold visual secret sharing scheme)，使得機密訊息被分解成 n 份，想要還原其中的秘密訊息，則需要從 n 份中取出至少 k 份出來，將它們重疊後，就可以經由人類的視覺系統解密。最初這個機制是設計在黑白影像上的，機密影像會分解成 n 份(share)，分別呈現在投影片上。機密影像上的每個點在每一個分享影像上擴展成 m 個點，因此要分解機密影

像，首先需要兩個 $n \times m$ 的布林矩陣(Boolean matrix) S_{white} 和 S_{black}，分別代表機密影像上的黑點與白點的分享機制。

以(2, 2)-threshold 分享機制為例：

$$S_{white} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad S_{black} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

其中'0'代表分享影像上的白點，'1'代表分享影像上的黑點，根據上面這兩個矩陣，分別形成 C_0 和 C_1 兩個集合，而 C_0 (C_1) 中的每個元素代表 S_{white} (S_{balck}) 矩陣每一行重新排列的結果，因此 C_0 和 C_1 如下所示：

$$C_0 = \left\{ \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right\}$$

$$C_1 = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\}$$

若要分享機密影像上的黑點時，則從 C_1 中隨機取用一個矩陣，矩陣上的第一列的 2 個點填入第一張分享影像上，第二列的 2 個點則填入第二張分享影像上；反之，若要分享機密影像上的白點時，則從 C_0 中隨機取用一個矩陣。矩陣第一列的 2 個點就填入到第一張分享影像上，第二列的 2 個點則填入第二張分享影像上。

(k, n)視覺式機密分享機制為了達到絕對的安全性，對於 C_0 和 C_1 設定了三個需滿足的條件(Naor & Shamir 1994)：

(1)C_0 中任何一個元素 S，任取其中 k 列經過 OR 運算所產生的 m 維向量 V 必須滿足 $H(V) \le d - \alpha m$

(2)C_1 中任何一個元素 S，任取其中 k 列經過 OR 運算產生的 m 維向量 V 必須滿足 $H(V) \ge d$

(3)令 D_t ($t = 0, 1$)代表由 Ct ($t = 0, 1$)中的矩陣取其中 q 列所形成的 $q \times m$ 矩陣的集合，假設 $q < k$，則相同的矩陣在 D_0 和 D_1 這兩個集合中出現的次數是一樣的，表示 D_0 和 D_1 是無法分辨的(indistinguishable)。

其中 $H(V)$代表 m 維向量 V 的 Hamming weight，d 代表一個門檻值且 $1 \le d \le m$，α則是區別黑色與白色之間的相對差值。

上述的前兩個條件稱為對比條件，也就是說，代表白點的分享影像重疊後，其黑點的個數必須小於 $d - \alpha m$；而代表黑點的分享影像重疊後，其黑點的個數必須大於門檻值 d，因此，代表白點的分享影像重疊後，就會與代表黑點的分享影像有一定的色差。第三個條件稱為安全條件，也就是說，任何少於門檻值 k 的分享影像重疊後，在重疊影像上是無法得出任何有關機密影像的訊息。

本研究中便是利用(k, n)-threshold 視覺式機密分享的機制，將重要的訊息分解成 n 份，以避免重要資訊由一人掌握的弊病，取其中 k 份便可還原訊息，以避免部份資訊遺失，即無法還原機密資訊的問題。如果 $k = n$，則代表必須取得所有的分享訊息，才能還原機密訊息。但是在視覺式機密分享中針對的是黑白影像，在本研究中則是利用在機密文件的分享上。

參、加密演算法

一、建立秘密文件差值表

為了掩護秘密文件的傳送，我們隨意選取一段有具體內容的中文字做為欺敵的文件，將兩份文件以 Big5 的中文內碼系統，分別轉換成十六進位內碼。令 P 代表轉換成內碼的秘密訊息之字元集合，$P=\{P_i \mid i=1\sim p\}$，其中 i 代表秘密字元的位置索引值，p 代表秘密訊息內碼的字元個數。同理，令 H 代表轉換成內碼的欺敵訊息之字元集合，$H=\{H_j \mid j=1\sim q\}$，其中 j 代表欺敵字元的位置索引值，q 代表欺敵訊息內碼的字元個數。針對 P 中每一個元素 P_i，我們隨機選取 H 中的一個元素 H_j，並且計算這兩個內碼間的差值，以建立秘密訊息的 Big5 碼文件差值表(Code Difference Table, CDT)。由於 P 和 H 的字元集合只會包含'0'到'9'、'A'到'F'等十六個元素，因此所有差值的範圍限制在-15 到 15 之間。為了便於機密分享和隱藏，我們將差值表中所有的值轉換成 5 個位元的二進位值，以最高位元代表數值的正負號，'0'代表正值，'1'代表負值，剩下四個位元則代表'0'～'15'的數值，例如'-3'便轉換成 10010，我們將轉換後的差值表稱為二進位差值表(Binary Code Difference Table, BCDT)。

在我們計算字元差值時，是以隨機配對的方式，因此我們設定一個根據 *seed* 值產生虛擬隨機亂數的函數 *per*，這個函數可以將某一位置索引值 i 對映至另一位置索引值 $per(i)$。當我們要計算秘密訊息內碼的字元 P_i 之差值時，利用 *per* 函數我們可以將 P_i 對映至位置索引為 $per(i)$的欺敵訊息內碼，亦即 $H_{per(i)}$，因此我們便可得出代表秘密訊息內碼 P_i 的差值：

$$diff(i) = P_i - H_{per(i)} \tag{1}$$

這個差值會落在-15～+15 之間，可以用 5 個位元來編碼，因此整個 BCDT 的長度為 $5\times p$，其中 p 代表秘密訊息內碼的字元個數。將來要還原這個隨機對映的順序，必須使用相同的 *seed* 值，因此這個 *seed* 值也必須傳送至接收端。

二、機密分享

為了達到控管的目的，我們將 BCDT 分解成 n 份，分別由 n 個人持有，而且必須集合 k 份以上的分享訊息，才能還原出原來的 BCDT。根據我們所設定的(k, n)-threshold 視覺式機密分享機制，設計兩個 $n\times m$ 的位元矩陣 S_{white} 和 S_{black}，將這兩個矩陣的每一行重新排列，分別形成兩個矩陣集合 C_0 和 C_1，這兩個矩陣集合就可以分別應用在位元'0'和位元'1'的分享上。從 BCDT 中第一個位元開始，若遇到位元'0'時，則從 C_0 集合中隨機挑選出一個矩陣，將位元'0'分解成 n 份，將矩陣中第 i 列的 m 個位元分配給第 i 個持有者；若遇到位元'1'時，則從 C_1 集合中隨機挑選出一個矩陣，將位元'1'分解成 n 份，同樣地將矩陣中第 i 列的 m 個位元分配給第 i 個持有者，一直到 BCDT 中每個位元都分享完畢，便可以得出 n 份分享訊息 $R^1\sim R^n$。此時每一份分享訊息的長度

為 $5 \times m \times p$，其中 m 代表訊息擴展的倍數。

另外，我們也將建立 CDT 時所使用的 *seed* 值，依同樣的方法分解成 n 個值，然後分別傳送給 n 個人，因此還原差值表時，參與還原機密的人必須拿出他的分享訊息，以及他所分得的 *seed* 值。以(2, 2)-threshold 視覺式機密分享機制來舉例說明，假設我們所使用的 *seed* 值為 100，換算成二進位則為 1100100，因為第一個位元為'1'，所以我們從集合 C_1 中隨機選取一個矩陣，假設挑選到第二個矩陣，則將位元'1'拆解成兩個值，分別為 01 和 10，第二個位元亦為'1'，所以執行同樣的步驟，假設所得出的兩個值為 10 和 01；第三個位元為'0'，則從集合 C_0 中隨機挑選一個矩陣，假設挑選到第一個矩陣，則位元'0'分解成兩個值，分別為 10 和 10，之後的位元依此類推。當所有位元皆分解完畢後，便得到兩個二進位值，分別為 01101010010110 及 10011010100110，這兩個值轉換成十進位則分別為 6806 和 9894，這兩個值便分別傳送給分享差值表的 2 個人。

三、隱藏分享訊息

為了不引起駭客的懷疑，我們希望最後所傳送出去的都是有具體內容的檔案，因此我們將分解後的差值表(亦即 n 份分享訊息)，以 LSB 法分別隱藏在不同的影像當中。一張灰階影像的像素共有八個位元，通常最低的三個位元的變動，人眼是無法察覺到其中的差異的，因此我們將修改的範圍限制在影像中每個像素最低的第 0 個位元平面到第 2 個位元平面上。

首先我們選擇 n 張適當大小的灰階影像 $I^1 \sim I^n$ 做為掩蓋影像，將分享訊息 $R^1 \sim R^n$ 分別藏入影像 $I^1 \sim I^n$，隱藏的方式則是在第 0 個位元平面 $I^t{}_0$ 的前 32 個位元，儲存記錄分享訊息的長度 $5 \times m \times p$。從第 0 個位元平面的第 32 個位元開始分別儲存分享影像 R^t 的每個位元 $R^t[i]$，若第 0 個位元平面的大小小於分享訊息的長度 $5 \times m \times p$，則可以依上述的方法，依序將分享訊息儲存到第 1 個位元平面，甚至第 2 個位元平面，直到所有分享訊息的位元都處理完畢為止。

<u>加密演算法</u>

輸入：(1) 欺敵訊息
(2) 秘密訊息
(3) n 張灰階影像
(4) 虛擬隨機重排的 *seed* 值

步驟：(1) 將欺敵訊息和秘密訊息分別轉換成十六進位的 Big5 內碼，並計算秘密訊息的長度 p。
(2) 針對秘密訊息中每一個內碼，計算下列的差值：

$$diff(i) = P_i - H_{per(i)}$$

其中 P_i 代表位置索引值為 i 的內碼，per 為一個虛擬隨機函數，目的是將位置索引 i 對映至另一個位置索引 $per(i)$。

(3) 將所有差值轉換成 5 個位元的二進位值，其長度為 $5 \times p$。

(4) 將所有的二進位差值依(k, n)-threshold 視覺式機密分享機制分解成 n 份分享訊息，其長度為 $5 \times m \times p$。

(5) 將 *seed* 值轉換成二進位，並且依(k, n)-threshold 視覺式機密分享機制分解成 n 個十進位的值。

(6) 分享訊息的長度儲存在影像最低位元平面的前 32 個位元，並且將 n 份分享訊息分別隱藏至 n 張掩蓋影像中。

輸出：n 張偽裝影像及 n 個鍵值 $seed_1 \sim seed_n$

肆、解密演算法

要還原原來的 BCDT，則必須至少集合 k 個人，每個人從他們所分得的偽裝影像和 *seed* 中取出必要的分享訊息。也就是從偽裝影像第 0 個位元平面的前 32 個位元中可以得知每份分享訊息的長度 $5 \times m \times p$，然後依序由影像的第 32 個位元開始取出 $5 \times m \times p$ 個位元，因此會得到 k 份長度為 $5 \times m \times p$ 的二進位分享訊息。根據(k, n)-threshold 視覺式機密分享機制，機密訊息上的一個位元會擴展成 m 個位元，亦即分享訊息上每 m 個位元，代表機密訊息的一個位元，因此我們從 k 份分享訊息中，每次取出 m 個點形成一個 $k \times m$ 的位元矩陣，將矩陣的每一列做 OR 運算會得出一個 m 維的向量 V。根據(k, n)-threshold 視覺式機密分享機制的門檻值 d，如果 $H(V) \geq d$ 則向量 V 代表的是位元'1'，如果 $H(V) < d$ 則向量 V 代表的是位元'0'，其中 $H(V)$代表向量 V 的 Hamming weight。

同樣以(2, 2)-threshold 視覺式分享機制來舉例說明，此分享機制的門檻值為 2，而且機密訊息每個位元在分享訊息上會擴展成 2 個位元。假設 2 份分享訊息的前 2 個位元分別為 10 和 01，經過 OR 運算後得到一個 2 維向量[1 1]，由於 H([1 1]) = 2，所以此向量在機密訊息中所對應的位元值為'1'；假設前 2 個位元分別為 01 和 01，經過 OR 運算後得到一個 2 維向量[0 1]，由於 $H([0\ 1]) = 1 < 2$，因此此向量在機密訊息中所對應的位元值為'0'。依此類推，當所有分享訊息都計算完畢後，便可還原出原來的 BCDT。

還原出 BCDT，尚需轉換成原來的差值，由於我們是以 5 個位元代表一個差值，且最高位元代表差值的正負號，'0'代表正值，'1'代表負值，其它的 4 個位元代表差值的絕對值。因此將 BCDT 中每 5 個位元為一個單位，就可求出正確的內碼差值。

由於這個差值在計算時是利用 *per* 函數，將秘密訊息內碼 P_i 隨機對應至欺敵訊息的某一個內碼 $H_{per(i)}$，因此我們必須利用原來的 *seed* 值，才能得知原本的配對關係。因此我們取至少 k 個人的分享值 $seed_i$，同樣依照還原二進位差值表的方式，便可得出原來的 *seed* 值。利用這個 *seed* 值就可以正確的計算 $per(i)$，以還原出原來的十六進位秘密訊息內碼：

$$p_i = diff(i) + H_{per(i)} \tag{2}$$

還原出十六進位的秘密訊息內碼後，便可還原出中文的秘密文件。

解密演算法：

輸入：(1) k 張灰階偽裝影像

(2) 欺敵訊息

(3) $seed_1 \sim seed_k$

步驟：(1) 從灰階偽裝影像中的前 32 個位元，讀取分享訊息的長度 $5 \times m \times p$。

(2) 從 k 張灰階偽裝影像中的第 0 個位元平面的第 32 個位元開始，依序取出 $5 \times m \times p$ 個位元，得出 k 份長度為 $5 \times m \times p$ 的分享訊息。

(3) 將 k 份分享訊息依據(k, n)-threshold 視覺式機密分享機制的門檻值 d，以及擴展位元倍數 m，還原出原來的二進位差值表。

(4) 將二進位差值表以每 5 個位元為單位，還原出每一個秘密訊息的的差值 $diff(i)$。

(5) 將 k 個 $seed_i$ 值轉換成 k 個二進位值，並且根據(k, n)-threshold 視覺式機密分享機制的門檻值 d，以及擴展位元倍數 m，還原出原來的 $seed$ 值。

(6) 將欺敵訊息轉換成十六進位的 Big5 內碼。

(7) 利用差值表、$seed$ 值、以及欺敵訊息內碼，還原出原來的秘密訊息內碼：

$$p_i = diff(i) + H_{per(i)}$$

(8) 將秘密訊息內碼轉換成中文文件。

輸出：秘密訊息

圖 1顯示我們加密與解密的流程圖。

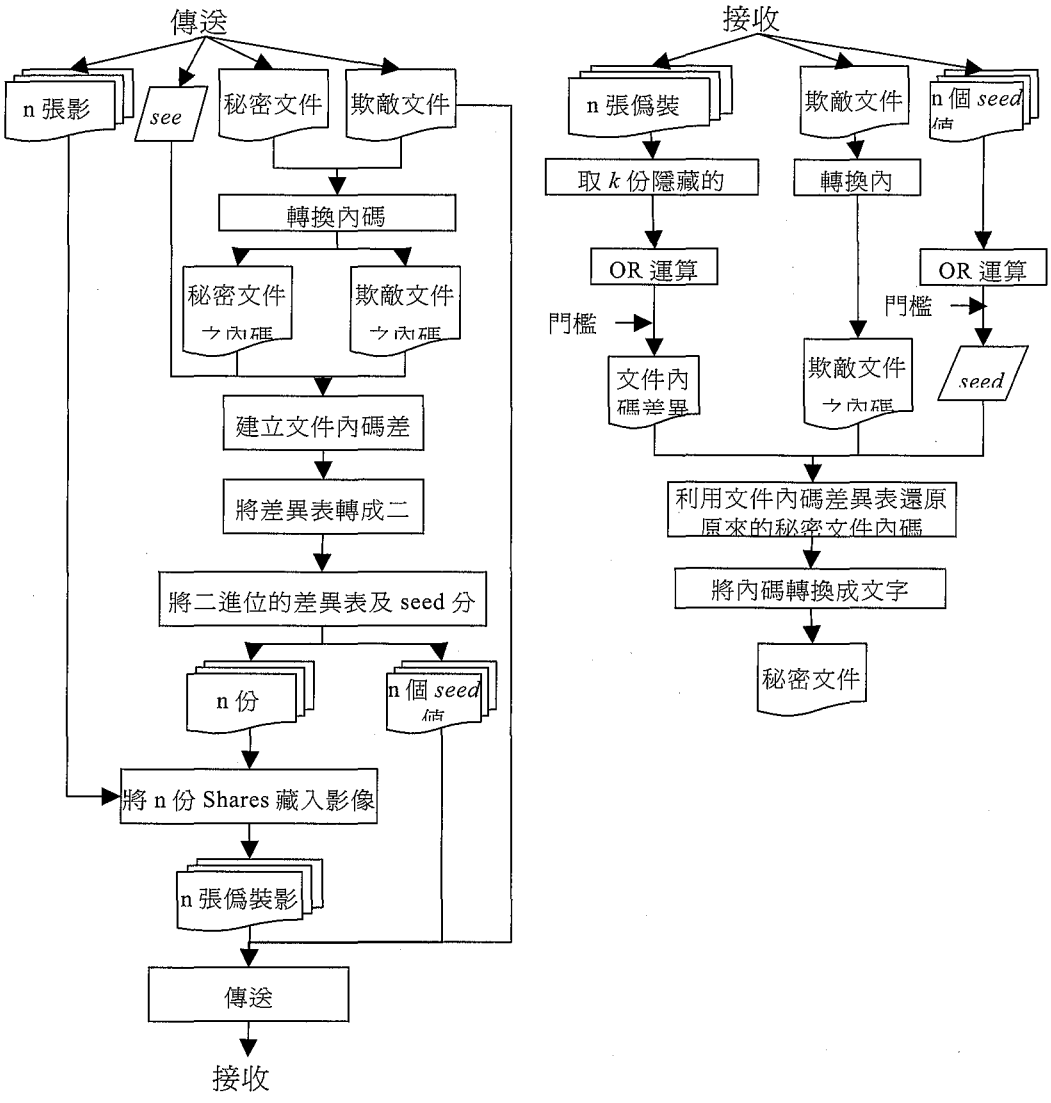


圖1：加密與解密的流程圖

伍、實驗結果與討論

我們以(2, 3)-threshold 視覺式機密分享機制為例，定義兩個 3 × 3 的矩陣：

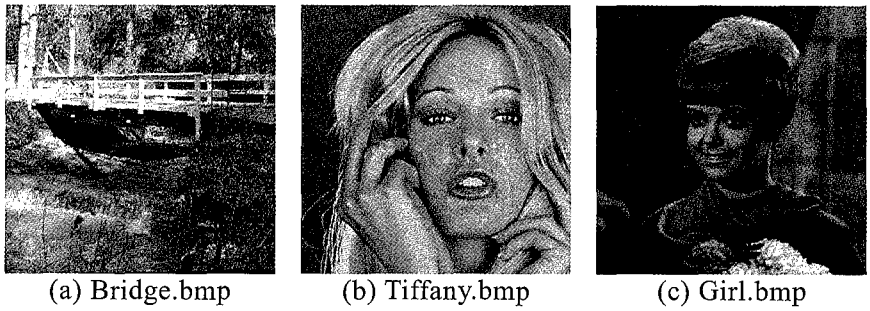
$$S_{white}=\begin{bmatrix}1&0&0\\1&0&0\\1&0&0\end{bmatrix},\ S_{black}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

由這兩個矩陣我們可以得出兩個矩陣集合 C_0 和 C_1：

$$C_0=\left\{\begin{bmatrix}1&0&0\\1&0&0\\1&0&0\end{bmatrix},\begin{bmatrix}0&1&0\\0&1&0\\0&1&0\end{bmatrix},\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}\right\}$$

$$C_1=\left\{\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix},\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix},\begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix},\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix},\begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}\right\}$$

由我們所定義的矩陣可以看出，將矩陣中的每一列經過 OR 運算後得出的 3 維向量中，屬於 C1 的矩陣都有 2 個以上的'1'，代表原機密訊息的位元為'1'，屬於 C0 的矩陣都只有 1 個 1，代表原機密訊息的位元為 0，因此判別'0'與'1'的門檻值可以設為 2。



(a) Bridge.bmp (b) Tiffany.bmp (c) Girl.bmp

圖2：三張 256 × 256 pixels 的灰階掩蓋影像

一、實驗一：中文秘密文件

表 1 為本實驗的欺敵訊息和秘密訊息，圖 2則為用來隱藏分享訊息的三張灰階影像。

表1：中文之欺敵訊息和秘密訊息

欺敵訊息	秘密訊息
飼養動物保護法公告禁止飼養的美洲巨水鼠，將遭強制沒收全部銷毀，並罰五萬元。	全球登山界這幾天正為人類首度成功攀登聖母峰五十週年舉行慶祝活動，不過世界自然基金會警告，溫室效應可能會造成喜瑪拉雅山地區的浩劫。世界自然基金會表示，聖母峰的冰帽因為溫室效應的影響，正逐漸融化，當地出現許多過去沒有的湖泊，而一旦融化的水量超過一定限度，湖泊的水可能會氾濫成災，直接淹沒山下的雪巴人村落。不只當地的地形地貌會出現變化，連村民的性命都飽受威脅。

兩則訊息轉換成十六進位 Big5 碼後，表示如下：

欺敵訊息：

B9 7D BE 69 B0 CA AA AB AB 4F C5 40 AA 6B A4 BD A7 69 B8 54 A4 EE B9 7D BE 69 AA BA AC FC AC 77 A5 A8 A4 F4 B9 AB A1 41 B1 4E BE 44 B1 6A A8 EE A8 53 A6 AC A5 FE B3 A1 BE 50 B7 B4 A1 41 A8 C3 BB 40 A4 AD B8 55 A4 B8 A1 43

秘密訊息：

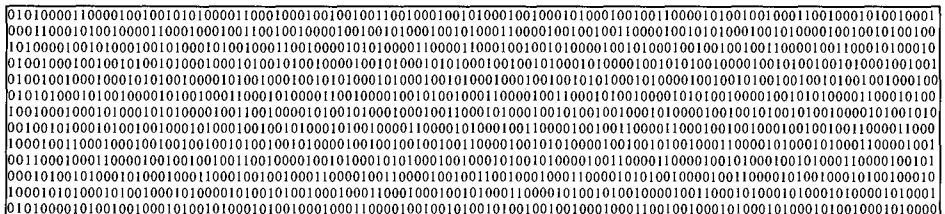
A5 FE B2 79 B5 6E A4 73 AC C9 B3 6F B4 58 A4 D1 A5 BF AC B0 A4 48 C3 FE AD BA AB D7 A6 A8 A5 5C C3 6B B5 6E B8 74 A5 C0 AE 70 A4 AD A4 51 B6 67 A6 7E C1 7C A6 E6 BC 79 AF AC AC A1 B0 CA A1 41 A4 A3 B9 4C A5 40 AC C9 A6 DB B5 4D B0 F2 AA F7 B7 7C C4 B5 A7 69 A1 41 B7 C5 AB C7 AE C4 C0 B3 A5 69 AF E0 B7 7C B3 79 A6 A8 B3 DF BA BF A9 D4 B6 AE A4 73 A6 61 B0 CF AA BA AF 45 A7 54 A1 43 A5 40 AC C9 A6 DB B5 4D B0 F2 AA F7 B7 7C AA ED A5 DC A1 41 B8 74 A5 C0 AE 70 AA BA A6 42 B4 55 A6 5D AC B0 B7 C5 AB C7 AE C4 C0 B3 AA BA BC 76 C5 54 A1 41 A5 BF B3 76 BA A5 BF C4 A4 C6 A1 41 B7 ED A6 61 A5 58 B2 7B B3 5C A6 68 B9 4C A5 68 A8 53 A6 B3 AA BA B4 F2 AA 79 A1 41 A6 D3 A4 40 A5 B9 BF C4 A4 C6 AA BA A4 F4 B6 71 B6 57 B9 4C A4 40 A9 77 AD AD AB D7 A1 41 B4 F2 AA 79 AA BA A4 F4 A5 69 AF E0 B7 7C A5 C6 C0 DD A6 A8 A8 61 A1 41 AA BD B1 B5 B2 54 A8 53 A4 73 A4 55 AA BA B3 B7 A4 DA A4 48 A7 F8 B8 A8 A1 43 A4 A3 A5 75 B7 ED A6 61 AA BA A6 61 A7 CE A6 61 BB AA B7 7C A5 58 B2 7B C5 DC A4 C6 A1 41 B3 73 A7 F8 A5 C1 AA BA A9 CA A9 52 B3 A3 B9 A1 A8 FC AB C2 AF D9 A1 43

欺敵訊息內碼和秘密訊息內碼的差值表則如表 2 所示：

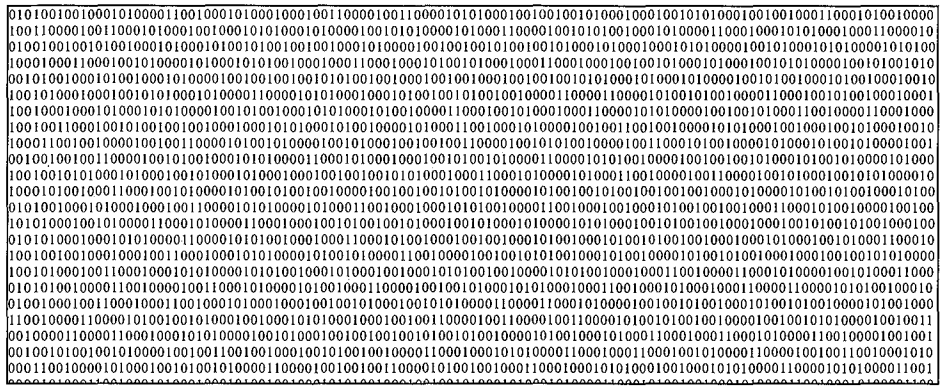
表2：中文秘密文件與欺敵文件之間的差值表

5, -5, 1, 10, 2, -11, -3, 1, 6, -2, -6, 4, 1, -7, 7, -7, -1, 0, -2, 5, 1, -3, 5, 15, 6, -8, -5, 3, 9, 3, 3, -3, 3, -3, 1, 3, 0, 6, -3, -14, 7, -7, -10, -2, -2, -8, 11, 4, 0, 3, 7, -1, -5, 0, 8, -7, 4, -2, 0, -7, 0, -5, -9, 9, 8, -1, -4, 0, 4, -4, 2, 3, 1, -4, -5, -7, 5, -3, 4, -10, 1, 4, -3, -3, 10, -6, 9, 1, -4, 0, -6, -7, 1, -1, -2, -2, -3, -2, 6, 10, 7, -6, 3, 2, 4, -3, 7, 6, 0, 6, -4, 0, 9, 6, -1, 7, 2, -2, 6, -8, 0, -11, 7, 6, -1, -11, -3, -10, 2, -7, -1, -6, 7, -1, -6, 5, -4, -6, -6, -1, 6, -3, 0, -2, -3, -4, 1, 2, 8, -10, 1, 1, 10, -1, 15, -7, -1, -2, 2, -2, -3, 6, -1, 11, -2, 4, 7, -5, 4, 1, 1, -2, -2, -9, -6, -10, 6, 6, 1, -7, 0, 7, 7, -3, 6, 5, 1, -9, 6, -4, 8, 2, 9, 4, -5, -1, 5, 11, 4, -8, 0, 1, -7, 4, 1, -2, 6, 6, 2, 6, 1, 3, 0, -1, 2, 9, 1, 5, 8, 15, -1, 3, 3, -6, 0, -4, 6, 4, 0, -1, -5, -2, -5, -5, -2, -13, 6, -8, 0, 14, -1, 2, 7, -1, 4, 5, -6, -2, 0, -3, -6, -10, 5, -9, -6, -6, -8, -4, 3, -5, -7, -4, -4, 0, 6, -1, 9, -4, -1, 4, 5, 1, -7, -1, 0, -14, 7, -5, 0, 0, 11, -1, 7, -6, -4, 2, -1, 5, 0, 3, 6, -8, 7, 1, -1, -13, -6, -13, -1, -2, -8, -6, -1, -8, 1, 0, -1, 4, 6, -8, 0, 0, 4, 5, -1, -4, -5, -12, 5, -8, -5, 1, 0, -5, -6, 3, 5, -2, 3, -11, 0, -5, 8, 1, -1, 0, 0, -3, 2, 4, 2, -6, 9, -9, 6, -8, -1, 6, 11, -4, -2, 4, -7, -2, 12, -6, 1, -1, -1, -6, -6, -3, -4, 0, 1, 5, 3, 3, 2, 2, 3, 6, 2, -2, 7, 5, 8, -5, 2, 0, 2, 2, 0, -13, -3, -10, 7, -4, 10, 12, 0, -5, -7, -3, 0, -6, -5, 7, 6, -5, 1, 1, 1, -12, -1, 2, 0, -8, -5, -6, 1, -1, -1, 1, 0, -5, -2, 5, 0, -2, -1, -7, 0, -5, 0, 3, -1, 7, -4, 4, 1, 1, 9, -8, 3, 7, 6, 9, 0, -10, -5, -13, 6, -4, 2, 2, -4, 3, 0, -9, 0, 1, 11, 5, 4, 5, 2, 0, 0, -6, 4, -4, 1, -1, -1, 9, 6, -6, 4, -10, 1, -6, 3, -6, 1, -8, 5, -3, -1, 3, -6, 8, -5, -7, -4, -10, -4, 9, -3, -2, -1, 8, 0, 3, 0, 10, 2, -7, 0, -10, -6, -9, 2, -6, 8, -8, 0, 0, 6, -1, 10, 0, 0, 6, 5, -6, 5, -6, 4, -6, 2, -2, 7, 5, 2, -14, 7, -3, -3, -2, -2, 1, 2, -4, 2, -7, 2, 5, 4, -8, -1, -2, 9, -3, 6, 1, -1, -10, -7, -8, -2, 0, 7, 2, 1, -11, -2, -5, 1, -12, 1, 3, 3, 7, 1, -8, -2, -7, 1, -11, 7, -6, -1, -1, 6, 0, 3, 0, 1, -9, 8, -2, 5, -7, 1, 1, 6, -3, -9, -2, -1, 7, 7, 7, 10, 2, 5, 4, -5, -8, -6, -4, 7, -3, 0, -8, 5, -8, -3, -5, 0, 1, 14, 7, -1, 2, -2, -10, 2, 0, 2, -1, 4, -5, -5, -9, 6, -3, 12, 4, -2, 1, -5, -7, 7, 0, 6, 9, 7, -3, 3, 8, 4, -6, 0, 2, 4, -7, 3, 0, 2, -5, 4, 0, 0, -7, 3, 0, -4, 0, -6, 1, 2, -2, -3, -7, -1, -3, 8, -3, -1, -5, 2, -3, 0, 3, 7, 6, 0, -5, 11, 6, -3, -1, -7, -6, 5, -5, 6, -7, 11, -1, 0, -5, 9, -2, 14, 6, 0, 8, 2, -9, -1, 4, 1, 4, 5, -10, -1, 2

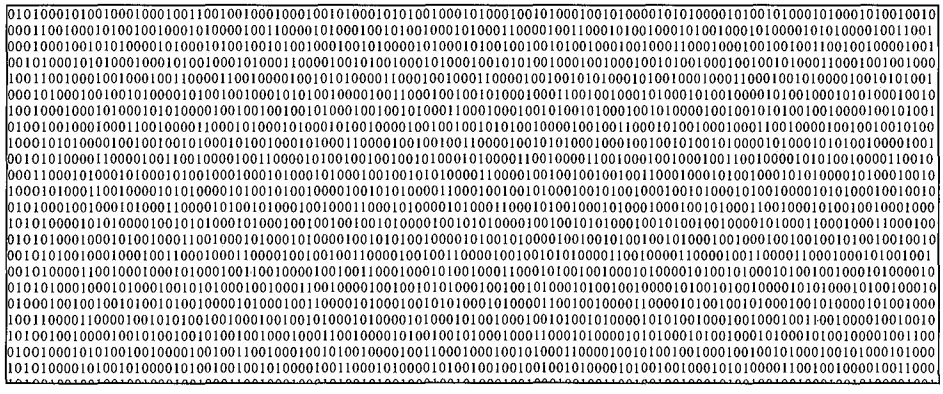
將表 2 的值轉成二進位後，利用(2,3)-threshold 視覺式機密分享機制，將二進位的差值表分解成 3 份分享訊息，圖 3中列出部份分享訊息的位元。圖 4為隱藏了分享訊息的偽裝影像以及 PSNR 值，不論從影像的外觀或是 PSNR 值，都可以說明人眼是無法察覺出影像有更改過的痕跡，因此在網路上傳送是不致引人懷疑的。



(a) 分享訊息 1

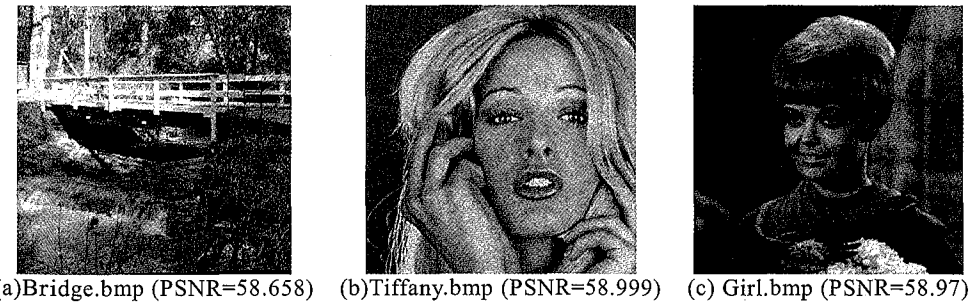


(b) 分享訊息 2



(c) 分享訊息 3

圖 3：三份分享訊息的前 3000 個位元



(a)Bridge.bmp (PSNR=58.658) (b)Tiffany.bmp (PSNR=58.999) (c) Girl.bmp (PSNR=58.97)

圖 4：三張 256 × 256 pixels 的灰階偽裝影像及 PSNR 值

二、實驗二：中英文夾雜之秘密文件

在這個實驗中，秘密訊息內含中文和英文，並且使用與實驗一相同的欺敵訊息，如表 3所示，以驗證我們的方法可以讓不同的秘密訊息，來共用相同的欺敵訊息，並且仍然以圖 2做為隱藏分享訊息的三張灰階影像。

表3：中文訊息與中英文夾雜之秘密訊息

欺敵訊息	秘密訊息
飼養動物保護法公告禁止飼養的美洲巨水鼠，將遭強制沒收全部銷毀，並罰五萬元。	法務部密令：從 7 月 30 開始展開代號"Golden Digging"的肅貪行動。

秘密訊息轉換成十六進位 Big5 碼後，表示如下：

秘密訊息：

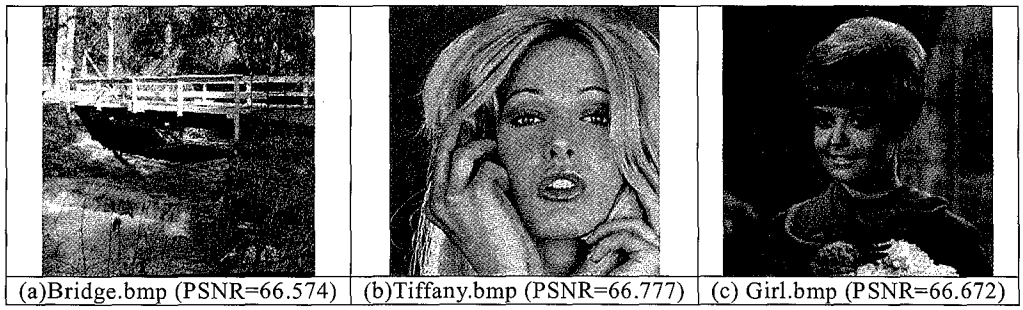
AA 6B B0 C8 B3 A1 B1 4B A5 4F A1 47 B1 71 37 A4 EB 33 30 B6 7D A9 6C AE 69 B6 7D A5 4E B8 B9 22 47 6F 6C 64 65 6E 20 44 69 67 67 69 6E 67 22 AA BA B5 C2 B3 67 A6 E6 B0 CA A1 43

欺敵訊息內碼和秘密訊息內碼的差值表則如表 4所示：

表4：雙語秘密文件與欺敵文件之間的差值表

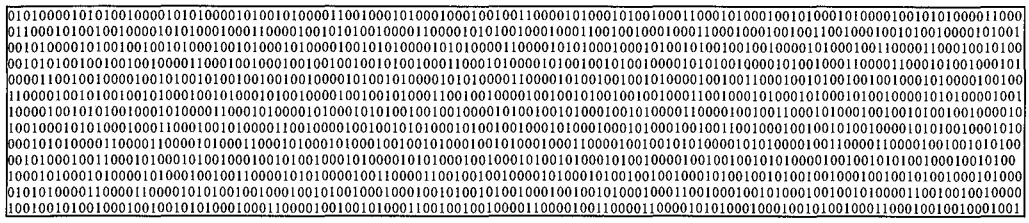
5, 0, -8, 7, 2, -13, 2, 0, 6, -4, -2, -9, 2, -10, 4, 1, -1, -7, -10, 11, 0, -5, 3, 7, 6, -11, -3, -4, 2, 6, 0, 0, 7, 3, -7, -9, -7, -6, -3, -8, 4, 2, -4, -1, -8, 1, 6, 4, -4, -1, 7, -5, -8, 2, 5, -9, -2, 6, 1, -7, 1, -1, -12, -1, 0, 3, -4, 4, -1, 3, 2, -7, -4, -7, -6, 3, -3, -8, -4, -6, -3, -1, -4, 4, 6, -3, 5, -3, -8, 10, -5, -1, -8, -5, 2, 1, -2, 2, 10, 1, 7, -5, 7, -7, 0, -2, 3, 6, 3, 0, 0, -9, 11, 1, -1, -4, -4, -11,

我們將表 4的值轉換成二進位，並且利用(2, 3)-thershold 視覺式機密分享機制，將二進位的值分解成三份分享訊息，並且分別藏入圖 2的三張灰階影像當中，結果如圖 5所示，每張偽裝影像的 PSNR 值都很高，而且從外觀上看起來，皆不致引人懷疑。圖 6則列出分享訊息的所有位元。

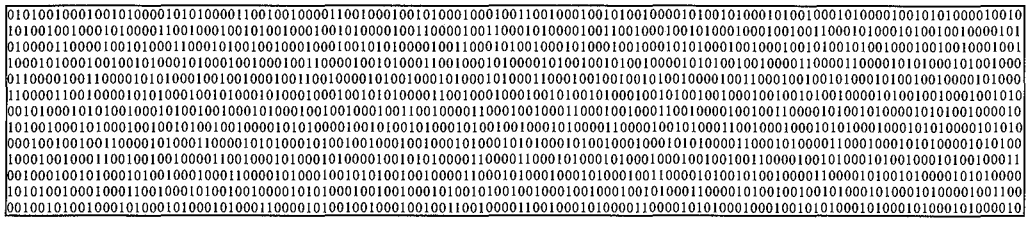


(a)Bridge.bmp (PSNR=66.574) (b)Tiffany.bmp (PSNR=66.777) (c) Girl.bmp (PSNR=66.672)

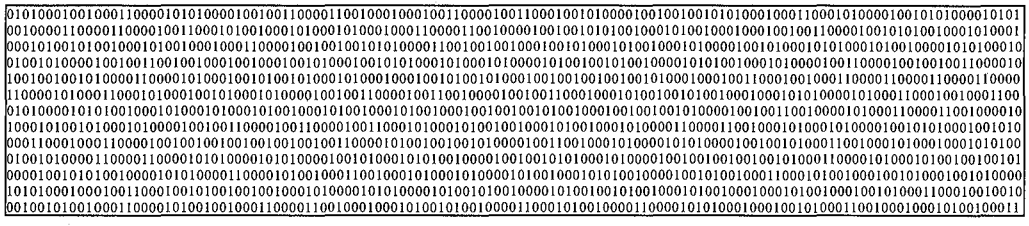
圖5：三張 256 × 256 pixels 的灰階偽裝影像及 PSNR 值



(a) 分享訊息 1



(b) 分享訊息 2



(c) 分享訊息 3

圖6：三份分享訊息的全部位元

三、討論

由表 5的結果我們可以發現，在欺敵訊息的內碼中'2'這個內碼從未出現過，但是不論是在實驗一或實驗二秘密訊息中的內碼，'2'都不只出現一次，因此若用 Yeh & Hwang 等人的作法，就必須再另外找尋一段欺敵的文字，或者是再加長原來的欺敵文字，直到有內碼'2'出現為止。雖然這也是一種解決辦法，但是不能隨意取用任何文字做為欺敵訊息，會使得我們必須一直更換欺敵訊息，而無法讓一些秘密訊息來共用同一段欺敵訊息。尤其我們並無法由中文字直接看出是否包含所有需要的內碼，有可能會試了好幾次之後，才能找到適合的欺敵訊息；又或者換過的欺敵訊息雖然包含'2'這個內碼，可是卻又少了別的內碼。因此不能取用任何文字做為欺敵訊息，事實上是 Yeh & Hwang 等人的方法中一個嚴重的缺點，但是在本研究中我們是計算兩份文件之間的內碼差值，任何內碼字元皆可做為計算的基準，因此我們的研究解決了這個嚴重的問題。

實驗一中的秘密訊息的文字比欺敵訊息多，因此轉換後的每個內碼字元出現次數也會增加，由表 5中我們也可以看出，實驗一的秘密訊息內碼的出現次數，全部都比欺敵訊息內碼的出現次數來得多，如果使用 Yeh & Hwang 的方法，在位置索引表中，相同的位置索引值勢必會出現許多次，因此便有露出蛛絲馬跡的可能。就算在實驗二中，秘密訊息的內碼個數少於欺敵訊息內碼個數的情況下，也有可能發生某些內碼在秘密訊息中出現的次數，比在欺敵訊息中出現的次數來得多。例如內碼'6'在實驗二的秘密訊息中出現 19 次，可是在欺敵訊息中卻只有 6 次。

表5：十六進位碼在欺敵訊息和秘密訊息中的出現次數

十六進位碼	在欺敵訊息中的次數	在實驗一之秘密訊息中的次數	在實驗二之秘密訊息中的次數
0	4	20	4
1	8	35	6
2	0	12	6
3	4	26	7
4	17	61	10
5	8	49	4
6	6	44	19
7	6	51	10
8	7	20	2
9	6	21	5
A	33	62	15
B	24	86	15
C	7	55	5
D	4	25	2
E	10	17	6
F	4	26	2
字元總數	148	610	118

由於英文的字集(Character Set) 有限，所以如 ASCII 這種單位元組(Byte)的編碼(Encoding) 系統，已可包含所有的英文字母加上一些常用符號。但是中文的字集非常的大，單位元組的編碼系統已不敷使用，必須使用雙位元組的編碼系統。Big5 編碼法即屬於一種雙位元組字集的編碼系統，在這個系統中亦保留了 21~7E 的範圍給 ASCII 編碼法所能處理的字元(鄭褚璋譯，Ken Lunde 著，2002)。因此我們將中英文夾雜的秘密訊息轉換成十六進位的 Big5 碼，就可以直接處理中英文夾雜的訊息，不需要如 Wang & Lu 的方法一樣，需要特定尋找一段雙語的欺敵訊息，來分別處理中文和英文的秘密訊息。雖然在我們的實驗中，只顯示了以中文欺敵訊息掩護雙語秘密訊息的傳送，但是本研究亦可應用在以雙語欺敵訊息掩護雙語秘密訊息的傳送。

BCDT 是還原秘密訊息的重要關鍵，因此我們使用(k, n)-threshold 的視覺式機密分享機制，將 BCDT 分成 n 份，給 n 個不同的人保管，除了可以減低差值表遺失的風險外，也可以達到控管的功能，防止機密被其中一個人洩露。我們將分享訊息隱藏在有

具體內容的影像中，亦可避免引起駭客的懷疑。在 Lin & Lee 等人的作法中，他們的秘密文件索引表都只有一份，遺失之後便無法還原原來的秘密訊息，而且也無法達到機密控管的效果。更何況 Lin & Lee 和 Yeh & Hwang 都是利用 IDEA 將索引表加密後再傳送出去，這種經過加密後沒有具體內容的檔案，更容易引起駭客的懷疑，而加以攔截。

在本研究中欺敵文件的作用只是被當作計算 CDT 的參考基準，因此並不需要包含所有秘密訊息的字元，因此它的字數可以少於秘密文件的字數。當欺敵文件的字數小到一定程度時，事實上我們未必需要欺敵文件。例如我們可以任選一個 32 位元長度的 *seed* 值，並且將這個 *seed* 分割成 8 個單元，每個單元是一個 4 位元長度的數值，這 8 個數值就扮演著欺敵文件內碼的角色。而秘密文件的內碼便隨機與這 8 個數值進行差值的計算，因為這個被當做計算差值表基準的 *seed* 值很短，因此也可以直接將它隱藏在分享影像中，就不需要欺敵文件，以減少資訊的傳遞量。

陸、安全性與效能分析

一、安全性

本研究將 BCDT 分解成 n 份，分別藏入 n 張影像的最低位元平面中，由於是藏入最低位元平面中，因此外觀上幾乎是看不出來有任何改變。還原原來的 BCDT，只需要其中 k 張影像便可，因此若在傳送過程中任一張偽裝影像遺失了，仍然可以得出原來的機密影像。換句話說，若被駭客偷走一張影像，縱使他知道訊息是藏在影像的最低位元平面，他也是無法還原出原來的訊息，因為在不足 k 張的分享訊息上，出現位元'0'與位元'1'的比例是相等的，因此駭客是無法由其中猜測出原來的機密訊息。由於資訊隱藏的精神在於，隱藏機密的偽裝影像無法由人眼看出外觀上的差別，因此不致引起駭客的懷疑而遭攔截，雖然 Wang & Lu 所使用的資訊隱藏技術具有較好的安全性，但是 LSB 方法對影像的破壞程度是最小的，只要我們所選擇的掩蓋影像夠普通，本研究的方法即使在不進行秘密分享的情況下，仍然能夠達成欺敵的目的。

如果所有的影像都被駭客偷走，那麼他也必須知道我們的分享機制，這樣他才能決定判別機密訊息的位元是'0'或'1'的門檻值 d，以及位元擴展的倍數 m。因為我們所分享的是一份文件，而非影像，所以無法靠視覺解密。而且文字必須要精確地還原出原來的'0'與'1'，才能得到正確的訊息，任何一個位元的錯誤，就無法得到正確的訊息。

如果駭客知道我們的分享機制，而得出正確的內碼差值表，那他也需要得知原先用來隨機重排的 *seed*。但是我們的 *seed* 已被分解成 n 個 *seed* 值，而且我們的 *seed* 值同樣具有 (k, n)-threshold 視覺式機密分享機制的安全性，如果駭客所取得的 *seed* 值不足 k 個，則仍然無法還原原來的 *seed* 值。同時，我們將二進位差值表與 *seed* 值分別以 (k, n)-threshold 視覺式機密分享機制分解成 n 份，再分開傳送，而不將兩者先合併一起分解再傳送，也可大幅降低差值表與 *seed* 值同時被截取到的機率。

另外，為了能夠徹底達到欺敵的效果，我們可以傳送許多份欺敵文件給對方，來混淆駭客的視聽，而真正的欺敵文件的檔名可以接在秘密訊息之後，再將秘密訊息加上欺敵文件的檔名一起轉成 Big5 碼後，產生 BCDT，並藏入影像檔中。等所有的欺敵文件傳送完之後，再傳送已隱藏了差值表的影像，但此時沒有欺敵文件的配合，就算被駭客攔截了檔案，也無法得知其中的秘密。而且欺敵文件有非常多份，駭客也必須能取得我們的 *seed* 值，才有可能知道真正的欺敵文件的檔案名稱。

若差值表被駭客取得，而駭客在無從得知 *seed* 值的情形下，秘密訊息被還原的機率為 $1/q^p$，其中 p 代表秘密文件十六進位碼的字元個數，q 代表欺敵文件十六進位碼的字元個數，因此當我們欲傳送的秘密文件很大時，或是欺敵文件長度夠長的話，便可使秘密訊息被還原的機率變得非常小，因而增加了機密被碼解的困難度。

二、效能分析

我們將秘密文件的十六進位內碼以差值來代表，由於差值的可能範圍為-15 ~ +15，每一個差值便可以用 5 個位元來表示，因此一個差值的長度只比一個內碼的長度多 1 個位元，若 p 代表秘密文件十六進位碼的字元個數，則整個差值表的大小只有 $5 \times p$ 個位元，僅僅為秘密訊息的 1.25 倍，而 Yeh & Hwang (2001b)的秘密文字索引檔為秘密訊息的四倍，因此他們的加密演算法必須多一道壓縮的步驟，來縮小秘密文字索引檔的大小。

二進位的差值表利用(k, n)-thershold 視覺式秘密分享機制分解成 n 份，每一份的大小為 $5 \times p \times m$ 個位元，分別隱藏在 n 張掩蓋影像的最低位元平面上。以一張 256 × 256 的灰階影像為例，一個位元平面便有 65536 個位元，即使扣除記錄每份長度的前 32 個位元，仍有 65504 個位元可以利用，因此這 n 張掩蓋影像便可隱藏 $13100/m$ 個秘密文件的內碼字元，代表 $3275/m$ 個中文字，假設 $m = 3$，則只需利用一個位元平面便可隱藏 1091 個中文字。而一張 256 × 256 的灰階影像的檔案大小約為 65KB，因此通信成本並不高。

柒、結論

本研究提出一個利用有具體內容的訊息，來傳遞中文秘密訊息的方法，我們的方法可以改進 Lin & Lee 等人的缺點，可以隨意取用任何文字做為欺敵訊息。除了使用文字做為欺敵的文件之外，也可以使用圖形當做欺敵文件。另外，我們將訊息藏入影像中，使得傳送出去的都是明圖和明文，因此可以降低引人懷疑的可能性，而且網路上傳送的資料量非常大，駭客也無法判斷哪些影像是帶有機密訊息的。除此之外，我們的訊息也是經過機密分享的機制，藏入不同的圖中，因此可以減低機密被偷取的風險，也可以達到機密控管的效果，而這是 Lin & Lee 等人所無法達到的。

就安全性而言，在我們的作法中，將差值表以(k, n)-threshold 視覺式機密分享機制，分解成 n 份，因此可以達到(k, n)-threshold 視覺式分享機制所擁有的安全性，雖然這個機制原先是用在以視覺來解密的方式，但是我們利用這個機制所設定的門檻值，便可以將它應用在文件的分享與還原上。另外 CDT 在建立的過程中亦經過隨機配對，而隨機配對的 *seed* 也利用(k, n)-threshold 視覺式機密分享機制分解成 n 個 *seed* 值，因此也可以防止 *seed* 值被他人取得，所以我們的方法是有多重保護及安全性的。

本研究是以傳送中英文雙語的秘密文件為主，未來我們將加入特殊碼的觀念，使得本研究可以延伸至多國語言文件的傳送。另外，本研究提出的是單一機密的分享機制，未來我們將擴展至多機密的分享機制。

參考文獻

1. 黃景彰，2001，資訊安全——電子商務之基礎，台北：華泰文化事業股份有限公司。
2. 鄭褚璋譯，Ken Lunde 著，2002，中日韓越資訊處理，台北，美商歐萊禮股份有限公司台灣分公司。
3. Lin, C.H. and Lee, T.C. "A Confused Document Encrypting Scheme and Its Implementation," Computers & Security (17:6) 1998, pp:543-551
4. Naor, M. and Shamir, A. "Visual Cryptography," Advances in Cryptology —— Eurocrypt '94, Lecture Notes in Computer Science (950) 1994, pp:1-12
5. Neil, F.J. and Sushil, J. "Exploring Steganography: Seeing the Unseen," IEEE computer (31:2) 1998, pp:26-34
6. Wang, S.J. and Lu, C.K. "A Scheme of Non-sensible Document in Transit with Secret Hiding," Journal of Information Management (9:2) 2003, pp:169-182
7. Wang, S.J. and Yang, K.S. "A Scheme of High Capacity Embedding on Image Data Using Modulo Mechanism," The Second International Workshop on Information Security Applications (WISA), Korea, Sep., 2001, pp:299-309
8. Yeh, W.H. and Hwang, J.J. "Hiding Digital Information Using a Novel System Scheme," Computers & Security (20:6) 2001a, pp:533-538
9. Yeh, W.H. and Hwang, J.J. "A Scheme of Hiding Secret Chinese Information in Confused Documents," Journal of Information Management (7:2) 2001b, pp:183-191