驗證影像之完整性--三維偵測法

簡永信(Yung-Hsin Chien)
靜宜大學資訊管理學系
g9134045@pu.edu.tw

王孝熙(Hsiao-Hsi Wang)
靜宜大學資訊管理學系
hhwang@pu.edu.tw

摘要

由於電腦技術的進步，我們可以將數位化之後的影像以圖形點(pixel)的方式來儲存，但存在許多缺點，因此如何保護重要資料免於被複製、竄改、偷窺和偽造是一個非常重要亟待克服的問題。為了證明影像或圖形是否為合法的擁有與使用，如何驗證影像(image authentication)及圖形的完整性(integration)便成為主要研究的課題，因此我們的研究的主要重點在於如何驗證該張影像或圖形的完整性以及精確地找出被非法竄改的圖形點。在本論文中我們將針對已發表的技術及方法做進一步了解，藉此提出我們的方法—三維偵測法，利用 RSA 的加密機制將圖形最重要位元所構成的區塊(MSB)透過秘密金鑰加密轉為數位簽章後，放入最不重要位元(LSB)所構成的區塊中，藉由公開金鑰來驗證圖形完整與否，並透過三維偵測法的運算後，偵測出非法竄改點的位置，經由我們的分析與實驗結果，更證明即使在被竄改一個圖形點的情況下，依舊能找出非法竄改點的位置。

關鍵詞：驗證影像、三維偵測法、RSA

一、前言

近年來，由於網際網路的蓬勃發展，使得電子購物、電子文件交換、線上廣播、隨選視訊等已被廣泛的運用在許多企業。雖然說相關技術的發展，將可為公司企業及其顧客帶來相當大的便利性，但在此一同時也衍生出許多資訊安全上的問題，例如冒名交易、竊取、竄改等。當中有許多問題可以運用密碼技術來加以解決，但是關於有價媒體(如影像、聲音、影片等)在網路上傳輸與交流，其智慧財產所有權之認證及驗證問題則仍有待克服。而在之前的研究中，有利用浮水印的技術來偵測出圖形被非法竄改的位置 [3,4,5,6,7,12,13,14]。但是所有提出來的方法如需顧慮安全性的話最多只能偵測出 172 個圖形點的範圍內被竄改與否，而無法確切的告知是哪一點有被竄改到。而 Tang，Hwang 及 Yang[11]進而提出利用數位簽章(digital signature)的方法運用在偵測圖形被非法竄改的位置，一開始他們所提出的方法最多只能偵測出 86 個圖形點的範圍內被竄改與否，後來他們又提出縱橫偵測法[1]使偵測出的圖形點範圍可以縮小至 1 個 pixel，但某些情況下仍然沒有辦法偵測出小至一個 pixel，因此我們提出一個新的方法-三維偵測法以做改進。

首先，以 RSA 的加密機制將經過雜湊處理的最重要位元構成的區塊(most significant bits(MSB)) 以秘密金鑰(private key)加密後得到數位簽章(digital signature)，然後放進圖形的最不重要位元(least significant bits (LSB))以做為隱藏數位簽章的地方，並藉由公開金鑰(public key)驗證影像或圖形的完整性，如果不具完整性，我們再透過影像的分割區塊化，將影像或圖形以三維偵測法運算，其結果可在 1 個 pixel 範圍內偵測出何處被竄改。

如果有人需要驗證該張影像圖形之合法擁有權與使用權、完整性的確認時，即可利用公開金鑰(public key)取出數位簽章(digital signature)[9]，並利用影像區塊的三維交集運算得出是否有被非法竄改，藉此驗證該張影像圖形的完整性。以下會詳細介紹我們的方法並提出分析和實驗結果來驗證我們的方法，既使在最壞狀況下，我們依然可以精確的找出非法竄改點的位置。

二、三維偵測法

我們所提的方法是為了改進縱橫偵測法所無法偵測至 1 個 pixel 的某些情況做進行研究，首先我們先要定義一下我們所預設的一些符號及設定狀況，以利我們描述及解說論文。

MSB1	*MSB2*	*MSB3*	*MSB4*	*MSB5*	*LSB1*	*LSB2*	*LSB3*

圖一

1.假設 I_{pq} 是一個 $p \times q$ 尺寸大小的灰階影像。

2.假設 I_{pq} 內的 *MSB*=5 位元/圖形點，如圖一的 *MSB1*~*MSB5*。

3.假設 *LSB*= *LSB1*+ *LSB2*+ *LSB3*，其中 *LSB1*=1 位元/圖形點，*LSB2*=1 位元/圖形點，*LSB3*=1 位元/圖形點，如圖一的 *LSB1*~*LSB3*。

4.每個區塊被雜湊函數處理過後須是 512 個位元，因為 RSA 加密系統金匙長度需達到 512 位元才會安全之限制[10]。

根據我們定義的三項假設後，我們的方法必須進行下列 10 個步驟來完成。首先，我們

先對 10 個步驟作個簡單的介紹:

1. 橫向第一個區塊的建立:如圖二的步驟一至三，建立橫向第一個區塊。
2. 橫向第二個以後的區塊的建立:如圖二的步驟三，繼續橫向區塊的建立。
3. 縱向第一個區塊的建立:如圖二的步驟四，建立縱向第一個區塊。
4. 縱向第二個以後區塊的建立: 如圖二的步驟四，繼續縱向區塊的建立。
5. 斜向第一個區塊的建立:如圖二的步驟五，建立斜向的第一個區塊。
6. 斜向第二個以後的區塊的建立: 如圖二的步驟五，繼續斜向區塊的建立。
7. 縱、橫、斜向區塊的雜湊處理:如圖二的步驟六，對三方向的區塊做雜湊處理。
8. 縱橫斜向區塊的數位簽章及隱藏處理:如圖二的步驟七及八，將雜湊處理過的區塊加密，放入 *LSB* 的區塊中，藉此我們可以得到加密過的影像。
9. 從縱、橫、斜向區塊內取出數位簽章:如圖三，同上述所提的步驟一至六，將雜湊處理過的區塊中取出數位簽章並解密。
10. 影像區塊的三維交集運算法:如圖三，透過比較區塊與數位簽章的步驟，符合即表示其圖形完整，並輸出空白的影像圖形；不符合便透過影像區塊的三維交集運算，找出非法被竄改的位置圖。

2.1 橫向第一個區塊的建立

首先我們將每個圖形點依據前面假設的第 2、3 點先行分離每個圖形點的 *MSB* 及 *LSB*，其步驟如圖二"將每個圖形點分離出 MSB, LSB_1, LSB_2, LSB_3"，之後我們將整張影像圖形所有 LSB_1, LSB_2, LSB_3 區塊值清除為 0，其步驟如圖二"將所有 *LSB* 值都設為 0"，然後以影像圖形的左上角第一個圖形點起算 512 點的 *MSB* 集合成為一個區塊(BR_1)，而第一個區塊是由 2 個較小區塊 br_1、br_2 (256 點)組成。而選擇 512 點構成一個區塊是為了要符合 RSA 加密系統金匙長度需達到 512 位元才會安全之限制。而橫向第一個區塊結構如圖四所示。其處理步驟如圖二的"橫向區塊建立(BR_X)"。

2.2 橫向第二個以後的區塊的建立

在我們第一個區塊建立以後，我們可以開始建立第二區塊至後的區塊，與第一個區塊不同的是第二區塊以後的區塊都需含前一區塊的較小區塊(256 個圖形點)，因此一個區塊是由三個 *MSB* 組成的較小區塊組成，以 BR_2 為例是由 br_1, br_2, br_3 組成，一個影像圖形總共可以製作出 BR_x，$x = (p \times q)/256$。其結構如圖四所示。其處理步驟如圖二的"橫向區塊的建立 (BR_X)"。但是每一張圖最後都會有不夠集合成一個大區塊的現象，因此我們只要將這些不夠的部份全部以 1 虛擬取代之，以讓其符合規則。

2.3 縱向第一個區塊的建立

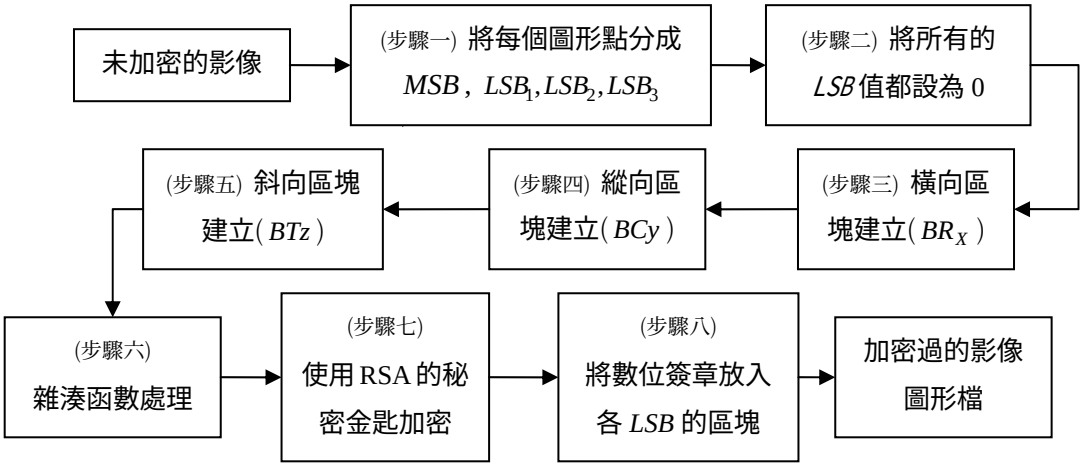
此步驟我們先以影像圖形的左上角第一個圖形點橫列起算 32、64、128、256 列為橫列標準，而直列也是以第一個圖形點為準起算 16、8、4、2 行為直行標準，但是橫列、直行各要選擇多少呢？這個問題就要看偵測目標的要求，如果偵測目標定在 1 個圖形點的話我們在橫列設為 256 列，直行為 2 行，因此行、列的設定完全依據偵測目標配合前面所提配對搭配。而整個縱向直行之 BC_1 是依據橫列之 *MSB* 與直行之 *MSB* 取其交集之部分為之。如果以橫列 256 列，直行 2 行為例，我們可得知兩個直行分別為 bc_1, bc_2。如果以橫列設為 128 列，直行為 4 行，則 bc_1, bc_2 各代表 2 行。因此 bc_1, bc_2 各代表多少行，完全視偵測目標而定。至於有多少個 BC_y 呢?我們可以得到 $y = (p \times q)/512$ 而整個縱橫對照比例是完全為了要符合 RSA 加密系統金匙長度需達到 512 位元才算安全之限制[2]。以此例算 256X2=512 剛好是 512 位元，因為 LSB_2 為 1 個位元。其結構如圖五所示。其處理步驟如圖二的"縱向區塊的建立(BC_y)"。

2.4 縱向第二個以後區塊的建立

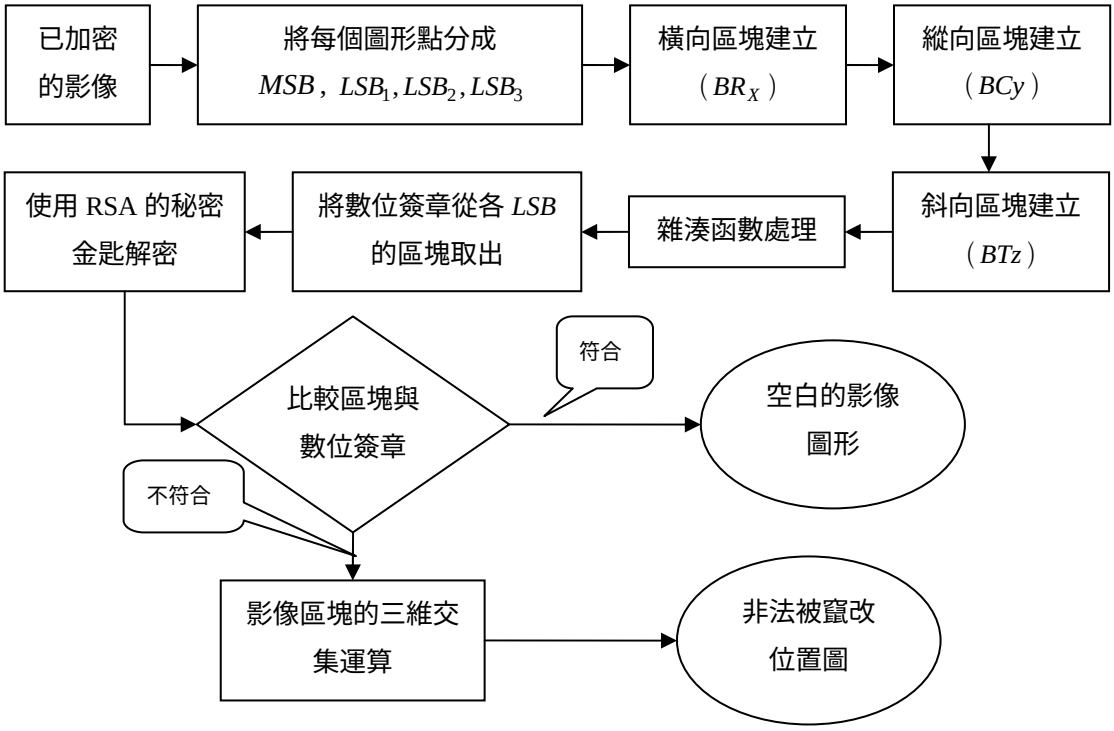
在我們第一個縱向區塊建立以後，我們可以開始建立第二區塊至後的區塊，與第一個區塊不同的是第二區塊以後的區塊都需含前一區塊二分之一的直行區塊，因此一個區塊是由三個直行 *MSB* 小區塊組成，以 BC_2 為例是由 bc_1, bc_2, bc_3 組成，一個影像圖形總共可以製作出 BC_y，$y = (p \times q)/512$。其結構如圖五所示。其處理步驟如圖二的"縱向區塊的建立 (BC_y)"。但是每一張圖最後都會有不夠集合成一個大區塊的現象，因此我們只要將這些不夠的部份全部以 1 虛擬取代之，以讓其符合規則。

2.5 斜向第一個區塊的建立

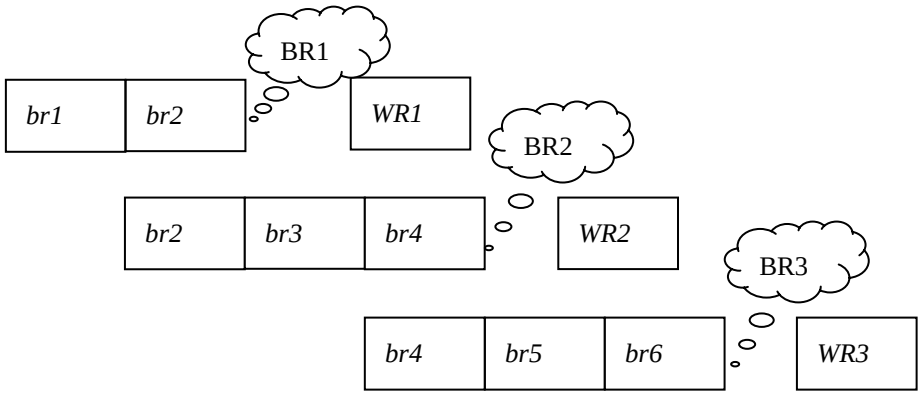
然後以影像圖形的左上角第一個圖形點起算依對角線斜下 512 點的 *MSB* 集合成為一個區塊(BT_z)，而第一個區塊是由 2 個較小區塊 bt_1、bt_2 (256 點)組成。而選擇 512 點構成一個區塊是為了要符合 RSA 加密系統金匙長度需達到 512 位元才會安全之限制。而斜向第一個區塊結構如圖六所示。其處理步驟如圖二的"斜向區塊的建立 (BT_z)"。



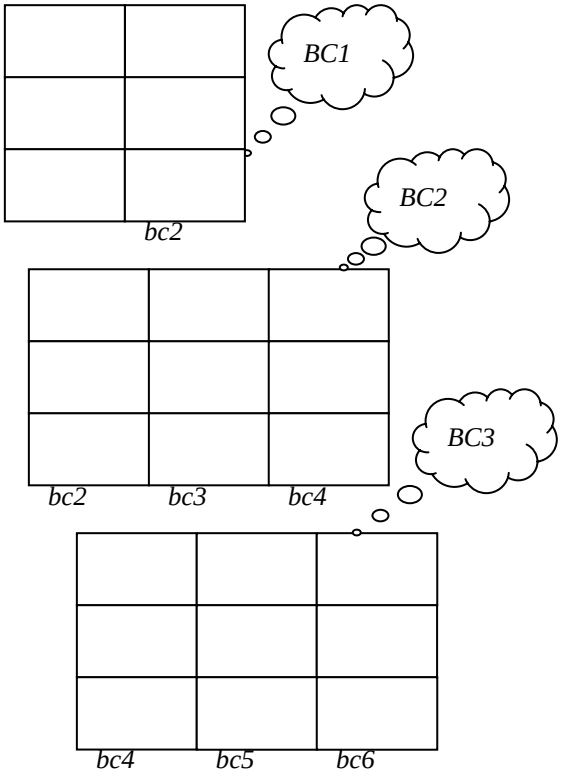
圖二：影像圖形加密步驟



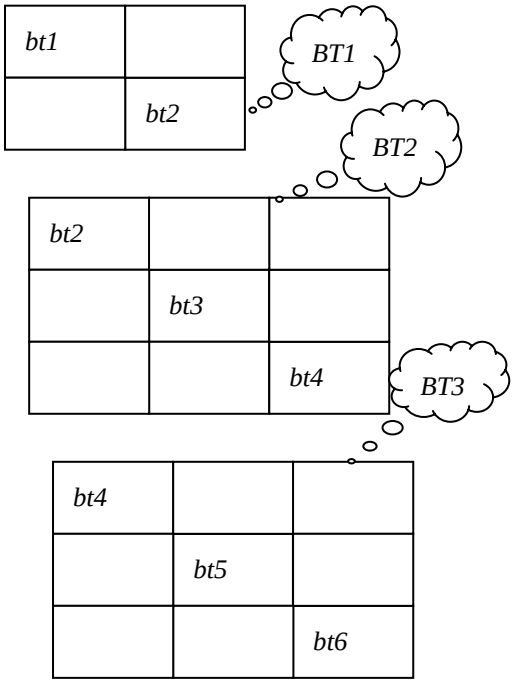
圖三：影像圖形之解密及竄改點偵測步驟



圖四：橫向區塊 BR_x 結構圖



圖五:縱向區塊 BC_y 結構圖

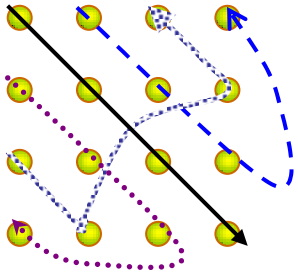


圖六:斜向區塊 BT_z 結構圖

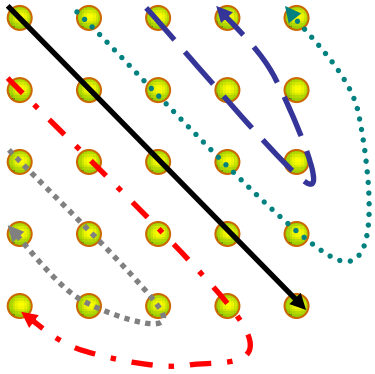
2.6 斜向第二個以後的區塊的建立

在我們第一個區塊建立以後，我們可以開始建立第二區塊至後的區塊，與第一個區塊不同的是第二區塊以後的區塊都需含前一區塊的較小區塊(256 個圖形點)，因此一個區塊是由三個 *MSB* 組成的較小區塊組成，以 BT_2 為例是由 bt_1, bt_2, bt_3 組成，一個影像圖形總共可以製作出 $BT_z, z = (p \times q)/256$。其結構如圖六所示。其處理步驟如圖二的"斜向區塊的建立 (BT_z)"。但是每一張圖最後都會有不夠集合成一個大區塊的現象，因此我們只要將這些不夠的部份全部以 1 虛擬取代之，以讓其符合規則。針對斜向區塊的建立，我們做了一些補充，因為斜向區塊不像縱橫區塊擁有固定的 pixel 個數，例如：一個 4X4 的區塊，第一列有四個 pixel，第一行也有四個 pixel，相對的第二、三、四列或行也都有四個 pixel，但是斜向部分有大至四個 pixel，也有小至一個 pixel，如此便造成程式的難以撰寫，因此我們想出可以用下列的方式來方便程式撰寫。

(1) 當 p,q 為偶數時，偵測的方式如下圖，每條線所連接的皆為四個 pixel，如此便不會有小至一個 pixel 的問題，而且偵測次數與縱向、橫向同為四次。



(2) 當 p,q 為奇數時，偵測的方式如下圖，每條線所連接的皆為五個 pixel，如此便不會有小至一個 pixel 的問題，而且偵測次數與縱向、橫向同為五次。



2.7 縱、橫、斜向區塊的雜湊處理

雜湊處理是一種加密處理之一種程序，而我們這個處理程序主要是要將縱向、橫向區塊、斜向區塊(BC_y, BR_X, BTz)壓縮達到 512 位元其所參考的是 MD5[8]所提出之處理程序，而 512 位元主要是要達到 RSA 目前所謂安全等級。其公式如下：

$$H(BR_x, BC_y, BT_z) = HWR_x, HWC_y, HWT_z$$
$$HWR_x, HWC_y, HWT_z = 512\text{位元} \qquad (1)$$

其處理步驟如圖二的"雜湊函數處理"。

2.8 縱橫斜向區塊的數位簽章及隱藏處理

我們的影像圖形加密程序主要是以 RSA 加解密系統為中心主軸，因此 RSA 加密系統金匙長度需達到 512 位元才會安全之限制。因此我們在做影像圖形加密程序是利用之觀念。

$$WR_x, WC_y, WT_z = E_k(HWR_x, HWC_y, HWT_z)$$

$$k = \text{秘密金匙}$$

由以上式子我們得知 WR_x, WC_y, WT_z 是代表橫向區塊、縱向區塊的數位簽章及斜向區塊的數位簽章，其處理步驟如圖二的"使用 RSA 的秘密金匙加密"。完成所有縱、橫、斜向區塊的數位簽章後我們將這些數位簽章一一放入 LSB_1, LSB_2, LSB_3 構成的區塊中，以達成影像圖形的加密程序。其步驟如圖二的"將數位簽章放入各 *LSB* 的區塊中"。但是遇到 2.2、2.4、2.6 有使用虛擬位元構成的區塊時，我們只將該區塊的數位簽章前段符合實際存放的部分放入(WR_x, WC_y, WT_z)中。

2.9 從縱、橫、斜向區塊內取出數位簽章

從這章節起即屬於影像圖形的解密及非法竄改偵測，圖形擁有者證明，完整性與否，如不完整被竄改點是哪裡的所有運作程序。首先圖形擁有者要確認一張影像圖形的完整性時，它先將該張圖形的各圖形點依據我們的定義分離出 MSB, LSB_1, LSB_2, LSB_3，之後我們再將整張影像圖形的所有 LSB_1, LSB_2, LSB_3 區塊建構起來，其步驟如圖三的"將每個圖形點分離出 MSB, LSB_1, LSB_2, LSB_3"；然後以影像圖形的左上角第一個圖形點起算 256 個圖形點的 *MSB* 集合成為一個區塊(BR_1)，其步驟程序同 2.1 所提，第二個至後的橫向區塊的建立亦依據步驟程序同 2.2，以上兩個步驟如圖三的"橫向區塊建立(BR_x)"。其後步驟我們先以影像圖形的左上角第一個圖形點橫列起算 32、64、128、256 列為橫列標準，而直列也是以第一個圖形點為準起算 16、8、4、2 行為直行標準，至於該選橫列、直行該選多少，完全依據第一個區塊的選擇為之建立起 BC_1 其依據 2.3 所提步驟完全一樣，第一個直行區塊建立之後再依 2.4 所提步驟建立起第二個至後的直行區塊。以上兩個步驟如圖三的"縱向區塊建立(BC_y)"。 然後以影像圖形的左上角第一個圖形點起算 256 個圖形點的 *MSB* 集合成為一個區塊(BT_1)，其步驟程序同 2.5 所提，第二個至後的橫向區塊的建立亦依據步驟程序同 2.6，以上兩個步驟如圖三的"斜向區塊建立(BT_z)"。當 BR_x, BC_y, BT_z 建立之後，我們就將三種區塊進行雜湊處理其步驟如 2.5。經處理完後我們即可獲得 HBR_x, HBC_y, HBT_z，步驟如圖三的"雜湊函數處理"。完成以上步驟我們即可以將圖形的數位簽章取出與之比對，以辨別該圖形之圖形擁有者、完整性等之問題。其步驟如圖三的"將數位簽章從各 *LSB* 的區塊中取出"，而取出之數位簽章還是加密過的，因此我們還需利用一 RSA 之公開金匙將之解密。以獲得 $DHWR_x, DHWC_y, DHT_z$ 即原始經過雜湊處理過的橫向區塊與縱向區塊，其解密法如下：

$$DHWR_x, DHWC_y, DHWT_z = D_k(WR_x, WC_y, WT_z)$$

$$k = \text{公開金匙}$$

其步驟如圖三的"使用 RSA 的公開金匙解密"。但是附帶一提的是遇有使用虛擬位元構成的區塊時，我們將經過雜湊函數處理的影像圖形區塊以相等於被取出解密過後的數位簽章位元數與解密後的數位簽章比較。

2.10 影像區塊的三維交集運算法

在完成了 2.9 的所有步驟後，(縱行區塊以橫列 256 列，直行 2 行為假設前提)我們即可進行原始經過雜湊處理過的橫向區塊與縱向區塊與斜向區塊($DHWR_x, DHWC_y, DHWT_z$)與該圖之橫向區塊、縱向區塊與斜向區塊(HWR_x, HWC_y, HWT_z)作一比較運算以獲得非法竄改點的偵測。當我們以一對一的比對去發現那一區塊有不符合的即給予標示出來，以獲得初步非法竄改點的標示。如果全部區塊都是符合的我們即可宣稱其完整性是正確的，並輸出一空白影像圖形以表示沒有發現非法竄改點，如果有誤的話我們即進行影像區塊的三維交集運算，其步驟如圖三的"比較區塊與數位簽章"。其規則如下：

2.10.1 橫向區塊的交集運算

如果上述一一比對有發現橫向區塊有出現不符合現象即需進行該單元運算，其步驟如圖三的"影像區塊的三維交集運算"。以下我們將依圖四為說明架構。其運算規則如下：檢視該區塊的左右橫列區塊依據 2.10 的程序是否有符合的如果有則以不合的區塊減掉左邊和右邊有符合的區塊。

我們針對此一規則作一實例解說，假設圖四的第 513 個圖形點被非法竄改，即 $513/256 = 2.007$ 代表 br_3 有被非法竄改那麼將導致 $DHWR_2$ 出現錯誤，而與解密程序解出之 HWR_2 不相同，那麼執行此一程序我們查看一下 $DHWR_1$ 與 $DHWR_3$ 是否與 HWR_1 和 HWR_3 相同，只要其中有一符合即可依據規則運作假設 $DHWR_1$ 與 $DHWR_3$ 與 HWR_1 和 HWR_3 符合，之後我們進行 $DHWR_2$ - $DHWR_3$ - $DHWR_1$ 即可獲得 br_3，因此我們可獲得此區域有被非法竄改，但無法達到一個位元範圍的

偵測效應，還需下個步驟的處理方可得到。

2.10.2 縱向區塊的交集運算

如果上述一一比對有發現橫向區塊有出現不符合現象即需進行該單元運算，因為橫向區塊出現不符即會導致縱向區塊不符出現，其步驟如圖三的"影像區塊的三維交集運算"。以下我們將依圖五為說明架構。其運算規則如下：檢視該區塊的左右縱行區塊依據 2.10 的程序是否有符合的如果有則以不合的區塊減掉左邊和右邊有符合的區塊。

我們針對此一規則再依據 2.10.1 的結果繼續做運算，作一實例解說，第 513 個圖形點對應到的直行是 $513/2=256.5$ 即 BC_{257} 的第一個直行有誤，我們把它對應到圖五的 bc_3，即 BC_{256} 對應 BC_1、BC_{257} 對應 BC_2、BC_{258} 對應 BC_3、以方便作一說明，因此 bc_3 有被非法竄改那麼將導致 $DHWC_2$ 出現錯誤，而與解密程序解出之 HWC_2 不相同，那麼執行此一程序我們查看一下 $DHWC_1$ 與 $DHWC_3$ 是否與 HWC_1 和 HWC_3 相同只要其中有一符合即可依據規則運作假設 $DHWC_1$ 與 $DHWC_3$ 與 HWC_1 和 HWC_3 符合，之後我們進行 $DHWC_2 - DHWC_3 - DHWC_1$ 即可獲得 bc_3，因此我們可獲得此區域有被非法竄改，但無法達到一個位元範圍的偵測效應，還需下個步驟的處理方可得到。

2.10.3 斜向區塊的交集運算

如果上述一一比對有發現斜向區塊有出現不符合現象即需進行該單元運算，其步驟如圖三的"影像區塊的三維交集運算"。以下我們將依圖四為說明架構。其運算規則如下：檢視該區塊的左右斜向區塊依據 2.10 的程序是否有符合的如果有則以不合的區塊減掉左邊和右邊有符合的區塊。

我們針對此一規則作一實例解說，假設圖四的第 513 個圖形點被非法竄改，即 $513/256=2.007$ 即代表 bt_3 有被非法竄改那麼將導致 $DHWT_2$ 出現錯誤，而與解密程序解出之 HWT_2 不相同，那麼執行此一程序我們查看一下 $DHWT_1$ 與 $DHWT_3$ 是否與 HWT_1 和 HWT_3 相同，只要其中有一符合即可依據規則運作假設 $DHWT_1$ 與 $DHWT_3$ 與 HWT_1 和 HWT_3 符合，我們進行 $DHWT_2 - DHWT_3 - DHWT_1$ 即可獲得 bt_3，因此我們可獲得此區域有被非法竄改，但無法達到一個位元範圍的偵測效應，還需下個步驟的處理方可得到。

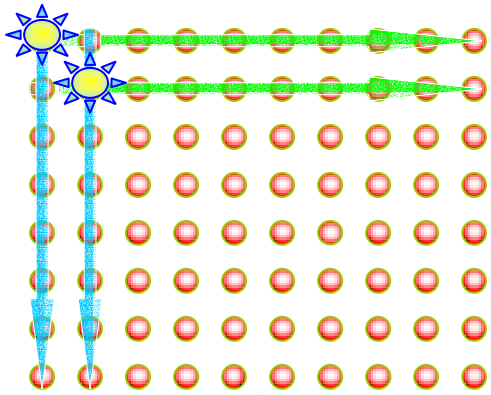
2.10.4 縱、橫、斜向區塊的縱橫交集運算

當 2.10.1~2.10.3 兩步驟做完後，我們即需將三個步驟的運算成果推疊在一起取其交集之部分即是我們所要偵測的被非法竄改點。因此當我們讓縱行區塊以橫列 256 列，直行 2 行為假設前提時即可輕易偵測出最小 1 個位元的非法竄改點。而如果我們讓縱行區塊以橫列 128 列，直行 4 行為假設前提時只能測出最小 2 個位元範圍內的非法竄改點，但是再配合斜向區塊則依舊可以偵測出最小 1 個位元的非法竄改點。

三、分析

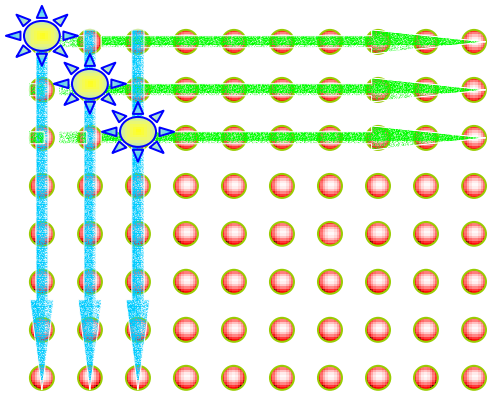
經由我們提出的三維偵測法我們克服了無法偵測至單一圖形點的瓶頸，接下來我們會針對縱橫偵測法所無法偵測出的狀況做分析及解決。

狀況 1.如圖七，當左上角斜線部分有兩個點被竄改時，則縱橫偵測法會偵測出第一列和第二列都有錯誤，第一行和第二行也有錯誤，所以便會產生所謂的最壞情況，因為縱橫交集的部分會是左上角四個點，但只要利用三維偵測法，就可以解決上述的狀況。



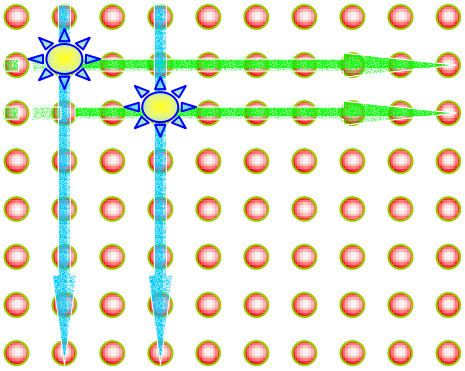
圖七

狀況 2. 如圖八，當左上角斜線部分有三個點被竄改時，則縱橫偵測法會偵測出第一、二和三列都有錯誤，第一、二和三行也有錯誤，所以便會產生所謂的最壞情況，因為縱橫交集的部分會是左上角九個點，但只要利用三維偵測法，就可以解決上述的情況。



圖八

狀況 3. 如圖九，斜線部分有兩個點被竄改(不在同條斜線)時，那麼縱橫偵測法會偵測出第二列和第三列都有錯誤，第二行和第四行也有錯誤，因此會誤判有四個錯誤，所以便會產生所謂的最壞情況，但只要利用三維偵測法，就可以解決上述的狀況。



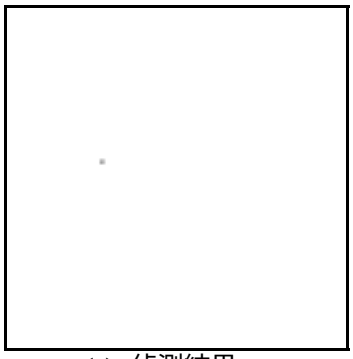
圖九

四、實驗

以下是我們利用三維偵測法，針對四張具有代表性的灰階圖形(圖十~圖十三)進行實驗，發現圖十至圖十三皆可偵測出被竄改的位置，如圖十和圖十一，我們在圖中個別加入一點和一直線，透過實驗結果被竄改的地方很明顯的被表示出來，相對地縱橫偵測法亦可，而在圖十二和圖十三中，我們加入了分析中所提到縱橫偵測法無法明確偵測出的情況，透過實驗結果我們依舊能偵測出被竄改點的位置，進而利用三維偵測法補足了縱橫偵測法所無法明確偵測出的情況。

(a) 原圖

(b) 竄改後的圖

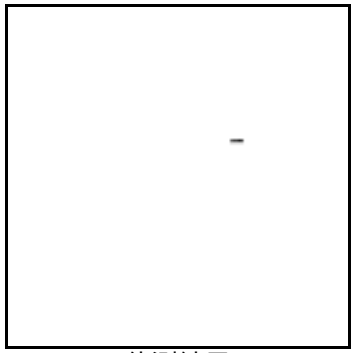


(c) 偵測結果

圖十

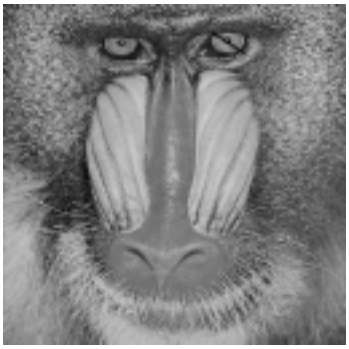
(a) 原圖

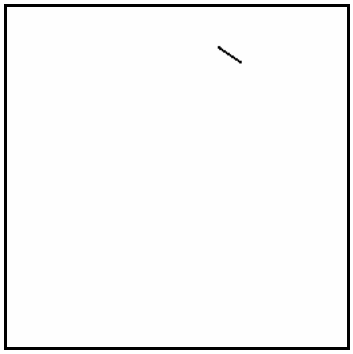
(b) 竄改後的圖



(c) 偵測結果

圖十一





(a) 原圖　　**(b)** 竄改後的圖　　**(c)** 偵測結果

圖十二

(a) 原圖　　**(b)** 竄改後的圖　　**(c)** 偵測結果

圖十三

五、結論

目前，影像已與我們日常生活息息相關，尤其現在利用電腦科技竄改、竊取或是偽造影像的問題層出不窮，因此要如何偵測出被竄改點，如何把原始圖像盡量的還原出來，便是我們本文的精神所在，我們針對目前在圖形非法竄改點的偵測問題方面，提出三維偵測法，利用這個方法使我們可以偵測出小到 1 個圖形點的非法竄改我們都有辦法偵測出來。而且由實際的實驗結果和分析我們發現，既使在最壞狀況下，我們依然可以精確的找出非法竄改點的位置。因為我們所做的實驗是利用灰階影像為主，因此未來的研究方向可朝向彩色影像或是找出更快速的方法來偵測被竄改點的位置。

六、參考文獻

1. Ching-Rong Yang, Min-Shiang Hwang, Yuan-Liang Tang, "偵測影像的被破壞點之研究--縱橫偵測法," Eighth National Conference on Science and Technology of National Defense, Tao-Yuan, Nov. 1999.

2. D. Atkins, M. Graff, A. K. Lenstra and P. C. Leyland, "The magic words are squeamish ossifrage," In Advance in Cryptology, Asiacrypt'94, pp. 263-277, Springer-Verlag, 1995.

3. C. C. Chang, K. F. Hwang, and M. S. Hwang, 2000. "A Digital Watermarking Scheme Using Human Visual Effects, " Informatica, 24 (4): 505-511.

4. G. L. Friedman: "The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image, " IEEE Trans. on Consumer Electronic, Nov.1993.

5. M. S. Hwang, C. C. Chang, and K. F. Hwang, 1999. "A Watermarking Technique Based on One-way Hash Functions," IEEE Transactions on Consumer Electronics, 45 (2): 286-294.

6. M. S. Hwang, C. C. Chang, and K. F. Hwang, 2000. "Digital Watermarking of Images Using Neural Networks," Journal of Electronic Imaging, 9 (4): 548-555.

7. Chang-Tsun Li, Der-Chyuan Lou, Te-Lung Yin, and Jiang-Lung Liu, "Image authentication via image-dependent watermarks," proceedings of the ninth national conference on information security, pp 99-102, May 1999.

8. R. L. Rivest, "The MD5 message digest algorithm," RFC1321, April 1992.

9. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, pp. 120-126,February 1978.

10. Bruce Schneier, Applied Cryptography, Second Ed., Wiley & Sons, 1996.

11. Yuan-Liang Tang, Min-Shiang Hwang, Ching-Rong Yang, "An Image Authentication Scheme Based On Digital Signatures," Pakistan Journal of Applied Sciences, vol. 2, no. 5, pp. 553-557, May 2002.

12. R. B. Wolfgang and E. J. Delp, "A Watermarking Technique for Digital Imagery: Further Studies," Proceedings of the International Conference on Imaging Science, Systems, and Technology, 1997, Las Vegas, pp. 279-287.

13. P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," Proc. IEEE Intl. Conf. On Image Processing, Chicago, USA, Oct.1998, pp.455-459.

14. M. Wu and B. Liu, "Watermarking for Image Authentication," Proc. IEEE Intl. Conf. On Image Processing, Chicago, USA, Oct.1998, pp.437-441.