

金利 Hide

(資 訊 隱 藏 應 用)

Data Hiding Application - Jin Li Hide

專題組員：吳宥蓁、張育齊、方莉雯、王譽霖

專題編號：PRJ-NTPUCSIE-101-002

執行期間：101 年 09 月 至 102 年 06 月

1. 摘要

由於科技的日新月異以及網際網路的蓬勃發展，讓資訊的傳輸與複製變得越加簡單且快速；同樣地也有越來越多秘密資訊遭到非法盜取、偽造或竄改等問題產生，因此「如何確保資訊的安全性」已經成為當今無可忽視的議題。目前普遍隱藏訊息內容的方法有兩種，分別是「密碼學」與「資訊隱藏」。

密碼學(Cryptography)是將欲保護的秘密資訊(原先的明文, plaintext)經過資料加密來改變檔案或訊息的內容，讓其從一個完整可讀的狀態轉換為一堆無意義且雜亂的密文(ciphertext)，唯有輸入正確密碼才可以將加密的檔案解出。但此加密的檔案並非隱匿，相當容易引起他人的懷疑；且在檔案傳送的过程中也可能遭受到有心人士惡意的監聽或入侵。雖然開發破解技術需要一段或長或短的時間，可是一旦加解密的技術被突破，裏頭的秘密資訊很可能被用於不正當用途，如此將導致不可預期的後果。

「資訊隱藏」(Data Hiding)是將秘密資訊不著痕跡的隱藏在所要傳送的影像當中(形成一種偽裝)，用來達成秘密傳遞資訊的目的。此種隱藏訊息的方式，無論是在交換訊息的途中或者將藏有秘密資訊的圖檔放置在網路上，都相當不易引起他人的懷疑與猜測，而這也是資訊隱藏與密碼學兩者之間最大的差異(圖 1)。

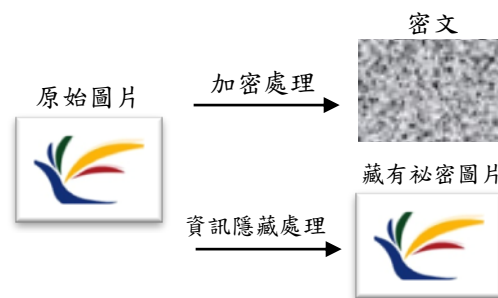


圖 1：密碼學與資訊隱藏比較

本專題是使用「資訊隱藏」當中的 LSB (Least Significant Bit) 方法[1]，透過密碼將秘密資訊隱藏於圖片當中，並利用簡易操作的圖形化介面方便使用者來使用。

關鍵詞：資訊隱藏 (Data Hiding)、最低有效位元、最低位元取代、最不重要位元、Least Significant Bit(LSB)

2. 簡介

本專題使用資訊隱藏(Data Hiding)技術。並採用當中最為普遍的LSB (Least Significant Bit, 最低位元取代或最不重要位元)方法,此方法藉由直接更改圖片的像素值來達到藏入祕密資訊的目的。其置換了原本圖片影像像素的低位元值,而改以欲藏入的重要祕密資訊來取代之。

使用者能夠選取任意數量、任何類型的祕密資訊,例如:文字檔(.doc)、音樂檔(.wav)、圖片檔(.jpg)等等,再搭配一組密碼,透過本程式,便能將其隱藏在一張看似平凡無奇的圖片當中。且藏入前與藏入後的圖片在視覺上幾乎沒有任何的改變,完全不會引起任何人的懷疑(圖2)。

取出祕密資訊的過程也相當簡單,僅需要一張藏有祕密資訊的圖片及正確的密碼,再透過本程式,便能將祕密資訊完整的還原至使用者指定的輸出資料夾。

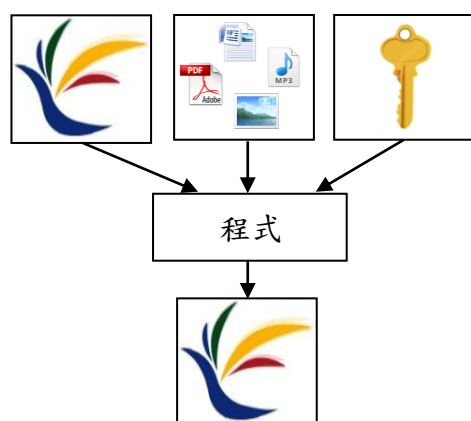


圖 2：專題目標示意圖

3. 專題進行方式

3.1 LSB 方法說明

本專題在決定要使用資訊隱藏中的LSB方法後,我們先參考幾篇與LSB相關的論文,了解LSB的原理。LSB隱藏方法是利用直接更改圖片中較低位元的像素值,並以欲藏入的重要祕密資訊取代之。

假設在圖片的每個像素值當中取最低的「三個」位元值來置換。例如,某一像素值為177 (=10110001₂),欲藏入的祕密資訊片段為6 (=110₂),我們直接修改原圖像素值的最低三位元,並以祕密資訊片段來取代之(最低三個位元001→110),藏完後新的像素值為182 (=10110110₂)。

選擇置換LSB的幾個位元值會影響到藏入祕密資訊後圖片影像的品質。置換的位元數越多,所造成失真的情形也越嚴重。使用PSNR評估方法衡量圖片影像品質,衡量的單位為db(Decibels),其計算公式定義如式(1)與式(2);通常PSNR值越高表示影像的品質越好。^[1]

$$PSNR = 10 \times \log \left(\frac{255^2}{MSE} \right) \quad \text{式(1)}$$

$$MSE = \frac{\sum_{n=1}^{FrameSize} (I_n - P_n)^2}{FrameSize} \quad \text{式(2)}$$

當PSNR>30db,人的肉眼不容易看出經過更動的圖片與原圖的差異。因此應盡量讓PSNR>30db,但PSNR值越高,不一定越好,有些還是須倚靠人的肉眼來輔助判斷影像的品質才較為正確。

3.2 程式設計摘要

為使 PSNR 值大於 30db，我們選擇置換較低的一到三個位元，其 PSNR 值約為 48.1、38.6、31.2(db)；為了達到最高的資訊隱藏量，本專題挑選最低的「三個」位元置換成祕密資訊的內容^[1]。

由於專題的核心技術牽涉到影像處理，因此我們在開發程式前閱讀了有關影像處理^[4]與學習 OpenCV 的相關書籍(Open Source Computer Vision Library，為一個跨平台的計算機視覺庫。主要用於圖像的處理、計算機視覺以及模式識別程序等等)^{[2][3]}。

確定本專題所要實作的方向之後，每位組員先在電腦上安裝開發程式的環境，接著進行主要功能的流程規劃。從檔案藏入開始，分別取出每個檔案的檔案名稱長度、檔案名稱、檔案內容大小、檔案內容四個資訊，依序放到欲藏進圖片的陣列當中，再分成每三個位元一組，將其藏入圖片像素值的最低三個位元；為了提升處理資訊的速度，陣列以 2KB 為單位，分批處理陣列當中的內容，同時也更加節省陣列宣告的空間(圖 3)。



圖 3：祕密資訊陣列

為了與新檔案之間區隔，我們以“-1”的檔案名稱長度來作為多個祕密檔案的結束，同時也代表祕密檔案已全數藏入。在藏入圖片的祕密資訊陣列最前頭，我們加入了“NTPU”四個字母，用來判斷取出祕密資訊時輸

入的密碼正確與否(圖 4)。輸入的密碼經過 Hash Function 產生之數值做為 Random 的種子，得到亂數順序，然後以該順序藏入每個像素值的後三個位元；若使用者輸入了錯誤的密碼，便會因為得到不一樣的亂數順序而無法成功取出最前頭的“NTPU”，在回傳錯誤的時，一同將原先設置的密碼提示傳回給使用者，讓使用者能重新輸入正確的密碼(圖 5)。

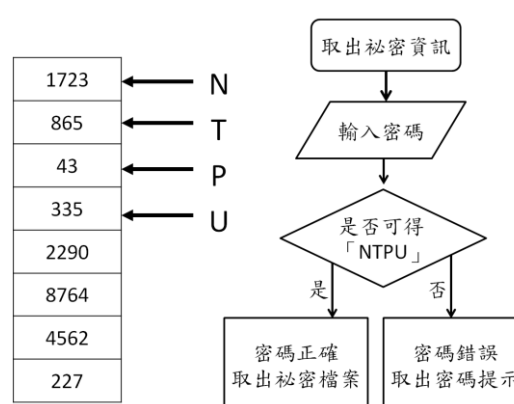


圖 4：判斷密碼正確與否

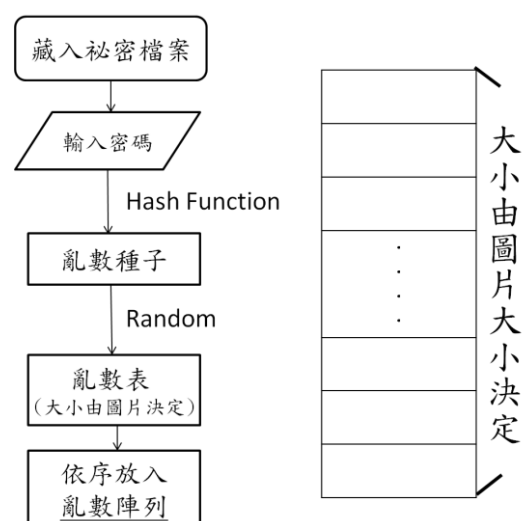


圖 5：產生藏入順序的亂數方法

不同於“NTPU”藏在秘密資訊陣列的最前端，密碼提示是被固定放置於圖片最右下方的像素值當中；由於密碼提示必須是不需要正確的密碼便可以取得的內容，因此將密碼提示另外存放，讓使用者在輸入錯誤密碼的同時，產生密碼提示訊息。

C 語言的核心技術程式皆已開發完成之後，為了與 C# 開發的圖形化介面結合，我們將原先的 C 核心技術程式包裝成 .dll 檔案型式；經 C# 程式呼叫，由 .dll 檔案執行核心技術部分，並回傳結果給 C#，再由圖形化介面輸出結果給使用者（圖 6）。

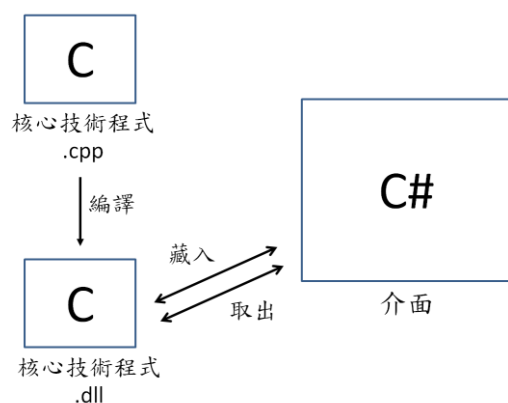


圖 6：C 核心技術程式與 C# 介面結合

3.3 人員配置與職責、時程規劃

在開發程式的最初我們是所有組員一同在開發程式，看誰能優先撰寫出循序流程的下一個功能；完成後組員將其設計的概念與方法與其他組員說明與討論，接著再繼續撰寫程式的下一個流程。

但這樣分工不佳的撰寫程式方式讓我們面臨到許多的問題：每位組員的程式設計習慣雖然相似但有些許差異，在變數的命名上大家採取自己習慣的命名方式，導致在結合程式時每次都需要重新修改變數名稱，另外

多人同時進行同一件事物是非常沒有效率且浪費時間的！

因此我們開始思考該如何能明確的分工，發揮團隊合作的精神，同時提高我們專題進行的效率（組員配置與職責請參照最末頁「表 1」，專題的時程規劃請參照最末頁「圖 7」）。

3.4 主要困難與解決之道

在技術部分比較困難的地方是在 OpenCV 的使用上；在完成所有的功能的建立後，核心技術程式能正確執行，但是偶爾會產生錯誤的情形。

經過一段時間的偵錯，最終發現可能是因為 OpenCV 在儲存圖檔時，會擅自更動像素值，這在大多數 OpenCV 的教學網站與書籍當中皆沒有提到，而是我們在不斷的找尋發生錯誤可能的原因時，發現到有這個的可能性。

發現到此可能的原因後，我們撰寫程式計算出各個狀態時的平均像素值，證實是因為 OpenCV 在執行儲存圖檔操作時，有時會擅自更改已藏入秘密資訊之後的像素值。為了解決問題，決定自行撰寫儲存圖檔的功能，才真正的解決了這個棘手的問題。

4. 主要成果與評估

「金利 Hide」是本專題最終成果的呈現，以下為六大特色。

- 一、接受任意數量、任何格式的秘密檔案。
- 二、支援 .bmp、.png、.tif 的圖片。
- 三、圖片在藏有秘密資訊後，看不出差異。

本專題因為使用 LSB 方法，以及僅置換三個位元，使藏入前

與藏入後的圖片，肉眼無法分辨；因此，設計「比較」介面，方便使用者比對圖片藏入前與藏入後的差異。

四、創意風格的圖形化介面。

此概念來自於「木製的寶物箱」，由密碼解出檔案就如同以一把金鑰匙開啟木製的寶物箱。

「木製」也成了創意的起點，我們開始設計一連串與木質、木頭相關的物品。從公告欄上張貼的老舊紙張、路上常見的指路路標、店家門前不同造型的吊飾牌等等，讓「金利 Hide」以木製風格風格呈現，使用者在執行程式的同時，也能一同欣賞到另類的創意設計。

五、密碼提示

想要成功藏入或取出祕密資訊，密碼是一個重要的角色。在要取出祕密資訊時，如果「忘記密碼」，將使得重要的祕密資訊檔案猶如沉入大海的寶箱一般那樣的難以取得；也因此金利 Hide 具備「密碼提示」功能，讓使用者在藏入祕密資訊的同時，也能一同輸入與密碼相關聯的提示字。

在取出祕密資訊時，若忘記或打錯密碼，金利 Hide 將會顯示出當初連同密碼一起輸入的密碼提示，讓使用者能夠快速的聯想起當初藏入該祕密資訊時所輸入的正確密碼。

六、貼心的流程配置與引導

在金利 Hide 的執行過程中，每個步驟都以適當的提示或視窗訊息來引導、提醒使用者，

讓其能按照明確的指示順利的完成欲執行的操作；且金利 Hide 的設計為使用者設想了各式各樣的情況，使用者可直接透過本程式刪除原始檔案、重新命名等。使用者也不需要一個一個檔案逐一的選取，裡頭皆附有全選與可同時多選功能；另外也增加圖片的拖曳功能，讓使用者能直接將圖片或檔案拖拉進程式，不需要再找尋並點選其所在位置，絕對在使用上更為快速。

在最後的測試階段，我們找了許多原先沒有接觸過、聽過金利 Hide 的許多朋友們來讓他們試著(模擬)操作看看。一開始他們都會先被一系列創意風格的木質設計所吸引，一旦閱讀過關於頁籤的簡要介紹後，便可直接可以開始使用「金利 Hide」；在我們進一步的說明其技術與應用後，使用者對「金利 Hide」皆有更深一層的了解，也同意能將「金利 Hide」用於日常生活中。

我們也在最後測試階段，反覆的測試各式的圖片及祕密資訊的藏入與取出的動作；一方面確認「金利 Hide」的各項功能皆已完善，另一方面也檢視當中是否有不順暢或錯誤的流程設計。

在這一年的專題當中，我們有著很大的成長，一開始只聽過資訊隱藏這個專有名詞，但相關知識也在閱讀相關的論文與書籍的過程中慢慢吸收，開始寫核心技術的程式又是進入一大挑戰，介面的程式及設計也是步步難關，報告撰寫、為成果發表進行準備和演練也都相當不易，每一個階

段都讓我們收穫良多。除了養成獨立思考與解決問題的能力、將理論實際操作提升學習的成效之外，也更佳了解到團隊合作的重要性以及分工合作的必要性。在每一次的專題製作過程中，所有的組員皆一同進行參與和討論；不僅養成了每位專題組員主動負責的精神，也逐漸地培養出愉快且良好的團隊合作氣氛。

5. 結語與展望

資訊隱藏為近年來相當受到重視的課題，而本專題也利用其方法將資訊隱藏實踐於日常生活中。

本專題所實作的 LSB 方法在其做法上是屬於不可逆的資訊隱藏 (Irreversible Data Hiding) 技術，不可逆指的是當我們將秘密資訊隱藏進偽裝的圖片時，我們是直接更改掉原圖(藏入前圖片)的像素值，並且沒有對原先的像素值做任何保存或記錄的動作，也因此取出隱藏於圖片當中的秘密資訊後，我們並無法將原圖回復成最初未被修改前的狀態。

在未來，本專題希望能以提升更高的資訊隱藏量以及如何能維持較高的影像品質為研究發展的方向，以期能進一步達成更佳層次的資訊隱藏辦法。也能增加研究其他可逆式資訊隱藏方法，讓其在未來能有更多方面與領域上的應用。

6. 特別感謝

最先要感謝張仁俊老師，在我們每個星期的專題會議中能夠和我們一同來檢視我們每週的專題成果；也教導我們以正面積極的態度來面對我們碰到的困難與問題，讓我們能更加發揮團隊合作的精神來思考並解決我們所面臨到的挑戰。老師在程式設計與學問上嚴謹態度和靈活的思考都是我們能夠多加學習且非常棒的典範。

也非常感謝身邊很多的同學、朋友以及學長姐們對於「金利 Hide」提供相當多寶貴的評論與建議。在我們設計程式遇到一些專業技術問題時，能夠給予我們許多的思考方向和意見；也讓「金利 Hide」在圖形化介面設計上，功能與整體呈現的效果能夠更加的流暢與完整。

7. 參考文獻

- [1] Chan, C. K., Cheng, L. M., "Hiding Data in Images by Simple LSB Substitution", Pattern Recognition, 37(3), pp. 469-474, 2004.
- [2] 劉瑞禎, "OpenCV 教程. 基礎篇", 2007
- [3] Gary Bradski, Adrian Kaebler, "Learning OpenCV", 2009
- [4] 王旭正, "數位影像處理與應用 =Digital image processing and applications", August 2012

表 1. 組員配置與職責

●主要負責 △次要協助

工作項目 人員配置	題目訂定	閱讀文獻	流程規劃	檔案藏入	檔案取出	多檔藏取	程式整合	C#介面功能	功能修改	介面設計	最後測試	報告撰寫	PPT製作	成果發表
吳宥蓁	●	●	●	△	●	●	△	●	●	●	●	△	△	●
張育齊	●	●	●	△	●	●	●	△	●	●	●	△	△	●
方莉雯	●	●	●	●	△	●	●	△	●	△	●	△	●	●
王譽霖	●	●	●	△	●	●	△	●	●	△	●	●	△	●

時間 工作項目	2012 年				2013 年					
	9 月	10 月	11 月	12 月	1 月	2 月	3 月	4 月	5 月	6 月
題目訂定										
閱讀文獻										
流程規劃										
檔案藏入										
檔案取出										
多檔藏取										
程式整合										
C#介面功能										
功能修改										
介面設計										
最後測試										
報告撰寫										
PPT 製作										
成果發表										

圖 7. 工作時程甘特圖