

Disguise

捉迷藏

專題組員:張睿昇 羅郁凱

執行期間:2014 年 9 月 至 2015 年 6 月

1. 摘要

網路和手機的使用日益普遍，幾乎人人都可以用手機藉由網路傳送資訊，然而，在傳送資訊的過程中，我們並無法防止有心人士竊取你的資料，只能藉由加密的方使讓人難以破解。但當你收到訊息時，很有可能不是你本人看到的，藉由本程式除了可以加密訊息以外，最重的是可以偽裝和欺騙，讓人在萃取真正的訊息之前，只能看到假的訊息，使不知情的人上當受騙，而且正常且有意義的文字訊息更不會讓人產生訊息被加密過的疑心。

2. 簡介

研究動機:

日常生活中，總是有許多只想要給朋友看的重要訊息，但是由於不確定是否是由本人接收訊息，甚至是在訊息傳送的途中被攔截，都有可能造成訊息的外洩，而普通的加密方法產生的密文容易辨認出是否經過加密，增加遭到破解的風險，因此我們決定除了傳統的加密藉以保護秘密資訊外，還打算用假訊息欺騙敵人，降低被發現以及破解的風險，也藉此防範傳輸中間竊聽者，而即便取得我們 APP 的程式碼也無法破解訊息，因為我們是以密碼做為亂數的種子，且不限定

密碼的長度，所以大大的降低被破解的風險。之所以我們開發會選用 iOS 做為主要 OS 是因為 APP Store 上的 APP 相較於 Windows mobile 或是更為主流、市占率為目前全球最高的 Android 而言其 APP 數量、品質、種類更為多樣化、更為精緻，再加上今日手機的使用量與需求量大增，很多人的生活已經離不開手機，因此我們決定把我們的專題寫成 iOS 的 APP。

3. 專題進行方式

3.1 基本架構

本程式主要分成三個部分:

1. 加密及偽裝
2. 傳送與接收
3. 解密及萃取

Part 1. 加密及偽裝:

加密是使用區塊加密的方式，以 byte 為單位，雖然我們選用最簡單的 EBC 加密模式，但是為了增加破解難度，程式讓使用者輸入一組沒有長度限制的英文和數字混合的字定密碼的 key 做為亂數的種子，再用亂數和偽裝的文字或圖片做 xor 而產生 data，每組 data 都沒有關聯性，讓人就算不小心獲得其中一個 data 也沒辦法得知其他的 data，進而增加破解的困難度。一般加密完成後的密文和偽裝的文字

一起傳輸的話，如果遭到攔截，很容易就被發現，該資料有經過加密，因此為了達到偽裝的目的，密文會被藏入一張使用者任選的照片或圖片中 (Steganography)，讓密文看起來只是一張普通的圖像。

Steganography

Steganography 是一門關於訊息隱藏的技巧與科學，所謂訊息隱藏指的是不讓除預期的接收者之外的任何人知曉訊息的傳遞事件或者訊息的內容。隱寫的訊息通常用一些傳統的方法進行加密，然後用某種方法修改一個「偽裝文本」(coverttext)，使其包含被加密過的消息，形成所謂的「隱秘文本」(stegotext)。如果載體文件 (cover file) 相對隱秘文件的大小 (指數據含量，以位元計) 越大，隱藏後者就越加容易，而我們所採取的方法是利用 LSB。數字圖像在大眾傳播媒體上被廣泛用於隱藏消息，例如：一個 24 位元的圖中的每個像素的三個顏色份量 (紅，綠和藍) 各使用 8 個位元來表示。如果我們只考慮藍色的話，即有 256 種不同的數值來表示深淺不同的藍色。而像 11111111 和 11111110 這兩個值所表示的藍色，人眼幾乎無法區分。因此，這個最低有效位就可以用來存儲顏色之外的訊息，而且在某種程度上幾乎檢測不到。依此類推可以在紅色以及綠色隱藏相同的量。

Part 2. 傳送與接收

傳送與接收的媒介，我們選用 Email，除了使用上的方便以外，Email 可以容納的文字量相較其他通訊軟體高，且可以通時傳送文字和圖片，且不同的 Email 用戶之間可以通訊，如果是用 LINE 這種通訊軟體，雖然方便，但僅限於 LINE 的用戶可以互相傳遞資訊，則方便性會減少。Email 的框架傳送端是使用 MessageUI 這個 framework，用 iOS 現有的框架來實現寄信的功能，然而，收件的方式就比較複雜，MessageUI 並沒有收取郵件的功能，於是我們選用 mailcore2 的框架來完成簡單的收取郵件的功能。

Part 3. 解密及萃取

Extraction 主要就是將 Hiding 的部分倒推一遍。剛剛有提及利用 Steganography 將 ciphertext 藏進 cover image 使其成為肉眼無法辨識差異的 Stego image，Extraction 的首要步驟就是先將 Ciphertext 從 Stego image 中還原，再倒著做 ECC 加密模式，最後產生 Secret。

3.2 Program algorithm

本程式主要分成四個部分：

1. Text Hiding
2. Image Hiding
3. Text Extraction
4. Image Extraction

Part 1. Text Hiding

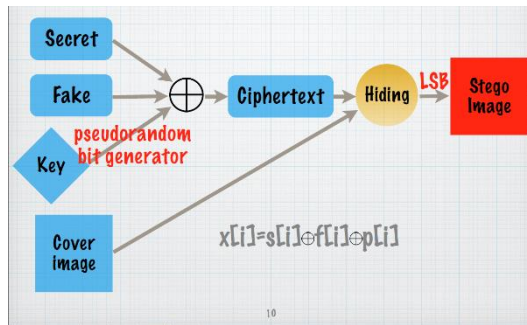


Fig 1. Text Hiding

首先我們將 Secret text 跟 Fake text 利用 ASCII 轉換成數字，再以 bit 為單位做 xor，然後再跟 key(俗稱的密碼)(可英文可數字)透過 pseudorandom bit generator 產生的 pseudorandom bit sequence 作 xor，會產生 Ciphertext，而這時會將 Cover image 與 Ciphertext 利用 LSB 的方法將 Ciphertext 存進 Stego image，並且以 3:3:2 的比例存進 RGB 之中，將 Cover image 變成 Stego image。又，將 ciphertext 的長度紀錄於第一、第二的 pixel 藉以降低時間複雜度，並且避免無謂時間的浪費。

Part 2. Image Hiding

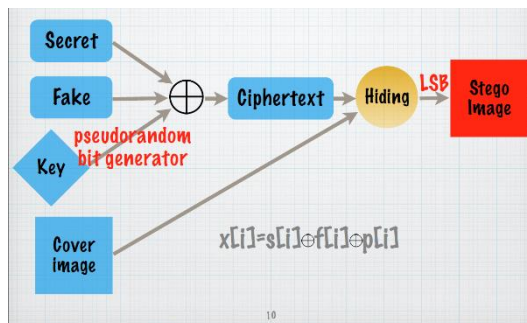


Fig 2. Image Hiding

首先將 Secret image 以及 Fake image 以 R 對 R、G 對 G、B 對 B 的方式以 bit 為單位做 xor，產生的 data 在跟 key 透過 pseudorandom bit generator 產生的 pseudorandom bit

sequence 作 xor，產生的 Ciphertext 的 RGB 各藏 6 個 bit 到 Cover image 中，並且以 Cover image 的 2 個 pixel 存 ciphertext 的 1 個 pixel 藉此避免 Cover image 失真過於嚴重，並將 Cover image 轉換為 Stego image。

Sending

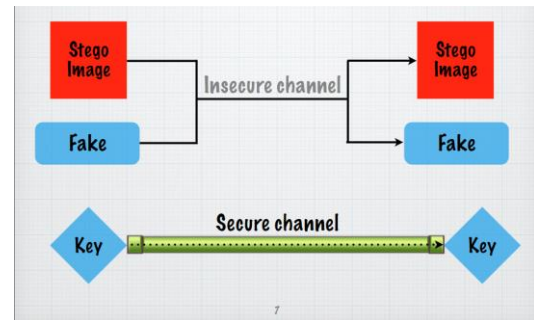


Fig 3. Sending and receiving

將 Fake 以及 Stego image 傳值並載入 Apple 的 MessageUI Framework 中，藉此減少使用者須自行輸入的不便，增進使用者的使用體驗。

Part 3. Text Hiding

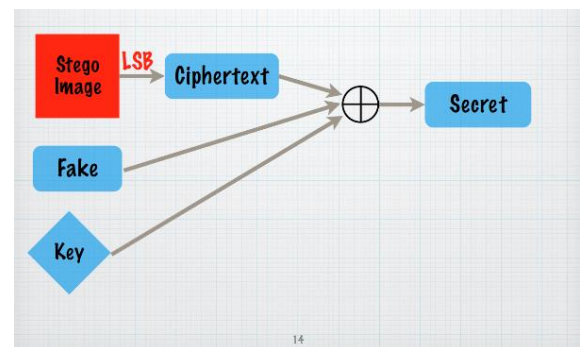


Fig 4. Text Extraction

剛剛在 Text Hiding 中有提到我們利用 LSB 的方法將 ciphertext 藏進 cover image 使其成為肉眼無法辨識差異的 Stego image，因此首要的步驟就是先將 Ciphertext 從 Stego image 中還原，再與 Fake 以及使用者自行輸入的 Key 作 xor，然後再與 ASCII 做對照

最後產生 Secret 並顯示在畫面上。

Part 4. Image Extraction

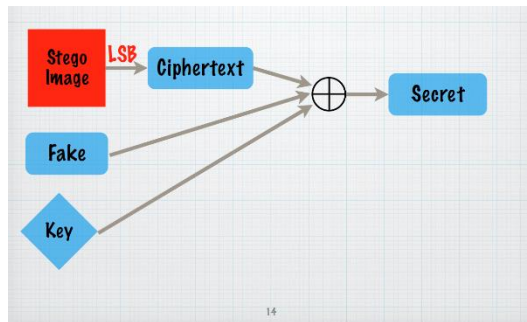


Fig 5. Text Extraction

首先以兩個 pixel 為一組，將 Stego image 中的 Ciphertext 還原，再與 Fake 以及使用者自行輸入的 Key 作 xor，一樣是 R 對 R，重新產生 Secret image、G 對 G、B 對 B 的方式以 bit 為單位，即可得到 Secret image。

4. 主要成果與評估

Disguise 的目的就是希望可以保護秘密的資訊，但是在保護秘密資訊的同時，又可以藉由假的資訊欺騙其他非接收者的人，也可以防範傳輸中間的竊聽者，即便取得我們的程式碼也會因為 key 做為最後一道屏障而無法破解該訊息。

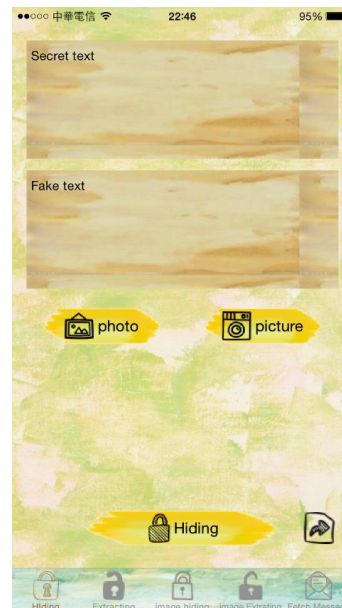


Fig 6. Text Hiding



Fig 7. Text Extraction



Fig 8. Image Hiding

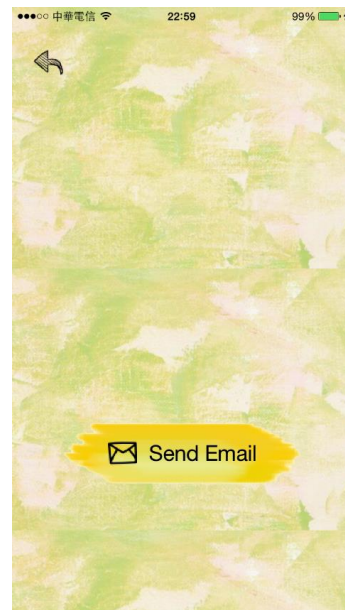


Fig 10. Sending



Fig 9. Image Extraction



Fig 11. Send Email



Fig 12. Fetch message 帳密登入畫面



Fig 13. Email TableView

5. 結語與展望

結語：

Disguise 既可以保護秘密資訊，而在保護秘密資訊的同時，又可以藉由假的資訊欺騙其他非接收者的人，也可以防範傳輸中間竊聽者，就算取得我們的程式碼也會因為 key 做為最後一道屏障而無法破解該訊息，因為

我們的 APP 不會將 key 存在 Disguise 之中，最後，為了增加使用者體驗，我們的 APP 是 All in one，結合了偽裝、加密、傳送、接收、解密、萃取於一，可以不用因為收 email 切換到郵件 APP 之中。

展望：

1. 推出其他平台版本

2. 偽裝影片

6. 銘謝

感謝指導教授們每個星期的專題會議，檢視我們的進度和引導我們以正面積極的態度來面對問題。

7. 參考文獻

[1] 朱克剛，iOS 8 程式設計實戰——205 個快速上手的開發技巧，第 56-326 頁，碁峰資訊股份有限公司，台北、台灣（2004）

[2] Deepesh Rawat and Vijaya Bhandari, “A Steganography Technique for Hiding Image in an Image using LSB Method for 24 Bit Color Image”, International Journal of Computer Applications, Volume 64 - No.20, pp. 15-19, Feb, 2013

[3] “MailCore/mailcore2” [Online], May 12, 2015, available at: <https://github.com/MailCore/mailcore2>.

[4] “MailCore An e-mail library for Mac OS X& iOS” [Online], May 12, 2015, available at: <http://libmailcore.com/>.

[5] Furnace iOS 程式設計中文學習站, “使用 UIImagePickerController 從圖庫中取得影片 / 相片的方法” [Online], Oct 10, 2014, available at:

http://furnacedigital.blogspot.tw/2011/12/uiimagepickercontroller_26.html.

[6] Johnny Sung, 製作xcode憑證,
available at:

<http://app-island.com/app/2560/%E8%A3%BD%E4%BD%9Cxc%E6%86%91%E8%AD%89%EF%BC%8C%E8%AE%93%E9%96%8B%E7%99%BC%E7%9A%84APP%E5%8F%AF%E4%BB%A5%E5%AE%89%E8%A3%9D%E5%88%B0iOS%E8%A3%9D%E7%BD%AE%E5%81%9A%E6%B8%AC%E8%A9%A6>

