Secret Sharing with Multi-cover Steganographic Audio Files

洪胤勛　吳坤熹
國立暨南國際大學 資訊工程學系[1]
{s108321019,solomon}@ncnu.edu.tw

摘要

祕密共享（secret sharing）[1, 2]為一種分享資料的技術，它會把資料拆成若干持份(sharing)，並且把 sharing 分給不同的人。若要回復原本的資料，則需要湊齊大於或等於一定門檻(threshold)數量的 sharing，否則不能還原出原本的資料。本論文將探討如何安全的把這些 sharing 分送給不同的人，並導入隱寫術，將 sharing 藏在聲音中，讓這些 sharing 在不易被察覺的情況下發送給不同的人。

I. 前言

隨著資安的議題日益重要，許多加解密的原理與工具更顯得它們的重要性。有些機密性的資料，不能夠直接赤裸裸的流通在網路上，因此有些人選擇給資料做了加密，防止資訊外洩。但在某些情況下，做了加密反而「欲蓋彌彰」。比如在一個被監控的網路環境中，或是網路會流經不受信任的節點，若你的封包攜帶了加密的資料，管理者通常能很快發現此封包異常，因此封包可能就此被丟棄又或是被選出來破解。為了避免封包被遺棄或是被別人嘗試破解，有人會選擇把機密的資料藏在一些不容易發現的地方，像是圖片或是聲音，甚至是封包的表頭，此方法又稱隱寫術（steganography）。

Secret sharing 在資安的領域也相當重要，它與加解密不同，secret sharing 為一種分享資料的技術，在這種架構下，它解決了祕密資料掌握在單一個人手上的保密性問題。比如需要多人授權的密碼，像是金庫的密碼，老闆可能會有緊急狀況，無法親自到金庫，因此需要部下幫他開啟，但如果把密碼只交給一個人，那將非常危險，因此理想狀況是需要多人才能開啟；又或是階層式授權的密碼，給經理和主管有開啟的權力，但也不能讓單一個經理或是單一個主管直接擁有密碼，而是讓一個經理和一個主管，或是三個主管決定要打開金庫時，才能得出密碼。這些問題都可以靠 secret sharing 來解決。

Secret sharing 同時也解決機密資料掌握在單一個人手上的另兩個問題。第一為資料的穩健性（robustness），在單一資料的架構下，當系統被駭客入侵，資料被竊取走，那就表示機密一定被駭客知曉；但在 secret sharing 的架構下，即使駭客竊走一份資料，他也無法得知祕密到底為何，他必須去找到一定數量的資料才有辦法還原出原始的機密資料。因此 secret sharing 有助於達成較佳的縱深防禦（defense in depth）。第二為資料的可靠性（reliability），在單一資料的架構下，資料若不小心遺失或是被破壞掉了，那祕密就此消失，再也無法取得；但在 secret sharing 的架構下，即使遺失一份或損壞一份資料，仍然可以靠其他的資料得以復原。

Secret sharing 的做法在1979年相繼被 Adi Shamir [1]和 George Blakley [2]發表。而他們的解法又常分別被稱為 Shamir’s Secret Sharing 與 Blakley’s Secret Sharing。 Shamir’s Secret Sharing 使用多項式來達成目的，而 Blakley’s Secret Sharing 使用有限域的幾何空間。雖然兩者都有 threshold 的設計，但 Blakley’s Secret Sharing 較複雜且效率較低[1]，因此本論文將著重在 Shamir’s Secret Sharing。

本論文除了使用 Shamir’s Secret Sharing 外，為了讓資料不被輕易察覺，還使用隱寫術，把產生的 sharing 藏在聲音檔裡面，並且派送至不同目的地。

II. 研究動機

Secret sharing 與一般的單一祕密架構不同，它會拆分成很多不同的 sharing，因為要收集到多份的 sharing 才能解開祕密，這使得它要被破解更困難；因為要破壞掉多份的 sharing，才能使祕密真正被揭露出來，這讓外人要破解它更艱辛。以上兩個原因，讓 secret sharing 比起單一祕密的架構更加強韌，更可以信任。

但也因為祕密會被拆分成多個 sharing，這讓傳遞和保存 sharing 時，需要花費額外的資源來儲存和傳送這些 sharing。如果 sharing 赤裸裸地呈現在封包內容裏，那麼可能在這些 sharing 到達目的地前，就會被偵測並刪除了，這並不是我們想要的結果。因此，本論文研究如何讓 sharing 能夠在安全、不易被察覺的情況下抵達目的地。

[1] 本研究成果感謝國立暨南國際大學與埔基醫療財團法人埔里基督教醫院產學合作之「埔暨計畫」(111-PuChi-AIR-006) 經費贊助

III. 相關研究

A. Securing matrix counting-based secret-sharing involving crypto steganography [3]

這篇的作者使用了 matrix-based secret sharing [4]，一種由 counting-based secret sharing [5]改進而來的 secret sharing 方法。它把由祕密產生的 sharing，使用三原色（RGB）的圖片，並使用兩種不同的隱寫術。第一種使用 LSB（Least Significant Bit），第二種使用了 DWT（Discrete Wavelet Transform）。

B. Secret sharing with multi-cover adaptive steganography [6]

這篇的作者提出了 multi-cover，跟傳統只藏在單一張圖片不同，作者把 sharing 隱藏到不同的圖片裡。藏的地方越多，單一張圖片儲存的資訊越少，因此也更不容易被發現。

C. Cyber warfare: steganography vs. steganalysis [7]

在此篇論文裡，作者針對圖片最低有效位元（Least Significant Bit，LSB）的隱寫術做了隱寫分析。他發現，通常在一張圖片上，一個點的 LSB 會與其鄰近點的 LSB 相似。因此只要抽取出整張圖片的 LSB，形成 LSB-plane，並針對此做比較，就可以很輕易發現這張圖片裡面是否藏有訊息，因為藏有訊息的 LSB-plane 中將會有某些部分是不連續的。

IV. 研究目標

本論文將使用 secret sharing 來提升祕密的穩健性和可靠性。除了 secret sharing 外，同時使用隱寫術來隱藏 sharing，讓 sharing 不容易被發現。

在 Cyber warfare [7]這篇文章裡，作者提到使用 LSB 把資訊藏在圖片裡面是件危險的事，因為此方法很容易被偵測。為此，本論文將使用聲音取代圖片來進行隱寫術，聲音除了沒有 LSB 連續的特徵以供偵測外，聲音也不像圖片一樣，能夠靜態地仔細觀察整張圖片；聲音只能在連續播放的過程中聆聽。

除了聲音之外，本論文還設計了一個 server 與多個 clients 的架構。server 把已經藏在音檔的 sharing 送到多個不同 client，如圖1所示，此方式讓想竊取機密的駭客，必需要能夠同時竊聽不同連線，才能有破解的一絲希望，因此更難去執行。

圖1. Server 與 client

V. 技術背景

A. Shamir's Secret Sharing

Shamir's Secret Sharing 的演算法共有兩個步驟。第一步驟是將原本的機密資料 D 拆成 n 份 sharing，並決定需要幾份 (k) 以上的 sharing 才能還原，故$1 \le k \le n$，並以 k 的值來決定一個多項式的最高冪為 k–1，而零次項的值則為 D。

第二步則從這 n 份 sharing 中湊得 k 份 sharing，並將它還原為原本的祕密，若湊得的份數少於 k 份，則無法還原。拆成 n 份 sharing 與需要 k 份 sharing 才能組回去原本祕密的情境，又稱為(k,n) threshold。

以下將以 threshold 為3，sharing 數 n=5，祕密 D=34為例。在 Shamir's Secret Sharing 的第一步驟中，將產生一個多項式，由於 k 為3，因此需要產生的多項式的最高次方為2，而零次項的值為34，其它次項的值則可隨機產生，以公式(1)為例，

$$F(x) = 4x^2 - 21x + 34 \qquad (1)$$

由於 n 為5，因此需要在這平面上隨機找5個點，F(2)=8、F(3)=7、F(4)=14、F(5)=29、F(6)=52。

而在 Shamir's Secret Sharing 第二步驟中，使用了拉格朗日差值法（Lagrange Interpolation）。如果收集到了3個 sharing，F(3)=7、F(5)=29、F(6)=52，並代入拉格朗日差值法(2)，即可得到以下式子，簡化後即可得出(1)，得出 D=34。

$$F(x) = 7\frac{(x-5)(x-6)}{(3-5)(3-6)} + 29\frac{(x-3)(x-6)}{(5-3)(5-6)} + 52\frac{(x-3)(x-5)}{(6-3)(6-5)} \qquad (2)$$

B. 聲音取樣

在音檔中，以下幾個是控制聲音格式的重要參數：channel 、sample、frame 、sample rate、sample format。

1. channel 代表著總共有幾個聲道，例如若是要輸出在電話的話筒，那麼 channel 數就設定為1，如果要輸出在有左右聲道的耳機，那麼 channel 數就設定為2。

2. sample 為儲存聲音的一個最基本單位，其大小由 sample format 控制，用來表示聲音當下的狀態。
3. frame 為所有聲道，在某個時間點的 sample 所成集合。如果是單聲道，那在某個時間點的一段 frame 裡就只有一個 sample。如果是雙聲道，那在某個時間點的一段 frame 裡就會有兩個 sample。
4. sample rate 為聲音的採樣頻率，單位為赫茲（Hertz，簡寫為 Hz）。sample rate 越高，代表聲音的品質越好，但也代表資料量越大。電話的頻率為八千赫茲，而 CD（Compact Disc）的音質大都是44100赫茲。
5. sample format 為呈現每個 sample 的數值，最常見的有8-bit、16-bit 和32-bit，bit 數越多，能呈現聲音的範圍越大，聲音越準確。

C. Least Significant Bit

Least Significant Bit 簡稱 LSB。指的是在二進位的數字中，最小的位數。由於最小位數的更動，代表著更動後的值只會相差0或1，不容易被發現，因此經常被拿來當作隱寫術隱藏資料的位置。

D. Peak signal-to-noise ratio [8]

Peak signal-to-noise ratio 簡稱 PSNR，用來測量訊號的雜訊比，單位為分貝。其定義如(3)

$$PSNR = 10 \log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (3)$$

以計算聲音的 PSNR 來說，MAX 值為 sample 的最大值。而 MSE（Mean Square Error）代表原始聲音與加入雜訊後的聲音之間的均方差，其定義如(4)

$$MSE = \sum_{i=1}^{n}(C_i - S_i)^2 \qquad (4)$$

其中 C 為原始聲音，S 代表加入雜訊後的聲音。我們用隱寫術藏進聲音中的這些 bit，在這公式中就被視為雜訊。

VI. 實驗架構

A. 決定參數

在此實驗架構中，將會有 server 和 client 兩種角色的架構。首先 server 需要先決定祕密 D，並與 client 約定好(k,n)的值，並由 Shamir Secret Sharing Scheme 的軟體 ssss [9]產生 n 份 sharing 後，等待 client 的連線。

B. server 錄音

當有 client 連上 server 時，server 則可以決定是否要傳送資料。當 server 決定要開始傳送資料時，server 會開始錄音。在本實驗中，將會以 sample rate 為8000、channel 為1，並用 sample format 為 unsinged int8來錄製5~10秒鐘的聲音，用來模擬一則語音訊息。

C. 隱寫術

錄完音後，會把 n 份 sharing 的其中一份藏進錄音檔裡。使用的方式是用 LSB 藏在 sample 裡，本實驗使用的 format 為 unsinged int8，所以每個 sample 就為一個 byte，也就是一個 byte 可以藏入一個 bit。如圖2所示，sharing 為**1-0bf53ca7dd**，長度為12 bytes。在隱藏資料前，我們需要先用一個 byte 來表示所藏資料的大小。假設共有12個字元，12用二進位表示為00001100。之後把00001100這8個 bits 藏進前8個 sample 中。接著把 sharing 用 ASCII 來表示，也就是一個英文字元用8個 bit 表示，所以一個字元會分別藏在8個 sample 裡，以上述的範例來說，則需要96個 sample 來藏。長度連同 sharing，全部共會用到8+96=104個 sample。

圖2. 隱藏範例

D. 包進 WAV 檔

把已經藏好的 sample，包成 WAV 檔，並使用 TCP 把此 WAV 檔送到 client。之後，server 又回到第二步，等待下一個 client 連線時送出下一份 sharing，直到所有 sharing 被領取完畢。

E. client 解析封包

client 收到 WAV 檔後，會先把所有的 sample 從 WAV 檔取出來，並從前八個 sample 取得 sharing 的長度。取得長度之後，就能知道總共要再取幾個 sample，以得出 sharing。

F. 取回祕密

在 n 個 client 中，他們手上都會有不同的 sharing。只要聚集 k 份以上的 sharing，就能透過 ssss [9]還原出祕密 D。

VII. 實驗結果

在實驗的步驟二裡，我們模擬了語音訊息傳送的情境，使用格式為 unsinged int8，channel 為1，並由 ssss

以(k,n)為(3,5)、產生 bytes 為12的 sharing 來藏入104個 sample 裡。以下 server 將分別測量錄音5~10秒，每次固定一個秒數，發送5個 sharing，並使用(3)與(4)比較它們之間的 MSE 和 PSNR。因為使用的 format 為 unsinged int8，所以 MAX 的值為255。結果如表 I、圖3和圖4所示

表 I
錄音秒數與 MSE 和 PSNR 對照

秒數	Sample 數	MSE	PSNR
5	40000	0.0012450	77.197667
6	48000	0.0011041	77.704725
7	56000	0.0009178	78.512879
8	64000	0.0008000	79.113535
9	72000	0.0007333	79.482942
10	80000	0.0006575	79.963869

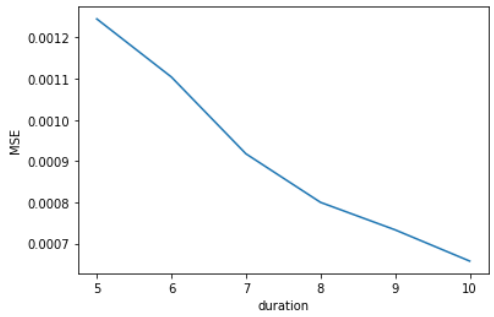

圖3. MSE 比較

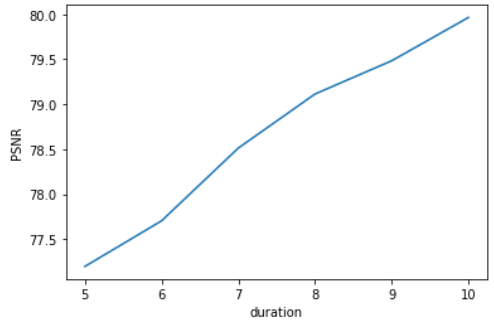

圖4. PSNR 比較

VIII. 結論

本實驗用來隱藏 sharing 的音檔皆是由 server 接收到 client 的連線後才開始錄音的，每次傳出去的音檔都是隨機的。因此，當有多個 sharing 要發送到不同地方時，每次發送的音檔都是不一樣，可用此來降低攻擊者的疑慮。

發送音檔的目的地皆不相同，因此要在多個接收端去攔截信號，且能夠偷到一定數量（k）以上的音檔，又為攻擊者加上一層阻礙。

IX. 未來展望

在聲音傳送的部分，用 TCP 來傳送仍不是最常見的形式，通常即時性的語音、視訊不會使用 TCP 來傳送，而是使用 UDP 搭配 RTP 來傳送。

但 UDP 和 RTP 並不像 TCP 一樣，能夠確保封包完整到達。如果在網路環境不好的情況下，封包傳送錯誤時，經常需要整段重傳，反而容易吸引攻擊者的注意。為了確保重要的 sharing 能夠傳達到目的地，未來考慮加入錯誤更正碼在 sharing 中，即使 server 在傳送過程中出現了錯誤，client 也能夠自動修正，從而還原出正確的 sharing。

參考文獻

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *Managing Requirements Knowledge, International Workshop on*, 1979: IEEE Computer Society, pp. 313-318.

[3] F. Al-Shaarani and A. Gutub, "Securing matrix counting-based secret-sharing involving crypto steganography," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[4] S. Porwal and S. Mittal, "A threshold secret sharing technique based on matrix manipulation," in *AIP Conference Proceedings*, 2020, vol. 2214, no. 1: AIP Publishing LLC, p. 020020.

[5] A. Gutub, N. Al-Juaid, and E. Khan, "Counting-based secret sharing technique for multimedia applications," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5591-5619, 2019.

[6] H.-D. Yuan, "Secret sharing with multi-cover adaptive steganography," *Information Sciences*, vol. 254, pp. 197-212, 2014.

[7] H. Wang and S. Wang, "Cyber warfare: steganography vs. steganalysis," *Communications of the ACM*, vol. 47, no. 10, pp. 76-82, 2004.

[8] M. Tayel, A. Gamal, and H. Shawky, "A proposed implementation method of an audio steganography technique," in *2016 18th international conference on advanced communication technology (ICACT)*, 2016: IEEE, pp. 180-184.

[9] B. Poettering. "ssss" (https://linux.die.net/man/1/ssss)