

# SVG 向量圖形資訊隱藏之研究

## Research of SVG Images Information Hiding

葉耀明

國立台灣師範大學資訊教育系  
ymyeh@ice.ntnu.edu.tw

吳秉鋒

台北市立華江高中圖書館  
pingfeng@tp.edu.tw

### 摘要

由於網際網路的快速成長，對於保護影像智慧財產權的需求與日遽增。影像資訊隱藏於數位內容典藏的應用已有多年的發展，但關於可變動向量圖形 (Scaleable Vector Graphic, SVG) 智慧財產權的保護機制，尚處於萌芽階段。並且因其圖形格式是以明文方式存在，更增加了保護的難度。

有鑑於此，本研究提出了一種資訊隱藏模型於可變動向量圖形保護上，資訊融入法 (Information Mixture)，其具有下列兩個特點：(1) 利用純文字明碼加密的技巧來隱藏資訊，讓合法使用者能驗證資訊 (2) 結合 Unicode[2] 編碼方式，提供亞洲字元驗證資訊的能力 (3) 系統實作在 JAVA 技術上可跨平台應用，結合 PKI 與 Client-Server 架構，提供一個針對 SVG 向量圖的線上智慧財產權保護機制。本方法使隱藏資訊後的 SVG 圖檔與原始檔案，執行時在瀏覽器端顯示出來的圖形完全一致，使用者也無法感覺顯示速度上的差異。

關鍵字：資訊融入法、可變動向量圖形、資訊隱藏

### ABSTRACT

With the rapid growing popularity of

the World Wide Web (WWW), the need of Intellectual Property Rights (IPR) for images is becoming steadily on the increase. The application of Image Information Hiding for Digital Content Archives has been developed for many years, but development of protecting mechanism for the IPR of Scaleable Vector Graphic (SVG) is just the beginnings. Since the format of SVG files is stored in the Plain Text type, the degree of difficulty in protecting files is getting higher.

Based on these facts, in the study, we propose one Information Hiding models for protection on SVG, the Information Mixture, respectively. It have the following two features. (1) The legal user can verify the information by the encrypted techniques of the plain-text. (2) Providing the ability to verify the information in Asia characters by combining the Unicode encoding. (3) Providing the mechanism of online protection of IPR by combining the public key infrastructure (PKI) and Client-Server architecture. In addition, between the original SVG file and the SVG file of hiding authenticated information, the visual perception is consistent and the execution speed is almost the same.

Keywords : Information Mixture、  
Scaleable Vector Graphics、Information  
Hiding

## 一、簡介

現今趨勢在著作權法倡導下，著作權意識抬頭，音樂有著作權、圖畫有著作權、文字著作都有著作權，凡是創作的作品都有著作權，相對就有保護著作權的相關措施，法律面不在此討論，也有很多人討論電腦圖形的著作權如何保護，但都侷限於傳統靜態點陣圖形的研究[3]，本研究特別針對 W3C 新制訂的網頁向量圖形格式 SVG（可變式向量圖形），設計出一套隱藏版權資訊機制：

1. 研究如何定義各種 SVG 圖形模式，使之可將特定受保護的資訊加入，藉此隱藏版權資訊。
2. 圖形創作者能自行決定要隱藏的資訊內容，能植入此特定資訊，過程中能加入私鑰，並取得公鑰，使呈現在螢幕上的圖形毫無分別。
3. 建立一網路平台，供圖形創作者登記版權，能準確分辨出特定隱藏的資訊，而後並可由此平台程式來提取特定資訊，擔任仲裁機構。
4. 提出一個隱藏字元成本（Cost of Hiding Character）的計算公式。

一般向量圖形，多由各大軟體廠商制訂封閉的格式，著作權的問題清楚而容易解決，只要圖形雷同，很容易由各家製作廠商的獨門格式辨認出是誰的著作權，特殊的封閉格式只有原軟體才能解譯，甚至在圖形完成時，就允許創作者直接存成二進位格式，此方式通常是電腦作業系統認可的可執行檔，一般使用者縱使取得原圖檔，還是無法任意修改或重製的。但 SVG 向量圖形是明碼文字格式，網路上容易取

得，也容易複製、修改，要在其中加入一些特定格式資訊，並且隱藏起來，適當時候能取出資訊，就能證明這個 SVG 圖形的著作權屬誰。

所以本研究的範圍與限制為：

1. 以 SVG 明碼圖檔為受藏資訊檔案，但檔案大於 0.7K Byte 以上，且內含同一種 tag 四個以上，（或某一種 element 配多個 attribute），視檔案內容可隱藏不同數量資訊。
2. 所隱藏資訊以可見字元為範圍，ASCII 碼 33~126。
3. 可接受雙位元語系字形，如中文簡繁體、韓文、日文等資料隱藏。
4. 開發程式以 Java 為主，搭配伺服器端 Apache+Tomcat 程式為平台後端。
5. 使用者以任何作業系統的圖形介面如 MacOS、Linux、Windows、OS2...，透過網際網路皆可。

## 二、可變動向量圖形（SVG）

### 2.1 簡介

SVG(Scalable Vector Graphics), 1.0 版由 W3C 於 2001 年 9 月正式正式推薦，是一種以 XML [1] 語言為基礎的新圖檔標準，主要用來改善網頁圖片的品質，可以讓瀏覽器上的圖檔更有效率、更多效果。SVG 可讓網頁上顯示出各種高品質的向量圖形，同時也支援文字、動畫、遮罩效果，對幾何圖形色彩、透明度、變形、修剪路徑、濾鏡效果都有充分支援。

網站的設計通常以點陣圖來處理圖形，若要達到檔案小、傳遞快，最好還有動畫，則考慮使用向量圖形，新興於網路

的 2D 向量圖格式有兩種，其一為 SVG 圖形格式，從瀏覽器觀賞網站的 SVG 圖形，需透過 Adobe 公司的外掛軟體 SVG Viewer；另一個是 swf 圖形格式，也需要在瀏覽器安裝 Macromedia 公司的 Flash player。這兩種向量圖形各有優缺點，前者是開放標準的純文字向量式向量圖形描述語言，後者佔有廣大市場，但為封閉格式。

## 2.2 SVG 的特色

由於 SVG 是 XML 的子集合，因此具備了一般網頁傳統使用圖形如 JPEG、PNG、GIF、BMP 等所沒有的特性：檔案小、傳輸速度快，還可支援下列各項：Type 1 及 TrueType 字型、XML（延伸標記語言）、CSS（樣式表）、互動式動作，以及動態 HTML 動畫。

對 SVG 而言，最大的優點當然是和其它開放標準相容，因此 SVG 擁有的許多特色分別介紹如下：

### 1. 基於 XML 標準

微軟的 office 套裝軟體無疑是最成功的商用軟體，而大量支援 XML，是微軟標榜 Office 2003 的賣點之一，軟體巨人的行動已經承認 XML 是下一代網路標記語言，與 HTML 一樣，XML 也源 SGML，它擁有 HTML 語言所缺乏的伸縮性與靈活性。SVG 設計之初就是以 XML 為基礎，這使得它能同 HTML、CSS、DOM、XSL、JavaScript、CGI 一樣成為一種新的標準。

### 2. 由純文字構成的圖形

SVG 是一種純文字構成檔案，經過瀏覽器解析後產生圖形。也就是說，可以不用任何影像處理工具，僅僅用文字編輯器就可以產生一個 SVG 圖形。

### 3. 向量圖形

SVG 向量圖形物件是由線框和填充物所組成，由電腦根據向量資料進行計算，然後繪製而成，不會因為縮放或解析度問題而有點陣圖形的鋸齒狀。因此向量圖形相對於點陣圖有以下特色：

(1) 文件的大小與圖形的純文字內容有關，而與圖形的具體尺寸無關。

(2) 圖形的顯示大小可以放大、縮小，變化後不影響圖形檔案內容影響，所以在圖形檔內容文字不多的情況下，向量圖形具有儲存文字量小，又可縮放的優點。

### 4. 開放式標準 (Open Standard)

SVG 之所以會出現是為了在網路上制定一種新的、大家認同的、可擴展的、開放的圖形格式。這個標準是由許多的廠商和 W3C 所共同制定的，所以 SVG 就是一個開放性的產業標準。

### 5. 多種物件格式靈活運用的文件

SVG 檔案由三部分組成：向量圖形、文字和影像，可以直接使用影像，也可以產生向量圖形和文字物件，還可以引入影像而製作出任何其他格式圖形所能達到的效果。由於文件格式是純文字形式，因此可以很容易重新修改，而且其中的圖形文字描述還可以在一個圖檔內重覆使用。

### 6. 可提供圖形檢索

一般點陣圖，只能憑檔名在網際網路上被瀏覽器、搜索引擎檢索，同一圖形可能重複命名，而相同命名卻又內容不同圖形，所以網路上搜尋文字容易，而搜尋適用圖片卻難，而 SVG 內嵌文字可供精確搜尋，此為 Flash 圖檔所不及。

### 7. SMIL 支援

製作網頁用的互動按鈕，其圖片和程式在以前是分開的，按鈕僅僅是一個圖片，按鈕的程式是由網頁的 Script 來作。SVG 改變這個規則，它經由 SMIL (Synchronized Multimedia Integration Language)，使得在圖片內時序元件可控制，也就是說 SVG 支援了互動性的網頁效果。

## 8. 內嵌字元資料

製作網頁的技術在 DHTML (Dynamic Hypertext Markup Language) 中，可使用系統中沒有的字體，該字體在瀏覽時，可透過 Web 即時下載，但是這對英文系統來說是適合的，因為英文字體的檔案大小一般在幾十 KB 左右，而對於中文系統來說，這種方法不適合，因為任何一種中文字形的檔案大小都在 2MB 以上甚至超過 5 MB，一般使用者不可能為了觀看幾個中文字而浪費大量時間去等待。SVG 採用了一種動態字體的方法，它嵌入了圖形中所出現的文字字型資料，用戶不需要下載所有的字形，這對於中文使用者來說是相當便利的，尤其用在圖書館典藏系統，當 BIG5 字碼不敷使用時，有新的方法可以解決中文問題。

## 三、資訊融入法

### (Information Mixture)

本章主要是分析 SVG 檔案內隱藏資料的方法，資訊隱藏 (steganography or information hiding) 是一種加密的方法，把重要資訊隱藏到一般資訊中。不同於純粹對重要資訊直接加密，資訊隱藏是藉由一個不會令人起疑的媒介，將機密資料隱

藏其中。相對於直接加密，資訊隱藏的方式更具有保密的效果，如果再搭配既有的一些加密方法，將欲保密的資料先作一層加密的動作，再隱藏至一般資料中，則保密效果就更好了，等於有了雙重的保密。

本研究提出資料隱藏法稱為 Information Mixture (資訊融入法)，將想要隱藏的資訊，以數值方式編碼混入圖檔程式碼內，和 SVG 元素的屬性數值融合在一起，外觀上無法察覺，需經由特定程式，才能萃取出有意義的資訊，再經由解碼程式判別出訊息。以此方法製作的 SVG 文件，內含 Invisible Watermark，適合隱藏圖檔版權資訊，具有實用價值。

本系統隱藏資訊使用字元範圍 ASCII 碼從 33~126，此為可見字元的一部份，目的為方便由鍵盤輸入，應用於 SVG 常用元素有下列六種更換 Attribute 值的方式：

- 轉置隱藏 (hide transform)
- 位移隱藏 (hide Move)
- 動畫隱藏 (hide Motion)
- 顏色隱藏 (hide Color)
- 外形隱藏 (hide Shape)
- 矩陣隱藏 (hide matrix)

### 3.1 轉置隱藏 (hide transform)

一般 SVG 圖檔在瀏覽時按下右鍵，可縮放四次，考慮圖形放大四次(16 倍)的情況下而不失真，更換 Attribute 的值，將字元編碼放在小數點下第四位和第五位，一些特定 SVG 元素對於 Attribute 的值採四捨五入取整數值，不影響原始數值資料在圖形上的表現。

假設座標  $A(x, y)$  在轉置隱藏下藏入數值  $\delta$ ，新座標  $A'(x+\delta, y-\delta)$ ，若放大 16 倍得座標  $A(16x, 16y)$  和新座標  $A'(16x+16\delta, 16y-16\delta)$ ，將  $16\delta$  控制小於

0.1(亦即 $\delta$ 小於0.00625),則 $16x+16\delta$ 和 $16y-16\delta$ 的值,將與 $16x$ 和 $16y$ 相當接近,在瀏覽器上A和A'呈現位置就難以分辨,例如座標A(20.786, 30.129)在轉置隱藏下,藏入資料68而得到新座標A'(20.786+0.00068, 30.129-0.00068) = (20.78668, 30.12832),若同時放大16倍得座標A(332.576, 482.064)和A'(332.58688, 482.05312),在瀏覽器上A和A'呈現位置都和座標(332.6, 482.1)相同,原始SVG程式片段如下:

```
<path d="M-40-30l 80 0l 0 60l-80 0l
0-60Z" style="fill:rgb(255,255,255);
stroke:rgb(0,0,255);"
transform="translate(20.786,30.129)"
></path>
```

隱藏資料的過程依照DOM Tree 逐項遊走遍歷element → attribute → methods → 改變Value,其中element可以是path、use、text或g等,attribute則是transform,methods可以是scale、translate、opacity等,attribute value可以兩位座標或單獨一個數值,隱藏資訊後程式片段如下:

```
<path d="M-40-30l 80 0l 0 60l-80 0l
0-60Z" style="fill:rgb(255,255,255);
stroke:rgb(0,0,255);"
transform="translate(20.78668,
30.12832)" ></path>
```

反解時DOM Tree 抓出20.78668和30.12832,依照特徵值取出68(以32比對驗證),還原成對照的可見字元,每一個元素的Attribute值可供轉置隱藏一個字元,第二個以後的字元隱藏在下一個元素,直到所有資訊隱藏完畢。

### 3.2 位移隱藏 (hide Move)

SVG的圖檔移動路徑,由A點直線移到C點,可以在直線路徑A、C之直線上增加B點,那麼由A點經B點到C點,或

由A點超越C點先到B點再回頭到C點,在瀏覽器畫面上產生的圖形路徑不變,都只看到結果是由A點移動到B點。

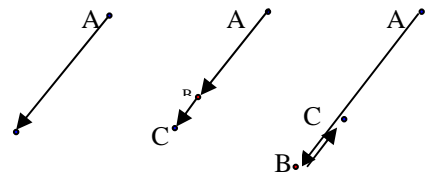


圖 3-1 位移隱藏示意圖

如果一個SVG圖檔是由很多路徑組合而成,可經由亂數取足夠位置,在眾多路徑中各自做位移隱藏,原始SVG程式片段如下:

```
<path style="fill:none; stroke:blue;"
="M512.5234375 428.349365 L525.
46875 406.698730" >
```

原始路徑要移動到絕對座標C(512.5234375,428.349365)然後畫出一條直線到D(525.46875 406.698730),現在藏入藏入資料67資料而得到新座標B(579.234375 428.349365),由B點到C點則是向量(-67, 0),因為向量AC=向量AB+向量BC,後程式片段如下:

```
<path style="fill:none; stroke:blue;"
d="M579.234375 428.349365 m-67
0 L525.46875 406.698730">
```

### 3.3 動畫隱藏 (hide Montion)

動畫時間以秒計時,增減一點點延遲時間是察覺不出來的,起始、結束的時間值同時加上延遲時間,則動畫經歷時間長度相同,或是增加或減少動畫經歷時間長度,只要保留起始、結束其中一個時間值。將隱藏字元的編碼放在小數點下第三位和第四位,些微差距不影響原始數值資料在動畫上的表現,原始SVG程式片段如下:

```
<animateMotion calcMode="paced"
dur="8.0S" repeatCount="indefinite"
begin="6.0S" rotate="auto">
```

動畫起始時間是在第 6 秒，歷經 8 秒動畫後於 14 秒時停止，藏入數值資料 67 後，調整起始時間是在第 6.0067 秒，歷經 7.9933 秒動畫後，依然於 14 秒時停止，隱藏資訊後程式片段如下：

```
<animateMotion calcMode="paced"
  dur="7.9933S" repeatCount="indefinite"
  begin="6.0067S" rotate="auto">
```

### 3.4 顏色隱藏 (hide Color)

RGB 三顏色數值為整數，對於小數點部分採取無條件捨去，將字元編碼放在小數點下第二位以後，繪圖填色時會自動忽略，不影響原始數值資料在顏色上的表現，原始 SVG 程式片段如下：

```
<line x1="20" y1="40" x2="220"
  y2="40" style="stroke:rgb( 255 , 127 ,
  63 ); stroke-width:4"/>
```

一次藏入 3 個數值資料 76、68、9 後，隱藏資訊後程式片段如下：

```
<line x1="20" y1="40" x2="220"
  y2="40" style="stroke:rgb(255.076 ,
  127.068 , 63.009); stroke-width:4"/>
```

### 3.5 外形隱藏 (hide Shape)

SVG 圖形的外形物件元素都有固定的屬性，其屬性值一般是配對座標如中心  $x$  和  $y$ 、長和寬、橢圓長軸半徑和短軸半徑、圓角矩形 Roundness- $x$  和 Roundness- $y$  等，依照 DOM Tree 逐項遊走遍歷 element  $\rightarrow$  attribute  $\rightarrow$  改變 Value，其中 element 可以是 Line、Rect、Circle、Ellipse 或 Text 等，attribute 則是固定配對格式如  $(x1, y1)$ 、 $(rx, ry)$ 、 $(cx, cy)$ 、 $(width, height)$ 、 $(x, y)$  等，原始 SVG 程式片段如下：

```
<rect x="50" y="100" width="150"
  height="20" style="stroke:#FF0000;
  fill:#CCCCCC"/>
```

每次藏入 1 個數值資料，本例將字元編碼 76 藏於小數點第二、三位數，不影響

原始數值資料在 SVG 圖形外形上的表現，隱藏資訊後程式片段如下：

```
<rect x="50.076" y="100.076"
  width="150"
  height="20" style="stroke:#FF0000;
  fill:#CCCCCC"/>
```

### 3.1.6 矩陣隱藏 (hide matrix)

元素 matrix 在 SVG 圖形中包含剛性運動和非剛性運動，剛性運動又稱保距變換，包含平移、旋轉和鏡射三種基本模式，非剛性運動指的是伸縮。

SVG 中所有的矩陣轉換都可以用一個

$3 \times 3$  矩陣  $\begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix}$  來表達，矩陣中真正

有用的數據只有六個，所以在 SVG 中用  $\text{matrix}[a \ b \ c \ d \ e \ f]$  來表示，舉例來說：

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ 表 matrix}$$

$[\cos \theta \ \sin \theta \ -\sin \theta \ \cos \theta \ 0 \ 0]$  或

$\text{rotate}(\theta)$ ，將  $x$  軸旋轉  $\theta$  度，座標  $(x, y)$  變成  $[\cos \theta - \sin \theta, \sin \theta + \cos \theta]$ 。

如果多個巢狀矩陣經過運算合併最後的結果還是一個矩陣，稱為 CTM (the current transformation matrix)，其中

$$\text{CTM} = \begin{bmatrix} a_1 & c_1 & e_1 \\ b_1 & d_1 & f_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_2 & c_2 & e_2 \\ b_2 & d_2 & f_2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \dots \cdot \begin{bmatrix} a_n & c_n & e_n \\ b_n & d_n & f_n \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_{\text{舊座標}} \\ y_{\text{舊座標}} \\ 1 \end{bmatrix} = \text{CTM} \cdot \begin{bmatrix} x_{\text{新座標}} \\ y_{\text{新座標}} \\ 1 \end{bmatrix},$$

本系統利用 matrix 來隱藏資訊，先假設 SVG 文件原來的 matrix 中有六個數值

分別是 a,b,c,d,e,f，記成 M ( a,b,c,d,e,f)，而要藏的資料假設 6 個字元為 m,n,o,p,q,r，放在 matrixA ( m,n,o,p,q,r) 之中，令 matrix B = (am+cn,bm+dn,ao+cp,bo+dp,aq+cf+e,bq+dr+f)，令 matrix C =

$$\begin{pmatrix} p & -n & -o & m & or-pq & -mr+nq \\ mp-no & mp-no & mp-no & mp-no & mp-no & mp-no \end{pmatrix}$$

$$\text{則 } \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} am+cn & ao+cp & aq+cf+e \\ bm+dn & bo+dp & bq+dr+f \\ 0 & 0 & 1 \end{bmatrix} \cdot \frac{1}{mp-no} \begin{bmatrix} p & -o & or-pq \\ -n & m & -mr+nq \\ 0 & 0 & mp-no \end{bmatrix}$$

即 matrix A=matrix(B • C)，可以用兩個矩陣 B 和 C 來替代原來矩陣 A，這樣得到相同結果，原始 SVG 程式片段如下：

```
<rect transform="matrix(1,1,0,1,50,100)"
x="50" y="80" width="150"
height="75" style="fill: #FFC; stroke:
#F00; stroke-width: 1.5px"/>
```

原本矩陣內資料 (a,b,c,d,e,f) = (1,1,0,1,50,100)，假設要藏入的資訊是 (m,n,o,p,q,r) = (1,2,3,4,5,6)，則 matrix B = (1,3,3,7,55,111)，matrix C = (-2,1,1.5,-0.5,1,-2)，隱藏資訊後程式片段如下：

```
<rect transform="matrix(1,3,3,7,55,111)
matrix(-2,1,1.5,-0.5,1,-2)" x="50" y="80"
width="150" height="75" style="fill:
#FFC; stroke: #F00; stroke-width:
1.5px"/>
```

因為

$$\begin{bmatrix} 1 & 0 & 50 \\ 1 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 55 \\ 3 & 7 & 111 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -2 & 1.5 & 1 \\ 1 & -0.5 & -2 \\ 0 & 0 & 1 \end{bmatrix} \text{代}$$

表 matrix(1,1,0,1,50,100) 運算等於 matrix(1,3,3,7,55,111)matrix(-2,1,1.5,-0.5,1,-2)，兩個矩陣合併運和前面單一矩陣有一樣的結果，所以原 SVG 圖形在瀏覽器上顯示結果相同，完全不會改變。

一個 SVG 文件中，令 M 為原始矩陣，

$$M = \begin{bmatrix} a_0 & c_0 & e_0 \\ b_0 & d_0 & f_0 \\ 0 & 0 & 1 \end{bmatrix}$$

在此文件中要插入雙位元文字 6n 個，分成 n 組每組藏 6 個編碼後的文字，[ a<sub>i</sub>, b<sub>i</sub>, c<sub>i</sub>, d<sub>i</sub>, e<sub>i</sub>, f<sub>i</sub> ]，亦即

$$A_i = \begin{bmatrix} a_i & c_i & e_i \\ b_i & d_i & f_i \\ 0 & 0 & 1 \end{bmatrix}, i \in [1..n]$$

則原始矩陣 M 改成 MA<sub>1</sub>•CTM•A<sub>n</sub><sup>-1</sup>，

還原隱藏資料 A<sub>k</sub> 的演算法如下：

```
Temp ← An-1
If k = n
then Return Temp-1
i ← n
while i ≠ k
do
i ← i-1
Temp ← (Ai-1•Ai+1)•Temp
If i = k
then Return Temp-1
```



#### 四、系統實作

當圖形創造者製作出原始 SVG 圖檔，以圖片形式放置在網頁上使用，在未經授權許可下很可能被盜用卻沒有任何保障，此時經由網際網路將 SVG 圖檔送到本系統 Server 端，使用者在 Client 端自選私

鑰，輸入想要隱藏資訊方式及內容，上傳到 Server 端後，即時產生加密檔送回 Client 端，同時 SVG 檔案內已經包含伺服器產生的公開金鑰和隱藏的資訊。此時送一份已經受保護的 SVG 圖檔到版權登記機構註冊，便可在版權保障下公開使用此圖檔。其 SVG 隱藏資訊系統流程如下圖 4-1：

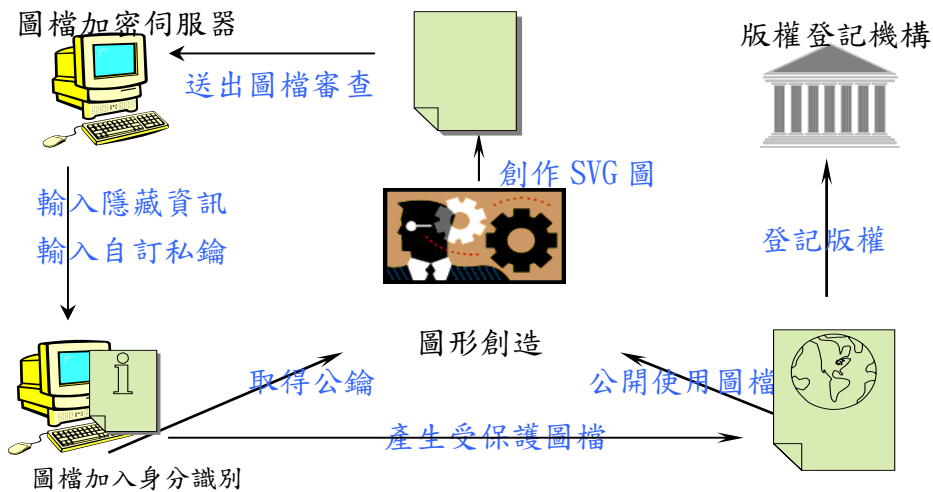


圖 4-1 SVG 隱藏資訊系統流程

根據以上流程，本研究實作出 SVG 資訊隱藏系統具有下列三大功能：

- 線上即時處理：透過網際網路，可即時取得已經隱藏資訊的 SVG 圖檔。
- 隱藏資訊多樣化：可自選想隱藏的字元、多國語言文字、圖形、動畫。
- 符合 PKI 架構：使用者可自訂私鑰，系統 Server 亂數產生公開金鑰。

Information Mixture 系統實際操作畫面如下圖 4-2、4-3、4-4：

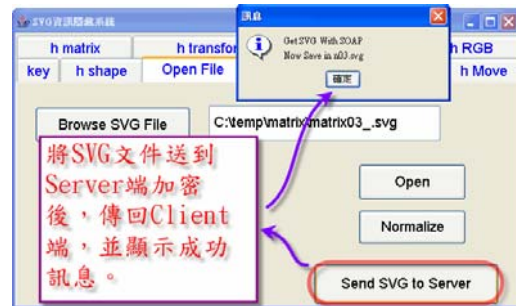


圖 4-2 Server 端加密送回 Client

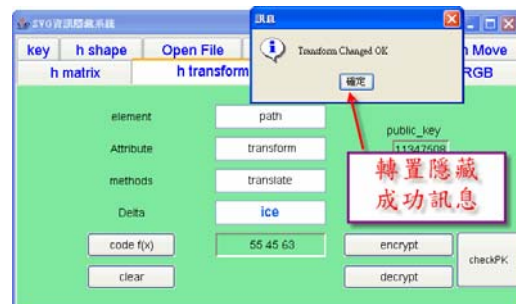


圖 4-3 Information Mixture 隱藏實作



亞洲語系雙位元字母，以 Unicode 編碼後隱藏資訊，再經由 hide 按鍵寫入 SVG 程式碼中，可以隱藏中文正體、簡體、韓文、日文。



圖 4-4 雙位元語系文字隱藏實作

## 五、分析

一個 SVG 檔案能夠藏多少資訊，要看這個 SVG 文件的內容而定，就 Information Mixture 方法能隱藏資訊的數量而言，可分為無限字元隱藏和有限字元隱藏：

表 4-1 Information Mixture 隱藏字元數量比較表

方法 數量	矩陣 隱藏	轉置 隱藏	位移 隱藏	動畫 隱藏	顏色 隱藏	外形 隱藏
有限		√		√	√	√
無限	√		√			

並不是每一個 SVG 檔案都能作矩陣隱藏或位移隱藏，必須依據文件內部是否有 matrix 和 move 元素而定，至少要有一個，那麼資訊便可無窮盡藏入，甚至超越本身檔案大小的資料都可以。本研究以每次增加 6 個隱藏字元，比較矩陣隱藏檔案大小與隱藏資料量的關係，如下圖 4-1：

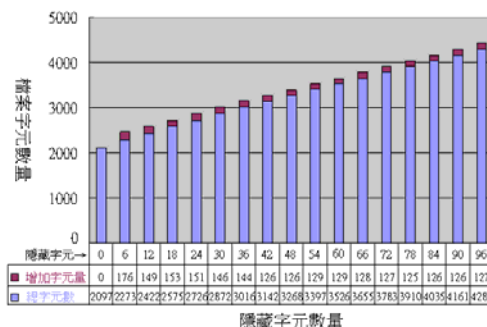


圖 4-1 Hide Matrix 隱藏字元增加數量與檔案大小

計算檔案大小不以所佔磁碟空間為依據，SVG 文件中空格如同 HTML 不會影響檔案執行結果，但會使檔案變大，因此不計檔案內空白字元、空白行距，純粹計算文件內有效字元。為了方便計算每隱藏一個字元，文件中需增加多少掩護的字元，本研究提出一個隱藏字元成本 (Cost of Hiding Character) 公式：

$$CHC = \frac{C_{\text{after}} - C_{\text{before}}}{H_{\text{number}}}$$

其中  $C_{\text{after}}$  表示文件隱藏資訊後字的元數， $C_{\text{before}}$  表示文件隱藏資訊前的字元數， $H_{\text{number}}$  表示隱藏的字元數。例如隱藏 NTNU 四個字元到一個原有 4928 字元的 SVG 檔案內，藏入後檔案共有 5002 個字元，即多出 74 的字元來隱藏此『NTNU』，則隱藏字元成本  $CHC = \frac{5002 - 4928}{4} = 18.5$ 。

不同的檔案，適用不同的方法，轉置隱藏、動畫隱藏、顏色隱藏、外形隱藏，只能藏入有限的資訊，其隱藏的容納量與該檔案相關的 element 數量成正比，而位移隱藏和矩陣隱藏可以融入非常大量的資訊。矩陣隱藏的 CHC 值雖然高於位移隱藏，但藏入資訊後的隱蔽狀態也優於位移隱藏，下表為 Information Mixture 的各種隱藏

方法在不同檔案下，隱藏相同數目字元的 CHC 值：

表 4-1 Information Mixture 隱藏字元 CHC 成本比較表

方法 數量	hide transform	hide Move	hide Montion	hide Color	hide Shape	hide matrix
6	10	11.33	10	5	8	29.33
12	10	11.33	10	5	8	27.08
18		11.56				26.56
24		11.00				26.21
30		10.73				25.83
36		10.67				25.53
42		10.60				24.88
48		10.52				24.40
54		10.54				24.07
60		10.57				23.82
66		10.67				23.61
72		10.69				23.42
78		10.69				23.24
84		10.68				23.07
90		10.66				22.93

## 六、 結論

本研究中我們對 SVG 檔案隱藏資訊提出 Information Mixture (資訊融入法) 的方法，針對一般可見字元 (ASCII33~126) 提出 6 種可利用的隱藏方法：轉置隱藏、位移隱藏、動畫隱藏、顏色隱藏、外形隱藏、矩陣隱藏。

在隱藏資訊的過程中，以明碼隱藏文字密文，將文字密文的內碼重新編碼，減少被識別出的機會，縱然找到編碼後的密文也不容易還原。Information Mixture 方法將資訊融入在一般數據中，更不容易被發現，可充分達隱蔽的效果，尤其 Hide Matrix 還能以 Unicode 藏大量中文，相當於內含 Invisible Watermark，拿來處理 SVG 圖檔的版權資訊很實用。

程式開發以 JAVA 為主，基於 XML 開放性的標準，具跨平台可擴充的特性，透過網際網路達到即時隱藏資訊的目的。系統採 Client-Server 架構，使用者從 Client 端上傳 SVG 圖檔，選擇想要隱藏的資訊和方式，輸入私鑰後，上傳到 Server 端，由 Server 端產生公開金鑰後，結合所有相關資料產生隱藏資訊後新的 SVG 圖檔，並傳回 Client 端，使用者以此加密後的圖檔用於網際網路上。

本系統將可用於 SVG 圖檔版權認證，若由公正的第三者或政府機關擔任 Server 端服務，同時提供圖檔註冊機制，使用者申請 SVG 圖檔專利認證後，將圖檔和公開金鑰均公告於此認證機構，比照現行智慧財產管理，供一般民眾日後申請、查閱，當有版權紛爭時，可拿出私鑰和圖檔，請求法律上的仲裁。

## 七、 參考文獻

- [1] 林錦雀,江高舉,"XML 基礎領航",金禾圖書,民國 90 年 5 月。
- [2] 曾士熊,"Unicode 與 ISO10646",中央研究院 Newsletter of Computing Center 計算中心通訊,資訊話題,第 16 卷 10 期民國 89 年 05 月。
- [3] 鄧慶齡,許成之,"應用於數位學習之版權管理系統規劃與建置",第三屆 DATF 數位典藏技術研討會,民國 93 年 8 月。