第 4 章 系統設計與分析

SVG 結構化資料檔案是 XML 的純文字格式的一種，以往的研究鮮少對文字格式檔案隱藏資訊，本研究希望透過 java 程式語言相關技術，建制 SVG 資訊隱藏的互動環境，讓使用者在本系統的輔助之下進行資訊隱藏，本研究提出的 SVG 資訊隱藏系統具備以下特色：

- 具公開金鑰架構（Public Key Infrastructure，PKI）架構。
- 線上即時認證。
- 線上即時加密。
- 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。
- 可接受圖形曲線、動畫、遮罩、標籤等資料隱藏。
- 開發程式以 Java 為主，搭配伺服器端 Apache+Tomcat+PHP 網頁為平台後端。
- 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。

4.1 系統環境與開發軟體

本研究採用具跨平台特性的 JAVA 程式語言，將 Client 端與 Server 端之間透過 internet 傳送，以純文字檔案傳遞訊息，文字檔案內以明碼隱藏文字密文，而文字密文可代表 SVG 的圖形、文字或動畫，進而透過網際網路達到即時隱藏資訊。基於 JAVA 跨平台的特性與 XML 開放性的標準，本研究將具備高度擴充性。

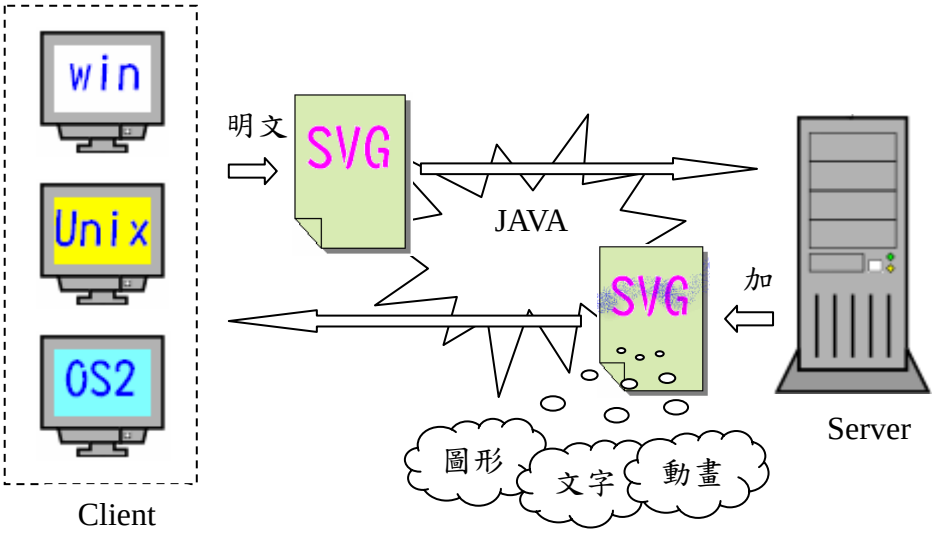

圖 4 - 1 SVG 資訊隱藏系統環境示意圖

本系統是以 Borland JBuilder 為主要開發工具，Borland JBuilder 是跨平台 Java 開發環境，可建構符合業界標準之 Java 應用系統，開發 EJB、Web、XML 以及資料庫等各類應用程式。藉由 JBuilder 雙向、視覺化之設計工具，本系統得以快速開發各種 SVG 資訊隱藏應用程式，並部署至多種應用程式伺服器，Server 端可架設於 Windows、OS2、Linux 或 Unix like 系統，Client 可使用圖形介面的瀏覽器於任何平台，並結合結合公開金鑰與私鑰以增加使用者資料的安全性。

4.2 擷取 SVG DOM Tree

XML 格式的 SVG 文件利用 Apache Batik 套件來解析成 DOM（Document Object Model）Tree，DOM 必須把整份 SVG 文件放進記憶體，然後以樹結構予以儲存，如果 SVG 檔案太大，則可能使作業系統出現記憶體不足的警告，使用 DOM 需要的記憶體和 SVG 文件的大小和複雜度成正比。

SVG 文件轉換 DOM Tree 過程分為兩步驟：

1. 首先是根據 SVG 文件構造源樹，然後根據 XSL 規則將源樹轉換為結果樹。目前，這種轉換協議已經日趨完善，並從 XSL 中獨立出來，成為 W3C 正式推薦的標準，稱為 XSLT（XSL Transformations）；
2. 生成結果樹，就可以對其進行解釋，產生一種適合顯示、列印或是播放的格式，這一步稱為格式化（Formatting）。

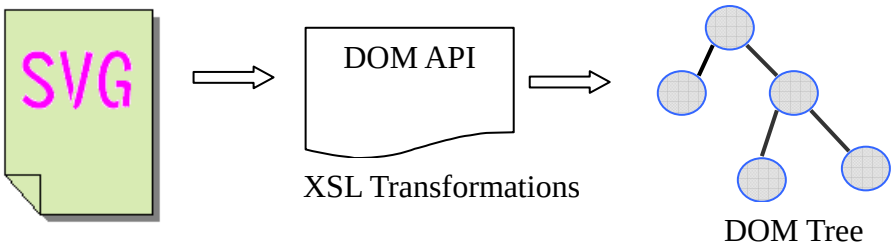

圖 4 - 2 SVG 樹狀模型產生程序

本研究使用 Batik 處理器負責實現轉換過程，首先 SVG 文件被解析成 DOM 樹存放在系統記憶體內，接著對文件進行分析，每一個 DOM 樹中的節點都會與一個隱藏資訊的模式相比較，當二者匹配時，就會按照模板中定義的規則進行轉換，否則繼續往下匹配。如此循環，直至整個 SVG 文件處理完畢。

Apache Batik 解析器要求處理對象是“well-form 格式良好”的 SVG 文件，有些還能根據 DTD 或 XML Schema 進行有效性驗証，Batik 的 DOM 解析器將 SVG 文件一次完整輸入在記憶體中解析，生成一個 DOM Tree 用以描述該文件。DOM 是一種與平台和語言無關的接口，可以進行動態地創建 SVG 文件的內容、結構和類型，瀏覽節點、添加、修改、刪除等操作。Batik 定義了一系列的對象和方法對 SVG DOM Tree 的節點進行各種隨機操作：

- Document 對象：作為樹的最高節點，Document 對象是對整個文件進行操作

的入口。

- Element 和 Attr 對象：這些節點對象都是文件某一部分的映射，節點的定級層次恰好反映了文件的結構。
- Text 對象：作為 Element 和 Attr 對象的子節點，Text 對象表達了元素或屬性的文本內容。Text 節點不再包含任何子節點。
- 集合索引：DOM 提供了幾種集合索引方式，可以對節點按指定方式進行遍歷。索引參數都是從 0 開始記數的。

DOM Tree 中的所有節點都是從 Node 對象繼承而來的。Node 對象定義了一些最基本的屬性和方法，利用這些方法可以從頭到尾遍歷 DOM Tree，同時，根據屬性還可以得知節點的名稱、取值並判斷其類型。

指定訪問 DOM Tree 某元素的流程

在文件載入記憶體完畢之就可以使用 getElementsByTagName 屬性訪問指定元素：NodeList EList = Doc.getElementsByTagName(TagName);

一旦建立了對 DOM Tree 中某個節點（例如根節點）的引用，就可以根據節點間的等級關係調用適當的方法進行遊走遍歷。下面以 circle.svg 為例說明各種方法的使用：

```
<?xml version="1.0" standalone="no"?>
    <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
    "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
    <svg>
    <circle cx="180" cy="140" r="3cm" style="stroke:rgb(128,0,128); fill:#C0C0C0;
    stroke-width:2 " />
    <ellipse cx="520" cy="140" rx="120" ry="60" style="stroke:#FF0000;
    fill:#00FFFF; stroke-width:3 " />
    </svg>
```

建立對< circle >元素的引用：

NodeList EList = Doc.getElementsByTagName(circle);

- 取得 circle 元素下的 attribute：

  Element TempElement = (Element) EList.item(i);

- 判斷指定的 Attribute 是否存在：

  TempElement.hasAttribute(AttName) == true;

  例如判斷 cx 是否存在：TempElement.hasAttribute(cx) == true;

- 取出指定的 Attribute：

  Attr TempAtt = TempElement.getAttributeNode(AttName);

  例如取出屬性 cx：Attr TempAtt =TempElement.hasAttribute(cx);

- 取出 Attribute 的值：

  String TempAttValue = TempAtt.getValue();

  TempAttValue 將會得到 cx 的值"180"

- 更換 Attribute 的值：

  NewAttValue = "250";

  TempElement.setAttribute(TempAtt, NewValueValue);

  此時元素"circle"的屬性"cx"將會得到新的值"250"

4.3 系統功能與實作

4.3.1 系統功能

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 4-3：

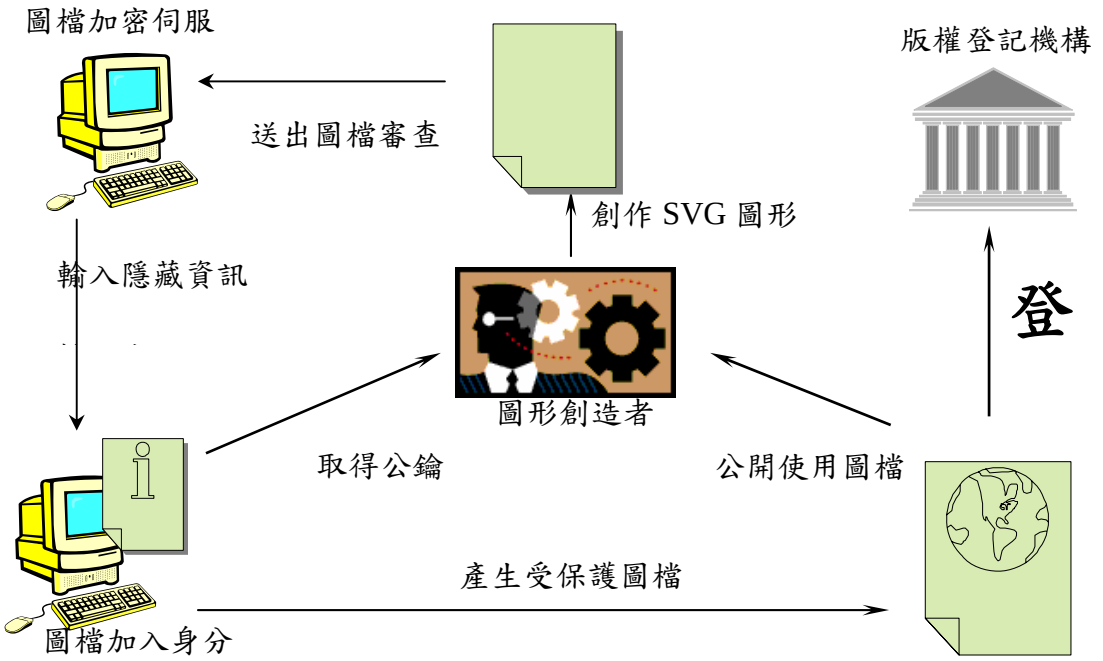

圖 4-3 SVG 隱藏資訊系統流程

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。

本系統實質操作過程如下：

首先使用者從 Client 端程式功能表上，標籤『Open File』進入，瀏覽目錄後選擇要上傳的 SVG 圖檔，接著開啟檔案，系統呈現確認對話框，如下圖 4-4 ：

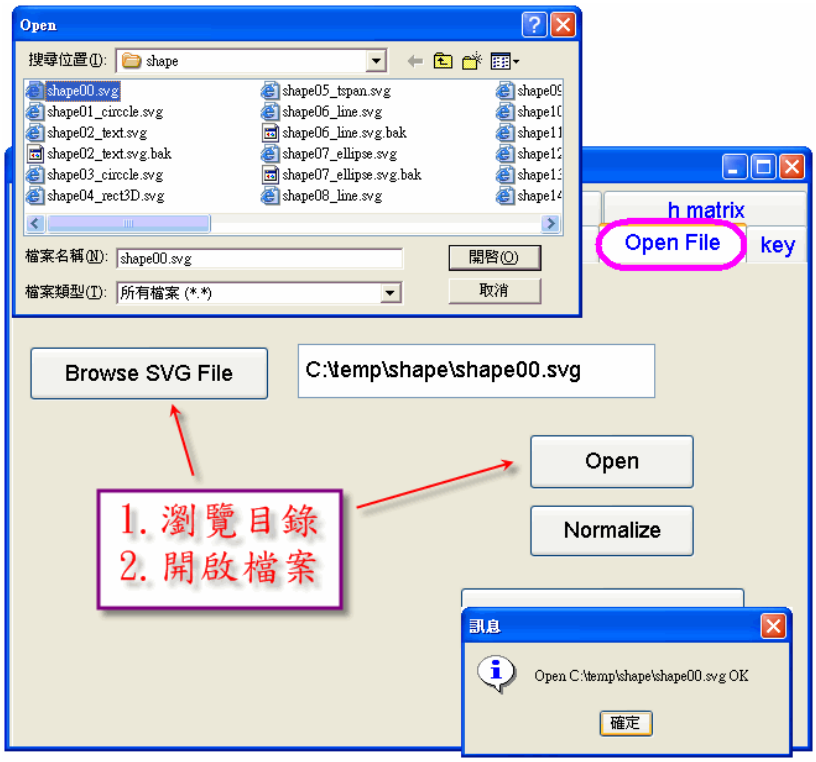

圖 4-4 系統開啟檔案過程

觀看開啟的 SVG 檔案內容可由標籤『SVF Text』點入，若點選『DOM Tree』則可看此 SVG 文件的 DOM Tree 結構，如下圖 4-5 和 4-6：

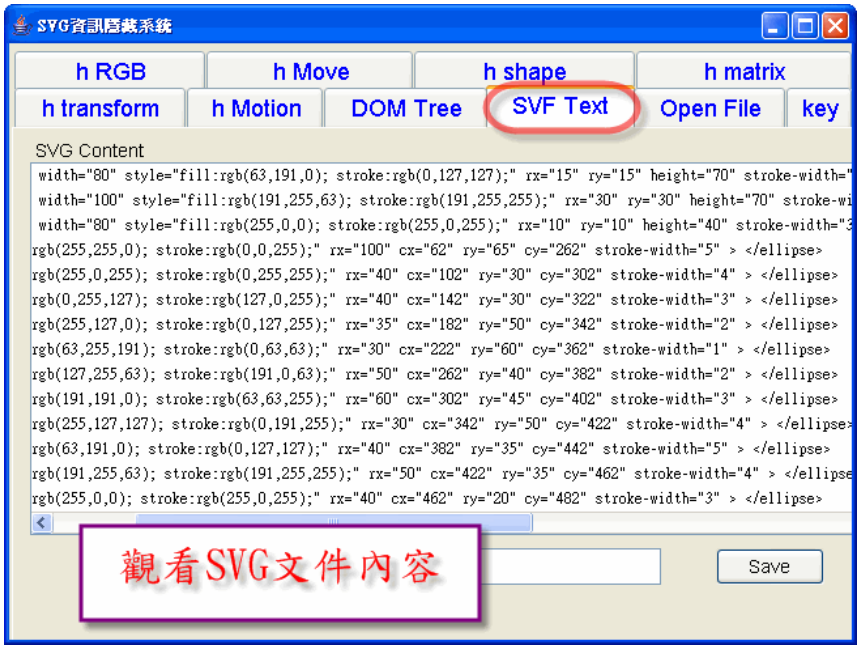

圖 4-5 系統顯示 SVG 文件內容

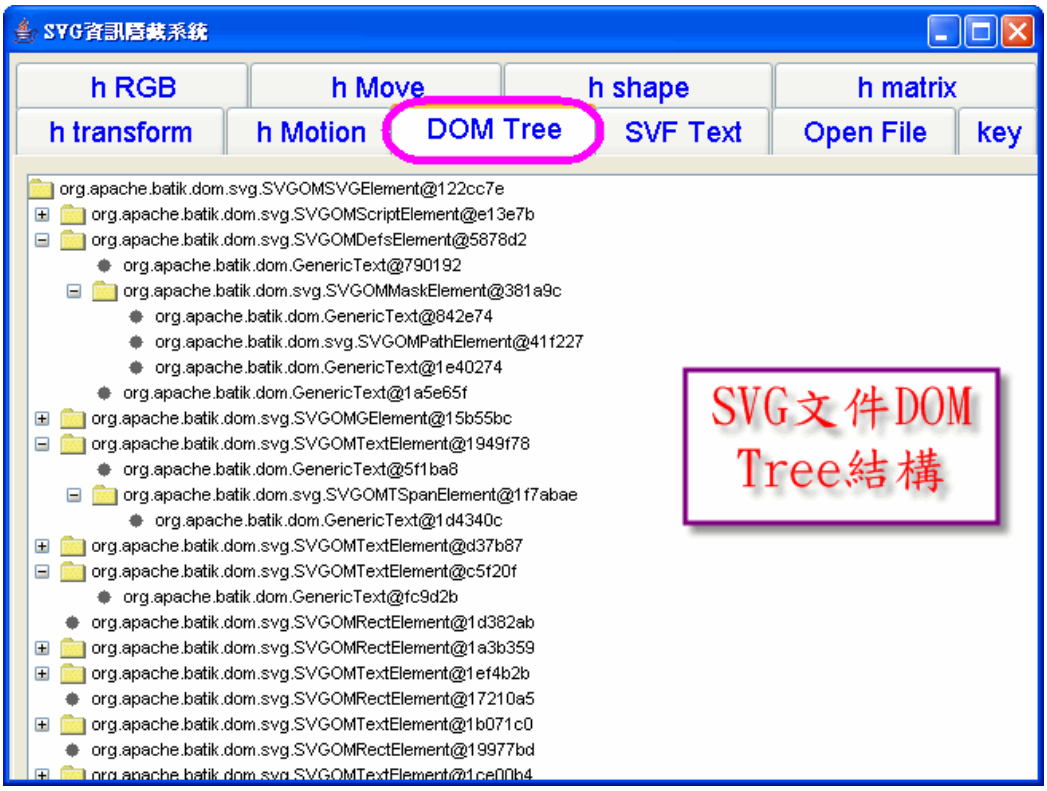

圖 4 - 6 系統顯示 SVG 文件 DOM Tree 結構

本系統使用 PKI 機制，倘若讀入已經受保護的 SVG 文件，會讀出文件裡面的公開金鑰，否則公開金鑰由系統產生，私鑰由使用者在 Client 端輸入，所有加密隱藏資訊的方法皆加入此兩 key 後，產生新的編碼後由 Server 端加密送回。

圖 4 - 7 公開金鑰與私鑰

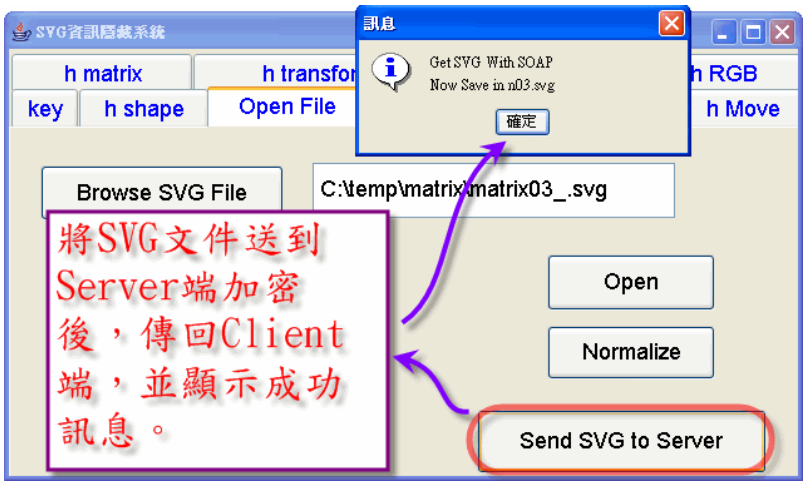

圖 4 - 8 Server 端加密送回 Client

4.3.2 實作一般字元隱藏

鍵盤上能夠直接打字在螢幕上的可見字元，位於 ASCII 碼（33～126），針對此可見字元，本系統實作出 6 種隱藏資訊的方法，實作過程敘述如下：

1. 轉置隱藏

假設想藏入字元為『ice』，先由 code f(x)按鍵產生新的編碼『55 45 63』，此新編碼已經參考公開金鑰與私鑰的 key 值，再經由 encrypt 按鍵寫入 SVG 程式碼中，解碼時需核對私鑰與公開金鑰，倘若任何一 key 錯誤則無法解讀出隱藏在 SVG 文件內的資訊。

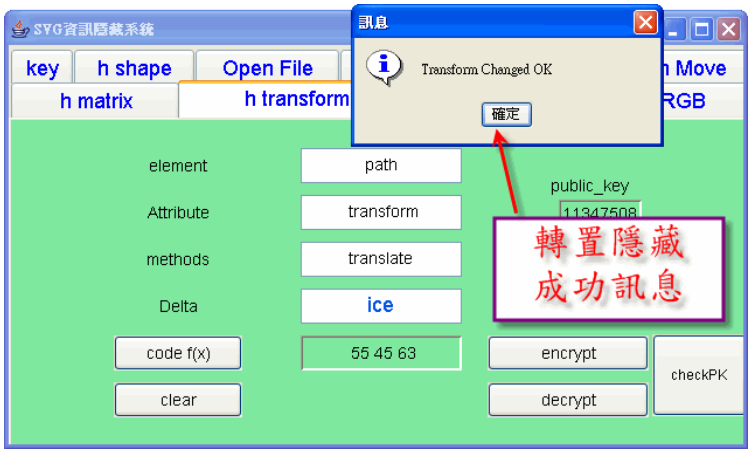

圖 4 - 9 轉置隱藏實作

2. 位移隱藏

假想藏入字元為『aaa*』，先由 code f(x)按鍵產生新的編碼『37 77 15 81』，雖然是三個相同的『a』但系統會產生不同編碼，Locate 按鍵以亂數找出隱藏的入口點，再經由 hide 按鍵寫入 SVG 程式碼中，解碼時按 recovery 鍵，系統將能正確將藏在 SVG 文件內的『37 77 15 81』解讀成『aaa*』。

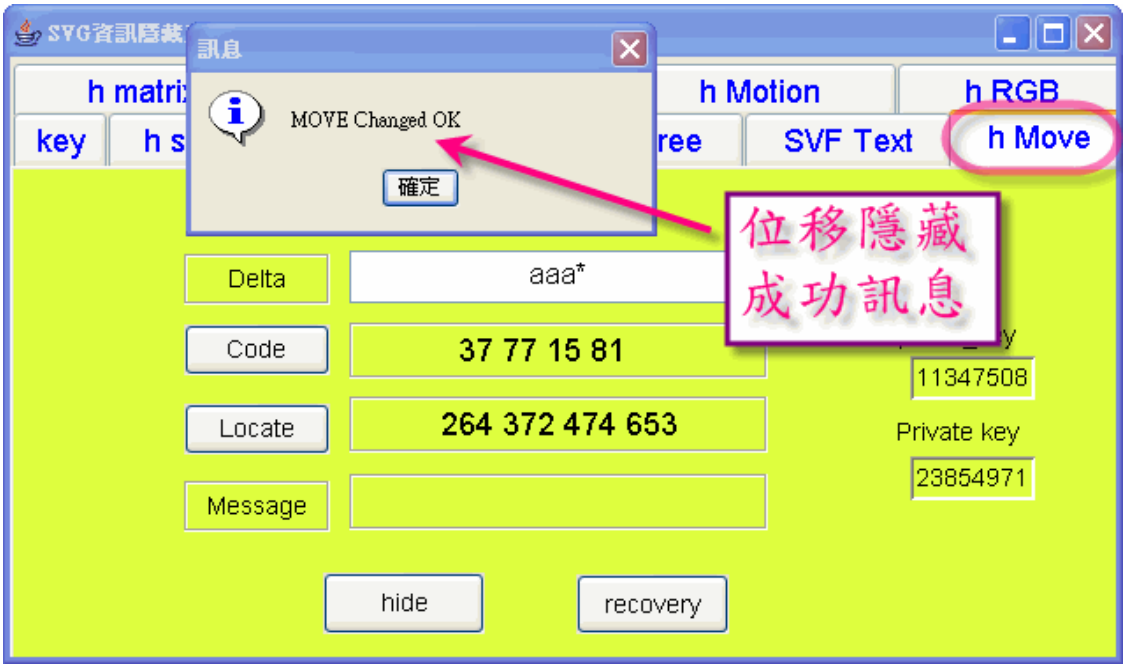

圖 4 - 10 位移隱藏實作

3. 動畫隱藏

動畫隱藏可針對常見 SVG 元素 animateMotion，先由 code 按鍵產生新的小數點編碼，Locate 按鍵以亂數找出隱藏的入口點，再經由 hide 按鍵寫入 SVG 程式碼中。若是針對元素 animate 時，設計另一個 animate 鍵，系統才能正確判斷不同動畫格式，還原時不論哪一種均可經由 Recover 按鍵解讀文件內的隱藏資訊。

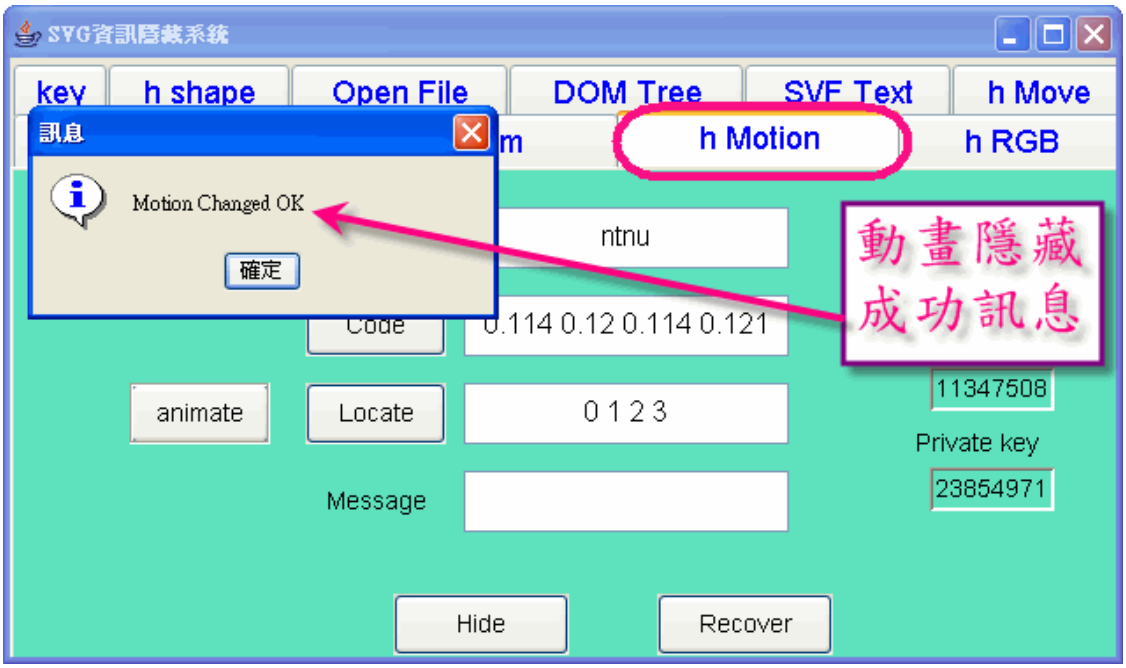

圖 4 - 11 動畫隱藏實作

4. 顏色隱藏

顏色隱藏操作模式與前面方法大致相同，下圖為系統正確還原隱藏的資料。

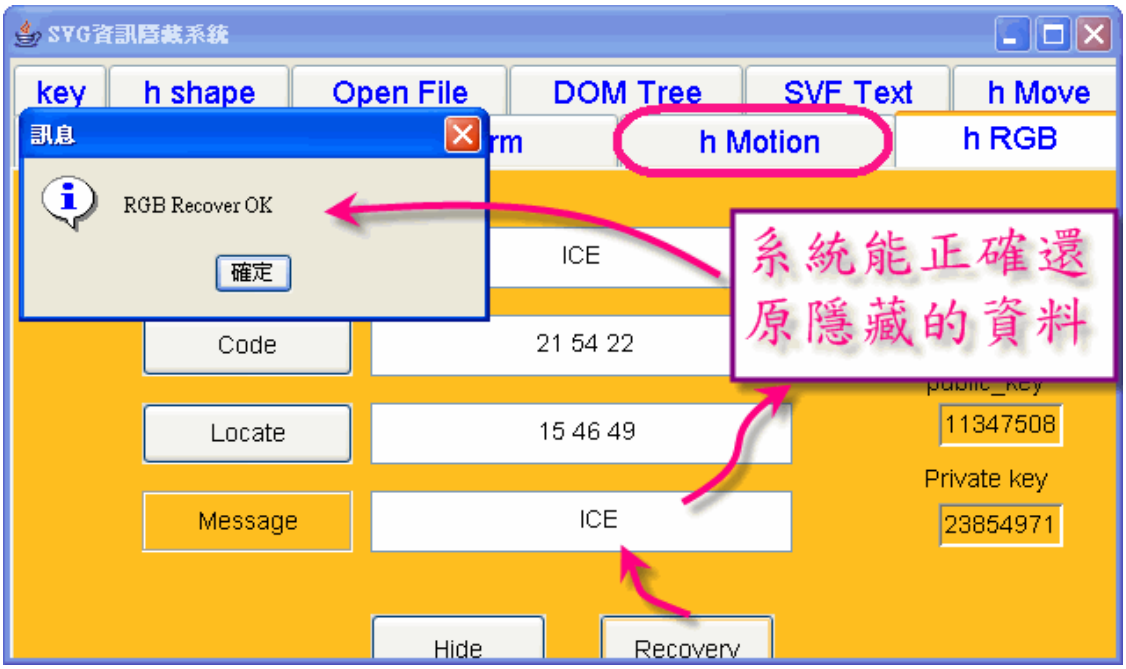

圖 4 - 12 顏色隱藏實作

5. 外形隱藏

針對常見 SVG 外形元素如 Eect、Circle、Ellipse、Line 或 Text，先由 code f(x) 按鍵產生新的編碼，再經由 encrypt 按鍵寫入 SVG 程式碼中。

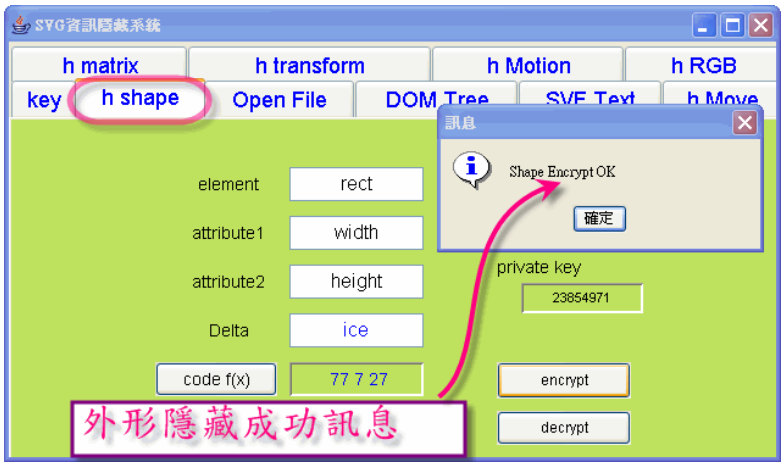

圖 4 - 13 外形隱藏實作

6. 矩陣隱藏

矩陣隱藏專門針對 SVG 的元素 matrix，Delta 中輸入 12 個字元『<%^&*(QwErTY』先由 code 按鍵產生新的編碼後，Locate 按鍵以亂數設定隱藏的入口點，再經由 hide 按鍵寫入 SVG 程式碼中。還原時 Recovery 按鍵解讀文件內的隱藏資訊，還原出『<%^&*(QwErTY』。

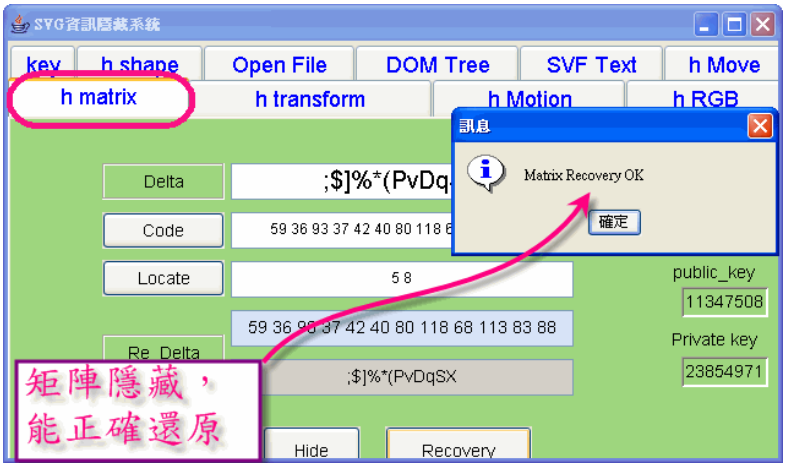

圖 4 - 14 矩陣隱藏實作

4.3.3 實作中文隱藏與偽裝

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。

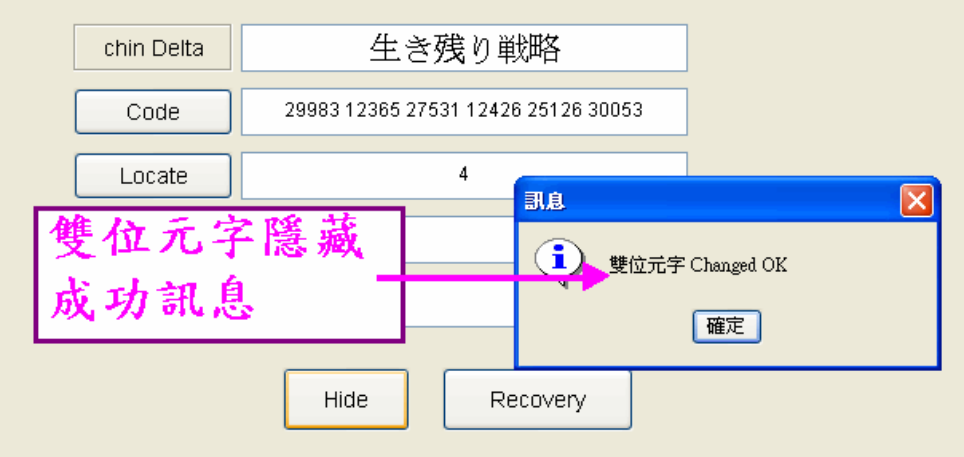

圖 4 - 15 中日文隱藏實作

自訂 Tag、偽裝 Animate、偽裝 Mask、隱藏 Curve 四種隱藏資訊的方法，實作上都以 Add Delta 按鍵提供使用者自行輸入要隱藏的資訊，再以 Add Tag 按鍵提供使用者自行輸入標籤，不經過編碼直接偽裝資訊，再經由 hide 按鍵寫入 SVG 程式碼中，可以隱藏 SVG 動畫、遮罩、曲線文字。

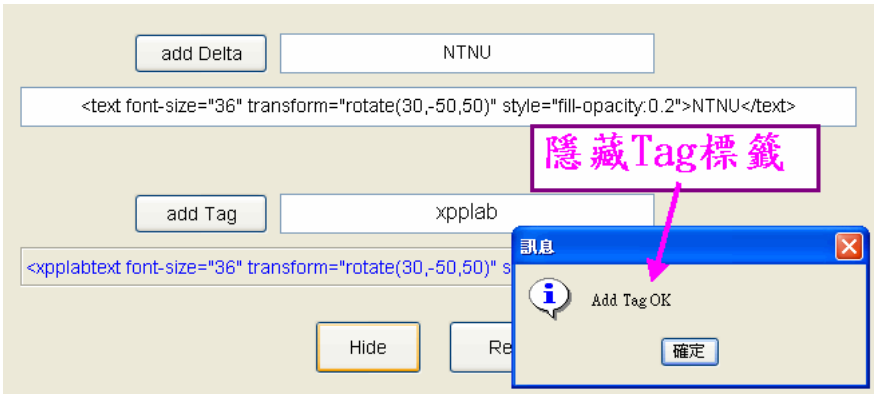

圖 4 - 16 附加 Tag 標籤實作

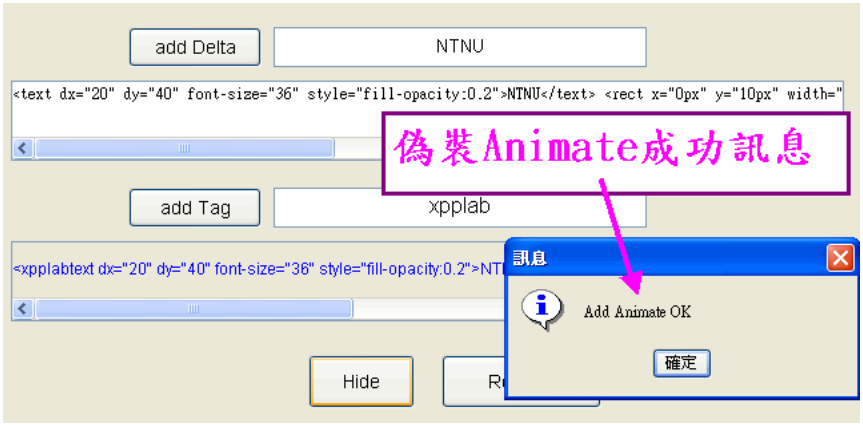

圖 4 - 17 附加 Animate 實作

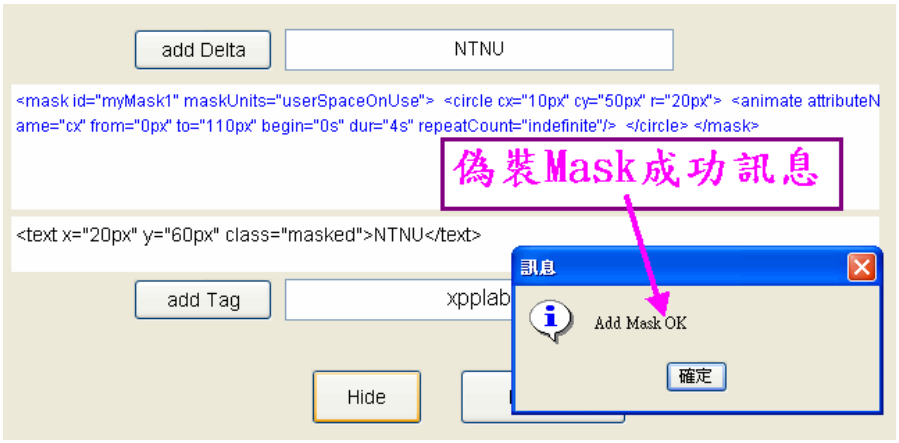

圖 4 - 18 附加 Curve 實作

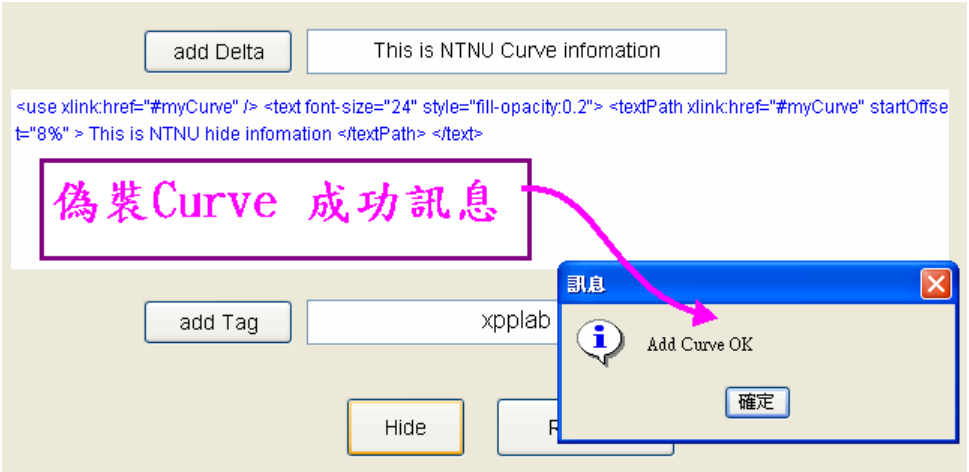

圖 4 - 19 附加 Curve 實作