國立台灣海洋大學資訊工程學系專題報告

二維條碼文件證書認證系統

09957007 簡偉智
09957036 周俊佑
09957049 謝宜辰
09957054 李威廣

指導教授：丁培毅 教授

中華民國 102 年 10 月 5 日

1.簡介

本專題運用 QR-code 儲存資料量多且讀取便利、不易失真的特性， 設計以 Android 手機讀取紙本文件上存於 QR-code 中的數位資料，使用者可以運用手機即時驗證文件中：時戳資料、簽章資料及文件內容的正確性，如此可以透過傳統紙本文件平行傳遞可驗證的數位資料，強化紙本文件的功能。

2.動機、大綱

一般的成績單、公告、能力測驗證書、或是保證書都是以印表機直接印出的紙本文件，這種方式成本較低且速度較快，可批次處理大量文件。但就資訊安全的角度來看仍有些疑慮，現今印刷技術的進步，使得普通的公告和紙本文件非常容易讓有心人士竄改。因此大部分重要的證書會加上一些防偽措施，如鋼印、浮水印、或用特殊材質的紙來印刷，但這些方法會使得證書的成本增加，一般人不易驗證，普通的證書使用這類的防偽措施就較浪費。

紙本文件有批次列印、生產速度快，和成本非常低的優點，但在資訊的分享傳遞速度較慢。公告的紙本文件僅能供位於文件前方的閱讀者接收，要再將資訊傳達較不方便，可能由閱讀者的手抄紀錄、電腦紀錄或將文件拍照分享。但過程可能有遺漏或被更改的不方便，因此在資訊傳遞分享的觀點上，紙本文件也是有不足的地方。

為了讓紙本文件可以以非常低的成本達到保護資料安全的效果，我們提出一個將紙本文件的內容簽章後放入文件中，讓任何有 Android 手機的使用者都可以驗證簽章的真偽，以此達到保護文件內容的安全。

其中，我們使用 QR-code。QR-code 以 資料類型(格式或字元組)、 版本(表示整個圖形的維度)和 錯誤修復等級三個項目作為區分。其最大的資料儲存格式為 40-L (version 40, error correction level L)，大約能儲存 23624 個 bits，即 2.8kb 的資料。將文件、文件簽章、時戳等資訊放入 QR-code，並嵌入到文件中生成 pdf 檔。文件將非常方便的產生、列印。

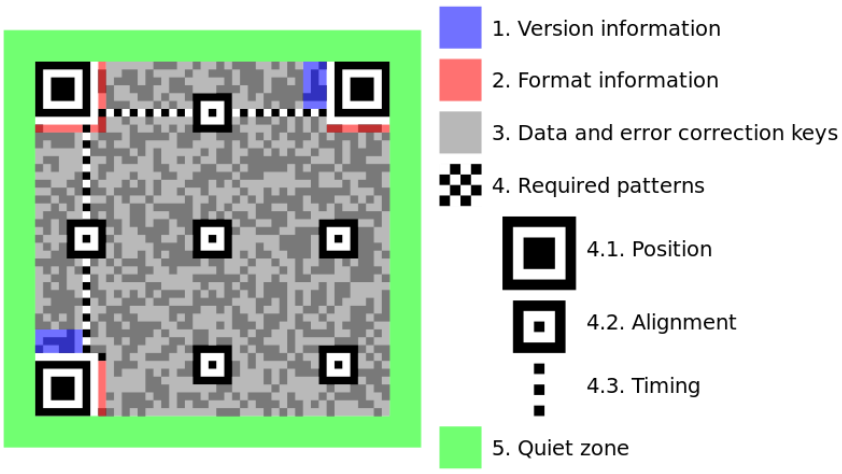
QR-Code 雖然方便，但實際操作時仍有一些問題：QR-code 最大的儲存容量(40-L)僅有 2.8kb，一份時戳資料便佔了 1.8kb，能使用的空間十分不足。且因為資訊太多、產生的 QR-code 若太複雜，使用解析度較低的手機相機掃描會非常的不方便。

因此，本專題以解決上列問題為目標，設計出一個能保護資料完整性(data integrity)、節省大量生產的成本、減低驗證方法的難度、跨平台的程式設計，使不同使用者都能方便的使用此系統、並且是讓資訊便於轉傳分享的系統，以達到真正紙本文件實作數位簽章的功能。

3.使用相關技術

QR-code 相關資訊

格式：



資料類型：

格式→	數字	字母數字	Binary/byte	漢字/假名
最大字元數	7089	4296	2953	1817
每個字元容量 (bits)	3 1/3	5 1/2	8	13
字元集合	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0–9, A–Z (只有大寫), 空格, $, %, *, +, -, ., /, :	ISO 8859-1	Shift JIS X 0208

(即一 QR-code 最大能儲存 23624 個 bits、2.8Kb 的資料)

錯誤修復等級：

Level L (Low)：可恢復 7%的資料
Level M (Medium) ：可恢復 15%的資料
Level Q (Quartile)：可恢復 25%的資料
Level H (High)： 可恢復 30%的資料

本專題在處理 QR-code 的方面使用 ZXING 這個 Open Source Project，此 Project 提供 Java SE、Android、iPhone、Actionscript、CPP 以及 JRuby 等等平台產生和讀取 QR-code 的模組。本系統將此工具應用在 Android 以及 PC 端的 Java 應用程式。

電腦端方面，我們使用 QR-code Encode API 做資料的編碼動作，將文件生產者的文件資料、驗證資訊及時戳的資料以 Java 的 String 格式讀入，經過 QR-code Encode API 產生 BitMatrix 形式，最後將 QR-code 資料輸出成 GIF 格式的圖檔。在手機端方面，我們使用 ZXING 的 Decode API，將由手機的相機讀入 QR-code，再取出 QR-code 中所儲存的字串。將字串中的簽章資料、時戳資料加以驗證。

時戳相關資訊

本專題使用 RFC3161 標準規範的電子時戳。在專題的一開始時，閱讀 楊中皇教授的網路安全的理論與實務教學，瞭解且實作架設 OpenTSA Server，後來因為時間和機器的一些問題，我們的專題暫時改用 http://time.certum.pl/ 這個 Time Stamp Service 測試。使用 Tomasz Widomski 先生在網路上的 Timestamping client RFC3161 Project，將我們想要的時戳程式實作。

在本專題中，簽章的部分使用 Timestamping client RFC3161 Project 中的 SenRequest，用 SendingRequest() 方法對資料加上時戳簽章。驗證的部分使用 Timestamping client RFC3161 Project 中的 VerTimeStamp ，以 VerifyTimeStamp() 方法驗證由 QR-code 中讀取到的時戳資料。

ZIP 壓縮相關資訊

本專題使用 java.util.zip 的工具實作 ZIP 壓縮的功能。主要原因是因為時戳資料加上簽章資料加上整份文件內容太過龐大，因此在將資料放入 QR-code 前，我們先將所有資料以 ZIP 壓縮的方式壓縮。雖有成效但時戳資料仍是太大，因此我們分析取得的時戳格式(PKCS7)，將較龐大且重複的簽章資料取出，預先置入程式中，便大大的節省了資料的使用空間。

我們使用 java.util.zip.GZIPInputStream 及 java.util.zip.GZIPOutputStream 這兩個 class 做 zip 壓縮的實作。

OPEN OFFICE 相關資訊

我們使用 Open Office 的 API 來實作將.doc、.docx、.odt 自動化生成 pdf 檔的功能。並在生成 pdf 時嵌入相對應的 QR-code，方便批次處理大量文件。

本專題建立了一個 PDFMaker class ，實作流程為：

1. 先呼叫 init() 方法將 PDFMaker 初始
2. 使用 getDoc() 方法讀取輸入的文件
3. 將取得的文件字串，透過 SenRequest 的 SendingRequest() 方法算出時戳
4. 將時戳資料透過 zip 壓縮，再使用 makeQRCode() 方法將時戳資料製成 QR-code 圖檔
5. 最後，使用 editDoc() 方法輸出 pdf 檔。

本專題使用的 OpenOffice pdf 輸出是參考以下網址，
https://wiki.openoffice.org/wiki/API/Tutorials/PDF_export，有較詳細的說明。

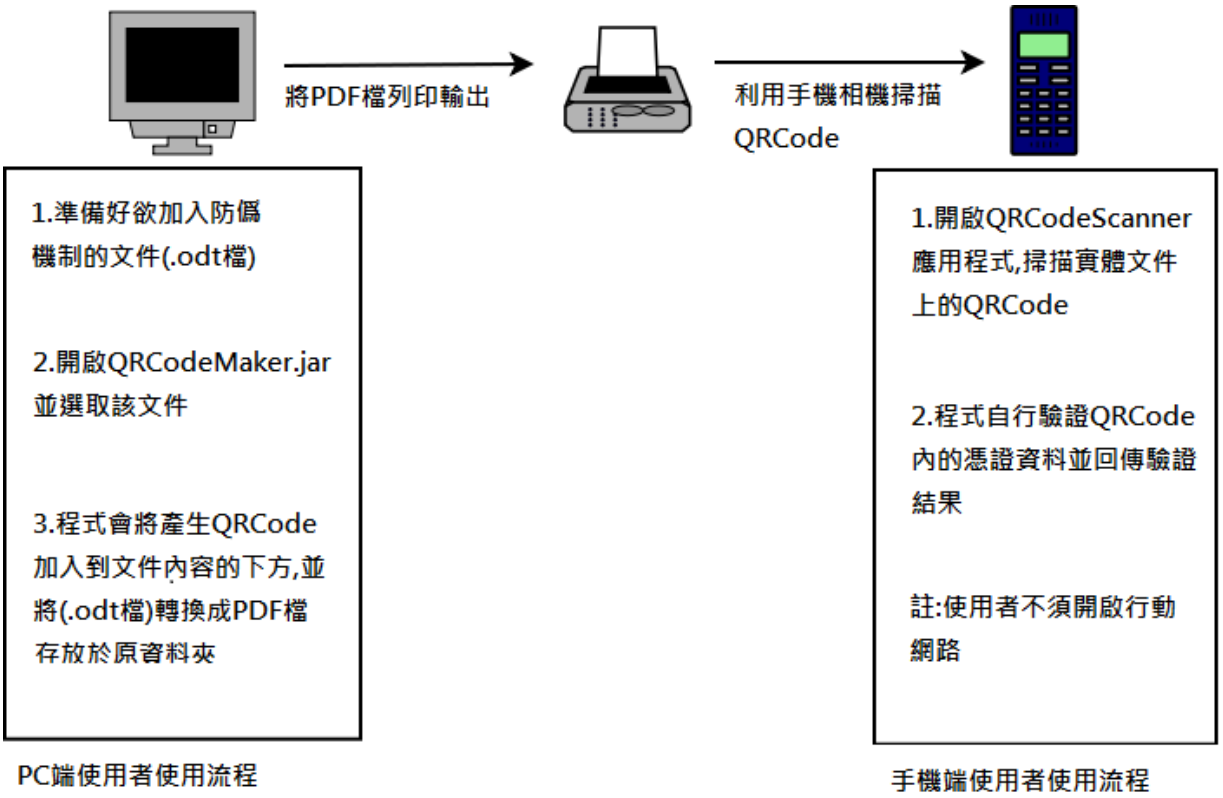
簽章相關資訊

本專題使用 Java 提供的 Java.security 來實作數位簽章的功能，先由 SHA1 的雜湊演算法計算出 hash 值，再使用 RSA 的數位簽章演算法簽章。

Android 部分

使用 Zxing project 的 IntentResult 類別操作 QR-code 讀取的工作，並利用 IntentResult 的 getContents() 方法取得字串中的內容，加以處理。

4.使用流程:



PC端使用者使用流程　　手機端使用者使用流程

(I) 文件發佈者程式操作流程

文件發佈者，即產品廠商、成績單發佈單位等等，使用本系統將依照下列流程操作：

1. 安裝

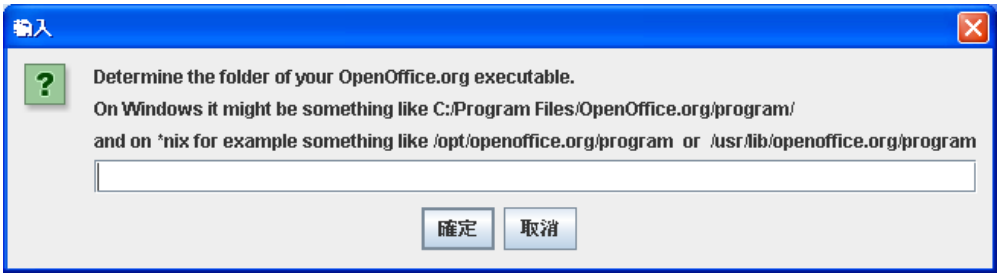
Apache OpenOffice 4.0.1 以上版本
Java Runtime Environment 7 以上版本

2. 輸入

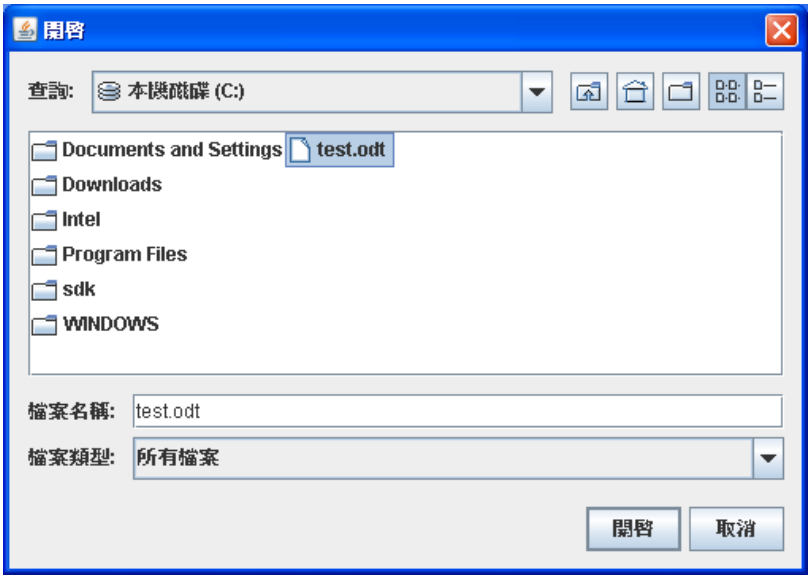
本系統支援以下檔案格式： .doc / .docx / .odt

3. 執行

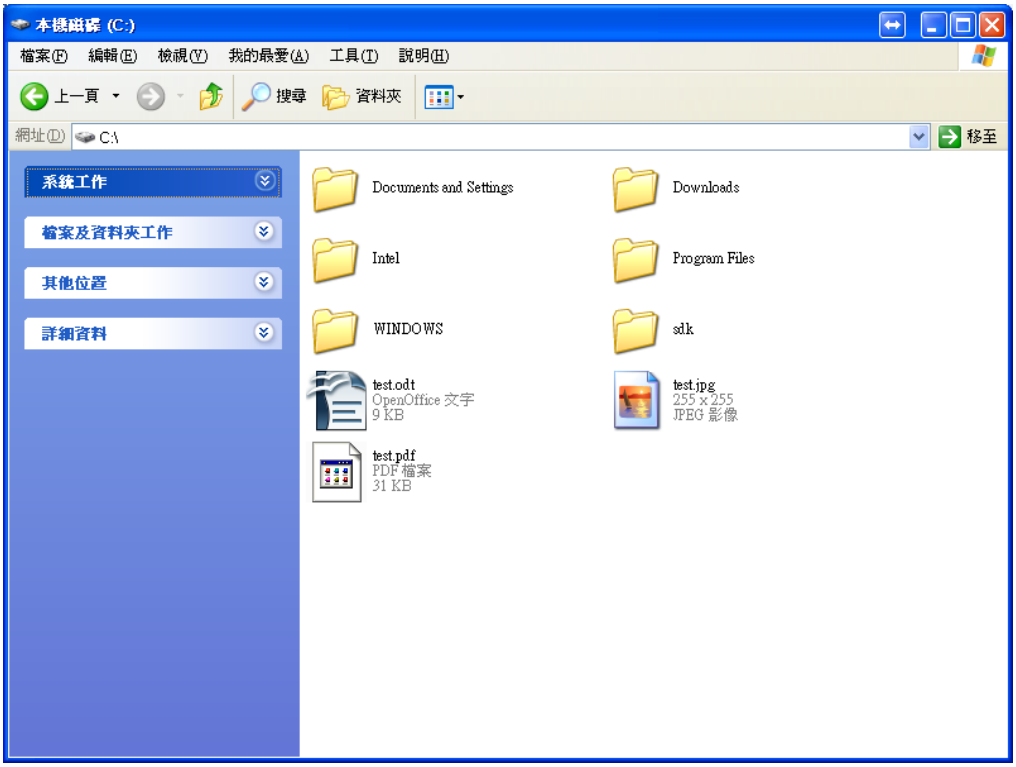
開啟 “ QRPDFMaker.jar ” 程式，接下來照著以下畫面操作：



1. 程式執行的開始，會要求使用者輸入 OpenOffice 執行檔的資料夾路徑
(Windows 系統可能於 C:/Program Files/OpenOffice.org(版本編號)/program/
Linux 系統可能於 /opt/openoffice.org(版本編號)/program)。
輸入後點選確定。



2. 選擇想要加上簽章的檔案後點選開啟。
   本系統使用圖形化的選單，讓使用者方便選取欲加上簽章的原始文件。



3. 系統將自動執行，若過程中沒有錯誤，最後程式會輸出一個內容為 QR-code 的 GIF 圖檔以及一份嵌入 QR-code 的 PDF 檔，文件發佈者便能用此 PDF 檔列印、公告。

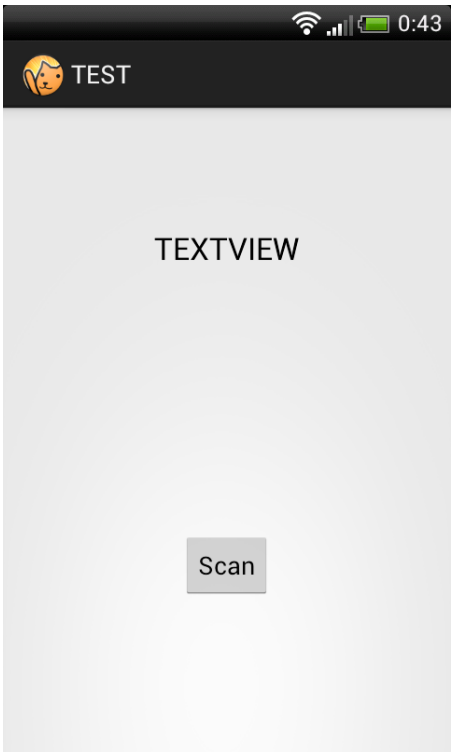
(II)使用者操作流程

欲驗證文件、公告的使用者，使用本系統將依照下列流程操作：

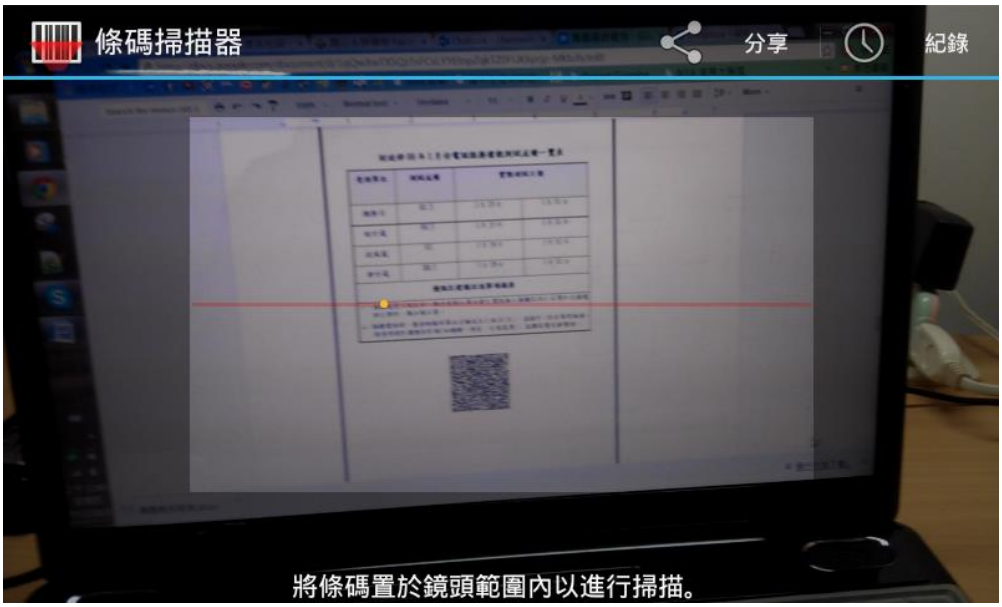
1) 使用具有照相功能的 Android 系統智慧型手機。

2) 安裝由 ZXing Team 開發的條碼掃描器 APP（程式下載連結：https://play.google.com/store/apps/details?id=com.google.zxing.client.android&hl=zh_TW）

3) 安裝本系統開發的 APP，安裝完成後開啟，接下來照著以下畫面操作。



1.程式初始畫面，點擊 Scan 鍵開始掃描。

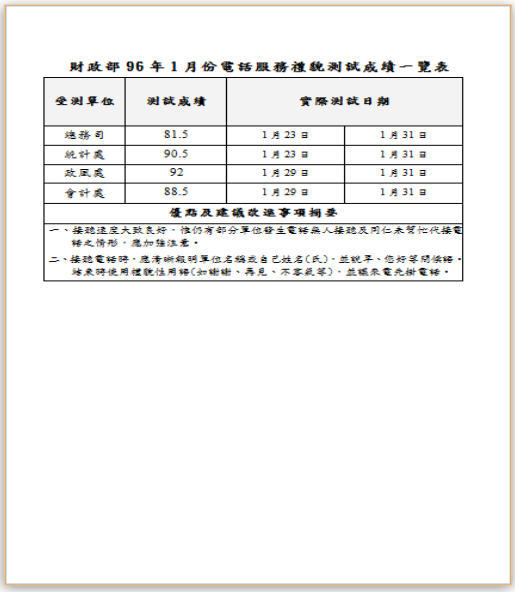


2.以手機的相機對準文件上的 QR-code 掃描，截取到 QR-code 後程式將自動返回初始畫面。

3.螢幕上顯示文件的時戳資料、簽章資料等驗證結果，並將文件中的資料以字串顯示在螢幕上，方便供使用者編輯或自己驗證資料的完整。

5.實際例子

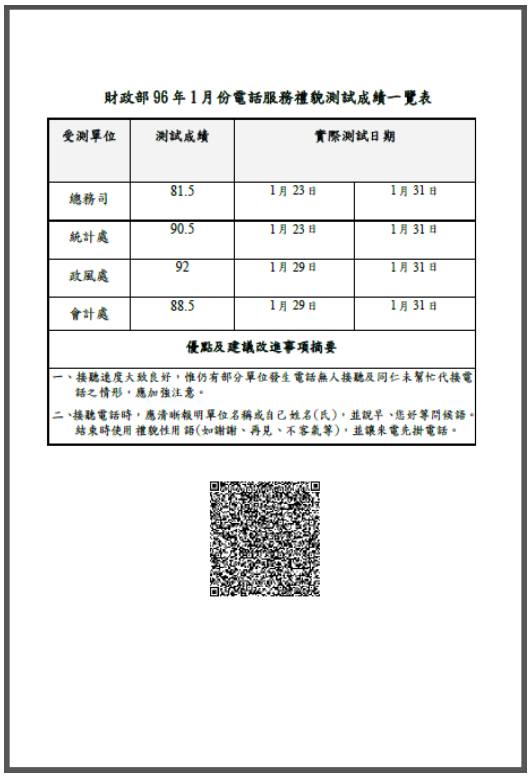
我們以這份網路上搜尋到財政部的某份報表作為例子，原始檔案 (https://www.dropbox.com/s/zsteg04n0tzmqu5/TEST.doc) 最後是沒有嵌上 QR-code 的，如下圖。



財政部 96 年 1 月份電話服務禮貌測試成績一覽表

受測單位	測試成績	實際測試日期	
總務司	81.5	1 月 23 日	1 月 31 日
統計處	90.5	1 月 23 日	1 月 31 日
政風處	92	1 月 29 日	1 月 31 日
會計處	88.5	1 月 29 日	1 月 31 日
優點及建議改進事項摘要			
一、接聽速度大致良好，惟仍有部分單位發生電鈴無人接聽及同仁未幫忙代接電話之情形，應加強注意。 二、接聽電話時，應清晰報明單位名稱或自己姓名(氏)，並親切、值好等問候語，結束時使用禮貌性用語(如謝謝、再見、不客氣等)，並禮來電先掛電話。			

接下來，執行本報告實作的程式後，可以自動產生一個下方嵌入 QR-code 的 PDF 文件，如下圖。



財政部96年1月份電話服務禮貌測試成績一覽表

受測單位	測試成績	實際測試日期	
總務司	81.5	1月23日	1月31日
統計處	90.5	1月23日	1月31日
政風處	92	1月29日	1月31日
會計處	88.5	1月29日	1月31日
優點及建議改進事項摘要			
一、接聽速度大致良好，惟仍有部分單位發生電話無人接聽及同仁未幫忙代接電話之情形，應加強注意。 二、接聽電話時，應清晰報明單位名稱或自己姓名(氏)，並說早、您好等問候語，結束時使用禮貌性用語(如謝謝、再見、不客氣等)，並讓來電先掛電話。			

文件下方的 QR-code，裡面記錄著所有文件內的資料。

6.碰到的問題、解決方法

QR-code 的能儲存的空間可能放不下一份完整的文件，因此我們想的解決方法是：將文件中的較重要的資料取出來，如證書中的時間，姓名，數量等資料製成表格，以便供客戶端檢查重要資訊。一方面節省資料的儲存空間，另一方面也幫使用者整理重點，方便使用者查詢。

7.結論

本專題為紙本文件設計了方便驗證的系統，但仍存在一些值得研究的問題。就整體而言，本專題已大致完成製作的目標，但我們還希望能再將很多問題補足，因此此專題還在繼續擴充當中。希望在未來我們或相關領域的研究者，能繼續的將這個系統製作的更加便利。

8.參考資料

[1] QR-code from Wikipedia
http://en.wikipedia.org/wiki/QR_code

[2] ZXING Project
https://code.google.com/p/zxing/

[3] Timestamping client RFC3161
https://sites.google.com/a/widomski.net/ntporgpl2/client-tsc-rfc3161

[4] Java's Generating and Verifying Signatures
http://docs.oracle.com/javase/tutorial/security/apisign/index.html

[5] Java security
https://www.ibm.com/developerworks/java/tutorials/j-sec1/

[6] 楊中皇 網路安全的理論與實務 第十六章 OpenTSA
http://security.nknu.edu.tw/textbook1ed/CHAP16-OpenTSA.pdf

[7] Open Office API for PDF export
http://wiki.openoffice.org/wiki/API/Tutorials/PDF_export