防止智慧型行動裝置之 APP 應用程式遭盜版之研究與實作

孫秉瑄[1]、朱麗芬[1]、陳冠宇[2]
[1]新竹市立建功高中
[2]清華大學資訊工程研究所

摘要

隨著資訊科技的日新月異，智慧型行動裝置，例如智慧型手機以及平板電腦，在過去數年間很快地崛起。使用者可以很便利的帶著這些智慧型行動裝置，無所不在地使用網路或是下載及執行 APP 應用程式。APP 應用程式可分為免費與付費兩種類型，免費版本沒有盜版問題，通常是付費版本的初階版或試用版，用以吸引使用者來使用，或者是程式開發者藉由 APP 使用量來獲取廣告商的廣告費。付費版本是程式開發者的智慧結晶，有好的想法就可以藉此賺錢，當然如何吸引使用者來使用就各憑本事了，例如知名的憤怒鳥遊戲就是一個很成功的典範。然而在盜版猖獗的社會中，APP 應用程式也成為了盜版者新的目標，所以在本研究中，我們將探討 APP 應用程式如何地被駭客破解而成為盜版 APP，我們分別就 Apple iOS 作業系統為核心的 iPhone、iPad 以及 Android 作業系統為核心的智慧型行動裝置來加以分析。我們接著提出一個嶄新的防止 APP 應用程式遭受駭客盜版的方法。我們的方法主要是針對 APP 應用程式執行時所需要的相關資源檔，例如圖片與音效等檔案進行存取管控。使用者必須在每次執行 APP 應用程式時先進行網路連線及驗證程序，通過驗證程序之後，APP 應用程式才能取得資源檔，這是一個在目前的時代中可以比較安全的防止 APP 應用程式被盜版的方法，可以帶給現在的正版業者一個比較好的智慧財產權保護機制。

關鍵詞：智慧型行動裝置、APP、反盜版

壹、前言

一、研究動機

在現今的社會中，科技的進步日新月異，智慧型行動裝置，包括手機與平板電腦，在最近幾年內發展迅速且廣泛的被使用，也因此 APP 應用程式的開發與使用也越來越普及。目前的兩大陣營包括 Apple 以 iOS 作業系統為核心的 iPhone、iPad 以及以 Android 作業系統為核心的智慧型行動裝置，例如: HTC、Samaung 的手機以及 Samaung、Amazon、Asus 與 Acer 的平板電腦。iOS 的 APP 應用程式必須由 Apple 的官方下載平台 Apple Store 做為下載安裝的唯一管道，這是一個較為封閉的系統，因為 Apple 公司並未公開其 iOS 作業系統的原始碼。Android 是走開放式系統架構，其作業系統是為開

放原始碼(Open Source)，有許多管道(官方及非官方下載平台)可以下載安裝 Android 的 APP 應用程式，其中目前最有名的是官方下載平台 Google Play(之前名叫 Android Market)，另有許多其他營運網站(非官方下載平台)也都可以下載安裝 APP 應用程式，例如中華電信的 Hami APPS 軟體商店。不論是 Apple 的 APP 應用程式或者是 Android 的 APP 應用程式，只要是付費的程式都有可能被盜版，例如: iPhone 的 iOS 裝置如果被越獄(Jailbreaking)成功之後，使用者可以將盜版軟體安裝在 iPhone 或 iPad 內而不透過 Apple Store 的管制，如此盜版軟體即可不付費而能使用。Android 由於是開放式系統，所以使用者只需要將 APP 應用程式複製到自己的裝置內即可安裝使用。儘管 Apple Store 與 Android Market 都有發展屬於自己的 APP 應用程式防盜措施，很不幸的，這些防盜措施無法有效的保護合法 APP 應用程式。

二、研究目的

本研究的主要目的是探討以兩大作業系統(Apple 的 iOS 與 Android 作業系統)為核心的智慧型行動裝置下，APP 應用程式是如何被駭客盜版使用，我們藉由觀察及了解駭客的破解方式，進而提出一個嶄新的方法來防止智慧型行動裝置上 APP 應用程式被盜版使用的問題。

三、研究架構

本研究之進行方式為首先介紹智慧型行動裝置的發展及現況，接著介紹現行 APP 應用程式付費驗證流程。之後介紹以 Apple 的 iOS 為核心的行動裝置以及以 Android 作業系統為核心的行動裝置對 APP 應用程式的防盜版保護機制。這些防盜版保護機制主要由行動裝置業者所提供。然後我們將探討這些機制如何被盜版者所破解。另一方面，我們也由 APP 應用程式開發員的觀點，探討程式開發員所設計的防盜版保護機制，以及如何被破解的方法，並由其中觀察出其關鍵點為何? 根據這些關鍵點，我們將設計新的防盜版保護機制，使得盜版的行為能夠得以控制。

貳、研究方法

一、智慧型行動裝置的發展

(一) 智慧型手機

智慧型手機有別於傳統的功能手機，除了打電話的功能外，它還可以像一般電腦一樣，具有連結網路、收發電子郵件、瀏覽網頁、播放影音等功能。更多的功能包括照相以及下載及安裝 APP 應用程式等。智慧型手機的發展開始於 2007 年，由 Apple 公司當

時的總裁賈伯斯發表了第一代的 iPhone 產品， 2012 年推出到 iPhone 5 系列。由於賈伯斯對 iPhone 設計理念的堅持，使得 iPhone 具有亮麗的外表與良好的觸控螢幕操作，讓使用者耳目一新，至今談起 iPhone 總是讓人們想起已過逝的賈伯斯 [1]，而 iPhone 至 2014 年下半年已推出到第六代。iPhone 使用的作業系統為 iOS，這是一個封閉式的作業系統，由於作業系統程式碼未公開，APP 應用程式也只能由官方伺服器所提供的 Apple Store 下載，所以其安全性較佳，但也因為其為封閉式系統，一些專業人員並不喜歡此模式。不讓 iPhone 專美於前，Google 與 34 家手機製造商、晶片製造商、軟體開發商和電信運營商共同建立了以Android作業系統為核心的開放手機聯盟 [15]。Android系統是一個開放式系統，其作業系統原始碼是公開的，因此也廣為人們所接受。目前有許多智慧型手機廠商採用Android系統做為其核心，例如台灣宏達電的 HTC 手機與韓國 Samaung 的手機等，從此智慧型手機便進入戰國時代。研究機構 Strategy Analytics 資料顯示，2014 年第 3 季 Android 手機全球市占率 84%，略低於第 2 季的 85%。第 3 季 iPhone 市占率 12%，Windows 手機市占率 3%，黑莓機市占率 1%[12]。就全球市占率而言，Android 手機聯盟領先蘋果 iPhone 手機有越來越擴大的趨勢，目前智慧型手機市場仍以 Android 手機及 iPhone 手機為主流，兩者全球市占率合計快接近九成，其中 Android 手機全球市占率遠大於 iPhone 手機。

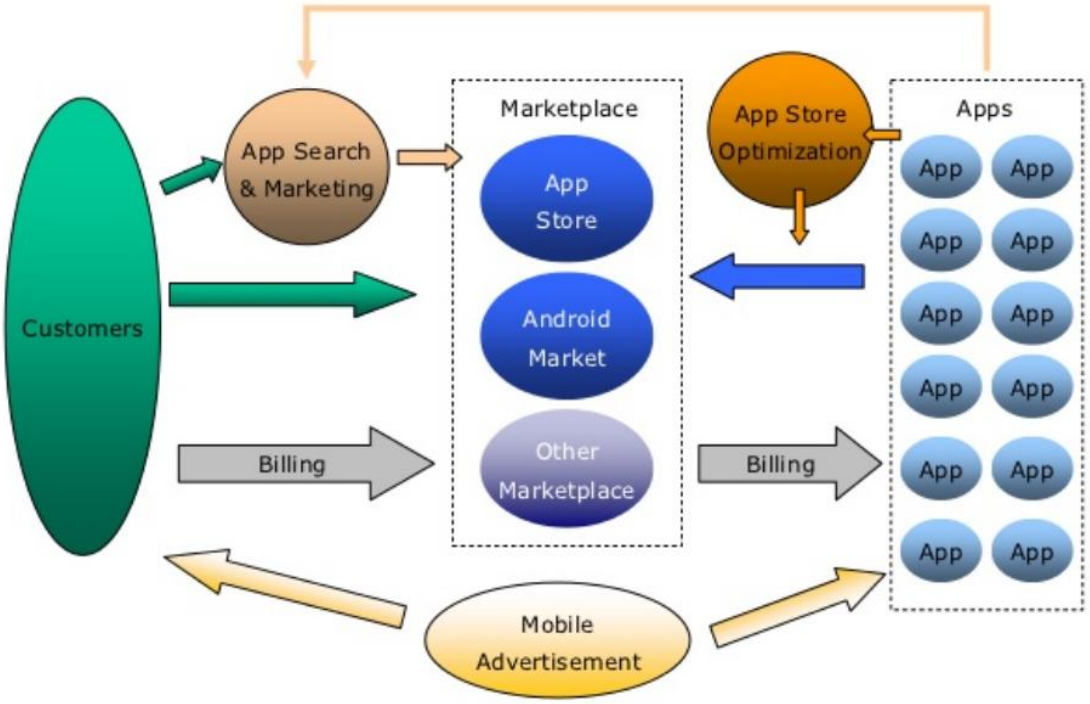
(二) 智慧型平板電腦

智慧型平板電腦有別於傳統的筆記型電腦與早期的平板電腦，除了體積變小，重量變輕，厚度變薄之外，它也採用觸控螢幕技術，使用手寫輸入或虛擬鍵盤取代實體鍵盤，使用手指觸控取代滑鼠，並可觸控縮放畫面與圖案。智慧型平板電腦的發展開始於 2010 年，由 Apple 公司發表了第一代 iPad 產品，到 2012 年 10 月 23 日已經推出到第四代 iPad。iPad 一推出時，全球市占率即高達五成以上。2012 年第二季 iPad 全球市占率已達 68%[6]，2012 年第三季 iPad 全球市占率則為 50.4% [7]。相較於 iOS，以 Android 作業系統為核心的平板電腦則出現百家爭鳴的狀況，市場研究機構 Strategy Analytics 發表 2014 年第一季全球平板市場報告，出貨量達到 5,760 萬台，與去年同期的 4,830 萬台相比，成長 19%。分析各作業系統所佔的比例：Android 平板佔 65.8%，大幅領先第二名 iPad 的 28.4%，Windows 平板則有 5.8% 的市佔率。然相較於 Apple 的 iPad 而言，仍然有很大的差距[4]。

二、現行 APP 應用程式付費驗證流程

伴隨著智慧型行動裝置普及率的日益增加，牽引著廣大的 APP 應用程式商業市場與利基，吸引 APP 應用程式開發者、廣告商及建基於 APP 應用程式形成多樣化消費形態。其中，APP 應用程式產品的商業模式特色在於多樣化產品交易，包括單純出售、廣告、附屬元件更新等，可依功能性分為遊戲型、應用程式型工具等，並兼具免費與付費兩類型 APP，特別是付費 APP 可能因完整的功能性與設計吸引大量點擊下載率，促成億萬收入。

在現行 APP 應用程式付費驗證流程中，使用者需要預先創建一組登錄帳號與信用卡支付資訊，當使用者成功進行交易手續後，使用者即可下載並安裝付費 APP。其中，開發商有兩種方式確認使用者付費，一種是透過官方伺服器(iOS 或 Android)，另一種則透過開發商自行架網站伺服器(僅 Android 作業系統有此方式)。我們現在將 APP 應用程式付費驗證流程由圖一表示。



圖一、行動 APP 付費驗證流程 (圖片來源: [13])

三、APP 應用程式防盜與盜版之方法

(一) 以 iOS 為核心的 APP 應用程式

iPhone 與 iPad 的 APP 應用程式下載方式有兩種，一種是透過個人電腦為中介者，使用者在他的個人電腦上安裝有 Apple 的 iTunes 軟體，當 iPhone 與 iPad 連接上電腦後，iTunes 將會將電腦與 iOS 的行動裝置同步處理，兩者的資料將同步備份。iTunes 軟體原是 Apple 公司所發展出來的一套數位版權管理機制[9]，原本是用來防止非法音樂與多媒體的下載，並且也是合法版權的音樂檔與多媒體檔的播放軟體。後來 iPhone 與 iPad 發展起來，APP 應用程式也用這套機制來進行數位版權保護，它記錄著使用者購買及下載的 APP 應用程式。iPhone 與 iPad 的 APP 應用程式是透過 iTunes 發佈，使用者下載後，後端伺服器會對 APP 應用程式產生一個對應的 iTunes 帳號的數位簽章[5]，數位簽章為密碼學原件之一，效力相當於手寫簽名一般，主要目的為達到數位資訊上的不可否認

性，表示此 APP 應用程式是相對於此 iTunes 帳號的擁有者所下載，沒有任何人可以偽造出如此一個簽章，因此可以做為認證的目的，檢查數位簽章是否真偽可以做為判斷 APP 是否合法的依據。另一種方式為由 iPhone 與 iPad 直接連接上 Apple Store，首次使用要先建立帳號及輸入信用卡號碼，管控方式與 iTunes 模式類似，使用相似的數位版權管理機制。雖然 Apple 公司創造了他們的數位版權管理機制，可是還是有許多方法可以破解或繞過此機制。

1、方法一

首先，對智慧型裝置進行 iOS 越獄(Jailbreaking)步驟，越獄完畢之後，安裝 Cydia，Cydia 是一個可以讓使用者在越獄後的 iPhone 或 iPad 上尋找及安裝各類軟體的軟體管理器[16]。然後在 Cydia 中新增 Hackulous 網站，Hackulous 為專門破解蘋果作業系統機制的駭客團體所架設的網站，做為軟體下載的來源，接著透過 Cydia 開始搜尋並載入適合 iOS 版本的“AppSync”，例如原執行 iOS 5.0.1，即可下載“Appsync for iOS 5.0+”，如此透過破解軟體可以繞過 Apple 的數位版權保護機制。此機制同時允許開啟使用者 iTunes，使 iPhone 與 iTunes 同步，將 APP 應用程式載入 iPhone 手機中[2]。

2、方法二

由外國網站 iReSign 推出，使用者可以不需要經過 Jailbreaking 進行盜版 APP 應用程式安裝。正規的 iOS 程式開發員都會得到 Apple 發出的電子證書，當該電子證書安裝到 iOS 後，就容許程式開發員安裝開發中的 APP 應用程式到 iPhone 或 iPad 上進行測試。而 iReSign 網站提供收費服務，將你的 iPhone 或 iPad 經由他們公司登記成程式開發員的專用機，安裝證書後就可以用開發員身份來安裝 APP 應用程式，用這方法來騙 iOS 這些都是開發版 APP 應用程式。不過此收費網站無保障，證書亦有有效期限，當證書過期後，所有經此管道的 APP 應用程式將不能使用[8]。

3、方法三

由 Alexey V. Borodin 提出，即以自行架設的伺服器取代 Apple 用於 In-App Purchase 的伺服器，讓 APP 應用程式所發出的請求實際上由 Borodin 架設的伺服器來處理，假的伺服器會回傳一個 Receipt，讓 APP 應用程式以為使用者已經付了錢，故使用者能取得所需的內容[10]。

(二)以 Android 為核心的 APP 應用程式

Android 作業系統雖然為開放式架構，但為了防止 APP 應用程式被盜用，Android Market 提供了一套授權服務機制，稱為 Android Market License Verification Library

(LVL)，使得每位 APP 應用程式開發員可以引用這套機制於他開發的 APP 應用程式中。有了這套機制，每一支 APP 應用程式都可以在執行時獲得它的被授權狀態，然後決定是否讓它繼續執行。然而根據Andrew Kameka的報導，Android Market License Verification Library 已經被破解[1]。由於 Android 主推開放式架構，所以除了官方網站 Android Market (現稱為 Google Play)可提供下載外，另有許多自營網站也提供 APP 應用程式下載服務，這些網站未必提供授權服務機制。Android 的 APP 應用程式流通性相當高。其它盜版方法如下：

1、方法一

連結至檔案分享網站 www.4shared.com，搜尋並下載 APP 應用程式的 APK 檔，並將該檔轉移至 Android 手機或平板電腦，接著使用檔案搜尋原儲存的資料夾，直接點選後即可安裝，這些都是被破解過的 APP 應用程式[3]。

2、方法二

使用者直接對一付費的 APP 應用程式進行反組譯的工作。首先將 APK 檔案解壓縮，將 classes.dex 檔解壓出來，然後使用工具反組譯出 Java 檔，即可看到原始程式碼了。接下來尋找驗證的指令，只要修改指令跳過驗證結果，使得不管驗證是否成功，程式皆能執行。

(三) APP 應用程式開發員設計的保護機制

除了由 Apple 與 Google 公司等供應商所提供的防盜版機制外，APP 應用程式開發員因為供應商的數位版權保護機制屢屢被破解，使得 APP 應用程式開發員不得不為自己所撰寫的程式加入保護機制。因為只要有一份 APP 應用程式被盜版，網路上馬上就數以千計的盜版 APP 程式出現。一般最常見的是 APP 應用程式開發員將他的 APP 開發成 APP 下載安裝時要註冊使用者帳號及密碼，後端伺服器儲存每位使用者的帳號及密碼的資訊，每當 APP 要執行時，使用者要先登入使用者帳號及密碼，透過與後端伺服器驗證是否正確，來決定是否讓 APP 應用程式繼續執行。這種方法非常簡單，也相當容易被破解。例如將合法者的帳號及密碼公開或給親朋好友，或者利用反組譯方式跳過帳號及密碼的驗證。

其它的保護方法包括 APP 應用程式開發員將他的 APP 開發成 APP 下載安裝時回傳行動裝置的裝置碼到後端伺服器，所謂行動裝置裝置碼是一種序號，此序號是唯一的，不會有別人的行動裝置跟你的序號相同。例如每支手機上有一個 IMEI 碼，IMEI 是國際移動裝備辨識碼，也就是一般俗稱的手機序號[17]。每當 APP 要執行時，先取得使用者行動裝置的裝置碼，透過與後端伺服器驗證裝置碼是否符合，來決定是否讓 APP 應用程式繼續執行。這種方法可以限制 APP 只能在同一台裝置上使用，縱使 APP 檔案被複

製到其它裝置，也會因裝置碼不符而無法執行。此方法雖然能夠防止一般使用者盜版，但對高明的駭客而言，仍然能夠利用反組譯方式跳過驗證裝置碼的驗證。

四、防止 APP 應用程式盜版之方法

我們可以觀察到駭客常用的手法為反組譯。反組譯的目的不外乎修改 APP 應用程式跳過網路連線或跳過驗證程序。因此一個防止 APP 應用程式被盜版的方法為強迫 APP 應用程式一定要網路連線，而且無法跳過驗證程序。在本研究中，我們提出的方法整合 APP 應用程式販賣商場平台，及 APP 應用程式開發員自己架設的伺服器。

本文提出的 APP 應用程式保護架構主要是針對 APP 應用程式執行時所需要的相關資源檔，例如圖片與音效等檔案進行存取管控。使用者必須在每次執行 APP 應用程式時先進行網路連線及驗證程序，通過驗證程序之後，APP 應用程式才能取得資源檔，駭客若反組譯使 APP 應用程式跳過網路連線或跳過驗證程序，此 APP 應用程式將會因缺乏這些資源檔而無法執行。詳細的方法論述如下。

(一) APP 應用程式購買安裝階段

1、步驟一

使用者在 APP 應用程式販賣商場購買一 APP 應用程式，透過 HTTPS 安全連線機制，裝置使用者成功完成付費 APP 應用程式交易。

2、步驟二

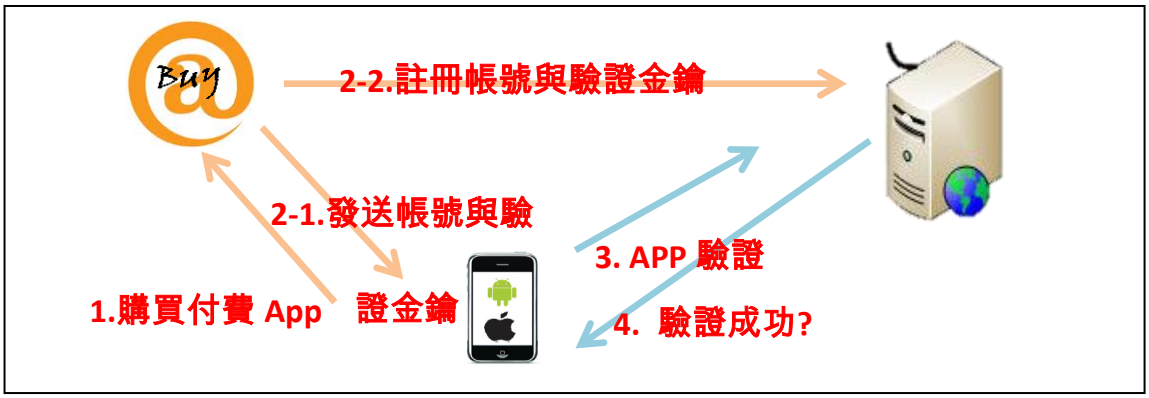
由 APP 應用程式販賣平台發送一組帳號與驗證金鑰給使用者及 APP 應用程式開發員所架設的後端伺服器。後端伺服器為此一使用者的這一個 APP 建立一組對應的帳號及儲存它的驗證金鑰以及資源檔。

3、步驟三在此同時，使用者開始安裝此 APP 應用程式，此 APP 應用程式安裝過程中，畫面提示輸入帳號及驗證金鑰，使用者輸入後，APP 應用程式將輸入帳號及驗證金鑰送到後端伺服器比對。

4、步驟四

若比對吻合，則完成安裝程序，並將帳號及驗證金鑰儲存於行動裝置中。

此安裝程序完成，後端伺服器會為此帳號進行註記，表示安裝已完成。未來若有以此帳號來進行安裝者接不接受，視為盜版。也就是說此帳號僅提供一次性安裝。整個購買安裝程序如圖二所示。



圖二、APP 應用程式購買安裝程序

參、研究結果

一、APP 應用程式執行階段

當 APP 應用程式安裝成功後，每次 APP 應用程式執行時，APP 會要求使用者裝置與後端伺服器進行網路連線(例如 WiFi 或 3G 網路等)以進行重驗證程序，每次連線都是建立在 SSL 管道上，即所謂的 HTTPS 連線，以確保每次的資料交換不會被窺視或監聽。重驗證程序如下:

1、步驟一

APP 應用程式執行時，先上傳帳號及驗證金鑰。

2、步驟二

當驗證成功後，後端伺服器選取一個隨機產生的驗證碼來更新驗證金鑰，並下載到行動裝置中供下次驗證使用。

3、步驟三

APP 回送 Ack 訊號表示收到更新驗證金鑰。後端伺服器複寫更新驗證金鑰到相對帳號中。

4、步驟四

將該 APP 所需用到的資源檔下載到 APP 程式中，使得 APP 程式能夠執行。

假如後端伺服器沒有收到第三步的 Ack 訊號，或者是再傳送中掉封包，後端伺服器將同時儲存原有的驗證金鑰與更新後的驗證金鑰，這是因為怕第二步的資訊如果因網路問題而無法傳遞到行動裝置中，則行動裝置中的驗證金鑰無法更新，或者是第二步的資訊有傳遞到行動裝置中，而第三步的 Ack 訊號沒有傳遞到後端伺服器，如此行動裝置中的驗證金鑰有更新。所以在後端伺服器沒有收到第三步的 Ack 訊號時，同時儲存原有的驗證金鑰與更新後的驗證金鑰以供下次驗證時使用。整個 APP 應用程式執行程序如圖三所示。

圖三、APP 應用程式執行程序

二、安全性分析

從駭客的角度來看，他勢必要取得資源檔，很不幸的是，他要是沒有依照 APP 應用程式執行程序，是無法取得資源檔的。這是因為駭客縱使利用反組譯方式來修改 APP 內部程序，跳過認證程序(步驟一)，步驟二、三、四皆無法執行。另有方法假設一駭客購買一合法 APP，然後複製檔案給他人，將會造成自己的 APP 無法使用，這是因為驗證金鑰每次都在改變，只有拿到最新的驗證金鑰之 APP 才能順利執行，通過認證程序。也就是說購買一份合法 APP，就只有一份 APP 可以正常執行。如果駭客購買一合法 APP，然後故意在步驟二或步驟三斷線，使得後端伺服器將同時儲存原有的驗證金鑰與更新後的驗證金鑰，他也得不到任何好處，因為他複製檔案給別人使用時，只要有一人完成四個步驟，這個 APP 將與後端伺服器重新共享一把新的驗證金鑰，其他版本的 APP 將無法通過認證使用。

肆、系統實作

我們實作的系統是使用 PHP 當後端語言使用，並在 Android 手機(Galaxy Nexus)以及 Emulator 上測試，程式使用 Android 4.4 版本的 API(此為最新相對穩定版本)，此系統成功實現了每次開啟程式都要透過一個驗證碼跟後端的伺服器驗證並且拿取資源檔，不管是驗證失敗或者是複製檔案，都只能同時有一個 Client 端來使用 APP 正常的

服務，並且實作了容錯的功能預防網路不穩而導致 Server 跟 Client 端的驗證碼沒有同步，並且資源檔是以 Streaming 方式傳送本機端不會有檔案讓攻擊者可以利用，而且可以每次都判斷使用者的權限或是付費分級，來決定要給使用者甚麼資源檔，沒有通過驗證的 APP 對使用者或攻擊者來說同樣都沒有任何功能可以使用。

一、步驟一

id	pw	check	check2
test	test	838326360	2049799634
aaa	aaa	1587743580	2064568513

圖四、後端資料庫內容

圖四為我們實作的系統的驗證部分的資料庫架構，除了帳號密碼以外有兩份驗證碼，兩份的原因是為了怕連線如果有異常則 Client 與 Server 兩端的驗證碼會不同步，才有辦法在不同步的情況下繼續作驗證。

二、步驟二

當驗證成功後，後端伺服器選取一個隨機產生的驗證碼來更新驗證金鑰，其中隨機產生的 Function 為 PHP 的 mt_rand，這裡可以再套用更嚴謹的密碼使用亂數，或是增加驗證長度，產生後的驗證碼下載到行動裝置中供下次驗證使用，並且同時把驗證後的金鑰備份。

三、步驟三

id	pw	check	check2
test	test	1731308304	838326360
aaa	aaa	1587743580	2064568513

圖五、後端資料庫內容(更新驗證碼後)

圖五為本實作的資料庫更新驗證碼後的格式。可以發現的是即使正常驗證都會把驗證後的驗證碼備份到驗證碼 2(也就是把 838326360 寫到 check2)，如果是錯誤的情況更新了新的驗證碼也仍有第二組驗證碼可以讓 Client 端重新驗證。其中顯示驗證碼在螢幕上只是方便展示驗證碼的亂數更新以及 Debug，實際運作驗證碼是存在 /data/data/<package name>/shared_prefs 中的 xml 檔，使用者並不會知道驗證碼以及背景的驗證程序，只知道他們有輸入登入的帳號密碼，正常來說隱私資訊儲存在程式的儲存空間是不會被其他程式拿到，但如果手機有 Root 或是有其他漏洞(如 Content Provider 漏洞)會有被竊取的風險，但此驗證碼如果被攻擊者知道，也只會讓攻擊者在另外一台設備得到一次性驗證的可能性(或是直接把整份檔案 Copy 搬移到另一台手機也可以達

到此功能)，並不會對我們的驗證系統產生任何破壞或是影響，他下次想要正常拿到資源檔還是得用新產生的驗證碼，或者是如果想限制使用者只能在一台設備中使用 APP 的功能也可以綁定 Device ID 並每次驗證都檢查。

四、步驟四

本實作是以 Streaming 方式把資源檔(音樂檔 mp3 格式) 傳送到手機 APP 上讓 Client 端可以撥放，所以 Client 端並不會有檔案保留，每次都需要做驗證才可以使用此服務(獲取音樂的服務，或是其他廣播或是語言學習 APP)，其中後端的處理權限是在另外一個 PHP 檔來實現，而且可以藉由判斷該使用者的權限，來決定要給使用者甚麼服務，比如說撥放片段試聽音樂或者是讓使用者存取完整的檔案。

伍、結論

目前我們所提出的防止盜版 APP 應用程式的機制，是一個非常有效的方法。這個方法提高了破解 APP 應用程式防盜機制的困難度，能夠有效的保障到正版的 APP 應用程式合法的使用。我們所使用的觀念為強迫 APP 應用程式一定要與網路連線，而且需經網路驗證程序，然後才能取得資源檔讓 APP 應用程式執行。縱使高明的駭客有反組譯的能力，能修改 APP 應用程式跳過網路連線或跳過網路驗證程序，亦無法取得資源檔，所以無法執行 APP 應用程式。如此不但可以保障那些下載正版 APP 應用程式使用者的權益，而且還可以同時保障到 APP 應用程式開發員的智慧財產權與應有的獲利。雖然現在我們可以用這種方法來暫時緩和盜版 APP 應用程式的問題，但人外有人，天外有天，或許這是可以暫時的解決盜版的問題，但這並不能保證是個永遠安全的保護正版 APP 應用程式的方法。在這個日新月異的世代裡，要持續地精進我們的防盜機制才可以維護合法的版權保護。

參考文獻

[1] Androinica。2012 年 11 月 12 日，取自 androinica.com/2010/08/google-responds-to-android-market-app-protection-cracks/
[2] BlogTechnika。2012 年 11 月 12 日，取自 www.blogtechnika.com/how-to-get-paid-iphone-apps-for-free/
[3] BlogTechnika (2012)。2012 年 11 月 11 日，取自 www.blogtechnika.com/how-to-download-paid-android-apps-for-free/

[4] Brian Chen (2014) 2014 年 Q1 平板市場：Android 市佔率 65.8%，大幅領先 iPad technews.tw/2014/04/29/strategy-analytics-android-tablet-shipments-65-8-q4-2014-ios-fell-28-4-windows-secured-5-8/

[5] Forouzan, B., (2008). *Cryptography and network security*. New York: McGraw-Hill.

[6] Kyle (2012)。 蘋果 iPad 於 2012 年第二季市佔率達 68%。 財團法人國家實驗研究院科技政策研究與資訊中心資訊服務處科技產業資訊室市場報導。2012 年 11 月 8 日，取自 cdnet.stpi.narl.org.tw/techroom/market/eecomputer/2012/eecomputer_12_049.htm

[7] Kyle (2012)。2012 年第三季 iPad 市佔率跌至 50.4%。 財團法人國家實驗研究院科技政策研究與資訊中心資訊服務處科技產業資訊室市場報導。2012 年 11 月 12 日，取自 cdnet.stpi.narl.org.tw/techroom/market/eecomputer/2012/eecomputer_12_069.htm

[8] PressByte。2012 年 11 月 12 日，取自 www.pressbyte.com/3443/iresign-install-ipa-jailbreak-download/

[9] Rosenblatt, W., Rosenblatt, B., Trippe, W. & Mooney, S. (2002). *Digital rights management: Business and technology*. New York: M&T Books.

[10] T 克邦。2012 年 11 月 12 日，取自 www.techbang.com/posts/10113-ios-app-billing-mechanisms-are-compromised-hackers-find-a-way-to-free-access-to-content

[11] 廖月娟、姜雪影、謝凱蒂 (譯) (2011)。賈伯斯傳。台北市: 天下遠見出版股份有限公司。

[12] 陳穎芃(2014)安卓全球市占登頂 難逾 85%極限。中時電子報報 11 月 4 日。

[13] 陳文淦 (2012)。App 的發展趨勢與語音搜尋的應用。2012 年 11 月 11 日，取自 www.slideshare.net/BlancChen/mobile-app2012-topology

[14] 葉亭均 (2012)。Android 市占率飆到 75%。經濟日報，11 月 3 日，A3 版。

[15] 維基百科 (2012)。2012 年 11 月 11 日，取自 zh.wikipedia.org/wiki/Google

[16] 維基百科(2012)。2012 年 11 月 11 日，取自 zh.wikipedia.org/wiki/Cydia

[17] 維基百科 (2012)。2012 年 11 月 11 日，取自 zh.wikipedia.org/wiki/IMEI