國立交通大學

電機資訊學院 資訊學程

碩 士 論 文

設計與實作以憑證為認證基礎的無線網路路由器

Design and Implement a Wireless Router with Embedded RADIUS Server and Certificate Authority

研 究 生：鄭宗益

指導教授：曾文貴 教授

中 華 民 國 九 十 三 年 十 二 月

設計與實作以憑證為認證基礎的無線網路路由器

Design and Implement a Wireless Router with Embedded RADIUS Server and Certificate Authority

研 究 生：鄭宗益　　Student：Tzung-I Cheng

指導教授：曾文貴　　Advisor：Dr. Wen-Guey Tzeng

國 立 交 通 大 學

電機資訊學院 資訊學程

碩 士 論 文

A Thesis

Submitted to Degree Program of Electrical Engineering and Computer Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

December 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年十二月

設計與實作以憑證為認證基礎的無線網路路由器

學生:鄭宗益　　指導教授：曾文貴 博士

國立交通大學電機資訊學院 資訊學程（研究所）碩士班

摘　　要

因為無線網路的日益盛行，當使用者在享受無線網路的方便性背後，同樣地也面臨到網路安全的問題，也就是可能會被竊聽的潛在危險。有心人士只要在無線網路覆蓋的區域內，利用掃瞄封包後加以儲存複製的方式，就有方法可以組合分析而得到傳輸的內容。

雖然 IEEE 802.1x 已經規定了憑證認證的機制，可是它必須與 RADIUS 伺服器和 CA 協同工作。這樣的架構是可以在中、大型企業內的有線與無線網路上實現，卻不是一般家庭使用者，甚至小型企業所願意負擔與管理的，因為使用者必須要另外架設RADIUS伺服器與CA，以及整套的運作機制後才有辦法與意願來加以架構與使用的。

在考量原無線路由器的可擴充性與可行性之後，筆者試著將 RADIUS 伺服器與 CA 整合進無線路由器當中，再加上容易操作管理的 WEB GUI 介面，不只成功的達到網路安全的需求，也讓使用者更方便的操作與管理。

關鍵詞：無線網路、802.1x、RADIUS、CA、憑證。

Design and Implement a Wireless Router with Embedded RADIUS server and Certificate Authority

Student: Tzeng-I Cheng Advisor: Prof. Wen-Guey Tzeng

Degree Program of Electrical Engineering Computer Science
National Chiao Tung University

Abstract

When accessing a wireless network, a user will face many security problems.

Someone can sniffer data packets within the wireless coverage. If data packets were not encrypted, a hacker can easily combine those packets and read the content. Although there's an authenticate mechanism base on certificate, it should work with a

RADIUS server and a certificate authority. Such kind of architecture can only be used within a large or medium scale enterprise. SOHO user or small company may not has such equipments to use. To protect the wireless network security more easily, i think it's necessary to build RADIUS server and certificate authority into a wireless router.

After done this program, a useful system came out. It can be easily configured and operated with IEEE 802.1x security policy. People won't need to waste their money and time to buy and install another RADIUS server and certificate authority. Only one wireless router can do such things.

Keywords: Wireless network, 802.1x, RADIUS, CA, Certificate.

誌　　謝

首先我要感謝我的指導教授曾文貴教授，在我的求學期間與研究過程中給予我的諸多教導與指引。另外還要感謝各位口試委員對本篇論文所提出的批評與建議，使學生我獲益良多。還有感謝我的父母，我的太太，我的家人給予我精神上的支持，讓我得以順利地完成此篇論文。最後謝謝我的夥伴吳俊達，藉由他的幫忙，讓我順利地解決了許多實作過程中遇到的問題。謝謝大家！

目　　錄

表目錄

圖目錄

頁次

圖片

一、 緒論

網路設備的發展趨勢，重心已經逐漸地從有線網路產品，轉移到應用無線網路的產品。這是因為在這快速變遷的時代，無線網路代表的是沒有束縛。隨插即用，隨處可用代表著訊息更能快速的掌握，越能快速掌握訊息，代表著可以更有效率，更能掌握致勝的先機。在無線網路的產品方面，最有代表性的有無線網路卡、無線存取點（AP，Access Point）、無線路由器（Wireless Router）等等。筆者有幸參與了無線路由器的發展計畫，有機會能將所學應用在發展過程當中。

1.1 研究動機

無線路由器原本就整合了無線存取點以及寬頻路由器的功能。在原有的系統中，無線網路安全的功能已很齊全。不過，在其中的 802.1x〔4〕憑證認證的部份，因為原來的設計只是想在無線存取點的基礎上增加安全的認證，所以它的身分只是傳遞認證者（Authenticator），將憑證驗證的功能留給後端的認證伺服器（Authentication Server）處理。而這個後端的認證伺服器，必須要在連線狀態，而且擁有使用者的認證相關資料。關於這個 802.1x 的部分，會在底下第 3.1 節詳述。

這個認證伺服器，確實對一般的使用者造成不小的困擾。因為若沒有了認證伺服器，於是在各種開放空間的使用情境下，皆無法享有無線網路安全（802.1x）的保障。怎樣讓使用者在不需要架設認證伺服器的情況下，也能享受無線網路安全（802.1x）的保障呢？我提出了將原有的系統增加內建 RADIUS（Remote Authentication Dial-In User Service）Server〔2〕的功能，而這個內建 RADIUS Server 的功能應該要可以減少使用者在架設認證伺服器所遇到的問題。

這樣一個簡單的構想開始了這個計畫。在這個構想之下，接著的問

題就是：是否可以把憑證認證（CA，Certificate Authority）的功能也一起提供？因為，如果能夠把 CA 的功能整合進來的話，使用者就可以不需自己管理各種憑證了。於是，整合 CA 的功能成了我的第二個目標。

假設有了 CA 之後，第三個問題是，使用者要如何減輕管理憑證的負擔？我的答案是必須要有結合 WEB 管理介面的管理工具，讓管理憑證的工作變得更輕鬆。這是我的第三個目標。

雖然有些使用者可能想說利用 VeriSign 這樣的憑證認證單位，或者是利用 Microsoft 的 Certificate Server 來幫忙簽發憑證。無論各種憑證的解決方案，我相信許多的使用者，都有可能會遇到使用上的瓶頸。所以我的出發點很簡單，只要把憑證輸出入的 GUI（Graphic User Interface）介面設計妥當，容易操作，就能夠減少使用者的麻煩。當然位在上層 GUI 與下層 Process 之間做為溝通橋樑的 CGI（Common Gateway Interface）也必須一併做好。這是我的第四個目標。

所以簡單來說，我的目標有四個：

甲、 內建 RADIUS Server

乙、 內建 CA

丙、 設計與新增容易管理憑證的 WEB GUI 介面

丁、 量身打造 CGI

1.1.1 簡介原有系統

以下將簡介原有的無線路由器功能：

- **無線存取點的部分**，就是提供無線端與有線端的介面轉換，以及封包（Packet）的 Forwarding，但不包括繞路（Routing）的功能。使用者透過無線網路卡與無線存取點連線，提供連線的需求。
- **路由器的部分**，就是在無線存取點的後端加上繞路（Routing）與連線上網的功能。所謂 Routing 的功能就是將封包由無線存取點所在的 LAN 端轉送至路由器的 WAN 端，另外由於 LAN 端使用的

Private LAN 的 IP Address 與在 WAN 端的 Public IP Address 並不在相同的網路上面，所以它也有 NAT/PAT 的功能。原有的系統提供了三種連接網際網路的方式：

1. 固定 IP 連接，通常用在 ISP（Internet Service Provider）給予固定 IP 的時候。
2. DHCP（Dynamic Host Configuration Protocol）連接，用在 ISP 給予非固定 IP 的時候，通常每次獲得的 IP 都有可能不同。
3. PPPoE （Point-to-Point Protocol over Ethernet），大部分的 ISP 都使用這種方式，尤其是採用 ADSL Modem（Asynchronous Digital Subscription Line Modem）撥接的時候。

關於無線網路的使用頻率與規格，目前 IEEE（Institute of Electrical and Electronics Engineers）訂了有 3 種規格（802.11b、802.11a、802.11g）：

用簡表來表示:

表一、802.11 的頻帶規格

項目 / 規格	頻帶	速率	最遠距離	相容性
802.11b	2.4 GHz	11 Mbps	100~150ft	與 11g 相容
802.11a	5 GHz	54 Mbps	25~75ft	與 11g、11b 不相容
802.11g	2.4 GHz	54 Mbps	100~150ft	與 11b 相容

原有系統因為只使用一支天線，所以在同一時間只能支援一種無線規格，例如只能選擇使用 802.11a 或 802.11g/b。802.11b 和 802.11g 因為頻帶相同，所以它們能相容。

1.1.2 擴充性與未來支援

當初在規劃原有系統的時候就已經考慮到未來的需求，所以採用了 Embedded Linux。最主要的著眼點就是未來的擴充與支援，還有就是網路上的 Open Source 相當的多，而且相容性都相當的不錯，目前在系統上執行的一些軟體，都只需要少部份的修改就可以整合進去。

1.2 研究方向

跟據我的目標，我將研究方向區分為四個。分別是安全性、相容性、多樣性與易操作管理，以下將分別描述。

1.2.1 安全性

在無線網路的網路安全上，一般是介在無線存取點與無線使用者端的部份，使用者必須要考慮的是使用現有的設備是否可以保護無線網路的安全，而系統開發者必須要考慮的是舊有規格的缺陷是否可以彌補，新的安全規格是否可以開發完成，並且通過認證單位的認證。在原有系統當中，使用者有各種不同的保護措施，以下列出各個不同等級的保護措施：

- 預設值：使用預設的 SSID（Service Set Identifier）是不會有任何的安全措施。任何使用者，只要是 802.11a 或 802.11g 模式相同都可以連上。
- 取消廣播 SSID：就是說在無線存取點定時廣播送出的 Beacon 訊息當中是不會帶有 SSID 的。不過只要偷聽者有心，還是會聽到無線使用者端送出的封包中帶著的 SSID。在前兩個等級所傳送的封包都還是明文的狀態。

- 啟動 WEP（Wired Equivalent Privacy）金鑰加密：傳輸的封包會被使用 RC4（1987 由 Ron Rivest 發展，RSA 公司）來加密。RC4 是一種資料串流加密的方法。
- 啟動 MAC 位址的認證功能：使用者可以在無線存取點或在 RADIUS Server 中保存一份使用者 MAC 位址的列表，這份列表可以是正向表列，允許存取網路。也可以是負向表列，不允許存取網路。在一般的應用中通常會以正向表列居多。
- 啟動 EAP-TLS （Extensible Authentication Protocol- Transport Level Security，RFC2716）〔1〕認證：這個認證動作，需要 CA 事先簽發的憑證，也需要 RADIUS Server 的共同參與，在認證過程中認證使用者的憑證。通過認證之後，系統還可以啟動動態更改加密金鑰的機制，間隔越短，安全性越高。
- 啟動 WPA（Wi-Fi Protected Access）：WPA 是新一代的安全加密標準，在新的 IEEE 802.11i 標準中收錄。

無線網路從使用 WEP key 來保護，然後再進步到使用 IEEE 的 802.1x 的架構。現今在頻寬也提高到 54Mbps 的情況下，更有硬體協助加/解密的動作，網路的安全將更可以確保。本論文所探討的部份就是屬於 802.1x 的 EAP-TLS 的應用環境的一個改良。它分成三個角色，將在第 3.2 節詳細解釋。在現有系統中的無線模組控制部份是由 Instant 802 這家公司所開發，這個部分（Hostapd）也是本系統的核心部分之一，僅次於 Embedded Linux Kernel。

我對本系統所做的是屬於架構上的“改變”，以及部份功能的新增，它還是擁有一樣的安全性，只是系統的可信賴度被我提高了，它不會因為與 RADIUS Server 和 CA 的連線斷線而無法運作，因為它們都在同一系統上。而且使用者在維護與操作上的複雜度可以減少許多，這是可以看得到的效果。

1.2.2 相容性

當我選擇有關 RADIUS Server 與 CA 的開放原始碼的時候，就有特別注意到相容性的問題。最後選擇了 FreeRADIUS 與 Openssl。這兩份開放原始碼在網路上已公開許久，應該不存在相容性的問題。事實證明，讓這兩份軟體在系統上執行並不是太大的問題，倒是這兩份軟體必須能夠透過 GUI 來設定與管理則必須要另做一番的規畫與串聯。

1.2.3 多樣性

雖然我將 RADIUS Server 和 CA 同時加在系統上，但是它們可以不需要同時執行。在第 4.6 節，我將會詳述 5 種不同組合的操作模式。為何要如此設計呢？基本上，必須考慮各種不同使用者的各種不同操作方式。也許使用者已經有一台 RADIUS Server 在自己的網路上運作，例如 Microsoft Windows 2000 Server 中提供的 Internet Authentication Service，他就可以不需要我們內建的 RADIUS Server。也許使用者想自己另外架設一台機器執行 Linux 作業系統，一樣也是執行 Openssl 來做為他的 CA，那也是可以的。只要他能記得將憑證輸出並送給適當的對象，無線網路的安全也是一樣可以確保的。只不過執行 Openssl 若是沒有配合適當的 WEB GUI 來做為輔助的話，全部的命令將會是從命令列輸入，那將會是非常辛苦的一件事。

1.2.4 易操作管理

使用者可能在操作外部的 RADIUS Server 或者外部的 CA 的時候，可能不小心選錯了一個選項，或者在命令列操作錯誤了之後，將會造成無法連線等等非預期的結果。而且接下來除錯的時間成本可能會更高，畢竟使用者並非程式開發者，甚至會不知道出錯的環節在那裡。針對這樣的問題，在我的網頁與 CGI 的設計當中，只有在需要使用者輸入的部分，才開放讓使用者輸入，伴隨著必要的錯誤警告與處理，讓使用者出錯的機率減少，這也是我當初設計的目標之一。

1.3 研究成果

在第一節中提到，我的研究目標有四個:

1. **內建 RADIUS Server**
2. **內建 CA**
3. **設計新增容易管理憑證與使用的 WEB 介面**
4. **量身打造的 CGI**

這四個目標到最後終於順利的達成。首先，在內建的 RADIUS Server 部份，我選擇了使用 FreeRADIUS 的開放程式碼，在 porting 的過程中，由於原有的系統是 Embedded Linux，所以我可以先將 Linux 安裝在 PC 上先行驗證，因為 FreeRADIUS 必須要使用到 Openssl 的 library，所以計畫中的內建 CA 就用 Openssl 來架構。於是憑證的功能有了之後，接著必須讓系統原有的 RADIUS Client 能夠與新的 Embedded RADIUS Server 溝通。這個部分，我必須要將 RADIUS Client 指向 RADIUS Server 的設定檔做部分修改，然後再讓 RADIUS Client 重新使用新的設定，就可以讓我的 FreeRADIUS Server 與原有的 RADIUS Client 溝通了。在完成溝通之後，接著的挑戰就是如何管理憑證與使用者的介面了。在 802.1x 的 EAP-TLS 的 Protocol 當中，使用者必須把他的憑證上傳給 RADIUS Server 做憑證的認證，這代表者 RADIUS Server 與 CA 都可以不必保留使用者的憑證。於是管理 RADIUS Server 與 CA 的畫面可以更簡單，可以參考第五章的實際操作說明。一個完整 CA 的功能其實還有其他的部分，例如有效與無效憑證的管理，憑證效期的延長，這些功能因為不是必要條件，不影響憑證的驗證，而且系統仍存在著 Flash 與 DRAM 的限制，於是乎暫列在未完成工作中。由於內建 CA 與 RADIUS Server 的功能，CGI 必須能夠提供憑證的匯入與匯出。這個部分比較複雜，必須重新設計新的 CGI。因為不容易整合到原來處理設定的 CGI，所以我另外撰寫了一個新的 CGI 程式，並且另外設計了網頁來特別處理

憑證的匯入與匯出。最後，在經過一番的整合性的操作之後，使用者終於可以透過EAP-TLS 與新的系統連線，憑證的認證功能已可以由內建的RADIUS Server 來提供。至此，證明了無線網路路由器整合內建了RADIUS Server 和 CA 是可行的。

二、 相關研究

2.1 802.11 無線網路標準（WLAN）

IEEE 這個組織到了 1997 年才規範了 802.11〔8〕無線網路的標準，1999 年通過了 802.11b 與 802.11a 的相關標準，在 2003 年最後通過了 802.11g 的標準。802.11 無線網路的主要優點，主要在於不用佈線，只要使用符合 802.11 規格的無線接取點以及無線網路卡，使用者可以在無線網路的覆蓋範圍內自由的使用。

802.11 標準中允許兩種通訊方式。第一種是使用一個無線存取點（AP，Access Point），讓無線的用戶端連上之後做為資料傳輸的中心，這種模式稱為**基礎模式**（Infrastructure Mode）；第二種是以點對點方式進行連接，這種模式稱為**點對點模式**（Ad hoc Mode）。

在**基礎模式**中，無線存取點做為一個轉送封包的角色，封包可以到另一個無線接取點或者經過路由器再上網際網路，這就像是把兩個不同的網路實體層，無線網路層與乙太網路層橋接（Bridge）起來一般。另外，無線接取點之間，甚至無線接取器與無線路由器之間還可以使用 WDS（Wireless Distribution System）連接起來。基礎模式的連接方式以下圖來表示：

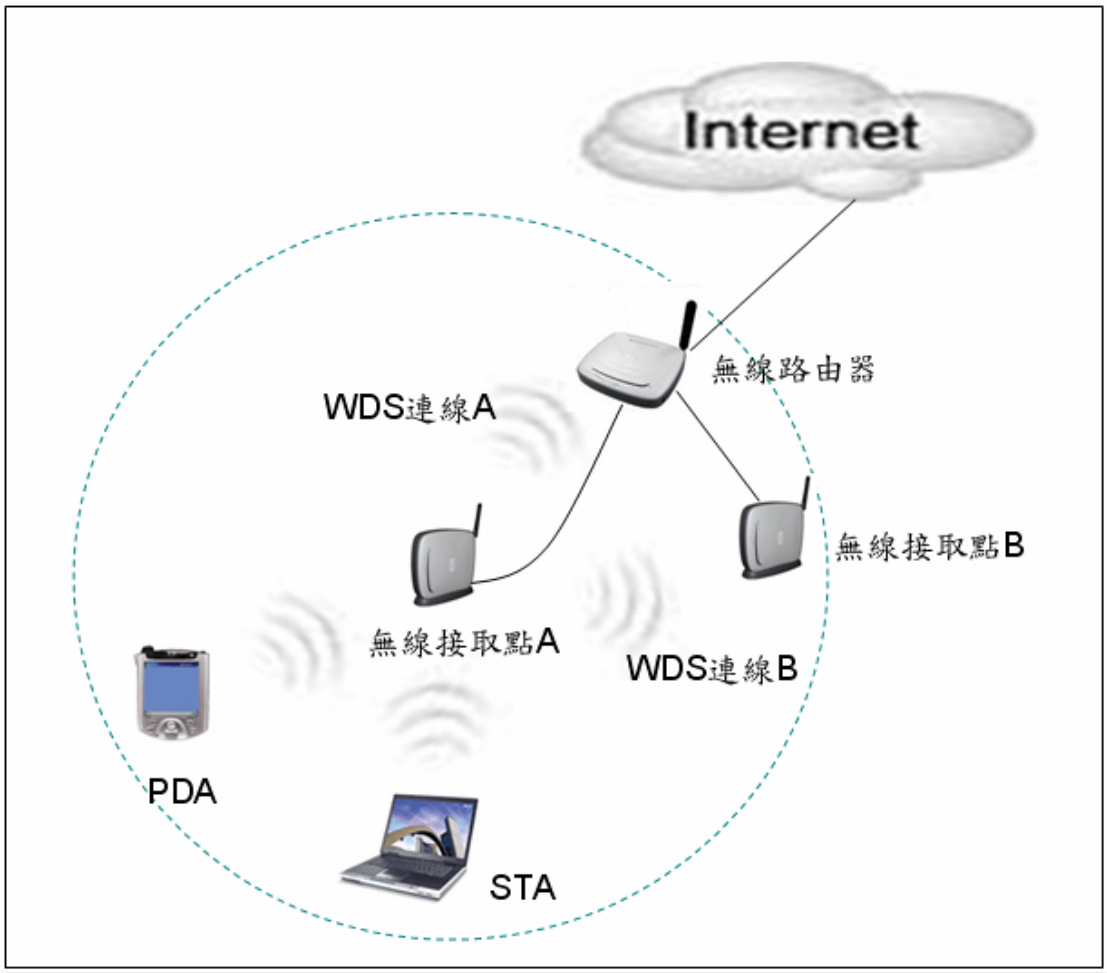

圖 1、基礎模式的連線可能方式

在**點對點模式**中，沒有了基礎模式中的無線接取點的角色，各個無線用戶端在一個可以互相無線電波可以覆蓋的範圍內直接通訊。通常這樣的應用是在一個短時間且短距離內的通訊，例如是在一間會議室中的互相通訊與傳輸檔案等等。通常通訊的發起者必須要設定一個 IBSS ID（Independent Basic Service Set ID）與頻道（頻率），如果還有必要的安全需求的話，互相通訊者就必須再加上使用共用的 WEP Key，來加密與解密無線傳輸中的資料。點對點模式的連接方式以下圖來表示：

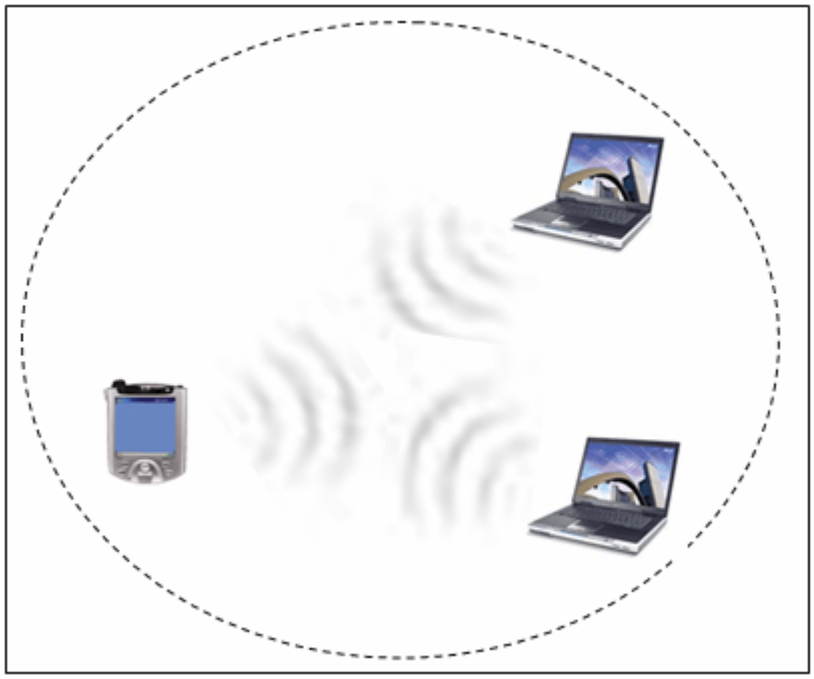

圖 2、點對點模式的連線可能方式

目前在符合 802.11 的通訊設備中，已經有相當多的設備種類，例如桌上型電腦、筆記型電腦、個人數位助理（PDA）、無線接取點與無線路由器等等。市面上的無線網路卡也更加地整合不同的介面，例如 PCI 介面、CardBus 介面、USB 介面、COMPACT FALSH 介面與 mini-PCI 介面等等。

2.2 802.11 無線網路安全機制

根據 IEEE 所定的標準，802.11 無線網路必須要有三項基本網路服務功能：

- 使用者認證（Authentication）

  使用者的身份認證在 802.11 網路中又分成三種：開放系統認證（Open System Authentication），封閉系統認證（Closed System Authentication）以及分享密鑰認證（Shared-key Authentication）。其中分享密鑰認證有時也會因為運作機制的因素，也被稱為挑戰與回應認證（Challenge-Response Authentication）。

- 資料保密（Confidentiality）

  使用者的資料在無線網路傳輸過程中必須要能夠有基本的安全防護，也就是加密，以防止資料的被竊聽。在加解密過程中主要使用 WEP（Wired Equivalent Privacy）與 RC4 的運算方法；我將於 2.3 節中詳細解釋關於 WEP 的運作方式。不過因為 WEP 運作的缺點被越來越被太多人知道與破解，所以又有一新的標準誕生，那就是 WPA（Wi-Fi Protected Access）。我將於 3.3 節中詳細解釋關於 WPA 的運作方式。

- 資料完整性（Integrity）

  無線網路傳輸的資料完整性是用 CRC Check Sum 的方式來確保的。在啟動 WEP 機制之後，又可以對傳送資料多一層保護。

2.2.1 使用者認證

使用者認證分為無加密認證與有加密認證兩種。

802.11 的使用者認證先以下面簡圖表示：

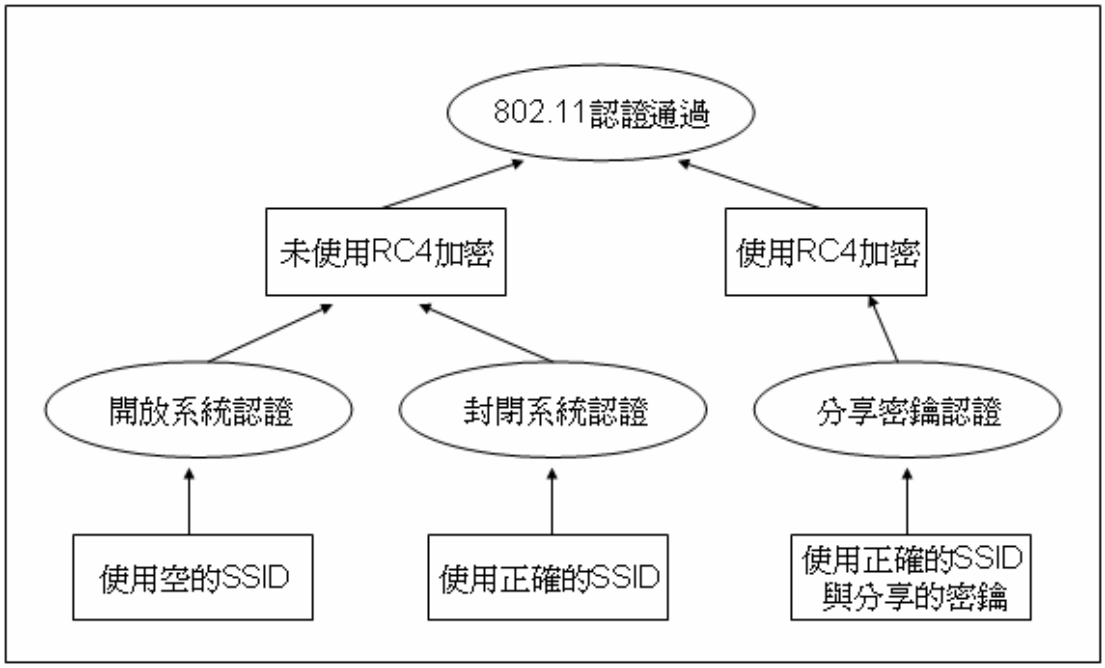

圖 3、802.11 的認證方式

- **無加密認證**

無加密認證是以 SSID（Service Set ID）做為認證基礎。SSID 是無線存取點提供服務的“服務名稱”。一個或多個無線存取點，可以因為用相同的 SSID 而擴大服務的領域。無線使用者只要將自己的網路卡設定使用相同的 SSID，就可以存取無線網路服務，假設在沒有其他的防護措施下。

在這樣的情況下，又可分成兩種認證方式：

第一種是開放系統認證（Open System Authentication），使用者可以用空白的 SSID 來與無線存取點溝通。而無線存取點可以對其回應，在回應的封包中會帶有無線存取點的 SSID。這種方式可以讓使用者透過掃瞄的方式，找到所有可提供開放系統認證的無線存取點。可以說完全沒有防護，但也不是說完全沒有優點，使用者本來就有可能事先不知道 SSID 的名稱，藉著這樣的尋找，是可以幫使用者回想起來。

第二種是封閉系統認證（Closed System Authentication），在這種認證的情況下，無線存取點是不會對空白的 SSID 回應，使用者必須知道正確的 SSID 才可存取。也就是說在尋找無線存取點的時候，是不會看到該無線存取點的 SSID。似乎很安全，可是只要有攻擊者利用可收聽無線封包的程式，例如 AiroPeek。在無線領域範圍內擷取封包後，並且分析其他使用者的連線行為，難保不會讓攻擊者推測出正確的 SSID。就好像【天方夜譚】的故事中，阿里巴巴聽到四十大盜所說的密語一樣，因為聽到四十大盜們每次進山洞所說的話都是“芝麻，開門”，離開的時候說“芝麻，關門”一樣，當認證的機制太簡單的時候，網路安全即形同虛設了。

- **有加密認證**

有加密認證是以分享金鑰做為認證基礎，運作方式是採用挑戰與回應的方式（Challenge and Response）。使用的是 WEP key 做為分享的金鑰，在認證過程中使用 RC4 運算方法做加解密。

先以簡圖表示之。

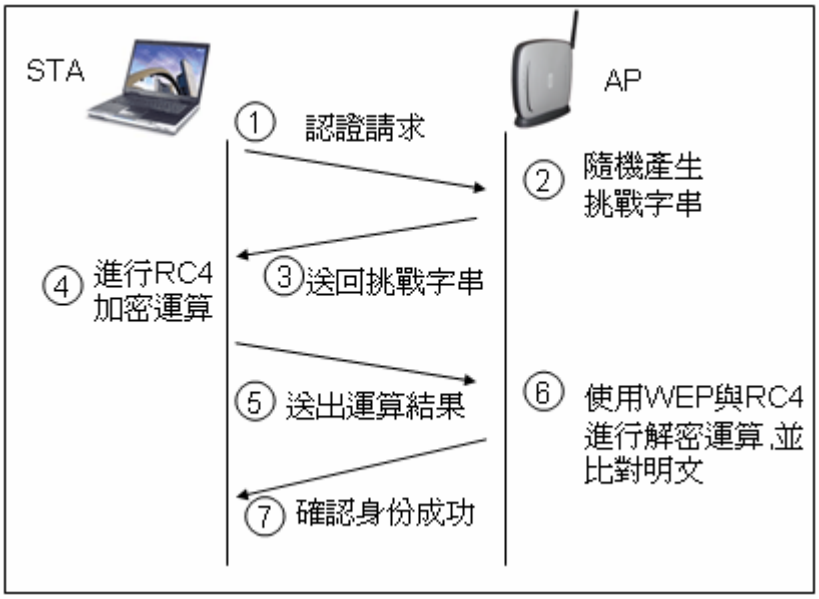

圖 4、分享金鑰的加密認證

在步驟一中，首先無線使用者（STA）對無線存取點提出認證的請求。

在步驟二中，無線存取點隨機產生了一個 128Bits 的挑戰字串（Challenge Text）。

在步驟三中，無線存取點將 128 Bits 的挑戰字串傳給無線使用者。

在步驟四中，無線使用者在收到挑戰字串後，用 WEP Key 及 RC4 來加密。

在步驟五中，無線使用者將加密後的密文回傳給無線存取點。

在步驟六中，無線存取點將收到的密文，用 WEP Key 及 RC4 來解密，然後與原來傳送的明文比較，如果相同，代表通過驗證，如果不同，代表沒有通過驗證。

在步驟七中，無線存取點送回驗證結果，代表願意接受連線或拒絕連線。

2.2.2 資料保密-使用 WEP

由於無線網路的無線電特性, 使得它比起有線網路更容易遭受竊聽。

因此 IEEE 設計了 WEP（Wired Equivalent Privacy），由字義上來看，是希望它能跟有線網路一樣有同等的機密性。關於 WEP 的詳細加解密過程將於 2.3 節中描述。基本上，無線使用者端與無線存取點共用同一把 WEP Key 做為加解密的依據，而且理論上，它不應該會被破解。

2.2.3 資料完整性

在 IEEE 802 的所有規格中，為確保資料的完整性，通常都是使用 CRC Check Sum 的方式來做為完整性的確認，這種方式可以有效防止傳輸過程中的不明因素干擾，最多錯了再重傳即可。傳送與接收的雙方都使用相同的 One Way Hash Function 將傳輸封包中的 Payload 部分進行運算，將得到的 CRC 值進行比對，若數值相同則資料是完整的。

在 802.11 中，傳送的一方可以在傳送資料前，先使用 CRC Check Sum 做資料完整性的確保，然後再使用 WEP Key 做資料的加密。而接收的一方可以在接收資料後，先使用 WEP Key 做資料的解密，再使用 CRC Check Sum 做資料完整性的比對。

2.3 WEP 加解密過程

根據 IEEE 所定的標準，WEP 的設計必需符合以下幾個條件：

1. 必須能夠抵擋暴力攻擊或字典攻擊法。
2. 必須要有自我同步（Self-Synchronizing）的特性。即使傳輸情況惡劣，依然要有保護資料的能力。
3. 可以軟體或是硬體來實現。
4. 必須要全世界能通用的，通過美國的加解密運算的出口管制。
5. 使用者可以選擇使用或不使用 WEP。

接著我們來談 WEP 的加解密的流程。先看下圖的說明：

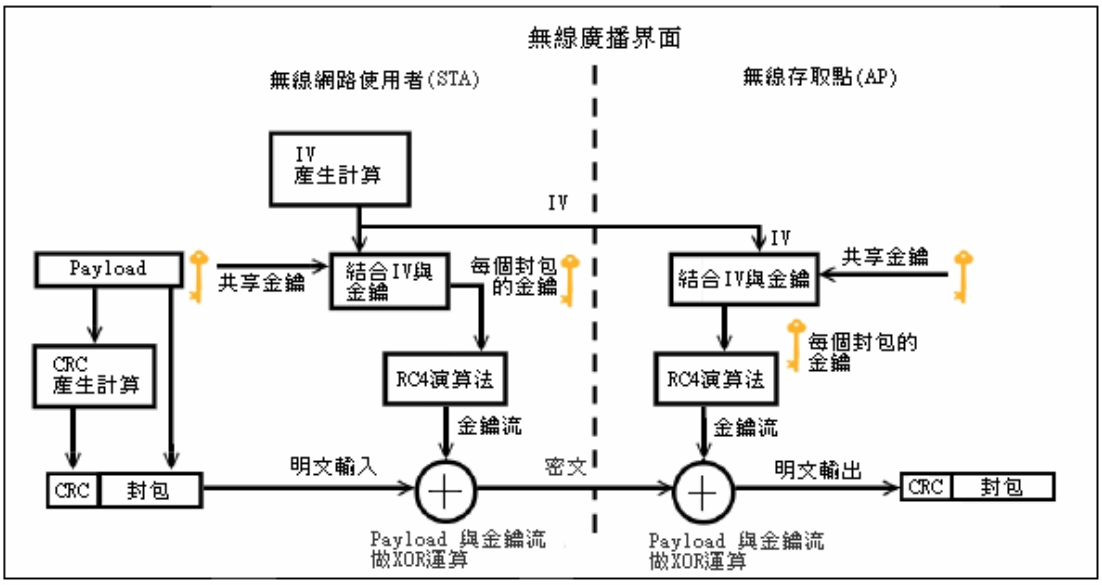

圖 5、WEP 加密流程

資料來源：NIST〔12〕

基本上，WEP 的加密與解密使用的都是同一把密鑰，它必須事先決定與存在通訊的雙方系統上，我們稱為對稱式加解密系統（Symmetric Cryptography System）。原始密鑰的長度可以是 40 位元或 104 位元。可是實際使用時，還必須要加上一個 24 位元長度的 Initial Vector（IV），由無線網路的使用者提供。使用這個 IV 的目的是為了不要讓同一把固定的 WEP Key 太容易遭受破解，所以加上這個 IV 藉以打亂 WEP Key 的組合。所以實際的密鑰長度會變成 64 位元或 128 位元。

2.3.1 WEP 加密流程

總共有五個步驟來描述加密流程：

步驟一，發送端將封包內的 Payload 進行 CRC 的運算，得到 CRC 的查核碼，然後附在封包中。

步驟二，發送端使用產生 IV 的演算法，算出一個 IV。

步驟三，發送端將 IV 及密鑰進行 RC4 運算後，取得加長的加密金鑰。

步驟四，利用加長的加密金鑰，與步驟一產生的封包進行 XOR 運

算，得到密文資料。再附上 24bits 長的 IV 到密文資料中。

步驟五，送出密文資料給接收端。

2.3.2 WEP 解密流程

解密的步驟就反過來：

步驟一，接收端先拿出附在封包中的 IV。

步驟二，接收端將 IV 與共用金鑰進行 RC4 的運算，得到加長的加密金鑰。

步驟三，接收端將收到的封包（密文資料），使用步驟二所獲得的加長的加密金鑰進行 XOR 運算後，便可還原明文資料。

步驟四，從明文資料中拿出 CRC 查核碼，做完整性的檢查。

2.4 802.11 的相關弱點分析

802.11 無線網路的安全問題大概可以分為三類：

- 無線通訊本身

  因為無線網路靠的是無線電波傳遞資料，只要是無線電波可以涵蓋的範圍內都有可能被攻擊者竊聽。如果使用者沒有對傳輸的資料進行加密，則竊聽者很容易就可以組合所得到的資訊。

- WEP 設計有錯誤

  由於 WEP 的某些設計實作上的錯誤，使得在使用 WEP 的效果上無法百分之百保證資料的機密性。例如 IV 向量太短或者固定。40bits 的 WEP Key 不敷使用，在很短的時間內只要攻擊者累積到一定量的封包，它是可以猜出 WEP Key 的。還有 RC4 的演算法也有弱點，WEP 的設計不應將密鑰的部份內容公開，如 IV。

- 設備管理困難度高

  有些網路管理者在管理無線存取點時，一定會遇到金鑰管理的問

題。如果使用者很多的話，4 把金鑰要如何管理呢？結果有些管理者不是只固定使用某些金鑰，再不然就是根本不使用 WEP。使用者的認證也是問題，只有靠 SSID 來辨識是不夠的。所以也才會有將來的 RADIUS Server 與 802.1x 的產生。

三、 背景知識

在上一章中，我們簡述了 802.11 天生無線的缺陷但還是有無線的優點，只有 WEP key 當初設計的缺陷一直無法彌補，所以才會有 802.1x 與 WPA 的發展。

3.1 802.1x

802.1x 是 IEEE 為了 802 網路家族所定的安全標準，它規範了以單個連線為單位的存取控制。不過我們現在把焦點集中在無線網路上，當 802.1x 與 EAP（Extensible Authentication Protocol）〔7〕合作的時候，它如何來保障無線網路上的存取安全。

3.2 EAP

EAP 是根基於 PPP（Point To Point Protocol）的一個擴充，因為無線網路的存取也可以被看成是點對點的存取，無線存取點對無線使用者，就好像是用 ADSL Modem 撥接上 Internet 一樣，ISP（Internet Service Provider）也必須對撥接上線的使用者做身份認證一樣。當 PPP 連線開始到認證成功，資料可以傳輸之前，必須要能先建立連線，然後在上面執行認證的協定，以便讓預備連線的雙方有時間來認證對方。等到認證成功之後，真正的連線才算建立完成。如果認證失敗的話，不只整個連線無法建立，甚至連原先用來認證的暫時連線還是會被斷線。這樣的機制看起來是非常適合無線網路的架構的。如果說，先讓無線使用者和無線存取點先建立暫時的連線，然後在連線上先執行認證的協定，先不討論如何認證的細節，等到認證成功再建立真正的連線，否則就把原來用來認證的連線斷線。這裡有三個角色在這樣的 EAP 連線中：

1.認證者（Authenticator），就是無線存取點，AP

2.認證伺服器（Authentication Server），RADIUS Server

3.被認證者（Supplicant），無線使用者（Wireless Client）

在 EAP 所支援的多種認證方法中，本系統選擇性的挑選 EAP-TLS（Transport Level Security）與 EAP-MD5（EAP-Message Digest 5）做為首要支援的對象。原因無他，在無線接取端的 Hostapd 有支援這兩項功能。

3.2.1 EAP Over LAN（EAPOL）

EAPOL 定義了 EAP 的封包在無線使用者端與無線存取點之間的無線 LAN 環境下封裝的規格。

EAPOL 封包的格式以簡表表示如下：

表 2、EAPOL 封包的格式

2 Bytes	1 Byte	1 Byte	2 Bytes	N Bytes
PAE Ethernet 種類	協定版本	封包種類	資料封包長度	資料封包

PAE Ethernet 種類：PAE(Port Access Entity) Ethernet Type，它佔了兩個 Byte 長，固定值是 88-8E（十六進位）。

協定版本：Protocol Version，它只佔了 1 個 Byte，這個值表示了傳送者所支援的 EAPOL 的版本號碼，目前它的值只有 01（十六進位）。

封包種類：Packet Type，它只佔了 1 個 Byte，這個值表示了傳送的這個封包的種類。封包種類共有以下五種：

1）EAP-Packet，其值為 00（十六進位），單純的 EAP 封包。

2）EAPOL-Start，其值為 01（十六進位），表示這個封包是啟始 EAP 協定的啟始封包。

3）EAPOL-Logoff，其值為 02（十六進位），表示這個封包是提出 Logoff 需求的封包。

4）EAPOL-Key，其值為 03（十六進位），表示這個封包是

EAPOL-Key 的封包。

5）EAPOL-Encapsulated-ASF-Alert，其值為 04（十六進位），表示這個封包，在資料封包的部份，包含了一個被 ASF（Alerting Standards Forum）所定義的 Alert 必須要送出。

資料封包長度：Packet Body Length，資料封包內容的長度，以 Byte 為單位。若其值為 0，表示沒有資料封包。

資料封包：Packet Body，資料封包內容。只有 EAP-Packet、EAPOL-Key 和 EAPOL-Encapsulated-ASF-Alert 這三種型式的封包會有資料封包這個部份。而且一次只能有單筆的資訊會存在，不允許複數筆的資料存在。例如一次只能允許一把 Key 的傳遞。另外，EAP 封包的最大長度取決於 MAC 層（Media Access Control）所能支援的長度。

3.2.2 EAP-TLS 與認證過程

EAP-TLS〔1〕，透過憑證的使用與 CA 的參與，提供了一個讓 Server 與 Client 來互相做認證的一個機制。只有當雙方的憑證同時都為合法的時候，連線才會被建立。

以下是實際的 EAP-TLS 的認證過程：

1. RADIUS Server 會送出 EAP 封包給無線使用者端，要求使用者端送出 ID。
2. 無線使用者端會回 ID。
3. Server 送出空的 EAP-TLS request 開始封包。
4. 無線使用者端送回 Client –Hello 的封包。
5. Server 送出包含 TLS 訊息的封包。例如
   (a)Server Hello
   (b)Client 端的憑證要求
   (c)Server finished
6. 無線使用者端送回包含 TLS 訊息的封包。例如
   (a)Client's Certificate.

(b)Pre-master cecret in key exchange message.

(c)Client Certificate verification information

(d)Change ciphter

(e)TLS finished

7.Server 送回其他 TLS 訊息的封包。例如

(a)Change Cipher

(b)TLS finished

8.Client 送回空的回傳封包

9.Server 送回 EAP Success 封包結束 EAP 交握過程

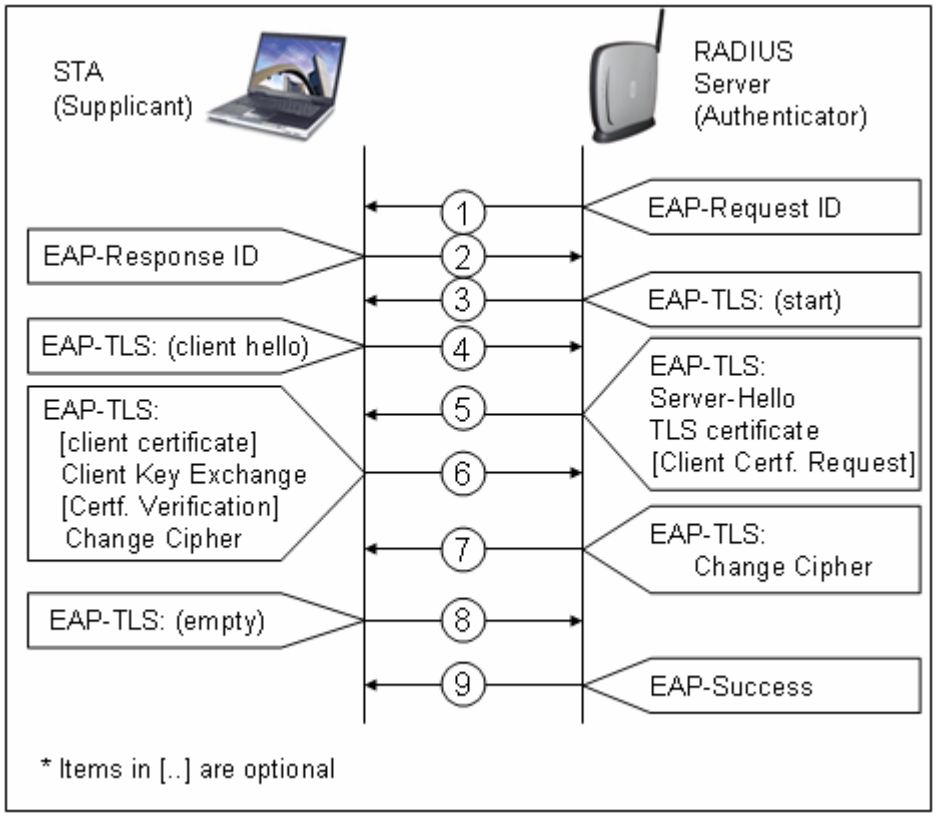

圖 6、EAP-TLS 憑證認證過程

資料來源：參考 RFC2716

3.2.3 **EAP-MD5（Message Digest 5）**

EAP-MD5 有一個最大的缺點，那就是它只讓 Authenticator 對

Supplicant 做認證，但是反過來，Supplicant 卻沒有對 Authenticator 認證。這對 WLAN 的安全來講，其實是一個漏洞。

EAP-MD5 雖然採用的是密碼檢驗的方式，可是它並不是在網路上傳輸密碼，它採用的是“證明”持有正確密碼的方式。

步驟如下：

1. Authenticator 送出一個 Challenge 給 Supplicant，這個 Challenge 包含了一個字串與流水號。
2. 而 Supplicant 要證明它持有密碼的方法就是對 Challenge 與密碼做 HASH，然後將 128bits 的 HASH 結果送回給 Authenticator。
3. Authenticator 在收到 Response 之後，也做相同的運算，當然在做運算之前，Authenticator 必須已知正確的 Supplicant 的密碼。
4. Authenticator 在算出相同得 HASH 結果之後，告知 Supplicant 通過了使用者的認證。

除了剛剛所講的缺點以外，它還有其他的缺點，例如：它必須要有地方來保存每一個需要認證的 Supplicant 的密碼，還有它並沒有產生 WEP session key 以及讓 session key 可以動態更換（Rekey）的功能。

只是就一般的使用者操作而言，它可算是在不使用憑證下的一個替代方案。

3.3 WPA（Wi-Fi Protected Access）

IEEE 制定了 802.11i〔5〕的規格，WPA〔5〕算是其中的一個子集合。802.1x 的推出主要是符合中大型企業的架構，所以一般大眾只好退而求其次的使用 WEP，可是一般當啟用 WEP 之後，傳輸量馬上降低，在 WEP 聲名已相當不好的情況下，反而有些使用者乾脆選擇不使用 WEP。所以 Wi-Fi 小組才想說早點推出 WPA 規格，讓一些硬體及軟體廠

商能及早開發出符合 WPA 認證的產品。

WPA 以 802.1x 和 EAP 作為其認證機制的基礎。在大型企業的網路中，認證程序必須要有使用者資料庫以及身份檢查的部分，通常是由 RADIUS 伺服器來完成。但由於 WPA 是打算讓所有的 WLAN 使用者都能使用，於是它也提供了一個比較簡單、不需要新增 RADIUS 伺服器就能使用的模式。這個模式－稱為預先共用金鑰（WPA-PSK，Pre-Share Key）模式，我將在以下的第一項解釋。

Wi-Fi 這個認證單位同時規定，2003 年 8 月開始送測的產品必須擁有 WPA 的功能，可是不在相容性測試項目中。但是，2004 年 4 月開始的 WPA 認證，就必須要通過完整的 WPA 相容性測試項目。

3.3.1 預先共用金鑰（WPA-PSK，PreShare Key）

預先共用金鑰，它是一組 8 到 63 個字元長度的字串。使用者只要在使用 WPA-PSK 的無線 LAN 環境下，將這支預先共用金鑰輸入到各個 WLAN 的節點，無線的用戶端就可以擁有 WLAN 的存取權限。

3.3.2 暫時金鑰完整性協定（WPA-TKIP）

暫時金鑰完整性協定，TKIP（Temporal Key Integrity Protocol），它可以算是下一代的 WEP，它使用新的加密演算法，如果使用的無線裝置有支援硬體加密的話，它也可以使用硬體加密來做運算。

TKIP 也提供下列項目：

1.決定加密金鑰後，驗證安全性設定。

2.同步處理每個框架的單一傳播加密金鑰變更。

3.決定每個預先共用金鑰驗證的唯一開始單一傳播加密金鑰。

3.3.3 進階加密標準（WPA-AES）

進階加密標準，AES（Advanced Encryption Standard），它是 WPA

所使用的另外一個加密方式。

各個不同的無線使用者端是可以各自使用不同的加密方法，比如說 AP 與 User A 使用 TKIP，與 User B 使用 AES，這是可以的。但是在原來的 802.11 規格與 802.1x 的規格中並沒有允許這樣的用法。

3.4 電子憑證（Certificate）

這一節我們來談在 802.1x 中佔了很大比重的電子憑證的部分。透過電子憑證，可以讓 RADIUS 伺服器與無線的使用者互相驗證身份，然後進而互相傳送必要的資料來進行加密的通訊。

3.4.1 X.509 Certification

X.509 Certification 最初被 CCITT 制定於 1988 年，那時稱為 v1。1933 年修訂了 v2，增加了兩個項目。後來由於 v1，v2 兩版的缺陷，1996 年由 ISO/IEC/ITU 和 ANSI X9 改進為 v3 版，然後就一直用到現在。X.509（Public-Key Infrastructure）不只是對電子憑證制定規格，它也對整個公開金鑰的安全架構，運作方式都有規範，舉凡憑證認證方式，憑證格式內容，CA 的階層架構，以及數位簽章所採用的演算法都有標準。

3.4.2 電子憑證格式 PKCS

PKCS（Public-Key Cryptography Standards），是 RSA 這家公司為了推廣公開金鑰的使用，於 1991 年訂立了第一版。目前 PKCS 中定義了 12 種不同種類的格式，原來的 PKCS #2 以及 PKCS #4，已經被整合到 PKCS #1（RSA Cryptography Standard）的格式中了。回想當使用 EAP-TLS 認證的時候，使用者必須擁有兩個憑證，一個是 RootCA 的憑證，另一個是 CA 簽給使用者的憑證，兩個憑證的格式都是使用 PKCS #12(Personal Information Exchange Syntax Standard)。只是 RootCA 的憑證並沒有包含私密金鑰的部份，只包含了 RootCA 的公開金鑰以及必要

的資訊，例如金鑰長度(512Bits 或 1024Bits)，Common Name(RootCA 的名字），憑證的起始日期及有效日期。CA 簽給使用者的憑證則包含了公開金鑰以及使用密碼加密的私密金鑰，一樣也有憑證的起始日期及有效日期。

四、 系統設計與實作

原來的系統只支援 RADIUS Client 端的功能，它必須將無線 Client 端的認證需求轉送至外部的 RADIUS Server，使用者必須另外架設 RADIUS Server，實在不便。雖然使用者可能想以 Internet 上的 RADIUS Server 來節省成本，可是別忘了當 Internet 連線斷了之後無線使用者就無法通過憑證認證了。然而筆者的設計對於這個問題是可以避免的，因為 RADIUS Server 與 RADIUS Client 都執行在同一設備上，可以看成是一封閉系統。對於新增的 CA 與 RADIUS Server 的功能，馬上就面臨到 DRAM 與 FLASH 空間可能不夠的問題。原來的 FLASH 存放的有 Boot Loader(開機程式)、使用者的設定資料、以及整個系統壓縮過的程式碼，幾乎已完全佔滿了 4M Bytes 的空間。DRAM 大小雖然有 32M Bytes，可是在原有的系統上也已非常接近滿載。

系統的 Kernel 選用的是 Embedded Linux，不只因為 Linux 是 Free 的，而且是可以與 PC 上的 Linux 相容的。這樣做有幾點好處，那就是可以事先排除硬體問題的干擾，再來就是各別模組可以分別開發，最後再整合在一起，可以節省不少時間。還有全部的程式幾乎都可以找到開放原始碼，不只問題少，就算發現問題也可以很快解決。然後我試著將 RADIUS Server 與 CA 先在原來的 PC 平台上模擬實作，實作成功後再轉到硬體平台上。在記憶體空間已明顯不足的情況下，最後，只有將 FLASH 與 DRAM 各增加一倍的空間，也就是 8M Bytes 的 FLASH 與 64M Bytes 的 DRAM。

4.1 軟體系統架構

因為使用了 IDT 的 79RC32438 這顆整合通訊處理器，當然也必須使用 IDT 所提供的 Embedded Linux Kernel，它的版本為 2.4.18，其他附屬

的程式集有 BusyBox version 0.60.5，Cross-compile 是 MIPS 系列 CPU 所使用的 mipstools。在 PC 上可以找到 RedHat 7.3 也使用相同的 Linux Kernel 版本，於是可以先在 PC 上模擬，模擬成功後再轉移到硬體平台上。在使用平台的轉換過程中，最大的不同是 Compiler 的差異，除了某些與系統比較有直接相關的模組需要特別改動以外，其他部份，大抵上是沒有什麼問題的。

以簡圖來表示：

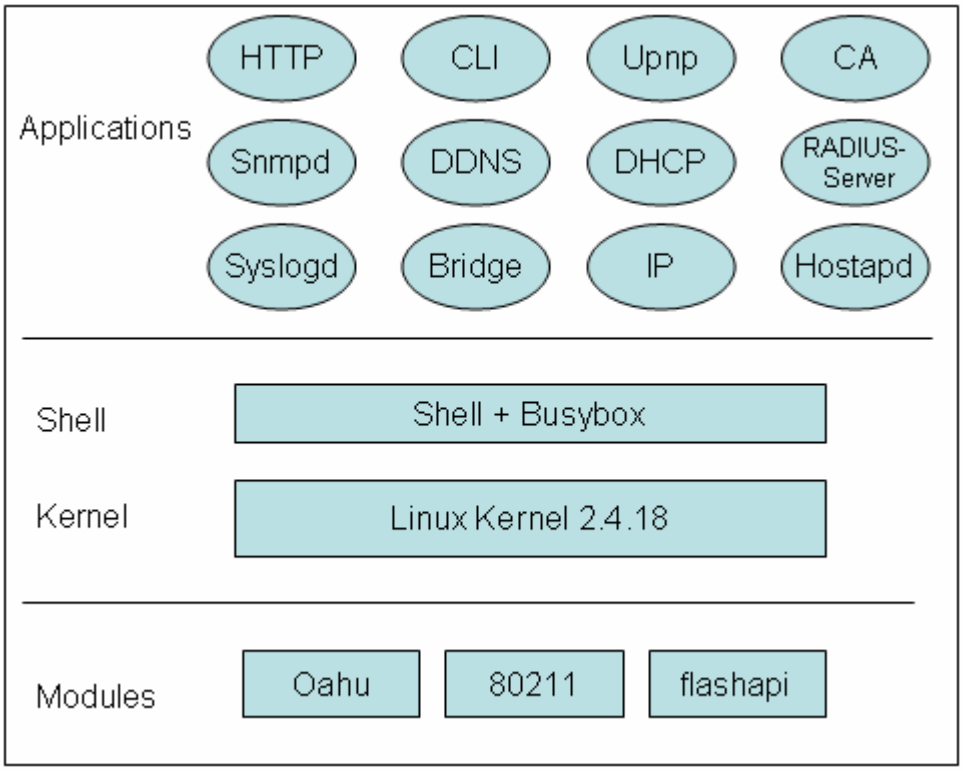

圖 7、軟體系統架構

基本上，與硬體有關的會是在 Kernel 與 Modules 這一層。然而大部份的修改是與硬體無關的，並且會發生在 Application 這一層。圖上並未列出所有的 Applications，只列出與本文有關的部分。注意圖上右上角 Applications 中的 RADIUS Server 與 CA 是我們接下來要探討的部份。

4.2 GUI 的架構

其實設計一台無線路由器，還有一個相當重要的部份，那就是如何

提供方便好用的 GUI 來讓使用者設定。早期的路由器都是透過命令列的方式來設定的，通常是系統管理者，他必須規劃網路環境，也必須對艱深的命令列語法相當熟悉才能架輕就熟。

那一般家庭使用者呢？ 誰會非常熟悉命令列語法？

我選擇的是使用 WEB GUI 加上 CGI 的方式來設定路由器。

使用者透過 WEB Browser 的方式，經由 HTTP〔9〕，瀏覽內建的網頁，不只可以更系統化的了解目前無線路由器的設定值，同時也可以很方便的修改它。WEB Browser 在各種平台都有，只要能夠與無線路由器連線，都可以被用來設定它。

讓我以簡圖來表示 GUI 的 Data Flow：

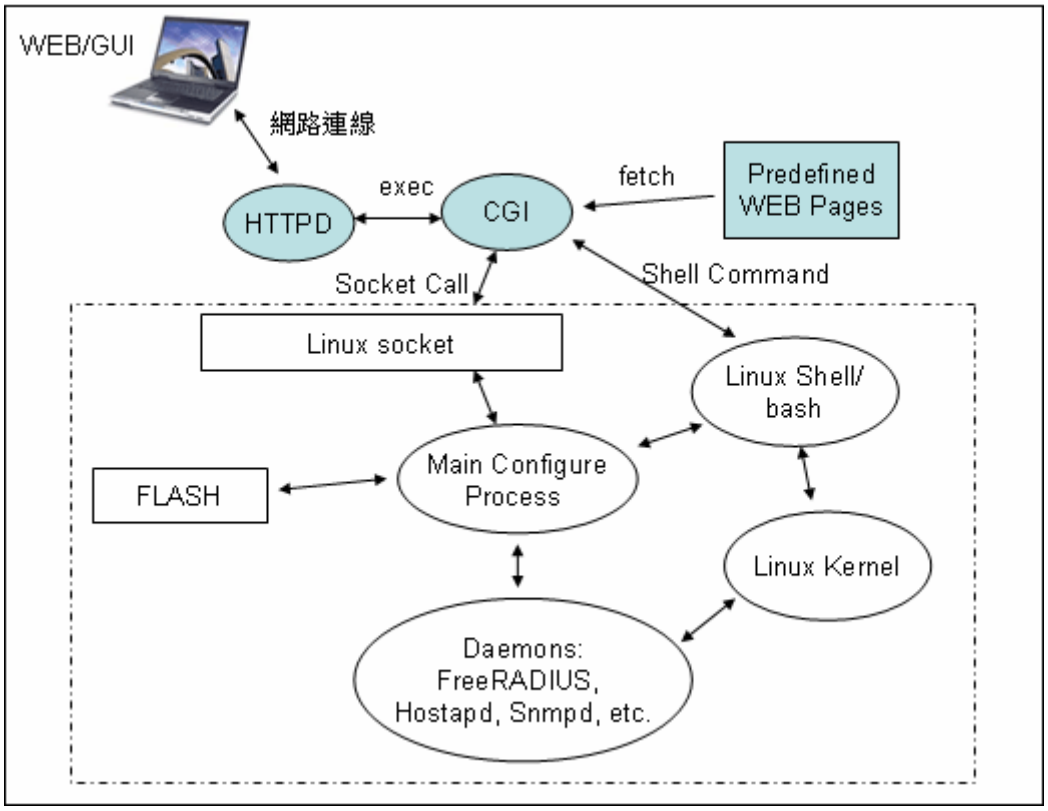

圖 8、GUI Configuration Structure

系統透過 HTTPD 這個 Daemon 來接受使用者傳送過來的 HTTP 需求，也透過 HTTPD 來回傳網頁。舉個例子來說：

例如使用者想看 RADIUS Server 設定的網頁。它的流程是：

1. WEB Browser 送出 Get RADIUS page 的 HTTP

Command。

2. HTTPD 收到了之後，將 HTTP Command 轉送給 CGI 程式並執行。
3. CGI 程式判斷要讀取 RADIUS Server 的相關設定。
4. CGI 程式透過 socket call 的方式向 Main Configure Process 要求目前的設定值。執行過程中的設定值都由 Main Configure Process 保持。
5. CGI 得到 Main Configure Process 的回傳值後，繼續讀取 RADIUS Server 的網頁，並且與回傳值結合，成為反應目前設定值的 WEB Page。
6. CGI 將新的 WEB Page 回傳給 HTTPD。
7. HTTPD 再將 WEB Page 透過有線/無線網路，回傳給 WEB Browser。
8. WEB Browser 再將 WEB Page 顯示給使用者看。

又例如使用者想修改 RADIUS Server 網頁上的設定值。它的流程會是：

1. 使用者在網頁上的可設定欄位變更設定值，然後點選“設定”按鈕。
2. WEB Browser 送出“Post setup.cgi”的 HTTP Command 給系統上的 HTTPD。
3. HTTPD 收到了之後，執行 CGI 程式，並且將 HTTP Command 轉送給 CGI 程式。
4. CGI 程式判斷要修改 RADIUS Server 的相關設定。
5. CGI 程式透過 socket call 的方式向 Main Configure Process 要求變更設定值。
6. Main Configure Process 判斷必須對 Hostapd 與

FreeRADIUS 變更設定。

7. 變更設定後重新執行 Hostapd 與 FreeRADIUS。
8. Main Configure Process 將變更後的值寫回到 FLASH。
9. 正常執行的話，把正確值回傳給 CGI。
10. CGI 得到 Main Configure Process 的回傳值後，繼續讀取 RADIUS Server 的網頁，並且與回傳值結合，成為反應目前設定值的 WEB Page。
11. CGI 將新的 WEB Page 回傳給 HTTPD。
12. HTTPD 再將 WEB Page 透過 Ethernet 封包，回傳給 WEB Browser。
13. WEB Browser 再將 WEB Page 顯示給使用者看。

以上是兩種主要的讀取與設定系統值的簡要操作過程。這兩種主要過程在讀取網頁的部分幾乎是雷同的。於是，我就將它們兩者合併設計，如下圖：

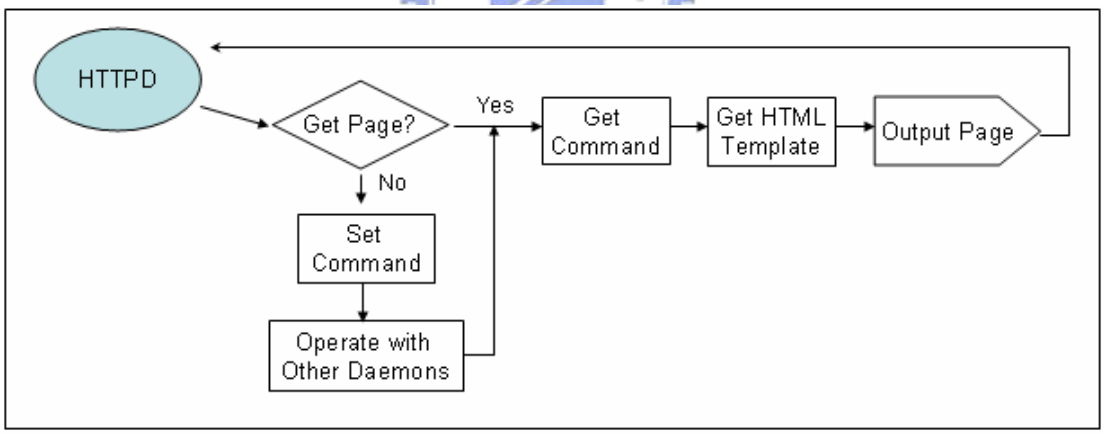

圖 9、CGI/Page Data Flow

雖然每一個網頁的功能各不相同，但是它們的設定/顯示的流程卻是相同的。所以最重要的資料流貫通之後，其他的網頁就可以比照辦理。

4.3 RADIUS Client

RADIUS 是 Remote Authentication Dial-In User Service 的縮寫，由字面上來看，它就是要對 Remote User 做認證，再加上使用在 Wireless

Network 上，無線網路安全更有保障。

4.3.1 RADIUS Client 的原始設定

原來的系統本來就有支援 RADIUS 的認證，只是原來的系統只做為 RADIUS 認證的其中一個點，我們稱為 RADIUS Client，另外兩個認證角色是前端的 Wireless Client 與後端的 RADIUS Server。在圖八的軟體系統架構中的 Hostapd 就是做這個 RADIUS Client 的角色。

Hostapd 的設定檔在屬於 RADIUS Authenticate Forward 部份的設定有：

```
#1: ieee8021x=1
#2: own_ip_addr=192.168.1.2

#3: auth_server_addr=127.0.0.1
#4: auth_server_port=1812
#5: auth_server_shared_secret=secret

#6: auth_server_addr=1.2.3.4
#7: auth_server_port=1812
#8: auth_server_shared_secret=abcdef
#9: auth_server_addr=2.2.3.4
#10: auth_server_port=1812
#11: auth_server_shared_secret=ghijkl
#12: radius_failover_limit=3
#13: radius_retry_primary_interval=3600
```

圖 10、Hostapd 的設定檔(部份)

因為這個 Hostapd 的設定檔是以 text 方式存在，所以它也相當容易修改與擴充。比如第 3 行到第 5 行的參數設定，它的意思就是在讓 Hostapd 知道 RADIUS Server 所在的 IP Address 與溝通的 Port 以及使用的 Share Secret。如果要讓 RADIUS Server 執行在相同的機器上，那這裡的參數必須要修改。

4.3.2 RADIUS Client 的修改設定

實作上，我使用 Loop back interface 的 IP Address（127.0.0.1），搭配修改過的網頁讓 Embedded RADIUS Server 實現。然後又因為實作

上讓 RADIUS Server 與 RADIUS Client (Hostapd) 在同一系統上，所以 Server Port 與 Share Secret 是可以被預先設定成固定值的。如圖 10 的第 4 行到第 5 行所示。

4.4 RADIUS Server

RADIUS Server 負責驗證無線使用者的憑證，以及驗證完後，WEP Session Key 的產生與傳送。目前規劃使用到的功能只有 EAP-MD5 以及 EAP-TLS 這兩樣功能。

目前找到 FreeRADIUS 支援 802.1x，而且也支援的最完整，許多大型的開發計畫也都拿 FreeRADIUS 來做測試。於是我決定使用它做為新系統內的 RADIUS Server。

決定使用 FreeRADIUS 之後，接下來的問題是如何整合到系統之中。我的計劃是這樣：

1. 因為使用的是 Linux 平台，所以先在 PC 環境上安裝與除錯，並且驗證實作的可能性。這個時候就要記錄使用的檔案結構以及被連結到的程式庫，做為移植的參考。
2. 如果成功的話，就按照在 PC 上模擬的結果，修改 Makefile 然後編譯出可以在系統上執行的 RADIUS Server。
3. 如果這個部份也成功的話，最後再修改系統上的開機順序檔，將 RADIUS Server 的啟動時間點安插在啟動過程中最適當的位置。
4. 交錯驗證 PC 版本以及 Embedded 版本的執行情況，做為除錯的參考。

計劃示意圖如下：

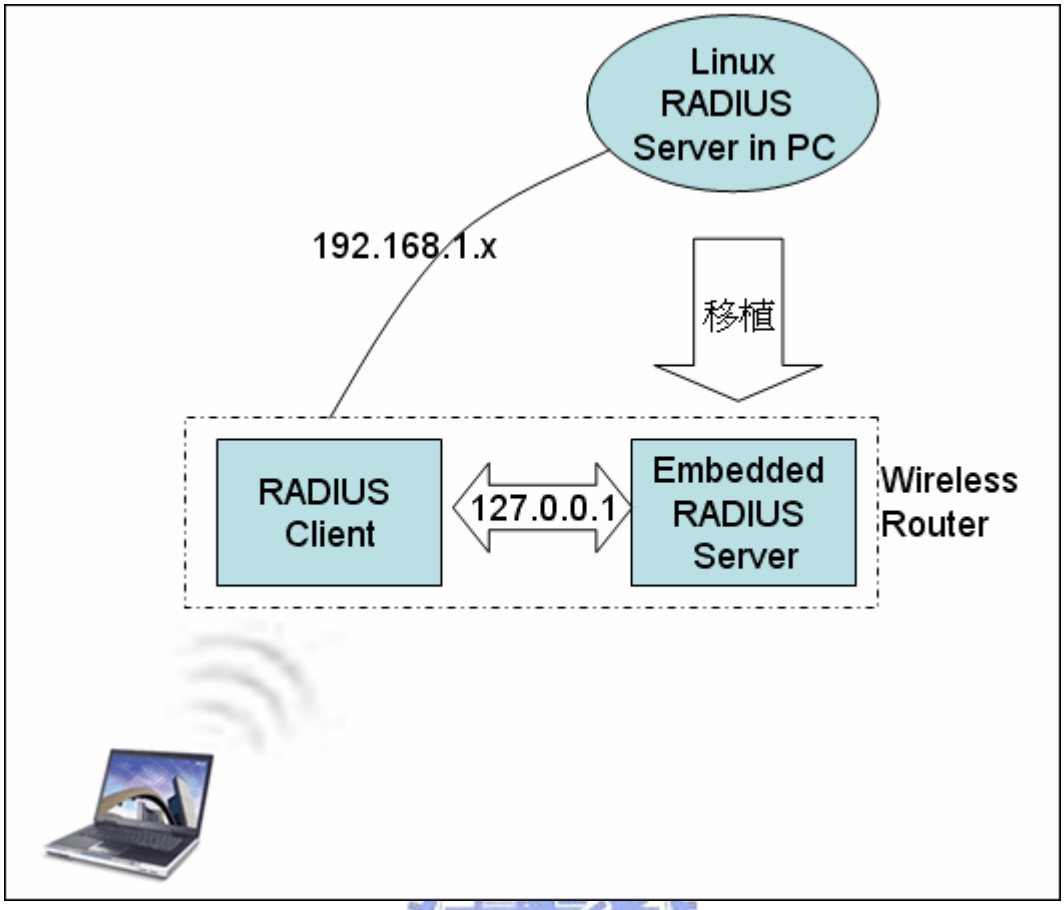

圖 11、RADIUS Server 移植計劃

實作過程還算順利，也因為可以先在 PC 上模擬及除錯，應該節省了不少開發時間。

在編譯 FreeRADIUS 的過程當中，發現到它需要兩個 Extend Library 的支援，那就是 Openssl 當中的 libcrypto.a 與 libssl.a 這兩個檔案。等於宣告要移植 FreeRADIUS 的同時也必須移植 Openssl 的程式庫。不過幸好 CA 的部份也決定使用 Openssl，剛好也可以節省部份的程式空間。

4.5 憑證授權單位(Certificate Authority)

憑證授權 Certificate Authority（簡稱為 CA）。一個 CA，基本上應該要有： a、產生憑證 b、驗證憑證 c、憑證管理的基本功能。最後我認為 Openssl 應該可以符合需求，至少在 a、b 兩項。至於第三項的憑證管理功能，可以再區分出有效憑證的管理、失效憑證的管理、延長憑證的

有效期限等等。我認為這些功能在使用者的使用需求上比較沒有那麼必要，通常使用者只要有能簽出憑證的功能。所以在一開始使用了比較長的有效期限。至於如何做好憑證的管理，是必須要花比較長的時間來規劃，例如在一台獨立的無線路由器上，FLASH 的空間已然不足的情況下，是不可能無限制的保留已簽發過的憑證，更何況已超過有效期限的失效憑證表列。反倒是延長憑證的有效期限是一個比較可行的輔助功能。還有一項就是如何在憑證的有效期限快到達之前，如何讓使用者與系統管理者都可接收到相關的訊息，然後來做延長憑證的有效期限，或者換發新的憑證，都是一些可行的方案。至少在預設的憑證有效期限到來之前，可以好好來規劃一些新的功能。我先把這部分的設計列為未來的待做事項。

基本上，CA 的種類有兩種。一種是 Signed CA，另一種是 Self-signed CA。這兩種 CA 的差異在於自己所使用的憑證是否需要其他 CA 的認證。所以，Signed CA 所使用的憑證是被別的 CA 所認證；而認證 signed CA 的 CA 我們稱為上層 CA。通常這個上層 CA 都會找比較有名的 CA，例如 VeriSign（www.verisign.com）。

另外，Self-signed CA 的憑證就是由自己替自己簽發的憑證，也就是由自己所認證。於是 Self-signed CA 沒有上一層的 CA，則自己就是 Root CA。現在在系統中的 CA，就是所謂的 Root CA。它必須自己替自己簽發憑證。

接著，我們來看在實作系統上的各種不同憑證的產生方式：

4.5.1 產生給 CA 自己的憑證

因為這是一個 Root CA 憑證的產生方式，而且它完全在無線路由器的系統上執行，執行的時機有兩種：一種是第一次起動 Embedded CA 時，這個第一次起動不是每次開關機後的第一次起動，而是當系統設定值完全沒有存在任何憑證的時候。而且只要這一次開機後，產生的憑證

就會存在系統設定值當中，在以後的開機過程中，經過讀取系統設定值，發覺已經有RootCA的憑證存在的情況下，就不再重新產生給自己的憑證。另一種是回復系統預設值之後，接下來的第一次開機時。如何回復系統預設值有兩種操作方式，第一種是從WEB GUI上的回復預設值頁面操作，第二種是操作硬體的RESET按鈕。雖然兩次重新起動時的預設值皆相同，但是因為其中有一組 Random 值，在不同的兩次啟動時是不會相同的，因此兩次所產生的憑證也會不相同。所以從原來RootCA簽發出去的使用者憑證與 RADIUS Server 的憑證將會無法用新產生的RootCA 的憑證來驗證成功。必須要將舊的使用者憑證換成由新的 Root CA 簽發出來的使用者憑證才可以通過驗證。這個問題，目前我把它歸類為設計規格的限制中。

產生 Root CA 憑證時的次序是:

1. 先利用 Openssl 產生金鑰對，密碼是預設值，Common Name 是用"RootCA"與 MAC Address 所組合而成，例如："RootCA_00:00:00:11:22:33"。
2. 然後利用產生的RootCA的私密金鑰來簽出自己的憑證。
3. 將金鑰對與 RootCA 的憑證檔案儲存為系統設定值，當下次開機時再自動讀取進來。

因為一般使用者是藉由Windows來管理 Root CA所使用的憑證，而Windows所顯示的就是Common Name，藉由不同的MAC Address產生不同的Common Name，以避免互相干擾。

下圖列出已經輸入在使用者電腦上的RootCA的憑證範例：

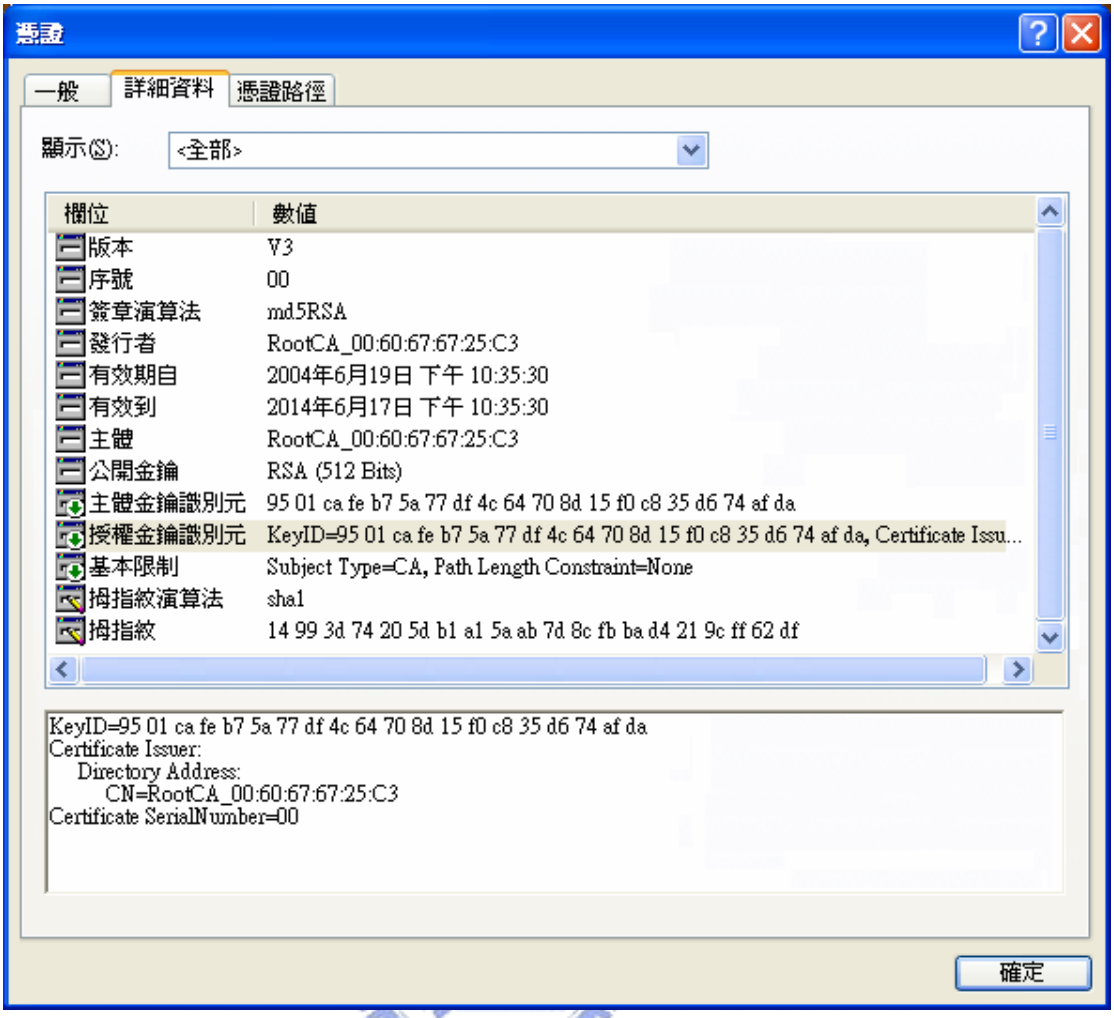

圖 12、使用者輸入的 RootCA 憑證範例

4.5.2 產生給 RADIUS Server 的憑證

當 Root CA 簽發完自己的憑證之後，它就可以簽發憑證給在相同機器上的 RADIUS Server，也可以簽發憑證給不同機器上的 RADIUS Server。給自己機器上的 RADIUS Server，只要將檔案存在的路徑傳給 RADIUS Server 即可。但是簽發憑證給不同機器上的 RADIUS Server，必須要由使用者透過手動將 RootCA 的憑證與 RADIUS Server 的憑證，透過檔案以及 WEB GUI 的方式輸入。

以下是 Root CA 對兩種不同 RADIUS Server 的憑證的產生方式：

第一種是簽發憑證給自己機器上的 RADIUS Server，它是在 Root CA 產生完自己的憑證之後自動產生的，系統使用預設

的”username”以及預設的”password”來產生，產生完的憑證再自動儲存在內部 RADIUS Server 可以存取的路徑上，然後修改 RADIUS Server 的 Configure file，對其發出重新啟動的訊號，等到 RADIUS Server 重新啟動完成之後，RADIUS Server 就會自動讀取 RootCA 以及自己的憑證，在有需要使用到 802.1x 傳輸憑證與驗證憑證的時候。

第二種是不同機器上的 RADIUS Server，通常是由系統管理者從 WEB GUI 輸入外部 RADIUS server 的 ”Username” 及”Password”來產生，然後將產生出來的憑證透過 WEB GUI 的輸出功能存成檔案，再拿給外部 RADIUS server 做輸入的動作。這個透過 WEB GUI 的輸出動作，可以再區分為幾種可能：

2a.在單獨一對一的連線下進行

2b.在路由器的 LAN 端進行

2c.透過 WLAN 進行

2d.透過 WAN 來進行傳輸

於是就存在不同的安全考量。

第一種方式是建議的運作方式，它幾乎沒有任何安全上的問題。還有第二種方式當中的 2a 與 2b 的連線，也是非常安全的，因為我們在 LAN 端使用的是 Switch 的 Chip。至於透過 2c 和 2d 的方式，我們並不建議使用。因為透過 WLAN 或透過 WAN 來傳輸 RootCA 所簽發的憑證，有外洩出去的可能，而這外洩的憑證有可能會被假冒的 RADIUS Server 或使用者拿去使用，造成安全上的問題。

不過，只要系統管理者能做好產出憑證的管理工作，事實上，並不會存在安全的問題。

4.5.3 產生給 Wireless Client 的憑證

目前新的系統只有一種方式來產生給 Wireless Client 端的憑證。它也是透過系統管理者從 WEB GUI 輸入 Wireless Client 端的"Username"及"Password"來產生，然後將產生出來的憑證存成檔案，再拿給外部 Wireless Client 端的機器做 Import 的動作。在傳輸憑證的安全考量上，與前一節所探討的是相同的問題。

4.6 各種操作模式的組合

由於原來的系統本身就已經有支援 802.1x 的 RADIUS Client 端的認證，只是沒有加入 Embedded RADIUS Server。現在我把 Embedded RADIUS Server 及 CA 加進來之後，操作情況變的有點複雜。以下將分別來探討各種組合：

4.6.1 內部 RADIUS Client+內部 RADIUS Server+內部 CA

這是建議的操作方式，也是本論文要達到的目的。如圖：

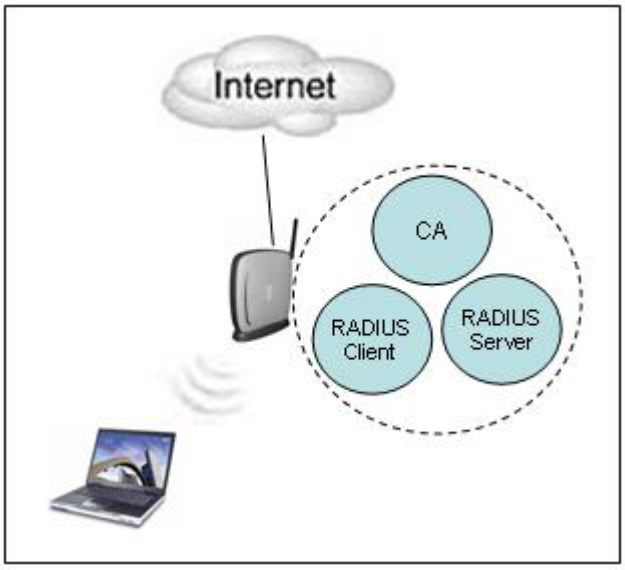

圖 13、建議的操作方式

在圖中間的無線網路路由器正在執行著 RADIUS Client、RADIUS

Server，至於Embedded CA，通常是Offline執行的。它的設定組態是將內部 CA 與內部 RADIUS Server 都設定在有使用的狀態。因為 CA 是Root CA，它已經對自己簽發憑證，同時它也對內部RADIUS Server 簽發憑證。至於無線的使用者端的憑證必須透過CA的簽發憑證機制將使用者端的憑證輸入至使用者端的機器內，還有另外一個 Root CA 的憑證也必須輸入至使用者端的機器內。

當使用者想讓他的網路使用 802.1x 的時候，如果他只能無線連接時，可能的做法是：

1. 先行以WEP Key方式連線，操作WEB GUI，透過"操作CA"的畫面，輸入使用者姓名與密碼，產生無線使用者端的憑證，並且連同 RootCA 的憑證，分別兩次的操作，將憑證先行安裝到作業系統內。
2. 透過WEB GUI，選擇改變無線網路路由器的WLAN連線的安全模式為 802.1x 的 EAP-TLS 認證，並且可自由選擇是否由RADIUS Server自動提供WEP session Key。
3. 啟動改變WLAN連線為802.1x，接著無線網路會斷線，因為已改變認證方式。
4. 使用者透過改變自己的WLAN Client軟體的連線Profile，將連線認證方式也改成 802.1x 的 EAP-TLS 認證，並且指定使用的使用者憑證與 RootCA 的憑證，別忘了輸入使用者名稱與密碼。
5. 重新啟動連線，過了幾秒鐘之後，當無線使用者端與RADIUS Server互相驗證完憑證之後，接著RADIUS Server會傳送將來使用的WEP session Key給無線使用者，然後無線使用者就用新的WEP session Key來正常上網了，如果網際網路的連線也一樣正常連線的話。

當使用者想讓他的網路使用 802.1x 的時候，而且他也能使用有線連

接時，設定情況會比較簡單，可能的做法如下：

1. 先透過有線連接上的 PC 或 Notebook，連上無線路由器，操作 WEB GUI，透過"操作 CA"的畫面，輸入使用者姓名與密碼，產生無線使用者端的憑證，並且連同 RootCA 的憑證，分別輸出到兩個檔案。
2. 繼續操作 WEB GUI，選擇改變 WLAN 連線的安全模式為 802.1x，並且可自由選擇是否由 RADIUS Server 自動提供 WEP session Key。
3. 啟動改變 WLAN 連線為 802.1x。注意，這時有線的 LAN 連線並不會斷線。
4. 將步驟一所產生的兩個憑證輸入到無線使用者端的機器。用各種方式傳給使用者端的機器，包括 e-mail、ftp、或者是磁片檔，我們可以不用擔心安全的問題，因為使用者的 Private Key 是被密碼保護著。
5. 使用者透過改變自己的 WLAN Client 軟體的連線 Profile，將使用者端的連線認證方式改成 802.1x 的 EAP-TLS 認證，並且指定使用的使用者憑證與 RootCA 的憑證，別忘了輸入使用者名稱與密碼。
6. 啟動無線使用者端的連線，過了幾秒鐘之後，當無線使用者端與 RADIUS Server 互相驗證完憑證之後，接著 RADIUS Server 會傳送將來使用的 WEP session Key 給無線使用者，然後無線使用者就用新的 WEP session Key 來正常上網了，如果網際網路的連線也一樣正常連線的話。

4.6.2 內部 RADIUS Client+內部 RADIUS Server+外部 CA

這個架構使用到一台外部的 CA，這個外部 CA 的可能性與連接性有

好幾種。如圖：

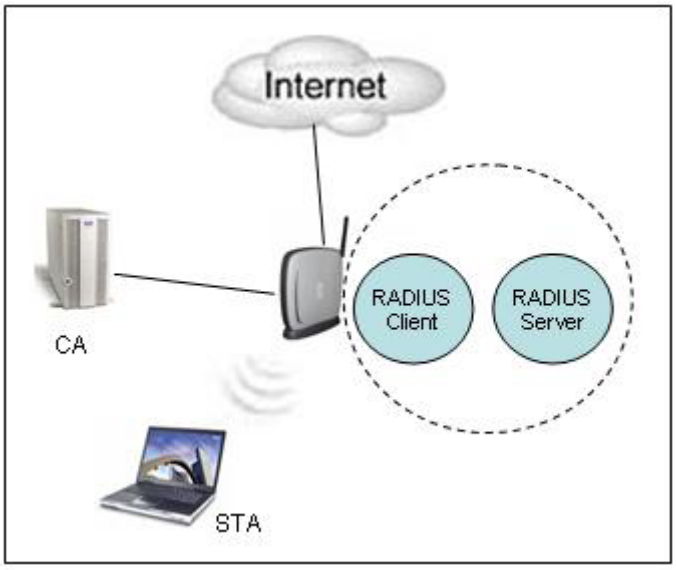

圖 14、結合外部 CA 的操作方式

在圖中間的 Device 中，只有執行 RADIUS Client 以及 RADIUS Server。CA 卻是在別的機器上執行，可能在 LAN，也可能在 WAN，有可能是 Windows 2000 server 的 IAS，也有可能是網際網路上的商業化的 CA。

它的設定組態是將內部 CA 設定為停止的狀態，但是內部的 RADIUS Server 是設定在執行的狀態。

在這裡，CA 有無及時連線並不重要，重要的是我們是否有外部 CA 所簽發的使用者憑證以及 CA 的憑證，還有外部 CA 所簽發給 RADIUS Server 的憑證。以這種方式，可能成本必須增加，操作與維護的成本也相對提高。

當使用者想讓他的網路使用如上圖的架構時，最重要的一點是，記得透過 WEB GUI 將內部 CA 的功能關閉，然後在 RADIUS Server 的選項當中，會出現兩個檔名及路徑的空白欄位讓使用者輸入外部 CA 所簽發的 RADIUS Server 的憑證以及 CA 的憑證。

接下來的做法，分別是設定無線路由器端的 802.1x 認證方式，以及無線使用者端的 802.1x，EAP-TLS 的認證方式。操作步驟如前述 4.6.1 操作方式的第一項所描述的。

4.6.3 內部 RADIUS Client+外部 RADIUS Server+外部 CA

這樣的架構，使用者將會多花兩台電腦的建制費用，其中一台是 CA，一台是 RADIUS server，當然使用者也可以只使用一台電腦，來達到 CA 及 RADIUS Server 的功能，例如 Windows 2000 Server 來達到。這個架構等同於原有系統所支援的 802.1x 的架構。如圖：

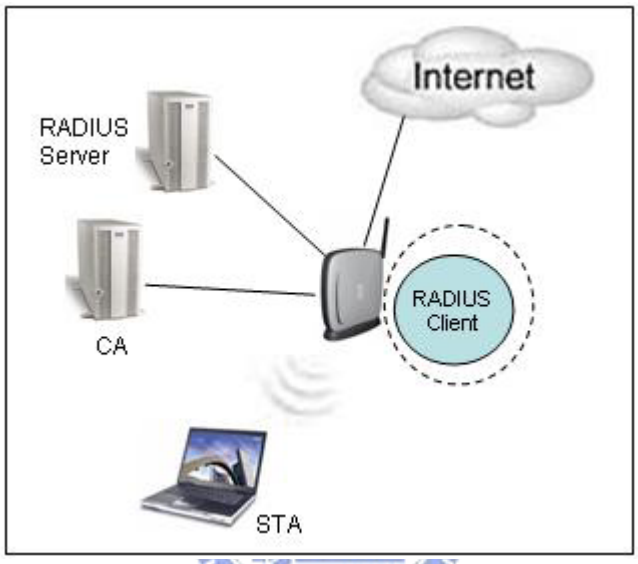

圖 15、結合外部 CA 與外部 RADIUS Server 的操作方式

在圖 15 中間的無線路由器中，只有執行 RADIUS Client 而已。在 WEB GUI 的設定中是將 Embedded CA 與 RADIUS Server 都設定在關閉的情形下，另外還必須將 RADIUS Client 所指向的 RADIUS Server 路徑由使用內部的 RADIUS Server 改成外部的 RADIUS Server。

在憑證的管理與設定上，外部的 RADIUS Server 必須要有外部 CA 所簽發的憑證以及外部 CA 本身的憑證。還有使用者端也必須事先要有外部 CA 所簽發給無線使用者的憑證和外部 CA 本身的憑證。還有，必須設定在 RADIUS Server 與 RADIUS Client 在溝通時所用到的共用密文（Share Secret）。

接下來的做法，分別是設定無線路由器端的 802.1x 認證方式，以及無線使用者端的 802.1x，EAP-TLS 的認證方式。操作步驟如前述 4.6.1

操作方式的第一項所描述的。

4.6.4 外部 RADIUS Client+內部 RADIUS Server+內部 CA

這樣的架構，不只節省了 CA 與 RADIUS Server 建置的費用，同時間更可以擴展無線網路的覆蓋範圍！如下圖：

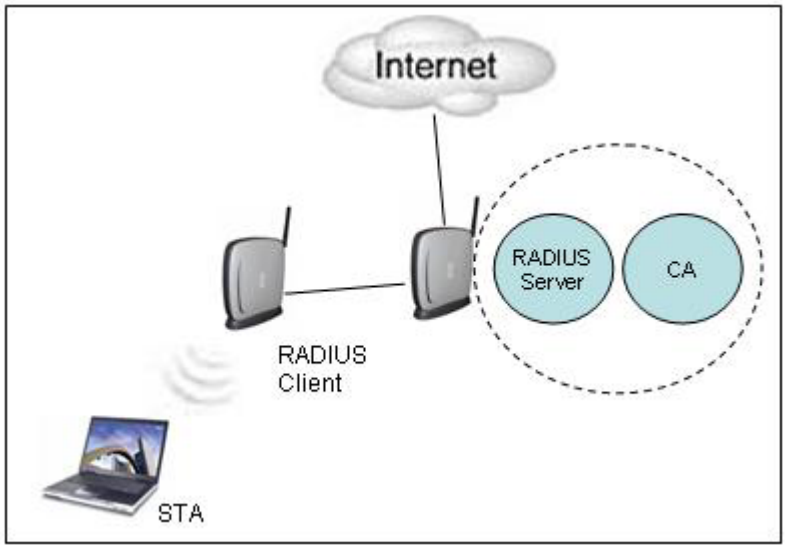

圖 16、兩台相同設備的搭配方式一

使用者可能因為空間比較大的關係，又或者因為某些原因，例如分級的管理，屬於 Group A 的使用設備甲，屬於 Group B 的使用設備乙。不管怎樣，我們的系統是有可能被整合起來使用的。就這個 Case 來說明，我們會希望以一個 RADIUS Server 來做為認證的中心，其他的無線存取點當做是 RADIUS Client。最好不要把所有的 RADIUS Server 以及 CA 都各自啟動執行，這樣混亂的架構對使用者來說是很不好的，因為可能使用者必須要自己管理好幾份的憑證。

在圖 16 中間的無線路由器中，正在執行著 RADIUS Server 與 CA，它一定也有執行著 RADIUS Client。不過我們所討論的 RADIUS Client 是在另一台設備上執行。

底下是角色設定的說明：

1. RADIUS Client，設定使用 802.1x 認證，將 RADIUS Server 的 Link 指向內部的 RADIUS Server，選定共用密文（Share

Secret）。

2. RADIUS Server，新加一項 RADIUS Client，選定共用密文（Share Secret），設定認證方式為 EAP-TLS。
3. Wireless Client，輸入 CA 所簽發的憑證和 CA 本身的憑證，使用 802.1x 的 EAP-TLS 連上 RADIUS Client。

4.6.5 外部 RADIUS Client+內部 RADIUS Server+外部 CA

與前一項架構很像，只是 CA 換成使用外部的 CA。如圖所示：

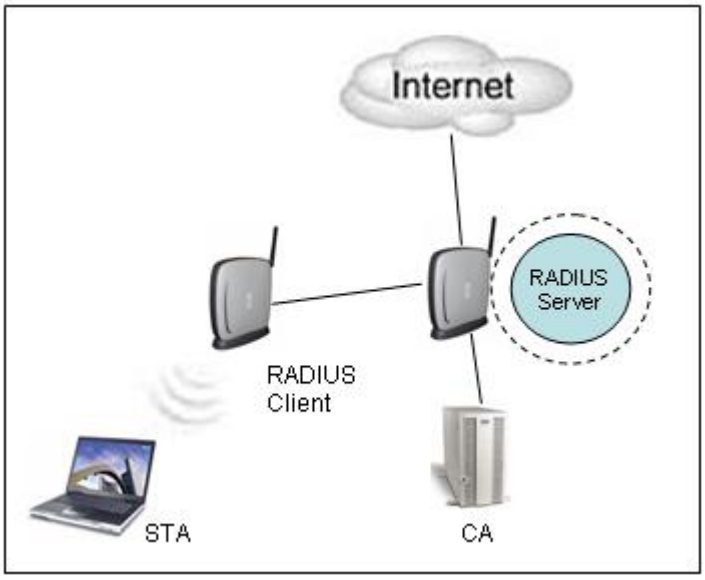

圖 17、兩台相同設備的搭配方式二

使用者可能因為管理上的理由，可能會採用這種方式，同時也方便憑證的管理。例如小公司使用 Windows 2000 Server 做為 CA 的話。

底下是角色設定的說明：

一、 RADIUS Client，設定使用 802.1x 認證，將 RADIUS Server 的 Link 指向另一台機器內的 RADIUS Server，選定共用密文（Share Secret）。

二、 RADIUS Server，新加一項外部 RADIUS Client 的連接設定，選定共用密文（Share Secret），設定認證方

式為 EAP-TLS。關閉與 RADIUS Server 同在一起的 CA，輸入外部 CA 所簽發給 RADIUS Server 的憑證和外部 CA 的憑證。

三、 Wireless Client，輸入外部 CA 所簽發的憑證和 CA 本身的憑證，使用 802.1x 的 EAP-TLS 連上 RADIUS Client。

總結前五項的設定，這是為了說明新的系統的規劃設計是可以符合使用者的各種不同的架構與應用。所以在相容性上，是沒有問題的。

五、實際操作

原先的規劃是在 802.1x 的架構下，實作 Embedded RADIUS Server 與 Embedded CA，然後提供 EAP-TLS 認證與 EAP-MD5 的認證。透過 WEB GUI 的操作與設定，讓使用者可以很方便的架設出一個安全的無線網路應用環境，前提是只“看得到”一台無線網路路由器，以及一台擁有 802.11a/11g 的無線網路卡的 Notebook。於是，使用者可以很方便的使用安全的 802.1x 的無線網路環境。

5.1 GUI

我設計的 GUI 是透過 WEB 瀏覽器，藉由瀏覽預先存在的網頁，並配合 CGI，達到與系統溝通的目的。以下將只針對提供無線連線與憑證認證相關的網頁做介紹。

5.1.1 CA 的管理

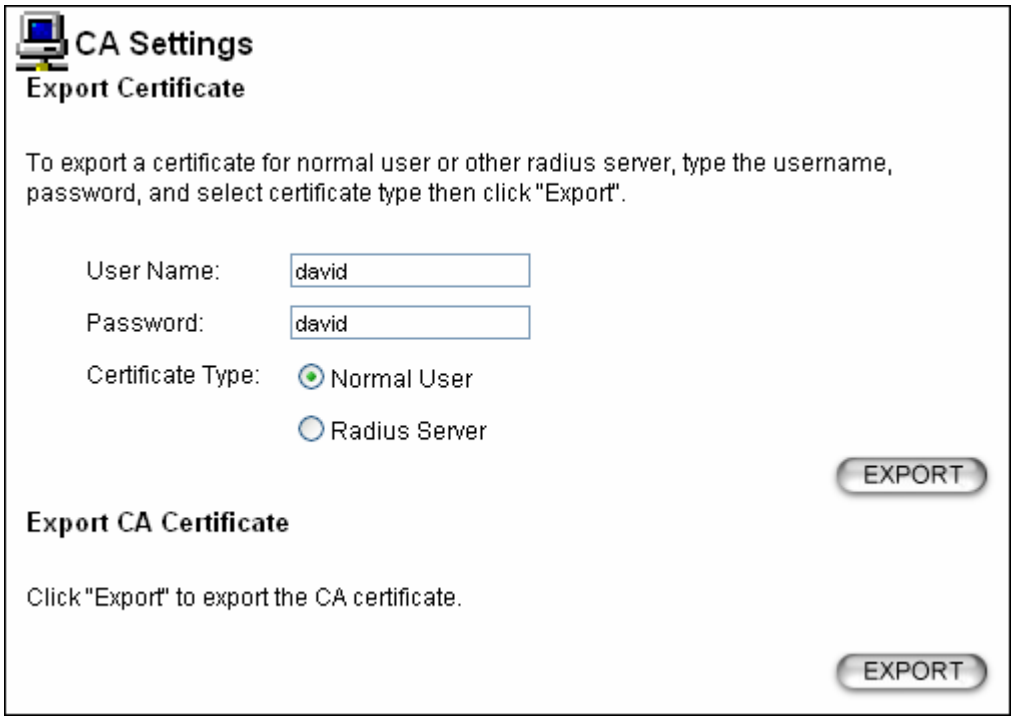

圖 18、CA 管理憑證的 Web GUI 畫面

如圖，對於 CA 的管理，其實相對比較簡單。因為只要能夠輸出 3 種憑證即可。

如果是給一般使用者的憑證，只要照著畫面上所顯示的，輸入使用者名稱(User Name），密碼（Password），以及選擇憑證的類別（Certificate Type）為一般使用者（Normal User），然後按輸出的按鈕（EXPORT），就會出現如下圖的對話盒，告訴你正要儲存使用者的憑證。注意，憑證的附檔名是 p12，檔案類型屬於個人資訊交換。

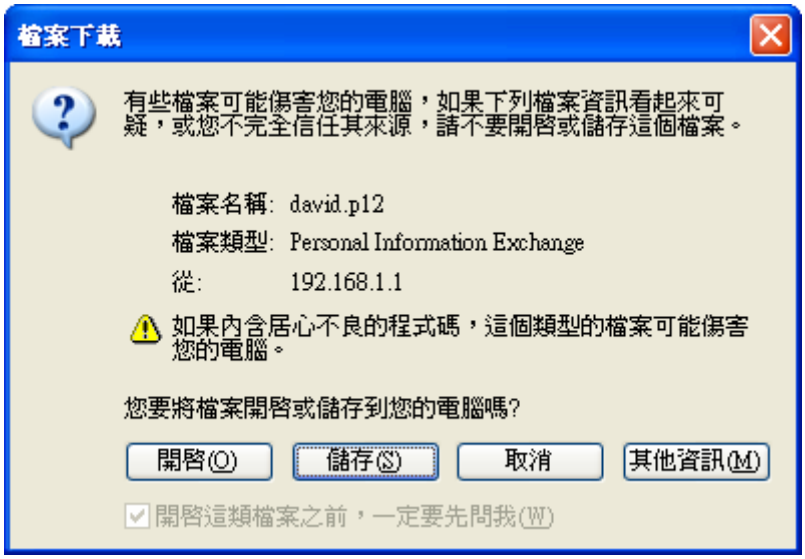

圖 19、輸出給使用者使用的憑證並儲存到電腦中

如果是給 RADIUS Server 的憑證，只要照著畫面上所顯示的，輸入使用者名稱（User Name），密碼（Password），以及選擇憑證的類別（Certificate Type)為 RADIUS Server，然後按輸出的按鈕(EXPORT)，就會出現如下圖的對話盒，告訴你正要儲存 RADIUS Server 的憑證。注意，憑證的附檔名是 pem。

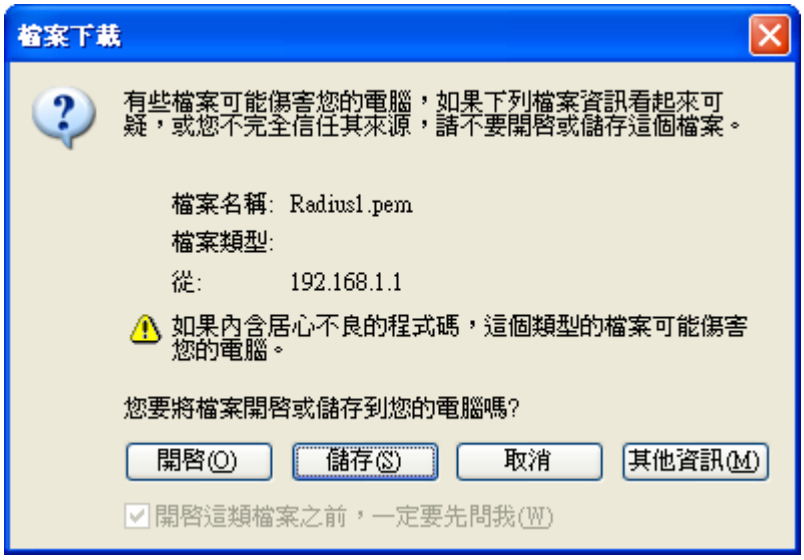

圖 20、輸出給 Radius Server 使用的憑證並儲存到電腦中

如果是輸出 Root CA 的憑證，只要照著畫面上所顯示的，直接點選輸出 RootCA 憑證的按鈕（EXPORT），也就是畫面中的第二顆按鈕，就會出現如下圖的對話盒，告訴你正要儲存 RootCA 的憑證。注意，憑證的附檔名是 cer，檔案類型是安全性憑證。

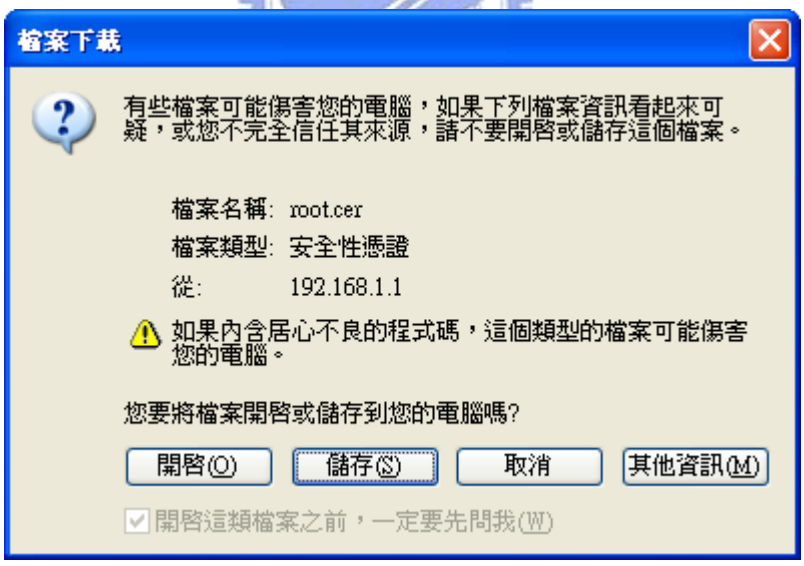

圖 21、輸出 RootCA 的憑證並儲存到電腦中

5.1.2 輸出憑證的管理

在前項 CA 的管理中，已知使用者可能輸出 3 種型式的憑證。通常使用者的操作是會先將憑證暫存在電腦硬碟的某個目錄中，但是這樣做的

話，作業系統其實還不知道有憑證的存在，一定要有“輸入”的動作，這樣作業系統才會對憑證有一個管理的動作。當使用者的程式有需要用到憑證的時候，系統就會根據管理憑證的目錄，去找到正確的憑證來使用。

舉 Windows XP 為例，使用者只要將滑鼠游標移至該“使用者憑證”的檔案，點擊滑鼠右鍵，在跳出的畫面上會出現有“安裝 PFX”的選項，點擊該選項後，根據“憑證匯入精靈”的指示操作即可。

又如果要匯入 RootCA 的憑證，點選滑鼠右鍵，在跳出的畫面上會出現有“安裝憑證”的選項，點擊該選項後，一樣根據“憑證匯入精靈”的指示操作即可。

如果要匯入RADIUS 伺服器的憑證，它是PEM 的檔案格式，目前只支援簽發憑證給相同系統的 RADIUS Server。基本上，並不考慮支援讓使用者透過我們的 RootCA 來更新 Windows 2000 Server 上的 RADIUS 伺服器的憑證。通常在這樣的假設條件下，使用者已經有 Windows 2000 Server 以及 CA 的功能，所以不建議使用者再透過我們內建的 RootCA 來簽發憑證。

5.1.3 RADIUS Client 的管理

RADIUS Client 的管理，如下圖所示：

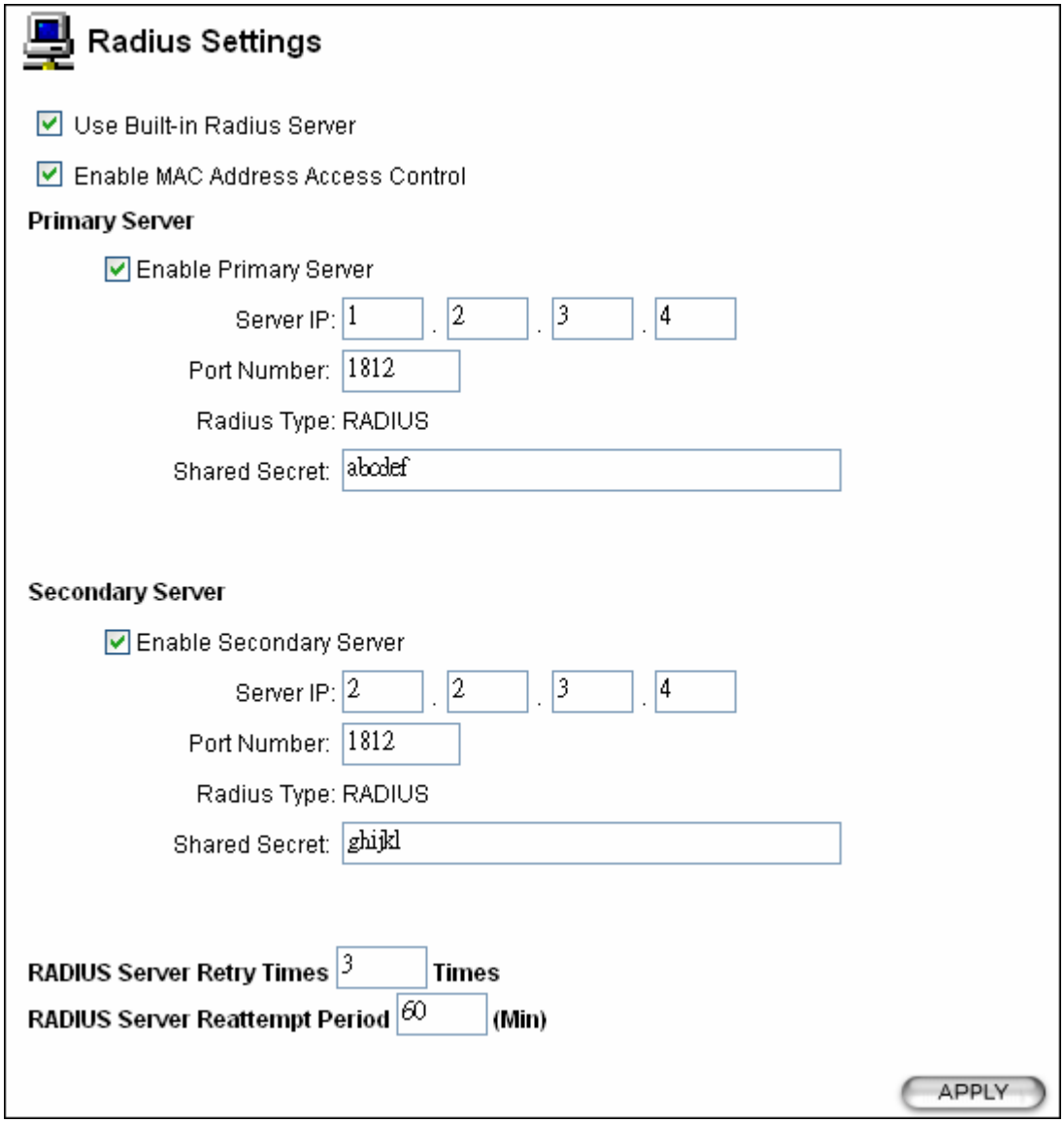

圖 22、RADIUS Client 的管理畫面

在這個 RADIUS Client 的網頁內，最上面有一個選項（Use Built-in RADIUS Server），它被用來指定是否要使用內建的 RADIUS Server。

如果使用者有打勾的話，則在 Hostapd 的設定檔當中就會指向三個 RADIUS Server。同時，如果使用者對無線網路安全使用 802.1x 的時候，RADIUS Client 尋找 RADIUS Sever 的優先順序會是“內建 RADIUS Server”，“Primary RADIUS Server”，最後才是“Secondary Server”。

如果使用者沒有打勾的話，則在 Hostapd 的設定檔當中就會指向兩個 RADIUS Server。對於尋找 RADIUS Sever 的優先順序會是“Primary RADIUS Server”，然後才是“Secondary Server”。

5.1.4 RADIUS Server 的管理-使用 TLS 認證及內建 CA

RADIUS Server 的管理就比較複雜。如下圖是使用內建的 CA 以及認證方式為 TLS 的情況：

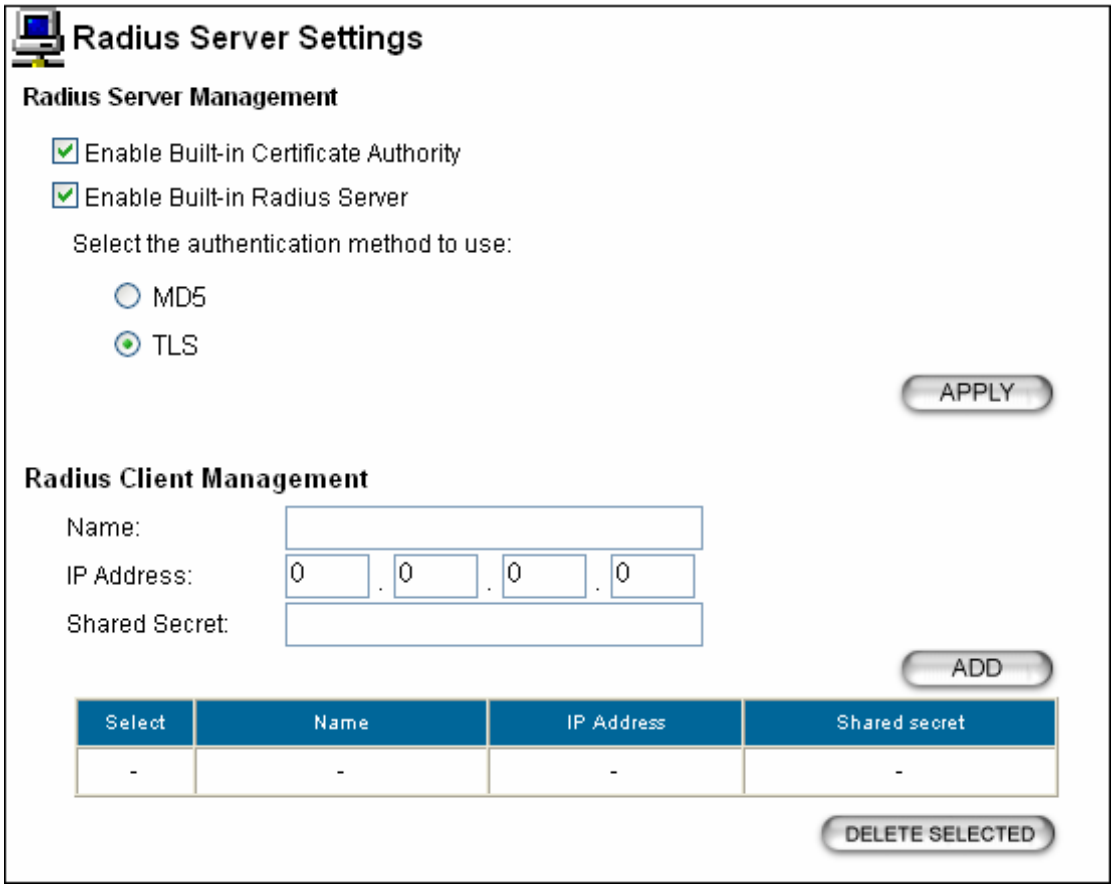

圖 23、RADIUS Server 的管理畫面-使用 TLS 認證及 CA

因為使用內建 CA，所以不需要從外部輸入憑證。還有，可能會有外部的 RADIUS Client 要求連線的情況，所以我設計提供了一個表格，方便 RADIUS Client 的管理。

5.1.5 RADIUS Server 的管理-使用 MD5 認證及內建 CA

如下圖是使用內建的 CA 以及認證方式為 MD5 的情況：

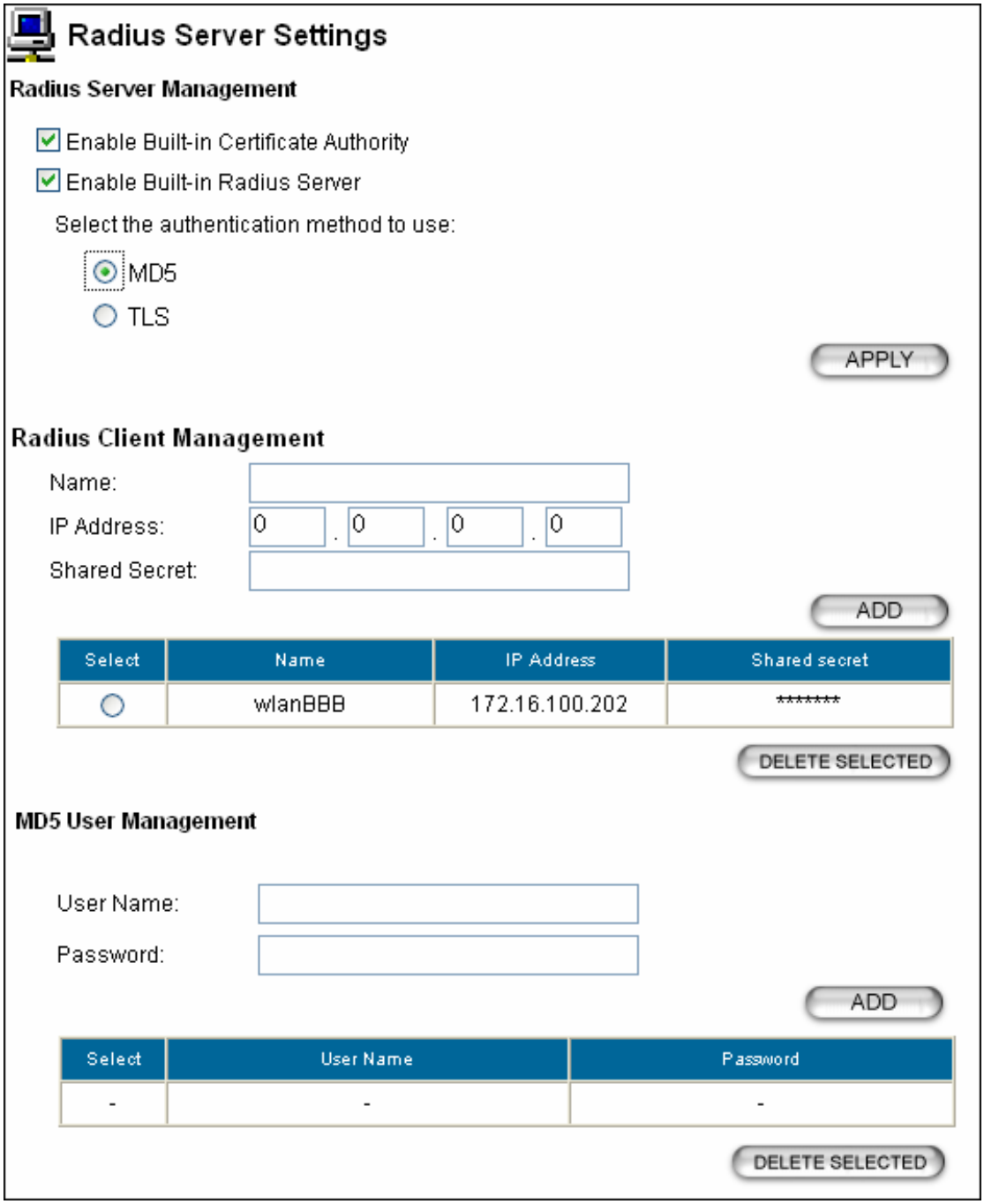

圖 24、RADIUS Server 的管理畫面-使用 MD5 認證及 CA

因為同樣使用內建 CA，所以不需要從外部輸入憑證。還有，我提供了一個記錄使用 MD5 認證的使用者名稱與密碼的表格，方便使用 MD5 認證的使用者的管理。

5.1.6 RADIUS Server 的管理-使用 TLS 認證但不使用內建 CA

如下圖是不使用內建的 CA 以及認證方式為 TLS 的情況：

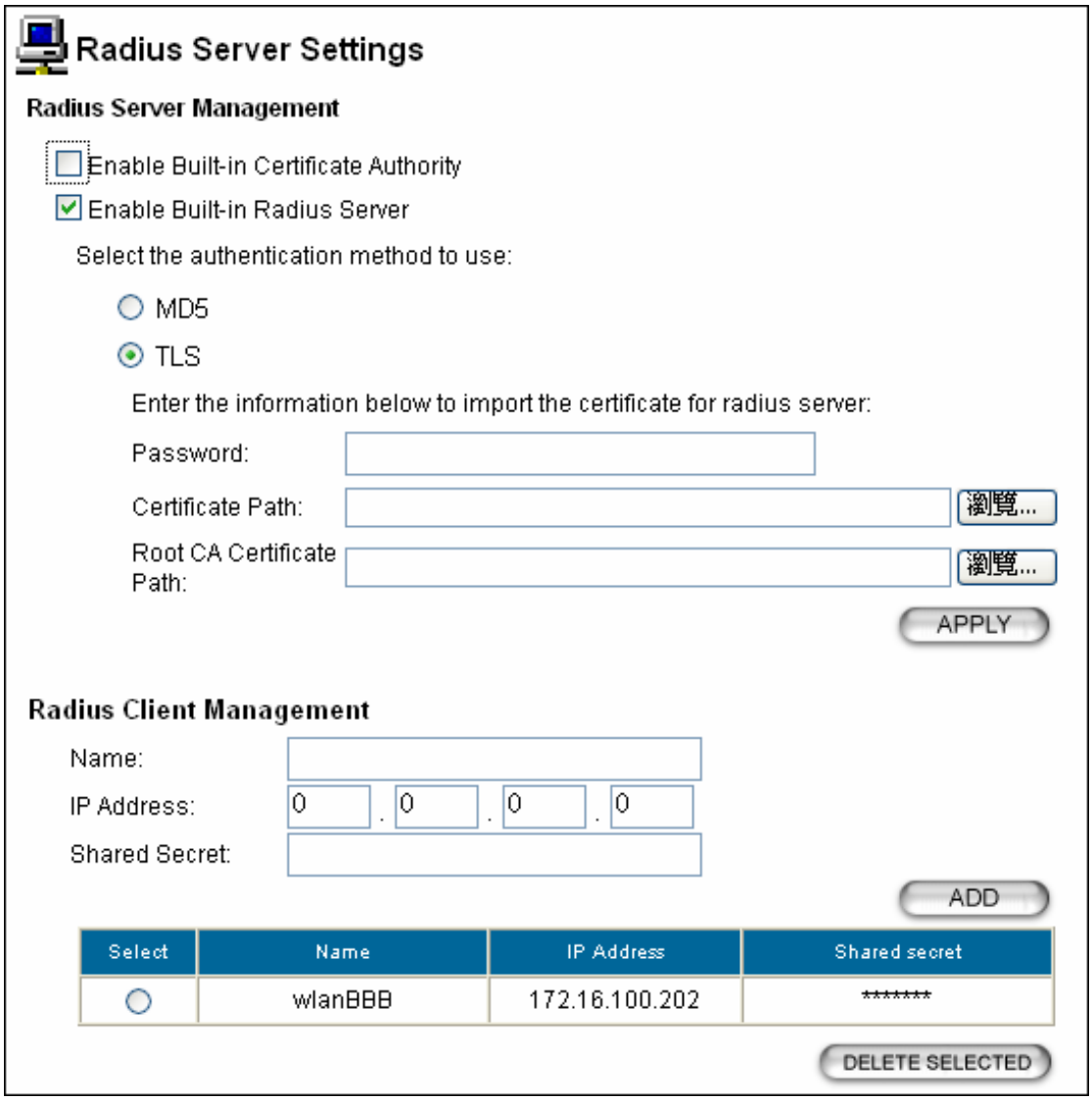

圖 25、RADIUS Server 的管理畫面-使用 TLS 認證但不使用 CA

與 5.1.4 不同的是，因為不使用內建的 CA，所以必須輸入外部 RootCA 的憑證以及給 RADIUS Server 的憑證，還有解開私密金鑰的密碼。以上三個選項只要輸入一次，然後點選“APPLY”按鍵即可。

5.1.7 RADIUS Server 的管理-使用 MD5 認證但不使用內建 CA

如下圖是不使用內建的 CA 以及認證方式為 MD5 的情況：

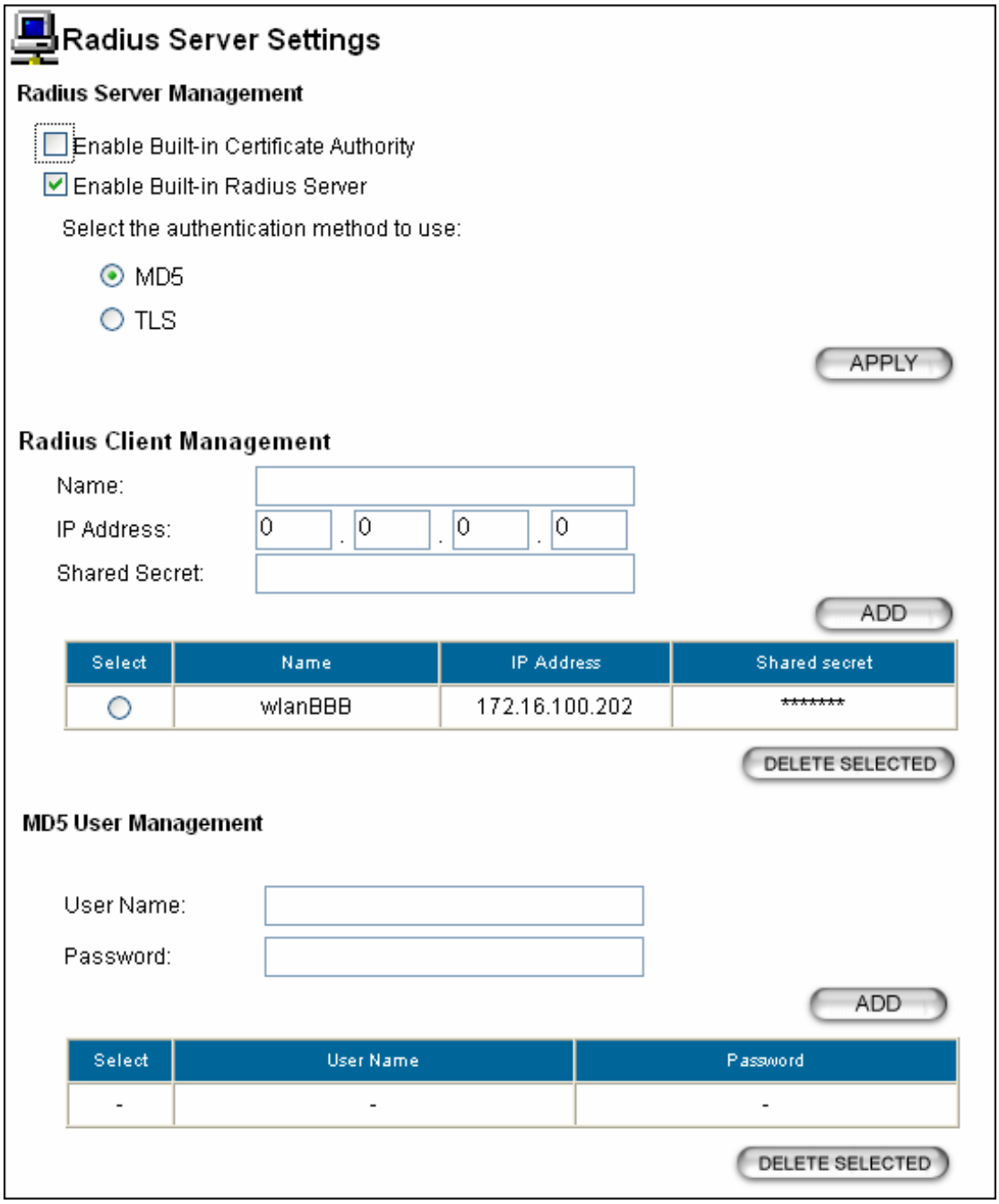

圖 26、RADIUS Server 的管理畫面-使用 MD5 認證但不使用 CA

因為不使用內建的 CA，所以理由同 5.1.6。因為使用 MD5，所以理由同 5.1.5。

5.1.8 **Wireless 連線的管理**

以下是基本的 Wireless 連線的部份。如圖所示：

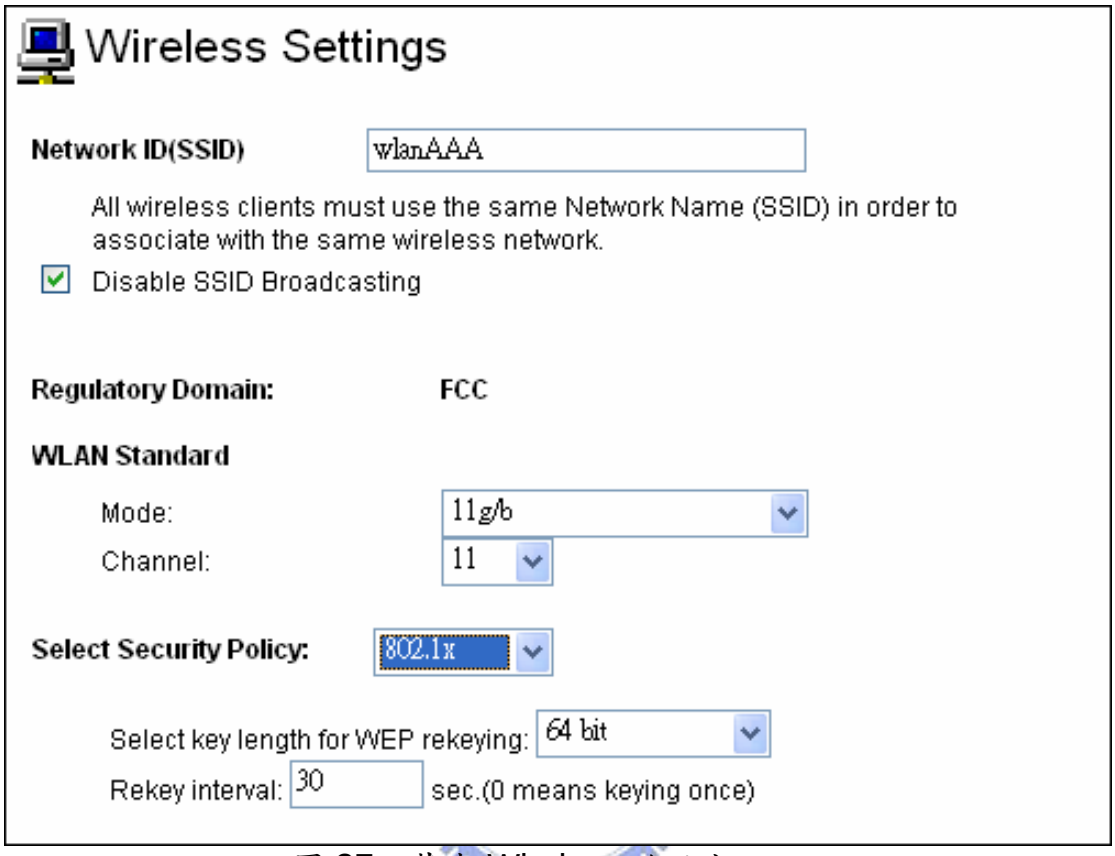

圖 27、基本 Wireless 的設定

圖中的設定值必須要靠使用者的正確設定，才能正常的工作。例如上圖所顯示的 SSID、802.11a/b/g、Channel、以及無線安全的對策。不只如此，我們還可以設定 RADIUS Server 給的 WEP rekey 的長度，還有多久 Rekey 一次，以秒為單位。

5.2 **Windows 的憑證管理**

之前提到過，憑證必須要透過 Windows 的管理，目前大部份的使用者是使用 Windows 作業系統。如何輸入憑證？在 5.1.2 已提到過。但是，怎樣確認憑證已經正確的在系統中了呢？在下圖中可以對照。使用者只要從“開始”、“控制台”進入，點選“網際網路選項”，然後再點選“內容”、“憑證”，即可看到如圖中的“個人”項目，其中的“8867578”的憑證，

就是由系統的 RootCA 發出的，RootCA 的名字為

“RootCA_00:60:67:67:25:C3” 。

圖 28、檢查 Windows 系統內的無線使用者憑證

別忘了同時也檢查一下，是否 RootCA 的憑證也在系統的憑證管理中？可以到“信任的根憑證授權”的頁面中尋找。如下圖所示：

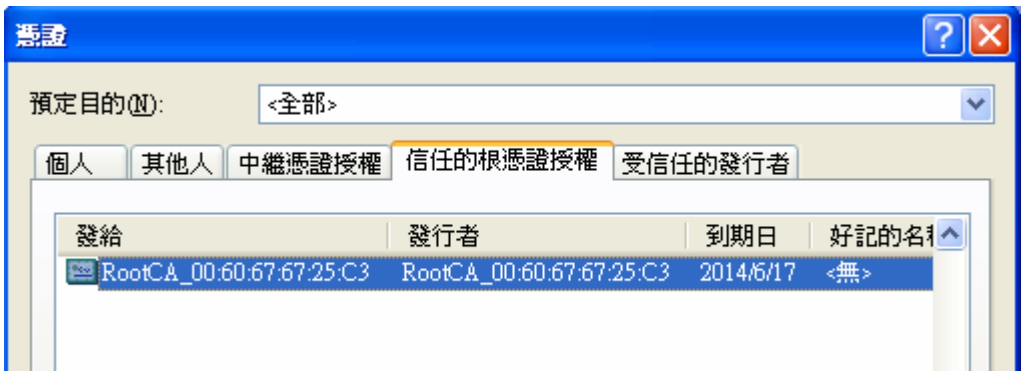

圖 29、檢查 Windows 系統內的 RootCA 的憑證

5.3 Wireless Client Utility

設定完無線路由器，憑證也已輸入之後，接著就要設定使用者端的

無線的連線環境。使用者端的設定其實也有點複雜，首先使用者必須要準備有支援 802.1x 認證的無線網路卡，有些早期的卡並未支援，通常有支援 802.11a 或 802.11g 的卡應該都可以使用。在安裝 Windows 的驅動程式的同時，應該也會安裝由製造廠商所提供的連線工具程式。廠商所提供的工具程式通常會比較實用。不過我還是藉由 Windows 本身提供的設定工具程式來說明。

通常在“網路連線”當中會有一個新增的“無線網路介面”。點選它的“內容”進入設定。請注意圖中圈起來的部份。

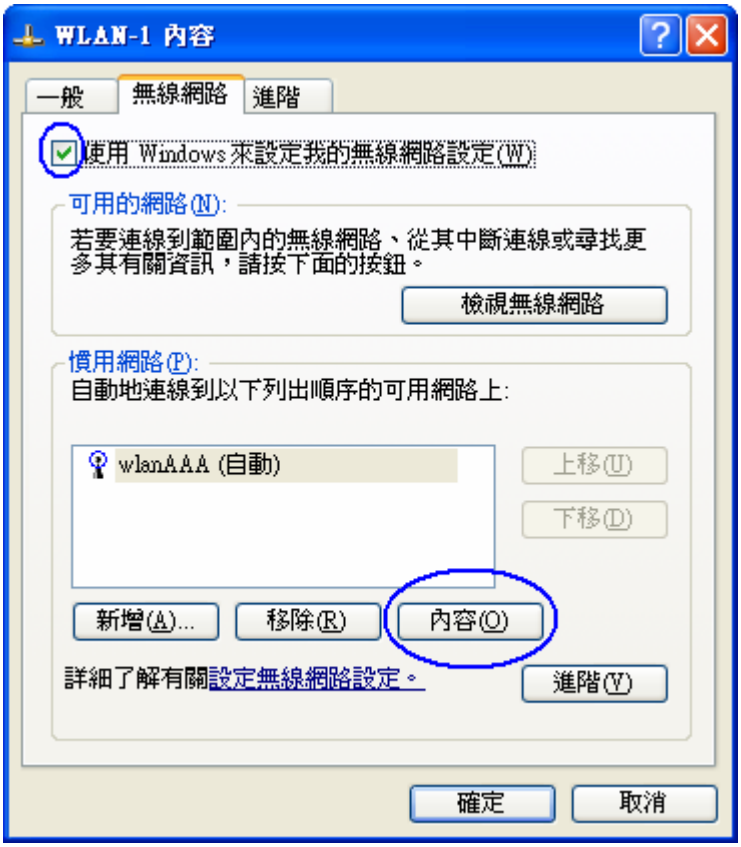

圖 30、使用者端設定畫面之一

點選“內容”進入下圖。

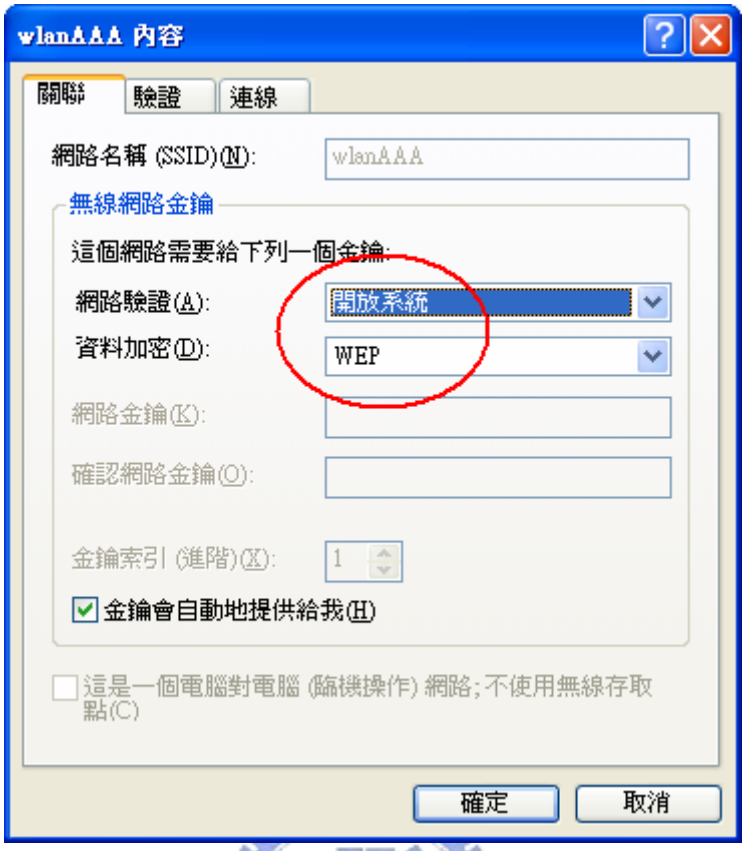

圖 31、使用者端設定畫面之二

如圖 31，點選“開放系統”網路驗證和“WEP”資料加密。因為例子是要使用 WEP key 的 Rekey 功能。然後在驗證的頁面點選啟用 IEEE802.1x 驗證，如下圖。

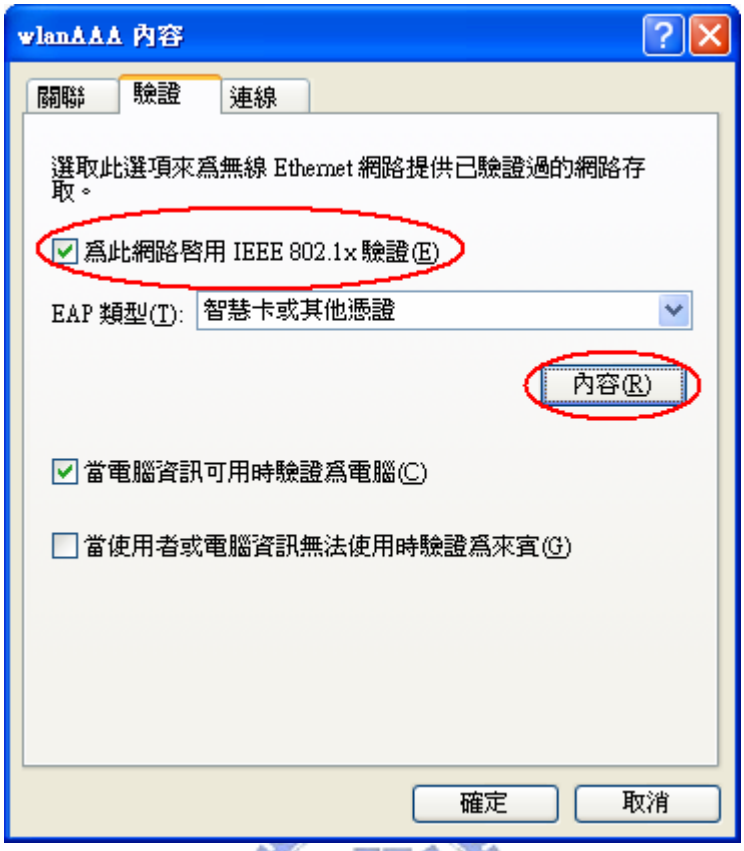

圖 32、使用者端設定畫面之三

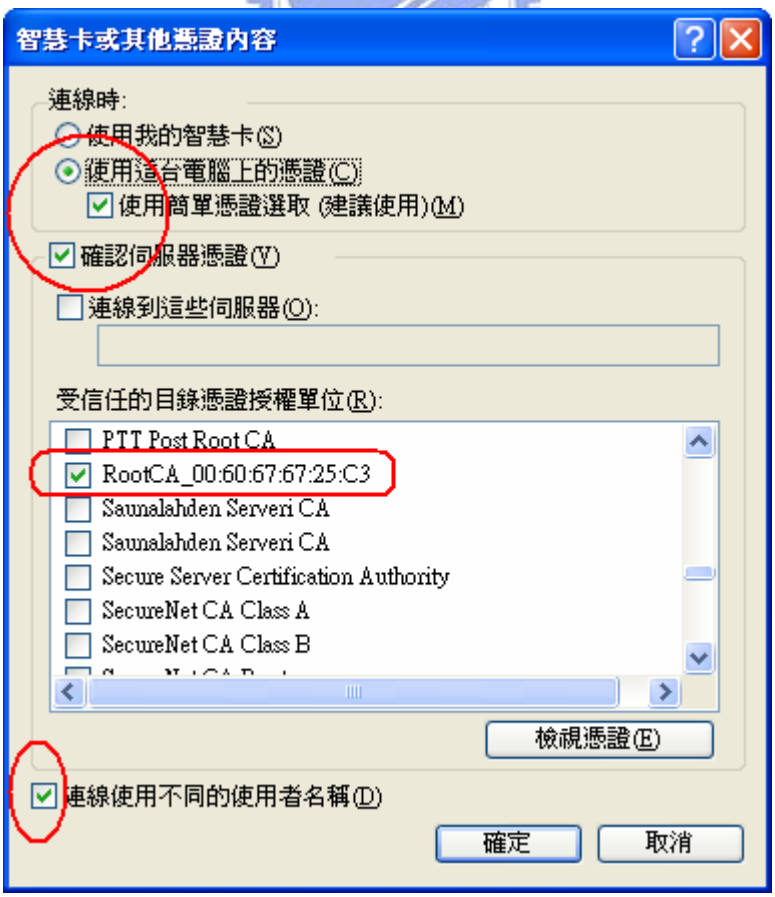

圖 33、使用者端設定畫面之四

如圖 33，注意其中使用到的憑證描述，點選”連線使用不同的使用者名稱(D)”，代表的意思是它會在認證過程中可以讓使用者自己去點選欲使用的憑證。按完確定之後，接下來就可以實際連線了。

因為系統使用 802.1x 的 EAP-TLS 的連線，並且有啟動 Rekey 的機制，當固定的時間一到，系統會自動更新 WEP key，於是整個網路安全就可以確保。這樣簡單方便的、安全的無線網路使用環境，就是本論文所提出的願景。

5.4 無線網路封包之擷取

為了確保無線網路環境是否有按照設定在正確地工作著，在這一節當中，我將使用 WildPackets 這家公司所開發的 AiroPeek 這一套擷取無線網路封包的工具程式，來看看 802.1x 無線網路封包的傳輸情形。

因為是要擷取無線網路封包，所以一定要使用兩張無線網路卡才可以。一張無線網路卡做為正常連線使用，另外一張無線網路卡就做為擷取封包使用。

下圖就是透過 AiroPeek 擷取到的封包，為了識別方便，我將無線路由器的 MAC 位址用“Wireless Router”來表示，Notebook 端的 MAC 位址用“Client”來表示：

P..	Source	Destination	Size	Relative Time	Protocol
1	Wireless Router	Client	71	00.000000	802.11 Probe Rsp
2	Wireless Router	Client	71	00.037912	802.11 Probe Rsp
3	Wireless Router	Client	71	00.906032	802.11 Probe Rsp
4	Wireless Router	Client	71	00.907627	802.11 Probe Rsp
5	Wireless Router	Client	71	00.909161	802.11 Probe Rsp
6	Wireless Router	Client	71	00.991685	802.11 Probe Rsp
7	Client	Wireless Router	41	01.034539	802.11 Auth
8	Wireless Router	Client	34	01.035901	802.11 Auth
9	Wireless Router	Client	34	01.036447	802.11 Auth
10	Wireless Router	Client	34	01.037097	802.11 Auth
11	Client	Wireless Router	64	01.038134	802.11 Assoc Req
12	Wireless Router	Client	50	01.039827	802.11 Assoc Rsp
13	Wireless Router	Client	50	01.040888	802.11 Assoc Rsp
14	Wireless Router	Client	45	01.043712	EAP Request
15	Client	Wireless Router	41	01.078372	EAPOL-Start
16	Wireless Router	Client	45	01.079596	EAP Request
17	Client	Wireless Router	52	01.080968	EAP Response
18	Wireless Router	Client	46	01.087745	EAP Request
19	Client	Wireless Router	28	01.109122	802.11 Data
20	Client	Wireless Router	170	01.120020	EAP Response
21	Wireless Router	Client	967	01.142365	EAP Request
22	Client	Wireless Router	1028	01.153749	EAP Response
23	Wireless Router	Client	101	01.368486	EAP Request
24	Client	Wireless Router	46	01.369013	EAP Response
25	Wireless Router	Client	44	01.382739	EAP Success
26	Wireless Router	Client	89	01.406456	EAPOL-Key
27	Wireless Router	Client	89	01.452869	EAPOL-Key
28	Wireless Router	Client	89	29.239795	EAPOL-Key
29	Wireless Router	Client	89	29.287078	EAPOL-Key

圖 34、使用 AiroPeek 擷取的無線網路封包

從擷取到的封包中可以看出，真正開始 802.1x 溝通是從第 7 個封包開始，到第 27 個封包結束，這時候使用者已可以連線網際網路。整個溝通時間約在 0.5 秒內結束。第 28、29 個封包是 Rekey 機制啟動的結果，它的 Rekey 區間是每 30 秒一次。

5.5 效能分析

關於效能的部分，我所使用的工具程式是 NetIQ 這家公司所開發的 Chariot 這個程式，版本是 v4.3。測試環境是，路由器的 LAN 端與 PC 相連，Notebook 與路由器之間是透過 Wireless LAN 連接。測試數據是

Wireless LAN 上的數據，TX 是路由器送出的部份，RX 是路由器收進來的部分。不同的 Radio mode，數字也許會不同，至於 Channel 的部分因為幾乎不會有影響，所以我挑選 11a mode 的 channel 36 和 11g mode 的 channel 6 做為測試的 channel。

其實不容易做到精確，各種因素都有可能影響，例如不同的無線網路卡，無線空間的干擾，以及路由器和 Notebook 和對測的 PC 的效能也都多少有關係。

下表列出平均值做參考，數字單位是 Mbps：

表三、WLAN 之效能表

測試項目	11a mode, Channel 36		11g mode, Channel 6	
	TX	RX	TX	RX
None	25.99	23.31	25.67	22.98
WEP64	24.88	*15.50	22.50	*14.94
WEP128	23.98	*15.54	23.58	*14.87
802.1x without rekey	25.56	23.99	24.85	22.76
802.1x with rekey function for every 60 seconds	25.13	22.93	21.80	21.13
802.1x with rekey function for every 300 seconds	25.58	22.95	20.72	20.15

由上表可以看出，在使用 WEP 之後，TX 的部份會下降是正常的現象，因為必須要做 Encryption。但是 RX 的部份，數字卻有明顯的下降，雖然系統此時需要做解碼的動作，但是理論上差別不會這麼大，這是目前已知仍未解決的問題之一。另外，有否使用 802.1x，有無啟動 rekey 的功能或者 rekey 的間隔時間多寡並不太會影響系統的效能。

另外，我再提出一個數據，那就是產生憑證所需的時間。當 CA 在產生憑證給使用者的時候，其實並沒有規定必須使用多少 bits 的金鑰。於

是，我就做了一個實驗，分別測量不同長度的金鑰與實際產生憑證的時間。結果產生 512bits 金鑰憑證的時間約在 20~30 秒之間。而產生 1024bits 金鑰憑證的時間約在 40 ~50 秒之間。為了安全性的考量起見，使用長度較長的金鑰是比較保險的，雖然使用者必須在第一次產生金鑰時等待比較久的時間。

六、 總結

本論文所設定的目標是提供使用者一個夠安全與容易架構使用的無線網路環境。誠如論文題目所說的，實作以憑證為認證基礎的無線網路路由器，我將 Free RADIUS 與 Openssl 的功能整合進一台小小的路由器當中。使用者將不再需要其他的 CA 以及 RADIUS Server，只要這台無線網路路由器就可以了。它讓使用者可以透過憑證，802.1x 的 EAP-TLS，與無線網路路由器溝通，確保了無線傳輸的安全性。至於設定使用的部分，我所設計的 WEB GUI 以及 CGI，則提供了使用者一個方便好用的圖形化介面。

6.1 回顧

這個無線路由器系統大約是兩年前才開始使用 Embedded Linux 系統的。最主要的兩個 Process：Main Configure Process 和 CGI 都是從零開始發展的。CGI 和 WEB/GUI 的部分是我負責設計與開發的，等到主要架構與資料流貫通完成了之後，後段的工作就可以以頁面為單位來完成。

回到主題的 FreeRADIUS 與 Openssl 這一個部份，個人覺得最大的因素在於規劃，事前的分析工作絕對不可以少。例如它的工作模式分析以及程式架構的規劃，與已有的各個模組間如何協同工作，還有最重要的可行性分析。在可行性的分析中，由於本系統也可以有 Linux PC 平台的關係，於是我可以先在 Linux PC 平台來實驗與開發，實驗成功後再導入到實際系統上，節省了不少平台轉換的時間。

6.2 未來的工作

前面曾提到幾項未完成的工作以及有問題的部份，我現在再把它們

整理一下：

1. 尚未支援有效憑證與無效憑證的管理。
2. 再加長 RootCA 憑證，以及使用者憑證的效期。
3. 系統重置（Clear configuration and Reset）之後，舊有憑證目前無法保留。
4. 沒有完整支援 Windows 2000 Server 憑證的簽發與匯入。
5. 802.1x 的認證部份只支援 EAP-MD5 和 EAP-TLS，未來考慮支援 EAP-TTLS 和 EAP-PEAP。
6. 尚未支援 WPA2 的規格。
7. 尚未支援未來的 802.11i 的規格。

參 考 文 獻

〔1〕B. Aboba, D. Simon, RFC 2716, PPP EAP TLS Authentication Protocol, IETF, October 1999。

〔2〕C. Rigney, S. Willens, A. Rubens, W. Simpson, RFC 2865, Remote Authentication Dial In User Service (RADIUS), IETF, June 2000。

〔3〕C. Rigney, W. Willats, P. Calhoun, RFC 2869, RADIUS Extensions, IETF, June 2000。

〔4〕Institute of Electrical and Electronics Engineers Standard 802.1x, IEEE Standard for Local and metropolitan area networks – Port – Based Network Access Control, June 2001。
URL:http://standards.ieee.org/getieee802/download/802.1X-2001.pdf。

〔5〕Jon Edney，William A. Arbaugh，Real 802.11 Security: Wi-Fi Protected Access and 802.11i，1st Edition，Addison-Wesley，July 2003。

〔6〕Jonathan Hassell，RADIUS，1st Edition，O'REILLY，Oct 2002。

〔7〕L. Blunk，J. Vollbrecht, RFC 2284, PPP Extensible Authentication Protocol (EAP)，IETF，March 1998。

〔8〕Mattbew S. Gast, 802.11 Wireless Networks: The Definitive Guide, O'REILLY, April 2002。

〔9〕R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, RFC 2616, Hypertext Transfer Protocol – HTTP/1.1, June 1999。

〔10〕T. Berners-Lee, R. Fielding, H. Frystyk, RFC 1945, Hypertext Transfer Protocol -- HTTP/1.0, IETF, May 1996。

〔11〕T. Dierks，C. Allen, RFC 2246, The TLS Protocol - Version 1.0，IETF，January 1999。

〔12〕Tom Karygiannis, Les Owens, Wireless Network Security 802.11, Bluetooth, and Handheld Devices, http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf, November 2002。

自　　傳

我從小於台南善化的農村長大，爸爸曾擔任過公務員，目前退休在家，媽媽務農，還有我們四個兄弟姐妹，我排行老么。家境小康，一家人彼此相處融洽。家父一直教育我們要用功讀書，不要花時間和金錢在玩樂上。雖然一度曾有難關，家母也曾做家庭代工賺錢。在那段時間，我學會了縫雨傘，也才深深體會到靠勞力賺錢的辛苦，以及學問知識的重要。在國、高中時期，別人在補習上課，我則是自己多做練習，不會的再向老師請教。所以從來不曾讓爸媽擔心課業的問題。後來，考上交大資訊工程系，當時稱為計算機工程系，從此與電腦結下不解之緣。

剛上大學的時候，因為自己從未接觸過電腦，計算機概論的程式作業每每讓我痛苦萬分。有感於自己將來必須靠電腦吃飯，於是發奮要練好英文打字及中文打字，甚至連倉頡輸入法也是靠自己摸索的情況下練就的。很感謝交大資工的師長們四年來的教誨，讓我有著豐厚的學識基礎，來面對未來的工作挑戰。

服役的時候，在宜蘭縣的某兵工廠當資訊士，每天在機房，工廠以及網路維修中渡過。兩年下來，學的東西也不少。尤其是人事系統，當我嘗試用交大所學的系統分析對它做資料庫系統的重新規劃，撰寫的報告還得到聯勤總部的肯定，讓我有機會對各個兵工廠的校、尉級軍官及其他士官們分享系統分析與規劃的經驗。

退伍後，在交大資科系擔任數位實驗助教與系計中的助教。感謝資科系的莊仁輝教授，讓我在這一年中又學到與眾不同的經驗。實驗室的管理，以及系計中機房及網際網路的架構與管理。

後來，進入全譜電腦開發多功能事務掃瞄機。當時開發的掃瞄機是可以直接接上印表機的，我負責開發的是 PC 端的 Bypass Printer Driver，這樣的點子在當時應該是非常創新的，可惜在當時沒有客戶有興趣。直到我進入鴻友科技開發數位相機的兩年後，市面上才出現了 HP 的 Office Scanner 的機器，多功能事務機的市場才真正起來。這時我才知道，多功能事務機必須要能整合 Scanner、印表機還有 FAX 的功能在一起才有市場啊。

鴻友科技開發數位相機的時間點算很早的，我從第一代數位相機的 RS232 Driver 開始做起，一直做到第三代八十萬畫素的 USB 機種。我從 PC 端最底層的 Driver 做到 TWAIN Data Source 以及後端的影像處理。當我想到未來應該是網路通訊的天下時，於是我進入了連碁科技。

在連碁科技，我參與了 ISDN TA、ISDN Router、Broadband Router、Access Point、以及 Wireless Router 的計畫。也就是在連碁，我得知了交大在職專班招生的訊息，然後也順利的考取專班。重拾書本之後，再次當學生的心情真的很不一樣，有一種更踏實的感覺，在求學、做計劃與寫報告的過程中。

雖然說學業、工作與家庭，三方面想要同時兼顧，有時真的很難。曾經因為工作的關係，我中斷了學業，結果還是無法避免的犧牲了許多與家人相處的時間。最後，非常感謝我的父母、我的太太、以及兩個小孩，這樣的體諒我及支持我。有人說，小孩子的成長只有一次。將來，我會用更多的時間陪伴他們。