結合Java卡與PKI之Ipv6 IPSec系統設計與實作

梁家瑋，高雄師範大學資訊教育研究所

楊中皇，高雄師範大學資訊教育研究所

摘要

由於網際網路的快速發展，傳統的 IPv4 位址數目已不敷使用。為了滿足未來網際網路的使用量，因而產生了新的 IPv6 網際網路協定。IPv6 與 IPv4 大大不同，而其中已內建了 IP 安全機制 (IPSec)，能確保網路傳送資料的安全性。Java 卡是新一代的智慧卡，並具備密碼學相關的 API，本研究利用 RSA 公開金鑰演算法來達成 IPSec 中 IKE 的認證，並利用 Java 卡來儲存重要的私密金鑰 (Private Key)與憑證(Certificate)，在進行 IPSec 通訊時，私鑰便可透過讀取 Java 卡來取得，通過認證後，建立安全連結(Security Association,SA)，保障伺服端與客戶端通信的安全。除此之外，本研究中加入了 OpenCA 這個開放原始碼的程式當作一個第三方的公開憑證管理中心，負責存送使用者的公開金鑰 (Public Key) 與管理發送憑證，讓系統符合 PKI 的架構，使得認證的機制更具安全及便利。

本研究伺服端的作業系統為FreeBSD，並修改開放原始碼套件KAME來建立IPv6與IPSec環境，客戶端為Linux作業平台，除此之外，我們還搭配一個發卡系統，來產生伺服端與客戶端對應的Java卡。

關鍵字： IPv6、IPSec、Java卡、RSA、IKE、OpenCA、KAME、PKI

1. 緒論

在網際網路上有著多不勝數的伺服器，使用者要連結到網站都必須透過IP位址來建立連線，在傳送資料封包的同時，可能會出現有心人士在其中監看並擷取封包。有關使用者的重要資訊被盜取後，駭客可能會進行竄改的動作，導致使用者的損失，由此可見，IP安全機制(IPSec) [9,17,18]的存在是有其必要性的。

IPSec所提供的功能讓我們在區域網路、公開的廣域網路或者網際網路上都可以進行安全的通訊。在不久的將來，我們的網際網路將會全面升級至 IPv6 [1]，網路位址的數量相當充足，共有 2 的 128 次方個位址，假設全世界的人口約略估記為一百億，平均一個人約可分到3495 個IP，由此觀之，迎接IPv6 的到來是勢在必行的。目前的作業系統大部份都已支援IPv6，例如Windows、FreeBSD及Linux新版之核心均已支援，即使沒有獨立的硬體設備，也可透過Tunnel Broker [2]的方式連上IPv6 的網際網路，對於IPv6 來說，IPSec已預設為內建機制，也就是說，網路安全在網際網路中扮演著相當重要的角色。

Java 卡 [6,20]是最近極熱門的智慧卡類型，它可以利用 Java 程式語言來設計卡片上的應用程式，稱之為 Applet，並利用其運算能力，作簡易的資料運算；程式的開發也比一般 IC 卡來得容易，包括了加密、解密、雜湊函數、數位簽章等密碼學的 API 可用。

近年來政府所倡導的自然人憑證，即為 PKI [14,16]的一種應用，在 PKI 之中，CA (Certification Authority)為一重要的成員，我們希望系統能加入公正的第三方，也就是憑證管理中心，提供申請與管理憑證的功能。在本研中利用開放原始碼的套件 OpenCA [13]來架設憑證管理中心伺服器。

本研究以設計與實作的方式係於伺服端 FreeBSD平台與客戶Linux平台之間建立起IPSec，網路環境是在IPv6之下，並使用Java卡來儲存私鑰，配合RSA數位簽章達成認證的機制，透過憑證管理中心核發憑證及儲存使用者的公鑰；IPSec的支援則是利用開放原始碼KAME [8]套件，並希望將Java卡及OpenCA整合入IPSec系統。

2. 文獻探討

本章節就本研究所用到的理論、演算法與程式技術做探討；其中包含了相關的密碼學演算法、IPv6 網路協定、IPSec 機制、Java 卡、KAME、憑證管理中心、PKI 等理論技術及簡介。

2.1 相關的密碼學演算法

RSA [17,18]演算法是一種非對稱式密碼學演算法，也是公開鑰匙加密演算法的一種，是由美國麻省理工學院的三位教授Ron Rivest、Adi Shamir與Len Adleman所提出，可用來作為加密演算法，亦可用為數位簽章來驗証身份。它是一種基於分解因數的指數函數以做為單向暗門函數的演算法，且指數函數具有可逆性與交換性。本研究則是利用RSA數位簽章當作認證的演算法。

2.2 IPv6網路協定

IPv6 [1]是網際網路通訊協定的新版本，是為了

因應IPv4網路位址的不足設計而成的，主要的改變包括了位址數目的增加、封包標頭規格、擴充性的支援、流量標記的能力、認證功能與保密性。IPv6出現後，IPv4並不會消失，而是採取共存的狀態，主要因應的機制包括了Dual Stack(雙重協定架構)、Tunneling(隧道技術)、Tunnel Broker(隧道代理人)、網路地址與協定轉換等；以下會針對本研究所用的Tunneling相關技術作介紹，其他IPv6技術協定則不再贅述。

2.2.1 Tunneling

Tunneling [4]的作法是在IPv6封包進入IPv4協定的網域時，將IPv6封包當作資料在前面加上IPv4標頭，再送入IPv4網域。當資料由IPv4網域離開進入IPv6網域時，再將IPv4的標頭移除，還原回原本的IPv6封包。隧道技術適合用於網路的兩個終端是IPv6協定網域，而中間都是IPv4協定的網域。如此作法可以透過現有的IPv4協定骨幹網路，將局端的IPv6主機相互連結，在IPv6運用初期是極為方便的技術。

2.2.2 Tunnel Broker

Tunnel Broker [2]技術可說是隧道技術的其中一種，它可被視為虛擬的 IPv6 ISP，主要提供 IPv4 網際網路使用者連接上 IPv6 網路；使用者必須具備一個 IPv6 的主機或路由器，申請建立隧道，Tunnel Broker 會自動地挑選一個恰當的 Server，然後自動地建立、修改和刪除 tunnel。本研究係透過中研院的 Tunnel Broker 伺服器，取得一個區段的 IPv6 位址以供測試使用。

2.3 IPSec

IPSec [9,17,18]機制提供了 IP 層與 socket 層之間安全的通訊方式，保障了IP資料封包傳遞的安全。IPSec包含了AH [10](Authentication Header)、ESP [11](Encapsulation Security Payload)、IKE [5](Internet Key Exchange)等幾個元件以及傳輸(transport mode)與通道(tunnel mode)兩個模式。

AH提供數據的認證的功能及資料完整性，ESP則是負責保密性、資料來源驗證、抗重送服務，IKE用以協商溝通雙方，並包括金鑰交換的資料及支援各種鑰匙的交換技巧等。IKE 有兩個協商階段，在第一階段（phase 1），通信雙方間建立了一個已經通過身份認證和安全保護的通道。在第二階段（phase2），就可用第一階段的通道，來建立起IPSec SA [12] (Security Association)。

SA 是兩個通信實體經協商建立起來的一種協定，在IPSec 中使用IKE 協商出SA，並且存入SAD(Security Association Database) [17,22]中。SA 決定了用來保護資料封包安全的IPSec 協定、加密方式、密鑰值、密鑰長度以及密鑰的有效存在時間等。IPSec 體系中有另一個元件，稱之為SPD(Security Policy Database) [17,22]。SPD記載了IPSec SA中定義了什麼模式或使用什麼協定、如何處理封包等，與SAD均為IPSec重要的元件。

2.4 Java卡技術

Java卡 [6,20]是智慧卡的一種，包括了內嵌的中央處理器以及多種記憶體，有些智慧卡還包含有數學輔助運算器。Java卡與一般IC卡不同之處在於它允許智慧卡與其他有著記憶體大小限制的裝置上，都可以執行使用Java程式語言所開發的應用程式(applets)。其特點包括了軟體易於開發、安全、硬體獨立性、儲存與管理多個應用程式的能力與智慧卡標準的相容性。

2.5 KAME

KAME [8]是一支開放原始碼的軟體套件，是一個由日本的幾間大公司所共同參與開發的計劃，計劃從1998年開始，並持續在更新套件的程式。而KAME主要提供的功能包含了IPv6、IPSec、以及一些進階的網路功能，像是ATM (Asynchronous Transfer Mode) 網路，網路行動性(mobility)等。而KAME是本研究伺服端支援IPv6與IPSec的主要核心套件，並將修改其原始碼符合其設計的系統。

2.6 憑證管理中心

數位憑證是一份經過了數位簽署、每個憑證都包含了某位使用者的公鑰，並且都已經用可信賴機構的私鑰簽署過而成一個電子文件。而憑證管理中心(Certification Authority,CA) [14,17,18]為一公正的第三方，負責管理與發放憑證，只要是可信任的CA所發的憑證便可證實某一方的身份，保障其安全性。

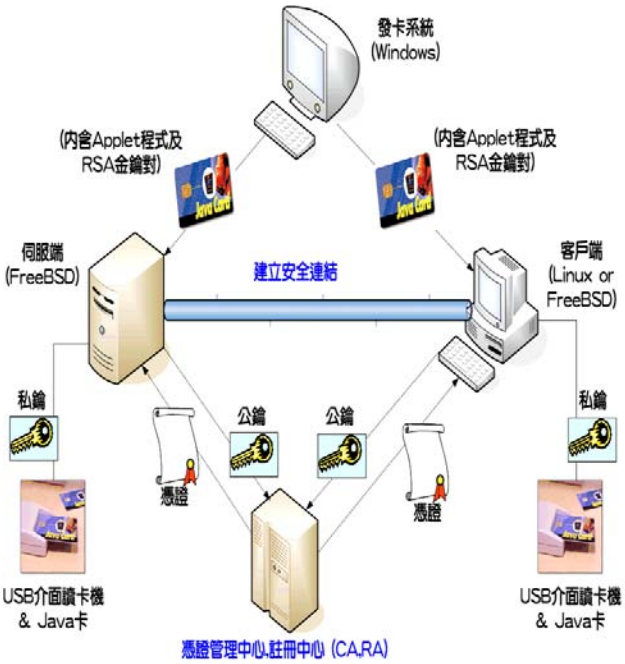
本研究CA的架設乃使用開放原始碼的軟體OpenCA [13]架設而成的，OpenCA 是一個以Perl語言設計的開放原始碼軟體，底下包括CA、RA(Registration Authority) 、 LDAP(Lightweight Directory Access Protocol)等功能。

2.7 PKI

公鑰密碼系統不需要傳統對稱式秘密交換或分發共用密鑰，而是使用兩把金鑰非對稱式的加密演算法。RSA是現今使用率最高的公鑰密碼系統；而公開金鑰基礎建設（Public Key Infrastructure，PKI）[14,16]是運用公開金鑰加密及認證的方法，以確保資料通訊的安全性及確認通訊對方身分的一個機制，因此本研究所採用了CA與公開金鑰演算法，其目的便是符合PKI的基本精神。

3. 系統架構

本研究的系統架構分為四部份，有一端為FreeBSD系統，如同電子商務系統，提供使用者服務的同時也建立其IPSec機制，因此在此稱為伺服端。客戶端則是Linux或FreeBSD系統皆可，在此客戶端平台則為Linux Fedora Core3，CA架設在Linux RedHat 9.0平台之上，發卡系統環境則為Windows XP，整個系統架構如圖一所示。



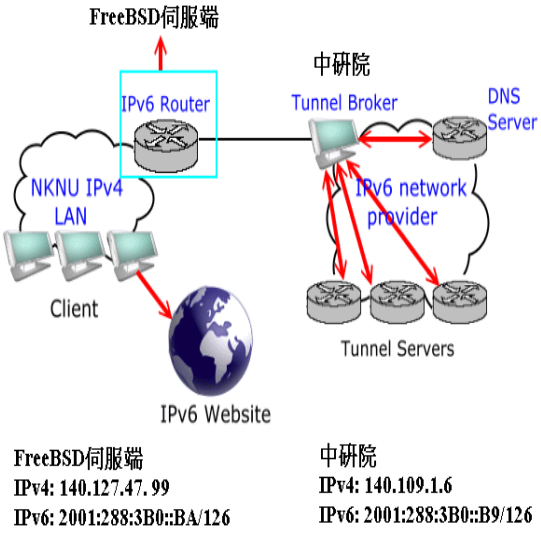
圖一 本研究系統架構圖

在發卡系統方面，我們開發Terminal(主機端)與Applet(卡片程式端)程式，利用Applet產生RSA金鑰對，可讓伺服端與客戶端利用RSA公鑰向CA申請憑證，待CA核準後，客戶端及伺服端均可連結上CA下載憑證，並把憑證與RSA私鑰透過讀卡機存於Java卡之中，並用加密演算法及PIN碼保護。

在伺服端與客戶端方面，我們利用開放原始碼套件KAME底下的一支程式Racoon來達成IPSec，對伺服端來說，安裝KAME即包含了這支程式；對客戶端而言，在Linux Kernel版本2.6以後，底下的IPSec tools套件就已包含了Racoon程式。我們修改Racoon程式，在IKE phase1的認證方式使用RSA數位簽章，認證時將RSA私鑰與憑證由Java卡匯出，當認證通後並可開始建立兩端間的IPSec。

3.1 建立IPv6環境

本研究透過Tunnel Broker的方式來建立IPv6環境，在此我們向中研院申請此服務，並透過更改KAME套件的設定使得伺服端具備IPv6路由器的功能，因此在區域網路內的電腦均可連結上IPv6網路，網路架構圖如圖二所示。



圖二 本研究IPv6網路架設圖

3.2 CA伺服器

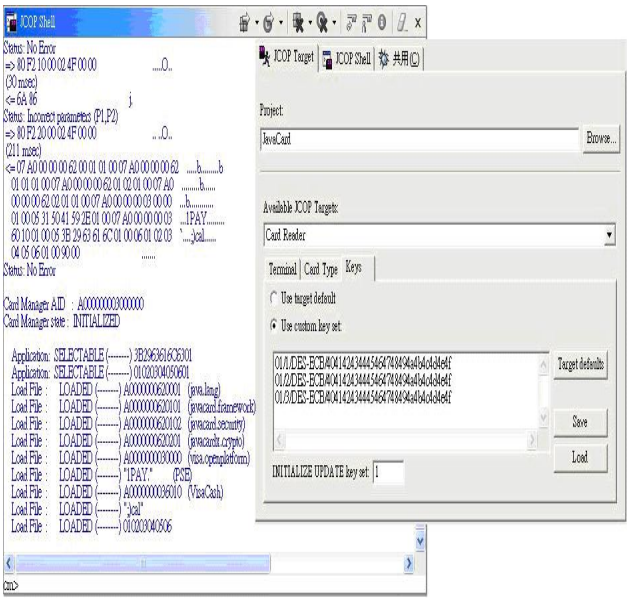
利用開放原始碼軟體OpenCA我們可以架設一個獨立的CA伺服器，客戶端及伺服端均可利用HTTP協定連到CA伺服器申請憑證，如圖三所示。

圖三 CA伺服器申請憑證畫面

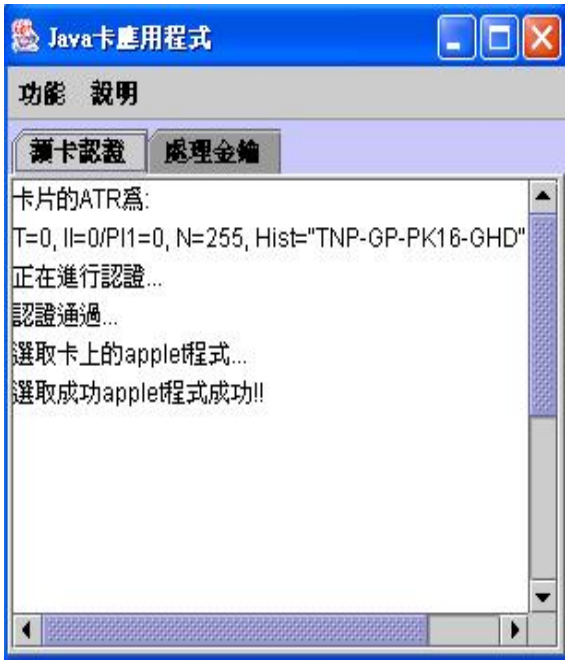
3.3 發卡系統

發卡系統的環境為Windows XP，使用Java程式語言搭配Eclipse [3]及JCOP tools [7]開發Java卡Applet與Terminal程式。JCOP tools為IBM公司針對智

慧卡所設計的一套開發工具，包括了卡片模擬、認證、載入程式、開發與除錯Java卡Applet等功能。與Java卡溝通之前必須先通過認證，才能存取卡上的相關資源與上傳應用程式。發卡系統功能為開發applet程式，與上傳CAP(Converted Applet)檔案到Java卡之上，建置完整的卡片相關功能。圖五為Terminal程式讀取與認證Java卡畫面。



圖四 發卡系統上傳applet至Java卡



圖五 Terminal程式讀取與認證Java卡

3.4 IPSec客戶端

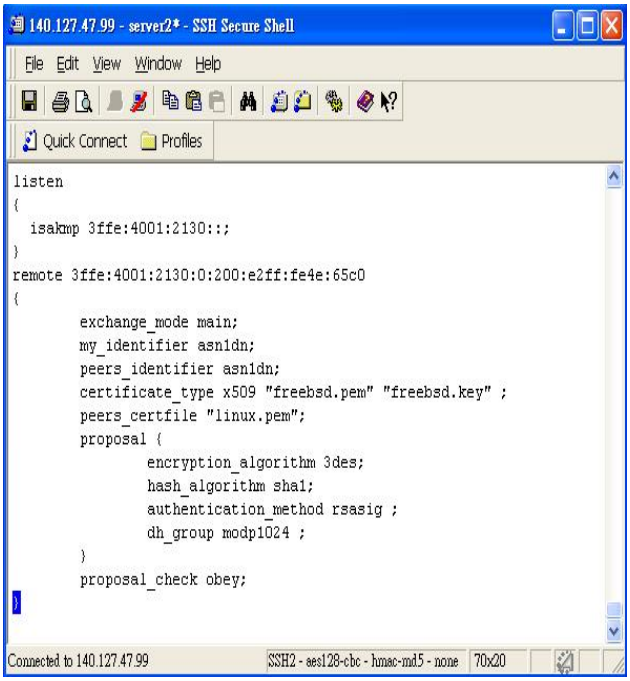
對客戶端來說，建立IPSec的步驟如下：向發卡系統索取卡片，卡片上包含RSA金鑰對，接著用RSA公鑰去向CA申請憑證，私鑰則可以存入Java卡之中保存以防被竊取；當要申請IPSec服務時，可以利用Java卡中設計的程式(Applet)及私鑰將資料做RSA數位簽章，再把簽章值傳送給伺服端；伺服端確認無誤後，即可對客戶端提供IPSec的服務，圖六為Terminal程式憑證與私鑰從Java卡匯出之畫面。

圖六 由Java卡匯出憑證與RSA私鑰

3.5 IPSec伺服端

本研究的伺服端乃於FreeBSD下安裝及修改KAME套件，除此之外，IPSec的建立也是利用KAME這個套件，並且接上USB介面IC讀卡機以供存取私鑰之用。

與客戶端建立安全連結前，首先要先去CA下載客戶端的憑證，憑證內包含客戶端的公鑰資訊等。在KAME底下有一應用程式Racoon，利用Racoon可以定義IKE的認證方式及演算法，我們所使用的認證演算法乃是RSA數位簽章，有關私鑰的讀取方面，我們修改Racoon的原始碼，改成由Java卡透過讀卡機導入RSA私鑰。當收到客戶端發出的IPSec請求，伺服端向CA下載客戶端的憑證，憑證內包含客戶端的RSA公鑰，可以將客戶端傳送的資料解密並檢查比對雜湊值，假設無誤後，便可設定認證的演算法和讀取RSA私鑰以建立IPSec。核心設定檔如圖七所示。



圖七 Racoon核心設定檔

3.6 系統測試

我們在伺服端以一個簡單的密碼認證網頁來測試，如圖八所示，是IPv6格式的。當伺服端與客戶端建立起IPSec後，便可進入網頁。有關SA的資訊階存於SAD內，如圖九所示。在傳送資料封包的同時，我們使用tcpdump指令檢測封包的狀況，檢測的結果如圖十所示，可以清楚看到封包已包含AH、ESP協定，表示IPSec已順利啟動，並且已對封包作加密的處理。

圖八 IPv6&IPSec系統測試網頁

```
3ffe:4001:2130:: 3ffe:4001:2130:0:200:e2ff:fe4e:65c0
        esp mode=transport spi=65538(0x00010002) reqid=0(0x00000000)
        E: 7  6b616d65 6b616d65
        seq=0x000000b5 replay=0 flags=0x00000040 state=mature
        created: Mar 19 18:29:45 2005   current: Mar 19 18:42:31 2005
        diff: 766(s)    hard: 0(s)      soft: 0(s)
        last: Mar 19 18:33:04 2005      hard: 0(s)      soft: 0(s)
        current: 22752(bytes)   hard: 0(bytes)  soft: 0(bytes)
        allocated: 181  hard: 0 soft: 0
        sadb_seq=2 pid=499 refcnt=2
3ffe:4001:2130:0:200:e2ff:fe4e:65c0 3ffe:4001:2130::
        ah mode=transport spi=131073(0x00020001) reqid=0(0x00000000)
        A: md5  74686973 20697320 74686520 74657374 206b6579
        seq=0x000000b4 replay=0 flags=0x00000040 state=mature
        created: Mar 19 18:29:45 2005   current: Mar 19 18:42:31 2005
        diff: 766(s)    hard: 0(s)      soft: 0(s)
        last: Mar 19 18:33:04 2005      hard: 0(s)      soft: 0(s)
        current: 22608(bytes)   hard: 0(bytes)  soft: 0(bytes)
        allocated: 180  hard: 0 soft: 0
        sadb seq=1 pid=499 refcnt=1
```

圖九 系統SAD

```
18:32:41.304710 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0x9d): ESP(spi=0x00010002,seq=0x9d)
18:32:42.304410 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0x9d): ESP(spi=0x00010001,seq=0x9d)
18:32:42.304554 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0x9e): ESP(spi=0x00010002,seq=0x9e)
18:32:43.304254 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0x9e): ESP(spi=0x00010001,seq=0x9e)
18:32:43.304398 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0x9f): ESP(spi=0x00010002,seq=0x9f)
18:32:44.304100 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0x9f): ESP(spi=0x00010001,seq=0x9f)
18:32:44.304244 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa0): ESP(spi=0x00010002,seq=0xa0)
18:32:45.298066 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa1): ESP(spi=0x00010002,seq=0xa1)
18:32:45.298304 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa0): ESP(spi=0x00010001,seq=0xa0)
18:32:45.303911 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa1): ESP(spi=0x00010001,seq=0xa1)
18:32:45.304009 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa2): ESP(spi=0x00010002,seq=0xa2)
18:32:46.303787 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa2): ESP(spi=0x00010001,seq=0xa2)
18:32:46.303935 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa3): ESP(spi=0x00010002,seq=0xa3)
18:32:47.303630 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa3): ESP(spi=0x00010001,seq=0xa3)
18:32:47.303775 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa4): ESP(spi=0x00010002,seq=0xa4)
18:32:48.303475 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa4): ESP(spi=0x00010001,seq=0xa4)
18:32:48.303631 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa5): ESP(spi=0x00010002,seq=0xa5)
18:32:49.303318 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa5): ESP(spi=0x00010001,seq=0xa5)
18:32:49.303467 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa6): ESP(spi=0x00010002,seq=0xa6)
18:32:50.303163 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa6): ESP(spi=0x00010001,seq=0xa6)
18:32:50.303316 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa7): ESP(spi=0x00010002,seq=0xa7)
18:32:51.303007 3ffe:4001:2130:0:200:e2ff:fe4e:65c0 > 3ffe:4001:2130:: AH(spi=0x00020001,seq=0xa7): ESP(spi=0x00010001,seq=0xa7)
18:32:51.303151 3ffe:4001:2130:: > 3ffe:4001:2130:0:200:e2ff:fe4e:65c0: AH(spi=0x00020002,seq=0xa8): ESP(spi=0x00010002,seq=0xa8)
```

圖十 封包檢測畫面

4. 結語

本研究在IPv6網路環境下實作了一個符合IKE(RFC 2049)認證的一個IPSec機制，使用Java卡對私鑰的保存添加了安全性，CA讓收發憑證更具便利性。IPv6網路協定是未來的趨勢，而對IPv6來說，IPSec機制是不可或缺的；目前Windows系統平台對於IPv6 IPSec僅支援AH，對於ESP並不支援，但對於IPv4網路環境下的IPSec，Windows XP、Server 2003是支援的；因此本研究的客戶端也以自由作業平台Linux來作測試，相信未來Windows作業平台對於IPv6 IPSec支援後，本系統仍然適用於Windows平台的客戶端，達成跨平台的目的。

5. 參考文獻

1. Deering S. and Hinden R., RFC 2460 IPv6 Specification, IETF, http://www.ietf.org/rfc/rfc2460 , 1998.

2. Durand A., Fasano P., and Lento D., RFC 3053 IPv6 Tunnel Broker, IETF, http://www.ietf.org/rfc/rfc3053, 2001.

3. Eclipse tool platform, http://www.eclipse.org.

4. Gilligan R. and Nordmark E., Transition Mechanisms for IPv6 Hosts and Routers, IETF RFC1933, http://www.ietf/rfc/rfc1933 , 1996.

5. Harkins D. and Carrel D., RFC 2409 The Internet Key Exchange, IETF, http://www.ietf.org/rfc/rfc2409, 1998.

6. Java Card Platform Specification, Sun Microsystems, http://java.sun.com/products/javacard/specs.html, 2003.

7. JCOP embedded secure software, IBM, http://www.zurich.ibm.com/jcop/.

8. KAME Project, http://www.kame.net.

9. Kent S. and Atkinson R., RFC 2401 Security Architecture for the Internet Protocol, IETF, http://www.ietf.org/rfc/rfc2401, 1998.

10. Kent S. and Atkinson R., RFC 2402 IP Authentication Header, IETF, http://www.ietf.org/rfc/rfc2402 , 1998.

11. Kent S. and Atkinson R., RFC 2406 IP Encapsulating Security Payload (ESP), IETF, http://www.ietf.org/rfc/rfc2406, 1998.

12. Maughan D., Schertler M., Schneider M., and Turner J., RFC 2408 Internet Security Association and Key Management, IETF, http://www.ietf.org/rfc/rfc2408 , 1998.

13. OpenCA project, http://www.openca.org.

14. Public Key Infrastructure Specification, Object Management Group (OMG), http://www.omg.org/issues/, Feb 2001.

15. Shacham A., Monsour B., Pereira R., and Thomas M., RFC 3173 IP Payload Compression Protocol, IETF, http://www.ietf.org/rfc/rfc3173, 2001.

16. Skarmeta A., Perez G., Reverte O., and Millan G., "PKI Service for IPv6," Internet Computing, IEEE Computer Society, May・June, 2003.

17. Stallings William. Cryptography and Network Security: Principles and Practice,3rd edition,Prentice. Hall,2003.

18. Stingson Douglas R., Cryptography: theory and practice , Boca Raton : Chapman & Hall, 2002.

19. Tsukamoto K., Matsushima M., Matsuki K., and Takano Y., "An Experimental Study on IPSec," The Institute of Electronics Information and Communication Engineers (IEICE) Transactions on Fundamentals, VOL.E85-A, NO.1, Jan 2002.

20. 周利欽、翁御舜譯，Zhiqun Chen著，智慧卡技術實務 - 使用Java Card，2002 年 4 月，初版。台北市：碁峰資訊股份有限公司。

21. 張瑞雄、陳俊良、陳彥文、趙涵捷、賴威光、賴溪松、陳錦洲、陳懷恩， IPv6 新世代網際網路協定暨整合技術，2004 年 7 月，台北：旗標出版有限股份公司。

22. 賴溪松、韓亮、張真誠，近代密碼學及其應用，2003 年，初版，台北：旗標出版股份有限公司。