東海大學
資訊工程與科學研究所

碩士論文

UDIDT 下建構於安全機制之入侵追蹤系統

Intrusion Traceback System on Security Mechanisms under UDIDT

指導教授：呂芳懌博士

研 究 生：洪 嘉 鴻

中華民國九十二年七月

致謝

本論文能順利完成，要感謝很多人，包括那陪我一路走來於背後支持我的長輩和朋友。首先，要感謝我的父母，於求學過程中不斷地給予我鼓勵與關懷。再者，要感謝指導教授呂芳懌老師於求學上給予很大的啟發與教導，並指引我正確的學習態度與方向。亦感謝口試委員林宜隆、連志誠、張文貴、林熙禎等教授能於繁忙之餘，不辭勞苦撥亢前來指導論文，予以我很多很寶貴的意見。

另外，也要感謝好友坤義及實驗室的義樺學長、政瑋學長、俊維學長、真真、仁傑、清健及耀中、國煒、子逸、品璁等學弟等，於學業上給予指點和豐富了研究所的兩年生活，沒有大家相伴與鼓勵，要走過那奮鬥的歲月勢必更加辛苦。

最後，慶幸能於東海這個美麗做研究，有幸認識許許多多相扶持的好友，並陪我一路走來度過很多的挫折與難關，終能在此留下美好的回憶。願在此與我的父母、師長及所有好友分享我的榮耀，也祝福他們永遠都健康快樂。

摘要

就目前的網路安全機制而言，例如，Firewall、IDS，對於攻擊者的入侵行為只有警示作用，沒有嚇阻效果。事實上，惟有找到攻擊者，訴諸法律，才能有效地嚇阻攻擊事件的發生。

一個完善的區域防禦機制應該包含入侵偵測與追蹤系統，藉由前者，可偵測出攻擊的行為，再以後者進行追蹤，俾迅速地找到攻擊者，切斷攻擊來源，以保障區域內的網路安全。

在本論文中，將探討區域防禦機制內的入侵追蹤系統。我們所提出的入侵追蹤機制，能適用於目前的網路環境。方法是將龐大的網路環境分為多個網路管理區域，以方便追蹤管理。追蹤系統透過各區域間的相互合作以追蹤攻擊者，而彼此間的追蹤是依據事先記錄在各區域以雜湊函數產生的識別值。

追蹤系統必須架構在一安全的環境上，以免遭受攻擊。本研究是透過 CA（Certification Authority）與 SSL（Secure Socket Layer）等安全機制，在追蹤系統各單元相互通訊時，保障其身分的識別、傳送的訊息與資料本身的安全性。使整個追蹤機制，能夠快速、正確且安全地找出攻擊者，並保障系統的堅固性，以免遭受攻擊者的破壞而無法正常的運作。

關鍵字：網路安全、入侵偵測、入侵追蹤、識別值、認證

Abstract

Currently security mechanisms, such as Firewall, Intrusion Detection System, only focus on caution, prevention and detection. In order to prevent an information system from illegal attacks, finding and punishing malevolent hackers are perhaps the most effective ways.

A perfect section defense mechanism should include intrusion detection system and intrusion traceback system. When receiving an alert from intrusion detection system, it can trace the intruder by intrusion traceback system. As an intruder is found, the section defense mechanism will cut off the network connection in order to protect the section.

In this paper, we design and construct the intrusion traceback system of the section defense mechanism, which is applicable to current environment of a network system. In this research, we divide a network system into many network management unit (NMU) for tracing intruder and convenient management purpose. Each NMU cooperatively trace the intruder with one another based on the identification code produced by hash function and pre-recorded in each section.

Intrusion traceback system needs a secure environment to perform its tracing. CA (Certification Authority) and SSL (Secure Socket Layer) are those mechanisms to guarantee safe authentication and confidentiality in each NMU. Under such an environment, the intruder can be quickly and correctly found. Of course, the system will be robust enough to protect itself from hackers and intruders.

keyword：Network Security, Intrusion Detection, Intrusion Traceback, identification code, authentication.

目錄

致謝 ..... **I**

摘要 ..... **II**

ABSTRACT ..... **III**

目錄 ..... **IV**

圖目錄 ..... **VI**

表目錄 ..... **VII**

第 1 章　前言 **1**

1.1　背景 1
1.2　研究動機 1
1.3　研究方法 2
1.4　論文架構 5

第 2 章　文獻探討 **6**

2.1　攻擊模式 6
2.2　入侵追蹤系統 10
2.3　入侵偵測系統 12

第 3 章　系統架構 **14**

3.1　系統架構 14
3.2　系統元件 16
3.2.1.　TM 各元件 *16*
3.2.2.　LM 各元件 *17*
3.2.3.　TA 各元件 *18*
3.2.4.　系統分工 *19*
3.3　追蹤方法 19
3.4　追蹤方法比較 22

第 4 章　系統安全 **25**

4.1　認證系統 25
4.1.1　公開金鑰基礎建設 *25*
4.1.2　電子憑證 *27*
4.1.3　認證系統架構 *30*
4.2　資料保密 32

4.2.1 SSL 機制 32
4.2.2 追蹤流程保密 34
第 5 章 追蹤系統機制 36
5.1 流量導入 36
5.2 封包處理 37
5.3 入侵偵測與資訊儲存 43
5.4 追蹤管理 44
5.5 訊息協定 47
第 6 章 系統實作 51
6.1 實驗環境 51
6.2 系統架構 52
6.3 實驗結果 54
第 7 章 結論 59
7.1 系統比較 59
7.2 結論與未來發展 60
參考文獻 63

圖目錄

圖 1-1 系統追蹤流程....4
圖 3-1 UDIDT 架構圖....15
圖 3-2 多個 Port 的 Backbone Router 之 LM 架構....16
圖 3-3 追蹤方法....20
圖 3-4 追蹤流程....21
圖 4-1 公開金鑰基礎建設....26
圖 4-2 X.509 憑證規格....28
圖 4-3 CA 簽證過程....29
圖 4-4 驗證憑證過程....29
圖 4-5 TA 之架構....30
圖 4-6 TA 之階層式架構....32
圖 4-7 SSL 的安全協定....34
圖 5-1 簡化的 NMU 架構....36
圖 5-2 具 Mirror Port 交換器結構....37
圖 5-3 HP 結構....38
圖 5-4 Ethernet 標頭格式....39
圖 5-5 IP 標頭格式....39
圖 5-6 白色部份為 IP 標頭追蹤資訊....40
圖 5-7 Trace Record....41
圖 5-8 HP 處理流程....42
圖 5-9 IDP 架構....43
圖 5-10 RS 架構圖....44
圖 5-11 TM 各元件....45
圖 6-1 實驗環境圖....52
圖 6-2 HP 實作架構圖....53
圖 6-3 TM 架構圖....53
圖 6-4 模擬封包傳送....54
圖 6-5 NMU2 的識別值....55
圖 6-6 NMU3 的識別值....55
圖 6-7 NMU4 的識別值....55
圖 6-8 No IP Spoofing 追蹤結果....56
圖 6-9 IP Spoofing 追蹤結果....57
圖 6-10 資料庫資料量對追蹤時間的影響....58
圖 6-11 Hop 數與資料量對追蹤時間的影響....58

表目錄

表 2-1　各個追蹤方法 10
表 3-1　UDIDT 特色 23
表 5-1　Detect Record 各欄位 42
表 5-2　追蹤協定的訊息格式 49
表 5-3　查詢協定的訊息格式 50
表 5-4　同步協定的訊息格式 50
表 6-1　實驗平台 51
表 7-1　AMN 與 UDIDT 比較 59

第 1 章 前言

1.1 背景

近年來，由於網路的快速發展，越來越多的商業活動都透過網際網路來進行，若無比較安全的機制，連上網路的使用者、主機或提供網路服務的伺服器，無疑都將直接暴露於網路上，對於駭客而言這是一種極大的誘惑。駭客攻擊的目的，早期是以獲得大型主機的使用權帳號為主，現今的入侵，則以資料竊取與破壞性攻擊為大宗，駭客攻擊的威脅也因此日益嚴重。造成攻擊行為氾濫的的重要因素之一是駭客工具易於取得，許多的駭客網站均提供自動化入侵工具與入侵教學供下載。攻擊者甚至不須具備足夠的系統或網路等專業知識，就能利用自動化工具發動攻擊，甚至可以聯合若干主機一起發動大規模的攻擊行動。駭客的攻擊行為往往造成個人或企業難以估計的損失，Yahoo、Amazon、EBay 等網站於 2000 年遭受 DoS 攻擊的事件，損失都是上億元以上。

安全的觀念逐漸受到重視後，防禦的機制，例如，防火牆（Firewall），開始被用於防範惡意攻擊者，其原則是以存取控制，限制使用者使用網路資源，做法從簡單的阻擋 port 的封包出入、限制某些 IP 位址的存取到複雜的訂定存取規則等，視管理上的需求而定。然而，防火牆的機制似乎不夠完備，所能阻擋的攻擊有限，有經驗的駭客更可輕易的穿透並進行攻擊。因而入侵偵測系統（Intrusion Detection System, IDS）的研究也逐漸展開，IDS 係偵測異常行為或攻擊行為之特徵，並給予入侵警示，即時的 IDS 常將結果以 E-mail 或簡訊回覆給管理者，提醒其留意。相較於防火牆，IDS 能知道遭受何種攻擊，唯不能如防火牆般做存取控制，所以，常與防火牆搭配，共同組成一道入侵的防線。

1.2 研究動機

目前已發展的入侵防禦機制，都是扮演被動式的防禦與警示角色，對於攻擊的駭客而言，只是增加攻擊的難度，並未有效的根絕攻擊的來源，而具備主動反制與追蹤入侵者的追蹤系統，能實際追蹤到入侵者，而將之繩之以法，或訴諸道德規範，才是真正嚇阻駭客的不二法門。

入侵追蹤係指利用一些資訊作逆向反查，以追查攻擊封包來源的行為。入侵追蹤最大的困難在於駭客會利用 TCP/IP 協定制定時，未限定封包標頭不能更改的缺點，逕行偽造封包的來源 IP 位址來隱藏自己。若追蹤系統未能分辨 IP 的真偽，則可能為駭客所所誤導，而降低追蹤結果的可信度。

然而，追蹤的機制不可能是完美無缺，有經驗的攻擊者在將來也有可能找到方法來規避追蹤，但是對於經驗不足或只會使用駭客工具的攻擊者而言，至少可以讓他們不輕易去嘗試攻擊的行為。因此，追蹤機制的功能除了希望對網路攻擊具有反制力之外，在平時也能有效的嚇阻駭客攻擊事件的發生。

追蹤系統在機制上應該搭配一入侵偵測系統，否則，只能是在攻擊事件發生後，人為的方式進行申訴再行追蹤，如此，往往會因為相隔時間過長，而失去追蹤時效，甚至是追蹤系統裡有效的資訊因時間久遠而被新的資料所取代，而無法執行追蹤任務。況且，受害者往往無法準確地評估實際遭受攻擊的時間。在時間與目標均不明確的情形下，花費追蹤上的時間將更長。因此，如能有一個即時有效率且正確地執行追蹤任務的追蹤機制，搭配一個能有效偵測入侵行為的偵測系統，必然為一極佳的組合。偵測系統能在入侵者來犯時，提供正確的警示，並立即通知系統追蹤之。本研究為整體研究的一部分，重點在於設計一相容於目前網路架構的追蹤機制，並搭配另一位研究人員所研製的入侵偵測系統，兩者各司其職卻共享相同的資訊與資源，減少資訊的重複蒐集，亦能達到即時追蹤的效果，並互補雙方在功能上之不足。

1.3 研究方法

要追蹤入侵者，勢必透過犯罪者遺留的證據及經過的痕跡。本論文所提出的方法是透過「刻意製造的痕跡」來達到反向追蹤的目的。這個「痕跡」就是各區域所記錄的：哪些封包行經過該區域。各封包在各區域記錄下來的資訊必須一致，才能分辨是同一封包的「痕跡」，也才能反向追蹤之。而各區域間的相互協助是必要的，本研究是以畫分區域的方式分散所要記錄封包的資訊量，並增加管理的效率。追蹤系統係由偵側系統來啟動，在偵測失敗時，方由網管人員向追蹤系統發出追蹤請求。整個流程如圖 1-1 所示，偵測系統或管理人員發出追蹤要求後，附近的區域接到此要求，會先判斷攻擊封包是否經過該區域。此時有兩種情況，若否，則回應追蹤結果給追蹤發起者；若是，則判斷是

否由本區域發出。相同的，此時也有兩種情形。若封包由本區域發出，表示追蹤完成，而必須將追查到的駭客主機相關資訊通知該網路之管理者，並回應追蹤完成訊息給追蹤發起者。若否，則回到前面步驟，繼續往外層詢問，直到追蹤到攻擊者。因為本系統需透過不斷的向駭客方向傳送追蹤資訊，為了增加查詢的效率，我們以雜湊函數將追蹤資訊轉換為雜湊值，以降低資料量。

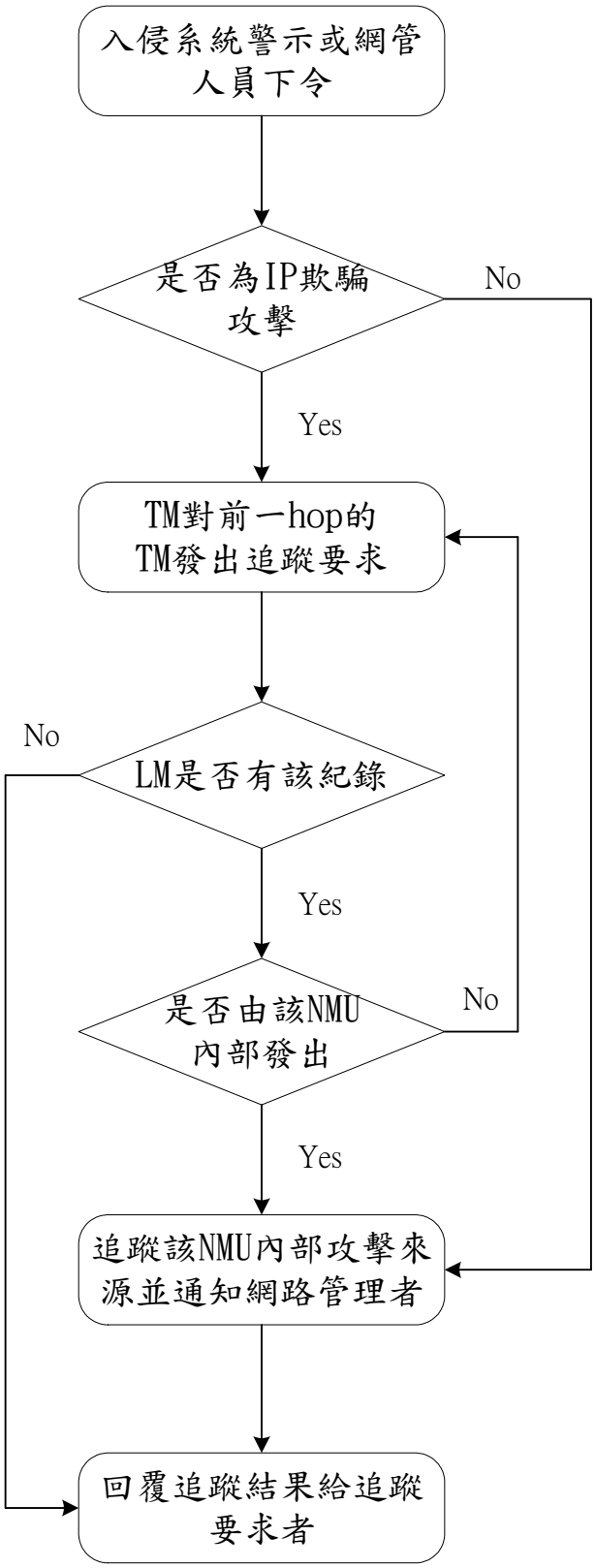
入侵系統警示或網管人員下令
是否為IP欺騙攻擊
No
Yes
TM對前一hop的TM發出追蹤要求
No
LM是否有該紀錄
Yes
是否由該NMU內部發出
No
Yes
追蹤該NMU內部攻擊來源並通知網路管理者
回覆追蹤結果給追蹤要求者

圖 1-1　系統追蹤流程

1.4 論文架構

本論文第二章是文獻探討，列出常見的攻擊模式、相關的入侵追蹤方法和入侵偵測系統類型；第三章則介紹本論文提出的追蹤系統 UDIDT 之架構，並敘述其追蹤流程；第四章敘述如何引入認證機制，以增加區域間訊息的可信度，並陳述認證機制在追蹤上所扮演的角色；追蹤系統各個元件之細部結構，及各元件間訊息傳遞的協定，則在第五章中陳述；第六章總結本論文，並提出未來的改良機制及入侵追蹤的展望與發展。

第 2 章 文獻探討

2.1 攻擊模式

隨著網路技術的發展，攻擊者所發動的攻擊模式也不斷翻新。設計一入侵偵測與追蹤機制的首要考量，是攻擊者可能發動哪些攻擊，其攻擊模式為何[14-17,25]。系統設計時則必須考量這些攻擊行為是否能被有效地偵測，又該如何追蹤。一般而言，攻擊可分為：單一封包攻擊、分段式攻擊及超載（overload）攻擊等三種模式。以下則分別介紹之：

1. 單一封包攻擊

攻擊者所發出的單一封包中蘊含攻擊行為之謂，這類攻擊包括：CGI（Common Gateway Interface）漏洞、緩衝區溢位、UniCode 漏洞…等，通常是利用系統漏洞來癱瘓受害主機或藉此取得系統的權限。以下簡述常見的單一封包攻擊：

(1) CGI 漏洞：CGI 是外部應用程式與 WEB 伺服器溝通的一標準介面，大部分的網站伺服器包括微軟的 IIS 與 Apache 都有支援。CGI 的安全漏洞有很多，以下僅列舉幾項常見的模式：

- 不正常表單數據：為 CGI 最常見的問題，攻擊者提交一些不正常的表單資料，當伺服器在解譯這些資料時，就可能發生意想不到的結果，例如，使用者回應給伺服器的表單含有不是伺服器提供的選項資料，伺服器不知如何處理。
- ../../問題：在路徑中單個點（.）和兩個點（..）的序列分別代表”目前的目錄”和”目前目錄之父目錄”。攻擊者可藉此存取受害者主機的檔案，例如，攻擊者可以建立像”../../../../../etc/passwd”的串列，就可以進入”etc”目錄去獲取 passwd 檔。
- 處理檔名的問題：提交的檔名如果包含路徑，例如”/etc/passwd”或”c:\WINNT\SYSTEM32\KRNL32.DLL”，則可藉由伺服器在處理該檔名

時，存取到該路徑之檔案。因此，若未小心處理此問題，則攻擊者可藉此漏洞存取系統中的任意檔案。

(2) 緩衝區溢位：係利用程式設計缺陷製造緩衝區溢位，來獲取系統權限。一般的作法是將規模大於緩衝區大小的資料放入程式變數中，緩衝區溢位後，系統會以溢位後執行程式的使用者身份（通常是 root）來執行攻擊者所佈設的惡意程式。常見的攻擊方式有：

- 攻擊者利用修改過的 ping 指令，將過長的資料送到某台主機後，若接收端主機並未對過長的資料做適當處理，而發生緩衝區溢位，結果是造成系統主機的癱瘓，此即為常見的 Ping of death 攻擊。
- 攻擊者可發送一個長度超過 65535 位元組的資料，一般而言，網路的封包大多小於 65535 位元組。因此，當攻擊者發送一個長度超過 65535 位元組的資料，伺服主機收到後會在緩衝區裡重新組合它們，若緩衝區小於該封包，則在重組該封包時可能會造成緩衝區溢位，而覆蓋到其它部份的資料或程式，如此，就有可能造成伺服器主機發生錯誤而當機。

(3) Unicode 漏洞：統一編碼（Unicode）的功能跨越了平台、程式、以及語言的藩籬，將所有的字及字體都統一編碼，例如「/」符號，若用統一編碼來表示即變成冗長的「%2f」。然而因為 IIS（Internet Information Server）的安檢功能無法辨識過長的名稱，所以 URL 上過長的統一編碼，便可繞過 IIS 的安全檢測。例如，http://x.x.x.x/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+type+c:\A.txt，即可利用 type 指令印出主機"X.X.X.X"之文字檔 A 的內容。相同的，只要變更「c+type+c:\A.txt」此路徑內容，即可任意的存取遠端主機資料。

(4) 連線劫持（Connection hijack）：是利用 TCP/IP 協定本身的疏失，劫持原本已經連接的連線。例如，TCP 的 Three -way handshaking 就存在序號(Sequence number)容易被猜測的弱點。駭客可以假冒連線的建立者，趁機下了一個 r 系列（用於遠端登錄、存取）的指令，把自己的系統設定為信任主機之一。攻擊者 H 在 A 和 B 兩主機已建立的連線中，對 A 主機將自己偽裝成 B 主機，對 B 主機則將自己偽裝成 A 主機，趁機竊取連線所傳送之內容，這種攻擊稱為 Man-in-the-middle

攻擊，為一種連線劫持攻擊。

2. 分段式攻擊

利用分段（Fragment）的封包進行攻擊之謂， Teardrop 和 Jolt 為兩個典型的例子，此類攻擊是利用 IP 層封包重組演算法的漏洞。

(1) Teardrop 攻擊：在 IP 層中封包分割和重組的原則是：分割後的封包大小必須小於傳輸介面的最大傳送單位（MTU, Maximum Transfer Unit），並且符合以 8 byte 為單位的倍數。當大封包在網路上傳輸時，由於 MTU 的限制，使得該封包需切割成數個小封包，再以一個個互相接續的方式傳入目標主機，由該主機的 IP 層將其重組回原資料段。Teardrop 攻擊就是使得有些片段位置重疊，刻意製造不正常封包序列，而造成某些作業系統，例如舊版的 Linux 系統，或是 WinNT/95，因此無法處理而當機或暫停服務。

(2) Jolt2 攻擊：亦是利用封包分段進行的攻擊。攻擊者利用駭客程式 Jolt2，以 ICMP 或 UDP 協定不斷地發送大位移（offset）封包分段，使得 IP 層重組該封包時，發現其超過 IP 封包的最長限制 65535 位元組，因而使得某些作業系統無法處理而當機或暫停服務。例如，未安裝 Service pack 的 Win2000 系統，系統在經由此攻擊後，會浪費大量資源處理攻擊封包，往往使得 CPU 的使用率過高，甚至導致系統當機。

3. 超載攻擊

攻擊者採用的攻擊手段是讓資源耗竭，而無法服務其他合法的使用者，稱為超載攻擊，也就是常見的 DoS 或 DDoS 攻擊。超載攻擊可分為兩種模式，一種是耗竭主機系統資源，另一種則是耗盡網路頻寬。以下分述之：

(1) 耗竭系統資源：此類攻擊是去消耗 CPU、記憶體、檔案系統配額等系統資源。在這些資源耗竭時，系統可能會引發當機、檔案系統飽合或行程（process）被懸宕。常見的手法有：

- TCP SYN Flood 攻擊：當建立一個 TCP 連線要時，必須完成 Three-way handshaking。攻擊者可從主機 A 送出一個含假冒來源位址 C 的 SYN 封包

給受害主機 B 的某一埠 P，如果 B 是存活的(Alive)， 則 B 會試圖送 SYN/ACK 封包給 C。若 C 是一存活的主機，則會發出一 RST 封包給 A 告知未打算建立此連線，否則，B 永遠收不到 C 的回應，這個連線要求就會一直保持在 SYN_RECV 狀態，並被放入連線佇列（Connection queue），等時間終了（Timeout）才終止這個連線要求。等候時間最短 75 秒，最長可達 23 分鐘，攻擊者只要 10 秒鐘內連續送出一群 SYN 封包，即可塞滿 A 在 P 的連線佇列，使得 A 經由 P 所提供的服務，例如，HTTP 及 SMTP…等暫時停止。

- Land 攻擊：是利用作業系統對封包處理時的漏洞所進行的攻擊。攻擊者向受害端 B 的埠 P 發送來源及目的 IP 位址均為 B 的大量封包，B 若不知如何處理自己送給自己的封包，在解譯時會佔用大量系統資源，而導致該類服務甚至整個系統癱瘓。舊版本的系統未對這項漏洞進行妥善處理，則可能遭受此攻擊。

(2) 耗盡網路頻寬：此類攻擊是設法塞爆網路頻寬，使得要求服務或服務封包無法順利抵達目的地，當然，使用者亦無法獲得應有的服務。通常的手法是利用大頻寬的系統去消耗小頻寬的系統資源，亦可同時發動若干較小頻寬以分散式的方式佔據較大頻寬的整個頻寬。常見的手法有：

- UDP Flood 攻擊：UDP Flood DoS Attack 又稱為 Fraggle 攻擊，攻擊者不斷送出假造來源 IP（例如，F）的 UDP 的廣播（Broadcast）封包（目標 port 為 7，表回應（Echo）服務）至目標網域，當該網域內的存活的主機以 UDP 回應封包回應給 F 後，會產生大量的流量，而造成 F 主機網路資源的耗竭甚至造成該區段網路的壅塞。即使某些 IP 位址沒有回應，但因路由器會為之回應 ICMP 封包（type 3, Destination Unreachable），仍可以達到 DoS 攻擊的效果。

- Smurf 攻擊：為 ICMP Flood 的一種攻擊形式，手法類似 UDP Flood。攻擊者會先假冒目標主機 F 向路由器發出廣播 IP 的 ICMP Echo Request 封包，路由器則對該區域網段內的所有主機發出相同的 ICMP 封包，所有的主機收到此訊息之後，會對 F 送出 ICMP Echo Reply 回應。所有的 ICMP 封包

在極短的時間內湧向 F，不但造成網路壅塞，更會使 F 因為無法反應如此多的系統 interrupt 而當機或暫停服務。若長時間一直有此攻擊封包進入該網段，則所有主機，甚至包括，路由器，都會因為網路壅塞而無法提供服務。

2.2 入侵追蹤系統

近年來所發表的入侵追蹤方法[1-7]，依其追蹤模式大約可分為主動式和被動式兩種類型，如表 2-1。主動式者在封包傳送過程即已將追蹤訊息基植於網路機制中，當有入侵行為，只須將收集之封包或訊息經過組合，即可找出此攻擊來源。被動式不須改變網路機制，至多只做紀錄，而在攻擊發生後，以事前紀錄之資訊追蹤到攻擊來源。

表 2-1 各個追蹤方法

主動式入侵追蹤	被動式入侵追蹤
IP Marking	Hop by hop
ICMP Message	Digests
	AMN

1. IP Marking

本方法由 S. Savage 和 D. Wetheral 等[1-2]於 2001 年所提出，其方法是以目前 IP 協定為基礎，在路由器中加入一標記（Marking），使得封包經過每個路由器 R 時，R 會以一固定的機率，決定是否標記某一段行經路徑（R 與下一路由器之間），並用 XOR 方式壓縮，而紀錄於 IP 標頭的 Identification 欄位內。採用隨機標記是為了減少路由器沉重的負擔，其缺點則是，若要重整完整的攻擊路徑則需蒐集相當多的封包標記方得以完成。因此，IP Marking 的方法只適於追蹤以大量封包所進行的攻擊的情形，例如，DoS 及 DDoS 等。

2. ICMP Message

是由 B.M. Leech 和 T. Taylo[3]於 2001 年所提出，利用路由器轉送封包 P 時，產生

另一與其目標 IP 相同的 ICMP 訊息封包 C，傳送到相同的目的端，而 C 的內容有 P 的部份資訊。ICMP Message 亦採取一個固定機率（1/20000）來產生該訊息，此方法如同 IP Marking 一般，必須要有足夠的封包，亦只能追蹤 DoS 及 DDoS 等大量封包的攻擊。

3. Hop by Hop

由 R. Stone[4]於 2000 年所提出，是由距離被害主機最近的區域網路骨幹路由器（Backbone router, BR），判斷攻擊封包是由哪些鄰近的路由器轉送過來，決定出這些相鄰的路由器 R 後，再由 R 以相同的方式找到更外層的轉送路由器，如此遞迴地執行，一直追蹤到網路邊緣，或定義出攻擊封包的可能路徑方止。本方法必須在每個路由器中紀錄各轉送封包之特徵，如此，將會造成路由器相當大的負載，而且目前亦無統一規格之路由器紀錄檔，對於 DDoS 等分散式地大規模攻擊時，難以分辨出多條入侵路徑。

4. Digests

此方法以雜湊碼（Digests）當做追蹤的基礎之機制，由 A. C. Snoeren 和 C. Partridge 等[5]於 2002 年所提出。在各骨幹路由器設置一個 DGA（Data Generation Agent），負責將經過的每個封包經由雜湊函數（Hash function）轉成一筆筆的 Digests，而 SCAR（SPIE Collection And Reduction Agent）則位於每個區域內，負責蒐集區域內各個 DGA 的資訊。當攻擊發生時，SCAR 會依照所控管的 DGA 產生出攻擊路徑的一部分，整個追蹤機制由一個 STM（SPIE Traceback Manager）掌控，負責將各 SCAR 所追蹤的部分攻擊路徑整合為一完整攻擊路徑，並藉此追蹤到攻擊者。

5. AMN

是由 T. Baba 和 S. Matsuda[6-7]於 2002 所提出，作者將整個 Internet 大區域切分成許多稱為 AMN（Autonomous Management Network）的小區域。各個 AMN 之追蹤元件包括 Sensor、Monitoring manager（MM）及 Tracer 三部份。Sensor 之功能如網路式入侵追蹤系統（NIDS, Network-based IDS）能偵測網路的攻擊行為。Tracer 會將經過該 AMN 之封包的一些特徵紀錄下來。MM 是區域追蹤系統的管轄者，當接到 Sensor 的入侵警示，即將此特徵送交給鄰近 AMN 區域的 MM，臨近區域之 MM 會與其 Tracer 中的資料相比對，若有此資料則重複此動作直到追蹤到攻擊者區域。

2.3 入侵偵測系統

入侵偵測系統若依偵測對象，可分為主機型和網路型兩類：

1. 主機型入侵偵測系統（HIDS）：因偵測該 IDS 所在之主機 S 是否遭受攻擊而得名，偵測對象則是作業系統日誌或應用程式日誌[8,18]，亦可監視系統呼叫（System Call）[19]，以獲得目前程式的執行狀況，或觀察 S 是否執行了不正常的程式，或被植入木馬程式。HIDS 安裝於主機上，比較能和作業系統密切整合，一般均依據安全稽核政策對作業系統之使用者管理、檔案存取權限、系統核心運作、及應用程式執行狀況，提供安全防護與即時回應，因此可以提供較直接的稽核機制。然而，因安裝於主機上，對系統的效能有很大的影響，通常只限於監測主機本身的安全。

2. 網路型入侵偵測系統（NIDS）：以網路上流動的封包為偵策對象[8,18]，判斷是否有攻擊行為。由於網路上的封包都遵循共同的通訊協定，因此不會如主機型因為電腦平台或作業系統不同而使應用程式或管理介面有所差異。NIDS 會拷貝及分析封包，並與資料庫中事先建立的攻擊特徵資料做比對，如果符合攻擊特徵，則向網路管理者發出警示。NIDS 的限制是必須面對大量的網路封包，資料處理若未即時，則會有資料遺漏的情形，相對於 HIDS 也會有較多的誤判。

若依偵測技術則可分為誤用偵測與異常偵測兩類：

1. 誤用偵測（Misuse detection）：所採取的偵測方式為特徵比對，將首先分析各種攻擊行為，並歸納出其特徵（Signature），以為封包比對的對象[9]。其比對的方式又分為：

(1) 狀態轉移（State transition）：運用有限狀態機圖形（finite automata）來描述入侵模型。唯須先將攻擊特徵以有限狀態機表示，比對到某一攻擊特徵後，將狀態轉移到下一狀態，作更進一步的特徵比對，最後若能符合終止條件（到達終止狀態），則判定為某一攻擊。

(2) 規則為基礎（Rule-based）：以規則的方式定義攻擊模式，擁有較高的偵測率，但須制定完整的攻擊規則資料庫。缺點是比對相當耗時，若要即時偵測，需耗用大量系

統資源。

2. 異常偵測（Anomaly detection）：此偵測模式會先建立使用者正常的行為資料檔（profiles）[10-11]，例如，CPU 使用率、記憶體使用率等，此檔案可由人爲定義，也可觀察、統計及歸納該系統之一般行為而獲得，以統計的方式取得。偵測系統會利用行為資料檔與使用者行為作比較，俾指出不符合正常行為的「異常行為」。採用這種方式必須確定使用者的行為模式在短時間不會改變，否則常會有誤判的情形。

第 3 章 系統架構

3.1 系統架構

本論文提出 UDIDT（Union Defense of Intrusion Detection and Traceback System）[20]，其架構如圖 3-1 所示，由入侵偵測機制與追蹤機制所組成。入侵偵測單元負責比對警示，入侵追蹤單元則擔任追蹤至攻擊來源的工作。若將整個 Internet 視為一追蹤系統，其在追蹤管理上勢必因範圍太大而有所困難，為了降低追蹤複雜度，本研究將網路區分為許多 NMU（Network Management Unit）單位。NMU 係一個網路管理單位，可為校園或企業網路等具區域性質的個體，各 NMU 均建制一 UDIDT 系統。

一個 UDIDT 包含有“追蹤管理者”（Traceback Manager , TM）和“區域網路管理者”（LAN Manager , LM）兩部份。依所扮演的角色而言，TM 是各 UDIDT 內追蹤系統的管理元件，負責處理所有追蹤要求（Trace Request），和輔助系統其它元件的運作，並藉由事先記錄在 LM 的資訊與其它 NMU 之 TM 協助，共同追蹤攻擊來源。LM 則負責處理、偵測及儲存經過 NMU 的封包資訊，適時地提供 TM 追蹤之用。

依所在位置來分， TM 位於 NMU 內部網路內，若有內部防火牆（Firewall），則置於防火牆的非軍事區（DMZ, De-Militarized Zone），並在防火牆的存取控制下，接受其它 TM 之追蹤要求。LM 位於骨幹路由器（BR, Backbone Router）前端所設置的交換器（Switch）S，其功能為記錄經過 BR 的流量，包含：進、出 NMU 和經由 BR 轉送到其它 NMU 區域之 BR 的封包。S 若置於 BR 後端，則只能記錄進、出 NMU 封包，會遺漏經由 BP 轉送的封包，導致追蹤資訊不完整而無法建立聯防的機制。且 S 必須具備 Mirror Port 的功能，方能將封包流量複製並導入 LM 區域內。

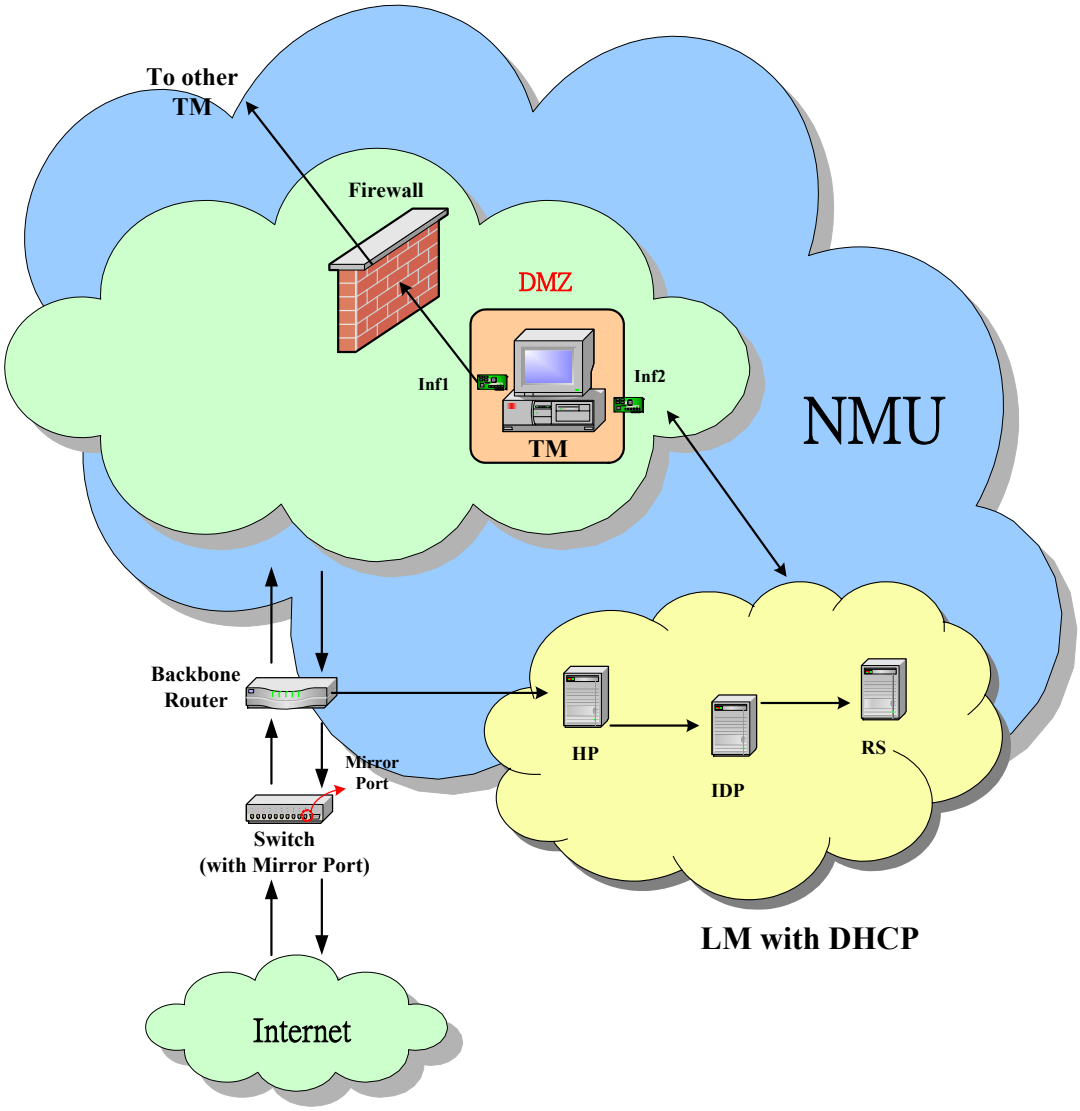

圖 3-1　UDIDT 架構圖

LM 之各元件之間以及與 TM 之溝通，均是以 DHCP（Dynamic Host Configuration Protocol）方式配置。TM 具有兩個網路介面（Interface），分別為 Inf1 和 Inf2。Inf1 具有真實 IP 位址能與外部網路溝通， Inf1 也是與其它 NMU 的 TM 交換訊息的管道。Inf2 則是 DHCP 伺服器的介面，會分配給 HP、IDP 和 RS 等主機虛擬 IP 位址，是為 DHCP 的客戶端。以 DHCP 配制的優點是外部網路使用者並不能直接存取 LM 的資料，以保障資料的安全。

一般而言，一個 NMU 會連接多個 NMU，而採用多個 port 的骨幹路由器，每一個 port 必須各自連接一個具 Mirror Port 的交換器，見圖 3-2，再由各自的 HP 處理和 BP 偵

側後，彙集到 RS 做緩衝區偵測與寫入資料庫，緩衝在同一時間或短暫的時間內大量封包湧入資料庫中，俾調節資料庫的交易處理。

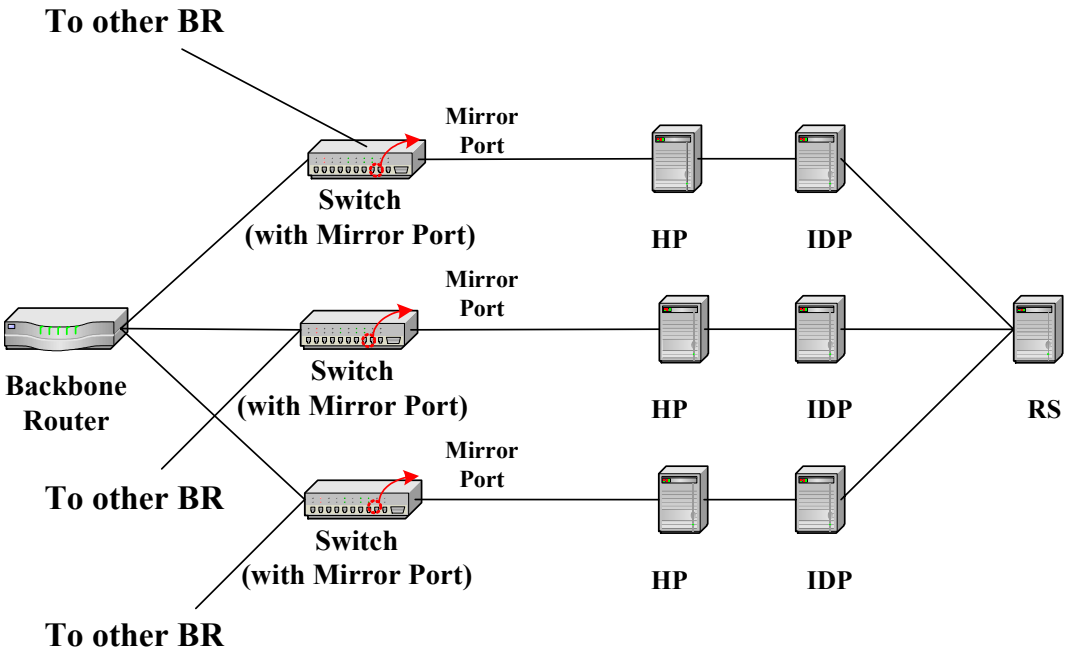

圖 3-2　多個 Port 的 Backbone Router 之 LM 架構

3.2 系統元件

UDIDT 的主要架構由 TM、LM 和認證機制 TA（Trace Authority）所構成，分述如下：

3.2.1. TM 各元件

基本上，一個 NMU 的 TM 由 ARP（Address Resolution Protocol） Table Set（ATS）、SSL 機制、DHCP 服務、NTP（Network Time Protocol）服務與追蹤代理人（Trace Agent）所組成。

1. ATS

ATS 主要提供內部 NMU 的 IP 與 MAC 位址的對應，當 TM 需追蹤內部使用者時，

可藉由此資料得到對應的 IP 位址，並採取進一步的因應措失，例如，發 Email 警告該駭客，甚至找出當事人而訴諸法律。ATS 的資料取得有兩種方式，一是由 TM 定時地蒐集並儲存該 NMU 中各路由器的 ARP Table，提供過去各時間點的 ARP 查詢，避免舊的資料在 ARP Table 內容更新後遺失，而無法追蹤到入侵者。另一則由網管人員所提供，在申請某 IP 時即提供相對應的 MAC 位址。

2. SSL

SSL 可保障兩應用程式之間資料交換連線的私密性，和身份確認等的機密性、可靠性及安全性。SSL 也必須搭配認證機制來保障 TM 與 TM，以及 TM 與認證機制間資料與訊息傳遞的安全性。

3. DHCP

LM 採用 DHCP 協定，其 Client 端會在開機時自動向 DHCP Server 要求相關的資訊，其後 Client 自動設定 Server 所給予的虛擬 IP 位址，以避免 LM 中所處理及儲存之追蹤與偵策資訊遭受外部攻擊。LM 的 DHCP 服務實際上是由 TM 之一網路介面來提供。

4. NTP

TM 以 NTP 服務與最接近的 NTP 伺服器校準網路時間，並由 TM 校準 HP 的時間，以便能夠給予一致且正確的時間戳記。

5. 追蹤代理人

負責發出追蹤要求與處理來自各 NMU 的追蹤要求。追蹤代理人有自主式和人工式兩種介面。前者是在接獲偵測機制的追蹤要求時，自動進行追蹤的任務，必要時則向其他 TM 發出追蹤要求；後者是在系統實際遭到攻擊而偵測機制未能偵測出來時，網管人員藉此介面向 Trace Agent 提出追蹤要求。

3.2.2. LM 各元件

LM 負責（1）將封包資訊轉換成追蹤資訊並儲存，（2）偵測與發出自主式追蹤要求給 TM，（3）提供 TM 追蹤資訊。LM 內部包含有標頭處理器（Header Processor, HP）、即時偵測處理器（Immediate Detecting Processor, IDP）及後端伺服器（Rear Service, RS）

等元件。

1. 標頭處理器

處理經由 Mirror Port 送過來的封包。除了將封包中可識別的部份轉換成雜湊碼以供追蹤識別之外，並在封包中擷取資訊，提供偵測機制進行入侵偵測。再將兩者匯集為一筆封包資訊（Packet Record, PR），目的是便於送交給後端的 IDP。

2. 即時偵測處理器

將 HP 送過來的 PR 利用 IP MASK 機制，依每個 PR 的來源 IP 位址及目的 IP 位址判斷，是由外向內、由內向外與經由 BP 轉送的封包，再分別送入 In_DQ、Out_DQ 與 Foreign_Queue 三個佇列中。LM 會以即時偵測器（Immediate Detector）檢視 In_DQ 和 Qut_Queue 中之 PR，以偵測是否有攻擊的行為，Foreign_Queue 之 PR 則直接存入 RS，以待後續處理。

3. 後端伺服器

In_DQ、Out_DQ 與 Foreign_Queue 三佇列中之 PR 會分別彙入 In-bound Temporary Area（IB TA）、Out-bound Temporary Area（OB TA）和 Foreign Temporary Area（FTA）等緩衝區中，待緩衝區全滿後再分別寫入資料庫的 In-bound Table、Out-bound Table 與 Foreign Table 等三個資料表單（Table）中。緩衝區除了可緩衝同時有多筆資料寫入資料庫之外，而以輔助偵測器 Auxiliary Detector（AD）對其資料進行偵測 In_DQ 中偵測不到之由內對外之 DDoS 攻擊。在資料庫中則有一 Analyzer 對資料做關聯性分析比對。

3.2.3. TA 各元件

認證機制負責各 TM 間的認證與告知 TM 該封包來源之 BR 位於那一個 NMU 及其 TM 之 IP 位址。TA 包含有 TCA（Trace Certificate Authority）與 TQA（Trace Query Agent）兩元件。

1. TCA

TCA 是一 PKI（Public Key Infrastructure）機制，負責 TM 與 TM 間的認證，即認證各 TM 的身份，並接受各 NMU 的註冊。它採階層式的架構，上層也可以為下層做認證。

因此，所發的憑證有兩種，一種是為 TM 認證，另一種則替下層 TCA 認證。

2. TQA

TQA 負責蒐集各 NMU 向 TCA 註冊的部份資訊，包含：NMU 名稱、NMU 內 BR 的 MAC 位址與其內部 TM 之 IP 位址。也負責為各 TM 提供其它 NMU 中之 TM 的位址。

3.2.4. 系統分工

UDIDT 結構頗複雜，因此係與另一位研究生分工，本論文將探討 TM 各元件、LM 中的 HP 元件和 TA 各元件，主要任務則是建立追蹤系統，包括追蹤系統之架構及追蹤方法，再加上認證機制。而 UDIDT 中的偵測單元與其偵測方法則由另一同學在其論文中深入研究。

3.3 追蹤方法

UDIDT 的追蹤方法如圖 3-3 所示。當 TM 接受到追蹤的要求和追蹤資訊，包含來源 IP、來源 MAC 與封包識別值（雜湊值）時，會依一流程進行追蹤：

1. 首先，以來源 IP 進行追蹤，藉此查詢到其所在 NMU 之 TM 的 IP 位址，並對其發出追蹤要求。
2. 協助追蹤之 TM 接到追蹤要求後，查詢是否有該紀錄，藉此判斷是否為 IP 欺騙攻擊。
3. 若查詢到該記錄，則表示非 IP 欺騙攻擊，並往內部追查到該攻擊來源。
4. 若判定為 IP 欺騙攻擊，則回覆給追蹤發起者，並繼續以 MAC 進行追蹤。
5. 追蹤發起者接到協助追蹤之 TM 的通知，認定為一偽造 IP 的攻擊，則以來源 MAC 查詢到其來源所在 NMU 之 TM 的 IP 位址，並對其發出追蹤要求。
6. 協助追蹤之 TM 接到追蹤要求後，查詢是否有該紀錄，若是，表示經過該 NMU 之封包，則判斷是否由內部所發出。

7. 若判斷來源 MAC 非本 NMU，則查詢該 MAC 所屬 NMU 之 TM 的 IP 位址，並對其發出追蹤要求，並重複步驟 6 和步驟 7，直到追蹤到發出攻擊之 NMU。

8. 若在追蹤過程於某節點發現查詢不到該記錄，則以人工方式向周圍之 NMUj 的 LMj 查詢是否有該識別值記錄，若否，則進行例外處理。

9. 若在周圍 NMUj 查詢到該筆記錄，則繼續重複步驟 6 和步驟 7，直到追蹤到發出攻擊之 NMU。

10. 當追蹤完成時，將追蹤結果回覆給追蹤發起者，並完成追蹤。

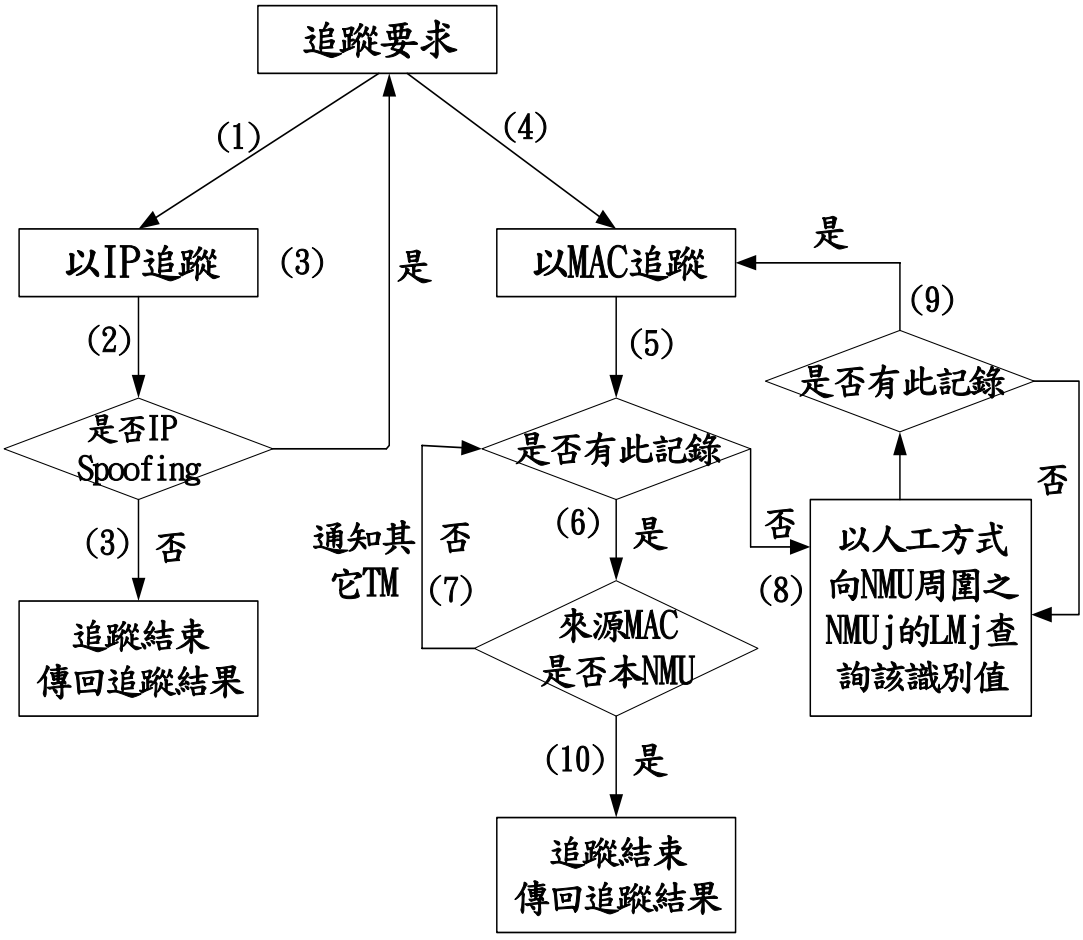

圖 3-3　追蹤方法

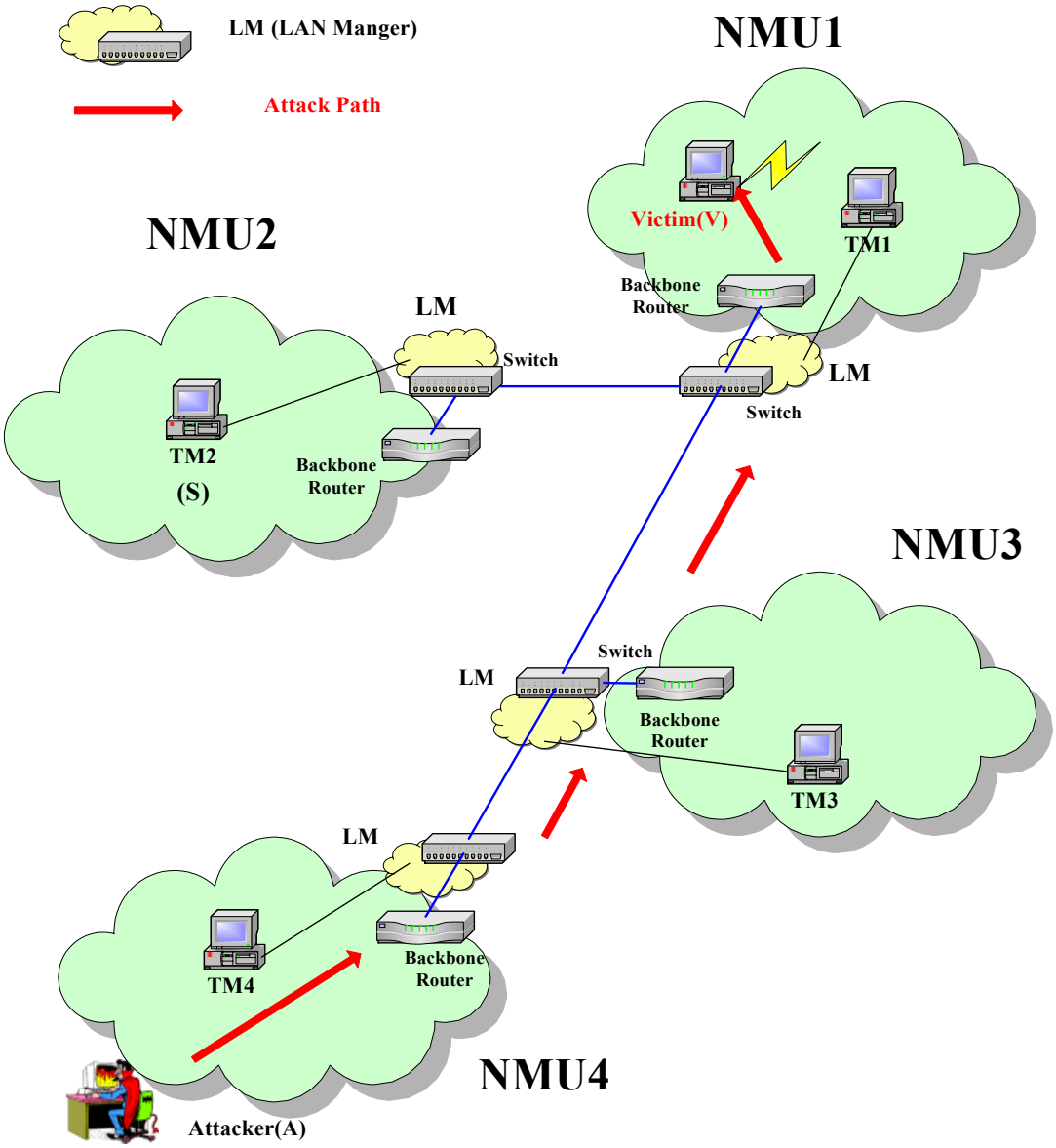

圖 3-4　追蹤流程

以下以圖 3-4 為例說明追蹤流程。圖中包含四個 NMU 的環境，各個 NMU 分別有其 TM 和 LM。NMU1 分別與 NMU2 和 NMU3 相互連接，NMU3 和 NMU4 也相連， NMU2 和 NMU3 則不直接相連。假設攻擊者 A（Attacker）位於 NMU4 內部，攻擊對象是 NMU1 的某受害主機 V（Victim），攻擊路徑（Attack Path）如箭頭所示。攻擊者可能有兩種攻擊模式，其一以自己的 IP 位址而直接攻擊受害主機，其二以偽造來源 IP 位址攻擊之。假設是偽造 NMU2 的主機 S 攻擊 NMU1 的受害主機 V。再假設該攻擊由 NMU1 的入侵偵測系統所發現，並警示及通知 TM1 發動追蹤，或由受害者申訴並由網管人員由 TM1 發動追蹤，以下分述之：

1. 非偽造 IP 攻擊：TM1 接收到偵測系統的警示和追蹤記錄 RD（包含封包識別值、來源 IP 位址和來源 MAC 位址），則發動追蹤。若是網管人員發出之追蹤要求，則 TM1 需先到 LM1 找到該筆追蹤記錄 RD。TM1 取得 RD 後，會向認證系統查詢 RD 之來源 IP 所屬的 NMU 和 TM 之 IP 位址，查證方式將於第四章中描述。此時會得知該 IP 位於 NMU4，並由 TM1 向 TM4 發出包含 RD 的追蹤要求。TM4 在驗證過 TM1 身份後，會依識別值向 LM4 查詢是否有此記錄。LM4 會查到該記錄，則 TM4 可確認該攻擊確實是由 NMU4 所發出，藉由該 IP 與 ATS 資訊比對找到內部攻擊者，並通知網管人員。最後並回覆 TM1 追蹤結果。

2. 偽造 IP 攻擊：因為 TM1 無法判定是否為偽造 IP 攻擊，因此，先透過認證系統查詢來源 IP 位址 S，並通知 S 所屬之 TM2。TM2 在 LM2 中找不到該筆記錄，並將結果通知 TM1。TM1 得知該攻擊是偽造 IP 攻擊，接著以 hop by hop 的方式繼續追蹤。TM1 會以該筆記錄的來源 MAC 位址向認證系統查詢，得知是 NMU3 的骨幹路由器位址，便向 TM3 送出協助追蹤之要求。TM3 依追蹤訊息所包含的攻擊封包之識別值向 LM3 查詢，發現有該筆紀錄，而該筆紀錄並非由內部所發出（因在 NMU1 之中）。因此，TM3 則重複 TM1 的步驟，依 LM3 中該紀錄的來源 MAC 位址向認證系統詢問，得知是由 TM4 送來，並向 TM4 發出協助追蹤要求。TM4 在 LM4 查詢到該紀錄後，可藉由 ATS 的比對找到其來源 MAC 位址，並進一步找到攻擊者。之後，TM4 將結果通知 NMU4 的網管人員和回覆 TM1。

3.4 追蹤方法比較

本論文提出的系統與被動式的追蹤機制如 Hop by hop、AMN 和 Digests 等追蹤機制來比較：

1. Hop by hop 的方式係以逆向的方式向相連接的路由器詢問是否有該攻擊紀錄，進而追蹤該攻擊來源。缺點是所依據的記錄檔在各路由器可能不一致，而且當有大規模的多路徑攻擊時，難以分辨出多條入侵路徑。

2. AMN 強調自治區的觀念，利用小區域易於管理的優點，再透過各區域間的相互協

助來達成追蹤的目的。然而每個封包經過 AMN 所存的資料欄位較多，每次比對會花較多的時間，比對效率較差。

3. Digests 的方法是將封包轉換為各筆 Digests 並儲存之，在追蹤的過程以 Digests 為追蹤依據，只須比對 Digests 而非各個封包資料欄位，有效的增加比對的效率，然每次追蹤均需整合各 SCAR 的部份追蹤路徑，追蹤所需的時間較長。

表 3-1　UDIDT 特色

	UDIDT 特色
1	強調區域管理的特性,以NMU為單位
2	以Hash Code為追蹤依據以減少比對次數
3	以MAC為追蹤基準,並搭配IP以減少追蹤時間
4	加入一有效的入侵偵測機制，已使得本系統亦能做即時追蹤
5	加入一認證系統,可認證各追蹤要求，並能協助追蹤
6	透過DHCP機制保障內部資料不易遭外界存取,並透過SSL保障訊息傳遞的隱密性與完整性

UDIDT 希望能擷取各方法的優點，並改良其不足之處，其特性如下，表 3-1 所示為整理後之清單：

1. UDIDT 同 AMN 亦強調區域的概念，將網路規劃為各個網路管理單位 NMU，NMU 可為校園或企業網路等具有區域管理特性之單位。透過各 NMU 的相互協助，可有效的追蹤到攻擊來源。

2. 每一筆封包紀錄在各 NMU 並非多個欄位，而是採取 Digests 機制的做法，利用雜湊函數產生 Digests 來增加比對效率。

3. 如同 hop by hop 機制般，是以逆向 Hop by hop 的方式由鄰近 NMU 往外一層層追蹤。為了避免 IP 欺騙攻擊，追蹤時以各 hop 的 MAC 位址為基準往外追蹤，因為 MAC 位址比 IP 位址不易造假。然而，若每次均採取 hop by hop 的方式追蹤時間勢必會較長，因此，UDIDT 亦搭配 IP 位址追蹤，在判定非 IP 欺騙攻擊時，直接委由該 IP 所在 NMU 之 TM 進行追蹤，以減少追蹤不必要節點所浪費的時間。

4. UDIDT 加入一有效的入侵偵測機制，除了採 NIDS 的做法偵測網路封包之外，當偵測出高危險等級攻擊時，會主動發出追蹤訊息要求追蹤該封包來源。另外亦採取如同 HIDS 的做法，針對資料庫記錄下來的資訊採關聯性的分析比對。UDIDT 是一個兼具入侵偵測與追蹤攻擊者來源的區域防禦系統，且具有較高的追蹤效率。

5. UDIDT 加入一認證系統，在各 NMU 相互協助追蹤時扮演一公證第三者的身份，以保障各 NMU 的追蹤要求都是可被信賴的。UDIDT 的認證機制中亦有一查詢機制，告知 NMU 該向那個鄰近區域要求追蹤，而不是全面性地要求相鄰者。

6. 透過 DHCP 以保障封包紀錄不易遭外界存取，另外，透過 SSL 連線以保障訊息傳送的完整性與隱密性。

第 4 章 系統安全

在追蹤機制下，每個 NMU 之 TM 必須和其他 NMU 的 TM 訊息溝通，並在透過其他 TM 的協助下，才能有效的追蹤到攻擊者。TM 與 TM 在傳遞訊息和追蹤資訊時，若無安全機制，則攻擊者就可假造大量的追蹤要求給 TM，當 TM 處理假造的追蹤要求時，會浪費大量系統資源，甚至癱瘓，因而淪為 DoS 或 DDoS 的受害者。所以，追蹤系統尚需藉由一認證機制來為各個 TM 作認證，當 TM 接到追蹤要求時必須先驗證對方 TM 之憑證，在確定身分後才執行追蹤。因此 TM 與 TM 間需要一公正的第三者來為彼此作認證，而 TM 和 TM 間的訊息與資料傳遞亦需加密，以免被竊取。認證的機制在 UDIDT 獨立出來而為一認證系統，包括有 PKI 架構與憑證的規格，於系統內元件通訊時做一認證動作。資料保密的機制則藉由 SSL 連線，TM 和 TM 及 TM 和認證系統間都需要有此服務。本章則依序介紹認證與資料保密機制。

4.1 認證系統

認證系統[21-23]主要的工作除了認證每個 TM 的身分外，亦提供 TM 之位置。以下將介紹公開金鑰基礎建設之架構、憑證規格與應用於 UDIDT 認證系統之架構。

4.1.1 公開金鑰基礎建設

認證系統之認證架構採公開金鑰基礎建設，提供了公正的身分認證機制，可以提供加密（Encryption）、簽章（Signature）、鑑別(Authentication)、完整性(Integrity)與不可否認性（Non-repudiation）的服務。PKI 機制可證明此公開金鑰是屬於某一位特定使用者所擁有，其最重要部份是 CA（Certificate Authority）的建制與管理。CA 必須為傳輸雙方所信賴，以下特介紹 PKI 的各元件（請參考圖 4-1）[12]：

1. 終端實體（End Entity）：終端實體是憑證的使用者，在 UDIDT 裡，終端實體為各個 TM。

2. 憑證中心（CA）：為 PKI 機制的核心，擔任可信任的公正第三者，負責 TM 間憑證、憑證廢止清冊（CRL, Certificate Revocation List）之發行與管理等工作。

3. 註冊中心（RA, Registration Authority）：為了避免 CA 負擔過重，CA 可授權某些管理功能給 RA。RA 位於 CA 之前端，TM 可向其申請憑證，不過此憑證需經過 CA 簽署才算完成。

4. 憑證/憑證廢止清冊儲存區（Certificate/CRL Repository）：為公開金鑰基礎架構中的資料儲存體，負責儲放憑證、憑證廢止清冊之資料。各個終端實體可透過儲存體查詢遭廢止的最新清單。

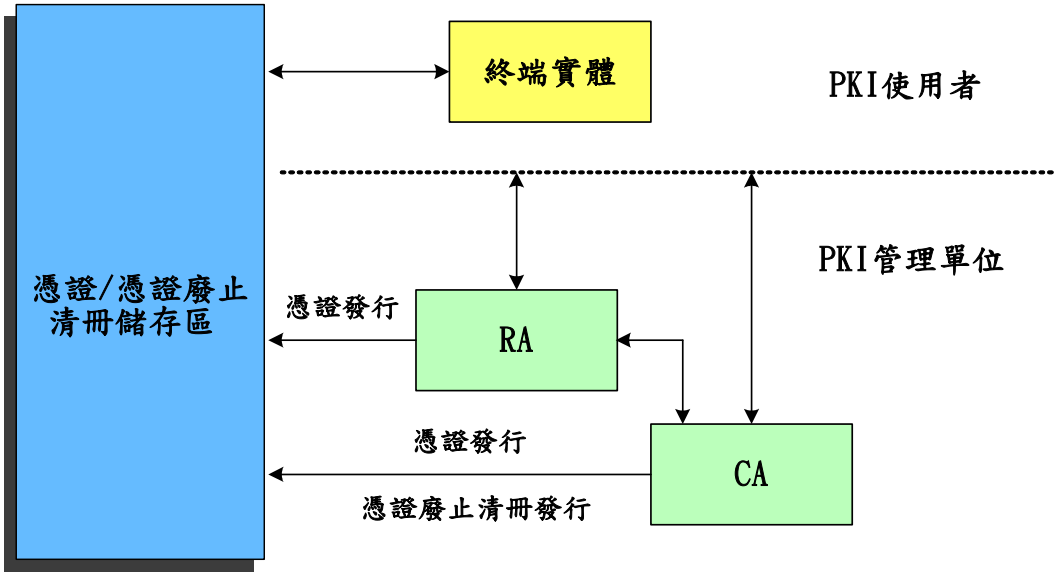

圖 4-1　公開金鑰基礎建設

在 UDIDT 機制下，CA 負責簽發憑證給各個 TM，保證追蹤訊息確實由某 TM 所發出，以建立具有機密性(Confidentiality)、鑑別、完整性、不可否認性的資料傳遞安全環境。取得憑證的步驟，必須先向 RA 註冊，在 RA 驗證過個人資料後會產生經過認可但未簽證的憑證。未簽證的憑證將交由 CA 在憑證上簽章，而後儲存於儲存區並發一份給使用者。當憑證出現問題或依使用者要求廢止該憑證，CA 會產生一憑證廢止清單，亦是存放於儲存區上。

4.1.2 電子憑證

除了 PKI 架構外，認證系統最重要的就是憑證的規格。UDIDT 的憑證是採用 X.509 的憑證規格。X.509 標準為國際電信組織（ITU-T）所制定的 X.500 系列文件的一部分。憑證就如同身分證，可以證實連線使用者的身分，避免有假造的情況，所以 PKI 架構和憑證必須互相搭配，在運作上是缺一不可的。X.509 的憑證規格經過三次的改變，在一開始的「版本一」中只定義憑證的標準欄位；在「版本二」中則添加 Issuer Unique Identifier 和 Subject Unique Identifier 兩個欄位，以支援目錄存取控制；在「版本三」添加了 Extensions 欄位，是一選擇性的擴充欄位。X.509 的憑證規格[26]，可如圖 4-2 所示，它包含了以下幾種欄位：

1. 版本（Version）：此憑證發行的版本，用來區別不同版本間的憑證規格。
2. 序號（Serial number）：憑證中心對此憑證所發予的序號，此序號在該憑證中心是唯一的。
3. 簽章演算法(1)（Signature algorithm identifier）：憑證中心用來簽署此憑證使用之簽章演算法代號，用以驗證簽章時使用。
4. 發行者名稱（Issuer name）：發出此憑證之憑證中心名稱。
5. 有效期限（Period of validity）：憑證之有效期限，包含起始日期與終止日期。
6. 憑證對象名稱（Subject name）：擁有此憑證之使用者名稱。
7. 憑證對象之公開金鑰資訊（Subject's Public key information）：擁有此憑證之使用者之公開金鑰資訊，包括有公開金鑰和演算法代號(2)。
8. 發行者唯一識別（Issuer unique identifier）：如果有其他物件與 CA 名稱相同，就利用此欄位來辨識身份。
9. 憑證對象唯一識別（Subject unique identifier）：如果有其他物件與憑證擁有者名稱相同，就利用此欄位來辨識身份。
10. 延伸欄位（Extensions）：一個以上的擴充欄位，在版本三才有此欄位。

11. 簽章（Signature）：包含了其他欄位的雜湊碼，是憑證中心以其私密金鑰對此憑證所作之簽章。此欄位也指出所採用的簽章演算法(3)。

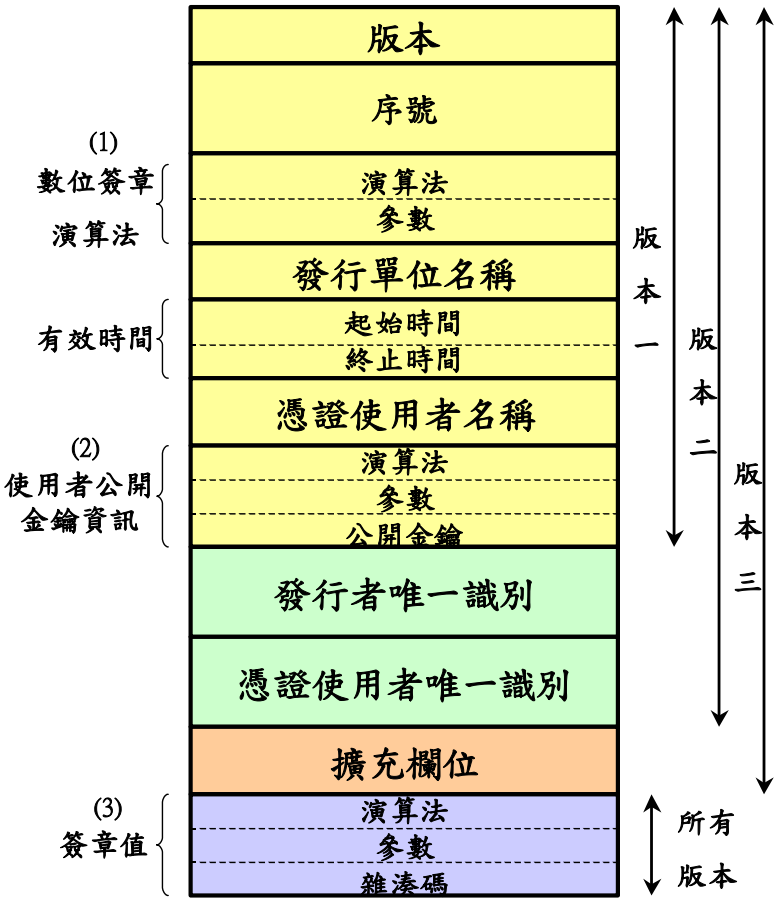

圖 4-2　X.509 憑證規格

CA 在使用者通過註冊程序，加入簽章的過程可如圖 4-3 所表示[12]。

(1) 憑證上的全部憑證欄位會經過一個雜湊函數產生一訊息雜湊碼（MAC, Message Authentication Code）。

(2) 接下來 CA 會以其私密金鑰依照憑證所指定的簽章演算法對訊息雜湊值加密，產生一筆簽章值。

(3) 最後，CA 會將此簽章值記載於憑證的簽章欄位（3）。

在 PKI 中憑證的認證採階層式架構，因為 CA 所作之簽證可保證憑證無法任意變造，

因此除了發給使用者外，亦可發給其它 CA 或放置於儲存體供使用者查詢。

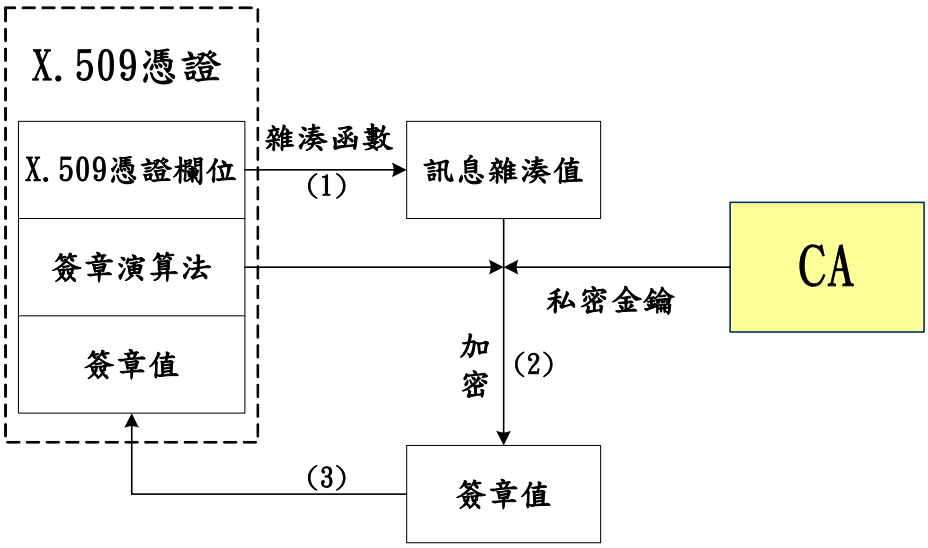

圖 4-3　CA 簽證過程

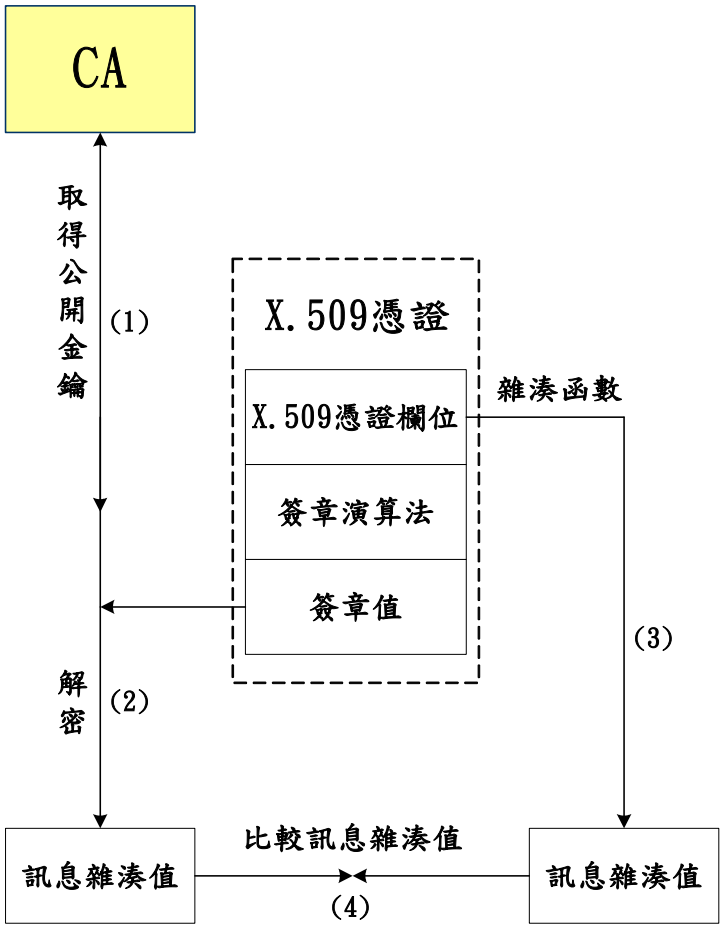

圖 4-4　驗證憑證過程

當使用者需要驗證對方的憑證時，其流程如圖 4-4 所示。

(1) 首先，使用者需向 CA 拿到該憑證的公開金鑰。

(2) 接下來利用此公開金鑰對對簽章值做解密，解密後是一訊息雜湊值。

(3) 此時，原本的憑證中的憑證欄位會經過同一雜湊函數計算而產生另一訊息雜湊值。

(4) 最後，將兩筆雜湊值拿來比較是否相同，若相同則表示通過完整性驗證。

除了驗證憑證的完整性之外，憑證的有效期限也是很重要的驗證對象。對期限之驗證，首先是驗證憑證內的有效日期是否過期，另外，也須對照 CA 的廢止清單是否此憑證已被廢止。

4.1.3 認證系統架構

UDIDT 的認證系統 TA 之架構如圖 4-5 所示，是以 PKI 機制為基礎發展的認證架構，其所發行的憑證是採取 X.509 的憑證規格。TA 的核心為 TCA 與 TQA。TCA 的功能如同 CA，是 TA 中執行認證的機制，負責發行憑證與憑證廢止清冊。TQA 是協助追蹤的機制，當 TA 要求其他 NMU 繼續追蹤時，必須向 TQA 發送訊息詢問，才能得知下一個要追蹤的 NMU 之 TM 所在。

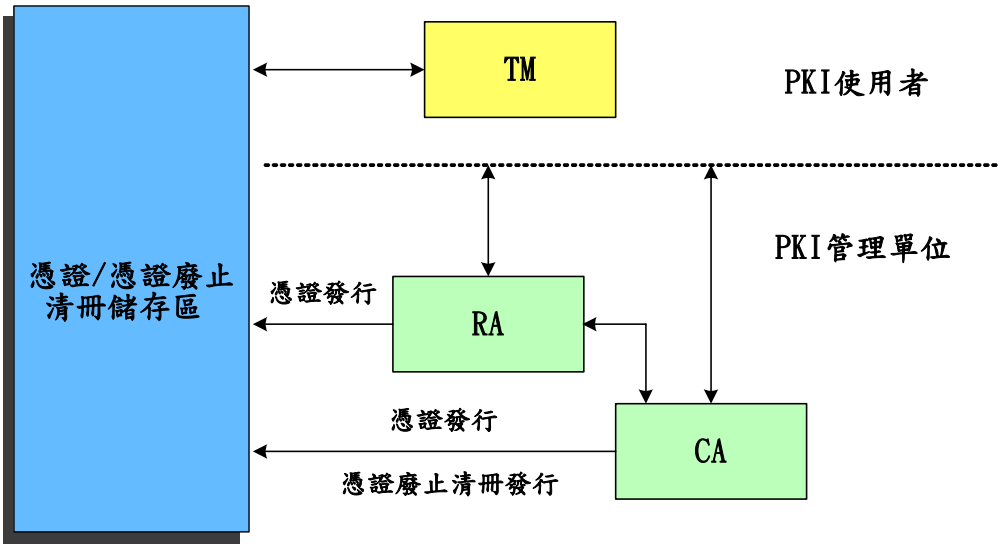

圖 4-5 TA 之架構

TQA 的追蹤資訊是由 RA 所提供，而 RA 為負責各 NMU 註冊的註冊中心。各 NMU 之 TM 需向 RA 註冊，其註冊的資訊共七項，包含：

(1) NMU 管理單位名稱或代號

(2) NMU 管理人員名稱或代號

(3) 通訊地址

(4) 連絡電話

(5) E-MAIL

(6) NMU 所管轄之所有 BR 之 MAC 位址

(7) 該 NMU 之 TM 的 IP 位址

項目（1）之資訊為一張憑證之「憑證使用者」欄位。（2）~（4）項則為備查之資料。（1）、（6）和（7）項則為 RA 需提供給 TQA 的資訊，TQA 將之轉交給 TM。當某個 TM（TM*i*）追蹤攻擊封包 P 時，會從 LM*i* 查詢到 P 是由某個 BR 之 MAC 位址的轉送過來，接著向 TQA 查詢，該 BR 屬於那個 NMU（假設是 NMU*j*），及其 TM *j* 之 IP 位址。TM 得到此資訊後即向 TM*j* 發出協助追蹤訊息。

UDIDT 之 TA 是採階層式的認證架構，如圖 4-6 所示。係依地域性規劃，將數個地理位置相近的 NMU 交由一 TA（第 n 層）管理，稱為區域 TA（Local TA），若干第 n 層的 TA 再由某一第 n-1 層的 TA 認證。在複雜的 UDIDT 系統裡，階層式的架構可能會有很多層，此時上層 TA 就要為下層作認證。因此第 *i* 層的 TA 中的 TQA 必須彙整以該 TA 為根節點之子樹（sub-tree）中之所有 TA（TQA）的註冊資訊。當 TM 在其所屬之最下層之 TA 查詢不到資訊時，則由該 TA 向上一層之 TA 查到更完整的資訊。

因此，UDIDT 在 TA 機的制下，不僅可在 TM*i* 向 TM*j* 要求追蹤時，由 TM*j* 透過 TA 驗證 TM*i* 的身份，亦可在 TA 的協助下繼續追蹤。因為各區域 TA 所管轄 TM 數量不同，TM 數量較少之區域 TA，其 RA、TCA 及 TQA 可合併成 RA/ TCA/TQA 元件，目的是減少建構成本。

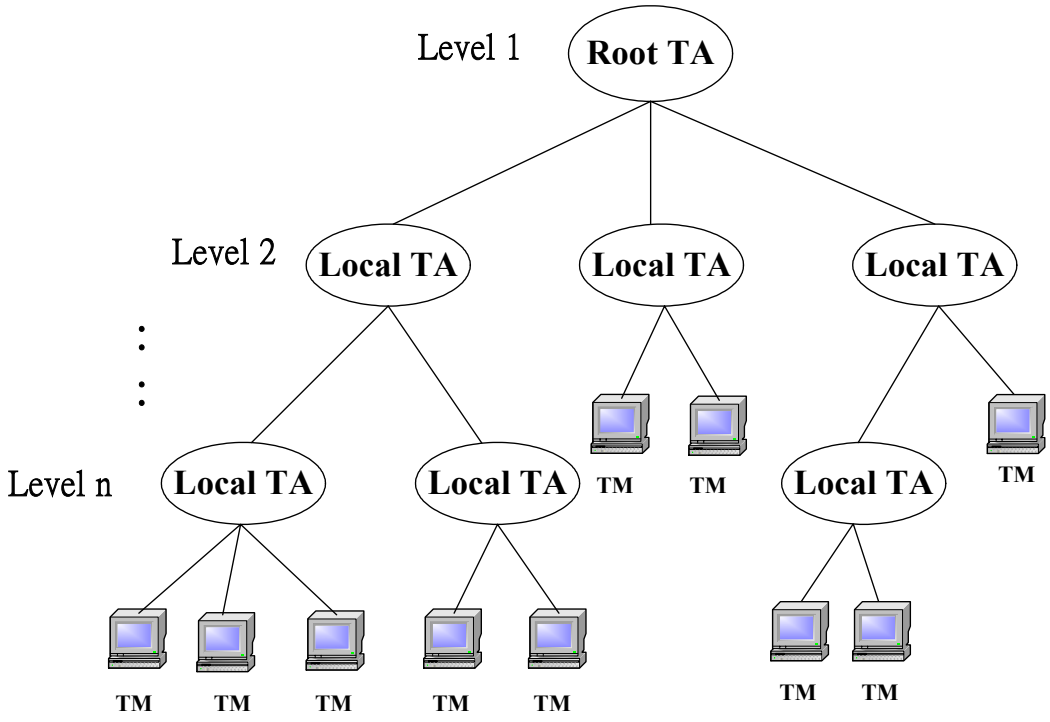

圖 4-6　TA 之階層式架構

4.2　資料保密

TA 的任務之一是認證 TM 的身分，然而，在資料與訊息交換過程只有認證機制安全性似乎不夠，資料或訊息還是有可能被偽造或竊取。UDIDT 系統利用 SSL 協定來為通訊雙方提供一安全通道，並以加密的方式來保障傳輸雙方資料與訊息安全。以下描述 SSL 與其於 UDIDT 上的應用[21]。

4.2.1　SSL 機制

SSL 安全協定最早是由 Netscape 在 1994 年所發表[24]，是架構在 OSI（Open System Interconnection）的應用層（Application Layer）與傳輸層（Transport Layer）間的安全協定。SSL 提供兩通訊端在應用層的可靠和安全的通道，其連線具有以下特性：

1. 連線隱密性（Confidentiality）：通訊的資料都會經過加密，假設有人竊取資料但若無法解密，也將無法還原原本資料。連線的雙方在一開始會以一交握協定（Handshake Protocol），協議雙方的加密演算法與加密金鑰。

2. 身份鑑別（Authentication）：在交握過程中，雙方可協議做身份驗證，係利用 PKI 架構所發行的 X.509 憑證來互相驗證身份。

3. 資料完整性（Integrity）：利用像是 SHA1 或 MD5 等安全雜湊函數對傳輸訊息產生訊息驗證碼 ，利用此驗證碼驗證訊息傳遞時的完整性。

SSL 協定是由兩階層的協定所組成，如圖 4-7 所示，其包含四個通訊協定，分別為：

1. 交握協定：協議雙方的加密演算法與加密金鑰，兩通訊端間亦可相互要求對方的電子憑證作身分驗證。

2. 改變密碼規格協定（Change Cipher Spec Protocol）：通訊兩方在做交握時，雙方用以交換加密演算法與加密金鑰的協定。當交握時，雙方皆會依此協定傳遞改變密碼規格訊息來通知對方，此後傳送的訊息皆會被協議的密碼規格與金鑰所保護。

3. 警報協定（Alert Protocol）：定義通訊雙方在 SSL 連線過程發現有錯誤時，所發出的警報等級與描述。警報等級分為警告（Warning）與致命（Fatal）兩個，當收到致命警報，則雙方均關閉連線，而警告警報的訊息內容的描述有：憑證錯誤、錯誤的訊息驗證碼與非預期錯誤等。

4. 記錄協定（Record Protocol）：通訊端將欲傳輸的資料經過分割（Fragment）、壓縮（Compress）、加入訊息驗證碼（MAC）並加密（Encrypted），最後加入 SSL 記錄標頭（SSL Record Header），再經由傳輸層傳送。當接收端接到後再經由記錄協定反向的解密（Decrypted）、驗證（Verify）訊息驗證碼與組合（Reassemble）資料後，送交至應用層。

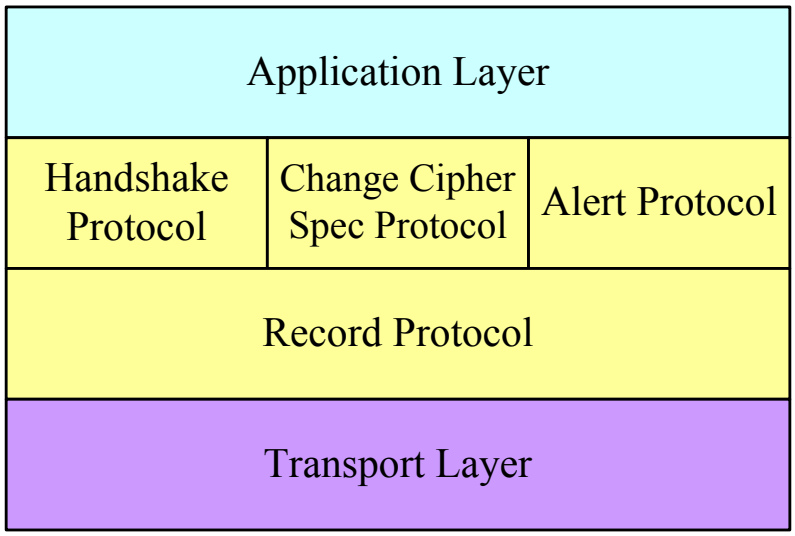

圖 4-7　SSL 的安全協定

4.2.2 追蹤流程保密

SSL 與 TA 搭配是希望能夠保障追蹤系統的連線隱密性、身份確認與訊息和資料的完整性。

UDIDT 機制裡需作 SSL 連線的對象有：

1. TM 和 TM：當 TM*i* 欲傳遞追蹤要求給其他 TM*j* 時，TM*j* 會要求建立起一段 SSL 連線。兩方會做一交握協定時，用以交換加密演算法與加密金鑰。之後，雙方交換由 TA 所發予的憑證，並驗證對方的身份，如此則完成 SSL 的交握協定。接著，TM*i* 以雙方協定過的加密演算法將訊息加密並送交 TM*j*，TM*j* 在解密獲得此訊息即中斷此 SSL 連線。

2. TM 和 TQA：做法同於 TM 和 TM 之通訊，TM 在向 TQA 查詢時雙方亦必須建立 SSL 連線，並在驗證雙方身份後以加/解密方式獲得所需資料。

透過 SSL 與 TA 之搭配，可確信雙方的身份認定，亦可保障訊息於傳遞過程不被修改，以確保資料的完整性。否則：

1. 若無身份的認證，攻擊者可輕易的向 TQA 察詢到資料，查到各 TM 的位址並予以攻擊，而造成系統的癱瘓。

2. TM 和 TM 的通訊若無安全機制則亦面臨到遭受攻擊的危險，攻擊者可偽造為一假的 TM*i* 對其它 TM 不斷的送出追蹤的要求，而使得 TM*i* 需耗資大量資源處理這些偽造的追蹤要求，而淪為 DoS 或 DDoS 的受害者。

3. 相同的，若無 SSL 的保護，攻擊者可輕易修改我們所傳送的訊息，除了可竊取到雙方通訊的內容，亦可偽造假的訊息，而達到攻擊的效果。

因此，為了確保 UDIDT 追蹤系統的安全，我們加入了系統的安全機制，藉以提昇系統強固性。

第 5 章 追蹤系統機制

本章將對 UDIDT 的追蹤系統相關元件做一詳細敘述。首先介紹如何由 Mirror Port 將流量導入到 LM 及 HP 元件如何處理各個封包，以產生有用的追蹤與偵測資訊。其次說明封包寫入資料庫的過程，其中牽涉到偵測系統的偵測，因為偵測機制不在此論文探討範圍，故僅介紹簡單流程，以保持章節連貫性。再者陳述追蹤管理單元 TM 之機制，最後，則探討整個追蹤系統所需的通訊協定。

5.1 流量導入

一個 NMU 架構可簡化如圖 5-1 所表示，主要由 TM 和 LM 所組成。經過骨幹路由器 BR 的每筆進出封包均利用具有 Mirror Port 功能的交換器導入 LM 區域，並經過 LM 內部元件的處理才能在追蹤時提供適當資訊。

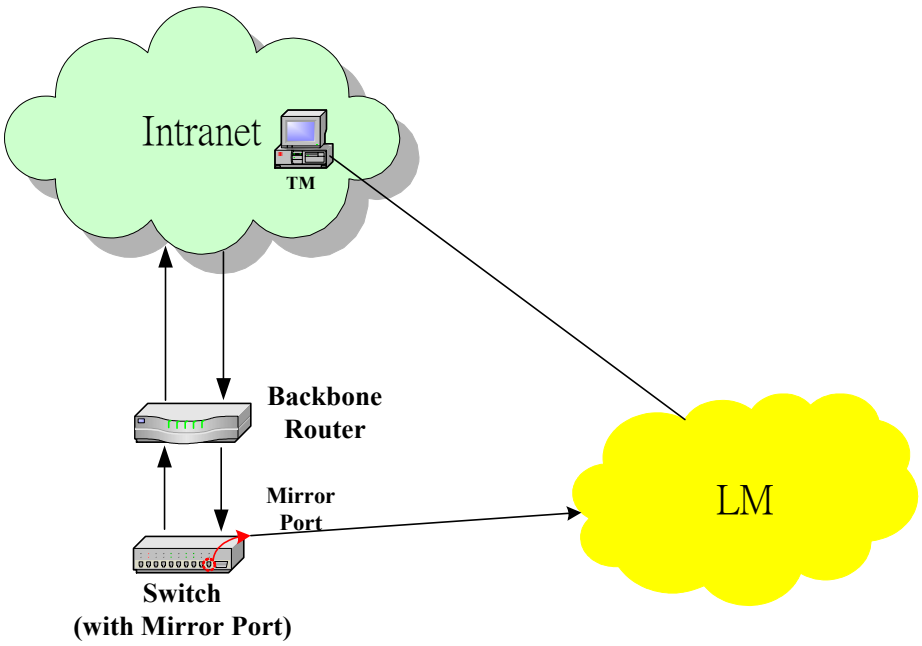

圖 5-1 簡化的 NMU 架構

一般而言，功能較強的交換器都具有 Mirror 封包的功能，能夠將某個或所有其他埠（Port）的網路流量導入某一個指定的埠中，這個埠稱為鏡射埠（Mirror Port）或 Span Port，其結構如圖 5-2。網管人員可以指定哪些 Port 是要導入，所有經由這些指定 Port 的封包在經過 Switch Backplane 時，除了將之送往目的 Port 外，也會在此複製一份送交給 Mirror Port。每個交換器依其功能的不同，有些只能 Mirror 單向流量，功能較強者則能 Mirror 進、出該交換器的雙向封包流量。因為 LM 必須紀錄進、出 BR 的所有封包資訊，因此務必選擇具有能夠 Mirror 所有雙向流量的交換器。

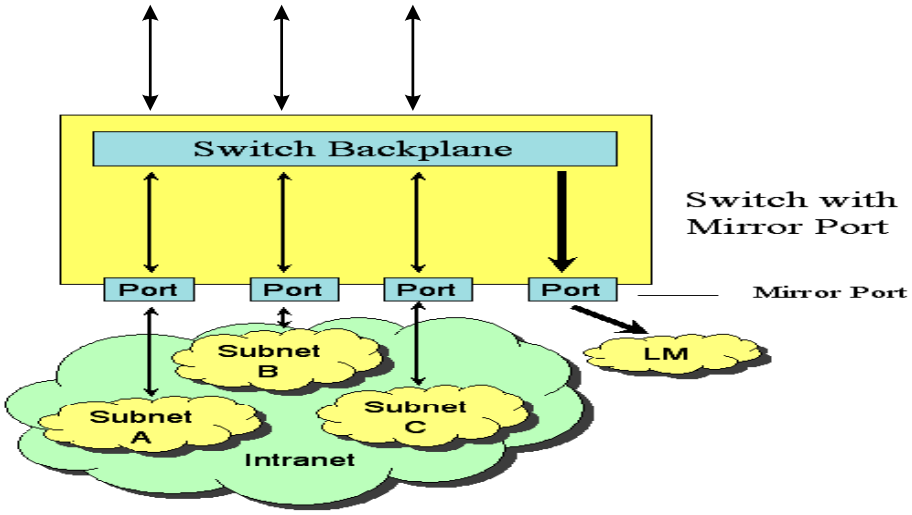

圖 5-2　具 Mirror Port 交換器結構

為了能將所有送往 BR 的封包資訊都能有效地導入到 LM，交換器的位置便很重要。若交換器置於 BR 之後端內部網路中，則因交換器只收到內部網段 IP 的封包，轉送的封包則不會進入 LM 中，因此無法協助其他 TM 進行追蹤，改善的方式就是將交換器放置於 BR 的前端，所有送往或離開 BR 的封包，會被記錄一份下來。

5.2　封包處理

導入至 LM 的封包，會經過 HP 做前置處理。HP 由兩個元件所組成，分別為 Sniffing AP 和 Hash Processor，如圖 5-3，而對封包做兩階段的處理。

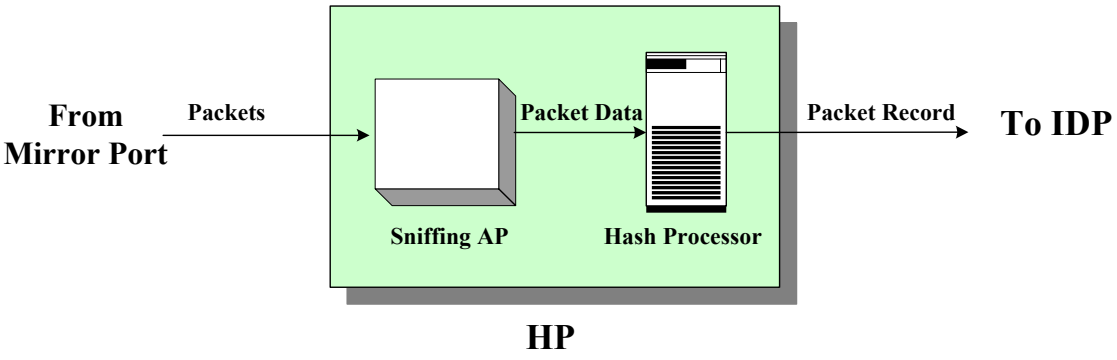

圖 5-3　HP 結構

在第一階段的處理中，Sniffing AP 會接收與處理 Mirror Port 送來的封包，其處理程序有二：

1. 設置網路介面於錯亂模式(Promiscuous Mode)：HP 必須有一設置於錯亂模式的網路介面與交換器之 Mirror Port 相連，才能接收所有送至該介面的封包，不論該封包目的 IP 位址是否為該主機。因為在正常模式下每台主機只會收取屬於自己的封包。經由錯亂模式抓取下來的封包，則先置於 Sniffing AP 的緩衝區中待處理。

2. 彙集所需資訊：擷取緩衝區內的封包追蹤資訊與偵測資訊，將述於後。Sniffing AP 會將資訊匯集成一筆封包資訊（Packet Data），再送往 Hash Processor 做第二階段處理，並丟棄該原始封包，以減少不必要的資訊量。

Sniffing AP 對資料的處理可詳述如下：

1. 封包型態

Sniffing AP 的處理是 HP 最繁重的負荷（load），效率非常重要，因此，以多執行緒的方式進行，可同步處理多個封包。必須處理的資訊包含：二、三、四層標頭及部分的資料[13,24]。以 Ethernet 的網路環境而言，Sniffing AP 會先拆解其第二層（Ethernet）標頭，見圖 5-4，計有：目的 MAC 位址、來源 MAC 位址和封包型態（TYPE）三個欄位。

目的 MAC 位址代表該封包所欲送達主機或路由器之 MAC 位址，其可能情形有：(1)

如是由其他區域經由 BR 送往本 NMU 的封包，則為目地位址為 BR 的 MAC 位址（2）若是由本 NMU 對外發送的封包，則為其目的 MAC 為下一個 hop 的 BR 位址。

來源 MAC 位址代表發送該封包的主機或路由器的 MAC 位址，其可能情形有：(1) 若由該 NMU 經 BR 往外發送，則為 BR 之 MAC 位址（2）若係由外界經 BR 送往本區域的封包，則為上一 hop 的 BR 位址。

TYPE 欄位可判斷出該封包是封裝何種類型的第三層(IP 層)標頭，計有 0800、0806 和 8035 三種值，分別表示封裝的是一個 IP、ARP 和 RARP 封包。後兩者表用於 IP 位址與 MAC 位址的相對應(Mapping)查詢。因此，攻擊封包大都是 IP 封包(Type=0800)。

圖 5-4　Ethernet 標頭格式

IP 標頭（見圖 5-5）包含：版本、標頭長度、服務類型、封包總長度、封包認證碼、旗標、區段位移、存活時間、通訊協定、總合檢查碼、來源 IP 位址、目的 IP 位址及選項等欄位。

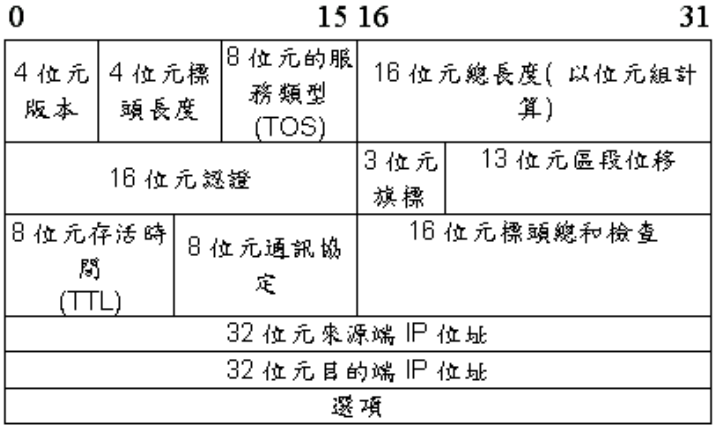

圖 5-5　IP 標頭格式

Sniffing AP 在拆解 Ethernet 標頭時，由 Type 欄位值判定所包裝的是否為 IP 封包。

在拆解 IP 標頭時，利用通訊協定欄位值，可判定封裝的是哪一種型態的第四層（傳輸層）封包， ICMP 為 1、TCP 為 6、UDP 為 11 等。再依不同種類標頭分別處理。唯 EGP、OSPF..等協定用於路由器交換路由資訊時，攻擊者不易以此協定攻擊，故予以忽略。攻擊者常用的是 UDP、TCP、ICMP 協定，因此，只處理此三類的封包，其它者則予以丟棄，以減少不必要的資料量。

2. 追蹤資訊

Sniffing AP 所擷取的追蹤資訊，會交由 Hash Processor 產生各封包的識別值，以為封包在網路上流動的識別依據。其所需資訊包括：封包的 IP 標頭與部份資料。IP 標頭某些欄位在傳輸過程會被其它路由器所改變，例如，存活時間欄位（TTL, Time To Live）在經過每個路由器時，其值會被遞減 1，而旗標和區段位移欄位也會因封包被路由器分段而有不同值。因此，為了讓各區域的 NMU 能對同一封包產生相同的識別值，則必須選取不會變動的欄位，要選取的 IP 標頭欄位，圖 5-6 之白色部份即所取部份，共計 14 個位元組。而同一台主機在不同時間向某一台主機所發出的封包， IP 標頭欄位有可能完全相同，但資料部份又是相同的機率則較低。所以，為了降低識別碼的碰撞率，而加入 40 位元組的資料。其另一個目的則是配合偵測資訊所須擷取的資料位元數，盡量以夠用即可為原則。

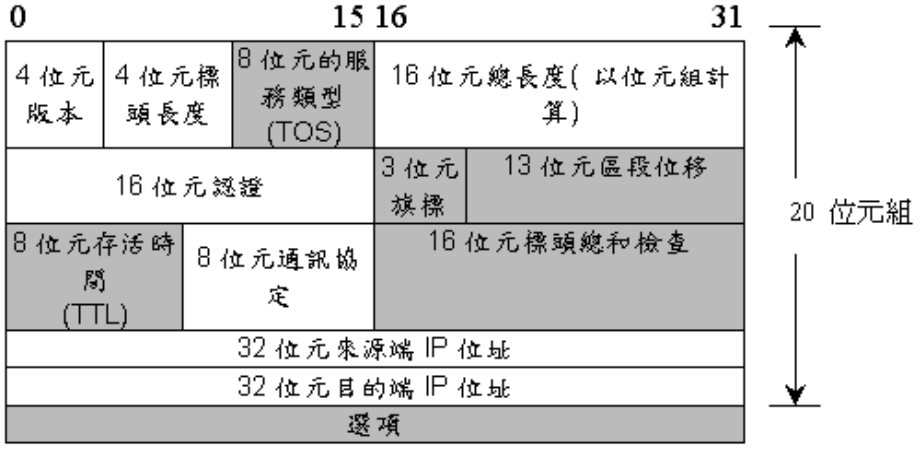

圖 5-6　白色部份為 IP 標頭追蹤資訊

因此，一個 Trace Record（TR）共計 54 個位元組，見圖 5-7。

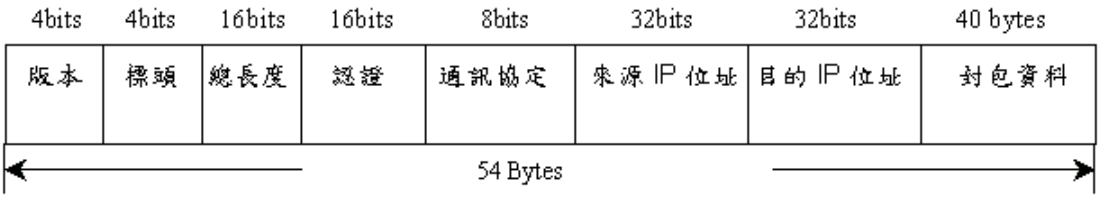

圖 5-7　Trace Record

3. 偵測資訊

Sniffing AP 所需擷取的偵測資訊，包含：來源 MAC 位址、目的 MAC 位址、封包的類型、封包大小、來源 IP 位址、目的 IP 位址、來源 Port 號碼、目的 Port 號碼、序號（SYN）、回報序號（ACK）、時間戳記及 40 位元組的封包資料等，詳細者請參考表 5-1。這些資料散落在 Ethernet 標頭、IP 標頭、傳輸層標頭（TCP、UDP 或 ICMP）與資料部份，Sniffing AP 會將這些資料彙集並以系統時間加上一時間戳記，再依 TCP、UDP 和 ICMP 不同型態封包，分別產生最大 91、83 和 81 位元組的偵測資訊 Detection Record（DR），用以支援後端的偵測系統進行偵測，而其中一部份，例如，來源 MAC 與 IP 位址則為追蹤之用。

表 5-1　Detect Record 各欄位

欄位編號	標頭	欄位名稱	欄位大小(bytes)
1	Ethernet	來源 MAC 位址	6
2	Ethernet	目的地 MAC 位址	6
3	IP	封包大小	2
4	IP	識別（Identification）	2
5	IP	分割（Fragment）	2
6	IP	IP 封包型態（Protocol）	1
7	IP	來源 IP 位址	4
8	IP	目的地 IP 位址	4
9	TCP/UDP	來源 Port	2
10	TCP/UDP	目的地 Port	2
11	TCP	序號（SYN）	4
12	TCP	回報號碼（ACK）	4
13	ICMP	Type	1
14	ICMP	Code	1
15		封包資料	40
16		時間戳記	4

之後， Hash Processor 以一單向雜湊函數（One-way Hash Function），例如，MD5 或 SHA1（本研究採 MD5），計算 TR 產生一個雜湊值，以為該封包的識別值，也就是 Digests。以雜湊值追蹤的優點在於將來追蹤時不須一一比較 TR 中的各個欄位，只須比較 Digests 即可 ，以提高追蹤效率。另外，資料量也能在雜湊函數的壓縮下，TR 由原本的 54 位元組大幅減少為 8 個位元組資料（以 MD5 的演算法）。最後，Hash Processor 將識別碼和 DR 合成一筆 Packet Record（PR），送交給 IDP 處理，如圖 5-8 所示。

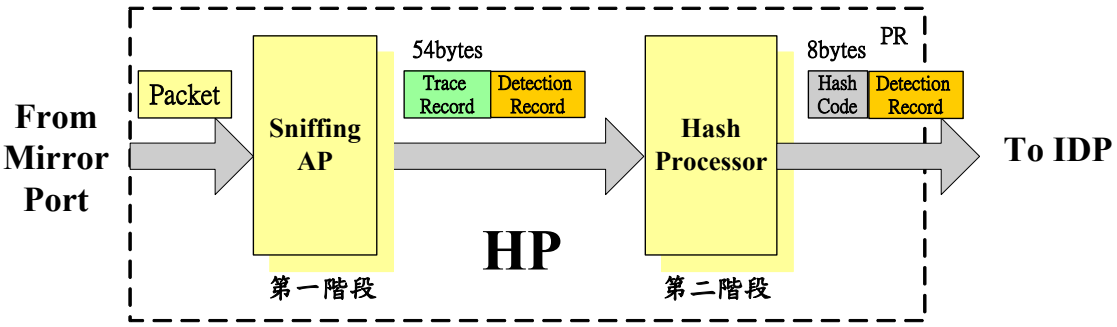

圖 5-8　HP 處理流程

5.3 入侵偵測與資訊儲存

HP 處理後的 PR 會送至後端的 IDP 與 RS 作偵測與寫入資料庫。IDP，如圖 5-9，有三個佇列群組，HP 送來的 PR 先經由網路遮罩（IP MASK）判斷該 PR 是進、出該 NMU 或只是經由 BR 轉送，而分別放入 In_DQ、Out_DQ 和 Foreign_Queue 三佇列中：

1. In_DQ 群組：存放外部進入該 NMU 的 PR，每一目的 IP 有一 IP_In_DQ，因此，NMU 中若有 n 個 IP，則 IP_In_DQ 數量最多應該是 n 個。網路遮罩若判斷目的 IP 位址為本網段 IP 而來源 IP 非本網段 IP 時，則動態產生 In_DQ，並將 PR 放入該佇列。若該佇列已存在，則直接將 PR 放入佇列中。

2. Out_DQ 群組：存放由該 NMU 對外發送的 PR，每一目的 IP 有一 IP_Out_DQ，其數量最多亦是 n 個，但任一時候和 IP_In_DQ 數量不一定相同。作法同 In_DQ 群組，差別在於以來源 IP 來分群組。

3. Foreign_Queue 群組：存放經 BR 但非進出該 NMU 的封包資訊。若目的 IP 位址與來源 IP 位址均非本網段 IP，則將之放入系統唯一的 Forign_Queue 中。

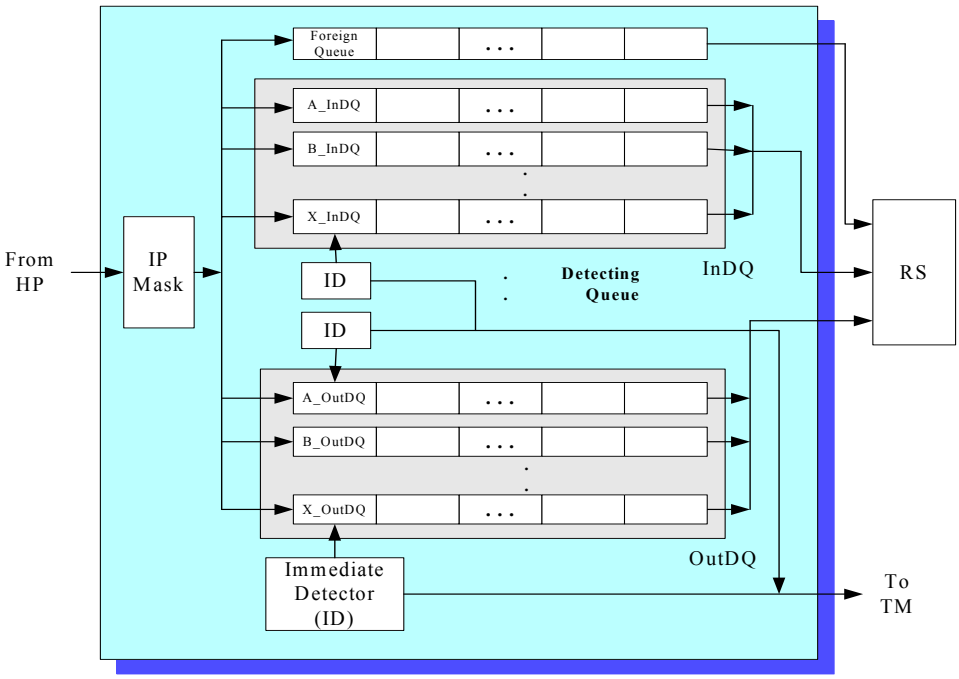

圖 5-9　IDP 架構

IDP 有一 Immediate Detector（ID）會偵測 In_DQ 和 Out_DQ 內各佇列，俾即時偵測入侵攻擊。IDP 各佇列群組處理完的 PR 則送交 RS 處理。RS 以三個緩衝區 In-bound Temporary Area（IBTA）、Out-bound Temporary Area（OBTA）和 Foreign Temporary Area（FTA）做寫入資料庫前的緩衝處理，等緩衝區滿時，再分別寫入資料庫中。RS 有一 Auxiliary Detector 偵測 OBTA 中是否有 DDoS 攻擊。另外，亦有一 Analyzer 偵測分析資料庫內容。各 PR 在寫入資料庫後即完成封包的處理流程。

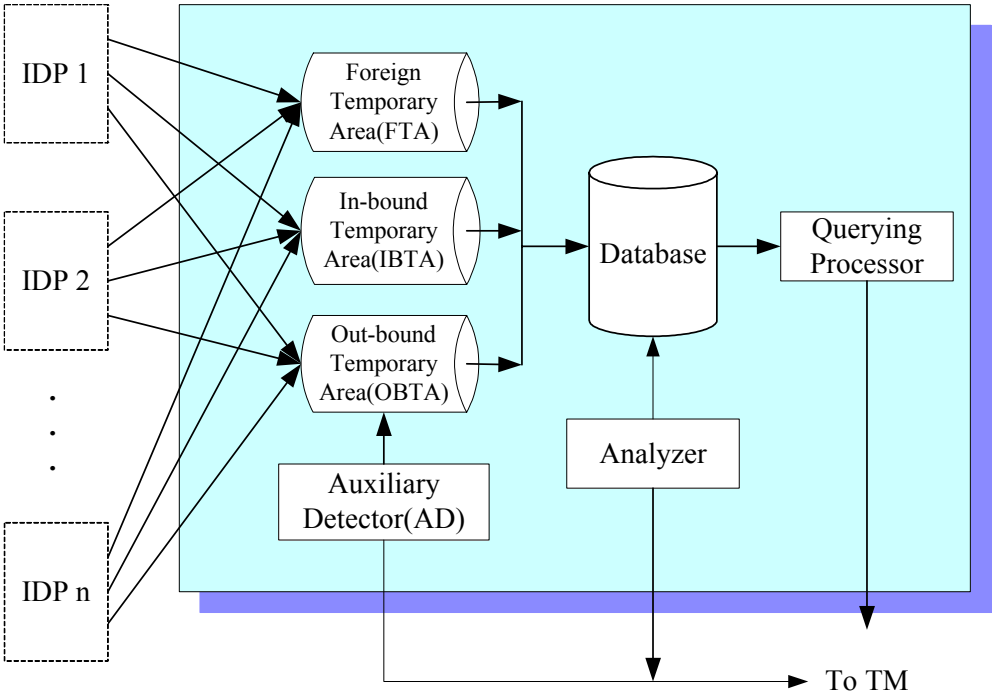

圖 5-10 RS 架構圖

5.4 追蹤管理

各 NMU 區域中，LM 是負責提供追蹤資訊與執行封包偵測的機制，而 NMU 的管理機制則由 TM 來執行。整個入侵追蹤，是透過各 TM 和 TM 間的相互合作，才得以完成。在追蹤過程中，其他的機制，例如，TA 的認證、查詢和 LM 的追蹤資訊，都只是扮演協助的角色。TM 的內部架構如圖 5-10 所示，共包含五個子元件，分別為 SSL、DHCP、NTP、ATS 與 Trace Agent，分別詳述於下：

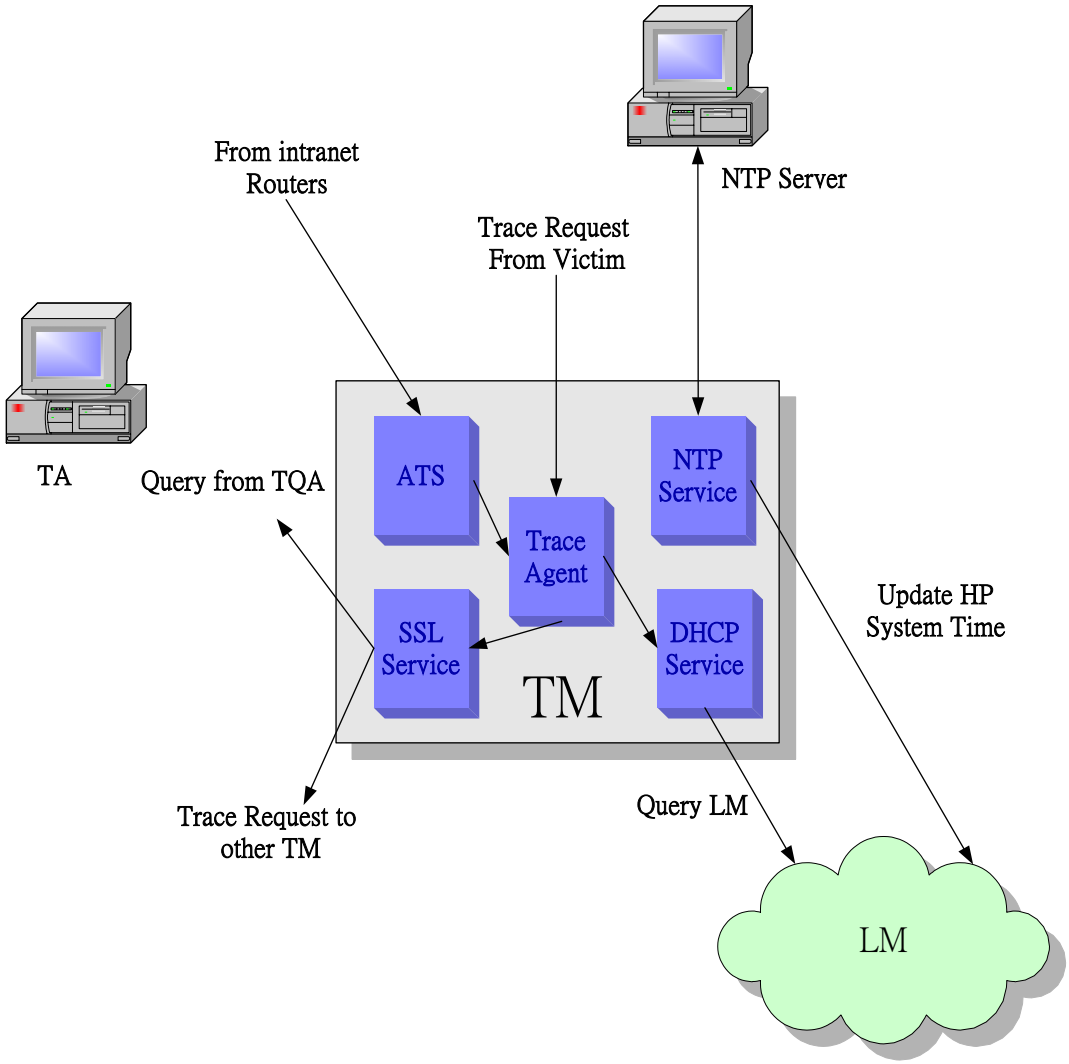

圖 5-11 TM 各元件

1. SSL

如前述，在入侵追蹤機制裡，藉由 SSL 協定保護傳輸安全，包括 TM 向 TA 的 TQA 單元查詢某 BR 所屬之 NMU，以及 TM 與 TM 間的追蹤要求。TM 向 TQA 查詢所建立的 SSL 是由 TQA 擔任連線的伺服器端，而 TA 是接受服務的客戶端。而在 TM 和 TM 間的追蹤要求，各 TM 均有機會擔任提供 SSL 連線服務的角色。發送追蹤要求的 TM 是連線的客戶端，其會先向所欲發送要求的對象先發出 SSL 連線要求，透過交握協定，對方會回應要求並給予憑證，發出端在驗證對方身分後也會傳送本身憑證給對方，在雙方均驗證過身分後，要求追蹤的 TM 才將追蹤要求與識別碼 透過加密的方式傳送給對方，對方在解密後即能處理此要求。因此，TM 本身需能提供 SSL 服務才能保障追蹤過程的安全。

2. DHCP

如前述，各 TM 均有兩個網路介面，一個介面是與其他 TM 溝通，而另一個則可為 LM 各主機提供 DHCP 服務[27]。TM 提供 DHCP 的網路界面擔任 DHCP 的服務端，其分配 IP 的方式有兩種，分別為動態分配（Dynamic Allocation）與自動分配（Automatic Allocation）。動態分配為最常見的分派方式，客戶端獲得 IP 後並非永久使用，當使用時間到就得釋放，並重新跟服務端租用其他 IP。自動分配則在獲得 IP 後即能一直使用，直到服務端要求收回。在 LM 各主機關機後，會要求分派 IP 位址時，TM 會以自動分配方式分別發給 HP、IDP 與 RS 一個固定的虛擬 IP 位址，使三者均得以永久使用，直到 LM 關機，TM 會再將之收回。在 DHCP 機制中，TM 是唯一能直接存取 LM 的主機，並能直接向 LM 查詢所需的追蹤資訊。而偵測系統的追蹤要求，也是透過 DHCP 機制傳送，因 DHCP 能在架構上保護訊息與資料不被外界所竊取，是故 LM 與 TM 的訊息傳遞不需以 SSL 連線。

3. NTP

NTP（Network Time Protocol）[28]是由美國德拉瓦大學的 D.L. Mills 教授於 1985 年提出，除了可以估算封包在網路上的往返延遲外，還可獨立地估算電腦時鐘偏差，達到在網路上高精準度電腦校正時間的目的。NTP 可以精確到 2 的負 32 次方秒，實際應用在廣域網路可達數十毫秒。NTP 是利用 UDP 封包來傳輸，使用 Port 123 和其他 NTP 伺服器或客戶端交談。而 NTP 採用階層式架構，第一層伺服器直接同步國家標準時間，第二層伺服器透過第一層伺服器同樣是同步國家標準時間，第三層伺服器則透過第二層伺服器，依此類推。在 UDIDT 機制裡，各 TM 會透過 NTP 的服務，定時向同一個 NTP 伺服器要求校正時間的服務，之後，各 TM 則轉而校正居於同一 NMU 之 LM 中的 HP 的系統時間，以同步各 NMU 之 HP 的系統時間，保障 PR 中所加入的時間戳記的精確性，避免因時間誤差而產生追蹤與偵測上的誤判。

4. ATS

當各 TM 發現攻擊來源來自自己的 NMU 內部時，透過其 LM 的資訊，網管人員必須取得攻擊來源主機的 IP 與 MAC 位址，以之追蹤攻擊來源。在網路有良好的管理的環境下，例如，使用者在申請 IP 位址時，必須告知 MAC 位址且須註冊其基本資料，若是如此，此時只要將由 LM 所得到的資訊與事前已建立的網路使用者資訊相比對，甚至可

將比對自動化，則可迅速追蹤到攻擊者。若是在一個非良好的管理政策下，網管人員必須以人工的方式分析及比對 IP 及 MAC，以找出攻擊者。一方面，攻擊者往往會偽造 IP 位址，因此，必須要有正確的 IP 與 MAC 位址的對應，才能找出正確的攻擊者，否則，常會有誤判的情形。TM 中則會有一機制定時地蒐集內部主機 IP 和 MAC 位址的對照表，也就是 ARP Table，存放在 ATS 的儲存體中。在做內部追蹤時，直接以追蹤資訊比對儲存體的資料，即可追蹤至攻擊者。

5. Trace Agent

Trace Agent 是 TM 中處理追蹤要求的單元，其需有兩種追蹤模式：

(1) 自主式追蹤：由偵測系統所發出或由其它 TM 經過認證過的追蹤要求，Trace Agent 會以自主式追蹤模式自動進行追蹤。追蹤要求會放於一工作佇列中，只有在最終的 TM 發送追蹤已完成、起始 TM 接收已完成追蹤的訊息，或者在中間過程的 TM(TM*i*) 在判定非其區域而相前繼續發出追蹤要求訊息後，位於 TM*i* 工作佇列的該工作項目才會完成。

(2) 人工追蹤界面：對於偵測系統未能偵測出來的攻擊，當受害者向其 NMU 區域的網管人員申訴時，網管人員可依受害者提供的資訊，例如，遭受攻擊的時間、受害主機 IP 位址等，向 LM 查詢資料。經證確實有未偵測到的攻擊時，可由網管人員透過此模式發出追蹤要求，此後的流程如同自主式追蹤模式。因此，人工追蹤界面包含有查詢 LM 資料庫與發出追蹤要求的功能。網管人員透過此模式發出的追蹤要求亦須經過 SSL 加密與認證的程序。另外，對於經由此模式所發出的追蹤會記錄該追蹤資訊 PR，以便將來能透過此記錄補強偵側系統，換言之，UDIDT 具備蒐集新的攻擊模式的功能。人工追蹤介面也負責處理最後的追蹤結果的回覆，網管人員也在追蹤到最後使用者後，透過此介面將最終的追蹤描述回覆給初始追蹤要求者（即受害者）。

5.5 訊息協定

在 UDIDT 系統裡，各元件的溝通必須靠共通的訊息協定，否則元件與元件之間若

無溝通方式就無法相互的協調運作。系統裡的訊息協定依其作用可分為三種類型的協定，其分別為追蹤、查詢與同步等協定，以下分別敘述之：

1. 追蹤協定

追蹤協定主要是用於追蹤要求處理時的溝通，包括有 Trace Request from DS（DS 指 LM 的偵測器）、Trace Request from TM、Trace Request between TM、Trace Notice、Trace Result 和 Trace Alert 等六種訊息格式，其訊息名稱、訊息內容/參數和描述如表 5-2 所示。

(1) Trace Request from DS：由偵測系統發出的主動追蹤要求，當其 NMU 的 TM 收到此訊息會先查看是否為內部追蹤，若是，則透過人工追蹤界面通知網管人員。若需要其他 TM 協助追蹤則發出 Trace Request from TM 訊息給下一個 hop 的 TM。

(2) Trace Request from TM： Trace Request from TM 是追蹤發起者發出的追蹤訊息，當其他 TM 收到此訊息會查看訊息中的 Digests 是否在自己的區域內，否則，向其他 TM 繼續發出協助追蹤要求，並以 Trace Notice 向追蹤發啟者之 TM 回報。

(3) Trace Notice：協助追蹤的 TM 回報追蹤結果給追蹤發啟者。

(4) Trace Result：當追蹤完成時，網管人員利用人工追蹤界面發出最後追蹤結果描述的 Trace Result 訊息給追蹤發起者，如此則完成一追蹤流程。

(5) Trace Alert：作一例外處理，萬一追蹤流程中斷，該中斷點的 TM 則發出一 Trace Alert 訊息通知追蹤發啟者之 TM 所發生的例外狀況，並由中斷點的 TM 以人工方式向外圍追蹤。

表 5-2　追蹤協定的訊息格式

訊息名稱	訊息內容/參數	訊息描述
Trace Request from DS	**Digests, Source MAC address, Source IP address**	**Detect System -> TM(Start)**
Trace Request from TM	**Digests, TM(Start) IP Address**	**TM(Previous) -> TM(Next hop)**
Trace Notice	**Next NMU name, Next TM IP**	**Between TM and TM**
Trace Result	**Trace Result Description**	**TM(Start) -> TM(Next hop)**
Trace Alert	**Exception Description**	**Between TM and TM**

2. 查詢協定

查詢協定是 TM 向 LM 查詢 RS 中之資料庫，以及向 TA 的 TQA 查詢下一個 TM 之用。查詢協定包含 Query Record、Query Result、Query Next NMU、Query Next NMU Result 和 Query Alert 等五種訊息協定，其訊息名稱、內容/參數和描述如表 5-3 所示。

(1) Query Record：TM 以 Query Record 查詢 LM 中 RS 的資料庫。

(2) Query Result：有兩種 Type 的訊息格式，Type 1 表示查無此記錄，而 Type2 則表示已查到該記錄。

(3) Query Next TM：用於 TM 向 TA 的 TQA 單元查詢 BR 的 MAC 所對應的 NMU 名稱/代號及其 TM 之 IP 位址。

(4) Query Next TM Result：表示查到該紀錄，TQA 會將查詢結果回傳給 TM，其中包含有 NMU 名稱及其 TM 之 IP 位址。

(5) Query Alert：若查詢不到，則以 Query Alert 訊息通知查詢者。

表 5-3　查詢協定的訊息格式

訊息名稱	訊息內容	訊息描述
Query Record	**Timestamp**	**TM->RS**
Query Result	**Type1: No such Result Type2: Have such Result, Source MAC Address, Source IP Address**	**RS->TM**
Query Next TM	**Source MAC Address**	**TM->TQA**
Query Next TM Result	**NMU name, Next TM IP Address**	**TQA->TM**
Query Alert	**Query Exception Description**	**TQA->TM**

3. 同步協定

同步協定用於各 TM 向 HP 要求校正系統時間，其包含 Synchronous System Time 和 Synchronous Finish 等兩種訊息格式，其訊息名稱、內容/參數和描述如表 5-4 所示。

(1) Synchronizing System Time：當 TM 向 NTP 伺服器校正完自己的系統時間後，馬上發出一個 Synchronous System Time 訊息給 HP 要求校正系統時間。

(2) Synchronization Finish：HP 在校正系統時間完成後則回應一個 Synchronization Finish 訊息通知 TM。

表 5-4　同步協定的訊息格式

訊息名稱	訊息內容/參數	訊息描述
Synchronizing System Time	**System Time**	**TM->HP**
Synchronization Finish	**Finish**	**HP->TM**

第 6 章 系統實作

本章將探討以前述的系統架構，在實際的網路環境上，以模擬實驗的方式，驗證系統之架構與理論的可行性與正確性。

6.1 實驗環境

實驗平台如表 6-1 所示，以四台主機模擬為四個 NMU 環境。程式撰寫工具為 Borland C++，資料庫以 Oracle 9i 建制。

表 6-1 實驗平台

主機IP	CPU	記憶體	作業系統
140.128.101.102	Intel Pentium 1.8 GHz	512MB	Windows 2000
140.128.101.104	Intel Pentium 1.5 GHz	512MB	Windows 2000
140.128.101.109	AMD Athlon 1 GHz	256MB	Windows XP
140.128.101.224	Intel Pentium 1.4 GHz	512MB	Windows XP

實驗是在 Ethernet 的網路環境上架設，透過 Ethereal 封包監聽工具蒐集封包，為了實驗的單純化，僅彙集 TCP、UDP 和 ICMP 協定的封包。經由 HP 處理並直接進入後端的 Oracle 資料庫，環境架設如圖 6-1 所示。

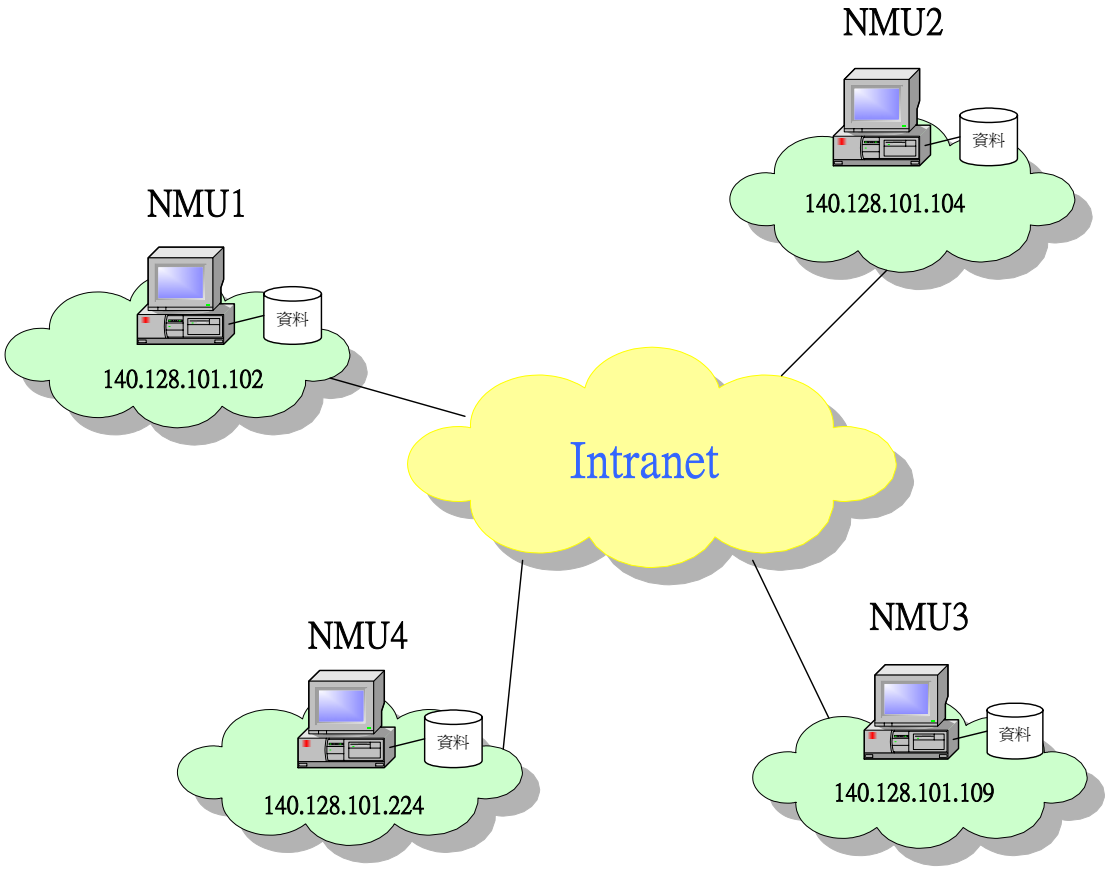

圖 6-1　實驗環境圖

6.2　系統架構

HP 之實作架構如圖 6-2 所示。Ethereal 工具收取送來之 TCP、UDP 和 ICMP 封包的原始資料後，交給 HP 處理，HP 會將原始資料經過四個程序的處理：

1. 格式轉換：將 bit stream 的原始資料轉換成 16 進制資料，以便於 HP 封包拆解與識別。
2. 拆解標頭：將每一個封包之 Ethernet、IP、傳輸層標頭及資料逐一拆解出來，並擷取所需的資訊，包括：追蹤及偵測資訊。
3. MD5 雜湊函數：拆解下來的追蹤資訊交由 MD5 雜湊函數產生 Hashcode（Digests）。
4. PR：每一筆 PR 資料會寫進後端的 Oracle 資料庫，亦預設一功能再將資料送交偵測系統 IDP 處理。

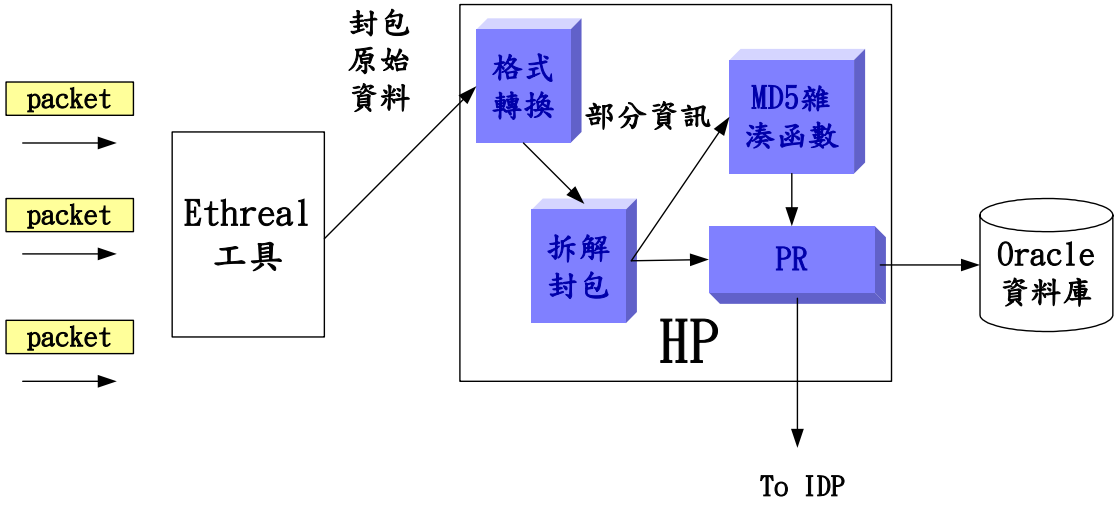

圖 6-2　HP 實作架構圖

TM 的架構如圖 6-3，包括收、發訊息的 Socket Server 與 Socket Client，在此，設定追蹤系統以 Port 8889 作為通訊埠。收到後所發出的訊息會經追蹤演算法判斷，是何種追蹤訊息或追蹤結果。

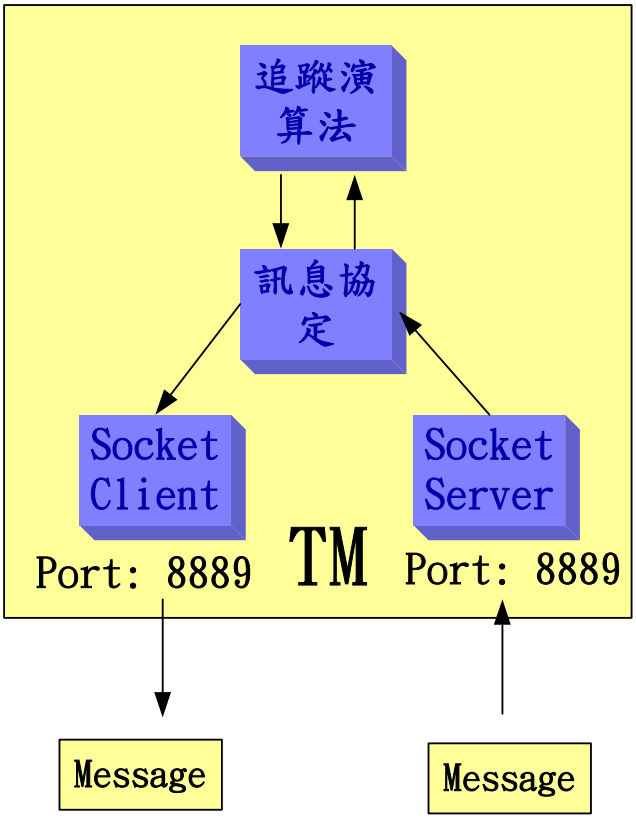

圖 6-3　TM 架構圖

6.3 實驗結果

共做了三個實驗以驗證系統的可行性與正確性。首先，為了能夠模擬封包在網路上傳送，設計了一支程式，具有三個模式，能夠以 Port 8889 發出、轉送與接受訊息。在實驗環境下，以預先設定的路徑，讓 NMU4 發出此訊息給 NMU3，當 NMU3 接到此訊息會轉傳給 NMU2，如圖 6-4 所示。在發出訊息前，每個 NMU 均開啟 Ethereal 工具以接收封包。

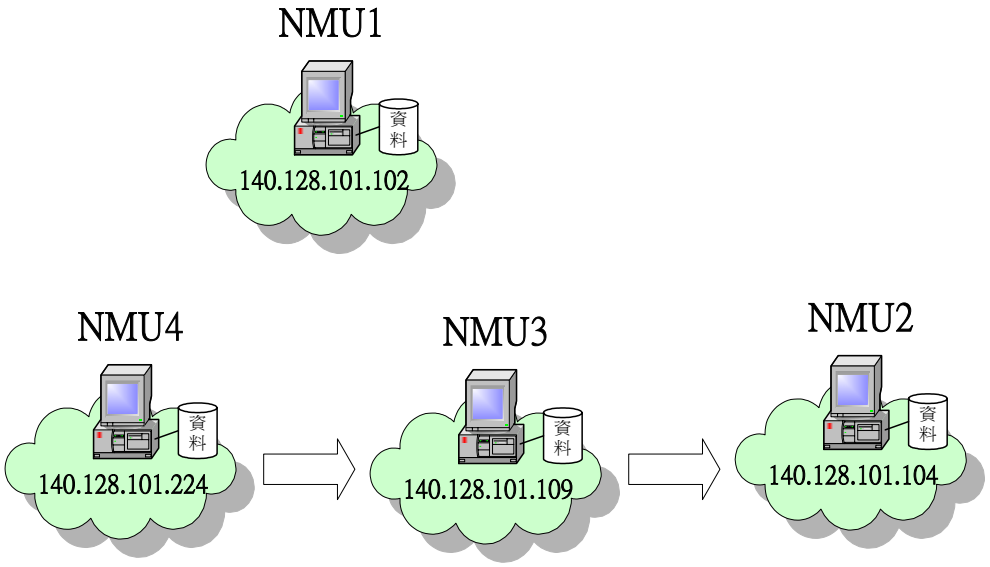

圖 6-4　模擬封包傳送

1. 識別值驗證：首先，驗證追蹤的封包的識別值是否能在 NMU2、NMU3 和 NMU4 識別為同一結果，否則，無法依此作為追蹤。因為是用程式模擬封包傳送，並非實際上同一封包，因此，程式在產生識別值部份要稍作修改，原本要經過 MD5 運算的追蹤資訊，需將其來源及目的 IP 和 Identification 欄位拿掉再作運算(因各自 NMU 產生之封包三者會不相同)。依此修改後，我們分別到 NMU2、NMU3 和 NMU4 的資料庫中查尋，是否有此筆訊息封包。

(1) 在 NMU2 中，封包是由 NMU3 所送來，因此在 NMU2 可以依來源、目的 IP 及所傳送訊息資料，而識別一筆識別值為 201BB9C953E00EEF6489E65E0B3E3DEE 的資料，如圖 6-5 所示。

Row #	HASHCODE	DESTMAC	SOURMAC	TLEN	IDENT	FRAGMENT	PROTOCOL	SOURIP	DESTIP	TYPE
1	BCD83E923E80F2B1A72DC267D96C6E3A	00055DE6AD9A	0000E2706D95	0030	D321	4000	06	8C8065E0	8C80656D	
2	8BD696F02503DB8D116DEEC3387875F1	00055DE6AD9A	0000E2706D95	0028	D322	4000	06	8C8065E0	8C80656D	
3	201BB9C953E00EEF6489E65E0B3E3DEE	00055DE6AD9A	0000E2706D95	0055	D324	4000	06	8C8065E0	8C80656D	

圖 6-5　NMU2 的識別值

(2) 相同的，在 NMU3 中也依此查詢，亦可找到相同的封包有相同的識別值，如圖 6-6，不過可找到兩筆記錄，這是因為以程式模擬時，未將 Foreign Queue 中重複者刪除之故（一筆先由 NMU4 進入 NU3 時記錄，由 NMU3 進入 NMU2 時又記錄之 ）。

Row #	HASHCODE	DESTMAC	SOURMAC	TLEN	IDENT	FRAGMENT	PROTOCOL	SOURIP	DESTIP	TYPE
53	7FFFA74FB93510CD4A3FC44D3AA24963	FFFFFFFFFFFF	000C6E05165E	004E	00BE	0000	11	8C806542	8C8065FF	
54	040658C9087792C59C0765B5D4223759	FFFFFFFFFFFF	0002B332331A	004E	30C1	0000	11	8C8066D6	8C8066FF	
55	201BB9C953E00EEF6489E65E0B3E3DEE	00055DE6AD9A	0000E2706D95	0055	D324	4000	06	8C8065E0	8C80656D	
56	201BB9C953E00EEF6489E65E0B3E3DEE	0040F43A092E	00055DE6AD9A	0055	D74E	4000	06	8C80656D	8C806568	
57	3308B6962B657FDC1E3915F521DEA22D	0000E2706D95	00055DE6AD9A	0028	D74F	4000	06	8C80656D	8C8065E0	
58	7B8FB2B477CD62C4935D708373C3D405	00055DE6AD9A	0040F43A092E	0028	E370	4000	06	8C806568	8C80656D	
59	11F148B3C9E738A30F62DB64AC812A38	FFFFFFFFFFFF	0040F43AA183	00EB	462E	0000	11	8C80663A	8C8066FF	
60	BCF705C74F482678E2D9C4639AE8B457	080020129651	00055DE6AD9A	0049	D750	0000	11	8C80656D	8C806501	
61	48FE5A5954E02D2555DD4130916170A6	00055DE6AD9A	080020129651	009E	AA86	4000	11	8C806501	8C80656D	
62	EFE27D1EC7F836A5C18D72F5EE8CA98B	00D00491FFFC	00055DE6AD9A	004E	D751	0000	11	8C80656D	8C80663A	

圖 6-6　NMU3 的識別值

(3) 最後，在 NMU4 發現亦可找到同一筆識別值，如圖 6-7。

Row #	HASHCODE	DESTMAC	SOURMAC	TLEN	FRAGMENT	PROTOCOL	SOURIP	DESTIP	TYPE	CODE
53	7FFFA74FB93510CD4A3FC44D3AA24963	FFFFFFFFFFFF	000C6E05165E	004E	0000	11	8C806542	8C8065FF		
54	040658C9087792C59C0765B5D4223759	FFFFFFFFFFFF	0002B332331A	004E	0000	11	8C8066D6	8C8066FF		
55	201BB9C953E00EEF6489E65E0B3E3DEE	0040F43A092E	00055DE6AD9A	0055	4000	06	8C80656D	8C806568		
56	A0DBD1D29D397F8B844D81DCEE38EB29	00055DE6AD9A	0040F43A092E	0028	4000	06	8C806568	8C80656D		
57	11F148B3C9E738A30F62DB64AC812A38	FFFFFFFFFFFF	0040F43AA183	00EB	0000	11	8C80663A	8C8066FF		
58	3AEB8673724C37387A6E51F0571D6A18	080020129651	0040F43A092E	0049	0000	11	8C806568	8C806501		
59	0921D1EE317BEAB58A50083970A640B8	0040F43A092E	080020129651	009E	4000	11	8C806501	8C806568		

圖 6-7　NMU4 的識別值

因此，由實驗一可知我們可拿封包的識別值做為追蹤的依據。

2. 驗證非 IP Spoofing 攻擊追蹤：在前述假設下，因為封包最後是由 NMU4 送交給 NMU3，NMU3 再傳送給 NMU2，故理論上應判定為 NMU3 所送來。因此，我們以 201BB9C953E00EEF6489E65E0B3E3DEE 去查看其追蹤結果，從 NMU2（140.128.101.104）開始追蹤後，可發現確實是由 NMU3 傳來，追蹤費時 0.03 秒。

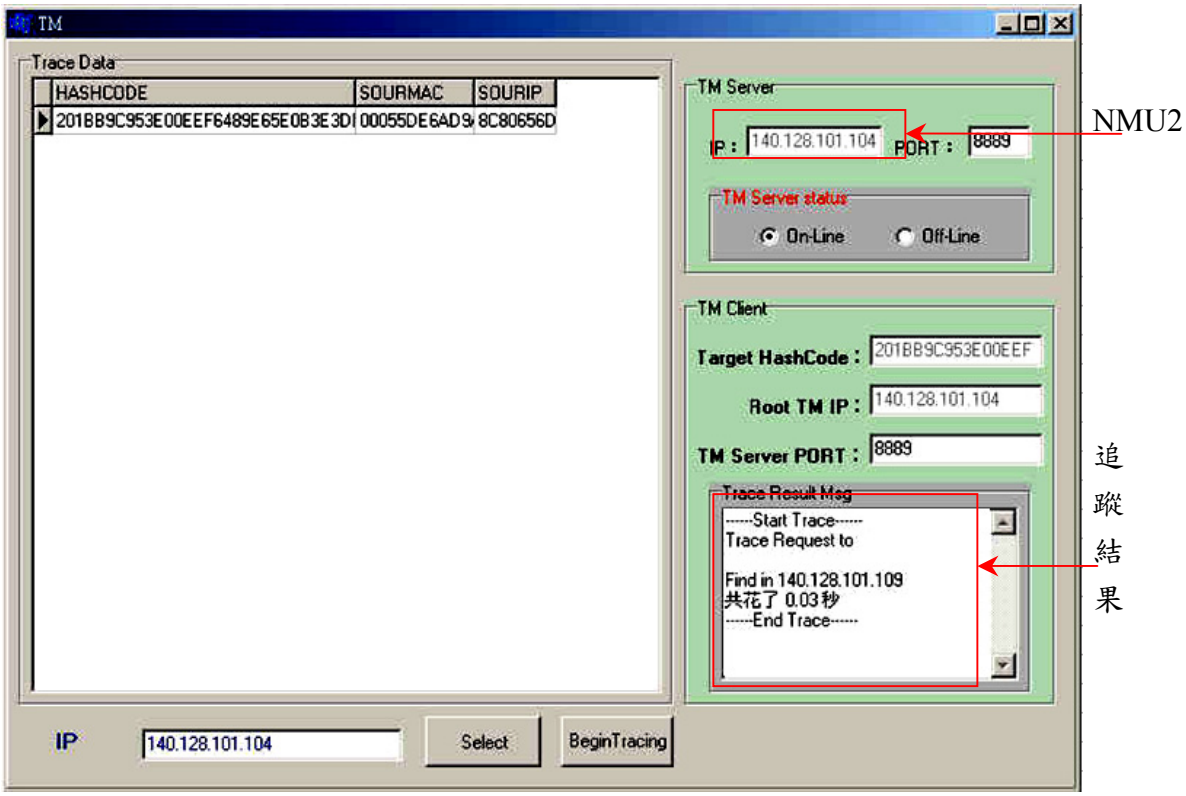

圖 6-8　No IP Spoofing 追蹤結果

3. 驗證 IP Spoofing 攻擊追蹤：在此，我們修改 NMU2 資料庫中的資料，偽造由 NMU1（140.128.101.102）所發送，此時 NMU2 會對 NMU1 發出追蹤要求，並由 NMU1 通知 NMU2 未找到，因此，判定其為一個 IP Spoofing 攻擊。於是進行第二段追蹤，以 MAC 位址做為追蹤條件，查詢 ARP Table（事先建置於資料庫中），找到其對應的來源 IP 後，以 hop by hop 的方式追蹤到攻擊者，最後會找到 NMU4，結果如圖 6-9。

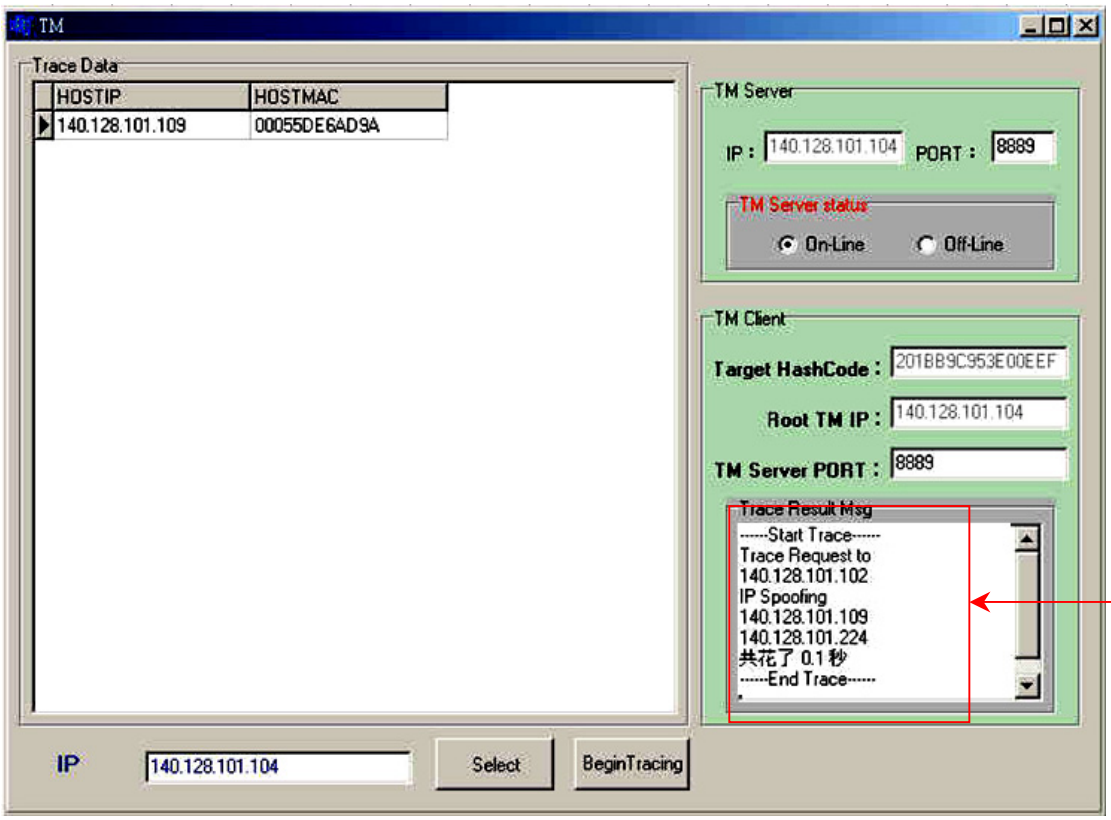

圖 6-9　IP Spoofing 追蹤結果

4. 資料量對追蹤時間的影響：為了觀察資料庫之資料量對整體追蹤時間的影響，故將各個 NMU 資料量由 200 筆增加至 200000 筆做了一測量，如圖 6-10 所示，上方的曲線代表只有一個 hop（No_IP_Spoofing）時的結果，下方則代表三個 hop（IP_Spoofing）的結果。資料量增加至 200000 筆對單一 hop 增加不到 0.16 秒，而對三個 hop 則亦只增加 0.36 秒左右的時間。因此，我們得知資料量對追蹤時間有影響，不過影響並不會很大，整體追蹤的時間亦在可接受範圍。

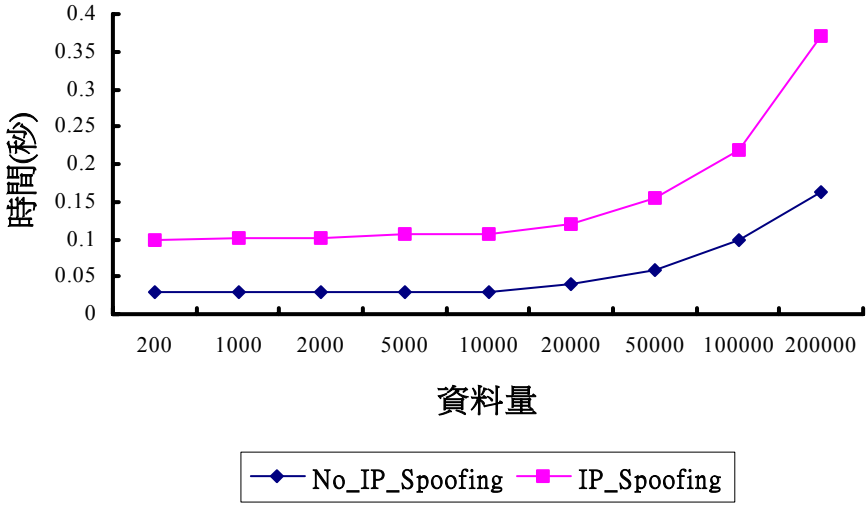

圖 6-10 資料庫資料量對追蹤時間的影響

5. Hop 數與資料量對追蹤時間的影響：Hop 數對追蹤時間的影響可如圖 6-11 所示，圖中曲線由下而上分別代表資料量由 10000、20000、50000、100000 增加至 200000 筆時，hop 數由 1 個增為 5 個時的影響。Hop 數對追蹤時間的影響呈一線性的增加，各個節點的資料量決定曲線的幅度。一般而言，一個 IP 封包到達目的 NMU 之前所通過的路由器數量不會超過 15（TTL），故由圖 6-11 類推所需時間，應仍在可接受範圍中。

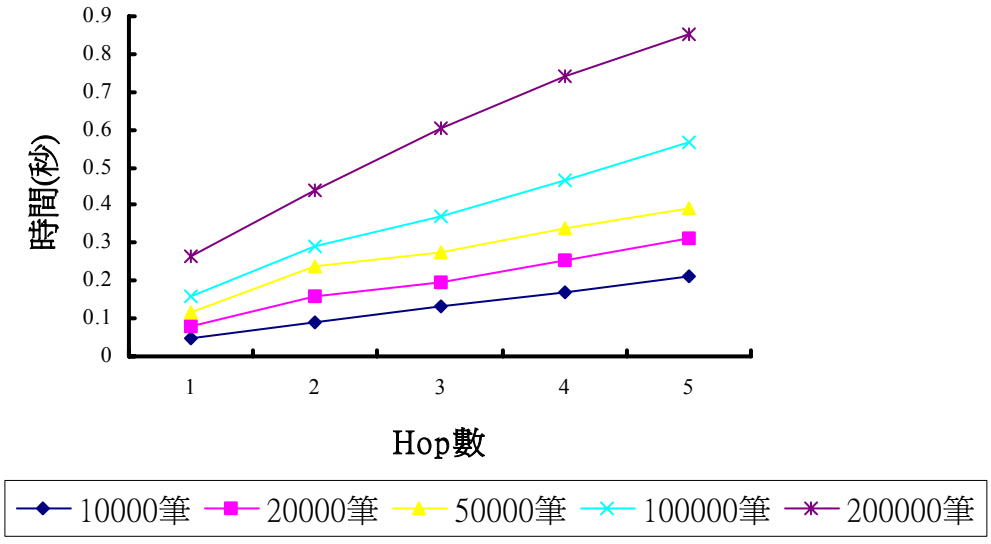

圖 6-11 Hop 數與資料量對追蹤時間的影響

第 7 章 結論

7.1 系統比較

在本 UDIDT 系統擬定後，發現 T. Baba 和 S. Matsuda 的 AMN 系統與本系統於方法上相近，遂在此與其做一分析比較，如表 7-1 所示。

表 7-1 AMN 與 UDIDT 比較

比較	AMN	UDIDT
決定下一個 Hop	必須向周遭與其相鄰的所有AMN區域詢問才能得知,平均要詢問半數相鄰者才能找到下一個AMN	以Source MAC向TQA查詢即可得知
追蹤時間複雜度	O(n),n為AMN之數量	若為IP Spoofing時為O(n),n為NMU之數量 若非IP Spoofing時為O(1)
比對效率	一次查詢需比對7次, 平均一次需比對4.3Bytes,及一次查詢比對30Bytes	一次查詢需比對1次, 需比對8Bytes
儲存體	存放於Memory,可適用於即時追蹤,但不在Memory者則無法追蹤	存放於記憶體與Database中,搭配偵測系統可做即時與事後追蹤
系統安全	無安全機制搭配	搭配DHCP, SSL, 與CA認證, 可保障LM資料安全與傳輸的隱密性,完整性與認證,也因此,傳遞時間比AMN長
硬體成本	成本較低	成本較高,需額外增加一交換器與資料庫

1. 追蹤方法比較：

(1) AMN 和本系統均是以 MAC 為追蹤依據，然而 AMN 並無一機制可告知下一個追蹤的 AMN。然，本系統可透過 TA 之 TQA 協助，在網路拓樸變動時，亦只需對 TQA

重新註冊，而無需更動 TM 資料。

(2) 在前置處理上，AMN 僅需拆解封包，其複雜度為 O(n)，而本系統則在拆解封包後需將追蹤資訊經過雜湊函數的處理，因此，時間複雜度為 $O(n^2)$，然，對追蹤系統而言，雖然本系統前置處理時間之時間複雜度較高，然，處理過後所儲存的追蹤資訊僅不到 AMN 的 1/3；追蹤時間從開始追蹤算起，不包含前置處理時間，僅需考量到追蹤演算法者。若考率到 n 個節點的追蹤，不管有無 IP Spoofing，AMN 的時間複雜度均為 O(n)，然，本系統在 IP Spoofing 時需 O(n)，但是在非 IP Spoofing 只要 O(1)，追蹤複雜度需考量到 IP Spoofing 出現的機率 $P_{ip\_spoofing}$ 與非 IP Spoofing 出現的機率 $P_{no\_ip\_spoofing}$，整體的複雜度為 T(IP_Spoofing)* $P_{ip\_spoofing}$+T(No_IP_Spoofing)* $P_{no\_ip\_spoofing}$，在最差情況下本系統的複雜度才會等於 O(n)。

(3) 在比對效率上， AMN 做一次比對需比對 7 個欄位，每各欄位平均長度為 4.3 個位元組，而本系統僅以識別值作一次比對，識別值的長度為 8 個位元組，因此，本追蹤系統在比對效率上明顯優於 AMN 系統。

(4) 在儲存體上，AMN 是以記憶體作為存放封包資料，雖可作即時追蹤，若網路流量大，記憶體空間有可能會被塞滿，況且，若各 AMN 因流量不同有些節點資料被覆概，而導至資料不一致，可能會因而無法追蹤到正確結果。本系統則透過偵測系統與資料庫的搭配，以偵測系統通知作即時追蹤，以資料庫儲存則不須擔心資料遺失導致資料不一致的問題。

2. 系統安全比較：

AMN 系統並無安全機制保護。本系統則透過 DHCP 機制保護 LM 的資料，並透過認證系統與 SSL 連線，各 TM 在傳送訊息可避免被竊取、修改，也可以避免浪費系統資源在駭客所偽造的追蹤訊息。

3. 本追蹤系統須比 AMN 額外增加一交換器和資料庫，在成本上比 AMN 高。

7.2 結論與未來發展

就追蹤機制而言，目前已提出幾種方法，而真正適用於目前的網路機制少之又少。

因此，本論文希望建構一入侵追蹤機制，能適用於目前網路環境的架構，能迅速且正確的追蹤。因此，提出了一個新的追蹤機制，除了整合各追蹤系統的優點，並增加一些機制以改善其在實作上的不足處，另外，也改進了追蹤方法與加入了安全機制，希望能建構出一個比較完善的追蹤系統。

本追蹤系統的主要特性有：

1. 在管理上，藉由分治的網路管理區域 NMU，能夠有效的蒐集追蹤資訊。
2. 在封包的處理上，各個 NMU 以 LM 處理經過該區域的封包，利用雜湊函數產生其識別值，以作為追蹤的依據，由於只需比對一次而易於比對。
3. 在追蹤方面，透過各 NMU 中之 TM 之間的相互協助，並利用協防的方式找到攻擊來源。對於 DDoS 攻擊，亦能在偵測系統偵測後，並以多路徑的追蹤，找出各個不同攻擊來源。
4. 事先判斷是否 IP Spoofing，以改善追蹤的效率，使得整體追蹤效率小於 O(n)。
5. 透過憑證中心與 SSL 連線等安全的機制，來保護 TM 和 TM 及 TM 與 TA 間的通迅安全與身份識別。另外，透過 DHCP 在架構上保護 LM 內的資料，免於遭受外界竊取或損毀。
6. 透過偵測系統的警示，達到即時的追蹤，亦能透過已記錄的資訊作事後的追蹤。

本系統的限制則有：

1. Mirror Port 的處理速度：若未能及時的將流量導入 LM，則可能會導致紀錄不完全，而在各 NMU 的 BR 前置一具有 Mirror Port 的交換器亦需考量不影響 BR 的功能。
2. HP 的處理速度:HP 在拆解封包時的處理速度亦影響到是否能即時將所有資訊紀錄下

來。

3. 認證、SSL 均須廢時，會影響整體的追蹤時間。

4. 一些安全機制，例如，IPSec、NAT，均限制本追蹤系統的追蹤。

在未來的工作，可朝幾個方向進行：

1. 對追蹤系統的效能與成本做進一步的改善。因為，系統係利用具 mirror port 的交換器複製封包並送交給 LM 處理，而 mirror port 會對 BR 的效能造成影響，其成本也是一大考量。因此，將來將研究如何在路由器中加入 Mirror 封包的功能，在影響路由器效能最低的情形下直接將封包導入 LM，又可節省交換器的成本花費。在 LM 中的封包處理希望亦能透過硬體方式來改善處理效能。

2. 對於追蹤路徑若中斷之問題，目前僅以人工方式向周圍的 NMU 查詢，在未來擬對此更詳加探討，以期使追蹤更有效率。

3. 對於追蹤到攻擊者的事後處理做一規劃。對於找出攻擊者的後續處理，主動的方面，能以法律或社會程序予以制約。根據中華民國刑法第三百六十條（民國九十二年六月三日通過立法程序）「無故以電腦程式或其他電磁方式干擾他人電腦或其相關設備，致生損害於公眾或他人者，處三年以下有期徒刑、拘役或科或併科十萬元以下罰金」，駭客的攻擊行為往往構成刑事責任，因此，追蹤系統在將來可用於警方的蒐證之用，以保障網路使用者其財產安全；在被動的方面，對於正在攻擊的連線可立即切斷其攻擊來源，以減少損害程度，方法是在路由器上封鎖此 IP 或重設該連線。

4. 對於內部攻擊的追蹤，目前尚在規劃如何在追蹤到攻擊者區域後，更有效的追查到內部攻擊者位置，以利於能立即找出現行犯。

參考文獻

[1] S. Savage, D. Wetherall, Member, IEEE, Anna Karlin, and Tom Anderson，"Network Support for IP Traceback"，IEEE/ACM TRANSCATIONS ON NETWORKING, VOL. 9, NO. 3, pp.226-237, JUNE 2001

[2] D. Xiaodong, Adrian Perrig，"Advanced and Authenticated Marking Schemes for IP Traceback"，Proceedings of IEEE INFOCOM'2001 (20th), vol.2, pp.878-886, March 2001

[3] Bellovin，"ICMP Traceback Messages"，Network Working Group Internet Draft，draft-bellovin-itrace-00.txt。

[4] R. Stone，"CenterTrack: An IP Overlay Network for Tracking DoS Floods"，Proceedings of 9th USENIX Security Symposium, pp. 199-212 Aug 2000

[5] A. C. Snoeren, C. Partridge, LA. Sanchez, CE. Jones, F. Tchakountio, B. Schwartz, ST. Kent, and W. T. Strayer, "Single-Packet IP Traceback"， IEEE/ACM TRANSACTIONS ON NETWORKING, Volume 10, Number 6, pp.721-734, December 2002.

[6] T. Baba, S. Matsuda，"Tracing Network Attacks to Their Sources"， IEEE Internet Computing, Vol. 6, No. 2, pp.20-26, March/April 2002

[7] S. Matsuda, T. Baba, A. Hayakawa, and T. Nakamura， "Design and Implementation of Unauthorized Access Tracing System" Proceedings of the 2002 Symposium on Applications and the Internet (SAINT 2002), IEEE Computer Society, pp.74-81, January 2002

[8] S. Northcutt, J. Novak, "Network Intrusion Detection：An Analyst' s Handbook"， New Rider Press

[9] S. P. Shieh, V. D. Gligor, “On a Pattern-Oriented Model for Intrusion Detection”，IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 9, NO. 4, JULY/AUGUST 1997, PP. 661-667

[10] D. E. Denning, “An Intrusion-Detection Model”，IEEE Transactions on Software Engineering Vol. SE-13, no. 2, pp. 222-232, February 1987

[11] E. Lundin, E. Jonsson, “Anomaly-Based Intrusion Detection: Privacy Concerns and Other Problem”，Computer Networks, Vol. 34, 2000, pp. 623-640

[12] R. Housley, W. Ford, W. Polk, D. Solo, “Internet X.509 Public Key Infrastructure Certificate and CRL Profile”，RFC2459, Jan1999

[13] W. Richard Stevens, TCP/IP Illustrated, vol. 1, The Protocols, Addison-Wesley, 2000.

[14] 沈文吉，”網路安全監控與攻擊行為之分析與實作論文”，台灣大學資訊管理研究所碩士論文，2001 年 6 月。

[15] 呂芳懌、蘇俊維、許惟翔，“內部網路安全威脅分析與防制”，2003 電子商務與數位生活研討會，2003 年 4 月 11-12 日。

[16] 閻雪，中國大陸的駭客技術，松崗電腦圖書公司，2000 年 8 月。

[17] 尤焙麟　譯，駭客現形第二版，麥格羅‧希爾，2001 年 7 月。

[18] 賴冠州，駭客入侵偵測專業手冊，旗標出版股份有限公司，2001 年 12 月。

[19] 呂芳懌、楊子逸，“A Host-based Real-time Intrusion Detection System with Data Mining and Forensics Techniques “，第三十七屆國際卡拉漢安全科技年會，2003 年 10 月。

[20] 呂芳懌、鄭真真、洪嘉鴻，“UDAIDTS：以 Hash 為基礎的主動式區域聯防入侵偵測與追蹤系統”，第十四屆國際資訊管理學術研討會，2003 年 7 月 11-12 日。

[21] 巫坤品、曾志光　譯，密碼學與網路安全原理與實務，碁峰資訊，2001 年 9 月。

[22] 張志忠，”中華民國海軍憑證中心之組織架構與建置”，交通大學資訊管理研究所碩士論文，民國 90 年 6 月。

[23] 朱建達，“建立於公開金鑰基礎建設的單一簽入系統”，交通大學資訊科學研究所

碩士論文，民國 89 年 6 月。

[24] 方盈，TCP/IP 通訊協定、入門與應用，博碩文化股份有限公司，200 年 4 月。

[25] http://www.cert.org.tw/，台灣電腦網路危機處理暨協調中心

[26] http://wp.netscape.com/security/techbriefs/ssl.html，Netscape network

[27] http://www.ietf.org/html.charters/dhc-charter.html，Resources for DHCP

[28] http://www.eecis.udel.edu/~mills/database/rfc/rfc2030.txt，ntp.org