國 立 交 通 大 學

資 訊 工 程 學 系

碩 士 論 文

利用資料分享方法以增進網路電話之安全性

A Data Sharing Approach for Secure VoIP Delivery

指導教授：蔡文能　教授

研 究 生：羅京城

中 華 民 國 九 十 三 年 七 月

利用資料分享方法以增進網路電話之安全性

A Data Sharing Approach for Secure VoIP Delivery

指導教授：蔡文能　　Advisors：Wen-Nung Tsai

研 究 生：羅京城　　Student：Ching-Cheng Lo

國 立 交 通 大 學

資 訊 工 程 研 究 所

碩 士 論 文

A Thesis Submitted to

Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Computer Science and Information Engineering

July 2004

Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 三 年 七 月

授權書

(博碩士論文)

本授權書所授權之論文為本人在 交　通 大學(學院) 資　訊　工　程 系所

＿＿＿＿＿＿組 九十二 學年度第 二 學期取得 碩 士學位之論文。

論文名稱：利用資料分享方法以增進網路電話之安全性

A Data Sharing Approach for Secure VoIP Delivery

1.□同意　□不同意

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利的附件之一，請將全文資料延後兩年後再公開。(請註明文號:　　)

2.□同意　□不同意

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

指導教授姓名：　蔡文能

研究生簽名：　　學號： 9117587

（親筆正楷）　　（務必填寫）

羅京城

日期：民國　93 年　7 月　15　日

---

1. 本授權書請以黑筆撰寫並影印裝訂於書名頁之次頁。
2. 授權第一項者，所繳的論文本將由註冊組彙總寄交國科會科學技術資料中心。
3. 本授權書已於民國 85 年 4 月 10 日送請內政部著作權委員會（現為經濟部智慧財產局）修正定稿。
4. 本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

國家圖書館博碩士論文電子檔案上網授權書

本授權書所授權之論文為本人在 交　通 大學(學院) 資　訊　工　程 系所
　　組 九十二 學年度第 二 學期取得 碩 士學位之論文。
論文名稱：利用資料分享方法以增進網路電話之安全性

指導教授：蔡文能

□同意　　□不同意

本人具有著作財產權之上列論文全文(含摘要)，以非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

研究生簽名：　　學號：9117587
（親筆正楷）　　（務必填寫）

羅京城

日期：民國　93　年　7　月　15　日

1. 本授權書請以黑筆撰寫，並列印二份，其中一份影印裝訂於附錄三之一(博碩士論文授權書)之次頁；另一份於辦理離校時繳交給系所助理，由圖書館彙總寄交國家圖書館。

利用資料分享方法以增進網路電話之安全性

學生：羅京城　　指導教授：蔡文能 教授

國立交通大學資訊工程學系（研究所）碩士班

摘　　要

近年來，由於網路傳輸的速率加快，許多需要即時(Real-Time)傳輸的應用程式開始盛行，最明顯的是網路電話(Voice over IP，VoIP)的崛起。VoIP 運用網路傳輸的優點，大幅降低了成本，所以頗受供應商以及用戶的好評。但是原本 IP 網路傳輸的本質爲“盡最大努力傳送”(Best effort)，對於傳輸的品質是很大的挑戰，而且近年來越來越多的網路攻擊行爲，如：阻斷攻擊(Deny of Service)、中間攻擊者程式(Man-In-Middle attack)、竊聽程式(Wiretap)…等，傳輸的安全性也因此受到很大的考驗。

傳統用來保護傳輸安全的方法，例如加解密，需要花費較多的時間在對明文加、解密上，並且對於用來加密的金鑰(Key)需要額外的保護，不適合用於時間敏感度高(Time sensitive)的應用上。本篇論文將視覺加密法(Visual cryptography)和資料分享(Data sharing)的概念運用於 VoIP 傳輸上，希望藉此達到防止他人中途竊聽。在不影響聲音品質下減少延遲時間，並且去除掉傳統加解密方法中，金鑰建立與傳送的手續。加上運用互斥多路徑路由(Disjoint multi-path routing)於分享資料(Share data)上的傳送，提高網路使用效能，以減少網路擁塞的發生，間接減少傳輸延遲，以利於即時應用(Real-time application)的傳遞。

A Data Sharing Approach for Secure VoIP Delivery

student：Ching-Cheng Lo **Advisors：Dr. Wen-Nung Tsai**

Institute of Computer Science and Information Engineering
National Chiao Tung University

ABSTRACT

In recent years, because the speed of network transmission increases, a lot of time-sensitive applications begin to prevail. The most obvious example is the network telephone (Voice over IP). The advantages of voice over IP network are lower cost and flexible. The IP network transmits packet using the “best effort” method, so the quality of VoIP should be taken in consideration by the supplier. More and more network attacks are discovered in recent years, like Deny of Service (DoS), Man-In-Middle attack, and the eavesdropping program (Wiretap).In this thesis, we will focus on the security of voice data in public IP network.

Traditional methods used for secure transmission, such as encryption, spend too much time in data encryption and decryption. Those encryption methods also spend much time in key establishment, key exchange and protection. When using visual cryptography and data sharing methods, we can reduce the influence of those problems that mentioned above. Therefore we use this concept in voice data transmission over Internet to prevent from the wiretap attack, and to transit the voice data through the network more efficiently. In addition, we utilize the disjoint multipath routing algorithm to send data, so we can reduce the occurrence of network hot-spot.

致　　謝

在兩年的努力之下，終於完成了我的畢業論文，中間經過了許多的困難與問題，也受到許多人的幫助，在此一一感謝。首先要感謝的是我的指導教授—蔡文能教授，在碩士班的兩年之間，他給我許多方面得指導，讓我受益良多，也成長了不少，在此十分的感謝。還有感謝我的父母，他們給我一個平穩的環境，讓我能夠在專心的學習，才有今天的我，感謝他們。接著要感謝的是實驗室的學長姐，感謝他們總是在我有疑惑時，給我忠告與建議，謝謝他們。接著感謝實驗室的同學與學弟們，大家一起工作、學習，互相幫忙，讓我有個難忘的碩士回憶，十分的感謝他們。最後，要感謝的人很多，如兩年來的寢室室友們、以前大學的好朋友們...等，在我沮喪或難過時，他們都能即時的給我幫助與支持，謝謝他們。

許多的感覺無法用言語形容，對於上面提到的所有人，在此致上我最深的謝意，謝謝你們。

目　　錄

表　　目　　錄

圖　　目　　錄

一、 緒論

近年來，隨著網路技術的進步，傳輸的速率以倍數的方式成長，再加上網路的架設範圍日增，使得許多對時間敏感度較大的應用(Time sensitive application)，例如網際網路電話(Voice over IP，VoIP)、視訊會議(Network meeting)、網路隨選電視(Video on demand，VOD)…等，也開始使用網際網路來當作傳送媒介。利用封包交換式(Packet-switch)的好處，使數位娛樂生活化的目標更進一步實現。另一方面，因爲 IP 網路本身是以盡最大努力傳送(Best effort)，加上一直以來存在著許多個網路攻擊情形不斷發生，要如何順應即時應用(Real-time application)本身的特性，而能達到品質佳且安全性高的效果，是值得我們研究的課題。

1.1 簡介

自 1995 年以後，網際網路快速的成長、擴大，由原先數十到數千個節點(Node)相連的小型網路，快速的成長到幾千萬以上個節點相連的龐大架構。其超越空間限制的特性，對人類生活造成一次巨大的資訊革命，讓人與人之間的溝通更加的便利與迅速，因此產生了不少的商機與利益。雖然曾經過度相信利潤的存在，使得 2000 年網路產生一次泡沫化(Internet bubble)情形，但是之後網路的應用仍然繼續的向四處延伸，至今已經與每個人的生活息息相關、密不可分了。

一開始，由於技術上的未成熟，傳輸的速率只適合用於對時間延宕不敏感的應用上，例如網頁的瀏覽(HTTP)、資料的傳輸(FTP)、信件的傳達(SMTP)…等應用，因爲時間的延遲(Delay)與延遲振盪(Delay jitter)並不會對於資料接收產生不良的結果，可用於較爲低速的網路狀況，所以很快的在世界上散佈開來，大量的被人們使用。近幾年來，網路技術的逐漸成熟，傳輸的速率也從先前的數十 Kbps，成長到數 Mbps 或是更快，使得對於時間敏感度高的應用(Time sensitive application)逐漸可被接受在一般使用者上，也因爲使用封包交換式(Packet-Switch)傳輸較原先電路交換式(Circuit-Switch)網路更能充份的利用到設備與資源(Resource)，所以被網路提供業者(Internet Provider, ISP)接受且開始廣泛的推廣，此類的應用有網際網路電話、網路視訊、網路電視隨選服務…等。

VoIP 是其中一項受到 ISP 注目的應用，可以取代傳統電話的功能，並且讓資料與聲音在同一個網路上傳輸，減少許多管理上的麻煩，並可以利用到封包交換式網路的好處，讓同一個時間可以接通的電話(Call)量增加許多，挾帶著諸多的好處，網路電話當然會成爲新的業界寵兒。爲了使網路電話能夠運作，兩大組織 IETF 與 ITU-T 分別推出了 SIP、H.323、MGCP、MEGACO…等 protocol，H.323 因爲是最早制定出來的協定，所以目前被廣泛的使用，雖然 SIP 起步較晚，但是因爲簡單與擴充性佳的緣故，備受囑目，MGCP 與 MEGACO 則是爲了與傳統電話相連接而制定的協定。

因爲一開始並不是設計用於即時應用，所以被廣泛使用的 IP network 是使用『盡最大努力傳送(Best effort)』的方法，並不是很適合用在即時傳輸的應用上。加上近年來，跟隨著網路的成長，潛伏在網路上的危機也越愈增多，諸如阻斷攻擊(Deny of Service，Dos)、中間攻擊者程式(Man-In-Middle attack)、竊聽程式(Wiretap)…等，使得傳輸的品質與安全性遭到重大的考驗。雖然用於非即時傳輸的應用已經有許多的解決方法，但是因爲需要消耗更多的處理時間(Process time)，尤其是安全度較高的非對稱加密系統(Asymmetric cryptography system)，所以不太適合於即時的應用上。傳統的加、解密需要對用來加密或解密的金鑰(Key)做建立、傳輸、保護，以及定期更換的動作，如此一來需要耗費更多的時間，一旦金鑰被竊取，更是危險，也不利於即時應用上的傳輸，因此需要一套更有效率，且能達到安全性佳的系統架構，來保障即時應用的傳輸。

1.2 動機與目的

資料分享(Data sharing)與視覺加密(Visual cryptography) [10] 是一系列發展已久，對資料進行處理後，達到安全傳輸不被破解的方法。因爲不需要一開始的溝通步驟，即可開始傳送處理過的資料，相對於傳統加、解密需要對金鑰建立、傳輸、保護的煩雜步驟可以省去，所以我們以此爲出發點，希望建立一套適合即時傳輸的架構。

因爲原先資料分割與視覺加密是運用於非即時傳送資料的應用上，加上對於傳送資料的性質有所不同，所以我們希望能夠針對聲音的特性，加上資料分割本身上的好處，發展出一套有利於即時聲音傳送上的架構與方法，來解決對於即時傳輸上安全性的問題(竊聽…等)。透過運用動態多路徑路由(Dynamic multi-path routing)的方式，由過去許多的研究結果顯示 [2,3,4]，可以達成減低網路擁塞(Network congestion)、並減少傳輸延遲的情形，更利於傳送即時性高的應用。

1.3 論文架構

本論文將以 SIP 爲網路電話的基本架構，參考聲音編碼方法(Voice codec)來幫助資料分享(Data sharing)的方法建立，最後以互斥多路徑路由(Disjoint multi-path routing)的方式傳送，用來解決上述的時間限制與除去金鑰的限制問題，達到資料快速且安全傳輸的理想。此論文的組織架構如下：

『第二章、背景介紹』將簡單介紹一些 voice over IP 相關的資料與通訊協定，以及聲音編碼(Voice codec)、傳統加解密，以及網路安全與 IPSec 協定等。『第三章、相關研究』介紹本論文的相關基礎研究視覺加密與動態路由以及其他的論文研究。『第四章、系統架構與方法』介紹資料分享(Data sharing)用於網路電話的架構與訊息傳送的細節。『第五章、模擬結果』顯示運用論文中提及的架構與方法模擬出來的結果與優劣比較。『第六章、結論與未來發展』介紹論文結論與論文架構未來的擴展可能性與需要增進的部份。

二、背景介紹

在這個章節，我們將介紹這篇論文的背景知識，如：網路電話(VoIP)、聲音編碼(Voice codec)、加密演算法(Encryption algorithm)，以及網路安全協定(IPSec)等。

2.1 網路電話 (Voice over IP，VoIP)

將類比式訊號電話系統轉變成數位式訊號電話系統，並且使用封包交換式傳輸網路來傳送聲音封包是近年來熱門的趨勢，許多歐美網路提供者(ISP)紛紛提出相關的產品。本節將介紹網路電話的基本知識，以及目前較常用的通訊協定與技術。

2.1.1 網路電話簡介

Voice over IP 簡單來說，即是將聲音數位化之後，以一個個封包的方式，透過IP(Internet Protocol)架構的網路傳送到收話端 [1]，相較於先前的電路交換式(Circuit-Switch)有著網路資源運用更加充分，以達到同時可處理更多通電話連線的好處，並且成本更加低廉，所以廣受業界的矚目與推廣。

網路電話的演進可以分成三個階段來看(如圖 1 所示)：

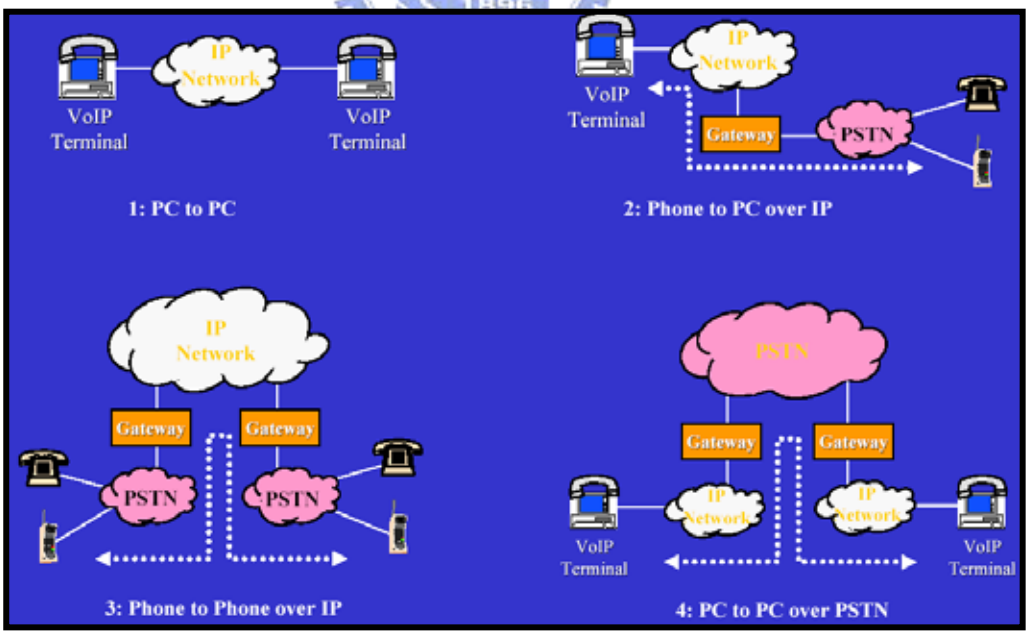


圖 1 網路電話演進圖

(1) PC to PC：最早被開發出來的網路電話系統，利用軟體在 IP 網路上建立傳送聲音封包的連線，可說是最簡單且容易的建構方法。第一個這類型的產品是

由 Vocaltec [38] 於 1995 年 2 月所開發出來的 Internet Phone，這個軟體的需求配備爲 486/33-MHz PC、一張音效卡、麥克風，以及調變解調器(Modem)，可以算是 VoIP 的第一個商業軟體。近年來，這類的產品大行其道，許多的軟體，如 Microsoft 的 MSN messager、Yahoo 的 Yahoo 即時通...等都提供有網路電話的功能。Skype [40] 於 2004 年 2 月推出了第一套以 P2P 爲基本架構的網路電話軟體，以不錯的通話品質獲得廣大的好評。

(2) PC to Phone：由於目前電路交換式的電話系統仍然是分佈最廣泛的系統，所以如何將兩種不同的網路系統互相溝通，使得使用者可以輕易的使用個人電腦打電話到一般的公眾交換網路(Public Switch Telephone Network, PSTN)上，是電話服務業者最爲重視的一個部份。這一類的應用最早由 Net2Phone 這家公司於 1995 年 11 月提出 PC-to-Phone 的網路電話計劃 [39]。1997 年 12 月 ITXC 公司開始大量生產交換機伺服器，將 PC-to-Phone 產品商業化。上一段所提及的 Skype [40] 軟體也同樣具有 PC to Phone 的能力。

(3) Phone to Phone：發展網路電話的最終目標是希望能將聲音與資料系統的整合，以方便管理與控制，所以 Phone-to-Phone 系統的開發就是希望能取代傳統的電路交換式系統。Net2Phone 公司於 1997 年 9 月開始在美國推出 Phone to Phone 透過 IP 網路傳送的服務[39]，許多的長途電話也漸漸的改用透過 IP 網路傳送以節省成本，可說是未來的趨勢與主流產品。2002 年 10 月 18 日，美國知名電話公司 AT&T 向 FCC 提出要求，希望 FCC 對 Phone-to-Phone 的 VoIP 規劃出一套合法的管理規則，但是 FCC 認爲 VoIP 是一套實驗與發展中的技術，所以對 AT&T 的提議持保留態度[41]。2004 年 2 月 12 日，FCC 開始注意到現行 VoIP 產品的架構與管理混亂，各家業者的實作方式皆有所不同，於該日起，開始著手訂定 VoIP 相關的架構與管理規則，期望使所有 VoIP 業者的實作有一定的規則，達到公用網路管理的目標[42]。

目前網路電話的服務以歐美國家推出的較多，因爲擁有較便宜的特性，頗受消費者的歡迎，所以每年成長也有一定的幅度，我們可以從 Allied Business Intelligent Inc.(ABI) 於 2003 年對 VoIP 做的研究與推測(圖 2)中看出，未來 VoIP 市場的成長幅度是樂觀的。

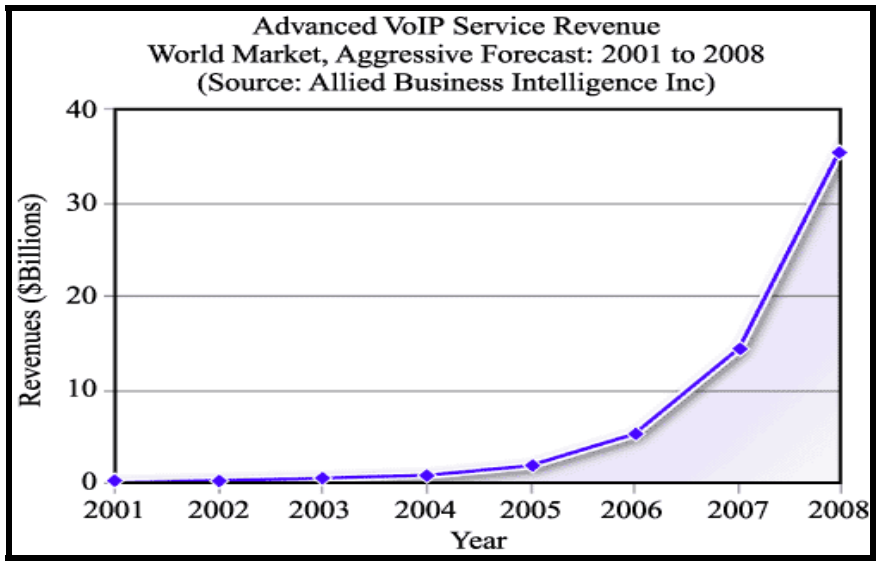


圖 2　世界網路電話市場成長預測圖
(資料來源：Allied Business Intelligence Inc)

同樣的，雖然亞洲的市場因爲電信業開放的腳步較爲緩慢，網路電話(VoIP)成長的速度相較於歐美國家來的遲緩，但是許多的調查與研究都顯示亞太地區網路電話的成長都是可以樂觀預期的，這方面可從 IDC 於 2002 年對亞太地區做的研究與預測(圖 3)可看出，成長的可能性。

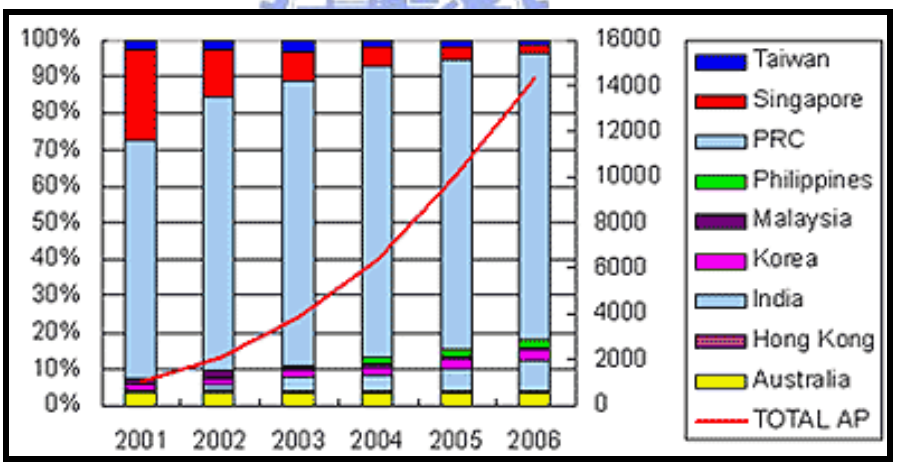


圖 3　亞洲網路電話成長預測圖
(資料來源：IDC)

2.1.2　網路電話通訊協定

爲了可以順利的在公眾的 IP 網路上建立連線並傳送聲音的封包，一套公認的通訊協定(Protocol)是必要的。網路上兩大網路通訊協定訂定組織：ITU (International Telecommunication Union)與 IETF(The Internet Engineering Task Force)分別都訂定了適用於網路電話連線建立與中斷的通訊協定，到目前爲止共有四個較爲廣泛使用的通訊協定，在此做簡單的介紹。

(1) RTP [14]：Real-time Transport Protocol，由 IETF 於 1996 年 1 月定義在 RFC1889，是一個用來傳送即時應用的通訊協定(包括聲音、影像…等)，定義有 RTP 封包標頭(Packet Header)與所傳送媒體相關的資訊…等。和 RTCP(RTP Control Protocol，RFC1890)搭配使用，是一套專門處裡媒體傳送的通訊協定，藉以和訊號封包分開傳送。

(2) H.323：Packet-based Multimedia Communications Systems，1995 年 3 月 28 日由 ITU-T 正式訂爲標準，是一套包含其他通訊協定的封包式多媒體通訊系統 [31] ~ [33]。因爲定義的較早，演化的版本也很多(於 2003 年 7 月出版第五版)，所以許多的系統都採用 H.323 的架構開發，如 Microsoft 的 NetMeeting 即是採用 H.323 的架構。H.323 最大的缺點爲系統龐大，內部結構複雜。

(3) SIP：Session Initial Protocol，於 1999 年 3 月定義於 RFC 2543，再版定於 RFC 3261。爲一套點對點(Peer to peer)以及主從式(Client/Sever)的傳輸架構，與 SDP(Session Description Protocol, RFC 2327)、RTSP (Real-Time Streaming Protocol, RFC 2326)、SAP (Session Announcement Protocol, RFC 2974)合爲 IETF 多媒體資料與控制架構(Multimedia data and control architecture) [15] ~ [17]。SIP 最大的好處即爲簡單、彈性佳，適合用於智慧型掌上產品的開發。

(4) MGCP：Media Gateway Control Protocol，於 1999 年 10 月由 IETF 定義在 RFC 2705，爲一套主從式(Master-slave)架構的通訊協定 [1] [19] ，運用於 Media Gateway Controller(MGC)與 Media Gateways(MGs)之間所傳輸的訊息規範上(請見圖 4)。MGCP 是一套純知識性的通訊協定，需要其他的通訊協定來詳細定出實作架構，Megaco/H.248 就是根據 MGCP 爲基礎所定義出的通訊協定。

(5) Megaco/H.248(Gateway control protocol)：爲 IETF(MEGACO)與 ITU-T(H.248)於 2000 年 11 月共同研究開發的通訊協定 [20] [34] ，與 MGCP 的用途類似，用於 MGC 與 MGs 的通訊上，亦爲主從式(Master-slave)的架構。

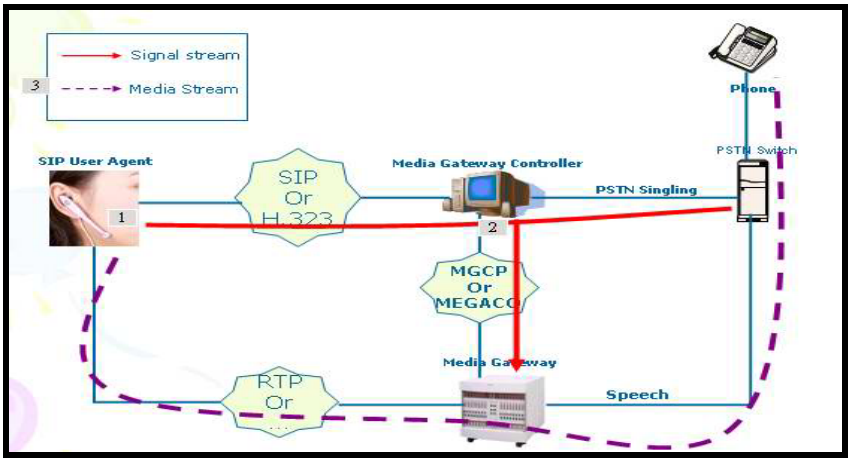


圖 4 各個 VoIP 通訊協定運作分工情形

以上爲目前用於網路電話的五個通訊協定，其中 H.323 與 SIP 主要用在通訊會期(Session)的建立與中止，以及媒體傳送的管理上。MGCP 與 MEGACO/H.248 主要是幫助網路電話與原先 PSTN 架構的相容性更佳所設計出來的。接下來的章節我們將特別介紹 SIP 架構的細節部份。

2.1.3 Session Initial Protocol (SIP)

Session Initial Protocol，簡稱 SIP [15]，是由 IETF 於 1999 年訂定出來的應用層(Application-layer)通訊協定，主要是爲了多媒體通訊的建立、修改與中斷而訂定。其最大的特點爲簡單、靈活度高以及適用於智慧型裝備。底下我們將分架構與訊息交換兩大部份來介紹 SIP。

2.1.3.1 SIP 架構

SIP 爲一個主從式(Client-server)的架構，由用戶端(Client)發出需求(Request)，伺服器端(Server)接收到需求後，因情形發出適當的回應(Response)給用戶端，底下我們分別介紹 SIP 架構中的每個實體(Entity)：

(1) Client：又稱 User agent client，爲一個應用程式，用來發出 SIP 訊息給伺服器或是其他的用戶。

(2) Server：主要用來對用戶端的需求做出正確的回應用，依功能可分成四類。

1. Proxy server：作用就像一般區域網路(LAN)中的代理伺服器(Proxy)一樣，將從用戶端收到的需求傳到正確的伺服器或用戶，可分成有狀態性(Stateful)或是無狀態性(Stateless)。

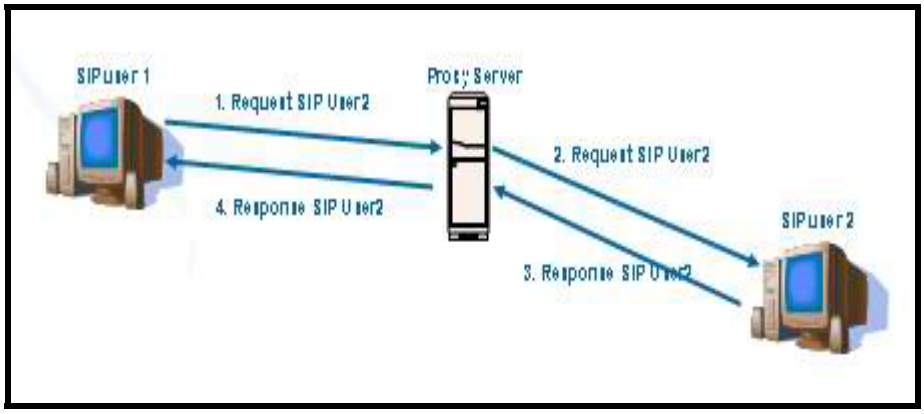


圖 5 SIP 代理伺服器(SIP proxy server)

2. Redirect server：從用戶端收到需求，依據伺服器資料庫內的紀錄，得到用戶查詢的對應位置，回傳給用戶端，此伺服器無法發出 Invite message 給其他的實體(Entities)。

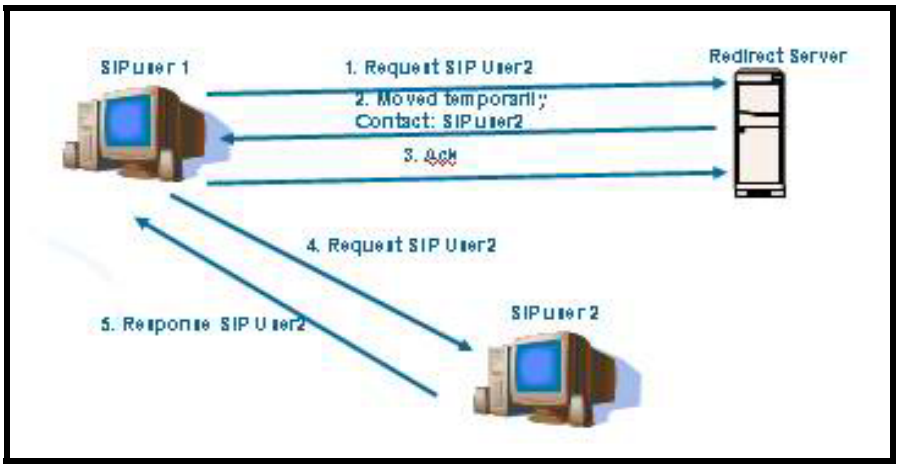


圖 6　SIP 位置轉向伺服器(SIP redirect server)

3. Register server：接受從用戶端發出的 register 訊息，讓用戶註冊於伺服器上，包含用戶端的一些資訊如：位置、暱稱…等，通常會與 proxy server 以及 redirect server 合併在一起使用。

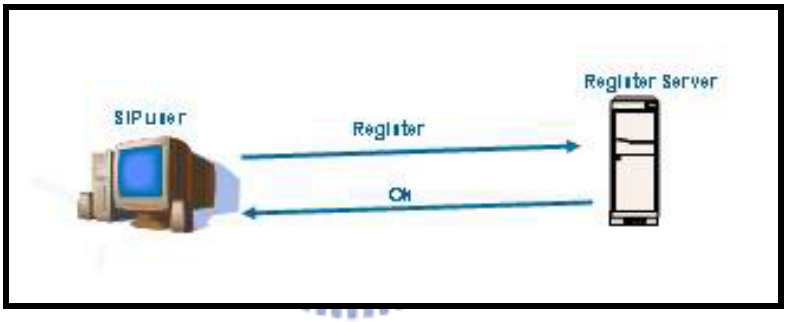


圖 7　SIP 註冊伺服器(SIP register server)

4. User agent server：接收從使用者(User)來的需求，並做出回應(發出需求到伺服器端，以得到回應)，通常與 user agent client 合併成一個裝置(Device)，稱做 User agent(UA)。

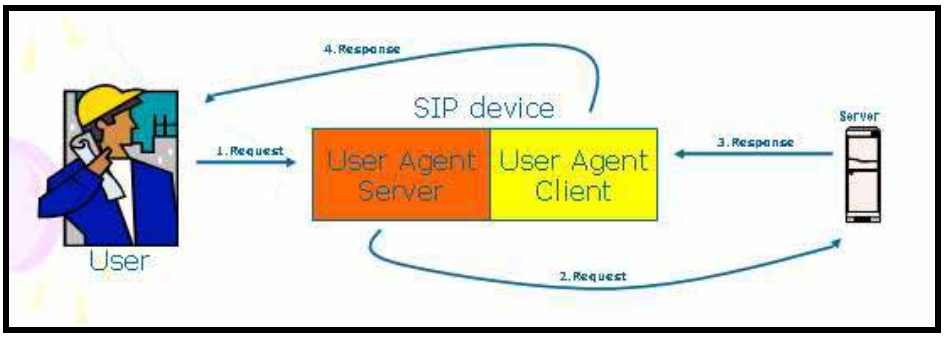


圖 8　SIP 使用者代理人(SIP user agent)

2.1.3.2 SIP 訊息傳遞

SIP 是一個以文字爲基礎(Text-base)的通訊協定，使用 ISO 10646 定義的字元，並使用 UTF-8 的方式編碼，所以 SIP 的訊息看起來類似於 HTTP 的訊息。用文字基礎傳送的缺點是與使用二進位方式表示的訊息相較，較占傳輸頻寬。

SIP 的需求(Request)訊息可以分爲六個基本類型：

(1) Invite：用來發出邀請、開起連線的訊息。

(2) ACK：用來確認最後的回應(Response)已經收到。

(3) BYE：用來結束連線用的訊息。

(4) OPTION：用來詢問伺服器的負載量(Capacity)等訊息。

(5) CANCEL：用來停止一個暫時行的需求(Request)。

(6) REGISTER：由 User agent client 發出，做爲登入(Login)或是註冊用的訊息，一個用戶可以對許多的伺服器註冊；對同一個伺服器也可接受同一個用戶多次的註冊動作。

底下我們用一個簡單的例子來介紹 SIP 建立一通連線的訊息交換：

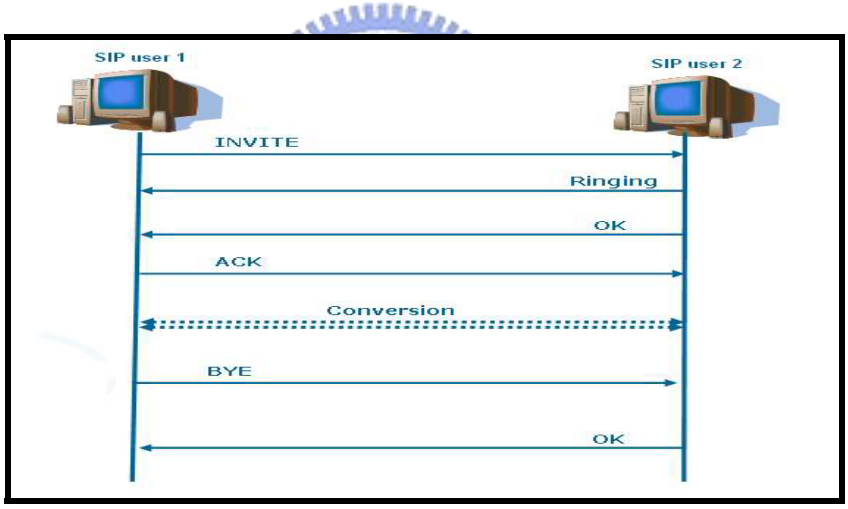


圖 9 簡易 SIP 通話訊息交換圖

首先由發話端(Caller)送出 INVITE 的訊息給收話端(Callee)，當收話端收到訊息後，會先發出鈴響，並傳回 Ringing 的訊息。當收話端拿起話筒回應時，會送出 200 OK 的訊息，發話端收到後，傳送 ACK 訊息，則通話正式建立。當通話結束時，任一端發出 BYE 訊息，由對方收到並回覆 200 OK 後通話正式結束。

2.2 聲音編碼 (Voice codec)

以往傳送聲音的傳送都是藉由類比訊號(Analog)，如果要使用 IP 網路傳送的話，便必須將類比訊號轉爲數位訊號，所以需要一套轉換的方式。聲音的編碼方式因爲不同的

需求，所以發展出許多的方法，底下我們將針對我們所需要的網路傳輸聲音編碼方法做進一步的介紹。

2.2.1 聲音編碼簡介

爲了要使聲音封包在網路上傳輸更加的方便，所以對於頻寬需求有一定的限制，同時也要兼顧到聲音品質的好壞。爲了有個評斷的標準，ITU-T 在 recommendation P.800 [35] 中提出了 MOS(Mean Option Score)的概念，將聲音由好到壞分成 1 到 5 這 5 個等級(MOS 5 品質最佳，MOS 1 品質最差)，由至少 30 個人聽過編碼後的聲音，給出分數後平均而得，是目前對聲音編碼評鑑的一個標準。(通常可以商業化的聲音編碼格式需要有 MOS 4 以上的分數)

將聲音由類比式轉爲數位式(Analog to Digital)的方法大致上可分爲三類，於底下一一介紹：

(1) Waveform codec：爲最簡單的一種聲音編碼方式，使用此編碼方法可得到不錯的聲音品質，但是需要一定量以上的傳送頻寬。編碼的過程大致上可分爲兩個步驟：

1. Sampling：將輸入的聲音波型從類比方式轉變成爲數位表示的過程，如下圖所示：

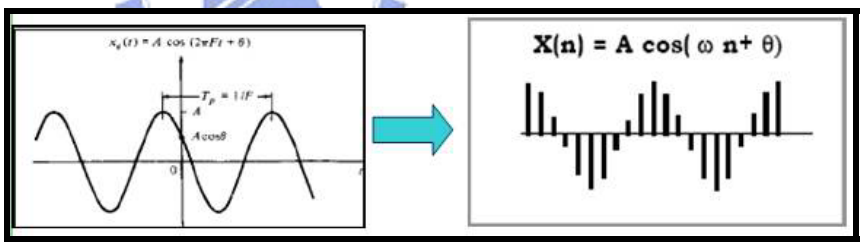


圖 10 聲音取樣示意圖(Voice sampling)

至於要每秒要取多少個樣本(Sample)數，目前最廣泛被使用的理論是 Nyquist sampling theorem，這個理論說明取樣頻率(Sampling rate)需要是原來輸入頻率的兩倍以上，才能在轉回類比時不失去原有的波型。

2. Quantization：決定要使用多少位元(Bits)來表示一個樣本，因爲無法完全表示每種樣本的數值，是故會產生所謂的量化誤差(Quantization error)，對於振幅越大的聲音，量化誤差影響越小，例如：樣本值爲 11.2，量化爲 11，則量化誤差比爲 $0.2/11 \approx 1.82\%$，較樣本值爲 2.2，量化爲 2 的量化誤差比($0.2/2 \approx 10\%$)來的小。量化(Quantization)可依據每個量化步驟(Quantization step)的差異分爲

均勻量化(Uniform quantization)與非均勻量化(Non-uniform quantization)。均勻量化顧名思義就是每一個量化步驟間距都相同。而非均勻量化在量化步驟較小時會用較小的間距。

(2) Source codec (Vocoders)：Source codec 試圖將輸入的訊號，對應到相對的數學模組，利用聲音被製造的原理來模擬，通常使用線性的聲道預測濾波器來處理，可以達到很低的位元率(Bit rate)，但是聲音普遍聽起來很明顯的是合成出來的結果，品質並不佳，就算提高位元率(Bit rate)來儲存也不會有明顯的效果與改善。

(3) Hybrid codec：Hybrid codec 試圖結合上述兩種方法的好處，有 waveform codec 高品質的好處以及 source codec 低頻寬的好處。

下圖表示上面三種 codec 的位元率(Bit rate)與 MOS 的關係圖。

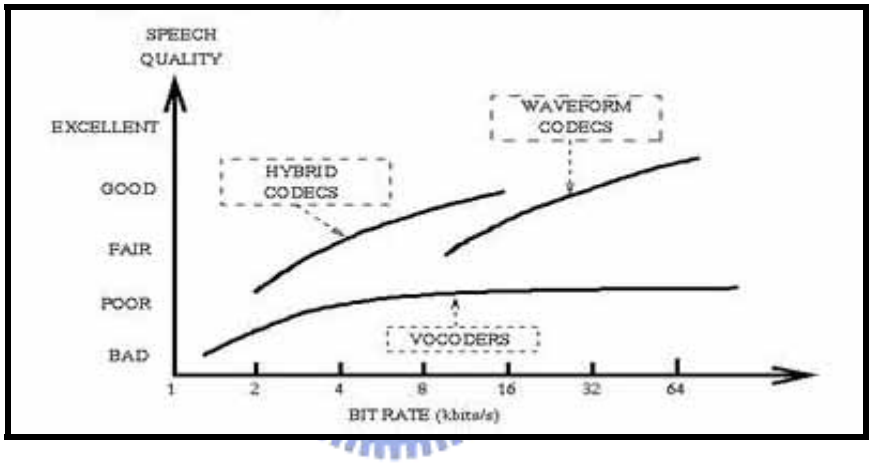


圖 11 各種聲音編碼方法的位元率(Bit rate)對聲音品質的影響圖
(資料來源：[1])

底下我們將介紹幾種常見用於即時應用於網路傳輸的 voice codec，都是定義在 ITU-T 的 G.7xx 系列中，其中 G.711 與 G.726 的觀念將會應用於論文中資料分享(Data sharing)處理的部份。

2.2.2 ITU-T G 系列 Codec

ITU-T G 系列是定義一些在數位網路上傳送與多媒體部份的組織架構(Transmission systems and media, digital systems and network)，而 G.7XX 定義有關聲音編碼方式以及傳送的相關資訊，底下我們將介紹其中四種：G.711、G.723.1、G.726、G.729 voice codec。

2.2.2.1 G.711

G.711 [27] 又稱做脈衝編碼調變(Pulse Code Modulation，PCM)，是目前最普遍最常用的聲音編碼方式，採用 waveform codec 方式編碼，MOS 約在 4.3 左右。因爲人類說

話的頻率分布在 300 到 3800Hz，依照 Nyquist sampling theorem，取樣頻率(Sampling rate)應該要大於 7600Hz 以上，G.711 所採用的取樣頻率爲 8000Hz。Sampling 後再做兩階段的量化(Quantization)處理，先使用 12 bits/sample 的均勻量化(Uniform quantization)處理，然後在對 12 bits 的樣本做非均勻量化(Non-uniform quantization)減化成每個樣本用 8 bits 儲存，所以 G.711 的位元率(Bit rate)爲 8000 * 8 = 64Kbps。

G.711 按照第二次量化(Quantization)的方法不同，可以分爲 A-law 與 μ-law 兩種。A-law 用在北美與日本等國，而 μ-law 則是被歐洲各國所採用。除了第二次量化(Quantization)的方法不同之外，A-law 在傳輸時會將資料中的偶數位元(2,4,6,8)反轉(Invert)。下面爲 A-law 與 μ-law 在兩次量化的列表資料：

表 1　G.711 編碼表

NUMERICAL VALUE	Bit Number	
	Mu-Law 12345678	A-Law 12345678
+127	10000000	11111111
+ 96	10011111	11100000
+ 64	10111111	11000000
+ 32	11011111	10100000
0	11111111	10000000
0	01111111	00000000
- 32	01011111	00100000
- 64	00111111	01000000
- 96	00011111	01100000
-126	00000001	01111110
-127	00000000	01111111

(資料來源：[27])

2.2.2.2　G.723.1

G.723.1 [28] 是 ITU-T 於 1999 年所提出的雙位元率(Dual bit rate)聲音編碼方式，可以產生位元率爲 5.3 Kbps 或 6.3 Kbps 的編碼方式，位元率較高的編碼方式可以得到較佳的聲音品質。一般而言，G.723.1 的 MOS 分數大約在 3.8 附近，是種頻寬需求低且聲音品質不錯的編碼方法。

G.723.1 是對取樣頻率(Sampling rate)爲 8000 Hz，16 bits/sample uniform quantization 的 PCM 輸入做處理。G.723.1 的編碼器(Encoder)每次處理 240 個樣本的區塊(Block)或框架(Frame)，即每個框架的時間長短爲 30 *ms*，加上 G.723.1 的編碼器會有向前參照(Look-ahead)的時間延遲 7.5 *ms*，所以 G.723.1 在聲音編碼上平均會有約 37.5 *ms* 的延遲。

由於 G.723.1 亦屬於 CELP 編碼一族，它擁有傳統參數編碼法和波形編碼法的優點，以一線性預測濾波器(Linear Prediction filter)來表示語音訊號的短時距(Short term)特性之

數學模型，並以基頻預測方式 (Pitch prediction)來找出語音訊號的長時距 (Long term)之類週期，再加上諧波雜訊整型器(Harmonic noise sharping filter)，過濾訊號中之雜訊的類週期性；最後，再以多重脈衝(Multi-pulse)訊號來逼近殘餘訊號，進行量化編碼。

G.723.1 的聲音編碼與解碼方式如圖 12 所示。首先將每個框架經過高通濾波器(High-pass filter)移除 DC 元素(Component)，接著分成四個子框架(Subframe)，每個子框架含有 60 個樣本資料，然後將每個子框架經過數個不同的操作，用來決定適合的過濾器(Filter)參數。最後使用算術運算碼激發線性預測(Algebraic Code-Excited Linear Prediction，ACELP)得到位元率爲 5.3 Kbps 的聲音編碼結果，或是使用多重脈衝最大可能之量化法(Multi-Pulse Maximum Likelihood Quantization，MP-MLQ)得到位元率爲 6.3 Kbps 的聲音編碼結果。

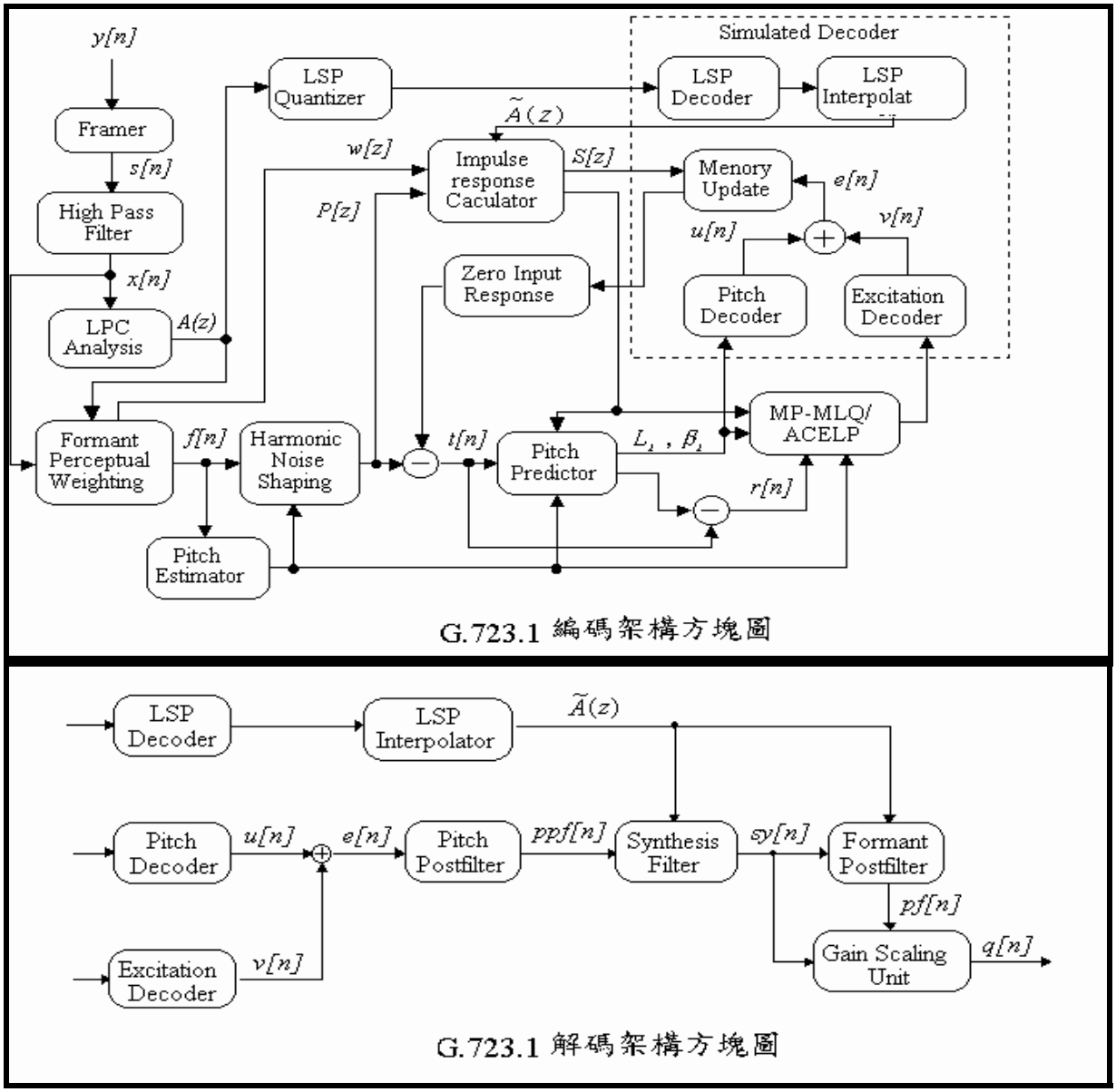


圖 12　G.723.1 編碼與解碼架構方塊圖

(資料來源：[43])

2.2.2.3 G.726

因爲 Pulse Code Modulation(PCM)編碼後所需要的傳輸頻寬(64Kbps)仍然不讓人滿意，所以許多方法便被研究發明來減少傳輸需要的頻寬。DPCM(Differential PCM)是將要傳送的 PCM 封包減掉前一個封包後，傳送兩者的差異值，因爲連續兩個樣本(Sample)的差異值通常不大，所以可以用較少的位元(Bit)來表示兩者的差異值，因此減少了傳輸所需要的頻寬。ADPCM(Adaptive differential PCM)利用前一個 PCM 資料 t-1 推測目前 PCM t 的預測資料t'，然後傳送預測的資料與原本的資料差$(t-t')$，如此一來只要選擇準確性高的預測演算法，所得到的$(t-t')$值較小，即可得到不錯的傳輸頻寬縮減效果。

G.726 [29] 就是採用 ADPCM 的方法來將聲音編碼，他取代了原先的 G.721 編碼方式(也是使用 ADPCM 的編碼方式)。將輸入的 A-law 或 μ-law 資料，做上述 ADPCM 的處理後，輸出成四種聲音格式，分別爲 16、24、32 或 40Kbps(分別爲 2 bits/sample、3 bits/sample、4 bits/sample、5 bits/sample)此四類，在 32Kbps 的編碼結果可以得到 MOS 4.0 的效果，可以算不錯的編碼方式。

下圖爲 G.726 encoder 端的流程圖：

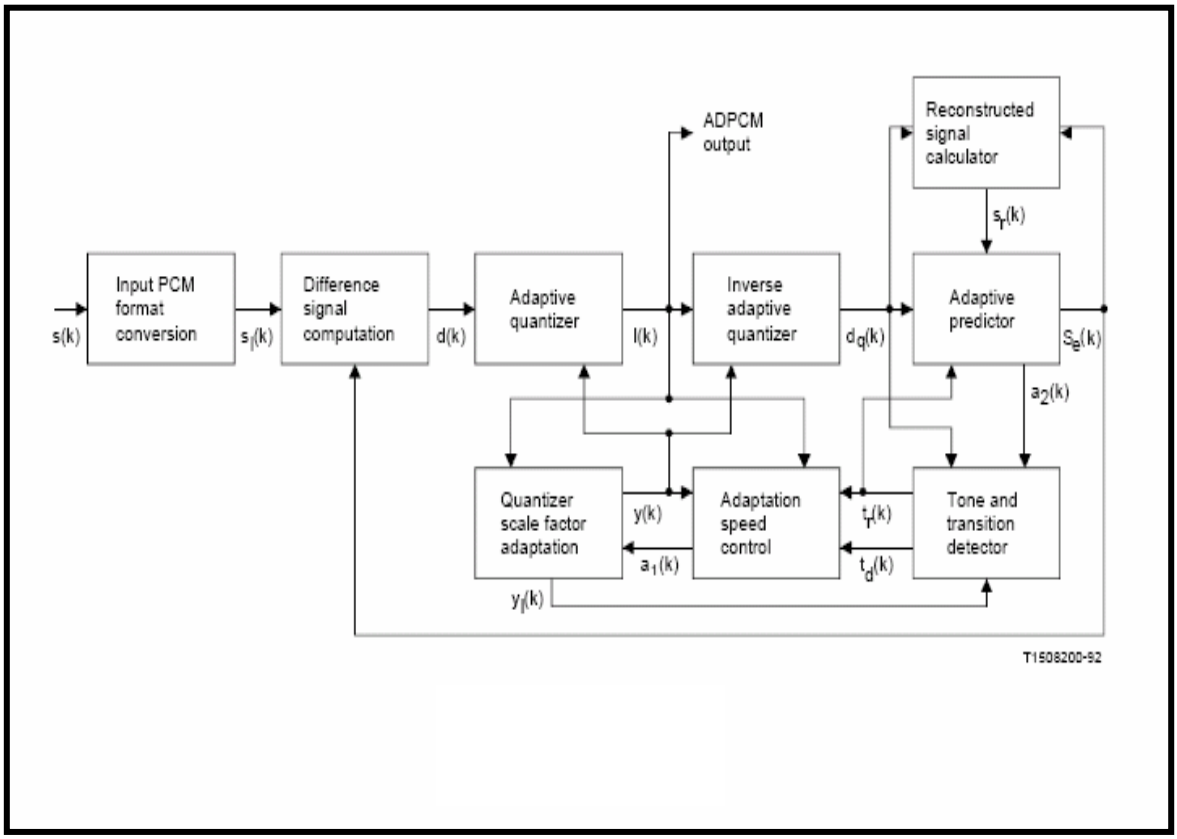


圖 13 G.726 編碼架構方塊圖
(資料來源：[29])

2.2.2.4 G.729

ITU-T Recommendation 中的 G.729 [30] 規定了聲音編碼爲 8Kbps，其所使用的是 Conjugate-Structure-Algebraic-Code-Excited Linear Prediction (CS-ACELP)的預測方法，用以幫助對所需頻寬的壓縮。G.729 的輸入爲 16 bits/sample 的 PCM 格式，每次處理爲一個框架(Frame)，每個框架(Frame)的時間長短爲 10 *ms*，即每次處理 80 個樣本(取樣頻率(Sampling rate = 8000Hz)，處理每個框架(Frame)所需要的時間約爲 15 *ms*，最後輸出 80 bits/frame，所以 G.729 傳輸所需的位元率(Bit rate)爲 80 (Bits/frame) * 10 *ms* = 8Kbps。表 2 爲 G.729 每個框架中，bit 分配的詳細表格，而圖 14 爲 G.729 block encoder 的示意圖：

表 2 8kbit/s CS-ACELP 演算法的位元配置表 (10 *ms* frame)

Parameter	Codeword	Subframe 1	Subframe 2	Total per frame
Line spectrum pairs	L0,L1,L2,L3			18
Adaptive-codebook delay	P1,P2	8	5	13
Pitch-delay parity	P0	1		1
Fixed-codebook index	C1,C2	13	13	26
Fixed-codebook sign	S1,S2	4	4	8
Codebook gains (Stage 1)	GA1,GA2	3	3	6
Codebook gains (Stage 2)	GB1,GB2	4	4	8
Total				80

(資料來源：[30])

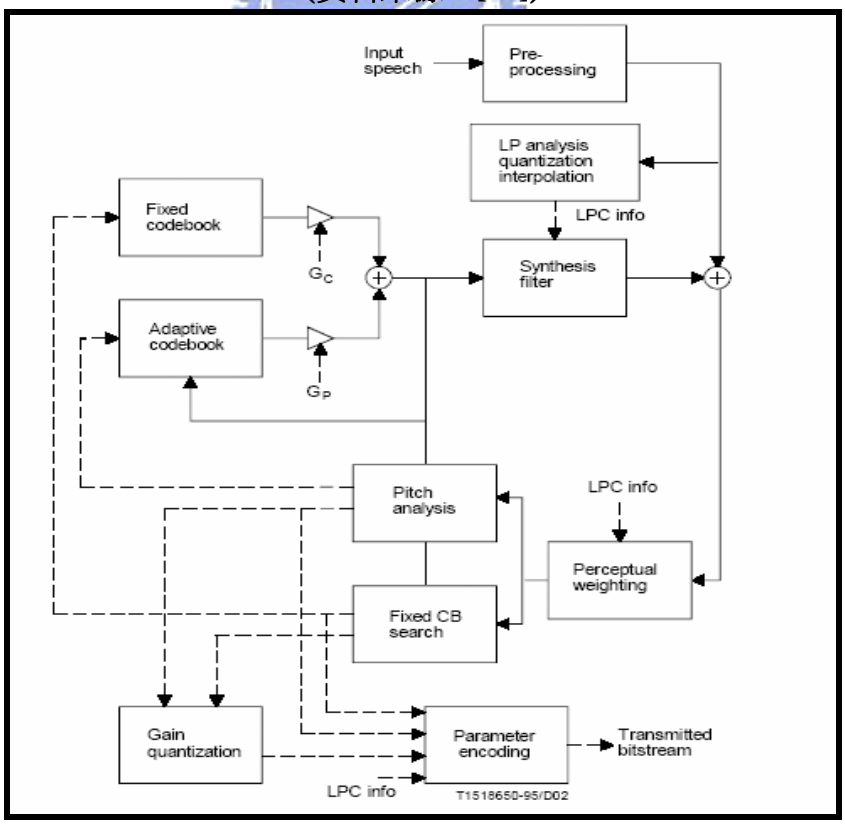


圖 14 G.729 編碼架構方塊圖
(資料來源：[30])

2.3 加密演算法 (Encryption Algorithm)

自古以來，對於傳送內有重要機密的文件時，爲了不讓其他人得知內容，許多的方法被產生來對文件內容做處理，使得其他人無法從處理過的文件看出原本的資料，尤其在戰爭時期，用在傳送軍事訊息格外的重要，如何安全不被視破的下達軍事命令，這些安全上的處理格外的重要，最有名的例子就是凱撒大帝所使用的加密方法。在這個章節我們將簡介密碼系統以及說明一些近期常用的加密演算法。[36]

2.3.1 密碼系統 (Cryptography System) 簡介

隨著網路的發達，許多資訊的傳輸都藉由網路來傳送，其中不乏許多重要的資訊，如商業機密、個人基本資料、國家機密文件…等，由於這些資料的價值很高，所以也有不少的人覬覦這些資訊，希望截取這些資訊後從中獲得利益，因此如何達到網路安全的話題，一直是近年來的重要議題。

將資料加密後傳送，是最常被大家所使用的方法，不論是簡單的對文字亂序排列，或是使用加密金鑰(Key)的方式對資料處理後傳送，都屬於密碼系統這類型的範疇。一般而言，密碼系統包含五個構成要素(M：明文，C：密文，K：加密金鑰，E：加密演算法，D：解密演算法)，並且需要符合下列三項需求：

1. 加、解密使用的演算法必須有效率的適合各種加密金鑰。
2. 系統必須能簡單的使用。
3. 系統安全性的高低應該只和加密金鑰相關，與加、解密演算法無關。

加密系統大致上可依加密金鑰的類型分成單向雜湊函數(One-Way Hash Function)、對稱金鑰(Symmetric key)密碼系統、非對稱金鑰(Asymmetric key)密碼系統等，底下將一一介紹：

1. 單向雜湊函數(One-Way Hash Function)：一種可以將原先不論長度有多少的訊息，經過單向雜湊函數的處理後，得到固定長度的訊息，稱爲『訊息摘要』(Message digest)，寄件者把原文的訊息摘要，利用私鑰產生「簽章」的動作，把這被簽章的資訊摘要和該文件一同傳遞出，收件者利用寄件者的公鑰去確認其簽章，來確定其文件是否確實寄件者所寫之內容，這類的函數常用的有 MD5、SHA-1…等，使用情形如下圖所示：

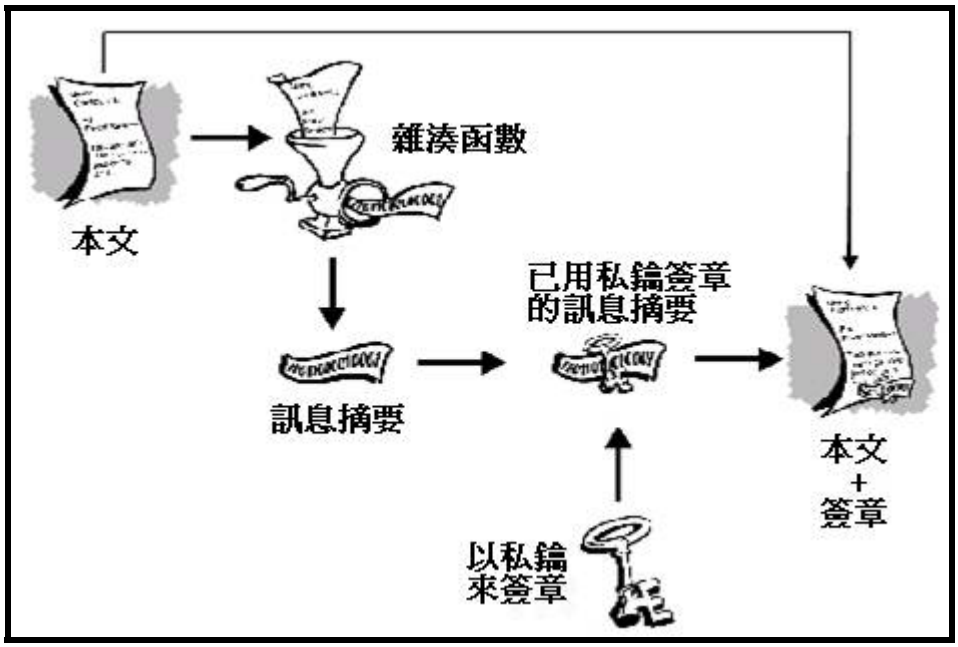


圖 15　單向雜湊函數使用範例
(資料來源：[37])

2. 對稱金鑰(Symmetric key)密碼系統：又稱秘密金鑰(Secret key)加密系統，傳送資料的雙方事先溝通好所使用的金鑰，稱爲秘密金鑰。使用這把秘密金鑰對要傳送的資料經過加密演算法做加密的動作後，傳送到接收的一端。最後利用這把金鑰經過解密演算法還原得到原本的資料，如下圖所示：

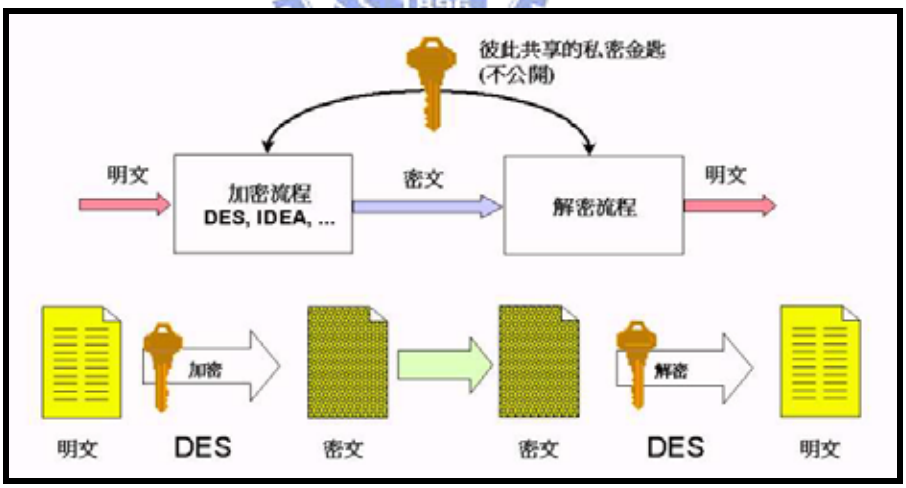


圖 16　對稱金鑰密碼系統
(資料來源：[37])

3. 非對稱金鑰(Asymmetric key)密碼系統：又稱做公鑰加密系統，爲了用來解決金鑰傳遞的問題(若秘密金鑰遭劫取，則資料即能被竊取)，而發展出來的系統。使用者擁有一對金鑰，一爲公鑰(Public key)，另一爲私鑰(Private Key)，公鑰是公開發佈的，用來對要傳送的資料加密用，只有擁有私鑰的人才能解回原文，所以只要保護私鑰不被取得即可，加、解密的方法如下

圖：

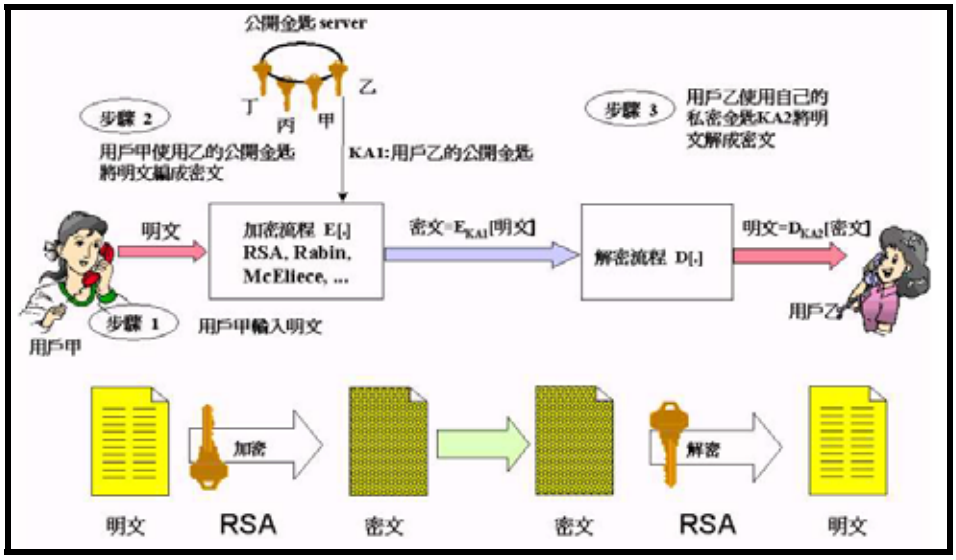


圖 17　非對稱金鑰密碼系統
(資料來源：[37])

以上是目前常用的加密系統，依照使用金鑰方式的不同加以分類介紹，接下來的幾小節中，我們將介紹將拿來與本文所提出方法做比較的幾種常見加密演算法，分別爲 DES、triple DES、AES、RSA 等。

2.3.2　常見的加密演算法

在這個小節中，我們將介紹一些常見的加密演算法，例如：對稱加密演算法(Symmetric encryption algorithm)中的 DES、triple DES、AES，以及非對稱加密演算法(Asymmetric encryption algorithm)的 RSA 等，做簡單的介紹。

2.3.2.1　Data Encryption Standard (DES)與 triple DES

1970 年代中期由美國 IBM 公司發展出來，於 1977 年經過美國國家標準局公布爲資料加密標準的一種區塊加密法(Block Cipher)，是一種密鑰加密演算法(對稱加密演算法)。由於近幾年電腦的運算速度增快，DES 加密過的資料已經可以在有限時間內被解開，所以才有 Triple DES 的出現。

DES 一次加密的區塊大小爲 64 bits，其密鑰大小亦爲 64 bits，但是其中有 8 bits 是用來做錯誤較正(Error correct)用，所以實際的金鑰大小爲 56 bits。如圖 18 所示，DES 的加密過程有 16 個回合(Round)，首先需要將使用者輸入的 64 bits 密鑰經過子金鑰的處理，生成 K1,…,K16 等子金鑰。接著將輸入的區塊做初始化排列(Initial permutation)後，分成 L0 與 R0 兩個各 32 bits 的區塊，L0 與子金鑰 K1 經過 f 函數的運算後，再與 R0 做 XOR 的運算得到 R1，而 R0 成爲下接段的 L1，如此 16 次運算得到輸出的資料。加密

與解密過程的差異只在於子金鑰的輸入順序相反，其他部份皆相同。至於細節的部份，如子金鑰的產生、f 函數等，在此不多做說明。

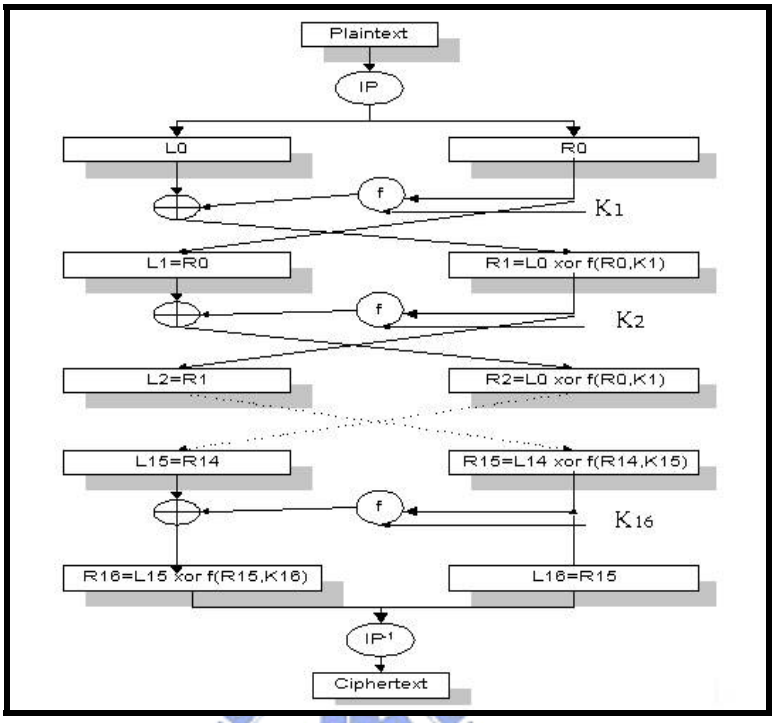


圖 18　DES 加密流程圖

近年來，由於個人電腦的運算能力大幅成長，用暴力法來破解 DES 加密過的資料已經不是什麼太耗時的事情，所以美國已經宣佈政府機構不再使用 DES 爲加密的標準，因此 triple DES 順勢而生。實際上 triple DES 並沒有改變原本 DES 的演算法，而是將 DES 的演算法重複執行 3 次，如此達到的安全性可以讓目前的電腦無法在有限時間內計算出原本的資料。

Triple DES 可依金鑰的不同分爲三類：

1. 第一類的 triple DES 使用三把金鑰，分別運用於每次的 DES 回合中，所以金鑰長度爲 192 bits(實際大小有用金鑰大小爲 168 bits)。
2. 第二類的 triple DES 使用兩把金鑰，第一次 DES 回合使用一把金鑰加密，接著使用另一把金鑰「**解密**」，再使用第一把金鑰加密後送出。
3. 第二類的 triple DES 也是使用兩把金鑰，與第二種 triple DES 不同的地方在於，第二回合的 DES 是採用第二把金鑰「**加密**」，其它部份相同。

2.3.2.2　Advanced Encryption Standard (AES)

由於計算機技術的大幅進步使得電腦的運算能力大幅提昇，對於使用 64 bits 的金鑰的 DES 而言，面臨了極大的考驗與威脅，因此 NIST 徵求下一代的加密標準(Advanced Encryption Standard, AES)。1997 年四月的時候，NIST 對外公布徵選 AES，2000 年底，

Rijndael 由十五個候選的方法中被選出，於 2001 年 4 到 6 月間，完成了標準制定與測試，2001 年 11 月，NIST 將 AES 定爲 FIPS197。在未來的三十年內，AES 將會取代 DES 成爲政府及民間企業通用的密碼加密標準，用來保護重要的資料。它的明文密文區塊爲 128 bit，而主金鑰的長度可分別有 128/192/256 bit 三種選擇。

Rijndael 是由比利時的 Joan Daemen 和 Vincent Rijmen 發明的，其命名是由兩人各自的姓取出一部分而來。Rijndael 是以方塊(Square)爲基礎發展而來的，原本的設計是區塊的大小可以有 128、192、256 位元三種變化，配上金鑰提供的選擇，可以有九種組合，但是最後實際應用時也許只有採用 NIST 要求的 128 位元。Rijndael 的運算是以位元組(Byte)爲單位的，其區塊和金鑰實際上可以是任何 32 的倍數。

首先，介紹一些基本定義，明文區塊和密鑰都可以表示爲 4*n 的陣列，其中每一個元素都是一個位元組，其中 n 與區塊或密鑰的長度有關，n = 長度 / 32。此外，區塊的 n 表示爲 Nb，密鑰的 n 表示爲 Nk，定義 Nr= max(Nb, Nk)+5，加密所要進行的回合數，是 Nr+1 次。

AES 編碼的過程中的每一回合，除了初始化和最後一回合，均爲四個步驟，分別爲：Byte Sub、Shift Row、Mix Column，以及與 Add Round Key。

- 在Byte Sub這一步驟中，所作的運算大概是這樣的，考慮這時的區塊的陣列表示元素，$\mathbf{a}_{ij}$，由 $\mathbf{a}_{ij} * \mathbf{a}_{ij}^{-1}=1 \mod m(x)$，其中 $m(x) = x^8 + x^4 + x^3 + x + 1$，算出來的 $\mathbf{a}_{ij}^{-1}$再去作如下的陣列運算，得到$\mathbf{b}_{ij}$，如此對每個元素都一樣運算，就是Byte Sub的變換過程。必須注意的是，AES演算法中多項式的加和乘是在有限場$GF(2^8)$上處理，與一般的代數運算不同。
- 在 Shift Row 這一步中，矩陣的每一列都會有不同位數的左移，左移數還與區塊長度有關，關係如表 3：

表 3 AES 列位移表(Shift row table)

Nb	列一	列二	列三	列四
4	0	1	2	3
6	0	1	2	3
8	0	1	3	4

(資料來源：[36])

- 在Mix column中，每列都被視爲$GF(2^8)$上的多項式，多項式與$C(x)='03'x^3+'01'x^2+'01'x+'02'$相乘後取mod$(1+x^4)$，得到D(x)，這個過程也可表示爲D(x)=C(x) ^ $\mathbf{a}_{ij}$。
- 最後，在Add Round Key這一步中，狀態陣列會和該回合的回合密鑰作XOR的運算，得到C(x)，其中$C_{ij} = A_{ij} \oplus Kn_{ij}$，$C_{ij}$, A_{ij}和Kn_{ij}分別爲三個陣列的一個元素。這四個步驟，會連續做Nr次，且最後一次不會做Mix column這一步。初始化只包含了Key expansion和Add Round Key這兩步。

2.3.2.3 RSA

基於 Diffe 和 Hellman 提出的以單向暗門函數(One-way Trapdoor Function)爲基礎的公開金鑰密碼系統，1978 年美國麻省理工學院三位教授 Rivest、Shamir 及 Adleman 首先提出一種基於分解因數的指數函數爲單向暗門函數的公開金鑰密碼系統，稱爲 RSA。

接下來我們將分兩階段介紹 RSA 系統的細節[36]，首先介紹加密用的金鑰產生過程，接著再說明利用公、私金鑰加密傳送的過程。

RSA 金鑰的產生可以分成下面三個步驟：

1. 使用者 A，任意選出兩個大質數p_A與q_A，並求出$N_A = p_A \times q_A$。
2. A 求出一任意整數e_A，使得$GCD(e_A, T_A) = 1$，其中$(p_A - 1) \times (q_A - 1) = T_A$，則$e_A$與$N_A$爲公開金鑰。
3. 計算出d_A，使得$e_A \times d_A \equiv 1 (\mathrm{mod}\, T_A)$，如此$d_A$即爲私人金鑰。

有了公鑰以及私鑰之後，接下來就是傳送資料時加、解密的過程了，底下將分成加密與解密兩個步驟來說明。

1. 加密：當使用者 B 欲傳送資料給使用者 A 時，明文$m(0 \le m \le N_A)$經過 A 的公開金鑰(e_A， N_A)處理後，得到密文爲$E_A(m) = C = m^{e_A} (\mathrm{mod}\, N_A)$。
2. 解密：當 B 將密文 C 傳到 A 後，A 利用其原有的私鑰d_A執行解密演算法：$D_A(C) = C^{d_A} = (m^{e_A})^{d_A} = m^{KT+1} = m (\mathrm{mod}\, N_A)$。

以上即爲 RSA 在處理加、解密，以及產生金鑰的過程說明。

2.4 網路安全與 IPSec 協定

網路上有許多攻擊方法是應用從網路層(Network layer)截取資訊，如來源位置(Source address)、目地位置(Destination address)…等，來對使用者連線做破壞的動作。爲了防止這一類的攻擊，IETF 建立了一套用來保護網路層傳輸傳送的架構，內含許多的通訊協定 [21] ~ [25]，被規定爲 IP v6 網路的基本要件之一。

IPSec 可以分成四個主要的元素，安全關聯(Security Association)、金鑰管理(Key management)、安全協定(Security protocol)，以及加解密或認證演算法(Encryption or authentication algorithm)，除了第四個部份已經在上一小節介紹過之外，其他將一一在下面的章節介紹。

2.4.1 安全關聯(Security Association)

安全關聯(Security Association，SA)的概念是整個 IPSec 的基礎觀念 [21]。簡單來說就是一個可以用來承擔安全服務(Security service)在上面傳送資料的連線(Connection)，這裡的安全服務包括將於 2.4.3 節中介紹的 ESP(Encapsulating Security Payload) [22]或是 AH(Authentication Header) [23]。

一個 SA 連線爲單向連線，如果需要建立雙向的連線，即需要建立兩條 SA 的連線。相同的，一條 SA 連線只提供一種安全服務，即 ESP 或 AH 兩者其中之一，若是要建立有 ESP 與 AH 的安全服務，需要建立兩條 SA 連線。因此 SA 也被稱爲點對點(Point-to-Point)的通訊方式，雖然也可以應用於點對多點(Point-to-multipoint)的通訊。每個 SA 連線存在著三個構成要素來唯一定義這個連線，分別爲 Security Parameter Index (SPI)、目的地位置(Destination IP address)與安全服務用的通訊協定識別號碼(Security protocol identifier)。

SA 可以分爲兩種模式，一種爲傳輸模式(Transport mode)，另一種爲隧道模式(Tunnel mode)，底下將分別介紹這兩種模式：

1. 傳輸模式(Transport mode)：建立 SA 於兩臺主機(Host)間。在 IP v4 的網路架構下，傳輸模式會將安全協定標頭(Security protocol header)至於 IP header 與 option 之後，在任何高層通訊協定(如：TCP、UDP…等)之前。在 IP v6 的網路架構下，安全協定標頭(Security protocol header)會在 IP header 與 extension 之後，但有可能在 destination option 之前。
2. 隧道模式(Tunnel mode)：一個隧道模式的 SA 本質上是將 SA 應用於 IP 隧道(Tunnel)上。如果有一端的端點爲安全閘道(Security gateway)的話，這一個 SA 連線一定使用隧道模式，所以無論是 SA 介於安全閘道與安全閘道之間或是介於主機與安全閘道之間，一定都是使用隧道模式。當然主機與主機間亦可使用隧道模式的 SA。

爲了能提供最小且適當的管理，SA 提供了兩種資料庫系統來幫助 SA 的建立與管理，一個爲安全方針資料庫(Security Policy Database, SPD)，另一個爲安全關聯資料庫(Security Association Database, SAD)。前者主要是規定從主機或是安全閘道送出的所有 IP traffic 要使 inbound 或是 outbound 的規則。後者則是用來儲存關於每個 SA 所需要用到的參數。以上是對安全關聯做簡單的介紹。

2.4.2 金鑰管理(Key management)

IPsec 系統由於提供十分彈性的封裝機制、安全協定、加解密演算法組合，因此通訊雙方必須事先「秘密分享」許多安全參數的設定（包含金鑰在內）。讓通訊雙方分享安全參數的方式有手動設定（Manual keying）與自動協調（Automatic Negotiation）兩種。而所謂的自動協調則是 IETF 提出的「自動金鑰交換機制」-- Internet Key Exchange（簡稱 IKE，RFC 2409） [24]。以實作的角度來看，IKE 是完全獨立於 IPsec 的應用系統，他會自動回應 IPsec 對安全參數的需求而與遠端的 IKE 進行安全參數的協調工作。由於多數廠商對於 IETF 所訂定的 IKE 標準並未達成完全的共識，因此歷年來所舉辦的 VPN 互通性測試皆是以 IKE 爲主。

IKE 是一個複合的通訊協定，結合了 RFC 2408 的 Internet Security Association and Key Management Protocol(ISAKMP) [25]、Oakely 提出的一系列金鑰交換的方法(The Oakely Key Determination Protocol，RFC 2413，稱爲 modes)以及一部份的 SKEME 金鑰交換技術。

IKE 用來建立一個認證過的交換金鑰可以分成兩個階段，第一個階段爲 IKE peers 建立一個安全且認證通過的通道(使用 ISAKMP 中定義的 parse 1)，可以讓 IKE peers 互相溝通。在第一階段中又可以分成兩種模式，一種爲主模式(Main mode)，而另一種爲主動模式(Aggressive mode)，兩者都是利用 ephemeral Diffie-Hellman 產生認證通過的金鑰材料，兩者的不同在於，主模式在溝通與傳送的參數上較主動模式來的多，所以花費的時間也較主動模式多，但是使用起來靈活度較大。第一階段有四種認證方法：預共用密鑰、數字簽名、公鑰加密、修正的公鑰加密。第二階段則是利用第一階段產生的安全關聯(Security Association)來溝通通道中所要使用的安全服務(如：IPSec)所需要的金鑰，或是其他需要設定的參數…等，用來建立 IPSec 的 SA。第二階段只提供一種模式，爲快速模式(Quick mode)，因爲第一階段所產生的 SA 爲雙向通道，所以第二階段快速模式的發起可由任何一端皆可。

此外，RFC 中還定義了一種名爲新群組模式(New Group Mode)，規定只能使用於第一階段，而且要在 IKE SA 建立前先進行，用來定義新的群組，但仍在發展中，尚未有完整的架構。以上即爲對金鑰管理用於 IPSec 的簡單介紹。

2.4.3 安全協定(Security protocol)

IPSec 中使用了兩種主要的安全協定：ESP (Encapsulating Security Protocol，RFC 2406) [22]以及 AH (Authentication Header，RFC 2042) [23]，底下將對兩種安全協定做較爲詳細的介紹。

ESP 主要是結合加密演算法（如 DES、3DES、或 AES）及雜湊函數(Hash function)，對封包內容加密後即可不用擔心封包遭到竊取，同時亦可有類似 AH 的驗證能力。但是 ESP 協定對於防止篡改或造假的能力並不如 AH 強大，因此雖然 ESP 亦有類似驗證功能，仍不可完全取代 AH。爲了配合 SA 的特性，ESP 同樣也分成兩種模式：傳輸模式(Transport mode)與隧道模式(Tunnel mode)，下圖爲兩種不同模式時，封包標頭(Header)的示意圖(以 IP v4 爲例)：

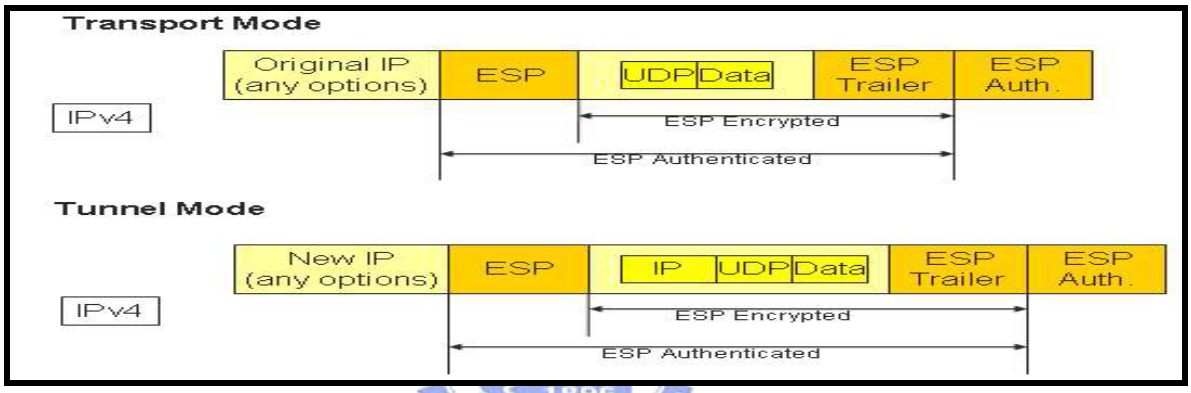


圖 19 ESP 傳輸模式與隧道模式封包標頭示意圖

AH 主要透過雜湊函數(Hash function)（例如 MD5 及 SHA-1）的技巧，提供封包來源的驗證、以及內容一致性的檢查，因此，當 IP 封包（包含 IP Header 及 Data Payload）在網路上遭人篡改或假造都可檢查出來，提供資料的正確性。爲了配合 SA 的特性，AH 同樣也分成兩種模式：傳輸模式與隧道模式，下圖爲兩種不同模式時，封包標頭的示意圖(以 IP v4 爲例)：

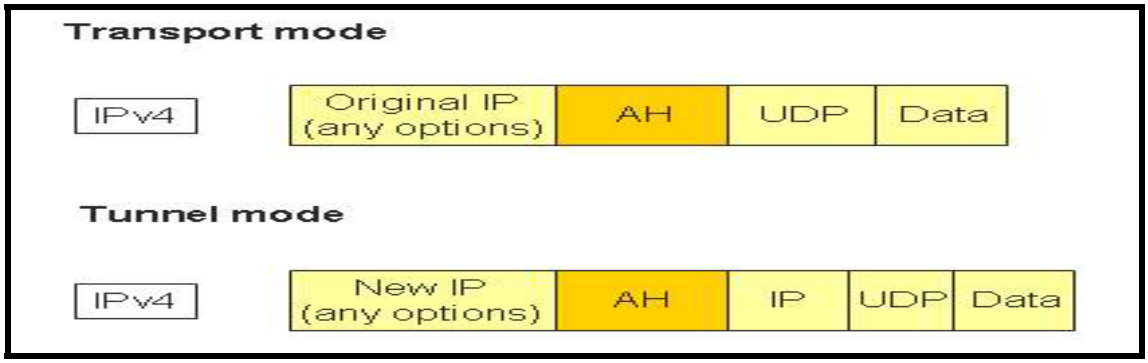


圖 20 AH 傳輸模式與隧道模式封包標頭示意圖

三、相關研究

本章節將介紹與本論文相關的一些研究。首先是傳輸聲音封包於 IP 網路上的特性研究，接著是一些使用 IPSec 傳送 VoIP 的相關研究與分析，以及 cIPSec 的介紹。最後是與本論文提出的方法相關的研究：視覺加密(Visual cryptography)與互斥多路徑路由演算法(Disjoint multipath routing algorithm)。

3.1 VoIP 效能需求

在這個小節中，我們將從過去的研究與試驗中，如 [7] [8] 等，獲得有關於網路上的參數對於 IP 電話的聲音品質影響，藉以用來做爲本論文所提出來的論述基礎，也更進一步了解 IP 電話需要克服的問題，提出對應的解決方法。

單向 Mouth-to-ear 延遲相當於聲音由發話端到收話端的延遲，大至上包含三個部份：將聲音數位化並且編碼與封裝、封包傳送時間，以及接收端消除 jitter 與解回類比資料的時間，如下圖所示。一般而言，一般的端對端(End to end)VoIP 延遲的範圍在 150 *ms* 到 400 *ms* 之間 [7]，而根據 [8] 的研究結果，Mouth-to-Ear 延遲時間在 200 *ms* 以內的 VoIP 產品即可用於商業用途上。ITU-T 也於 G.114 [26] 標準中提出，單向延遲(One-way delay)在 150 *ms* 內電話產品可以得到最佳的聲音品質(G.114 原先是爲 PSTN 聲音品質做測試的標準，同樣的也可以適用於 VoIP 聲音品質的測量上)。

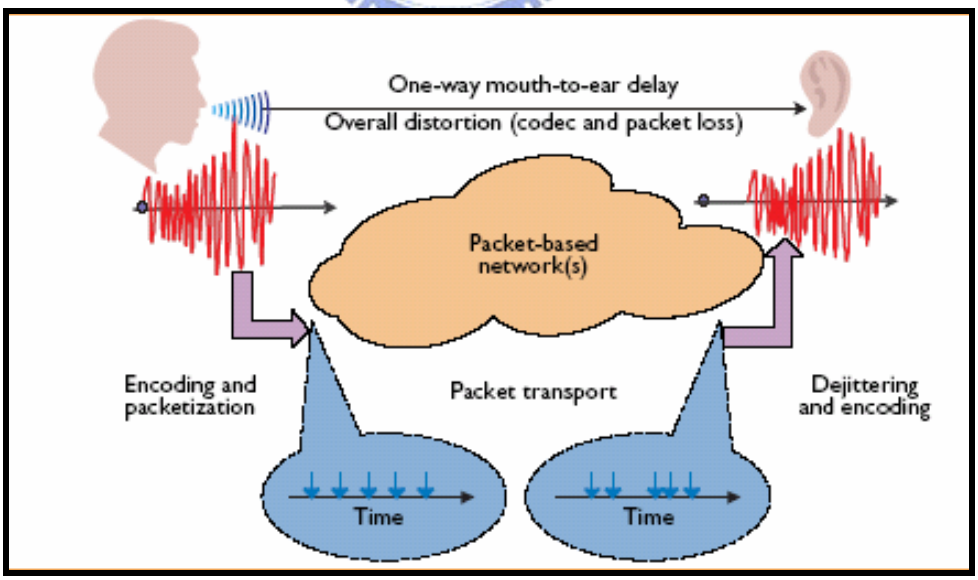


圖 21 單向 Mouth-to-ear 延遲示意圖

(資料來源：[8])

底下我們將從 [8] 與其它相關 [9] 的研究中，討論上圖中，聲音封包透過 IP 網路傳送時，會造成延遲的三個部份，與不同聲音編碼法的關係。

首先是聲音數位化與編碼的部份，我們可以從下表中看出，壓縮較佳的聲音編碼方式，通常需要較多的處理時間，但是可以得到較小的封包，使頻寬的需求量減少。同樣地，在接收端收到聲音封包後，解碼所需要的時間也和編碼所需時間相同。

表 4　各種聲音編碼法的負載大小(Payload size)與編碼延遲列表

Compression method	Bit Rate (Kbps)	Payload size (bytes)	Encode delay (*ms*)	MOS score
G.711 (PCM)	64	160 (default)	0.75	4.1
G.726 ADPCM	32	60 (default)	1	3.85
G.728 LD-CELP	16	40 (default)	3 to 5	3.61
G.729 CS-ACELP	8	20 (default)	10	3.92
G.723.1 MP-MLQ	6.3	24 (default)	37.5	3.9
G.723.1 ACELP	5.3	20 (default)	37.5	3.65

在封包傳送的時間延遲部份，可分爲兩類的延遲情形，一種爲最小網路延遲時間，這類延遲主要爲傳送延遲(Propagation delay，約傳送每一公里產生 5 microseconds 的延遲)。另一種爲排程延遲(Queuing delay)，這部份的延遲主要產生在封包經過路由器(Router)或是閘道器(Gateway)時，在路由器或是閘道器中等待被處理的延遲，通常需要視網路當時的流量而定，若是經過的路由器或是閘道器無法再容納更多的交通流量(Traffic)進入該路由器或是閘道器時，這些封包會被丟棄(Discard)，而產生封包遺失的情形。封包遺失對於每一種聲音編碼方式有著不同的影響，可由下圖中得知：

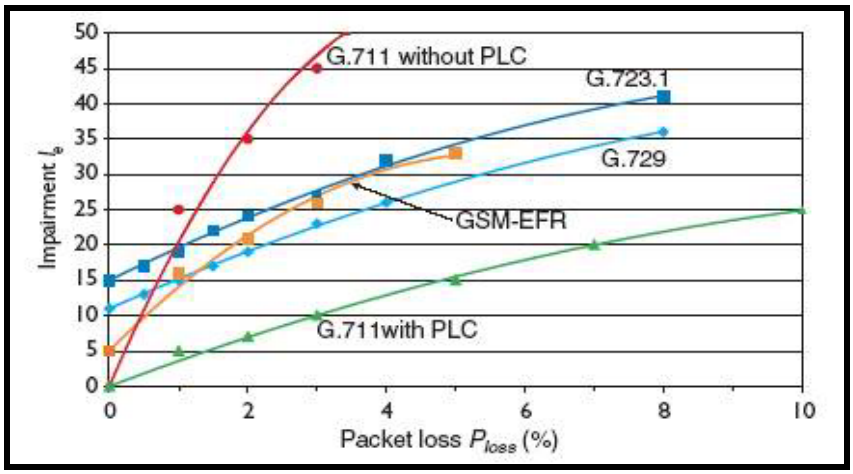


圖 22　封包遺失對不同聲音編碼方式的影響
(資料來源：[8])

在接收端，因爲每個封包傳輸的延遲不同，產生所謂的延遲振盪(Delay jitter)，爲了處理振盪(Jitter)，通常在接收端會使用消除振盪(Dejittering)的技術來消除振盪對聲音撥放的影響，最簡單的方法是將收到的封包放置於一個暫存區內(Buffer)，然後再按照撥放的順序依時間撥放。同樣的，使用消除振盪的技術同樣也會使 Mouth-To-Ear 的延遲增加，所以消除振盪的技術選用也是很重要的課題之一。

以上是對 VoIP 在傳送上的一些特質介紹，之後我們將使用並針對這些特質提出一套有效的安全方法，來解決本文一開始所提到的問題。

3.2 使用 IPSec 傳輸 VoIP

在這一節中，我們將介紹幾篇有關於運用 IPSec 來確保 VoIP 安全傳送的相關研究與分析[5][6]，並且在 3.2.2 小節中介紹[6]所提出的壓縮 IPSec 概念。

3.2.1 使用 IPSec 傳輸 VoIP 分析研究

IPSec 是近期被 IETF 提出來，用來保護網路層安全性的一套架構，可以增加 IP 網路的基本安全性。對於是否適用於對時間極爲敏感的即時應用程式，需要以實驗來驗證，對於這方面的分析與研究可從 [5] [6] 兩篇來看。底下我們將針對這兩篇論文的內容做介紹，並且在下一節介紹其中一篇論文 [6] 所提出來的方法 — cIPSec。

IPSec 在前面的章節(2.4)介紹過了，是一個透過內部金鑰管理來建立安全連線(Security Association)，接著在 SA 中使用傳輸模式或是隧道模式配合 Encapsulating Security Protocol (ESP)或是 Authentication Header (AH)以達到資料的安全與正確傳輸的目地。由上一節的結論得知，VoIP 在傳輸上對時間的要求有比較嚴格的限制，所以封包的大小也會因爲這個因素而有所限制，但是使用 IPSec 作爲安全傳輸的基礎，會在運用 ESP 或是 AH 時，會使原本的封包增加許多，我們可以由下圖(圖 23)中看出成長的幅度。

爲了了解在 IPSec 的架構下，在網路傳輸聲音封包的影響有多少，這兩篇論文都做了模擬的環境，借此來了解真實的情形，看看是否 IPSec 適合使用來增加 VoIP 的安全性，得到的結果如下頁圖 23、圖 24，以及圖 25 所示，可以看出有一定程度的影響，所以在 Reberto Barbieri 的研究 [6] 中，提出了 cIPSec(將於下一節介紹)的概念，試圖解決使用 IPSec 導致封包擴張而使延遲增加的情形。

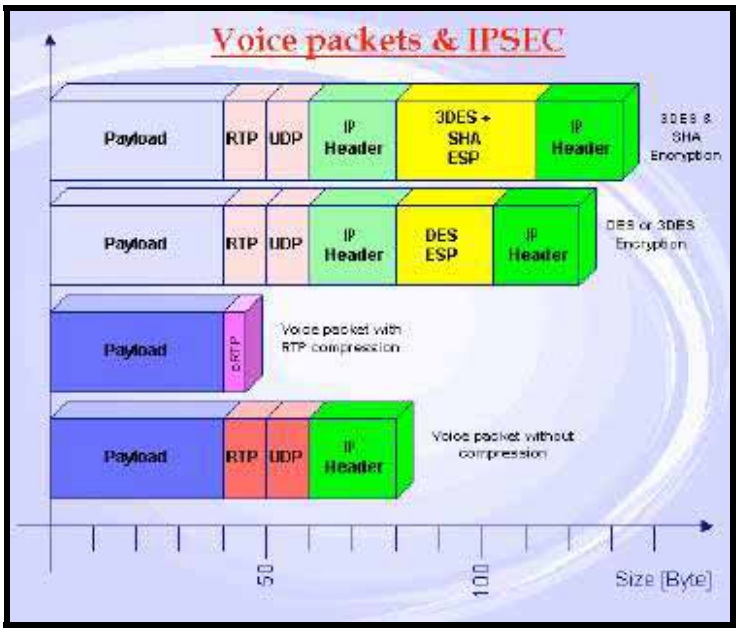


圖 23　使用 IPSec 於 VoIP 封包的標頭成長示意圖

(資料來源：[5])

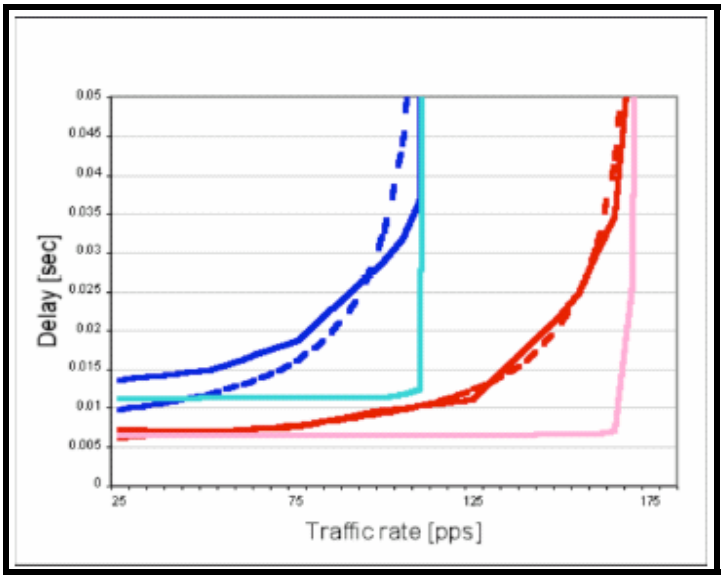


圖 24　封包延遲與網路流量的關係圖，最右邊的線段爲沒有使用IPSec的情形，最左邊的線段則是使用DES加密於IPSec的情形

(資料來源：[6])

由於IPSec不對加密與認證方法做限制，所以用不同的加密或是認證方法會對封包大小有不同的影響，但是不論是那一種組合，都對聲音傳輸有相當的影響，所以底下我們將介紹cIPSec來改善這個問題。

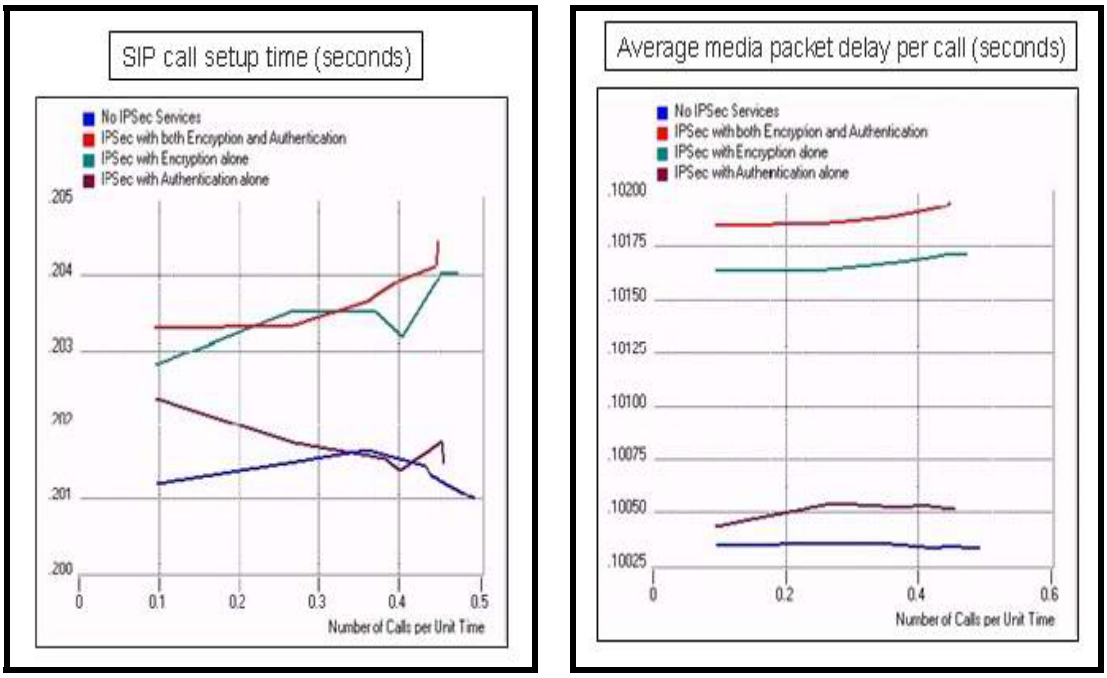


圖 25　左圖爲 SIP call setup 時間的模擬結果，右圖爲模擬傳送聲音封包延遲時間所得到的結果

(資料來源：[5])

3.2.2　壓縮 IPSec(cIPSec)

cIPSec 的概念是來自觀察封包傳送的標頭所得來的，有許多的欄位在通訊過程中根本不會變化，或是只和前一個封包相差 1 個位元的成長，如 sequence number，又或是某些欄位的值在更上層的通訊協定標頭(Protocol header)中已經存在…等情形，可以加以縮減，以減少所需要傳送的資料量，此種方法也可以用在低速網路連接上(cSLIP，Low-Speed Serial Links，RFC 1133)。

由上面的觀察，在 [6] 這篇論文中打算把隧道模式中 IP/UDP/RTP 的內部標頭壓縮成 4 bytes，方法是將上述的欄位不在網路上傳輸，以減少傳輸的資料量。爲了達到上面的目的，每個連線的兩端點都需要維護一些資訊以及連線內容等，這些資料用來幫助接收端回復原來完整的封包標頭。

儲存在加密兩端點上的資訊包含下列的幾個部份，主要是 IP header 中的來源與目的地位址、UDP 中的來源與目的 port 以及 RTP 中的 SSRC 欄位…等，如下所列：

- 尚未使用壓縮(cIPSec)前最後一個完整的 IP/UDP/RTP 標頭，使用內部的資訊來幫助壓縮後的封包重組。
- 尚未使用壓縮(cIPSec)前最後一個 sequence number 的後四位元，用來檢驗封包遺失於 compressor 和 decompressor 之間。

經過壓縮後的內部標頭(Internal header) (IP/UDP/RTP)如下圖所示，各欄位說明見於圖下所示：

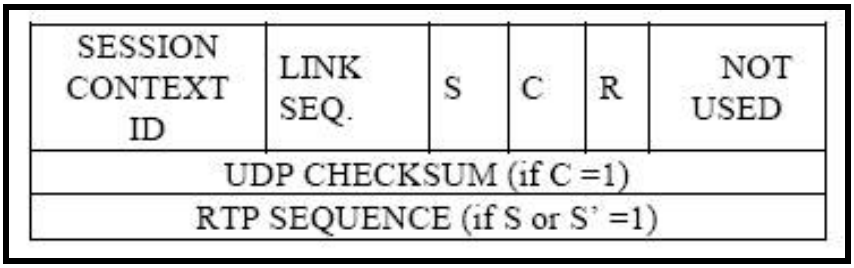


圖 26　cIPSec 標頭(Header)

(資料來源：[6])

- Session context ID：共有 16 bits，用來區分同樣兩個端點間，不同連線用，如此可在兩端點間同時存在 65536 通連線。
- Link sequence：8 bits，用來表示 RTP 標頭末 8 位的位元，如此可以讓接收端追蹤至多連續 256 個的封包遺失。
- S、C、R：各爲 1 bit，S 與 C bit 分別用來表示是否有 RTP sequence 與 UDP checksum；R bit 用來指出是否需要重送封包並且不經過壓縮。
- UDP checksum 與 RTP sequence：爲選擇性存在(Option)的欄位，即爲原封包的 UDP checksum 與 RTP sequence。

原封包格式爲 2.4.3 節中圖 19 的 ESP 隧道模式，經過 cIPSec 的處理後，將會被轉換成爲下圖的格式：

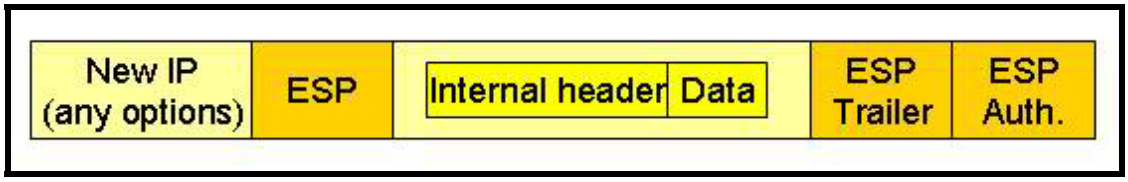


圖 27　cIPSec 用於 ESP 隧道模式的封包標頭示意圖

經過該論文的實作 cIPSec 與模擬之後，可以得到下圖的結果，很明顯的可以大大的減少頻寬的使用量。雖然可以減少頻寬的使用量，但是 cIPSec 還是使用需要加解密的過程並且需要用到金鑰的觀念，所以本篇論文僅將此方法用於訊號(Signals)傳遞的保護上。

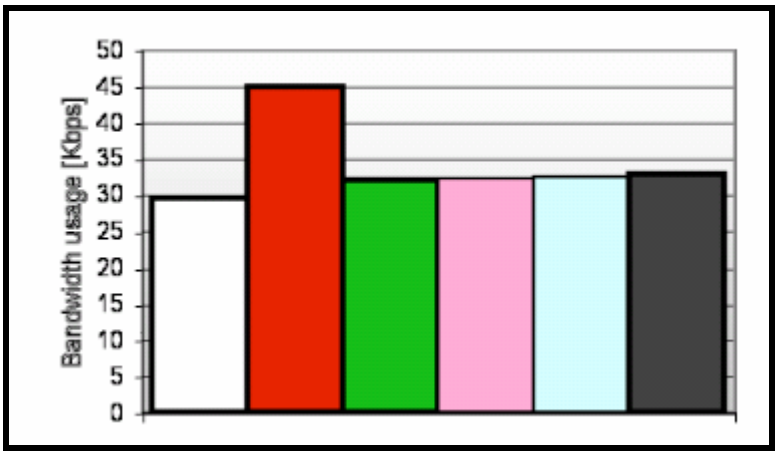


圖 28　由左到右分別爲：一般的聲音封包、使用 IPSec、使用 cIPSec、2%資料遺失率、5%資料遺失率，以及 10%資料遺失率的頻寬使用大小

(資料來源：[6])

3.3　視覺加密(Visual cryptography)

視覺加密法(Visual cryptography) [10]，是由 Moni Naor 與 Adi Shamir 兩人於 1995 年所提出，一種藉由人類視覺特性，將書面的材料(如：書面文字、手寫筆記、圖畫…等)加密的方法，最後直接由人類的視覺系統加以解密。這種加密方式基本上包括一張密文(以傳真或是郵寄的方式傳送)與一張列印出來的投影片(如同密鑰的功能)，收到的一方將兩者重疊觀看即可得到原本的明文(可能會有一些隨機雜訊的存在)，如此即可簡單與快速的將明文加密，而不需要經過繁複的處理過程。

最早被提出的視覺加密是運用在黑白兩色的圖文上，將原圖分成兩個相等大小分享資料(Share data)的方法如下表所示，當原圖(明文)上的點爲白色時，share-1 與 share-2 皆使用白色；當原圖(明文)上的點爲黑色時，share-1 與 share-2 使用 Black 一行中，三種方法的其中一種(黑白、白黑、黑黑)。這種方法的好處是可以節省空間，但最大的缺點是可以很容易的從分享資料(Share data)中看出原圖內容，如圖 29 所示。

表 5　每個像素使用 1 位元分割的視覺加密法(Visual crytography)

Original pixel	White	Black
Share-1	□	□ ■ ■
Share-2	□	■ □ ■
Stacking result	□	■ ■ ■

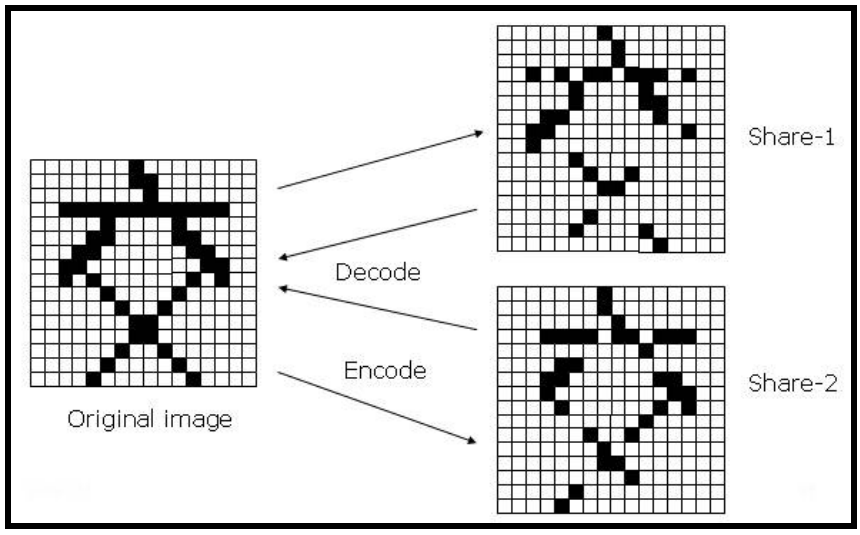


圖 29　每個像素使用 1 位元分割的視覺加密法範例

由於上述的方法有嚴重的安全缺點，所以經過研究以後，將原本在原圖上的 1 個像素(Pixel)擴充爲分享資料(Share data)上的四個像素來表示，對應的表格如下(表 6)所示，如此一來即可得到像雜訊般的分享資料(Share data)，增加安全性。此方法的缺點是，需要原圖四倍的空間來儲存分享資料，如圖 30 的例子所示。

表 6　每個像素使用 4 位元分割的視覺加密法(Visual crytography)

Image	Secret pixel (white)	Secret pixel (black)
Share-1		
Share-2		
Stacking result		

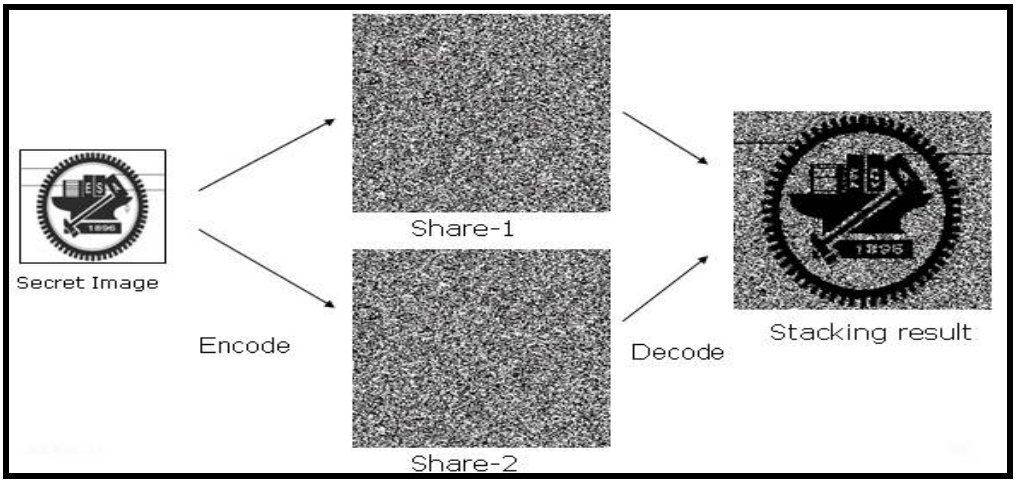


圖 30　每個像素使用 4 位元分割的視覺加密法範例

由視覺加密法的構想，我們可以引用於聲音安全加強的方面，將已經數位化的聲音依照視覺加密的方式，再根據聲音的特性使擴張倍數減小，如此可以產生一個不需要建立、傳送與保護金鑰的加密方法，此即爲本篇論文的基本構想，如下圖所示：

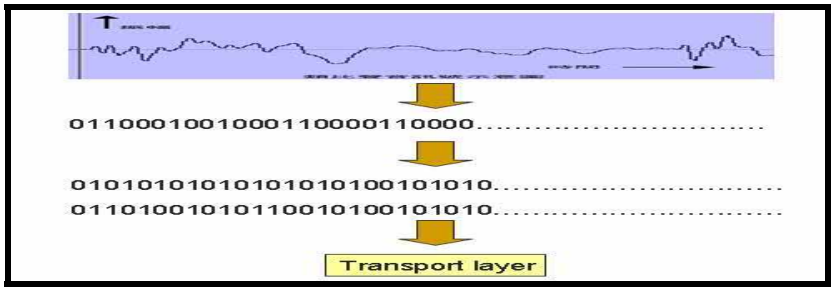


圖 31　運用視覺加密演算法於聲音傳輸的構想圖

3.4　互斥多路徑路由演算法(Disjoint multi-path routing algorithm)

由於上一節提到將資料分割傳送的概念原本並非使用於網路上傳輸，所以轉移到網路平臺上便需要一些技術的幫助，互斥多路徑路由(Disjoint multi-path routing)即爲其中一項。在分享資料的觀念中，兩個不同的分享資料需要在不同的路徑上傳輸，否則被攔截到的分享資料依原方法組合即可得到原本的資訊，如此就失去了資料分享(Data sharing)的安全性了。另外，根據許多的研究顯示[2, 3, 4]，動態多路徑路由(Dynamic multi-path routing)的方式有減緩網路阻塞，使傳輸延遲減低的好處，利於時間敏感度高的應用，如 VoIP、多媒體串流(Multimedia stream)…等，底下我們將介紹一些過去別人所提出的互斥多路徑路由演算法(Disjoint multi-path routing)與一些分析所得的結果。

對於尋找互斥集合(Disjoint-set)的路由路徑(Routing path)，已經有許多的研究與演算法被提出來，如相關研究中的 [11] [12] [13]等，各有各的優點，在此不一一列出。於 [12] 中所提出的構想與本論文相似，所以將此篇論文所提的演算法爲基礎，介紹 [12] 中所提出來的互斥多路徑路由演算法。

在介紹演算法之前，先對一些網路拓樸(Network topologic)的表示法介紹與說明。一個網路架構可以模型化爲一個無向圖(Undirected graph)，G=(V,E)，其中 V 表示一個有限的節點(Node)集合，E 爲一個有限的連線(Link)集合。一個 E 中的連線連接著一對的節點，x 與 y，記號爲(x,y)，而此連線的花費(Cost)記號爲 c(x,y)，其中 c(x,y)爲一個正數且 c(x,y)=c(y,x)。

給定一個 V 集合中的節點 t，稱爲目的節點，則一條路徑(Path) p 由 s 到 t，可表示爲(s,i,j,…,k,t)，爲 E 中一群連線的集合(s,i),(i,j),…,(k,t)，路徑 p 的花費記號爲 c(p)，爲經過路徑中所有連線的花費之和。若存在一個路徑 p 爲(s,x,…,y,t)，則唯一定義其 path

id，pid(p)=y。G 中存在一個以 t 爲根節點(Root)的最短路徑樹(Shortest path tree)，記號爲 SPT，其中 SPT 上的每一點延著 SPT 的連線到 t 的路徑爲花費最少的路徑。SPT(x,t)表示唯一存在的路徑由 x 到 t，若 y 爲路徑 SPT(x,t)上的其中一個節點，則 y 稱爲 x 的 downtree，x 稱爲 y 的 uptree。對於一個節點 x，若存在一連線(x,y)，則 y 稱爲 x 的 neighbor，且將所有 x 的 neighbor 記號爲 nbr(x)。若 y 爲 x 的 uptree 且 y∈nbr(x)，則 y 稱爲 x 的 uptree neighbor；若 y 爲 x 的 downtree 且 $y \in nbr(x)$ ，則 y 稱爲 x 的 parent；若 $y \in nbr(x)$ ，且(x,y)非 SPT 的 link，則稱 y 爲 x 的 horizontal neighbor。

在[12]中假設，每個節點都已經知道他的 uptree neighbor、parent 與 horizontal neighbor 等，並假設過程中沒有拓樸架構的改變。用來傳送的訊息結構如下：msg{mytype, nid, pid, cst, path}，其中 mytype 爲 message type 爲 0 或是 1，nid 表示發出訊息的節點 id，pid 爲 path id，cst 表示經過路徑的花費和，path 爲經過節點的集合。

此互斥多路徑的尋找演算法分爲兩個步驟。第一步由目的節點 t 發出訊息類別(Message type)爲 0 的訊息 msg{0, t, Φ, 0, (t)}到各個 neighbor，當網路中的節點 x 收到訊息 msg{0, nid, pid, cst, path}後，可分爲下列兩種情形：

1. 若是從 x 的 parent 所傳送來的訊息，則 x 將路徑加上 x 後，紀錄下來，並將訊息修改成 msg(0,x,(pid==Φ)?x:pid,cst+c(x,parent(x)),path+x)後，傳給 x 的 uptree 與 horizontal neighbor。
2. 若是從 x 的 horizontal neighbor 所傳送來的訊息，則檢查其中的路徑加上 x 後，是否與存在 x 節點中已知的路徑重疊，若無，則加入可傳輸的路徑內。此收到的訊息並不向外傳送。

第二個步驟是透過交換訊息類別(Message type) 1 的訊息來獲得更多的互斥路徑集合，訊息類別 1 的訊息由各個節點獨自發出。對於每個節點 x 記錄的替代路徑，x 將產生 msg{1, x, p.pid, p.cst, p}並且將此訊息傳給適當的 neighbor，所謂適當的 neighbor 需包含三個條件：(1) $v \in nbr(x)$ ；(2) $v \notin p$ ；(3)v 不爲 x 的 uptree neighbor。如此傳遞完之後即可得到互斥路徑集合。下頁圖 32、33 爲例：

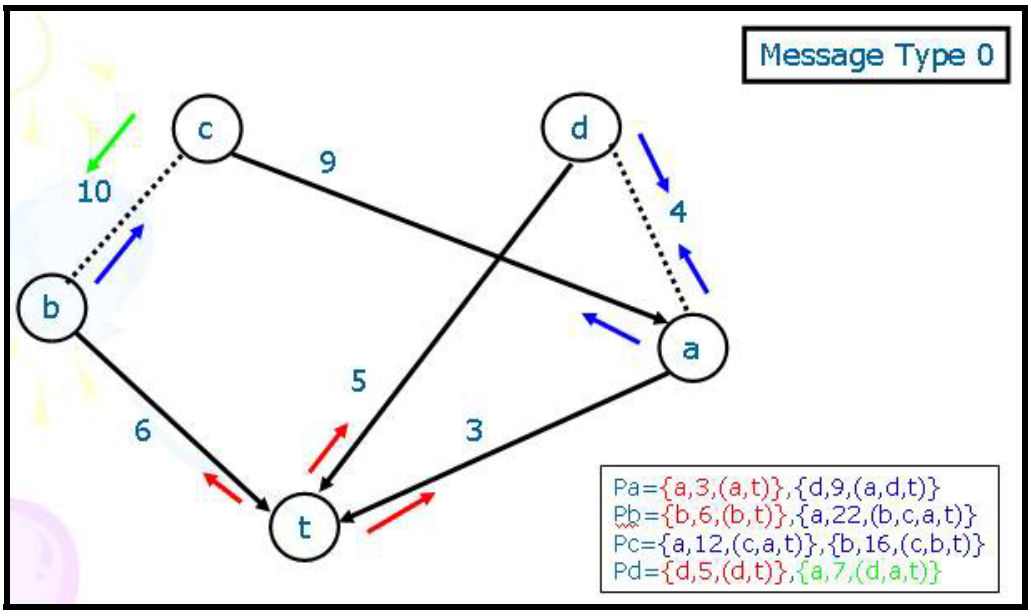


圖 32　互斥多路徑演算法訊息類別 0 傳送圖

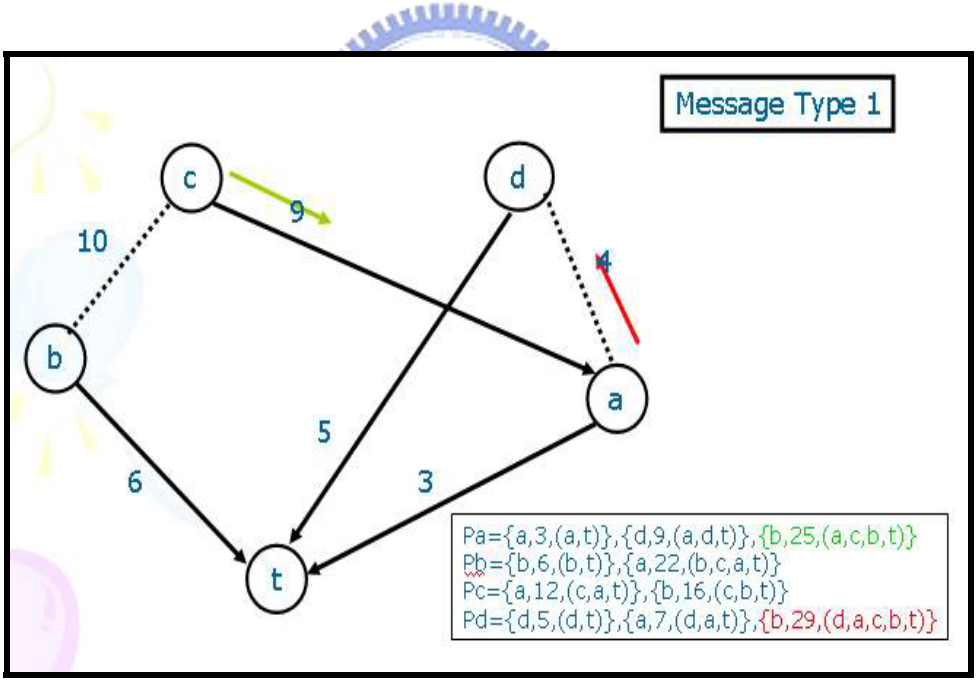


圖 33　互斥多路徑演算法訊息類別 1 傳送圖

四、多路由資料分享 VoIP 安全架構

這個章節將提出視覺加密概念與 SIP 結合後的系統架構，運用於網路電話的安全傳輸上，並且使用 cIPSec 於 SIP 訊號傳輸上的保護。首先簡介系統架構並提出一些系統的假設條件，接著介紹系統的使用流程以及詳細的訊號交換過程，最後介紹聲音封包分割、重組的處理方法。

4.1 系統簡介與假設

本系統的基本架構承襲於 2.1.3.1 節所介紹的 SIP 架構，採用主從式(Client-server)的架構，擁有的伺服器與用戶端種類也相同。爲了能和現有的網路架構相容，對於原先各個伺服器的功能與擺設位置做了一些改變，如下圖所示：

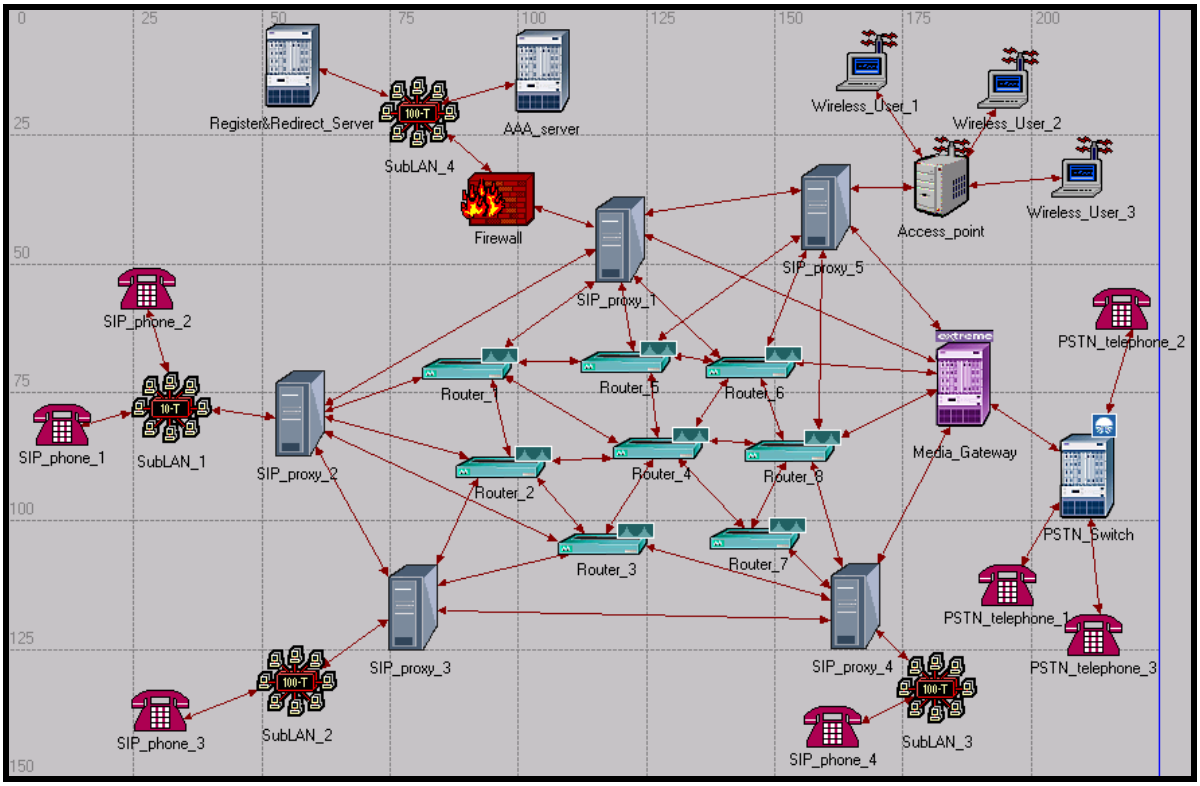


圖 34 系統架構圖

原先的 SIP 系統架構中，代理伺服器只擔任轉送封包的功能，並且可以與 register server 以及 redirect server 結合成一台伺服器。在我們的方法中，由於顧及目前所使用中的路由器大多是使用 Open Shortest Path First(OSPF)的路由通訊協定，若要全面改成採用互斥多路徑路由的可能性不大而且所費太高，所以將 SIP proxy server 獨立出來，並將原先只有封包轉傳的功能加以擴充，這個部份將於下一小節介紹。

除了 SIP proxy server 的改變之外，此系統有一些使用的預設條件以及限制，於底下一一列出：

- 區域網路(LAN)中的通話不提供安全保護的機制，此方法假設私人網路中的竊聽攻擊不存在或是很少，安全性較高，所以對聲音做資料分割傳送的部份只有在跨越不同的網路區域時，如圖 34 中，若是 SIP_Phone_1 與 SIP_Phone_2 建立通話連線，因爲不會透過外面的 SIP_Proxy_2，所以資料並不會被分割後傳送。本方法處理的情形如圖 34 中 SIP_Phone_1 與 SIP_Phone_4 通話時，才會在 SIP_Proxy_2 與 SIP_Proxy_4 間使用資料分享的方法以確保安全。
- 假設任兩節點間存在兩條以上互斥的路徑，爲了要讓兩個由同一個資料分割而成的封包不在同一條路徑上傳送而被攔截下來，所以我們假設同時有兩條以上的互斥路徑存在於網路上。爲了讓互斥路徑存在的越多，依據 [12] 所模擬的結果來看，平均每個節點的連結數(Degree)越高越能找到較多的互斥路徑。
- 假設 SIP proxy server 可以接受到附近大部份路由器所傳送的路由資訊訊息，如此可以幫助 SIP proxy server 在選擇傳送封包的兩條路徑時，可以得到較佳的路徑選擇(兩路徑延遲差距較小，且無相交之點)，如此可以減低傳輸上的延遲，並增加安全性。
- 假設訊息穿越防火牆或是網路位置轉換器(Network Address Translator)時，有相對應的解決方法，如 UPD hole punching 等，如此一來可以不用考慮封包穿越這些裝置時所產生的問題。

以上爲本論文中所提出新的 SIP 網路架構簡介，接下來我們將針對圖 34 中，SIP proxy server 的功能做詳細的介紹。

4.2 SIP 代理伺服器 (SIP proxy server)

由上一節的介紹得知，在我們規劃的網路電話安全傳輸架構中，爲了不更動現有網路實體的情形下，SIP proxy server 會有較大的改變，所以在這一小節中，將針對 SIP proxy server 做詳細的介紹。

除了原先擔任封包轉傳的功能之外，在我們提出的網路電話安全傳輸架構下，SIP proxy server 增加了下列的功能。

- 互斥多路徑路由(Disjoint multipath routing)，將 3.4 節所介紹的互斥多路徑路由演算法用於 SIP proxy server 上，並且可以接收原本路由器的訊息，如此可以在不必更動原本網路架構的情形下，達成多路徑傳送的目標。

- 資料分享(Data sharing)方法，將從使用者收到的聲音封包依照接下來章節(4.3)提到的方法加以分割後傳送；接收端的 SIP proxy server 則是將收到的兩個分割的聲音封包加以組合成原本的封包，再傳送到目的端的主機撥放。
- 與傳統電話網路 PSTN 相連接的部份可以與原先的媒體閘道器(Media gateway)結合，具有 SIP proxy server 與 PSTN media gateway 的功能，如上一節圖 34 中 media gateway 的部份。
- 幫助管理一個區域內的 SIP user 資料，此處類似行動系統(Mobile system)中的基地臺(Base Station, BS)的功能，幫忙紀錄通話時間以及使用者資料管理等，並且協助 user agent 認證與註冊的事宜。

接下來，我們將一一介紹 SIP proxy server 收到各種訊息後的處理方式：

- SIP proxy server 收到從 SIP phone 或是其他實體傳來的 SIP 訊息時，除了 Register 訊息將於下一節介紹有所不同外，其他訊息如：ACK、Ringing、OK... 等，與原先的 SIP proxy server 一樣，只做轉傳的動作。
- SIP proxy server 收到從一般路由器傳來的路由訊息後，與其他收到的路由訊息做整理後，得到目前網路路由情形表格，便利之後處理互斥多路徑傳輸上的使用。
- SIP proxy server 收到從其他 SIP proxy server 傳來的互斥多路徑路由訊息(3.4 節所介紹的訊息類別 0 與 1)後，依 3.4 節的演算法做相對應的處理，這個部份將於 4.5 節中介紹。
- SIP proxy 收到 RTP 封包時，會依據下圖所示的流程圖，做一系列的處理：

  1. 檢查 RTP 封包的來源位置是否爲該 SIP proxy server 所管轄的 SIP phone，若爲該 SIP proxy server 所管轄的 SIP phone 則進入步驟 2；若非，則進入步驟 3。
  2. 將 RTP 封包內的資料，以 4.4 節介紹的聲音封包處理後，封裝成兩個與原先封包標頭相同的 RTP 封包。然後依據 4.5 節方法得到的互斥多路徑資訊，取出最短的兩條路徑後，分別寫入兩個分享封包(Share packet)的 IP 標頭中。如果在 IP v4 的網路架構下，則放在 IP 標頭的 option 中；如果在 IP v6 的網路架構下，則放在 Hop by Hop 的擴充標頭(Extension header)中。最後將封包依路徑送出，離開流程。
  3. 檢查 RTP 封包的目的位置是否爲該 SIP proxy server 所管轄的 SIP phone，若是，則檢查暫存區是否存在另一個 share data，若存在，則組合後送給

SIP phone，若不存在，則置於暫存區中；若目的位置非所管轄的 SIP phone，則依據 IP 標頭中，option 欄(IP v4)或是 Hop by Hop 欄(IP v6)的資料傳送到下一個目標去。

4. 如果步驟 3 中，下一個目標與該 SIP proxy server 的連線已經中斷，則根據該 SIP proxy server 的路由資訊，與 RTP 標頭中 option 或 Hop by Hop 欄位比較後，取得一條互斥的路徑傳送。

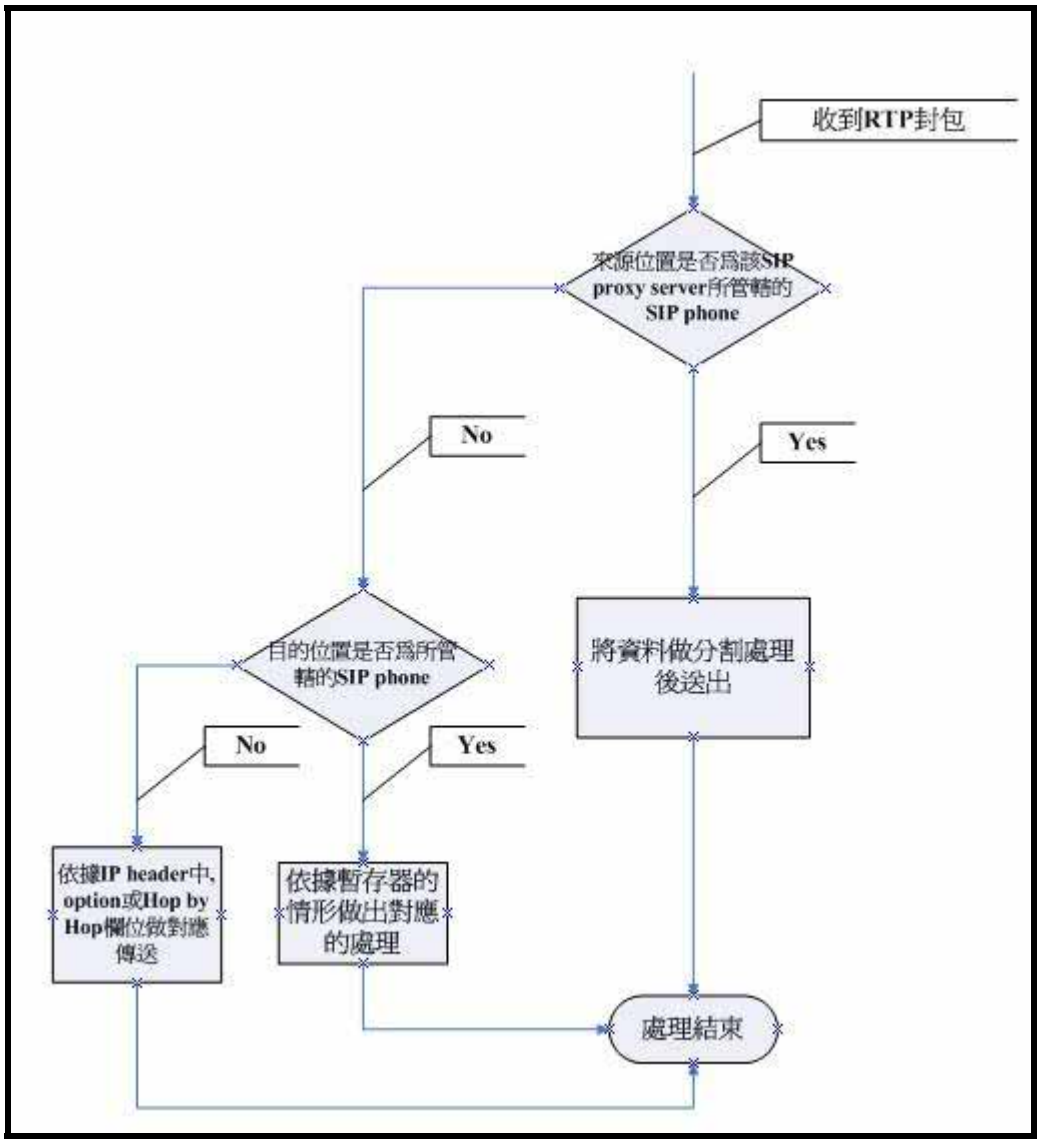


圖 35　SIP 代理伺服器 RTP 封包處理流程圖

以上是針對 SIP proxy server 在網路電話安全傳輸架構下，增加的功能與對封包處理的介紹。接下來的小節，我們將依不同的 scenarios，分別介紹在 4.1 節圖 34 的架構下，各種訊息與多媒體傳輸的情形。

4.3 系統流程與訊息交換

接下來我們就各種可能會發生的情形，分成不同的 scenarios，一一介紹其中的訊息交換與傳遞過程。在通訊建立訊息傳遞的部份，與原先 SIP 訊息傳遞的順序相似，最大的差異在於傳送媒體時的不同。

當一個 SIP phone 接上網路時，或是 SIP user agent software 被執行時，會先自動去尋找附近的 proxy server，設定成自己的 proxy server，接下來透過 proxy server 轉送 register 的訊息到 register server 註冊，scenario 1 介紹圖 34 中 SIP_phone_1 開起並執行 register 的訊息流程說明。

Scenarios 1 – 使用者註冊流程(User register)

Step 1：SIP_phone_1 與 SIP_proxy_2 建立 cIPSec 的安全關聯 (Security Association) 後，送出 REGISTER 的訊息到 SIP_proxy_2。

Step 2：SIP_proxy_2 收到 REGISTER 的訊息後，先暫存一些使用者資料後，與 SIP register server 建立 SA，並傳送 SIP_phone_1 的 REGISTER 訊息給 register server。

Step 3：當 Register server 通過認證並紀錄下 SIP_phone_1 的使用者資料以及所屬的 SIP_proxy_2 後，傳回 OK 訊息給 SIP_proxy_2。

Step 4：當 SIP_proxy_2 收到 OK 訊息後，將原先暫存的 SIP_phone_1 資料正式儲存，以方便後來的通話建立，然後傳 OK 訊息給 SIP_phone_1，register 動作完成。

Step 5：此爲選擇性(**Optional)**的部份，若要確保私人網路傳輸的安全，可以在 SIP phone 與 SIP proxy 之間使用簡單的加解密方法(XOR plaintext with key…)，此時可溝通用來加密的金鑰，由通訊雙方決定是否需要這個步驟。

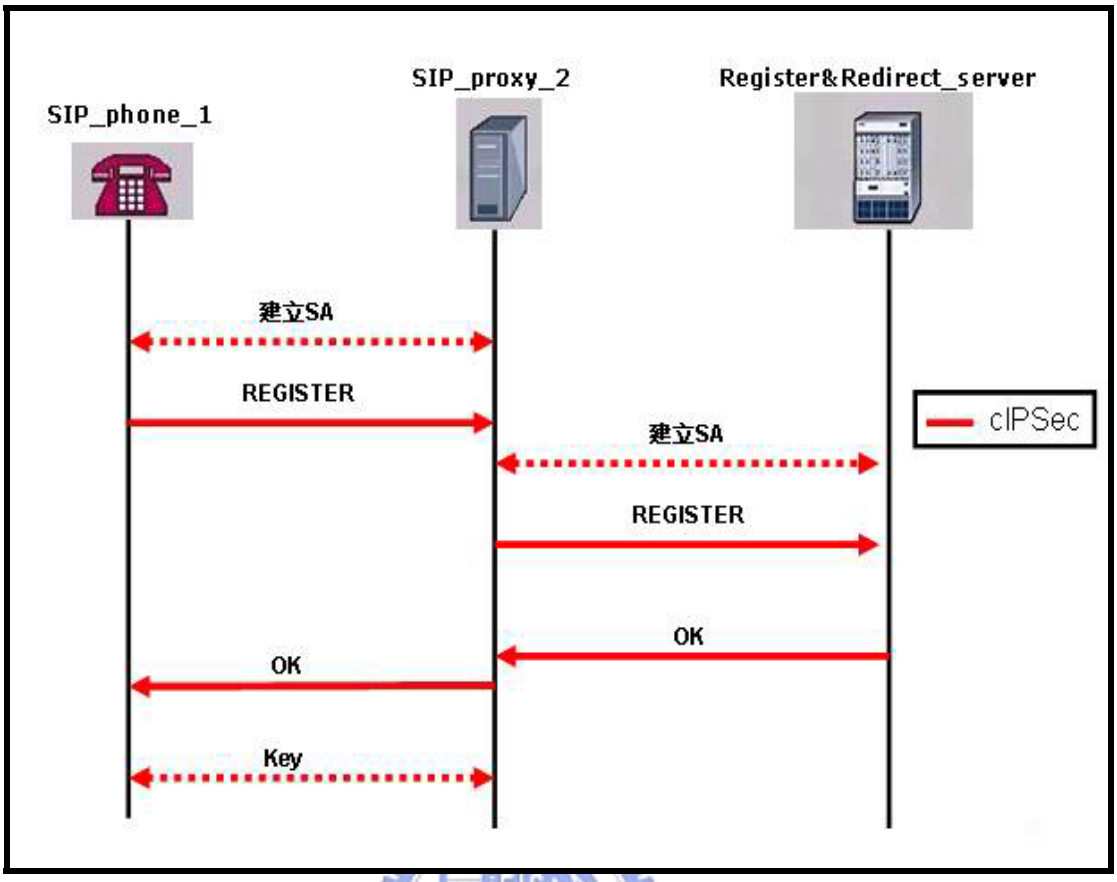


圖 36　場景一：使用者註冊訊息交換圖

Scenario 2 – 建立區域網路內的連線(Establish call in local area network)

在前一節(4.1)已經提過，在區域網路(Local Area Network)之間的 SIP call 不在本論文運用的範圍內，這裡只概略的列出在 LAN 建立通話的訊息傳遞情形。圖 37 中藍色線段使用的安全協定是由 SIP phone 雙方自行決定。

Step 1：SIP_phone_1 想要打電話給 SIP_phone_2，但是不知道 SIP_phone_2 的確實位置，所以送出需求(Request)訊息去尋問 SIP_phone_2 的位置。

Step 2：當 SIP_proxy_2 收到 SIP_phone_1 的需求(Request)後，會先去查詢本身的紀錄表格，如果 SIP_phone_2 的資料存在於 SIP_proxy_2 的紀錄中，則直接回應 SIP_phone_1。

Step 3：SIP_phone_1 收到 SIP_proxy_2 的資料後，回應 ACK 訊息給 SIP_proxy_2，並且發出 INVITE 訊息給 SIP_phone_2 建立連線。

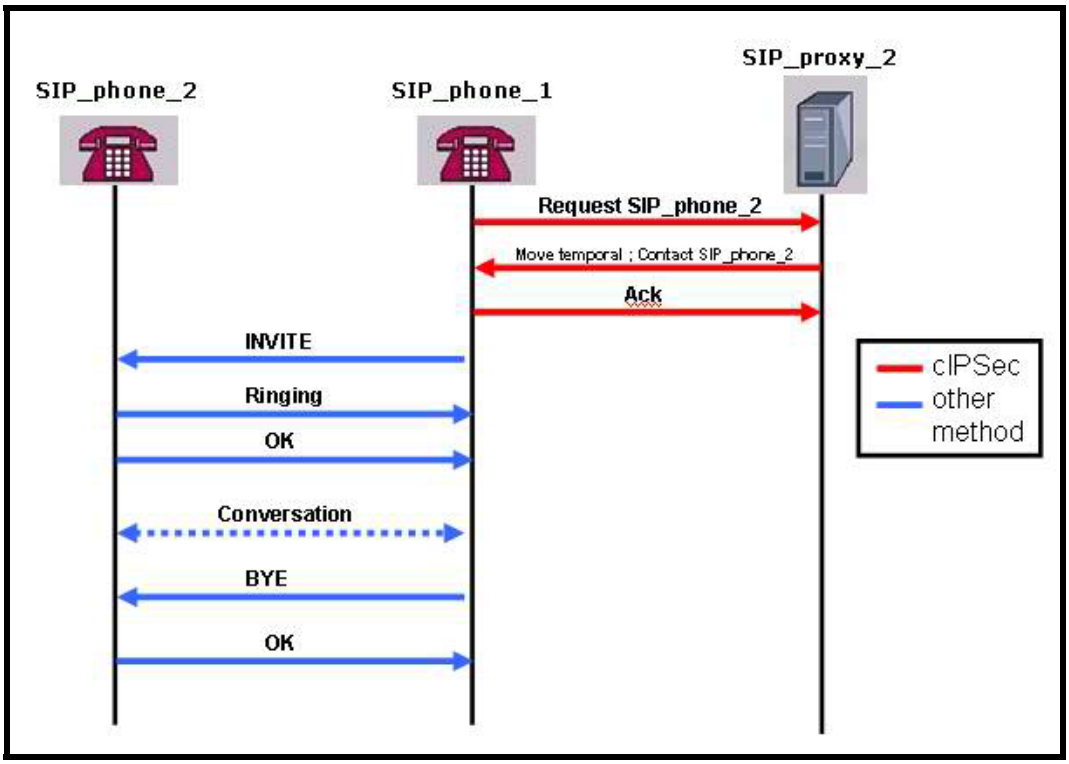


圖 37　場景二：建立連線於區域網路的訊息交換圖

Scenario 3 – 建立跨越網路的連線(Establish call across network)

Step 1：和 scenario 2 的 step 1 相同，SIP_phone_1 想要與 SIP_phone_3 建立通話，發出需求(Request)查詢 SIP_phone_3 的位置，SIP_proxy_2 收到後，因爲在自己的紀錄表格中查詢不到有關 SIP_phone_3 的相關資料，所以將需求轉向 Register&Redirect_server 來詢問，Register&Redirect_server 查詢得到結果後回傳給 SIP_proxy_2 與 SIP_phone_1。

Step 2：SIP_phone_1 依據收到的結果對 SIP_phone_3 發出 INVITE 訊息，經過 SIP_phone_3 傳回 Ringing 與 OK 訊息後，通話正式建立，而 SIP_proxy_2 開始紀錄通話相關資訊。(圖中 Ringing 與 OK 訊息一般而言，會分成兩次的訊息傳送，當收到 INVITE 時發出 Ringing 訊息，當收話端拿起話筒時才送出 OK 訊息，此處爲了繪圖方便才寫於同一個訊息上傳輸)

Step 3：通話開始後，當 SIP_proxy_2 從 SIP_phone_1 收到聲音封包後，會根據將於下一節(4.3)提到的方法，將封包分割成兩份，再根據目前網路的狀況(由路由器收到的資訊，加以整理後所得結果)在運用 3.4 節提到的互斥多路徑演算法找出最短的兩條路徑(可經過其他的 SIP proxy server)，分開傳送至

SIP_proxy_4，再由 SIP_proxy_4 將封包重組回原先的封包，傳送給 SIP_phone_3(如圖 39 所示)。

Step 4：通話期間，如果有其他訊息傳送，例如 SIP_phone_1 因爲偵測到網路可用頻寬減低，需要改變聲音編碼方式，則在訊息的傳送上，使用 cIPSec 來保護訊息的安全。

Step 5：當通話結束時，由發話或收話的任一端發出 BYE 訊息，收到 BYE 訊息的另一端則回應 OK 訊息後，通話正式結束，此時 SIP_proxy_2 紀錄這通電話的開始與結束時間，以方便之後電信公司查帳用與通聯紀錄查詢用。

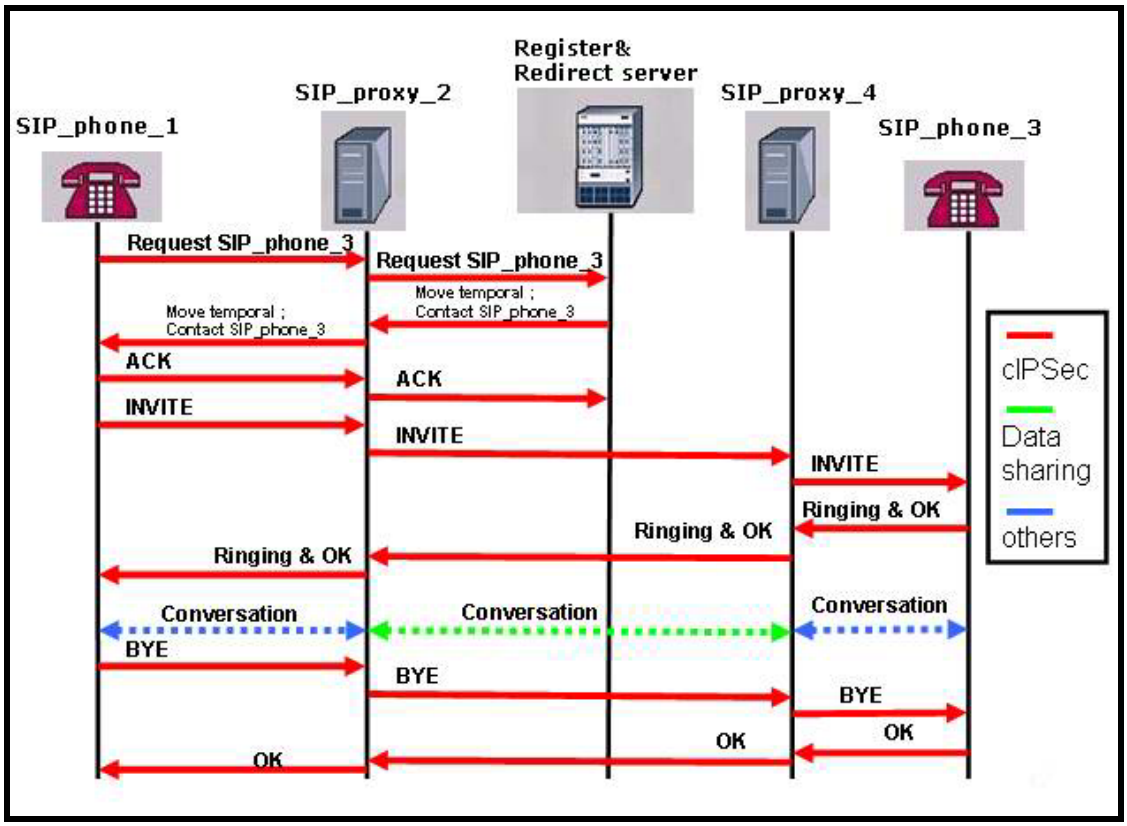


圖 38　場景三：建立跨越網路連線的訊息交換圖

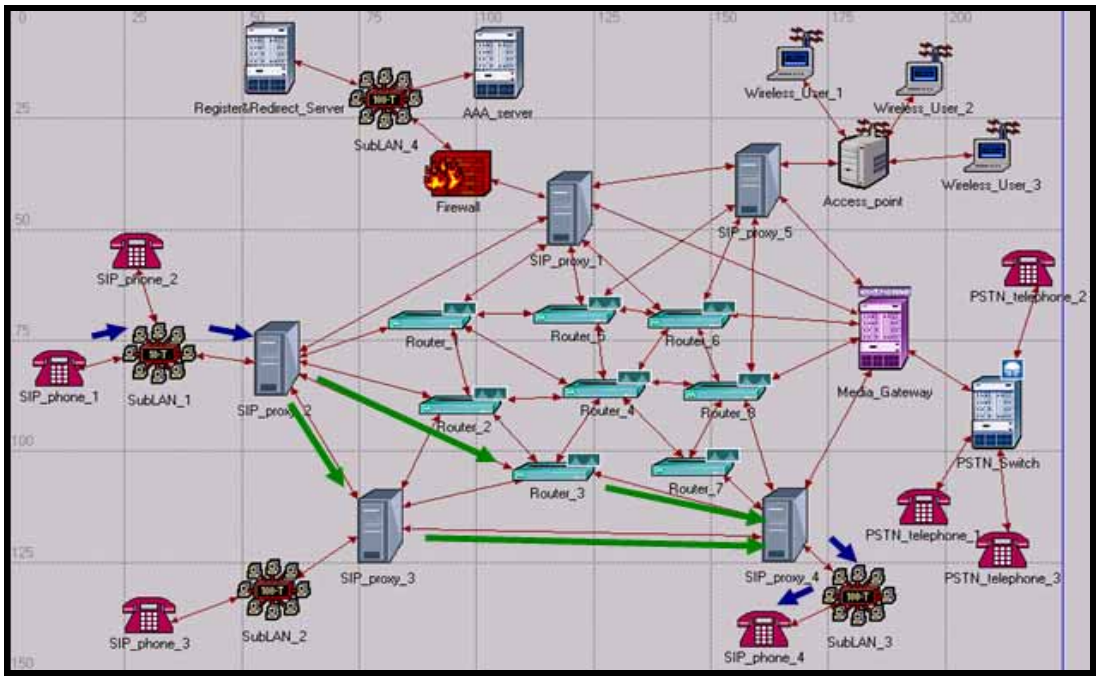


圖 39　資料分享用於聲音媒體傳送示意圖

Scenario 4 – 建立連線於 IP 網路與 PSTN 網路

這個 scenario 的訊息傳遞的情形與 scenario 3 相似，不同的地方在於原本 scenario 3 的 step 3 中，與 SIP_proxy_4 溝通的 INVITE message 要轉傳送給 Media Gateway Controllers(MGCs)，然後透過 MGC 來與 PSTN switch 溝通 call 的建立，並且保留 Media Gateways(MGs)上的資源給這通 call。

接下來傳送聲音封包的資料，一樣是由 SIP_proxy_2 將聲音封包分割，然後由 4.1 節圖 34 中的媒體閘道器扮演像 SIP proxy server 一樣的功能，將聲音封包組合回去，然後在 PSTN 網路上傳輸，如下圖所示：

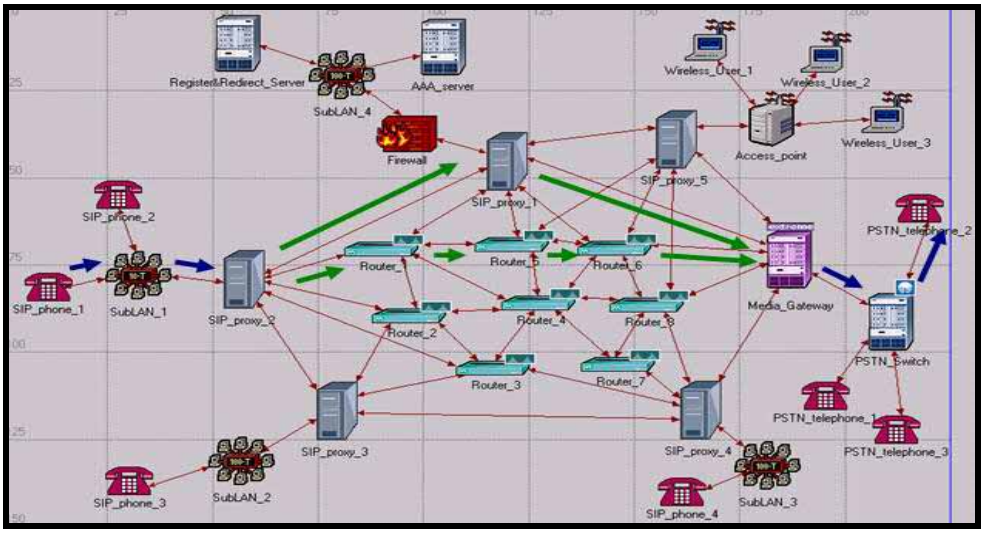


圖 40　資料分享方法與 PSTN 網路溝通示意圖

以上是幾種可能的通話建立情形用於新 SIP 架構上的訊息傳遞情形，至於如何對要傳送的聲音資料作分割，將於下一節中介紹。

4.4 聲音封包分割方法

本章節介紹以視覺加密爲基礎的資料分割方法，配合聲音編碼後的特性，發展出一套適合 VoIP 的資料分享(Data sharing)方法。底下以使用最廣泛的 G.711 聲音編碼方式爲例，介紹本論文使用的資料分割方法。

原本用於視覺加密的分割方式是爲了要配合人類視覺的特性，所以需要將原先的資料擴充爲原來的四倍，靠著兩張資料的重疊才能讀出原本的資料(請見 3.3 節)。如果直接運用在聲音資料的處理上，如此會產生原來資料的八倍大的空間，對於時間敏感度高(Time sensitive)的 VoIP 來說不是很適合，再加上數位化的處理不需要視覺上的特性，所以我們提出另一種分割的方式。

要縮小資料的增加量，最簡單的方式是將資料原先一個位元分割只分割成兩個位元，接下來就是要考慮分割需要用到的運算方式。一般常用的 AND 和 OR 都不適合，真值表如表 7 所示，以 OR 爲例，當獲得其中一個分享資料後，如果資料上的位元爲 1 則，原先同樣位置的位元值即爲 1，所以只剩下位元爲 0 的部份需要猜測，讓暴力法(Brute force)的可行性增加許多，所以需要其他的較公平的運算方法，來避免這種情形。

表 7　AND、OR 真值表

AND	0	1
0	0	0
1	0	1

OR	0	1
0	0	1
1	1	1

爲了使分割後不容易使用暴力法即能破解，所以需要一種運算方法，使真值表中得到的結果 0 與 1 的機率各爲 1 / 2，所以我們採用 XOR 的運算法。資料分割方法如下表所示，當原本的資料爲 0 時，share data 1 與 share data 2 的對應位元可分別 0、0 或是 1、1；原本的資料爲 1 時，share data 1 與 share data 2 的對應位元可分別 1、0 或是 0、1。當竊聽者只得到其中一個分享資料(Share data)時，若位元值爲 1，則原資料是 0 或是 1 的機率各爲 1 / 2。

表 8　資料分享方法

Original data	Share data 1	Share data 2	Result = share data 1 ^ share data 2
0	0	0	0
	1	1	0
1	1	0	1
	0	1	1

針對 VoIP 對時間的敏感度較高，所以我們希望將封包分割後能得到較小的成長幅度，如果將聲音封包的每個位元都依照表 8 的方式分割，則會得到兩個與原封包大小一樣的資料，則成長幅度爲兩倍，在這裡我們希望針對不同聲音編碼方式，做不同的處理，使用分割後封包大小的成長幅度較小。此處我們以 G.711 的編碼方式爲例，依照實驗的結果後(見 5.1 節)，可得到下圖中的資料分割方式，將原先給個樣本的前四個位元依據表 8 的方式分割，使得資料成長幅度爲原封包的 1.5 倍。(類似的方法亦可使用於 G.726 32Kbps 的編碼方式，只是將每個樣本分割的位元取前兩位)

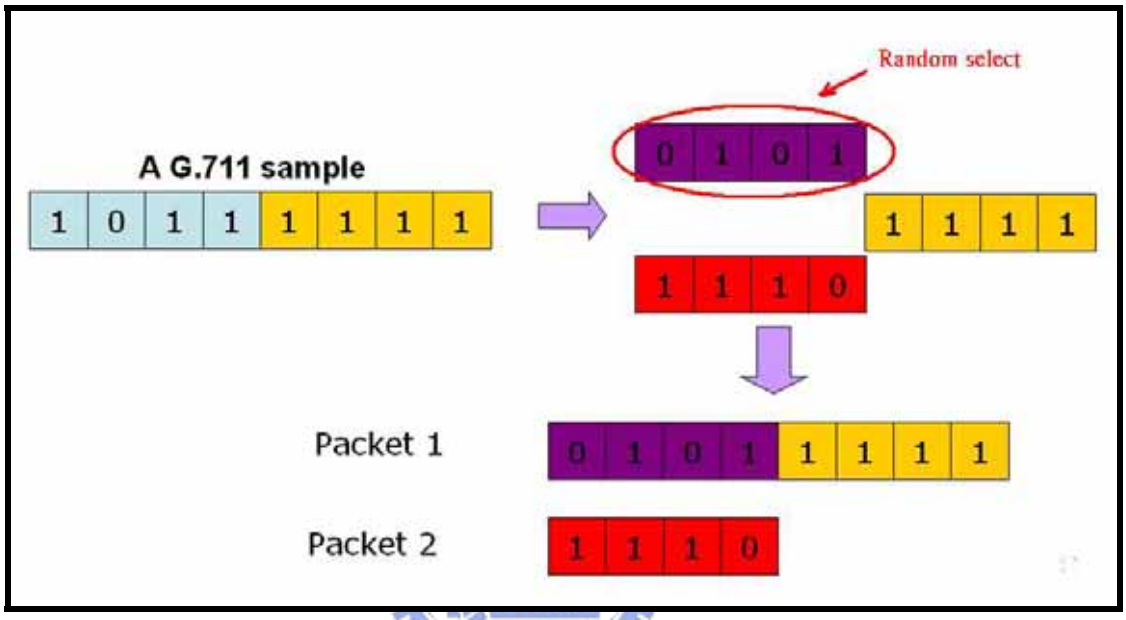


圖 41　資料分享方法範例

上圖中紫色的部份由亂數取得，接著將紫色部份的四個位元與原先封包相對應的位元(淡藍色部份)做 XOR 運算得 share data 2 資料(紅色部份)，黃色部份則是與原資料對應位元相同。

4.5　互斥多重路徑選擇演算法

在這個小節我們將以例子介紹如何在上述的架構下，找出互斥多重路徑的方法，主要引用的方法與 3.4 節所提的方法相似，所以網路的拓樸表示法也延用該節(3.4)的敘述，底下將以圖 42、43 與 44 爲例，分三個步驟一一說明。

Step 1：各個 SIP proxy server 盡可能的取到足夠多從各個路由器送出的路由資訊，然後計算出傳送到各個相鄰路由器到目的地會經過的路由器以及所需的花費，如下圖所示：

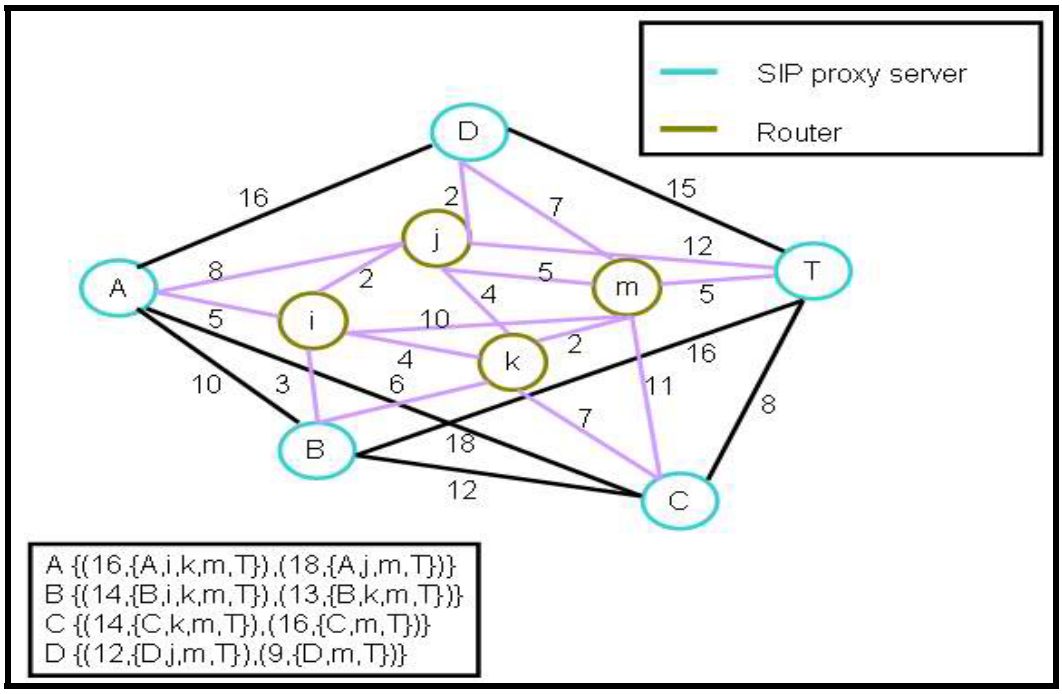


圖 42　互斥多路徑路由演算法步驟一

Step 2：與 3.4 節介紹的方法一樣，一開始由目的地的 SIP proxy server t 開始發出訊息類別爲 0 的訊號給相鄰的 SIP proxy server，收到的 SIP proxy server 依據訊息的內容計算是否有新的路徑可到達 t，如此一直傳遞到 SPT 的最後一個節點，結果如下圖所示：

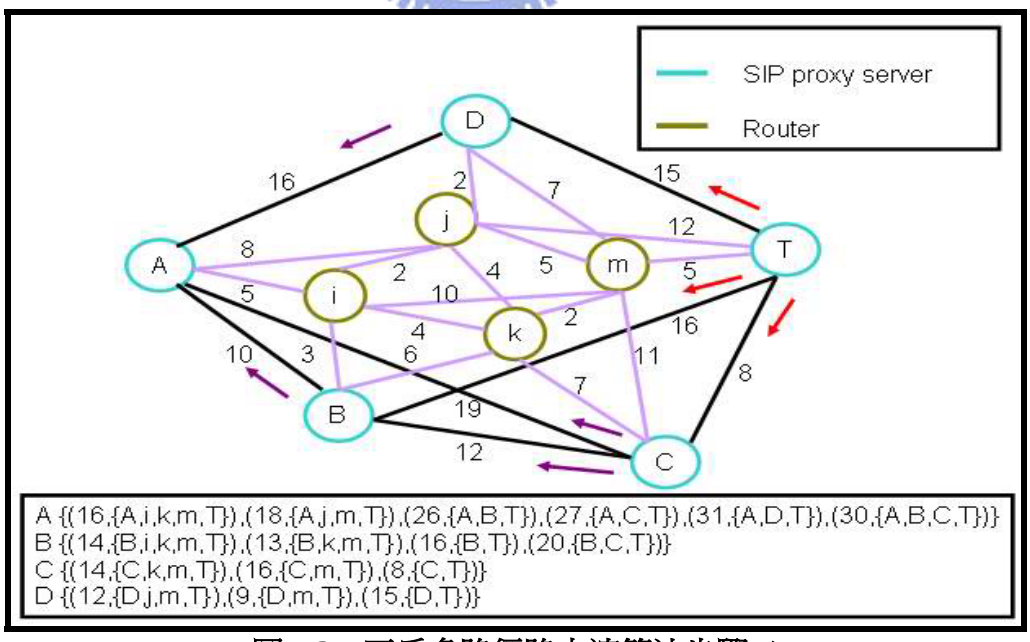


圖 43　互斥多路徑路由演算法步驟二

Step 3：接著由各個 SIP proxy server 決定是否發出訊息類別 1 的訊號給相鄰的節點，條件如同 3.4 節所敘述的一樣，如此可得到下圖的結果。

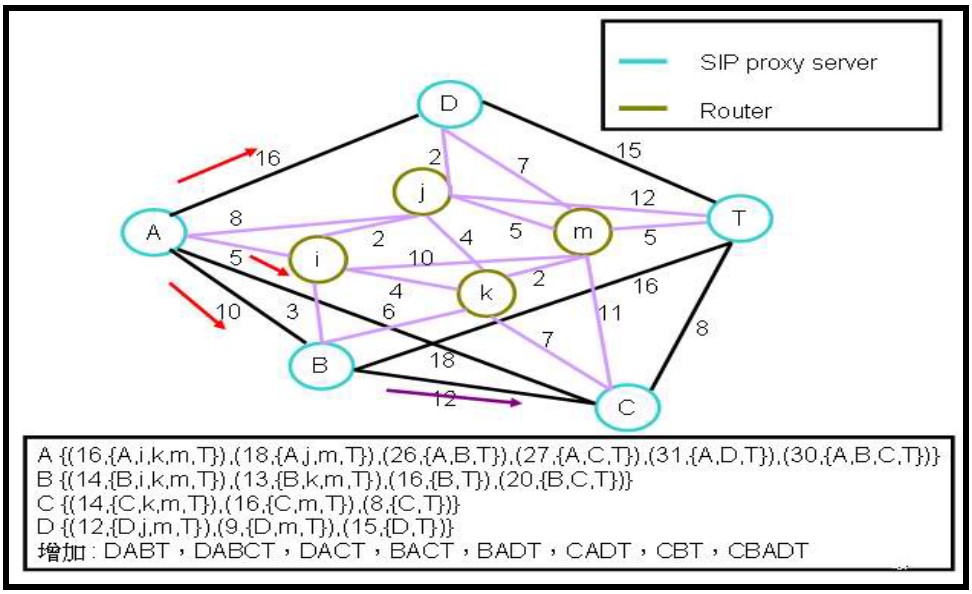


圖 44　互斥多路徑路由演算法步驟三

依照上述的步驟可以得到 SIP proxy server 之間的互斥多重路徑(Disjoint multi-path)，先決條件是各個 proxy server 可以收到足夠多的路由資訊，以 RIP(Routing information protocol)爲例，路由資訊最多可追溯到 16 個節點之遠，其他的路由方法則更多，如此幫助在路徑的選擇上避免經過重複的節點。

五、模擬結果

在這一個章節，我們將作一些模擬的實驗來證明上面提出方法的可行性，底下將分爲四個小節，分別介紹模擬環境、聲音分割的可靠性、其他加解密方法的處理時間比較，以及網路傳輸模擬。

5.1 模擬環境

在介紹各個模擬的結果之前，我們先介紹底下各小節所使用到的模擬環境。模擬所使用的硬體方面，CPU 爲 Intel Pentium-M 1.8 GHz 的筆記型電腦用 CPU，1GB 的 SDRAM。在軟體方面，使用的作業系統是 Microsoft 的 Windows 2000 server，撰寫 C 語言用的 IDE 爲 Microsoft 的 Visual C++，至於其他的應用軟體將於下面的章節中介紹。

5.2 聲音分割結果測試

在這個小節我們將對 4.4 節提出的聲音分割方法安全性做模擬，來驗證是否在該方法下，只獲得其中一個分享資料是不能聽出原來的內容。我們一個名爲 Vox Studio 3 的錄音軟體錄製 5 秒鐘 G.711 A-law 的格式，原本的聲音波型如下圖所示：

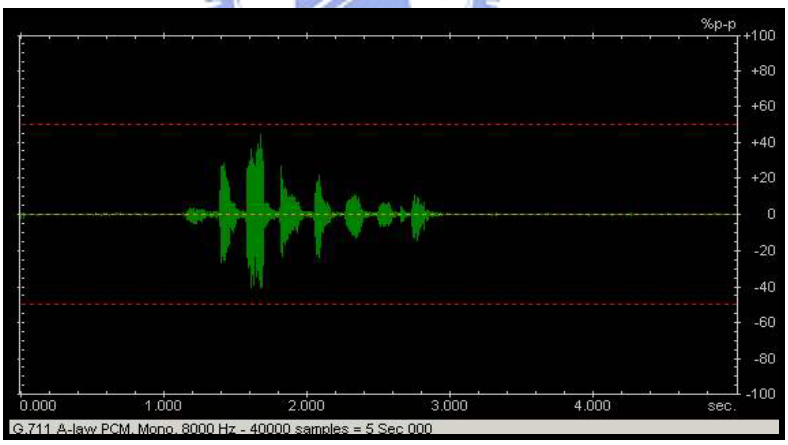
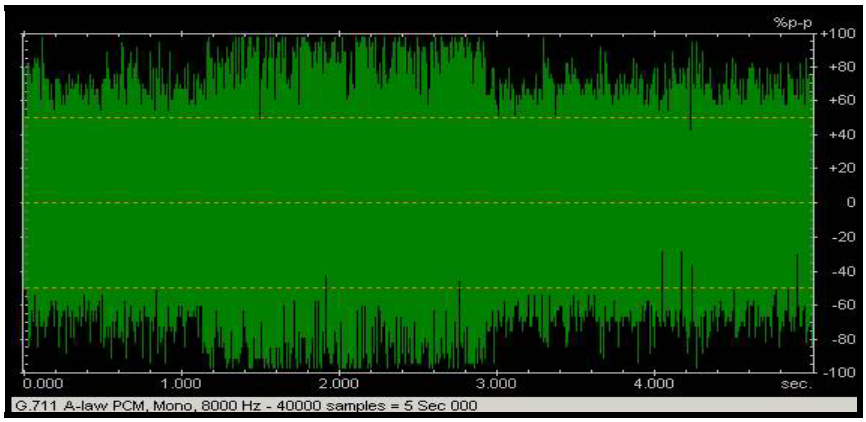


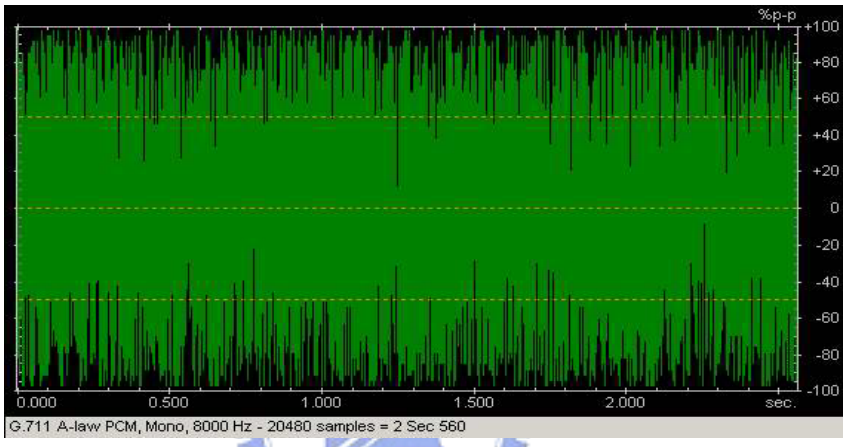
圖 45　5 秒的 G.711 A-law 聲音波型圖

經過 4.4 節的演算法處理後，share data 1(圖 46a)與 share data 2(圖 46b)的波形圖如下所示，聽起來與一般的雜訊相同，無法聽出確切的內容。

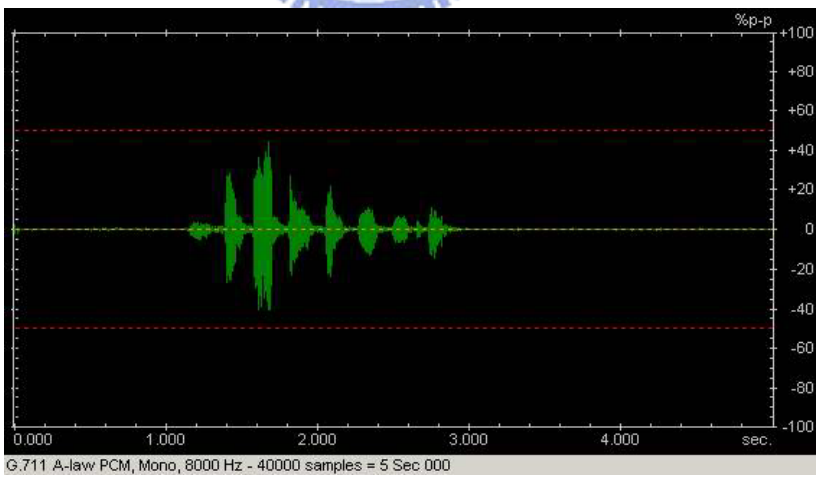
接收到 share data 1 與 share data 2 的一端，將 share data 1 每個樣本與 share data 2 取四個對應的位元做 XOR 後，可以得到下圖(圖 46c)的結果，即可得到原來聲音的原貌。



(a)



(b)



(c)

圖 46　使用資料分享方法於 G.711 聲音範例的波型圖

(a)Share data 1 (b) Share data2 (c) 將 Share data 1 and share data 2 組合後的結果

由 4.4 節的圖 41 中可以看出來，如果 share data 1 的封包被竊聽者攔截到，後四位的位元與原本資料的後四位元是完全相同，所以我們將原先的後四位元取出後，將前四

位元填 0，看看是否會得到原先的聲音資料，結果如下圖圖 47 所示，只能大略的分出那邊是有說話的部份，但無法取得原先的說話內容。

同樣的方法我們也應用於 G.726 32Kbps 的編碼方法上，因爲 G.725 32Kbps 是每 4 的位元爲一個樣本(Sample)，所以將原本的取前四個位元分割的方法改爲取前兩個位元分割，同樣的我們也得到無法取得原資料內容的結果，因此證實此方法可用於安全的防護上。

圖 47　將每個聲音樣本前四位元設爲零之後的波型圖

5.3　加密處理所需時間比較

由 3.1 節所介紹的聲音封包在網路上傳輸的特性來看，在收發兩端點對於封包的處理上，如果能減少時間的使用量，對整體而言是有不錯的幫助，此處所指對於封包的處理包括：聲音編碼、聲音資料的加密、封裝成封包、接收端的消除振盪(Dejittering)動作以及接收端解密與解碼的動作。如果在上面的步驟中，能有效的縮短花費的時間，對整體的傳送品質有不錯的幫助。

所以在這個小節中，我們固定聲音的編碼方式以及需要加密的聲音資料大小，針對資料處理上(加、解密)所花費的時間做模擬以及比較。底下列出這個模擬中，所有的環境條件以及使用的軟體等。

表 9　加密環境列表

CPU	Intel Pentium-M 1.8GHz
RAM size	1 Gigabytes
OS	Microsoft Windows 2000 server
Encryption software	KRYPTOS v1.0，爲 GMU educational software toolset 之一。

Complier environment for data sharing method	Microsoft Visual C++ 6.0
Encryption data	40 Kbytes G.711 A-law sample (圖 45)

而各個拿來比較的加密演算法(DES、triple-DES、Rijndael、RC6、RSA)的環境參數如下表所示：

表 10 加密演算法參數列表

Encryption name	Key size	Mode	IV size
DES	64 bits	CBC	64 bit
Triple DES	128 bits	CBC	64 bits
Rijndael	128 bits	CBC	128 bits
RC6	128 bits	CBC	128 bits
RSA	Public：1280 bits Private：5064 bits	CBC	2048 bits

底下是我們經過模擬後所得到的結果，每個演算法對同一個資料加(解)密處理共 1000 次後所得的花費時間取平均值，結果如下表與圖 48 所示。

表 11 資料加密時間比較表

對資料加密的部份：

Algorithm	DES	3DES	Rijndael	RC 6	RSA	Data sharing
Average Time (*ms*)	9.16	16.45	4.90	8.40	431.22	1.25

對資料解密的部份：

Algorithm	DES	3DES	Rijndael	RC 6	RSA	Data sharing
Average Time (*ms*)	9.70	17.30	5.33	7.00	14019.04	0.25

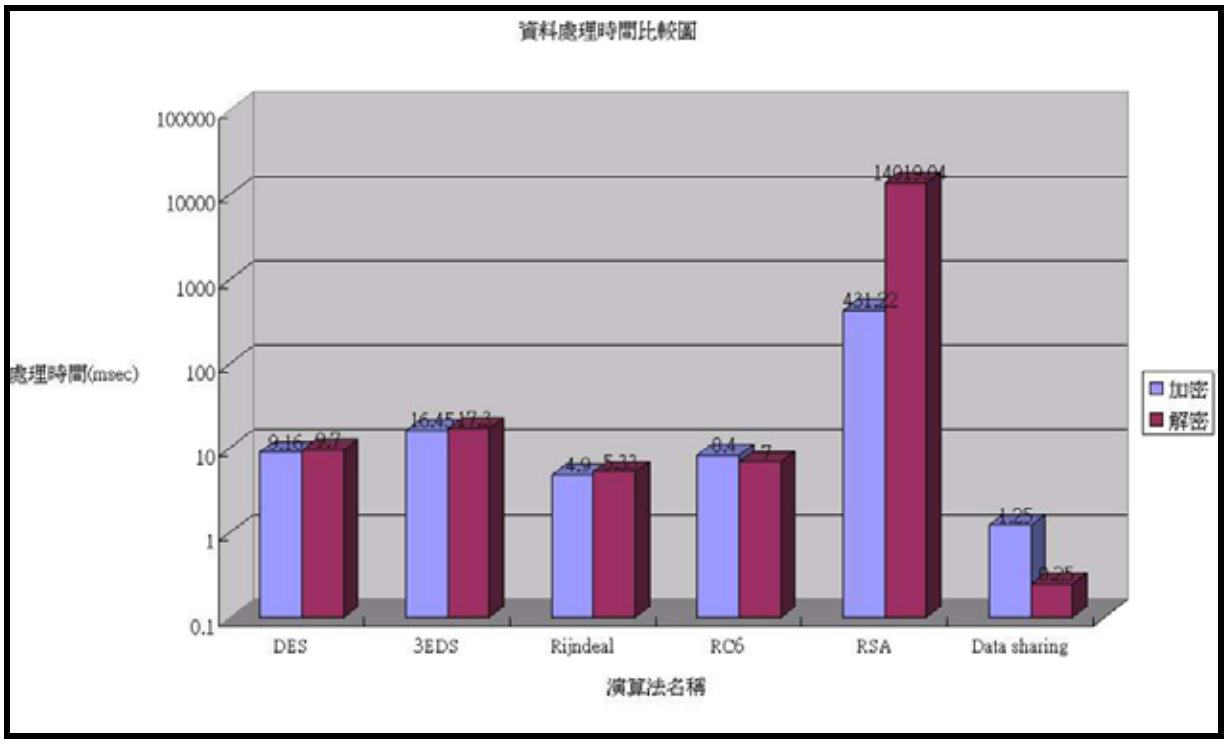


圖 48 資料加密時間比較圖 – 1

爲了能夠與現實的情形更加的相近，我們參照 3.1 節中表 4 的資料顯示，預設的 G.711 聲音封包大小：160 bytes，從原先的聲音封包中，取出 160 bytes 後，再以下面的環境(表 12)做了與上面相同的模擬，其中各個加、解密演算法的參數與表 10 所列相同沒有更動，經過每個演算法 1000 次的運算後，得到的結果平均後如表 13 與圖 49 所示。

表 12 加密環境列表 – 2

CPU	Intel Pentium-M 1.8GHz
RAM size	1 Gigabytes
OS	Microsoft Windows 2000 server
Encryption software	KRYPTOS v1.0，爲 GMU educational software toolset 之一。
Complier environment for data sharing method	Microsoft Visual C++ 6.0
Encryption data	160 bytes G.711 A-law sample

表 13　資料加密時間比較表 - 2

對資料加密的部份：

Algorithm	DES	3DES	Rijndael	RC 6	RSA	Data sharing
Average Time (μ s)	99	160	74	71	11121	6.0

對資料解密的部份：

Algorithm	DES	3DES	Rijndael	RC 6	RSA	Data sharing
Average Time (μ s)	101	160	71	74	140515	2.8

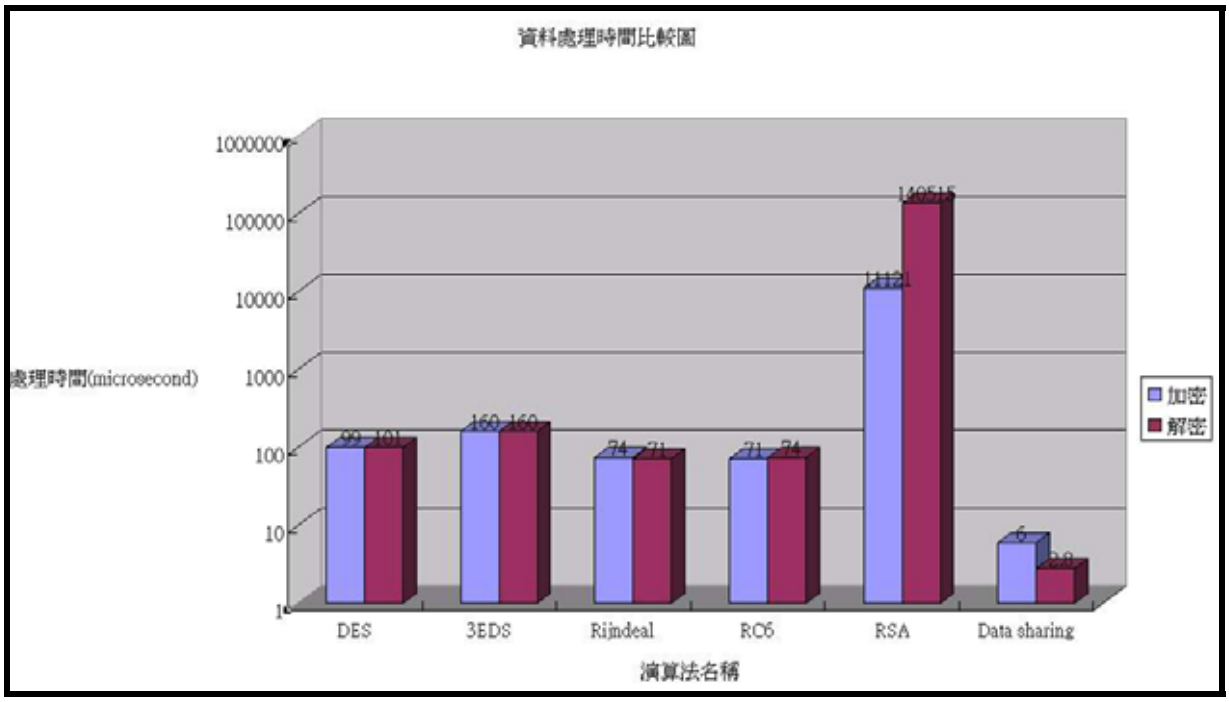


圖 49　資料加密時間比較圖 – 2

由上面的模擬結果來看，很明顯的本篇論文中所提及的 data sharing 的方法較傳統加、解密演算法所需要的處理時間來的少很多，以 5 秒的聲音爲例，相較於 3DES 加解密時間來的少約 32 *ms*。如此一來，在端點上的處理時間可以縮短很多，有利於減少 Mouth-to-ear 的延遲時間，增加整個 VoIP 的通話品質。

除了在端點處理聲音封包的過程需要花時間之外，最大的延遲發生就存在於網路傳輸上，因爲本文中提及的方法需要另一條路徑來傳送 share data 2 的封包，會經過非最短路徑的其他路徑，所以下一個小節我們將討論在網路上傳輸的延遲影響，並作對應的模擬。

5.4 網路傳輸模擬

在 3.1 節曾經提及，網路傳輸的延遲大至上可分爲傳送延遲(Propagation delay，5μsec/km)與排程的延遲(Queuing delay)，其中，傳輸延遲與資料大小無關，是存在於傳輸上的基本延遲，而排程延遲則發生在封包進入路由器(Router)到被送往正確的傳送連結上所需要的時間，這個部份與網路狀況以及封包的大小有著密切的關係，本節將針對這個部份作模擬。

由 3.2 節(圖 23)的研究來看，使用 IPSec 大致會使封包成長至約 1.5~2.0 倍，而在資料分享方法上，第二個分享資料的封包大小約爲原來的 0.5 倍，所以底下我們將針對封包大小分別爲 240 bytes 與 80 bytes 在經過網路節點時，在不同網路負載的情形下，所產生的延遲差異來討論。

我們以表 14 所列的環境以及圖 50 的網路拓樸排序來模擬上述的情形，測量封包大小爲 80 bytes 與 240 bytes 分別經過 3 個內部節點(Internal node)後(由圖 50 中的 Node 0 傳送到 Node 4)，到達目的地所需要花費的時間，其中所有經過的連線傳送延遲(Link propagation delay)皆爲 10*ms*(除了 node 5 與 node 6 之間的連線之外)，頻寬爲 100Mbps，並且分爲 no traffic in network、3 ftp traffic in network、6 ftp traffic in network、9 ftp traffic in network、20 ftp traffic in network 與 30 ftp traffic in network 六種情形來模擬，得到的結果如表 15 與圖 51 所示。

表 14 網路模擬環境列表

CPU	Intel Pentium-M 1.8GHz
RAM size	1 Gigabytes
OS	Microsoft Windows 2000 server
Simulation software	Network Simulation 2 (Version 2.27)
Packet size	80 bytes UDP/IP 240 bytes UPD/IP
Link state	Delay：10 *ms*，Bandwidth：100Mb，Queue：droptail

從模擬結果來看(圖 51)，平均封包在節點的延遲大致與封包大小比例成正比(1:3)，但是隨著網路的壅塞程度漸長，封包較小者的延遲成長幅度較大，而封包較大者延遲成長幅度較小，但是兩者在比例上仍然與封包大小比例不相上下。

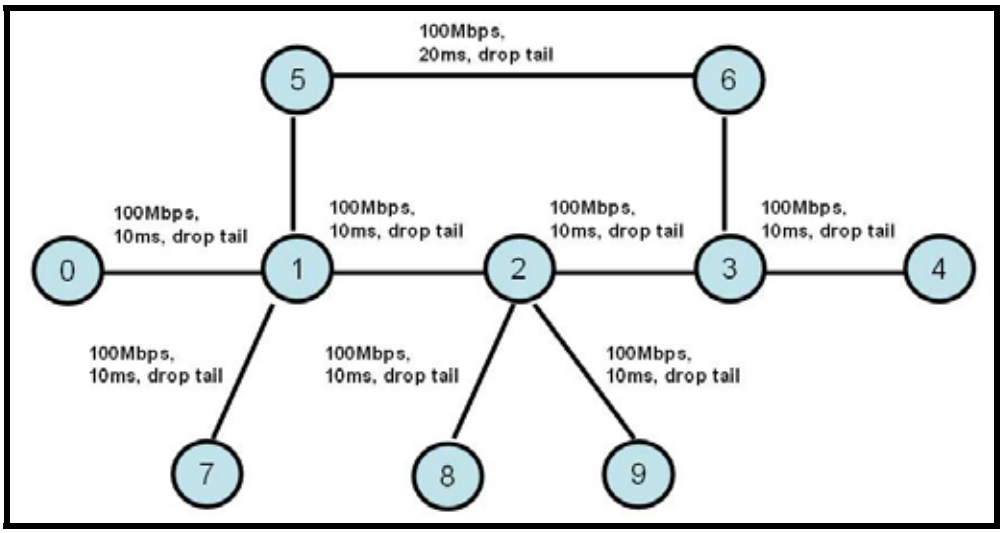


圖 50　網路模擬所使用的網路拓樸架構

表 15 網路模擬結果

Network state / Packet size	No traffic	Traffic with 3 ftp streams	Traffic with 6 ftp streams	Traffic with 9 ftp streams	Traffic with 20 ftp streams	Traffic with 30 ftp streams	Traffic with 40 ftp streams	Traffic with 50 ftp streams	Traffic with 70 ftp streams
80 bytes – average node delay (*μsec*)	8.533	8.533	9.324	9.406	9.686	9.905	10.378	12.087	16.280
240 bytes – average node delay (*μsec*)	25.60	25.600	26.335	26.366	26.354	26.775	26.989	28.98	35.236

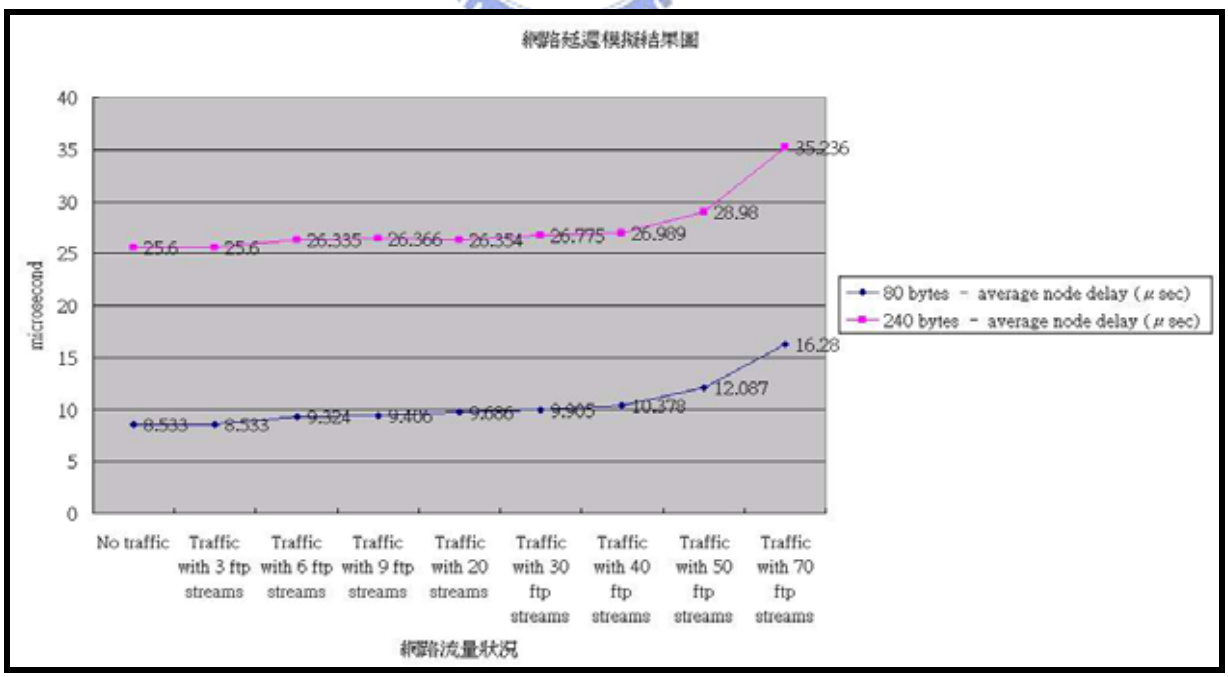


圖 51　網路模擬結果

六、結論

6.1 討論與結論

在這篇論文中，我們試圖結合視覺加密法(Visual cryptography)的基本精神與 VoIP，希望能夠達成一個簡單且快速的安全架構，用來取代原先需要對加密金鑰作保護的加解密方法。

經過第五章的模擬之後，從模擬得來的結果，我們列出這篇論文提出的方法可以達成的幾個項目：

- 可以提供安全不被竊聽的環境。由 5.1 節的模擬結果來看，在網路中間的攔截者無法從單一的分享封包得知原本封包的內容，如此在防止竊聽的安全性上，提供一個不錯的選擇。
- 減少在通訊建立時的訊息傳輸。因爲省略掉了一般加密方法(對稱加密演算法)中，金鑰的建立、傳送與保護的步驟，可以大大的減少安全通訊前需要傳送的訊息量，並且不需要定期的更換金鑰來保證安全性的存在。
- 可以快速的對需要保密的資料做處理。從 5.2 節中，我們可以看出本方法在對資料的處理上較一般常用的加密演算法來的快速，無論是在加密處理或是解密處理的過程。我們也可以從模擬的結果與 3.1 節的資料中看出，傳送 160 bytes 的封包，與 Rijndael 演算法相比較(所有列舉的加密演算法中最快者)，只要在路徑選擇時，兩條路徑的差異小於約 27 公里 ($(\frac{(74+71)-(6.0+2.8)}{5} = 27.24(km))$)，並且在選擇第二條路徑時，中間經過的節點數不大於最短路徑所經過節點數的 3 倍，則可較 Rijndeal 加密演算法來的快速，如此較爲適合如 VoIP 這類時間敏感度高(Time sensitive)的應用。
- 適合直接使用於現行環境中。在架構上我們考慮到目前的路由器所使用的路由方式爲單一最短路徑(Single shortest path)，如果直接更動的可能性不大，所以架構中將互斥多路徑演算法的功能放置於 SIP proxy server 上，不需要對原先網路的架構做極大的更動，適合本論文方法的實作，也減少更動所需要的成本。

6.2 未來工作

本論文提出一個安全傳輸聲音資料於 IP 網路的架構，但仍有一些細節的部份需要更進一步的研究或是可以擴充的地方，底下我們列出一些可改進或是需要填補的地方。

- 找出適合傳輸的互斥多重路徑。由於這個方法需要將封包分開傳送才能達到安全性，可是又要考慮到 VoIP 即時傳送的特性，所以需要找到兩條延遲時間相似的路徑，這邊需要一個更好的演算法來輔助。
- 尋找最佳的 SIP proxy server 配置。承襲論文提出的架構以及上一點的特性，如何配置 SIP proxy server，以及中間的連線關係需要一些統計與測試的結果，來找出最適合的配置，而所需要的花費最少。
- 運用於其他類型的多媒體傳輸安全。由於 VoIP 有即時傳輸的特性，所以可以將這種網路架構推廣到其他多媒體傳輸的安全確保上，如網路電視…等，針對不同媒體的特性做不同的分割，讓此網路架構可重覆利用，也大大的減少鋪設的花費，對業者或是使用者都能得到利益。

參考文獻

[1] D. Collins, Carrier Grade Voice over IP, McGraw-Hill, New York, 2003.

[2] Swades De, Sajal K. Das, "Dynamic Multipath Routing (DMPR): An Approach to Improve Resource Utilization in Network for Real-Time Traffic", University of Texas at Arlington, 2001.

[3] P.-J. Chuang, H.-Y. Tu, "Dynamic scheme for reducing hot-spot effects in multipath networks", IEE Proc. –Comput. Digit. Tech., Vol. 146, No. 4, pp.179-184. July 1999.

[4] Swades De, Chunming Qiao, "Does Packet Replication Along Multipath Really Help?", State University of New York at Buffalo.

[5] Mohan Krishna, Ranganathan, Liam Kilmartin, "Investigations into the Impact of Key Exchange Mechanisms for Security Protocols in VoIP Networks", National University of Ireland, Galway, Ireland, 2001.

[6] Roberto Barbieri, Danilo Bruschi, Emilia Rosti, "VoIP over IPSec：Analysis and Solution", Computer Security Applications Conference, 2002, Proceedings, 18th Annual, 9-13, pp. 261 - 270, Dec. 2002

[7] Haritha Phalgun, "The Effect of Voice Packet Size on End-to-End Delay in 802.11b Networks", M.S., University of Pittsburgh, April 29, 2003.

[8] Jan Janssen, Danny De Vleeschauwer, Maarten Büchli, Guido H. Petit, "Assessing Voice Quality in Packet-Based Telephony", IEEE Internet Computing, 1087~7801/02 ©2002 IEEE, pp. 48~56, May-June 2002.

[9] Fabrice Poppe, Denny De Vleeschauwer, Guido H. Petit, "Choosing the UMTS Air Interface Parameters, the Voice Packet Size and the Dejittering Delay for a Voice-over-IP Call between a UMTS and a PSTN Party", Alcatel Bell, 2002.

[10] Moni Naor, Adi Shamir, "Visual Cryptography", Eurocrypt '94 Proceeding LNCS, 950:1–12, 1995.

[11] Deepinder Sidhu, Raj Nair, Shukri Abdallah, "Finding Disjoint Paths in Networks", ACM SIGCOMM Computer Communication Review, Volume 21 41-53, September 1991.

[12] Wenjing Lou, Yuguang Fang, "A Multipath Routing Approach for Secure Data Delivery", IEEE Military Communications Conference (MILCOM 2001), Mclean, VA, USA, Oct 2001

[13] Bao Hong Shen, Bin Hao, Arunabha Sen, "On Multipath Routing using Widest Pair of

Disjoint sets", High Performance Switching and Routing, 2004, pp.134-140, Apirl 2004

[14] H. Schulzrinne, et al, "RFC1889: RTP: A transport Protocol for Real-Time Application", January 1996.

[15] J. Rosenberg, et al, "RFC 3261: SIP : Session Initial Protocol", June 2002.

[16] M. Handley, V. Jacobson, "RFC 2327: SDP : Session Description Protocol", April 1998.

[17] H. Schulzrinne, A. Rao, R. Lanphier, "RFC 2326: Real Time Streaming Protocol(RTSP)", April 1998

[18] M. Handley, C. Perkins, E. Whelan, "RFC 2974: Session Announcement Protocol", October 2000.

[19] M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett, "RFC 2705: Media Gateway Control Protocol (MGCP)", October 1999

[20] F. Cuervo, et al, "RFC 3015: Megaco Protocol Version 1.0", November 2000

[21] S. Kent, R. Atkinson, "RFC 2401: Security Architecture for the Internet Protocol", November 1998.

[22] ──, "RFC 2402: IP Encapsulating Security Payload (ESP)", November 1998.

[23] ──, "RFC 2403: IP Authentication Header", November 1998.

[24] D. Maughan, et al, "RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP)", November 1998.

[25] D. Harkins, D. Carrel, "RFC 2409: The Internet Key Exchange (IKE)", November 1998.

[26] Recommendation ITU-T G.114, "One-Way Transmission Time", Int'l Telecommunication Union, Geneva, 1996.

[27] Recommendation ITU-T G.711, "Pulse code modulation (PCM) of voice frequencies", Int'l Telecommunication Union, Geneva, November 1988.

[28] Recommendation ITU-T G.723.1, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s", Int'l Telecommunication Union, Geneva, March 1996.

[29] Recommendation ITU-T G.726, "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)", Int'l Telecommunication Union, Geneva, December 1990.

[30] Recommendation ITU-T G.729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)", Int'l Telecommunication Union, Geneva, March 1996.

[31] Recommendation ITU-T H.225 version 4, "Call signaling protocols and media stream packetization for packet-based multimedia communication systems", November 2000.

[32] Recommendation ITU-T H.245 version 8, "Control protocol for multimedia

communication", July 2001

[33] Recommendation ITU-T H.323 version 4, "Packet-based multimedia communications systems", Int'l Telecommunication Union, Geneva, November 2000.

[34] Recommendation ITU-T H.248, "Gateway Control Protocol", Int'l Telecommunication Union, Geneva, November 2000.

[35] Recommendation ITU-T P.800 "Methods for subjective determination of transmission quality", August 1996.

[36] 賴溪松、韓亮、張真誠，近代密碼學及應用，期標出版股份有限公司，臺北，民國92年。

[37] 近代密碼技術，URL：http://sna.csie.ndhu.edu.tw/~cnyang/RecentCrypto/

[38] Vocaltec，URL：http://www.vocaltec.com

[39] Net2Phone，URL：http://web.net2phone.com/home_inten.asp

[40] Skype，URL：http://www.skype.com/

[41] AT&T，URL：http://www.att.com/

[42] FCC，URL：http://www.fcc.gov/

[43] G.723.1 在數位訊號處理器的即時實現，
URL：http://roger.ee.ncu.edu.tw/chinese/pcchang/course98b/vq/g/