安全語音即時通訊系統的設計與實作

Design and Implementation of a Secure Voice Chat of Instant Messaging

蔡金龍
高雄師範大學資訊教育研究所
和平一路116號
高雄市802苓雅區
goadragonason@gmail.com

楊中皇
高雄師範大學資訊教育研究所
和平一路116號
高雄市802苓雅區
chyang@nknucc.nknu.edu.tw

摘要

現在的即時通訊，不只有文字通訊的功能，更具有語音通話的功能。提供使用者更快速的溝通方式。然而在2007年英國VoIP專家Peter Cox，發布了SIPtap，一種用來監聽VoIP的軟體，向大家證明現在多數的網路語音是可以被監聽的。現今主流的即時通訊軟體的安全機制，只有在用戶端登入主機時，針對其使用者帳號和密碼進行加密傳送，而身份確認後，之後的資料傳送都是明文型式。有鑑於此，Kikuchi, Tada 和 Nakanishi 提出了採用Diffie-Hellman Key Agreement Protocol以確保用戶端之間訊息的保密性。而M. Mannan, 和 P. C. van Oorschot 提出名爲IMKE (Instant Messaging Key Exchange) 的安全即時通訊協定，而郭宗益 改進 Mannan的IMKE協定，採用質數體橢圓曲線密碼學爲基礎，提出安全即時通訊與展現協定 (Secure Instant Messaging & Presence Protocol, SIMPP) 。

本研究以SIMPP爲基礎，進一步實做出安全的即時通訊語音功能。我們利用Winmm函式庫取得使用者的語音數據資料，引用開放原始碼OpenSSL密碼學函式庫，使用SIMPP產生的短期共同金鑰，以AES128進行語音加密後才將語音資料送出，受話方取得語音資料後進行解密，再將語音資料轉成聲波。若有心人士從中取得語音網路封包，也只能聽到噪音，無從得知其談話內容，可確保談話的安全性。

關鍵詞: 即時通訊、語音安全、橢圓曲線、Jabber、ECDH、AES。

Abstract

The current instant messaging (IM) has more and more functions. It not only has the text chat, but also voice chat. Voice chat provides users with a more rapid means of communication. However, the voice chat of IM is not secure. Voice data is easily monitored and recorded over network.

In order to protect the voice chat of IM from monitor, we designed and implemented secure voice chat of IM based on Secure Instant Messaging and Presence Protocol (SIMPP). Firstly, we use winmm library to get digital voice data. Secondly, voice data is encrypted with a key generated by SIMPP. This study uses open source OpenSSL cryptographic library for security. The encrypted voice data is send to another user. When being received, the encrypted voice data will be decrypted to original voice data with the key. After been decrypting, the voice data will be transformed to voice by winmm library. If cracker got the voice packets from network, he will hear noise and does not know the content of the voice talk when playing the encrypted voice data. Thus, this study makes voice calls of IM secure.

Keywords: Instant Messaging (IM), Voice Security, Elliptic Curve, Jabber、ECDH、AES

一、緒論

英國VoIP專家Peter Cox, 在2007年發佈了名爲SIPtap的網路語音監聽軟體。依照他的測試結果，長達將近三個月的測試，都能分析出網路中的語音封包，並且存成檔案播放[13]。現今越來越多人使用即時通訊的語音功能，卻很少人留意透由網路傳送語音資料比傳統電話更容易被監聽。所以本研究實做一個安全即時通訊語音功能，讓使用者可以享受網路語音的便利也能具有保密性，而不會被監聽軟體所側錄。

由於網路速度越來越快，且價格越來越低廉，在網路上傳送聲音影像變得相當容易，所以許多主流的即時通訊軟體像是Yahoo! Messenger、America Online Instant Messenger、ICQ 、Google Talk和 MSN Messenger不只是豐富了原本的文字傳送，增加動感的表情符號，更追加了語音功能。讓使用者可以透由電腦網路就可以和另一方語音通話，進行比文字更快速更便利的溝通。

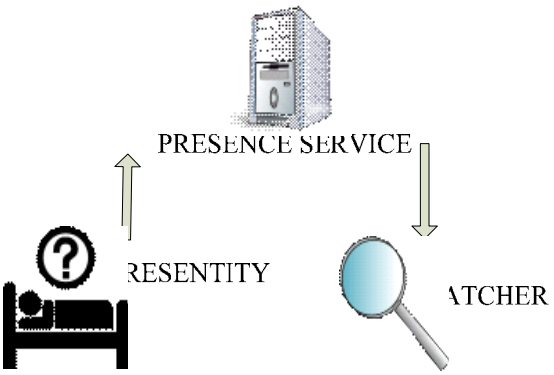
在各種溝通方式之中，語音是最常使用且快速的方式，傳統的遠距離語音只能採用電話，而這是會依使用時間以及距離的長短增加費用。而即時通訊軟體的語音功能只要通話雙方都具有電腦網路和喇叭麥克風，即可進行通話，由於只要支付網路費，因而改變很多人的通話方式，有許多企業組織都利用即時通訊軟體進行內部通話以節省通話費的支出。

二、文獻探討

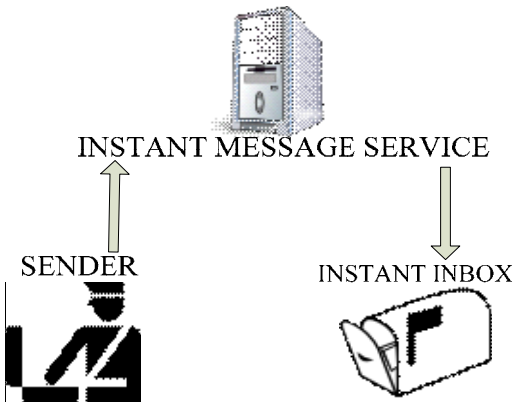
2.1 即時通訊

在2000年IETF爲即時通訊(Instant messaging)所供提的定義是能看見所選定的聯絡人名單的線上狀況，如果對方在線上，能進行即時的訊息交換。進一步來說就是即時通訊軟體具有同步和線上狀態。所以即時通訊軟體的功能具備即時的訊息交換，和觀察聯絡人名單的線上狀況[10]。

針對上點，IETF也定義RFC 2778標準[6][8]，即時通訊軟體需具有定位服務(Presence Service)和即時傳訊服務(Instant Messaging Service)。



圖一: RFC 2778所定義的定位服務(Presence Service)模型



圖二: RFC 2778所定義的即時傳訊服務(Instant Messaging Service)模型

從圖一可以看出，即時通訊軟體內有兩個服務來處理使用者狀態的機制，分別爲PRESENTITY和WATCHER。PRESENTITY將其使用者自己狀況傳送到PRESENCE SERVICE，而PRESENCE SERVICE會將其它使用者的狀態傳送給WATCHER。

可以從圖二看出，關於即時傳訊服務的部分，即時通訊軟體內有兩個服務來負責，分別爲SENDER和INSTANT INBOX。使用者要傳送的訊息透由SENDER傳送到INSTANT MESSAGE SERVICE，而INSTANT MESSAGE SERVICE再將訊息送到受訊方的INSTANT INBOX。而受訊方再從INSTANT INBOX取得訊息。

2.2 即時通訊協定

儘管即時通訊軟體相當便利，卻不一定能取代電子郵件，其主要原因是各家電子郵件提供者採用開放協定的SMTP和POP3。所以電子郵件的互通性是很好的。而即時通訊軟體部分，各家所採用的協定都不同，甚至有些是不公開的。所以形成即時通訊軟體的互通性比較匱乏。有鑑於此，IETF最早提出IMPP希望能制定出開放的即時通訊標準[10]。也爲IMPP (Instant Messaging and Presence Protocol)制定RFC2778、RFC2779、RFC3339、RFC3859、RFC3860、RFC3861、RFC3862、RFC3863。而現今主要的即時通訊開放協定爲SIMPLE和XMPP。

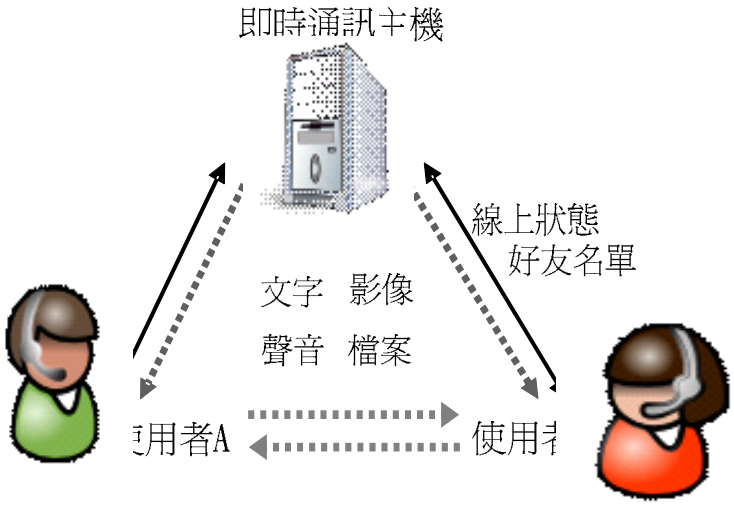
SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions)是利用SIP作爲基礎的即時通訊協定[21]。SIP (Session Initiation Protocol)爲會議啓始協定，是VoIP的通訊協定，故技術上已相當成熟。而SIMPLE就是利用SIP實作出符合IETF的即時通訊協定[10]。由於SIP本身就在發展多媒體服務，這使得SIMPLE有利於發展即時通訊語音功能。

另一爲XMPP (eXtensible Messaging and Presence Protocol)。最早爲開放原始碼通訊軟體Jabber的核心協定。以XML技術爲基礎，實做出符合IETF規範的即時通訊協定[6]。由於XML是具有很大的彈性，所以未來在技術的發展相當可觀。由於XMPP來自於Jabber，而Jabber是開放原始碼所以任何人都能自由取得並修改，所以在各平台都可以看到Jabber，因而很多開放性的即時通訊軟體都相容於XMPP。

由於SIMPLE是基於SIP所發展的，所以很多媒體的傳送直接依賴SIP就能進行，然而SIMPLE的各種訊息傳送是被包在SIP訊息裡面，所以一個簡單的SIMPLE訊息其實包

含了SIP訊息。而相對上，一樣的訊息，XMPP的封包就會比較小。不過對於多媒體以及點對點的傳送就需要重新打造。

而無論是SIMPLE或是XMPP協定，其即時通訊模型，都會有一台即時通訊主機，使用者透由即時通主機取得自己的好友名單並從自己的狀態回傳給主機，且收到好友們的狀態訊息，如下圖所示：



圖三: 即時通訊服務的通信三方模型

從圖三可以看出，使用者A和使用者B無論是採用SIMPLE或是XMPP都需要向即時通訊伺服器主機傳送用戶端的線上狀態和取得好友名單以及其線上狀態。如果是採用明文傳送，有心人士將能從中取得訊息，甚至發送竄改的文字訊息。所以SIMPLE和XMPP雖然是很好的即時通訊協定，但是缺乏好的金鑰管理機制。

2.3 即時通訊的語音功能

MSN Messenger是微軟公司所發佈的即時通訊軟體，經過不斷地改版之後又改名爲Windows Live Messenger，其通訊協定爲MSNP，目前的版本是MSNP 15。其主要功能有文字訊息、檔案傳送、電子白板、網路語音、視訊等。設計架構主要爲主從式架構，但是像檔案傳送、網路語音、視訊則是點對點運作。目前所提供的VoIP服務包含PC-PC, PC-Phone 和Phone-to-PC。而PC-Phone 和 Phone-to-PC是需要收費的。而其VoIP的功能是採用SIP協定所實現[6]。

圖四: Windows Live Messenger

Yahoo! Messenger是由Yahoo! 所推出的即時通訊軟體。具有文字訊息、檔案傳送、網路語音、視訊、電子白板等功能。其使用的通訊協定爲YMSG[6][20]主要是參考SIMPLE所做出來的協定。其中的語音功能又稱爲Yahoo! Voice包含PC-PC、PC-Phone 和Phone-to-PC共三種的VoIP服務。而語音通話是利用SIP協定，語音編碼是採用internet Low Bit Rate Codec (iLBC) [19]。

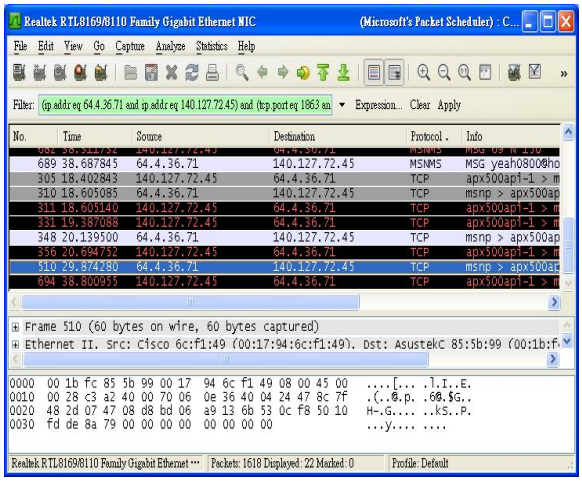
Google Talk是Google公司所推出的即時通訊軟體，其核心的通訊協定採用開放標準XMPP/Jabber，而多媒體傳送的部分則採用Jingle[5]協定。Jingle是XMPP延伸性的多媒體協定，可以用來傳送檔案，語音對話，視訊等功能[11]。Google依照Jingle協定實作出libjingle[4]函式庫，不只用來開發Google Talk，更將libjingle釋出原始碼。Google Talk除了基本的文字傳送之外，也具有檔案傳送，語音交談，視訊等功能。

無論是Windows Live Messenger或是Yahoo! Messenger本身都有穿越NAT的功能。但是語音資料的傳送方式是直接把壓縮過語音資料放在SIP封包中傳送，而SIP本身是明文傳送。雖然Google Talk不是採用SIP協定，但是其運作原理仍是取得聲音資料後，將聲音編碼，接著直接放入由libjingle所建立的點對點的session之中[4]。所以一樣是明文傳送。因此有心人士可以從中擷取封包，將封包重組後可以聽到其交談內容。

2.4 公眾即時通訊系統的安全性

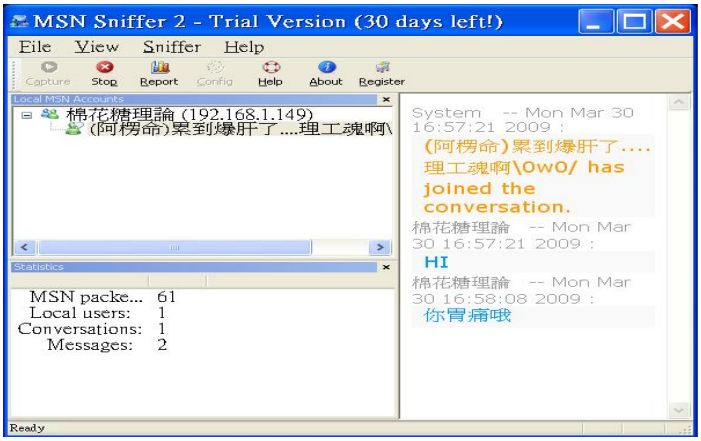
而目前多數的即時通訊軟體其安全性只針對帳號密碼進行保護，當用戶端登入即時通訊軟體主機時，有進行加密處理，等到身份驗證成功之後，接下來的訊息傳送都是明文型式。所以無法保證使用者A和使用者B以其主機之間通訊的保密性(Confidentiality)、完整性(Data Integrity)、可用性(Availability)、不可否認性(Non-repudiation)。

有心人士如果要取得即時通訊使用者之間的明文訊息，只需取得使用者之間和主機傳送的封包，就可以直接看到內容。像是使用Wireshark，一種開放原始碼跨平台的網路封包擷取分析軟體。可以識別多種網路協定，當然也包含即時通訊協定。不只如此，Wireshark可以針對現今主要的網路語音協定進行分析，像是SIP、H.225。



圖五: Wireshark監看MSN Messenger

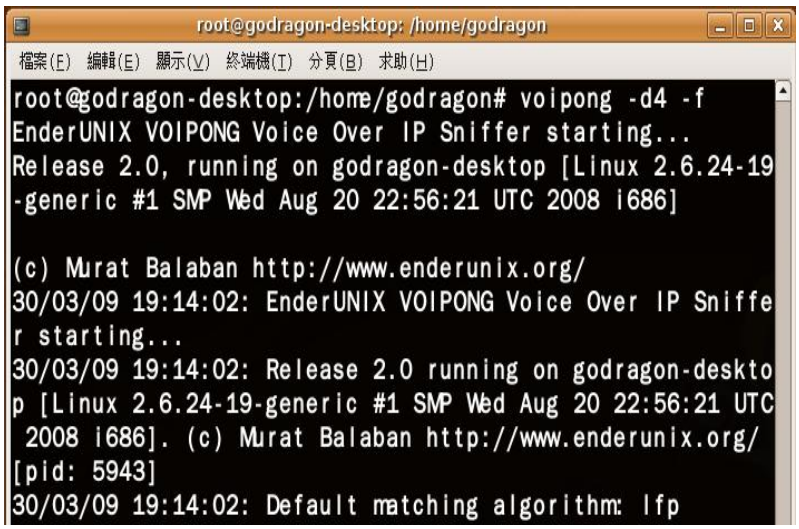
有些主流的即時通訊軟體，被成爲網路監聽軟體公司開發的對象。像是一間名爲EffeTech的網路安全監控公司，就針對MSN Messenger、America Online Instant Messenger、ICQ這些即時通訊軟體，研發出MSN Sniffer、AIM Sniffer、ICQ Sniffer這三套監聽軟體，更能方便的掌控其訊息的流向。



圖六: MSN Sniffer監看MSN Messenger

既然文字訊息可以被監看，那即時通訊的網路語音由於多數也採用明文傳送其語音資料，所以只要擷取完整的封包還原成原本的語音資料就可以即時地竊聽雙方的語音溝通內容。

在Linux平台上有一個名爲VoIPong[17]的開放原始碼的VoIP監聽工具，爲了效能而採用C語撰寫。能夠偵測到的VoIP種類包含H323、SIP、RTP、RTCP、Cisco's Skinny Client Protocol。可以將偵測到的VoIP語音錄成WAVE檔，存在本機中。而在效能上，官方網站宣稱網路流量在45 Mbit/sec的情況下，一台記憶體只有256MB，CPU只有Celeron 1700 MHz的Toshiba筆記型電腦都可以正確地偵測將區域網路中的VoIP，而其CPU的使用量只在66% - 80%之間。



圖七: VoIPong監聽網路中的VoIP

而在2007年英國的VoIP專家Peter Cox，發布了名為SIPtap的概念證明軟體(proof-of-concept program)，向大家證明VoIP是很容易被竊聽的。此舉引起很多人關注，Peter Cox希望透由這樣子的證明，帶給大家重視VoIP的安全問題。而其它開放原始碼像是Vomit以及OreKa，收費軟體Cain & Abel等等，都可以進行網路語音的監聽。

所以現今多數的即時通訊軟體其語音功能是很容易被竊聽的。如果只是普通的談話家常，那會失去個人的隱私性，如果在談話中涉及到重要機密，或是個人的身份證號碼，銀行提款密碼等等，將會產生嚴重的損失。所以即時通訊語音有安全性的需求。

2.5 即時通訊語音安全性

提升即時通訊安全的部分，關於文字訊息，目前已有許多做法。像是採用外掛軟體如IMSecure、SimpLite、Trillian，或是替代軟體如Gaim-Encryption，或是採用SSL/TLS，的方式來維護訊息的安全性[1]。

而維護即時通訊語音的部分，可以從四個面向著手。1是外掛程式。2是程式內建語音加密功能。3是採用SSL/TLS。4是採用安全的即時通訊交換協定。

2.5.1 外掛程式

使用外掛程式的好處是不用變動原本的程式。Zfone是由PGP的創始人Phil Zimmermann所研發的，支援多種常見的作業系統，其版權相當複雜，由於Zfone是由許多部分所構成的，而Phil將不同部分採用不同的版權宣告。但對普通使用者來說，仍是免費使用。Zfone的設計是把既存的VoIP 用戶端程式具有安全功能。只要先開啓Zfone再打開原本的VoIP 用戶端程式，Zfone就會自動建立安全的語音通話，且會把運作結果顯示在GUI上。

圖八: Zfone

2.5.2 程式內建加密功能

AJAH是商業的VoIP軟體，最近才開始具有128位元的語音保護。其主要特色是不只提供免費地PC對PC的語音通話，更提供免費每禮拜150分鐘PC對Phone的服務。

Skype一樣也是商業軟體，但仍提供免費的PC對PC的語音通話，文字傳送，檔案傳送等功能，對語音資料採用對稱式加密 AES 256演算法。Skype本身有自己封閉式通訊協定。

Twinkle是開放原始碼的VoIP軟體，採用GPL授權，運作平台爲Linux，無Windows版本。支援多種聲音編碼，採用ZRTP[22]和SRTP[15]達到語音安全性。

2.5.3 SSL/TLS

SSL (Secure Socket Layer)最早是由Netscape網景公司發展出來成爲web安全傳輸，後來被IETF標準化之後，在RFC3346中定義爲TLS（Transport Layer Security）。是以PKI爲基礎的安全傳輸層協定。由於PKI是建立在RSA基礎之上，故要運用到大量的模運算，而模運算的需要較多的計算[1]。

2.5.4 安全的即時通訊交換協定

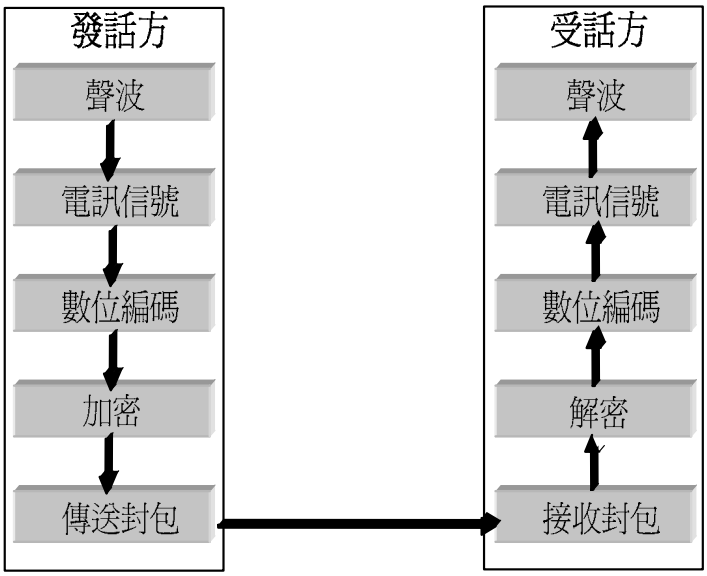
郭宗益(2007)提出的安全即時通訊與展現協定(Secure Instant Messaging & Presence Protocol, SIMPP)，改善Mannan與van Oorschot在2006年所設計的即時通訊金鑰交換IMKE協定。原本的IMKE協定是採用非對稱式RSA密碼系統爲基礎，改用採用質數體橢圓曲線密碼學爲基礎。不只增進計算的速度，而且在同樣的安全性之下，其金鑰的長度會小於RSA密碼系統。能確保主機和用戶端、用戶端和用戶端資料傳送的保密性和確認性。此外SIMPP的PAKE機制能抵擋攻擊者在使用者帳號註冊時的多種攻擊[1]。

三、即時通訊語音安全設計

文本研究是延續SIMPP的發展，所以雙方的訊息傳送都具有SIMPP的安全性[1]。具有訊息的保密性(Confidentiality)，完整性(Data Integrity)、確認性。但是不考量木馬和蠕蟲攻擊。

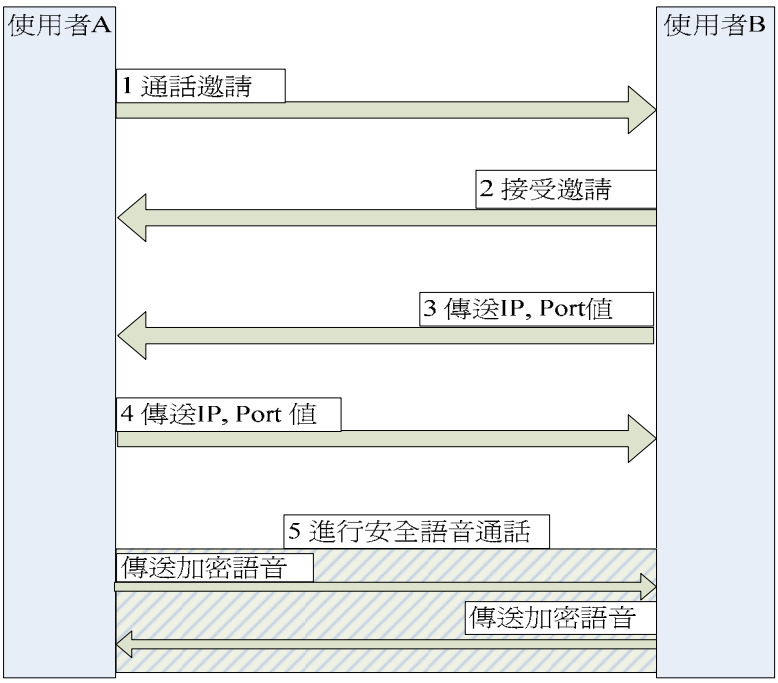
爲了達到安全的即時通訊語音功能，必須先透由音效裝置將聲波轉成電信訊號，再利用作業系統音效相關的API，把電信訊號轉成數位編碼，針對數位編碼進行加密，接著再把已加密的資料傳送給對方。對方收到資料後，需要先透由金鑰解密取得數位編

碼，再利用作業系統的API把數位編碼轉成電信訊號，最後由音效裝置將電信訊號轉成我們可以聽見的聲波。以下爲系統運作流程圖：



圖九: 系統運作流程圖

上圖中所使的加解密方法是對稱式密碼學演算法AES-128，ECB模式。因爲語音通話講究即時處理。密碼學演算法中可分爲對稱式和非對稱式，普遍來說對稱式演算法的速度比較快[2]，最主要的缺點是如何讓雙方同時擁有同一把對稱式金鑰。而這個問題，本研究透過SIMPP，可以安全地讓雙方都具有同一把對稱式金鑰，且不會被第三人所發現。



圖十: 語音邀請流程圖

當使用者A想要和使用者B進行語音通話時，使用者A會發出一個通話請求。使用者B收到請求之後，可以選擇拒絕通話或是接受通話。當使用者B接受通話將會將自己的IP位置和使用的語音Port傳送給使用者A，使用者A收到之後，先傳送IP和Port給使用者B，接著就將自己的語音資料加密後傳送給使用者B。使用者B收到IP相關資訊後，就接收來自使用者A的加密語音，使用共同金鑰解密。同時也將自己的語音資料加密送給使用者A，此時雙方開始進行安全的語音通話。

四、即時通訊語音安全架構與實現

本研究以SIMPP (Secure Instant Messaging & Presence Protocol)爲基礎，架構主從式的即時通訊系統，當使用者之間要進行語音通話時，先透由原本的SIMPP架構進行語音邀請以及傳送點對點所需資料，最後進行點對點的安全語音通話。

本研究的安全語音功能實現在SIMPP的用戶端程式，使得原本具有安全性的文字通訊功能更增加了安全語音功能。程式開發工具採用Borland公司出版的Borland C++ Builder，作業平台爲Windows XP。語音的擷取和播放採用Windows平台上內建的Winmm[18]多媒體函式庫，Winmm是Windows平台上用來處理多媒體的一組WINAPI，從Windows NT 3.1之後的作業平台，都是透由Winmm多媒體函式庫控制音效裝置。安全的部分引用開放原始碼OpenSSL密碼學函式庫，使用對稱加密演算法AES128，ECB進行加密。將語音資料進行加密後才發送給受話方，受話方收到語音資料必須先利用由SIMPP產生的對稱式金鑰進行解密，然後就可以把語音資料轉換成聲波。雙方即可進行安全的語音交談。

表一: 即時通訊安全語音用戶端程式規格表

程式開發工具	Borland C++ Builder
聲音函式庫	Winmm
密碼學函式庫	OpenSSL
語音加解密	AES128，ECB
對稱式金鑰來源	SIMPP

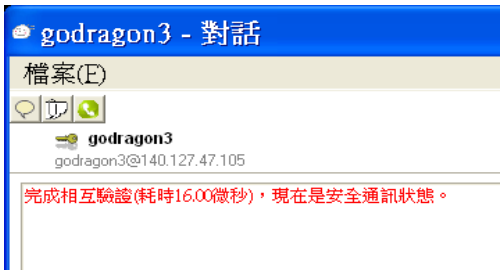
安全即時通訊伺服器程式採用SIMPP的伺服器程式。作業平台採用Ubuntu 8.04版。SIMPP的伺服器程式增進原本的jabberd-2的安全功能，改進jabberd-2只能以一般使用者的身份運作，以避免當jabberd-2被入侵時，立刻取得管理者權限。而資料庫管理系統採用開放原始碼MySQL，存放使用者基本資料和驗證資訊。所使用的密碼學規格如下表：

表二: SIMPP密碼學規格表

公開金鑰 密碼系統	$GF(p)$橢圓曲線 $(y^2 = x^3 - 3x + b \bmod p)$，
金鑰長度	Server 224位元， Client 192位元
金鑰交換方法	ECDH (Elliptic Curve Diffie-Hellman)
數位簽章演算法	ECDSA (Elliptic Curve Digital Signature Algorithm)

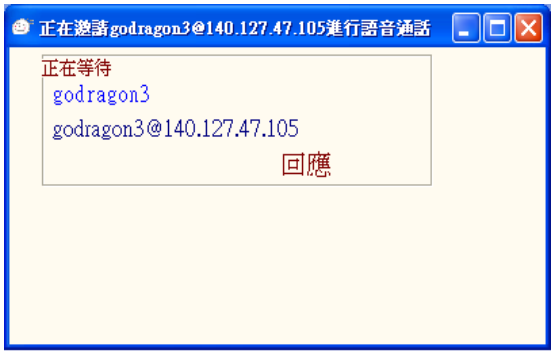
訊息加解密	128 Bits AES，CBC模式
單向雜湊函數	SHA-256

如果要進行安全語音功能，要先登入SIMPP的主機，點擊好友名單之後，會出現對話視窗。視窗上排有一個話筒的圖示，點擊之後就會邀請對方，等待對方回應後，就可以進行安全的語音通話。

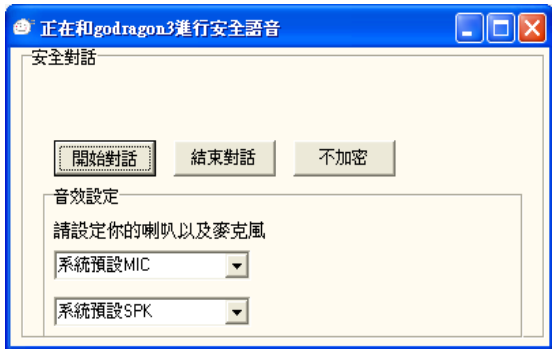


圖十一: 安全語音按鍵

上圖中，可以看出雙方已經透由SIMPP完成相互認證，雙方已具有短期的共同金鑰。所以雙方可以進行安全的語音對話。



圖十二: 等待回應



圖十三: 進行安全語音

當對方答應請求後，就可以進行安全語音通話，預設是啓用雙方的短期共同金鑰利用AES 128對語音資料做加解密。

五、結論

即時通訊軟體由於使用上相當方便，能即時地彼此互動，加上語音功能更促進溝通效率。然而現今絕大多數的即時通訊軟體所提供的語音功能都是不安全的。而本研究以安全即時通訊與展現協定(SIMPP)爲基礎，將語音資料加密後才傳送到對方，實際開發出安全即時通訊語音功能。即使有心人士擷取到語音封包，也只能聽到噪音，無從得知雙方談話內容，以確保談話的安全性。

參考文獻

[1] 郭宗益，”安全即時通訊系統設計與實現”，*國立高雄師範大學資訊教育研究所碩士論文*，2007
[2] 楊中皇，網路安全：理論與實務，台北： 金禾資訊，2006。

[3] Baumann, R., Cavin, S., and Schmid, S., “Voice Over IP - Security and SPIT,” *KryptDet Report*, 2006 (available online at http://www.schmiste.ch/wk06.pdf)
[4] Google Talk, http://code.google.com/intl/zh-TW/apis/talk/libjingle/libjingle_applications.html
[5] Jingle (XEP-0166), http://xmpp.org/extensions/xep-0166.html
[6] Khoshbakhtian, M., Darvishan, A. H., and Eghtedari, P., “Comparative Analysis of IMP services,” *3rd International Conference on Information and Communication Technologies: From Theory to Applications*, 2008: pp. 1-6
[7] Kikuchi, H., Tada, M., and Nakanishi, S., “Secure instant messaging protocol preserving confidentiality against administrator,” *18th International Conference on Advanced Information Networking and Applications (2)*, 2004: pp. 27- 30
[8] RFC 2778, http://www.ietf.org/rfc/rfc2778.txt
[9] Mannan, M., and Oorschot, P. C. van., “A protocol for secure public Instant Messaging (extended version),” *Financial Cryptography and Data Security 2006 (FC'06)*, 2006, (available online at http://www.scs.carleton.ca/~paulv/papers/FC06-preproc.pdf)
[10] McClea, M., Yena, D. C., and Huangb, A., “An analytical study towards the development of a standardized IM application,” *Computer Standards & Interfaces* (26:4), August 2004, pp. 343-355.
[11] Saint-André, P., “Jingle: Jabber Does Multimedia,” *IEEE Multimedia* (14,1), 2007: pp. 90-94
[12] SIP, http://www.ietf.org/html.charters/sip-charte.html
[13] SIPtap, http://www.techworld.com/security/features/index.cfm?featureid=3859
[14] SIPtap, news http://www.voip-news.co.uk/2007/11/23/siptap-software-can-eavesdrop-on-voip-calls/
[15] SRTP, http://en.wikipedia.org/wiki/SecureReal-time_Transport_Protocol
[16] Sun, L. F., and Ifeachor, E. C., “Voice Quality Prediction Models and their Application in VoIP Networks,” *IEEE Transactions on Multimedia* (8:4), 2006: pp. 809-820.
[17] VoIPong, http://www.enderunix.org/voipong/
[18] Winmm, http://msdn.microsoft.com/en-us/library/ms712636(VS.85).aspx
[19] Yahoo! Voice, http://en.wikipedia.org/wiki/Yahoo!_Voice
[20] YMSG, http://en.wikipedia.org/wiki/YMSG
[21] Zhang, Y. T., Liao, J. X., Zhu, X. M., Wu, W., and Ma, J., “Inter-working between SIMPLE and IMPS,” *Computer Standards & Interfaces*,(29, 5),2007: pp. 584-600
[22] ZRTP, http://en.wikipedia.org/wiki/ZRTP