Design and Implementation of a Secure Video of Instant Messaging Based on SIMPP

Yu-Ting Lo
National Kaohsiung Normal University
creeds2239@gmail.com

Chung-Huang Yang
National Kaohsiung Normal University
chyang@computer.org

***Abstract* — Instant Messaging is not only the most popular type of peer-to-peer computing network service currently available, but it is also a very important way of contacting people in daily life. However, most instant messaging programs do not provide security mechanism capable of ensuring Client to Client video service communications [2][13]. Therefore, this study designs and implements a secure video service based on the SIMPP (Secure Instant Messaging and Presence Protocol)[8]. The proposed service is compatible with the XMPP (eXtensible Messaging and Presence Protocol) Standard. If attackers intercept video packets from the network, they cannot load the encrypted video stream packet. This design not only achieves security in instant messaging services, but also avoids the rate of video transmission delay without affecting the user's video quality.**

***Index Terms* — Instant Messaging、SIMPP、XMPP、Jabber、Secure Video**.

I. INTRODUCTION

With the growth of computer networks in recent years, as well as the rapid growth of multimedia applications, instant messaging services have become an E-mail, Web of the third-largest network after the application services [14]. Nevertheless, most IM programs, such as WLM (Windows Live Messenger) [11] and Yahoo! Messenger [12] do not have any encrypt mechanism. If a user wants to protect the communications of IM programs, the most common solution is to use plug-in software like SimpLite-MSN, MSN Plus!, which provides a message encryption function to ensure message security. However, the existing plug-in software does not provide video and sound protection mechanisms [2][13].

To address this problem, the study designs and implements a secure video service for an Instant Messaging system based on SIMPP. The proposed use of cryptography TurboPower [12][16] library provided algorithms AES encryption for video streaming packets to ensure the availability, confidentiality, integrity of peer-to-peer transmission of video packets solves instant messaging multimedia transmission security problems [8].

II. LITERATURE REVIEW

The Internet Engineering Task Force (IETF) established an Instant Messaging and Presence Protocol Working Group (IMPP WG) and Instant Messaging and Presence Protocol (IMPP) [10] and proposed that the RFC 2778 defined the Instant Messaging system to be composed of two types of services, which are the Presence Service and the Messaging Service [3] :

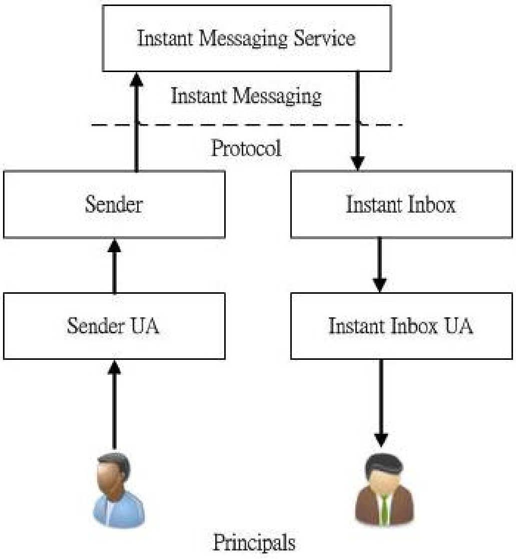


Figure 1. Instant Messaging Service models defined in RFC 2778

A. Messaging Service

Instant messaging software first sends a message to the server. The Instant Message Service accepts and delivers Instant Messages to an Instant Inbox. The client's PRINCIPAL sets up an Instant Inbox for these instant messages. When a client receives a message from the Inbox, it immediately displays it to the user to user immediately. This works has characteristics similar to E-mail Client / Server architecture.

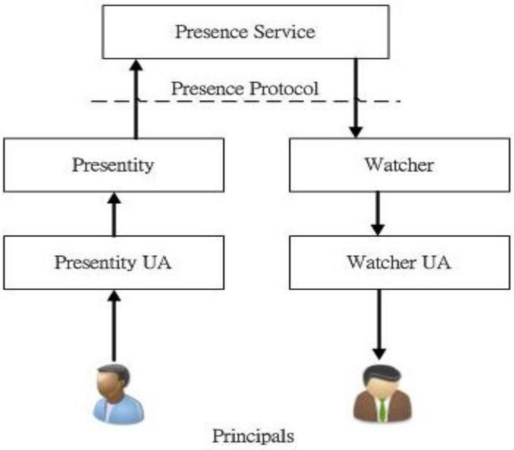


Figure 2. Presence Service models defined in RFC 2778

B. Presence Service

Presence Service is composed of PRESENTITY and WATCHER. PRESENTITY dispatches status of user for the presence message to WATCHER. The other user may understand the PRESENTITY status through the Presence Service [3].

C. Instant Messaging Standards

The IMPP has become the current SIMPLE and XMPP instant messaging protocol reference architecture.

IETF proposed RFC3428 and defined SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) standards in 2002. SIMPLE is based on SIP (Session Initiation Protocol), which is a real-time protocol. This is because SIP is used in consultation, dialogue management, and termination of media sessions and transmits data from a specific protocol to be completed such as RTP (Real-time Transport Protocol). The first multi-media channels to use SIP to establish a session, followed by the use of SDP (Session Description Protocol) to complete the functions of both ends of the exchange of media session. Finally, because RTP can be used to deliver multi-media (video, voice) packet, so SIMPLE is suitable for development instant messaging multimedia services [19].

The Jabber community created the first Extensible Messaging and Presence Protocol (XMPP) in 1999 [9][13]. After a period of correction and development, IETF proposed RFC3920-RFC3923, which redefined the instant messaging standard in 2004. The most important jabber's message architecture is characterized by the use of XML (eXtensible Markup Language) for the data format description language [9][13].

In addition, the Jabber network topology structure is similar to an e-mail system, in which each client needs a local server to receive and send messages. The client and the IM server communicate using TCP over Port 5222 to establish a connection, and servers connect to each other via TCP over Port 5269[13].

D. The threat of instant messaging

Although instant messaging provides a variety of services, popular IM software is not secure [13]. For instance, in Windows Live Messenger, any user that has successfully logged into the system can communicate in plaintext with other users [13][15], and communications are not properly protected.

This means that an attacker could easily capture the video service traffic content. As a result it is not possible to guarantee the confidentiality, integrity, availability, or non-repudiation of communications between the host and users [15].

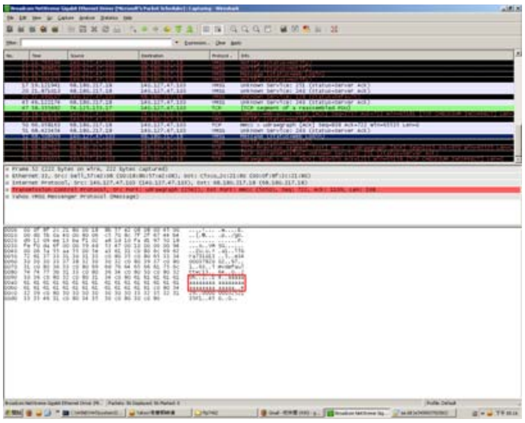


Figure 3. Sniffer Pro Package Interception

Many network security companies have developed sniffer software for instant messaging such as Figure 3,other including MSN Sniffer's IMDetect and AIM Sniffer [7].

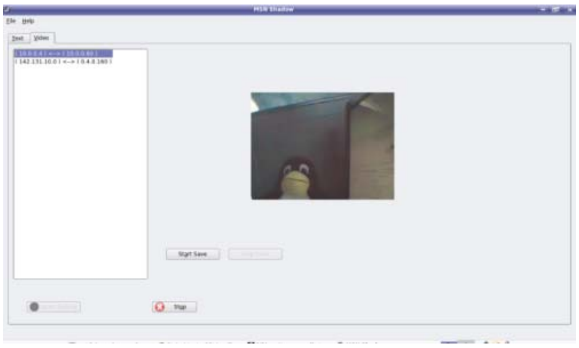


Figure 4. MSN Shadow Video Package Interception

Figure 4 shows that there are also many open-source programs. For example, MSN Shadow uses libmimic API and MPlayer for the binary code for video encoding and video decoding method, respectively and uses the AVI video format for video streaming [5][6]. Enterprises may be able to apply to protect the safety of commercial confidentiality, but an attacker can also use sniffer software to intercept packets.

E. Security for video service of Instant Messaging

Three strategies can be used to enhance the security of Instant Messaging [14]:

- Plug-in

First of all, encrypt a message to the majority of current through the plug-in software such as SimpLite, IMsecure, Pidgin-Encrypt text messages and so on to achieve the purpose of confidentiality. However, there are very few video encryption plug-ins for instant messaging programs. Therefore, this type of message encryption software cannot provide confidential and secure transmission of video content.

- SSL/TLS

SSL/TLS are based on PKI (Public Key Infrastructure) [11], perhaps use of SSL / TLS to establish secure communications Client-Server (for example, Google Talk)[4], although this solution maybe will ensure that Client-Server communications security, but when two client will using video or file transfer directly between clients without passing through a server is referred to as peer-to-peer communication, do not adopt SSL/TLS[8] [17].

- Built-in encryption mechanism of instant messaging software

In addition to the current instant messaging software, Skype has built-in AES 256 encryption algorithms, text, voice, and video to provide the function of encryption [1]. Other commercial types of instant messaging software include Live Communications Server for high security services, but for other mainstream instant messaging programs, there is no built-in encryption functionality.

III. Design And Of Secure Video Of Instant Messaging

This purpose of this study is based on SIMPP, so there is no viruses in the operating system environment, Instant Messaging server after the login, the two sides of the transmission of any message have a security, Confidentiality, Data Integrity, Authentication [18].

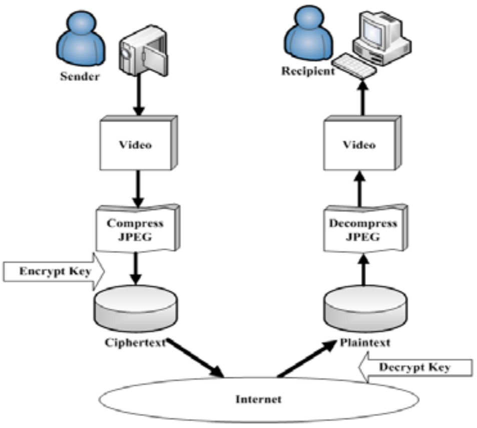


Figure 5. System architecture diagram

Figure 5 depicts the proposed system architecture. To complete the connection of communication, this design involves the following steps:

Step1: Capture the user's image through a webcam device.

Step2: Compress the image data to jpeg and save it in a video stream.

Step3: Encrypt the video stream with a shared key and send the encrypted video stream packet to the receiver.

Step4: The receiver decrypted the video stream packet must be encrypted with correct shared key to get original video stream.

Step5: Decompress the video streams through JPEG and display the images on the screen.

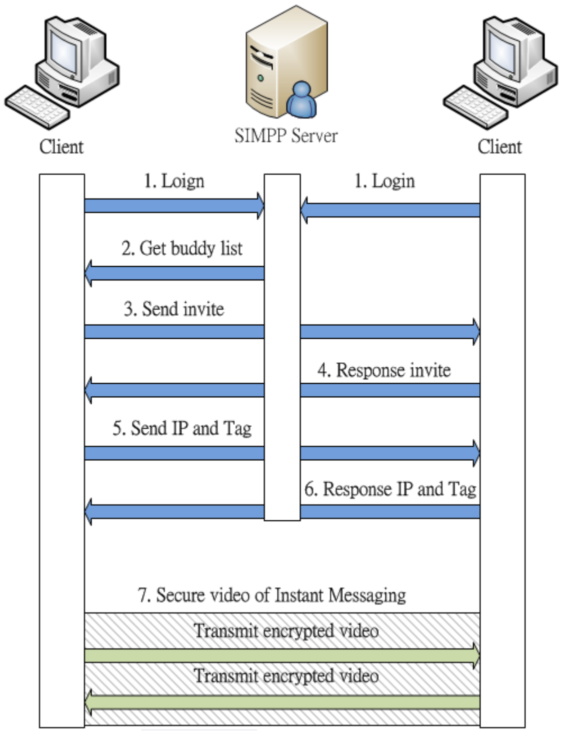


Figure 6. Video service flow chart

If client A and client B want to transmit video, client A sends a request to client B. When client B receives this request and accepts it, client B sends the IP information and Device Tag of video transmission services to client A. Client A sends the IP information and Device Tag to client B, and then client A's encrypted video stream is transmitted to client B. Client B receives the IP information from client A and encrypts video stream using the same symmetric key. At the same time two sides begin to security video of instant messaging service.

IV. Implementation Of Secure Video Of Instant Messaging

This study implements a secure video service of Instant Messaging based on the SIMPP (Secure Instant Messaging & Presence Protocol). When users want to enable video services, the client software sends a video service invitation and information about peer to peer data transmission through from the SIMPP architecture.

The proposed tax uses program development tools such as Borland C++ Builder, and uses the Microsoft Video for Windows library to capture webcam video. Secure part of video stream, using TurboPower company release open source LockBox cryptographic library to encrypt with symmetric encryption algorithm AES 128, CBC mode [12][18] to encrypt and jpeg compresses to save in video stream before sending it to the receiver.

When user receives the video stream, it must be decrypted by the generated SIMPP symmetric key. This enables both users to conduct secure video transmission in an Instant Messaging program. Table 1 shows the specifications of the proposed secure video service in an IM system.

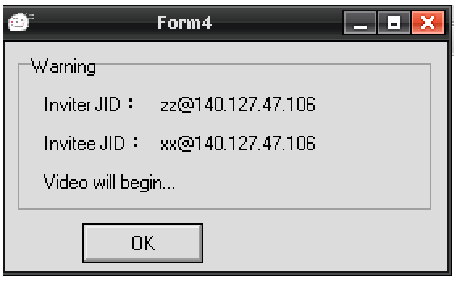
Table 1. Secure video of Instant Messaging specifications

Program Development Tools	Borland C++ Builder
Multimedia Library	Microsoft Video For Window
Cryptographic Library	TurboPower LockBox
Symmetric Cryptosystem	AES 128，CBC
Key Exchange Agreement	SIMPP

The secure Instant Messaging is the use of SIMPP server; it revised from open source jabberd and operated on Ubuntu 8.10. Using MySQL Database stores user information and authentication information. SIMPP server is more secure than original jabberd. SIMPP cryptography used in the specifications in the following table:

Table 2. SIMPP cryptographic specifications

Public-Key Cryptosystem	GF(p)Elliptic Curve ($y^2=x^3-3x+b \mod p$)
Size of Key	Server 224 位元 Client 192 位元
Key Exchange Agreement	ECDH
Digital Signature Algorithm	ECDSA
Symmetric Cryptosystem	128 Bit AES，CBC
One-way Hash Function	SHA-256



Warning

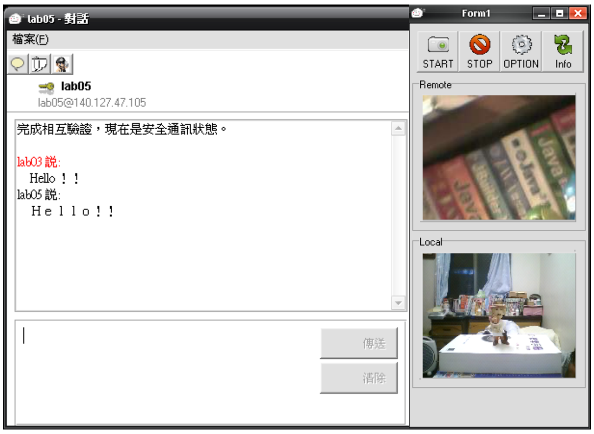
Inviter JID： zz@140.127.47.106

Invitee JID： xx@140.127.47.106

Video will begin...

Figure 7. Wait for response

By clicking on a particular and video service icon button or the buddy list, the receiver can display an invitation dialogue in the window. Figure 7 shows how the invitation will be displayed.



lab05
lab05@140.127.47.105

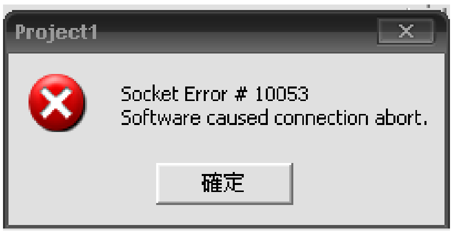
完成相互驗證，現在是安全通訊狀態。

lab03 說:
Hello！！
lab05 說:
Hello！！

Figure 8. Secure video of Instant Messaging

If a user attempts to send a video invitation while off-line, the invitation service will be system canceled. At the beginning of the video service, the user will record its own webcam device information and a verify tag of random, which will be immediately encrypted and saved as a tag. This will also be sent to the recipient through the SIMPP server, when the beginning of transmission peer-to-peer video streaming will be sent encrypted with the tag and video packets.

The recipient will decrypt the tag, and if the tag is valid, will begin decryption video packet. If the tag authentication fails, this indicates that the carrier is not the inviter, and system will not decrypt the video stream, but terminate video services.



Socket Error # 10053
Software caused connection abort.

Figure 9. Attempt to read the encrypted video streaming of the window

Even if attackers intercept packets, they cannot obtain the correct decryption key, and will not be able to restore the video stream packets. Figure 9 shows the resulting error message.

V. CONCLUSIONS

Instant Messaging programs are very convenient and important applications in our lives, but the major instant messaging programs that currently provide video service are not secure. For example, programs such as Windows Live Messenger generally contain no provision for message confidentiality. Therefore, this study designs and implements a secure video service for Instant Messaging based on SIMPP. This system uses the webcam device name as an authentication tag and encrypts video streaming and then transmits it through the network. Even if attackers intercept the video packets, they cannot return to the original video stream format to prevent attackers from spoofing video stream transmission, effectively achieving video service confidentiality and security.

REFERENCE

[1] T. Berson, A. Laboratories, “Skype Security Evaluation,” 2005.

[2] C. E. Dalton and W. Kannengeisser, “Instant Headache,” Hong Kong CERT/CC Security Bulletin, 2003, pp. 2-7.

[3] M. Day, Lotus , J. Rosenberg, dynamicsoft and H. Sugano Fujitsu, “A Model for Presence and Instant Messaging ,” RFC2778, 2000.

[4] Google , “SSL Connections,” Admin Terms of Service. http://www.google.com/support/a/bin/answer.py?hl=en&answer=100181.

[5] G.M. Nunes, " MSN Shadow AN INSTANT MESSAGING FORENSICS TOOL (MSN FORENSICS) ,"

http://msnshadow.blogspot.com/

[6] Gabriel, MSN Shadow, SourceForge.net, http://sourceforge.net/projects/msnshadow, 2008.

[7] IMDetect , MSN Sniffer., http://www.msnsniffer.com/, 2008

[8] T.Y. Kuo, “Design and Implementation of Se-cure Instant Messaging System,” MA Dissertation. Graduate Insitute of Information & Computer Education, National Kaohsiung Normal University, Kaohsiung, 2007

[9] S. Ludwig, J. Beda, P. Saint Andre, R. McQueen, S. Egan, and J. Hildebrand, “Jingle (XEP-0166),“ XMPP Standards Foundation, 2009, http://xmpp.org/extensions/xep-0166.html.

[10] M. McClea, D. C.Yena and A. Huangb, "An analytical study towards the development of a standardized IM application," Computer Standards & Interfaces, vol. 26, no.4, pp.343-355, 2004

[11] Mozilla.org, SSL/TLS, Mozilla Developers, 2006, http://www.mozilla.org/projects/security/pki/nss/ssl.

[12] Norden Logic Oy at Helsinki, “TurboPower VCL for CodeGear RAD 2007 C++ (and 2009) ,” http://www.nordenlogic.com.

[13] J. Rittinghouse and J. F. Ransome, JUN-2005, "IM Instant Messaging Security, " Elsevier Digital Press.

[14] TWCERT, “Instant Messaging of operating recommendation and security solutions,” Taiwan Computer Emergency Response Team / Coordination Center, 2009., http://www.cert.org.tw/document/column/show.php?key=95.

[15] The Wireshark team, Wireshark , 2009, http://www.wireshark.org/.

[16] TurboPower Software Company, LockBox, SourceForge.net, http://sourceforge.net/projects/tplockbox/.

[17] N. Williams, J. Ly, "Securing Public Instant Messaging (IM) At Work," Centre for Advanced Internet Architectures, Technical Report 040726A, Swinburne University of Technology.

[18] C.H. Yang, "Network Security: Theory and Practice," Basic Security Services Inc., Taipei, Taiwan, 2006(in Chinese).

[19] Y. C. ZHANG , "Standardized Instant Messaging Protocols : Comparative Analysis of SIMPLE and XMPP, " J. of Wuhan Uni. of Sci. & Tech. (Natural Science Edition) ,Vol.28 , No.4 , Dec, 2005.