Empirical Analysis of Security Protocol in Mobile VoIP System

Conference Paper · December 2010

CITATION
1

READS
1,323

6 authors, including:

Han Park
Yonsei University
4 PUBLICATIONS **9** CITATIONS
SEE PROFILE

Jaejong Baek
Yonsei University
14 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Empirical Analysis of Security Protocol in Mobile VoIP System

Han Park, Che Xuemei, Gyungho Chu,
Jaejong Baek and Jooseok Song
Department of Computer Science
Yonsei University, Seoul, Korea
jssong@emerald.yonsei.ac.kr

Yonjeong Kang
Korea Internet & Security Agency, Seoul, Korea
yjkang@kisa.or.kr

***Abstract*—With the rapid growing of Voice over Internet Protocol services, security issues of VoIP become more and more important. Also, popularization of smartphone, such as Android and iPhone, makes it necessary to provide security to mobile VoIP. However, because of the limitations in mobile environment, providing security could bring more overhead than in wired environment. Hence, it is necessary to develop proper security solution to mobile VoIP. In order to analyze whether standard security protocol is suitable to be used in mobile environment, in this paper, we set up a test-bed and implement security protocol using Asterisk as the server and Sipdroid as the client. The necessity of developing a proper mVoIP security protocol is also suggested.**

Keywords-mobile VoIP; security; protocol

I. INTRODUCTION

Due to the various benefits, such as low cost and convenience, a growing number of public institutions and corporations are introducing Voice over IP (VoIP) services. Also, with the spread of smartphone all over the world, mobile VoIP becomes a very attractive market.

However, with the development of VoIP technology and market, the threats related to the vulnerabilities of VoIP are growing as well. Hence, it is necessary to apply the VoIP security approaches. Achieving end-to-end security in a VoIP session is a challenging task. VoIP session establishment involves various protocols, all of which must inter-operate correctly and securely. Unfortunately, since the capacity of mobile devices is limited, when the existing VoIP security protocols are applied to mobile VoIP, the performance becomes so bad that it is difficult to provide VoIP service. To solve this problem, it is necessary to provide a lightweight VoIP security mechanism in mobile networks.

The main motivation of this paper is to implement VoIP protocols and the related security protocol in mobile platform using Java language and Android. Then, we compare the performance before and after security protocol is implemented.

The remainder of this paper is organized as follows: in Section 2, we briefly introduce the VoIP protocol stack including typical protocols in each layer. VoIP signaling and media transport security protocols are also explained in this section. The implementation environment, evaluation and result analysis are shown in Section 3. In Section 4, we make some conclusion of our work and explain our future work.

II. BACKGROUND

A. VoIP and Mobile VoIP

Fig. 1 shows the widely used VoIP protocol stack, which is divided into three layers: signaling, session description, and media transport.

Signaling is an application-layer control mechanism used for creating, modifying and terminating VoIP sessions. The dominant signaling protocol is Session Initiation Protocol (SIP) [1]. Session description protocol (SDP) [2] is used for initiating multimedia and other sessions, and often include key exchange as a sub-protocol. Media transport layer is the layer in which the actual voice datagram is transmitted. In this layer, Real-time Transport Protocol (RTP) [3] is used. Key exchange protocols are intended to establish a cryptographically secure session. Since secure media transport protocol, like Secure Real-time Transport Protocol (SRTP), doesn't deal with key exchange part, security of the media transport layer depends on the security of the key exchange mechanism.

Mobile VoIP (mVoIP) is an extension of mobility to a VoIP network. In this case, mobile device is implemented as a SIP and RTP client. Turning a mobile handset into a SIP client requires that the mobile handset support high speed IP communications, while in standard VoIP, this is provided by any broadband IP-capable wireless network [4]. Additionally, decent computing power, processing ability, and battery life should be provided, especially in the case of applying security protocols. In wired case, this may be not a big problem, while in mobile device case, this could result in unacceptable performance because of the limited resources.

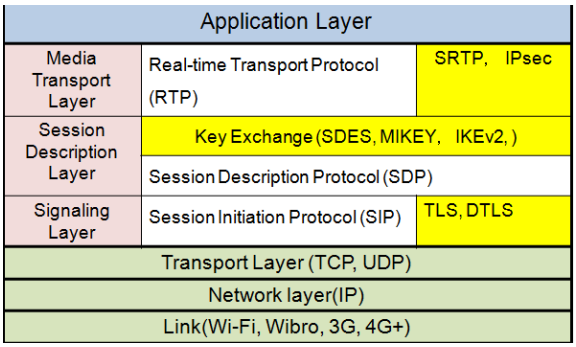

Figure 1. VoIP protocol stack

978-1-4244-8870-4 /10/$26.00 © 2010 IEEE

B. VoIP Security Protocols

Fig. 1 also shows security protocols, including signaling security, e.g. Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS), and Internet Protocol Security (IPsec), media security, e.g. Secure Real-time Transport Protocol (SRTP), and IPsec, and some key management protocols, e.g. Session Description Security (SDES), Multimedia Internet Keying (MIKEY), and Internet Key Exchange (IKEv2). This section describes signaling and media security protocols.

1) Signalling security protocol: TLS is the most widely used protocol for securing VoIP signalling. The primary advantage of TLS is that it is easy to provide security for an application protocol by inserting TLS between the application layer and the network layer [5]. Also, TLS offers hop-by-hop security. However, TLS is based on a reliable transport channel, like TCP, therefore cannot be used on top of UDP, which is widely used in VoIP.

DTLS is a modified version of TLS that runs on top of UDP. One of the advantages of DTLS is that since it is very similar to TLS, there is not compatibility problem with existing infrastructure. Another advantage is that it is easy to adapt protocols to use it, because it provides a familiar interface.

2) Media transport security protocol: Secure media transport aims to provide confidentiality, message authentication and integrity, and replay protection to media stream. An example of a secure media transport protocol used on VoIP communications is SRTP [6]. SRTP defines a set of default cryptographic transforms, and it utilizes Advanced Encryption Standard (AES) as the default cipher for encryption and decryption of the data flow. Though SRTP can easily accommodate new encryption algorithms, the SRTP standard states that new encryption algorithms besides those described cannot simply be added in some implementation of SRTP protocol [7].

SRTP creates encryption key from master key exchanged during call setup. However, SRTP does not deal with key management, which means key exchange protocols are needed.

III. IMPLEMENTATION AND EVALUATION OF VOIP SECURITY PROTOCOL IN MOBILE PLATFORM

A. Test-bed Setup

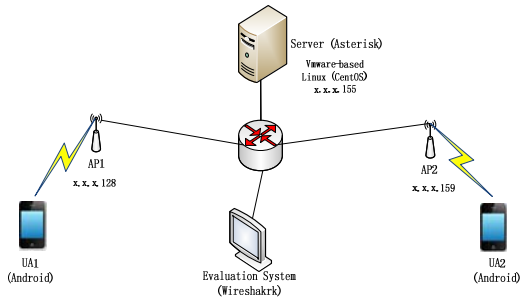

Figure 2. Test-bed environment

Fig. 2 shows the overall test-bed environment. To implement VoIP security protocol in mobile platform we use the open source Asterisk to set up the server and sipdroid to implement the user agent. To test the performance, we implemented TLS in both the server and client. Wireshark is used as a tool to evaluate the system.

1) Server: Asterisk is free and open source software that turns an ordinary computer into a voice communications server. Asterisk powers IP PBX systems, VoIP gateways, conference servers.[8] AsteriskNOW is the Asterisk software appliance, which is a combination of CentOS Linux, Asterisk and FreePBX. FreePBX is an easy to use GUI (graphical user interface) that controls and manages Asterisk.

In our system, we install Asterisk Now 1.6.7 on Vmware with 512MB memory and 10GB hard disk. After installing, we add two extensions (100 and 101) on the server. To make TLS work, we configure Asterisk in the following steps[9]:

- Creating our Certificate Authority.
- Creating our Server Certificate.
- Preparing the certificate for Asterisk.
- Configuring Asterisk.

2) Softphone: Sipdroid is a VoIP application for the Google Android operating system using the SIP. It is open source free software released under the GNU General Public License. We analyse the sipdroid source and implement TLS. Fig. 3 shows the result of the emulator. Besides UDP and TCP, TLS can be selected as the transport protocol and the corresponding port number (5061) can also be selected. Then, the communication between two Android phones can be done on top of TLS. The result and analysis will be shown in Section 4.

3) Wireshark: Wireshark is a free and open source packet analyser. It is used for network analysis, software and communications protocol development. It is able to display the encapsulation and show the message flow of

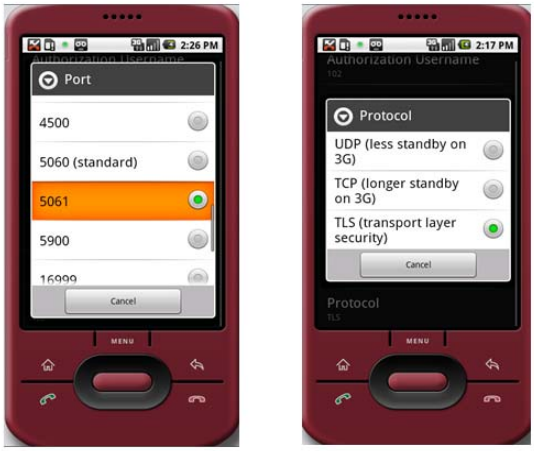

Figure 3. Android Emulator implementing TLS

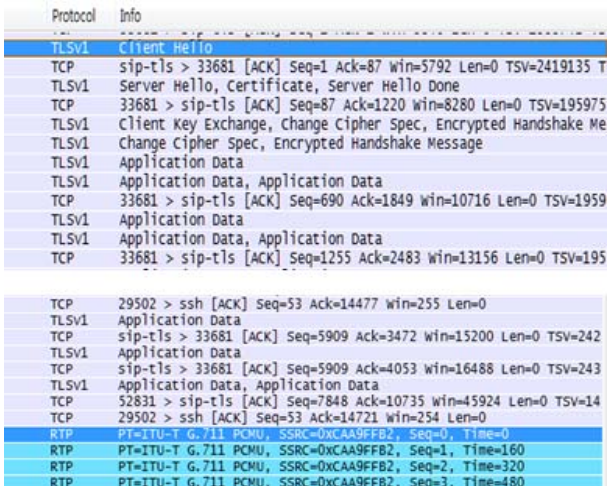

Figure 4. Wireshark captured TLS packets

VoIP calls. In this paper, we use wireshark to capture VoIP packet between Sipdroid and Asterisk server and then analyse the captured packets to get call set up delay, loss rate, etc. Fig. 4 is the result of packet capturing between two Android phones using TLS as the transport layer protocol.

B. Evaluation

As is shown in Fig. 5, before setting up any call, the client has to register on the server. When TLS is used, there is additional TLS handshake time before SIP registration. After SIP registration, the call is set up.

TLS client and server use a handshaking procedure to negotiate a secure connection. During this handshake, the client and server agree on various parameters [10]. The messages needed to exchange during handshake are shown in Fig. 6.

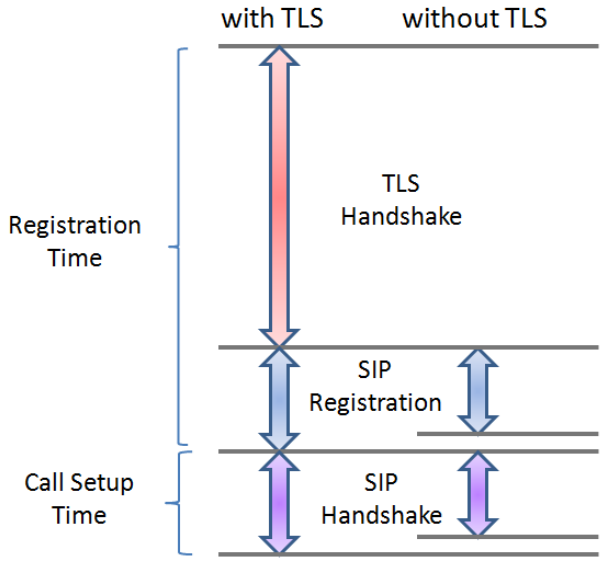

Figure 5. VoIP registration and call setup time

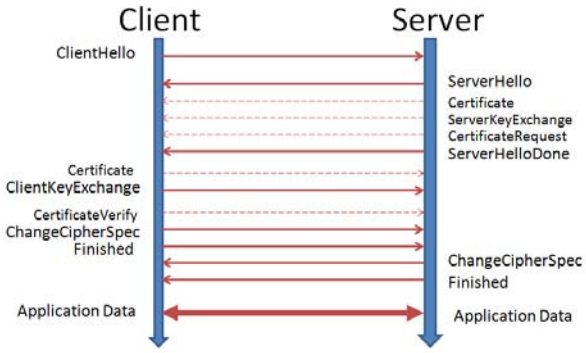

Figure 6. TLS handshake message flow

The TLS handshake protocol involves the following steps:

- Exchange hello message
- Exchange the necessary cryptographic parameters
- Exchange certificates and cryptographic information
- Generate a master secret from the premaster secret and exchanged random values
- Allow the client and server to verify that their peer has calculated the same security parameters.

TLS handshake takes a relatively long time, because the client has to do time consuming computation work. However, this process happens only once after the VoIP program is executed no matter how many calls are set up later.

During SIP registration, only four messages – Register, 401 Unauthorized, another Register, and 200 OK – are exchanged. Hence it takes a small amount of time in both SIP and SIP with TLS case, we did not consider the time of this part in our analysis. SIP call set up process includes INVITE, Trying, Ringing, OK, and ACK messages. Unlike registration, both caller and callee have to exchange messages with server to set up the call. Therefore, it takes a little longer than SIP registration.

C. Results and Analysis

This section shows the statistical result of TLS handshake time and SIP call setup time after a hundred times of experiments. We compare normal call setup case and SIP with TLS case by showing the average, maximum, minimum and standard deviation time of both cases.

TLS handshake time is usually ignored in other researches of VoIP security. Most of them just focus on the call setup time after TLS handshake. In this paper, we analyze this part to provide a reference to see whether it is acceptable to implement TLS in mobile platform. TLS handshake time is composed of the cryptographic computation time and the time needed to transmit the messages. As you can see in table 1, average TLS computation time is 10.756 seconds, which is the main part of TLS handshake time and the transmission time is only 0.176 seconds. Although total TLS handshake time is 10.931 seconds, this handshake stage happens only once after user starts up the VoIP program.

TABLE I. TLS HANDSHAKE TIME

	Computation (sec)	Transmission (sec)	Total(sec)
Average	10.756	0.176	10.932
MAX	19.276	0.595	19.437
MIN	8.418	0.091	8.685
STDEV	2.078	0.076	2.083

STDEV: Standard Deviation

TABLE II. SIP HANDSHAKE TIME COMPARISON

	Without TLS (sec)	With TLS (sec)
Average	1.428	2.001
MAX	1.906	2.488
MIN	0.827	1.45
STDEV	0.147	0.188

STDEV: Standard Deviation

SIP call setup time measures the time between SIP INVITE message transmission and the first voice message transmission. If normal SIP is used, the messages are exchanged between client and server in plaintext, while if TLS is used as the transport protocol, the messages are encrypted before they are exchanged. As is shown in table 2, when TLS is used, the average SIP handshake time is 2.001, while without TLS, it is 1.428 seconds, which is 0.573 seconds faster. In another word, using TLS takes about 40% more time than just use normal SIP.

Fig. 7 and Fig. 8 show the needed time distribution of normal SIP call setup and SIP with TLS call setup respectively. Most of the experimental results are between 1.3 seconds and 1.5, which is around the average value in normal case. Most of the results are also around the average value in SIP with TLS case.

IV. CONCLUSION AND FUTURE WORK

The dramatic increasing of VoIP services, for example, Skye, Vonage, Google Talk, etc, benefit both normal users and service providers. Moreover, smartphone is becoming the market trend and with the ever increasing performance of smartphone, mobile VoIP is also a growing market. However,

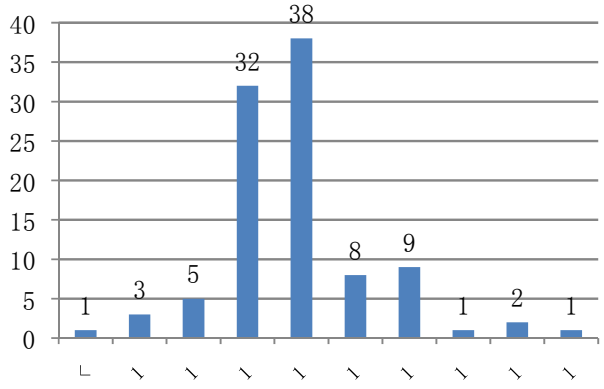

Figure 7. SIP handshake time distribution without TLS

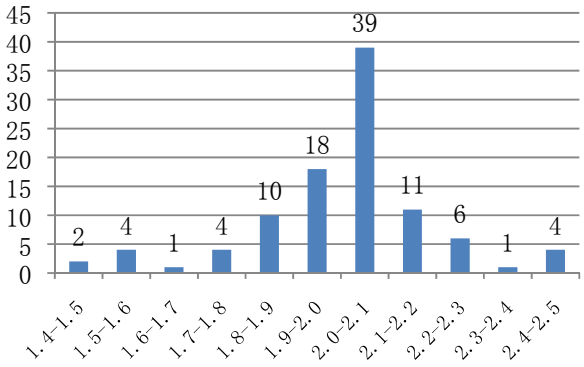

Figure 8. SIP handshake time distribution with TLS

security concerns have been bothering the users and service providers for a long time. The wireless network environment and the limited device performance would make it even harder to provide good service with security support.

In this paper, we implemented VoIP in mobile platform, which uses Sipdroid as the client and Asterisk as the server. Based on this, we implemented TLS in the same environment, and evaluated performance of both normal SIP and SIP with TLS. Our experimental result shows that it takes a relatively long time to register TLS. However, this is durable since this process happens only once. After TLS registration, users can make many calls as long as the VoIP program is still on. Through the comparison of call setup time between normal SIP and SIP with TLS, we can see that call setup time is increased for about 40%. Therefore, a more lightweight security protocol is needed to improve VoIP service performance.

In the future, we will implement SRTP with various key management protocols such as SDES, MIKEY and IKEv2, and compare their performance. Also, we will implement IPsec in both signaling and media transport layer to see if it is more efficient. As a result, this work will derive the lightweight mobile VoIP security mechanism.

ACKNOWLEDGMENT

This research was supported by the ICT Standardization program of MKE(The Ministry of Knowledge Economy).

REFERENCES

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston,J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. IETF RFC 3261, June 2002.

[2] M. Handley and V. Jacobson. SDP: session description protocol. IETF RFC 2327, April 1998

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. IETF RFC 3550, July 2003

[4] http://en.wikipedia.org/wiki/Mobile_VoIP

Γ. Dierks and E. Rescorla.TLS: transport layer security protocol IETF ℜFC 5246, August 2008.M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (SRTP). IETF RFC 3711, March 2004.

[6] http://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol

[7] http://www.asterisk.org
[8] http://www.remiphilippe.fr/2010/05/30/sips-on-asterisk-sip-security-with-tls/
[9] http://en.wikipedia.org/wiki/Transport_Layer_Security
[10] http://sipdroid.org/