The Authentication and Processing Performance of Session Initiation Protocol (SIP) Based Multi-party Secure Closed Conference System

Jongkyung Kim[1], Hyuncheol Kim[1], Seongjin Ahn[2], and Jinwook Chung[1]

[1] Department of Computer Engineering, Sungkyunkwan University, Suwon, Korea
{jongkkim,hckim,jwchung}@songgang.skku.ac.kr
[2] Department of Computer Education, Sungkyunkwan University, Seoul, Korea
sjahn@comedu.skku.ac.kr

Abstract. This paper presents, a new SIP based multi-party secure closed conference system. In the traditional participants, except of captain, conference system doesn't have a privilege for UA (User Agent) that accepts or declines new participants. On the other hand, closed conference system supports this function. We also propose for closed conference system authentication procedure. By means of a real implementation, we provide an experimental performance analysis of SIP security mechanisms.

1 Introduction

The system configuration for multi-party SIP based conference (including 3 parties or over) can be largely classified into three models. As shown in Fig. 1(a), an end user system should cover all the signaling and media mixing during the conference, resulting in a significant overload on the system. In the case of distributed multipoint conference model, as shown in Fig. 1(b), an additional terminal leads to an increase in multicast recipient. This means that an additional INVITE message is inefficiently required whenever such process is repeated. Finally, Ad-hoc conference model adopts a simple structure where the central media server or MCU (Multipoint Control Unit) processes all the signaling messages and performs media mixing so that each end user manages only his own traffics [1][2].

The SIP message contains information that a user or server wishes to keep private. Securing SIP header and body information can be motivated by two reasons. One is that we need maintain private user and network information in order to guarantee a certain level of privacy. Another is that we have to avoid SIP sessions being set up or changed by attackers faking the identity of someone else. The SIP UAC (User Agent Client), calling side, can identity itself to a UAS (User Agent Server), called side. Therefore, SIP authentication applies only to user-to-user, user-proxy or user-registrar communication [3]

A representation of the SIP digest authentication procedure is given in Fig. 2. The function *F()* used to compute the response specifies how to combine the input pa-

J. Cao et al. (Eds.): ISPA 2004, LNCS 3358, pp. 725–729, 2004.

rameter with some iterations of a digest algorithm [4][5]. The authentication procedure is run when the UAS, a proxy server, or the registrar server requires the calling side (UAC) to be authenticated before accepting the call, forwarding the call, or accepting the registration.

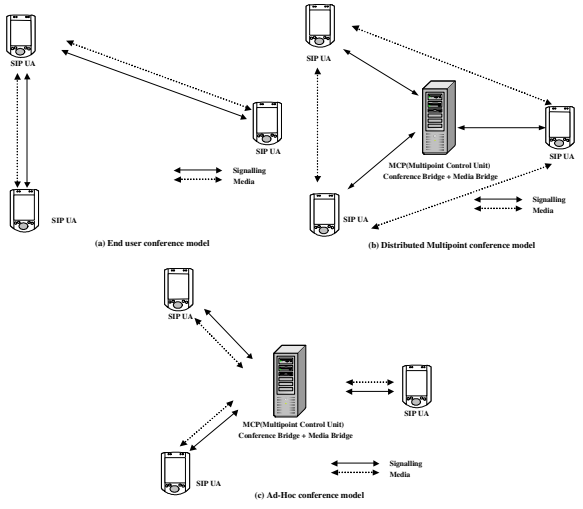

Fig. 1. Configuration for SIP-Based Multi-Party Conference System

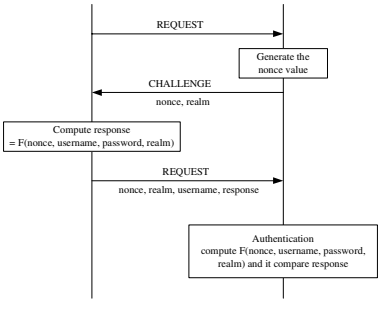

Fig. 2. The Procedure of Digest Authentication

In case of the existing conference system, as shown above, only the captain of conference has the right of invitation of additional participants. However, closed conference system needs all participants' anonymous agreement is needed to invite additional participants. In this paper, the closed conference model and the extended header will be proposed. The reminders of this paper are organized as follows. The authentication procedure for closed conference system is proposed in section 2. The section 3 describes the methodology for the evaluation of processing cost and experimental results. Finally, we make a conclusion.

2 Proposed Authentication Procedure for Closed Conference System

The simple expansion of 1:1 connection isn't able to guarantee satisfactory confidence. For instance, when a new participant joins conference session, existing participants and new one have to know the information of who has joined conference. That is why any one among them doesn't want to join conference together. Therefore, we need to know all participants of conference. We use the method that is the usage of INVITE message's SDP (Session Description Protocol) from the conference server [6].

Fig. 3 shows the procedure of the invitation of a participant in the closed conference system. To begin with Alice and Bob setup call session through authentication procedure. When the captain, Bob, invites Carol, Bob's UA sends NOTIFY message to Alice in order to agree an additional invitation. When Alice agrees to invite Carol, UA sends 200 OK message (F25). However, when he doesn't want to do that, it sends 603 declined message (F25). If Alice agrees the invitation of Carol, next time, Conference Server asks Bob to join conference. Bob checks the received information of participants. If he doesn't want to join that conference, he sends 603 messages (F43). This procedure presents with gray box in Fig. 3.

The states of the proposed UA can be divided into StateIdle (initial state), StateTrying, StateRinging, and StateInCall. The initial state is changed to StateRinging state when a phone is answered and followed by a receipt of ringing message, as shown in Fig. 4. When a ring is given someone, INVITE message is sent so that the status is set at StateTrying state. In the case of conference during the process, the REFER message is sent. If an INVITE message is received during StateIdle state, a negotiation of multimedia to be used in the session is followed by sending Ringing OK message (Real message: 180 Ringing). After that, it is changed to StateRinging state. At that time, if network resources are lacking or the phone is on the line, a receipt of PRACK message during StateRinging state results in sending 200 OK and INVITE OK(200) messages. After that, it is changed to StateInCall state.

3 Methodology for the Evaluation of Processing Cost and Experimental Results

In order to experiment with advanced feature in SIP, we have realized a test bed [7]. The goal of this test bed was twofold: firstly verification of the function behavior of the various elements and their interoperability; secondly the possibility of making some performance analysis. In particular regarding performance analysis, an interesting point is the evaluation of the cost to be paid in terms of performance for in the introduction of security mechanisms.

The results of our evaluation are reported in Table 1. The Third column reports the experiential average cost of twenty times in terms of second. Note that this throughput corresponds to 100 percent utilization of elements processing resources. The two rightmost columns are the most important ones and report the throughput value converted in a relative processing cost.

The result show that the introduction of SIP security accounts for nearly 30 percent of processing cost of no authentication procedure. This increase can be explained with the increase in number of exchanged SIP messages. Another interesting finding is that the incensement of processing time cost increase accidentally, when third attendant joins conference.

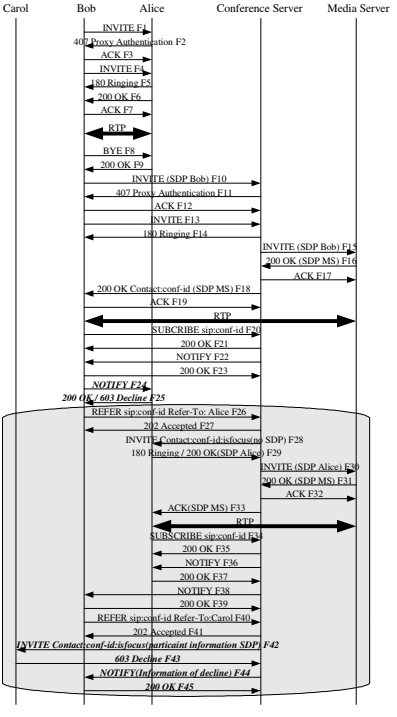

Fig. 3. The Procedure of Closed Conference System

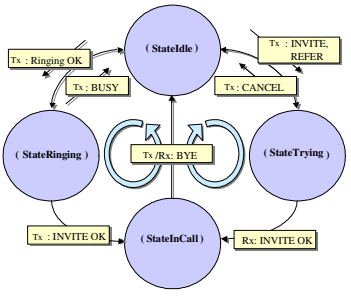

Fig. 4. Proposed SIP UA State Transition Diagram

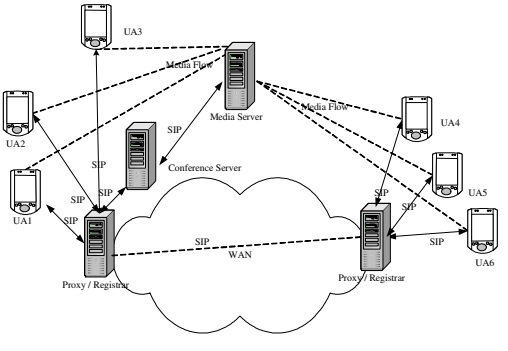

Fig. 5. The Test Bed Layout

Table 1. Experimental results

	Procedure/scenario	Processing time cost(second)	Relative cost
1	No authentication, 2 attendants	2.873 * 10^-2	100
2	No authentication, 3 attendants	4.022 * 10^-2	140
3	No authentication, 4 attendants	4.252 * 10^-2	148
4	No authentication, 5 attendants	4.453 * 10^-2	155
5	No authentication, 6 attendants	4.683 * 10^-2	163
6	Authentication, 2 attendants	3.735 * 10^-2	130
7	Authentication, 3 attendants	5.200 * 10^-2	181
8	Authentication, 4 attendants	5.545 * 10^-2	193
9	Authentication, 5 attendants	5.746 * 10^-2	200
10	Authentication, 6 attendants	6.119 * 10^-2	213

4 Conclusion

In this study, SIP based multi-party closed conference system and authentication procedure is described. The performance aspects of SIP authentication for closed conference system are considered with pure experimental approach. The processing costs of different security procedure/scenario are compared under a reference implementation. Although the performance results are obviously conditioned by the specific implementation aspects, they can be rough idea of relative processing cost of SIP security procedures.

References

[1] Jonathan Rosenberg, et al.: Models for Multi Party Conferencing in SIP, Jan. 2003
[2] A. Johnson: SIP Call Control-Conferencing for User Agents, Oct. 2003
[3] Rohan Mahy: A Call Control and Multi-party usage Framework for the Session Initiation Protocol (SIP), September 2003
[4] J, Franks et al.: HTTP Authentication: Basic and Digest Access Authentication, IETF RFC 2617, June 1999
[5] R. Rivest: The MD5 Message-digest Algorithm, IETF RFC 1321, Apr 1992
[6] Janet R. Dianda, et al.: SIP, Bell Labs Technical Journal 7(1), 3-23(2002)
[7] Guy J. Zenner, Mark H.Jones, Amit A. Patel: Emerging Uses of SIP in Service Provider Networks, Bell Labs Technical Journal 8(1), 43-63(2003)