The 6th International Conference on Wireless Communications, Networking and Mobile Computing

September 23-25, 2010 Chengdu, China

Volumes

Main Menu
Volumes
Sessions
Authors

Click on a volume for a list of sessions

Sessions by Volume

Vol 5. Network Protocols

- Media Access Control
- Network Protocol and Congestion Control
- QoS and Traffic Analysis
- Network Security
- Other Related Topics

Main Menu
Volumes
Sessions
Authors

Papers by Session

Network Security

Main Menu
Volumes
Sessions
Authors

Papers by Session

SIP-based IM and Its Security Solutions

Xin Cui
School of Business
Shandong University at Weihai
Weihai, China
chlgms@sdu.edu.cn

Yuan Zhang
School of Info. Sci. & Eng.
University of Jinan
Jinan, China
yzhang@ujn.edu.cn

Woo Jin Lee and Seok Joo Koh
School of Electron. Eng.& Comp. Sci.
Kyungpook National University
Daegu, Korea
{woojin, sjkoh}@knu.ac.kr

***Abstract*—SIP-based instant messaging (IM) is currently a quite hot issue in the SIP world. Two approaches, pager-mode IM and session-mode IM, have been explored for implementation. This paper provides an in-depth analysis of the first method through an example, and makes improvement to the second by presenting specific signaling and media messages. Solutions to security problems are also discussed in detail for the two methods, respectively.**

Keywords- IM; pager-mode; session-mode

I. INTRODUCTION

The Session Initiation Protocol (SIP) [1] is a relatively new IETF signaling protocol for establishing real-time sessions over Internet Protocol (IP) networks. Each session may include different types of data ranging from text (e.g. instant messaging) to audio (e.g. calls) and video (e.g. video-conferencing). SIP has been designed to be a general-purpose protocol, which is open and scalable. However, extensions are needed to make it a truly functional platform.

Instant messaging (IM) is defined as the exchange of content between a set of participants. In most practical cases, however, the scenario is peer-to-peer. Instant messages are usually grouped together into brief live conversations online. As such a nearly real time conversation shares quite a bit in common with SIP's design philosophies, the adaptation of SIP to IM seems a natural choice given the widespread support for the SIP standard. The proposed solutions are mainly under consideration within IETF's SIMPLE working group [2].

Experts have been exploring two ways to implement IM, namely "pager-mode" and "session-mode". Although IM is a powerful complement to e-mail and telephone communications, the introduction of direct peer-to-peer links over a public IP network dramatically changes the corporate security posture. But there is still no elaborate discussion on this vital issue.

II. SIP FOR IM

A. Pager-mode IM

In this model, each IM is sent as an independent message using the MESSAGE method. There is no setup or session establishment needed to send a message. There is no explicit association between messages. Figure 1 shows an example message flow to facilitate the understanding of pager-mode IM service supported by SIP.

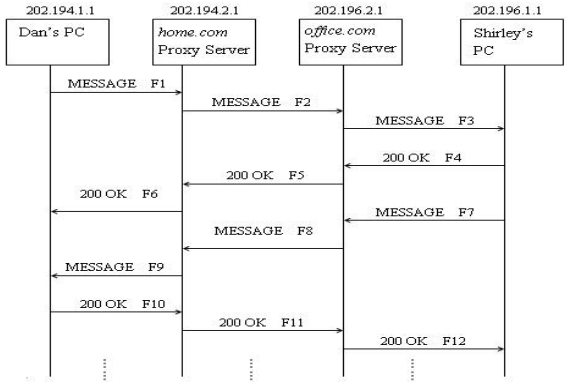


Figure 1. Pager-mode IM

Note that there should be no Contact header in the MESSAGE request. The *home.com* proxy server receives the MESSAGE request and forwards it on behalf of the requestor. It locates the proxy server at *office.com*, possibly by performing a DNS lookup, and thus obtains the IP address of the *office.com* proxy server to forward the MESSAGE request there (F2). The *office.com* proxy server recognizes that it is the server for *office.com*. Then it consults a database, generally called a location service, which contains the current IP address of Shirley. The *office.com* proxy server adds the third Via header field with its own IP address to the MESSAGE and proxies it to Shirley's PC (F3). The message is received by Shirley, displayed, and a response message (F4) is generated to be sent back.

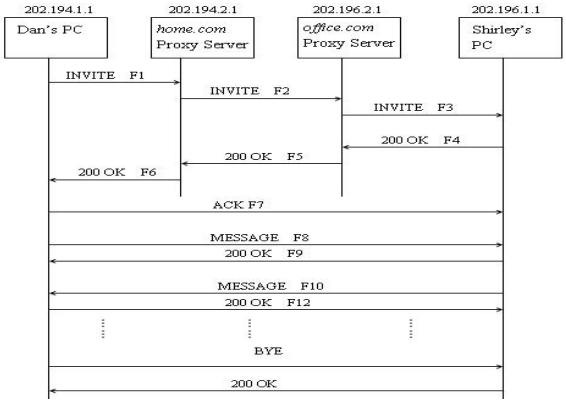
In this case most of the header fields are simply copied from the MESSAGE request. The *office.com* proxy server receives this response, strips off the top Via, and forwards it to the address in the next Via, *home.com* (F5). Finally, Dan will

B. Pager-mode IM

As was mentioned at the beginning of this paper, the IM traffic in this model is viewed as a "media stream" with a clear beginning and end. There is explicit association between messages. In this approach, a SIP endpoint that wishes to initiate a text chat session would send an INVITE request with an SDP body that describes the session. The sender and recipient then negotiate MESSAGE sessions using normal SIP

978-1-4244-3709-2/10/$25.00 ©2010 IEEE

conventions. To motivate the discussion, let us continue with



the former example (see figure 2).

Figure 2. Session-mode IM

This request has three main objectives: 1) user location: discovery of Shirley wherever located; 2) user availability: determination of the willingness of the called party, Shirley, to engage in communications; 3) user capabilities: negotiation and determination of the media formats to be used between Dan and Shirley. While the implementation of "user location" and "user availability" is managed much alike any other media session, there are some differences in the message session negotiation.

Instant messages usually have typically high volume. Due to the network congestion concerns and due to the reliable delivery of a message, message sessions must run over a connection-oriented, congestion-safe network transport, such as TCP or SCTP. But SDP provides no way to describe a session that uses protocols other than RTP or UDP.

The m-line in the INVITE message body specifies TCP for succedent media session. Whilst for message sessions, the media field must have the value of "message". The a-line indicates that Bob will both accept a TCP connection on the port number 5162, and that it will also initiate a connection to the port number on the m-line of Shirley's response. The 200 OK (F4) contains the SDP media description of the type of session that Shirley is willing to establish with Dan.

The a-line of the 200 OK response message will cause Dan to initiate a connection to port 6152 at 202.196.1.1, and Shirley to initiate a connection to port 5162 at 202.194.1.1. Whichever TCP connection succeeds will be used. If both succeed, one of the connections may be closed as an optimization. In order to minimize the chance that two connections are created, Shirley may opt to use the recommendation, which would result in events occurring in the following sequence:

- Dan sends SDP as listed above. He must enable a listener on port 5162 at this time, but is not able to initiate a TCP connection due to the fact that it does not have sufficient information from the Shirley.
- Shirley, upon receiving the SDP, immediately initiates a TCP connection to 202.194.1.1:5162.
- In order to minimize the chance of a duplicate connection, Shirley pauses for a short time to allow Dan to receive the TCP connection initiation.
- After the short pause, Shirley sends the SDP response as listed above.

Finally an ACK (F7) is sent from Dan to Shirley, by passing the two proxies, to confirm the reception of 200 OK response. Then their message session begins.

The following MESSAGE requests (F8, F10, etc.) are quite different from those in figure 1. In pager-mode, each message needs to include all of the SIP headers that are mandated by RFC 3261 [1]. However, some of these headers are not needed once a context is established for exchanging messages. For instance, the Via header indicates the path taken by the request so far and indicates the path that should be followed in routing responses. Since Dan definitely knows that Shirley now resides in the domain office.com, he can directly send the MESSAGE request to her through the existing connection opened by the INVITE/ACK transaction. As a result, Via header is not necessary. Note that Via, Cseq and Call-ID headers are all omitted. Considering the large volume of instant messages, less overheads in each individual message adds up to network congestion mitigation.

C. *Advantages of session-mode IM*

MESSAGE requests differ from other sorts of SIP requests in that they carry media as payload. The pager-mode by necessity causes MESSAGE to follow the SIP signal path, which is intended to manage sessions, not transfer content. While in session-mode, there is quite a potential to decouple the path of proxies used for the signaling from those possibly used as intermediaries for the MESSAGE requests comprising the session.

In pager-mode, since the MESSAGE is sent as a regular SIP request, there is nothing that could prevent a proxy from forking. But it does not make any sense. The sender will not know how to target the recipient for future messages. This problem can be resolved by establishing a session with INVITE. In that case, the caller does know who has received the session invitation, and can select which recipient to communicate with.

In addition, message sessions allow greater efficiency for secure message exchanges. In the next section, security problems including this aspect will be discussed in detail.

III. PROPOSED SOLUTIONS TO SECURITY ISSUES

Although IM provides a way for people to converse with their "buddies" in real time, as well as for enterprises to use as a valuable business tool, it brings forth wide acknowledged security problems. Even without realizing it, we open ourselves or the corporate infrastructure to a myriad of security threats including privacy issues (personal information leakage, IP address exposure, loss of confidential information, and eavesdropping), identity issues (impersonation), malware in transferred files (worms, viruses, Trojan horses, and other malicious software), and security bugs in the clients such as

buffer overflows that could expose users to any number of different types of attacks (denial of service attacks, worm infections, privilege-elevation attacks, Trojan attacks, etc.).

Consequently, IM session protocol must be compliant with the IM security requirements in RFC 2779 [4], which describes authentication, integrity and encryption of instant messages for implementation. Such security goals can be achieved in one of two general ways: 1) Data can be authenticated or encrypted at the level of the transfer protocol (e.g. using TLS) or, equivalently, at the bevel of the underlying transport protocol (e.g. using IPsec). This method requires the communicating parties to trust the intermediary servers to do their job properly. 2) Data can be authenticated and/or encrypted end-to-end, even though they do not enjoy a direct connection. This method does not require the communicating parties to trust any intermediaries, but does require them to agree on a security mechanism and key management structure.

In the opinion of the author, solutions to message integrity and confidentiality may vary with respect to different design philosophies of pager-mode and session-mode IM, which is to be discussed respectively as follows. Since authentication is quite appropriate to be fulfilled by IM's combined service, Presence, it is not the focus of this section.

A. Enhanced MIME body part for Pager-mode security

The SIP MESSAGE request inherits the S/MIME features of SIP, allowing a message to be signed and encrypted. RFC 3428 [3] specifies that MESSAGE bodies must be secured with S/MIME to provide end-to-end message integrity and confidentiality.

The S/MIME algorithms are set by RFC 3369 [5], which is derived from RFC 2315. Cryptographic Message Syntax (CMS) defines multiple content types. Of these, only the data, signed-date and enveloped-date content types are currently used for S/MIME. The Signed-Data content type is used to digitally sign the message body for integrity. Figure 3 illustrates the process by which Signed-Data is constructed.

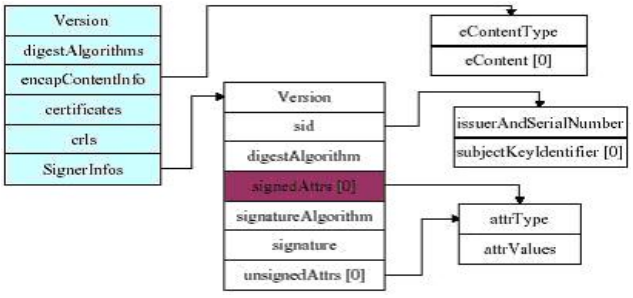


Figure 3. The process by which Signed-Data is constructed

The Enveloped-Data content type is used to apply data confidentiality to a message. Figure 4 to figure 7 are the pictorial description of the process by which Enveloped-Data is constructed.

When a user agent client (UAC) sends a MESSAGE request outside the context of a dialog, the UAC should structure the body as an S/MIME 'multipart/signed' CMS Signed-Data body. If the desired CMS service is Enveloped-Data (and the public key of the target user is known), the UAC should send the Enveloped-Data message encapsulated within a Signed-Data message.

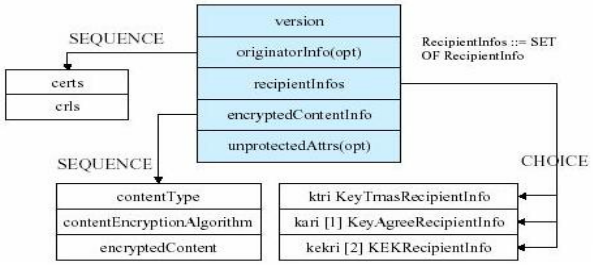


Figure 4. Description of the process

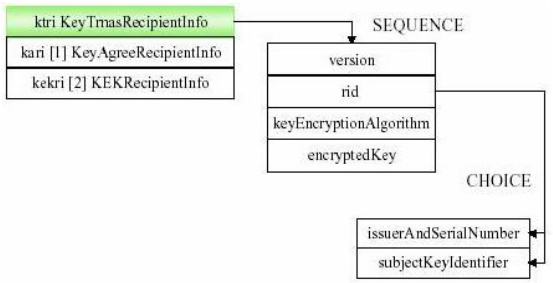


Figure 5. Procedure of KeyTmasRecipientInfo

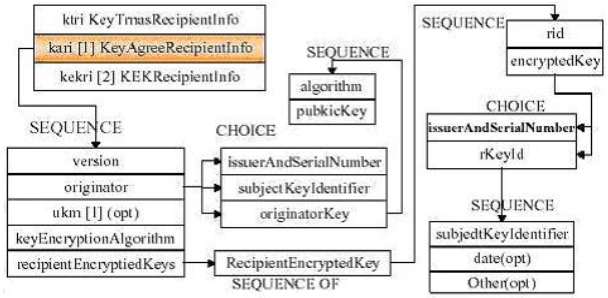


Figure 6. Procedure of KeyAgreeRecipientInfo

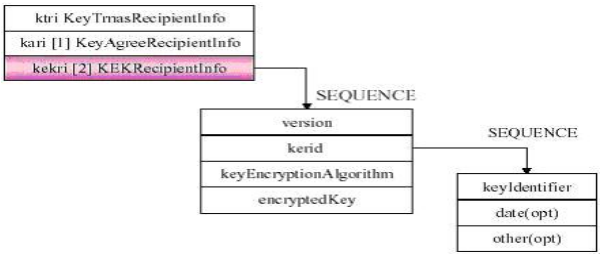


Figure 7. Procedure of KEKRecipientInfo

B. MIMEY for session-mode IM

The S/MIME approach requires public key operations for each message. With session-mode messaging, however, a session key can be established at the time of session initiation. This key can be used to protect each message that is part of the

session. Certain challenges exist in multimedia transactions: 1) It is largely real-time, 2) It is heterogeneous, 3) Interactive, and 4) Bandwidth computation intensive. More over, it is desirable that such session keys can be established through SIP. Consequently MIKEY is up to now a good choice. MIKEY, as a key management protocol, is designed to have the following characteristics:

- End-to-end security. Only the participants have access to the generated key(s).
- Simplicity.
- Efficiency: a) low bandwidth consumption; b) low computational workload; c) small code size; d) minimal number of roundtrips.
- Tunneling. Possibility to "tunnel"/integrate MIKEY in session establishment protocols (e.g. SIP).
- Independent of any specific security functionality of the underlying transport.

The key management procedure (setup a CSB and create a TEK/Data SA) is shown in figure 8.

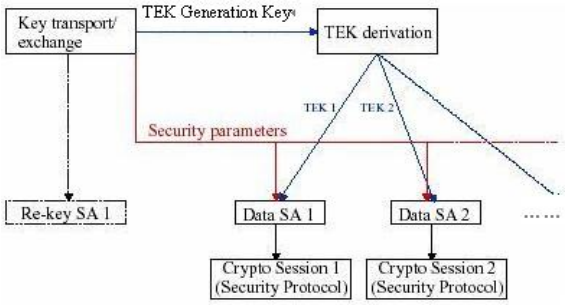


Figure 8. Overview of the key management procedure

MIKEY in particular defines three key management schemes that all finish their task within just on roundtrip.

- a symmetric key distribution protocol based upon pre-shared master keys;
- a public-key encryption-based key distribution protocol assuming a public-key infrastructure with RSA-based private/public keys and digital certificates;
- a Diffie-Hellman key agreement protocol deploying digital signatures and certificates.

However, considering its usage in IM, the first and the third method have their constraints and limitations respectively. The symmetric key distribution protocol is simple to implement but does not nicely scale in any larger configuration of potential peer entities due to the need of mutually pre-assigned shared master secrets. Moreover, the security provided does not achieve the property of perfect forward secrecy; i.e. compromise of the shared master secret would render past and even future session keys susceptible to compromise. Although the Diffie-Hellman key agreement scheme has the advantage of providing perfect forward secrecy, it is only a good alternative in the special peer-to-peer case in that the two involved parties generate a unique random key, making a group TEK establishment impossible. Thus the public-key encryption scheme, which depends upon a public-key infrastructure that certifies the private-public keys by issuing and maintaining digital certificates, is recommended.

IV. Conclusion and Future Work

To its credit, SIP has big-name support in the world of telephony and wireless messaging. With respect to IM, it is still very much in the development phase. Several vendors, e.g. Microsoft, AOL Time Warner, etc., have Pulled back from SIMPLE due in part to concerns about the robustness of security and other features. With core mechanics already got in place, there are many areas associated with previous discussion that needs improvement and revision. Only a few of them have been identified and are presented below.

- Due to network congestion concerns, pager-mode MESSAGE requests have a limitation of 1300 bytes, a prohibition against overlapping requests, etc. These are quite undesirable.
- Discussion on negotiating what MIME types are permitted in the bodies of session-mode MESSAGE request is needed. Likely candidates may include message/cpim.
- The largest outstanding defect with the S/MIME mechanism is the lack of a prevalent public key infrastructure for end users. If self-signed certificates are used, the SIP-based key exchange mechanism described in Section 23.2 of [1] is susceptible to a man-in-the-middle attack with which an attacker can potentially inspect and modify S/MIME bodies.
- Public-key infrastructures are not widely available and in general, implementations are significantly more complex. Is it possible to combine the security mechanisms to put forth a synergetic optimization?
- It would be really ugly to have different semantics for MESSAGE requests depending on whether they are in-session or stand-alone. Can session-mode be placed as a replacement for pager-mode in the future?

Acknowledgment

This research was supported by the ITRC program of MKE/NIPA (NIPA-2010-C1090-1021-0002) and the IT R&D program of MKE/KEIT (10035245).

References

[1] Rosenberg J., Schulzrinne H., Camarillo G., et al., 'SIP: Session Initiation Protocol', RFC 3261, IETF, June 2002

[2] http://www.ietf.org/html.charters/simple-charter.html

[3] Campbell B., Rosenberg J., Schulzrinne H., *et al*., "Session Initiation Protocol Extension for Instant Messaging", RFC 3428, IETF, Dec. 2002.

[4] Day M., Aggarwal S., Mohr G., Vincent J., "Instant Messaging / Presence Protocol Requirements", RFC 2779, IETF, February 2000.

[5] .Housley R.:, "Cryptographic Message Syntax (CMS)", RFC 3369, IETF, August 2002.