SIPPING
Internet-Draft
Expires: August 16, 2004

K. Ono
S. Tachimoto
NTT Corporation

February 16, 2004

Requirements for End-to-middle Security for the Session Initiation Protocol (SIP)
draft-ietf-sipping-e2m-sec-reqs-01

Abstract

A SIP User Agent (UA) does not always trust all proxy servers in a request path to decide whether or not to inspect the message bodies and/or headers contained in a message. The UA might want to protect the message bodies and/or headers from proxy servers excluding the particular proxy that provides some services based on their content. This situation requires a mechanism for securing information passed between the UA and an intermediary proxy, also called "end-to-middle security", which does not interfere with end-to-end security. This document defines a set of requirements for a mechanism to achieve end-to-middle security.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

Table of Contents

1. Introduction

The Session Initiation Protocol (SIP) [2] supports hop-by-hop security using Transport Layer Security (TLS) [3] and end-to-end security using Secure MIME (S/MIME) [4]. This assumes that a SIP UA trusts all proxy servers in a request path to decide whether or not to inspect the message bodies contained in a message.

However, there is a model where trusted and partially-trusted proxy servers are mixed along a message path. The partially-trusted proxy servers are only trusted by users in terms of the SIP routing. The proxy servers are not trusted by users to inspect data except routing headers. Hop-by-hop confidentiality services using TLS are not suitable for this model. End-to-end confidentiality services using S/MIME are also not suitable when the intermediaries provide services based on reading the message bodies and/or headers. This problem is described in Section 23 of [2].

One example of such services is a firewall traversal. A firewall entity that supports the SIP protocol or a midcom [5] agent co-located with a proxy server controls a firewall based on certain Session Description Protocol (SDP) attributes in a SIP transaction.

Another example is transcoding [6]. A transcoder related to a proxy server transfers coding based on certain SDP attributes in a SIP transaction or transfers text-to-speech based on a message body in the MESSAGE [7] method.

A third example is the archiving of instant messaging traffic, where the archiving function co-located with a proxy server logs the message bodies in the MESSAGE method. This service might be deployed for financial or health care applications, where achiving communications is required by policies, as well as other applications.

In these cases, a UA might want to protect the message bodies and/or headers from proxy servers excluding the particular proxy server that provides these services. Conversely, a proxy server might want to view the message bodies and/or headers to provide these services. Such a proxy server is not always the first hop for the UA. These situations require security between the UA and the intermediary proxy server for the message bodies and/or message headers. We call this "end-to-middle security".

End-to-middle security consists of authentication, data integrity and data confidentiality. Above examples mainly require data confidentiality for end-to-middle security. For authentication, proxy servers usually require to authenticate a user that sends a request

message. The user also requires to authenticate the proxy that has the user's credential. HTTP digest authentication described in [2] can be used for mutual authentication for the request message. The authenticating proxy is not limited to the first hop for the UA. Thus, HTTP digest authentication can be used for end-to-middle security. To avoid replay attacks, the HTTP digest authentication needs to be used with a security mechanism for confidentiality such as TLS. HTTP digest authentication does not support authentication for an originator of a response message. Digital signatures obtained from a Public Key Infrastructure, S/MIME Cryptographic Message Syntax (CMS) [8] SignedData body, can be used for the authentication. Since these mechanisms achieve authentication for end-to-middle security, the requirements are not discussed in this document.

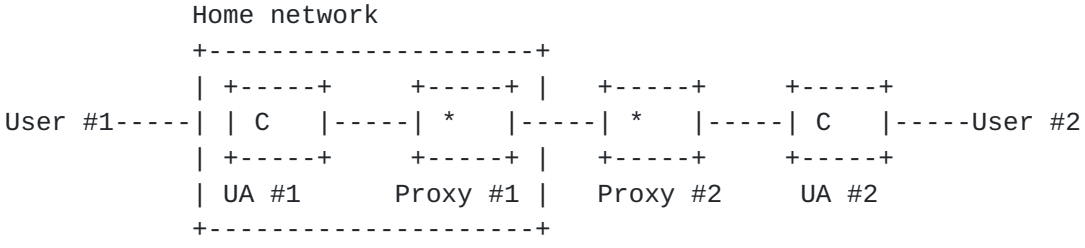
As for data integrity, proxy servers require to validate the content to be used for providing some services. The CMS SignedData body might be used in a mechanism for end-to-middle security. The CMS SignedData body can be created with the original data and the originator's private key, and anyone can verify the data integrity by using the originator's public key and the certificate. That is, proxy servers can verify the data integrity whenever they require. Thus, the CMS SignedData body could be used to implement end-to-middle security at the same time as using end-to-end security. Currently, proxy servers cannot require UAs to send a message with the CMS SignedData body. Some new mechanisms are needed to achieve data integrity for end-to-middle security.

This document mainly discusses requirements for data confidentiality and the integrity of end-to-middle security. Proposed mechanisms are discussed in [9].

2. Problems with the Existing Situations

We describe here examples of models in which trusted and partially trusted proxy servers both exist in a message path. These situations demonstrate the reasons why end-to-middle security are required in certain scenarios.

In the following example, User #1 does not know the services provided by or security policies of Proxy #1. User#1 sends an INVITE request including S/MIME-encrypted SDP for end-to-end security as shown in Figure 1. Proxy #1 may reject the request because it cannot offer a firewall traversal service. Or Proxy #1 may erase the encrypted data in the request based on a strict security policy that prohibits the forwarding of unknown data. Thus, the UA will need to discover if information requirements to receive intermediary's services or security policies will conflict with end-to-end confidentiality.

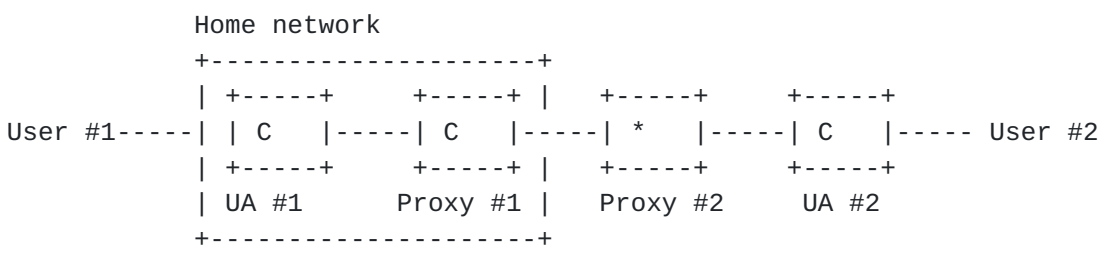


C: Content that UA #1 allows the entity to inspect
*: Content that UA #1 prevents the entity from inspecting

Figure 1: Deployment example #1

In the second example, Proxy server #1 (Proxy #1) is the home proxy server of User #1 using UA #1. User #1 communicates with User #2 through Proxy #1 and Proxy #2 as shown in Figure 2. UA #1 already knows the public key certificate of Proxy #1, and it allows Proxy #1 to inspect the message bodies in a request for some purpose. However, User #1 does not know whether Proxy #2 is trustworthy, and thus wants to protect the message bodies in the request. The UA will need to be able to grant a trusted intermediary permission to inspect message bodies while preserving their confidentiality with respect to other intermediaries.

Even if UA #1's request message authorizes a selected proxy server (Proxy #1) to see the message body, UA #1 is unable to authorize the same proxy server to see the message body in the response from UA #2. The originating UA will need to designate and share a key that can be reused as a content encryption key (CEK) for bidirectional exchanges of S/MIME-secured messages in SIP.

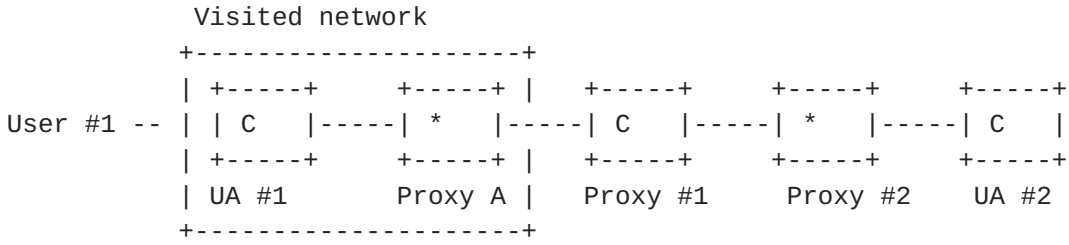


C: Content that UA #1 needs to disclose
*: Content that UA #1 needs to protect

Figure 2: Deployment example #2

In the third example, User #1 connects UA #1 to a proxy server in a Visited (potentially hostile) network, e.g. a hotspot service or a roaming service. Since User #1 wants to utilize certain home network services, UA #1 connects to a home proxy server, Proxy #1. However, UA #1 must connect to Proxy #1 via the proxy server of the visited network (Proxy A), because User#1 must follow the policy of that network. Proxy A may perform access control based on the destination addresses of calls. User #1 trusts Proxy A to route requests, but not to inspect the message bodies they contain as shown in Figure 3. User #1 trusts Proxy #1 both to route requests and to inspect the message bodies for some purpose.

The same problems as in the second example also exist here.



C: Content that UA #1 needs to disclose
*: Content that UA #1 needs to protect

Figure 3: Deployment example #3

3. Requirements for a Solution

We describe here requirements for a solution. The requirements are mainly applied during the phase of a dialog creation or sending a MESSAGE method.

3.1 General Requirements

Following are gerenal requirements for end-to-middle confidentiality and the integrity.

1. It SHOULD have little impact on the way a UA handles messages with S/MIME bodies.

2. It SHOULD have no impact on proxy servers that do not provide services based on S/MIME bodies in terms of handling the existing SIP headers.

3. It SHOULD have little impact on the standardized mechanism of proxy servers that provide services based on S/MIME bodies.

   When a proxy server receives an S/MIME message, it should be able to quickly and easily determine the necessity to investigate the S/MIME body. This can be restated as:

   + It SHOULD allow proxy servers to quickly and easily determine whether to handle S/MIME bodies and, if so, how and which ones.

4. It SHOULD allow a proxy server to notify a UA about the proxy server's security policy for a request/response.

5. It SHOULD allow a proxy server to notify a UA what data in a request/response is needed in order to provide a service.

3.2 Requirements for End-to-middle Confidentiality

1. The solution MUST be compatible with end-to-end encryption. The encrypted data can be shared with the end user and selected proxy server, if needed.

2. It MUST NOT violate end-to-end encryption when the encrypted data does not need to be shared with any proxy servers.

   For example, keying materials for secure RTP (SRTP) in SDP [11] can be included only in the end-to-end encryption, if the UA's policy is such.

3. It SHOULD allow a UA to discover which proxy server needs to view some data in a request/response message for a certain service, and discover what data is needed.

   This requirement is necessary when the UA does not know which proxy or domain provides the service in advance.

4. It MUST allow a UA to request selected proxy servers to view specific message bodies. The request itself SHOULD be secure.

5. It SHOULD allow a UA to request the recipient UA to disclose the same information that the requesting UA is providing to the proxy server to the same proxy server. The request itself SHOULD be secure.

   It is not reasonable to expect the recipient UA have knowledge of the public key certificate of the proxy server on the originating network. This can be restated as:

   + The solution SHOULD allow a UA to request the opposite-side UA to reuse a CEK in subsequent messages during a dialog.

   + It SHOULD allow a UA to request a selected proxy server to keep a CEK in a message during a dialog. The requests themselves SHOULD be secure.

6. It MAY allow a UA to notify the opposite-side UA which proxy server needs to view some data in a request/response for the services.

7. It MAY allow a UA to notify the opposite-side UA what data the proxy server is permitted to view in a request/response for the services.

   These last two requirements might be needed when there are a firewall in the network on UAS's side. A UAS need to notify a UAC to disclose the SDP in an INVITE message to a proxy server that control the firewall in the UAS side. Such notification might be applied to a registration phase.

3.3 Requirements for End-to-middle Integrity

1. It SHOULD work even with SIP end-to-end integrity service enabled.

2. It SHOULD allow a UA to discover what data in a request/response the proxy needs to verify in order to provide the service.

This requirement is necessary when the UA does not know what data is used to provide the service in advance.

3. It MUST allow a UA to request selected proxy servers to verify specific message bodies. The request itself SHOULD be secure.

4. It SHOULD allow a UA to request the recipient UA to send the verification data of the same information that the requesting UA is providing to the proxy server. The request itself SHOULD be secure.

5. It MAY allow a UA to notify the opposite-side UA what data the proxy server needs to verify in a request/response for the services.

4. Security Considerations

This documents present requirements including security viewpoints in Section 3.

5. IANA Considerations

This document requires no additional considerations.

6. Changes from 00.txt

- Reworked the sub-sections in Section 3 to clarify the objectives, separating end-to-middle confidentiality and integrity.

7. Acknowledgments

Thanks to Rohan Mahy and Cullen Jennings for their initial support of this concept, and to Jon Peterson, Gonzalo Camarillo, and Sean Olson for their helpful comments.

References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[3] Allen, C. and T. Dierks, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[4] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1992.

[5] Srisuresh, P., Kuthan, J., Rosenberg, J., Brim, S., Molitor, A. and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[6] Camarillo, G., "Framework for Transcoding with the Session Initiation Protocol", draft-ietf-sipping-transc-framework-00.txt (work in progress), February 2004.

[7] Campbell, Ed., B., Rosenberg, J., Schulzrinne, H., Huitema, C. and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[8] Housley, R., "Cryptographic Message Syntax", RFC 2630, June 1999.

[9] Ono, K. and S. Tachimoto, "End-to-middle security in the Session Initiation Protocol(SIP)", draft-ono-sipping-end2middle-security-01 (work in progress), Feb. 2004.

[10] Baugher, M., Carrara, E., McGrew, D., Naslund, M., McGrew, D. and K. Norrman, "The Secure Real-time Transport Protocol", draft-ietf-avt-srtp-09.txt (work in progress), July 2003.

[11] Andreasen, F., Baugher, M. and D. Wing, "Session Description Protocol Security Descriptions for Media Streams", draft-ietf-mmusic-sdescriptions-03.txt (work in progress), February 2004.

Authors' Addresses

Kumiko Ono
Network Service Systems Laboratories
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-shi, Tokyo 180-8585
Japan

EMail: ono.kumiko@lab.ntt.co.jp

Shinya Tachimoto
Network Service Systems Laboratories
NTT Corporation
9-11, Midori-Cho 3-Chome
Musashino-shi, Tokyo 180-8585
Japan

EMail: tachimoto.shinya@lab.ntt.co.jp

Intellectual Property Statement

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.