SIPPING
Internet-Draft
Expires: December 29, 2004

K. Ono
S. Tachimoto
NTT Corporation

June 30, 2004

Requirements for End-to-Middle Security for the Session Initiation Protocol (SIP)
draft-ietf-sipping-e2m-sec-reqs-03

Status of this Memo

Copyright Notice

Abstract

A SIP User Agent (UA) does not always trust all intermediaries in its request path to inspect its message bodies and/or headers contained in its message. The UA might want to protect the message bodies and/or headers from intermediaries except those that provide services based on its content. This situation requires a mechanism called "end-to-middle security" to secure the information passed between the UA and intermediaries, 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]. These security mechanisms assume that a SIP UA trusts all proxy servers along its request path to inspect the message bodies contained in the message, or a SIP UA does not trust any proxy servers to do so.

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 to provide SIP routing, but these proxy servers are not trusted by users to inspect its data except routing headers. A hop-by-hop confidentiality service using TLS is not suitable for this model. An end-to-end confidentiality service using S/MIME is 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].

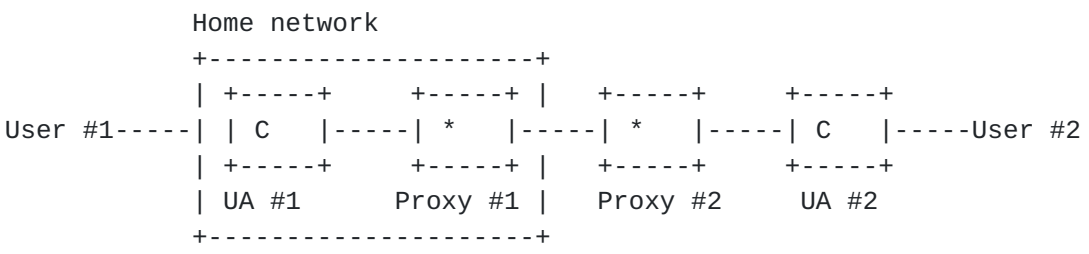
In some cases, a UA might want to protect its message bodies and/or headers from proxy servers along its request path except from those that provides services based on reading its message bodies and/or headers. Conversely, a proxy server might want to view the message bodies and/or headers to sufficiently provide these services. Such proxy servers are not always the first hop from the UA. This situation requires a security mechanism to secure message bodies and/or headers between the UA and the proxy servers, yet disclosing information to those that need it. We call this "end-to-middle security".

2. Use Cases

2.1 Examples of Scenarios

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

In the following example, User #1 does not know the security policies or services provided by Proxy server #1 (Proxy#1). User #1 sends a MESSAGE [5] request including S/MIME-encrypted message content for end-to-end security as shown in Figure 1, while Proxy #1 erases the encrypted data in the request or rejects the request base on its strict security policy that prohibits the forwarding of unknown data. For the MESSAGE request to correctly traverse Proxy #1, the UA will need to discover if end-to-end confidentiality will conflict with intermediary's services or security policies.

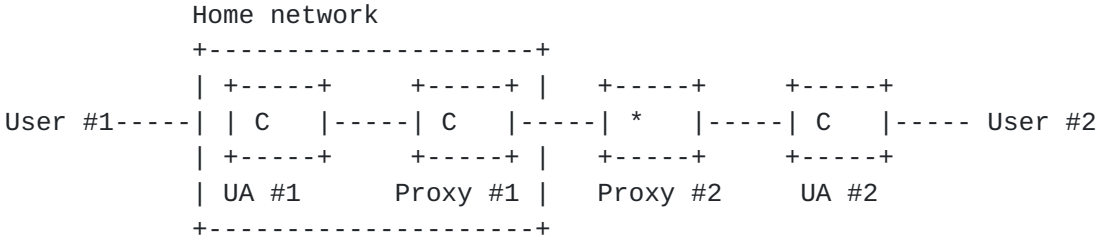


C: Content that UA #1 allows the entity to inspect
*: Content that UA #1 prevents the entity from inspecting

Figure 1: Deployment example #1

In the second example, Proxy server #1 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. Although User #1 already knows Proxy #1's security policy which requires the inspection of the content of the MESSAGE request, User #1 does not know whether Proxy #2 is trustworthy, and thus wants to protect the message bodies in the request. To accomplish this, UA #1 will need to be able to grant a trusted intermediary (Proxy #1) to inspect message bodies, while preserving their confidentiality from other intermediaries (Proxy #2).

Even if UA #1's request message authorizes a selected proxy server (Proxy #1) to inspect the message bodies, UA #1 is unable to authorize the same proxy server to inspect the message bodies in subsequent MESSAGE requests from UA #2.



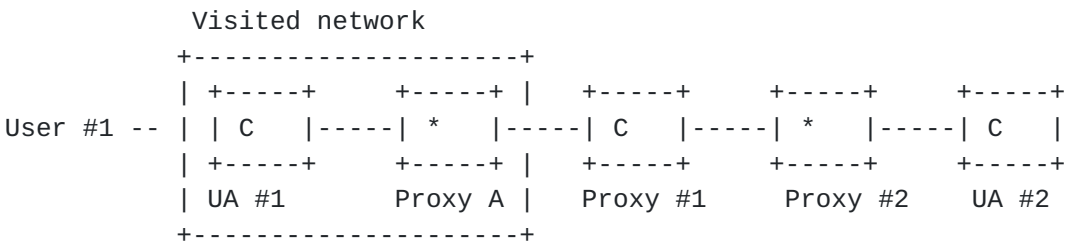
C: Content that UA #1 needs to disclose
*: Content that UA #1 needs to protect

Figure 2: Deployment example #2

In the third example, User #1 connects UA #1 to a proxy server in a visited (potentially insecure) 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 performs access control based on the destination addresses of calls. User #1 only trusts Proxy A to route requests, not to inspect the message bodies the requests 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

2.2 Service Examples

We describe here several services that require end-to-middle security.

2.2.1 Logging Services for Instant Messages

Logging Services are provided by the archiving function, which is located in the proxy server, that logs the message content exchanged between UAs. The archiving function could be located at the originator network and/or the destination network. When the content of an instant message contains private information, UACs (UA Clients) encrypt the content for the UASs (UA Servers). The archiving function needs a way to log the content in a message body in bidirectional MESSAGE requests in such a way that the data is decipherable. The archiving function also needs a way to verify the data integrity of the content before logging.

This service might be deployed in financial or health care service provider's networks, where archiving communication is required by their security policies, as well as other networks.

2.2.2 Non-emergency Call Routing Based on the Location Object

The Location Object [6] includes private information as well as routing information for appropriate proxy servers. Some proxy

servers have the capability to provide location-based routing. When UAs want to employ location-based routing in non-emergency situations, the UAs need to connect to the proxy servers with such a capability and disclose the location object contained in the message body of the INVITE request, while protecting it from other proxy servers along the request path.

The Location Object also needs to be verified for integrity before location-based routing is applied. Sometimes the UAC want to also send the Location Object to the UASs. This is another good example of the need for a UAC to simultaneously send secure data to a proxy server and to the UAS.

2.2.3 User Authentication

2.2.3.1 User Authentication using the AIBs

The Authenticated Identity Bodies (AIBs) [7] is a digitally-signed data that is used as way to identify users. Proxy servers that need to authenticate a user verify the signature. When the originator needs anonymity, the user identity in the AIB is encrypted before being signed. Proxy servers that authenticate the user need to decrypt the body in order to view the user identity in the AIB. Such proxy servers can be located at adjacent and/or non-adjacent to the UA.

The AIB could be included in all request/response messages. The proxy server needs to view it in request messages in order to authenticate users. Another proxy server sometimes needs to view it in response messages for user authentication.

2.2.3.2 User Authentication in HTTP Digest Authentication

User authentication data for HTTP digest authentication includes two types of information; potentially private information, such as a user name, and information that can be used for "replay-attacks", such as the "response" parameter that is created by a calculation using a user's password. The user authentication data can be set only in a SIP header of request messages. This information needs to be transmitted securely to servers that authenticate users, located either adjacently and/or non-adjacently to the UA.

2.2.4 Media-related Services

Firewall traversal is an example of services based on media information typically in a message body, such as the Session Description Protocol (SDP). A firewall entity that supports the SIP protocol, or a midcom [8] agent co-located with a proxy server,

controls a firewall based on media information. The SDP includes the address and port information for media streams and/or key parameters for Secure RTP (SRTP) [9]. Critical information contained in SDP requires UAs to encrypt the SDP for recipient UAs. If the SDP is encrypted for end-to-end confidentiality, the proxy server operating as a midcom agent will have no way to provide firewall traversal as it can not inspect the SDP. Therefore, there is a need for proxy server to be able to decrypt the SDP, as well as to verify the integrity of the SDP.

[Note: The validity of the use case depends on which mechanism is selected for session policies [10] by the SIPPING WG. If the session policy mechanism would require that UAs disclose media information to the policy servers using out-of-band messages, such as OPTIONS request, end-to-middle security is not required for these use cases. If the session policy mechanism employs in-band messages in order for UAs to disclose media information to the policy servers co-located with a proxy server, end-to-middle security is required. As the mechanism proposes to place subset of SDP into the header to be viewed by proxy servers, such as addresses and port numbers of media streams, these information need to be secured from entities except the policy servers.]

3. Scope of End-to-Middle Security

End-to-middle security consists of user authentication, data integrity, and data confidentiality. However, this document only describes requirements for data confidentiality and data integrity, since authentication is covered by existing mechanisms such as HTTP digest authentication [2], S/MIME Cryptographic Message Syntax (CMS) SignedData body [11], or an AIB.

As for data integrity, the CMS SignedData body can be used for verification of the data integrity by any entities. The CMS SignedData body could be used for end-to-middle security at the same time for end-to-end security.

Although a proxy server is able to verify the integrity of the data, there is no way for UAs to request a selected proxy server to verify a message with the CMS SignedData body. Therefore 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.

4. 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.

4.1 General Requirements

The following are general requirements for end-to-middle confidentiality and integrity.

REQ-GEN-1: The solution SHOULD have little impact on the way a UA handles S/MIME-secured messages.

REQ-GEN-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.

REQ-GEN-3: It SHOULD have little impact on the standardized mechanism of proxy servers in terms of handling message bodies.

REQ-GEN-4: It SHOULD allow a UA to discover security policies of proxy servers. Security policies imply what data is needed to disclose and/or verify in a message.
This requirement is necessary when the UA does not know statically which proxy servers or domains need disclosing data and/or verification.

4.2 Requirements for End-to-Middle Confidentiality

REQ-CONF-1: The solution MUST be enable an encrypted data to be shared with the recipient UA and selected proxy servers, when a UA wants.

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

REQ-CONF-3: It SHOULD allow a UA to request selected proxy servers to view specific message bodies. The request itself SHOULD be secure.

REQ-CONF-4: It SHOULD allow a UA to request that the recipient UA disclose information to the proxy server, which requesting UA is disclosing the information to. The request itself SHOULD be secure.

4.3 Requirements for End-to-Middle Integrity

REQ-INT-1: The solution SHOULD work even when the SIP end-to-end integrity service is enabled.

REQ-INT-2: It SHOULD allow a UA to request selected proxy servers to verify specific message bodies. The request itself SHOULD be secure.

REQ-INT-3: 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. Security Considerations

This document describes the requirements for confidentiality and integrity between a UA and a proxy server. Although this document does not cover authentication, it is important in order to prevent attacks from malicious users and servers.

The end-to-middle security requires additional processing on message bodies, such as unpacking MIME structure, data decryption, and/or signature verification to proxy servers. Therefore the proxy servers that enable end-to-middle security are vulnerable to a Denial-of-Services attack. There is a threat model where a malicious user sends many complicated-MIME-structure messages to a proxy server, containing user authentication data obtained by eavesdropping. This attack will result in a slow down of the overall performance of these proxy servers. To prevent this attack, user authentication mechanism needs protection against replay attack. Or the user authentication always needs to be executed simultaneously with protection of data integrity. In order to prevent an attack, the following requirements should be satisfied.

- The solution MUST support mutual authentication, data confidentiality and data integrity protection between a UA and a proxy server.
- It SHOULD support protection against a replay attack for user authentication.
- It SHOULD simultaneously support user authentication and data integrity protection.

6. IANA Considerations

This document requires no additional considerations.

7. Changes

7.1 Changes from 02.txt

- Changed the text about the use case of SDP-based service in order to decrease the dependency on session policies discussion. The title was changed to "media-related service".

- Simplified the "Scope of End-to-Middle Security" section.
- Removed some of the text that described detailed information on mechanisms in the "Requirements for a Solution" section.
- Closed open issues as follows:
  - Deleted an open issue described in the "General Requirements" section, since it is no longer an issue. The issue was concerning the necessity for the proxy server to notify the UAS after receiving a response, which is not necessary, because proxy servers' security policies or services have no dependencies on the information in a response.
  - Deleted an open issue described in the "Requirements for End-to-Middle Confidentiality" section, since it is not an issue of requirements, but that of a mechanism.
- Changed the last item of the general requirements from proxy-driven to UA-driven.
- Deleted the text in the requirements that describes the relation between the requirements and the service examples.
- Added some text in the "Security Consideration" section.
- Many editorial correction.

7.2 Changes from 01.txt

- Extracted use cases from the Introduction section, and created a new section to describe the use cases in more detail. The use cases are also updated.
- Deleted a few "may" words from the "Problem with Existing Situations" section to avoid confusion with "MAY" as a key word.
- Added the relation between the requirements and the service examples.
- Deleted the redundant requirements for discovery of the targeted-middle. The requirement is described only in the "Generic Requirements", not in the "Requirements for End-to-Middle Confidentiality/Integrity".
- Changed the 4th requirement of end-to-middle confidentiality from "MUST" to "SHOULD".
- Changed the 3rd requirement of end-to-middle integrity from "MUST" to "SHOULD".
- Added some text about DoS attack prevention in the "Security Consideration" section.

7.3 Changes from 00.txt

- Reworked the subsections in Section 4 to clarify the objectives, separating end-to-middle confidentiality and integrity.

8. Acknowledgments

Thanks to Rohan Mahy and Cullen Jennings for their initial support of

this concept, and to Jon Peterson, Gonzalo Camarillo, Sean Olson, and Mark Baugher for their helpful comments.

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

[6] Cuellar, J., Morris, J., Mulligan, D., Peterson, J. and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.

[7] Peterson, J., "SIP Authenticated Identity Body (AIB) Format", draft-ietf-sip-authid-body-03.txt (work in progress), May 2004.

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

[9] Baugher, M., McGrew, D., Naslund, M., Carrara, E. and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[10] Rosenberg, J., "Requirements for Session Policy for the Session Initiation Protocol (SIP)", draft-ietf-sipping-session-policy-req-01 (work in progress), February 2004.

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

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

Disclaimer of Validity

Copyright Statement

Acknowledgment

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