SIPPING Working Group
Internet-Draft
Expires: April 14, 2005

M. Garcia-Martin
Nokia
G. Camarillo
Ericsson

October 14, 2004

Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP)
draft-ietf-sipping-uri-list-message-01.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies how to request a SIP URI-list service to send a copy of a MESSAGE to a set of destinations. The client sends a SIP MESSAGE request with a URI-list to the URI-list service, which sends a similar MESSAGE request to each of the URIs included in the list.

Table of Contents

1. Introduction

SIP [4] can carry instant messages in MESSAGE [7] requests. The Advanced Instant Messaging Requirements for SIP [11] mentions the need for sending a MESSAGE request to multiple recipients:

"REQ-GROUP-3: It MUST be possible for a user to send to an ad-hoc group, where the identities of the recipients are carried in the message itself."

One possibility to fulfill the above requirement is to establish a session of instant messages with an instant messaging conference server. While this option seems to be reasonable in many cases, in other situations the sending user just want to send a small pager-mode instant message to an ad-hoc group, without entering the burden of setting up a session. This document focuses on sending a pager-mode instant message to a number of intended recipients.

To meet the requirement with a pager-mode instant message, we allow SIP MESSAGE requests carry URI-lists in 'recipient-list- body parts, as specified in the framework for URI-list services [10]. A SIP URI-list service, which is a specialized application service, receives the request and sends a similar MESSAGE request to each of the URIs in the list. Each of these MESSAGE requests contains a copy of the body included in the original MESSAGE request.

The UAC (User Agent Client) that sends a MESSAGE request to a URI list service needs to be configured with the SIP URI of the service that provides the functionality. Discovering and provisioning of this URI to the UAC is outside the scope of this document.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

MESSAGE URI-list service: SIP application service that receives a MESSAGE request with a URI-list and sends a similar MESSAGE request to each URI in the list. In this context, similar indicates that some SIP header fields can change, but the MESSAGE URI-list service will not change the instant message payload. MESSAGE URI-list services behave effectively as specialised B2BUAs (Back-To-Back-User-Agents). A server providing MESSAGE URI-list services can also offer URI-list services for other methods, although this functionality is outside the scope of this document.

In this document we only discuss MESSAGE URI-list services.

Incoming MESSAGE request: A SIP MESSAGE request that a UAC creates and addresses to a MESSAGE URI-list service. Besides the regular instant message payload, an incoming MESSAGE request contains a URI-list.

Outgoing MESSAGE request: A SIP MESSAGE request that a MESSAGE URI-list service creates and addresses to a UAS (User Agent Server). It contains the regular instant message payload.

Intended recipient: The intended final recipient of the request to be generated by MESSAGE URI-list service.

3. Overview

A UAC creates a MESSAGE request that contains a multipart body including a list of URIs (intended recipients) and an instant message. The UAC sends this MESSAGE request to the MESSAGE URI-List service. On reception of this incoming MESSAGE request, the MESSAGE URI-list service creates a MESSAGE request per intended recipient (listed in the URI list) and copies the instant message payload to each of those MESSAGES. Then the MESSAGE URI-list service sends each of the created outgoing MESSAGE request to the respective receiver.

The mechanism reuses the XML format for representing resource lists [8] to include the list of intended recipients. We define an extension to that list to indicate the capacity of each resource, which can be To, Cc or Bcc (in an analogy to e-mail). The MESSAGE URI list service can include a resource list in the outgoing MESSAGE request that contain those resources tagged with a To or Cc capacities (and not Bcc capacities). This allows the creator of the incoming MESSAGE request to identify if a resource should be receiving a copy of the MESSAGE request as a capacity of recipient (to), carbon copy (cc) or blind carbon copy (bcc). It also allows some the intended recipients to reply to the initial sender and all the visible recipients of the MESSAGE request.

4. URI-List document

As described in the framework for URI-list services [10], specifications of individual URI-list services, like the MESSAGE URI-list service described here, need to specify a default format for 'recipient-list' bodies used within the particular service.

The default format for recipient-list bodies for MESSAGE URI-list services is the resource list document format [8] . UAs (User

Agents) and servers handling recipient-list bodies MUST support this format and MAY support other formats.

Nevertheless, the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) resource list document provides features, such as hierarchical lists and the ability to include entries by reference relative to the XCAP root URI, that are not needed by the MESSAGE URI-list service defined in this document, which only needs to transfer a flat list of URIs between a UA and the server. Therefore, when using the default resource list document, UAs SHOULD use flat lists (i.e., no hierarchical lists) and SHOULD NOT use <entry-ref> elements.

Section 4.1 defines an extension to the XML format for representing resource lists [8]. This extension allows to provide an 'capacity' attribute to a resource. UAs (User Agents) and servers handling 'recipient-list' bodies MUST support that extension.

A MESSAGE URI-list service receiving a URI-list with more information than what has just been described MAY discard all the extra information.

4.1 Extension to the resource lists data format

An extension to indicate the capacity of a resource is included. We define a new <capacity> element that can take the values "to", "cc" and "bcc". A "to" value indicates that the resource is considered the recipient of the MESSAGE request. A "cc" value indicates that the resource receives a carbon copy of the MESSAGE request. A "bcc" value indicates that the resource receives a blind carbon copy of the MESSAGE request. The default capacity value is "bcc", that is, the absence of a <capacity> element MUST be treated as if the capacity was set to "bcc".

The <capacity> element SHOULD be included as a child element of any of the elements included in the <list> element of a resource list.

Figure 1 describes the format of the capacity element. Implementations according to this specification MUST support this XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:capacity"
    xmlns="urn:ietf:params:xml:ns:capacity"
    xmlns:rls="urn:ietf:params:xml:ns:resource-lists"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

    <xs:annotation>
      <xs:documentation xml:lang="en">
         Adds the capacity element to a resource list.
      </xs:documentation>
    </xs:annotation>

    <xs:element name="capacity">
       <xs:simpleType>
          <xs:restriction base="xs:string">
             <xs:enumeration value="to"/>
             <xs:enumeration value="cc"/>
             <xs:enumeration value="bcc"/>
          </xs:restriction>
       </xs:simpleType>
    </xs:element>
</xs:schema>
```

Figure 1: Extension to the resource lists data format

4.2 URI-list example

Figure 2 shows an example of a flat list that follows the resource list data format. Each resource indicates the capacity of a resource.

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
          xmlns:cp="urn:ietf:params:xml:ns:capacity"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@example.com">
       <cp:capacity>to</cp:capacity>
    </entry>
    <entry uri="sip:joe@example.org">
       <cp:capacity>cc</cp:capacity>
    </entry>
    <entry uri="sip:ted@example.net">
       <cp:capacity>bcc</cp:capacity>
    </entry>
  </list>
</resource-lists>
```

Figure 2: URI-List

5. Procedures at the User Agent Client

A UAC that wants to create a multiple-recipient MESSAGE request MUST create a MESSAGE request according to RFC 3428 [7] Section 4. The UAC SHOULD add a body part, whose Content-Disposition type is "recipient-list", which contains a URI-list with the recipients of the MESSAGE. The URI-list MAY also include the "capacity" extension to the URI-list specified in Section 4.1.

Multiple-recipient MESSAGE requests contain a multipart body that contains the body carrying the list and the actual instant message payload. In some cases, the MESSAGE request may contain bodies other than the text and the list bodies (e.g., when the request is protected with S/MIME [9]).

Typically, the MESSAGE URI-list service will copy all the significant header fields in the outgoing MESSAGE request. However, there might be cases where the SIP UA wants the MESSAGE URI-list service to add a particular header field with a particular value, even if the header field wasn't present in the MESSAGE request sent by the UAC. In this case, the UAC MAY use the "?" mechanism described in Section 19.1.1 of RFC 3261 [4] to encode extra information in any URI in the list. However, the UAC MUST NOT use the special "body" hname (see Section 19.1.1 of RFC 3261 [4]) to encode a body, since the body is present in the MESSAGE request itself.

The following is an example of a URI that uses the "?" mechanism:

sip:bob@example.com?Accept-Contact=*%3bmobility%3d%22mobile%22

The previous URI requests the MESSAGE URI-list service to add the following header field to a MESSAGE request to be sent to bob@example.com:

Accept-Contact: *;mobility="mobile"

6. Procedures at the MESSAGE URI-List Service

On reception of a MESSAGE request with a URI-list, a MESSAGE URI-list service SHOULD answer to the UAC with a 202 Accepted response. Note that the status code in the response to the MESSAGE does not provide any information about whether or not the MESSAGEs generated by the URI-list service were successfully delivered to the URIs in the list. That is, a 202 Accepted means that the MESSAGE URI-list service has received the MESSAGE and that it will try to send a similar MESSAGE to the URIs in the list. Designing a mechanism to inform a client about the delivery status of an instant message is outside the scope of this document.

6.1 Determining the intended recipient

On reception of a MESSAGE request with a URI-list, a MESSAGE URI-list service SHOULD determine the list of intended recipients, by inspecting the URI list contained in the body. In case two of those URIs are equivalent (section 19.1.4 of RFC 3261 [4] defines equivalent URIs), the MESSAGE URI-list SHOULD consider a single intended recipient.

6.2 Creating an outgoing MESSAGE request

Since the MESSAGE URI-list behaves as a UAC for outgoing MESSAGE requests, for each of the intended recipients, the MESSAGE URI-list service creates a new MESSAGE request according to the procedures described in Section 4 of RFC 3428 [7] and the following procedures:

- A MESSAGE URI-list service MUST include a From header field whose value is the same as the From header field included in the incoming MESSAGE request, subject to the privacy requirements (see RFC 3323 [5] and RFC 3325 [6]) expressed in the incoming MESSAGE request. Note that this does not apply to the "tag" parameter.
- A MESSAGE URI-list service SHOULD generate a new To header field value set to the intended recipient URI. According to the procedures of RFC 3261 Section 8.1.1.1, this value should also be equal to the Request-URI of the outgoing MESSAGE request

- A MESSAGE URI-list service SHOULD create a new Call-ID header field value.
- If a P-Asserted-Identity header field was present in the incoming MESSAGE request and the request was received from a trusted source, as specified in RFC 3325 [6], and the first hop of the outgoing MESSAGE request is also trusted, a MESSAGE URI-list service MUST include a P-Asserted-Identity header field in the outgoing MESSAGE request with the same received value. However, if the first hop of the outgoing MESSAGE request is not trusted and the incoming MESSAGE request included a Privacy header field with a value different than 'none', the MESSAGE URI-list service MUST NOT include a P-Asserted-Identity header field in the outgoing MESSAGE request.
- If a MESSAGE URI-list service is able to assert the identity of a user (e.g., using HTTP Digest authentication scheme [3], S/MIME [9], etc.) and the service implements a mechanism where it can map that authentication scheme to a user's SIP or SIPS URI, and subject to the privacy requirements expressed in the incoming MESSAGE request (see RFC 3323 [5], the MESSAGE URI list MAY insert a P-Asserted-Identity header with the value of the user's asserted URI.
- If the incoming MESSAGE request contains an Authorization or Proxy-Authorization header fields whose realm is set to the MESSAGE URI-List server's realm, then the MESSAGE URI-List service SHOULD NOT copy it to the outgoing MESSAGE request; otherwise (i.e., if the Authorization or Proxy-Authorization header field of incoming MESSAGE request contains a different realm), the MESSAGE URI-List service MUST copy the value to the respective header field of the outgoing MESSAGE request.
- A MESSAGE URI-List service SHOULD create a separate count for the CSeq header field of the outgoing MESSAGE request.
- A MESSAGE URI-list service SHOULD initialize the value of the Max-Forward header field of the outgoing MESSAGE request.
- A MESSAGE URI-list service MUST include its own value in the Via header field.
- A MESSAGE URI-List service SHOULD include any other header field expressed with the "?" mechanism described in Section 19.1.1 of RFC 3261 [4] and encoded in the intended recipient URI of the URI list.
- A MESSAGE URI-List service SHOULD preserve to the outgoing MESSAGE request any other header field not explicitly indicated in the above paragraphs.

6.3 Composing bodies in the outgoing MESSAGE request

When creating the body of each of the outgoing MESSAGE requests, the MESSAGE URI-list service tries to keep the relevant bodies of the incoming MESSAGE request and copies them to the outgoing MESSAGE

request. The following guidelines are provided:

- o A MESSAGE request received at a MESSAGE URI-list service can contain one or more security bodies (e.g., S/MIME [9]) encrypted with the public key of the MESSAGE URI-list service. These bodies are deemed to be read by the URI-list service rather than the recipient of the outgoing MESSAGE request (which will not be able to decrypt them). Therefore, a MESSAGE URI-list service MUST NOT copy any security body (such as an S/MIME [9] encrypted body) addressed to the MESSAGE URI-list service to the outgoing MESSAGE request. This includes bodies encrypted with the public key of the URI-list service.
- o If the URI-list of the incoming MESSAGE request include resources tagged with the <capacity> value of "to" or "cc", the URI-list service SHOULD include a URI-list in each of the outgoing MESSAGE requests. The format of such list SHOULD BE according to the XML format for representing resource lists [8] and the capacity extension specified in Section 4.1. This resource list MUST contain those elements categorized with the "to" or "cc" capacity and MUST NOT contain those resources categorized are "bcc" or lacking the capacity element.
- o If a MESSAGE URI-list service includes a URI-list in an outgoing MESSAGE request, it SHOULD use S/MIME [9] to encrypt the URI-list with the public key of the receiver.
- o The incoming MESSAGE request typically contains a URI-list body or reference [10] with the actual list of recipients. Section 6.2 contains procedures that determine when the MESSAGE URI-list service should include a URI-list body in the outgoing MESSAGE request.
- o The MESSAGE URI-list service SHOULD copy all the rest of the message bodies (e.g., text messages, images, etc.) to the outgoing MESSAGE request.
- o If there is only one body left, the MESSAGE URI-list service MUST remove the multipart/mixed wrapper in the outgoing MESSAGE request.

The rest of the MESSAGE request corresponding to a given URI in the list MUST be created following the rules in Section 19.1.5 "Forming Requests from a URI" of RFC 3261 [4]. In particular, Section 19.1.5 of RFC 3261 [4] states:

"An implementation SHOULD treat the presence of any headers or body parts in the URI as a desire to include them in the message, and choose to honor the request on a per-component basis."

SIP allows to append a "method" parameter to a URI. Therefore, it is legitimate that an the "uri" attribute of the "entry" element in the XCAP resource list contains a "method" parameter. MESSAGE URI-list

services MUST generate only MESSAGE requests, regardless of the "method" parameter that the URIs in the list indicate. Effectively, MESSAGE URI-list services MUST ignore the "method" parameter in each of the URIs present in the URI-list.

7. Procedures at the UAS

A UAS (in this specification, also known as intended recipient UAS) that receives a MESSAGE request from the URI-list service behaves as specified in RFC 3428 [7] Section 7.

If the UAS supports this specification and the MESSAGE request contains a body with a Content-Disposition header set to 'recipient-list', then the UAS will be able to determine who are the other intended recipients (visible) of the MESSAGE request. This allows the user to create a reply request (e.g., MESSAGE, INVITE) to the sender and the rest of the visible recipients.

8. Examples

Figure 5 shows an example of operation. A SIP UAC issuer sends a MESSAGE request. The MESSAGE URI-list service answers with a 202 Accepted message and sends a MESSAGE request to each of the intended recipients.

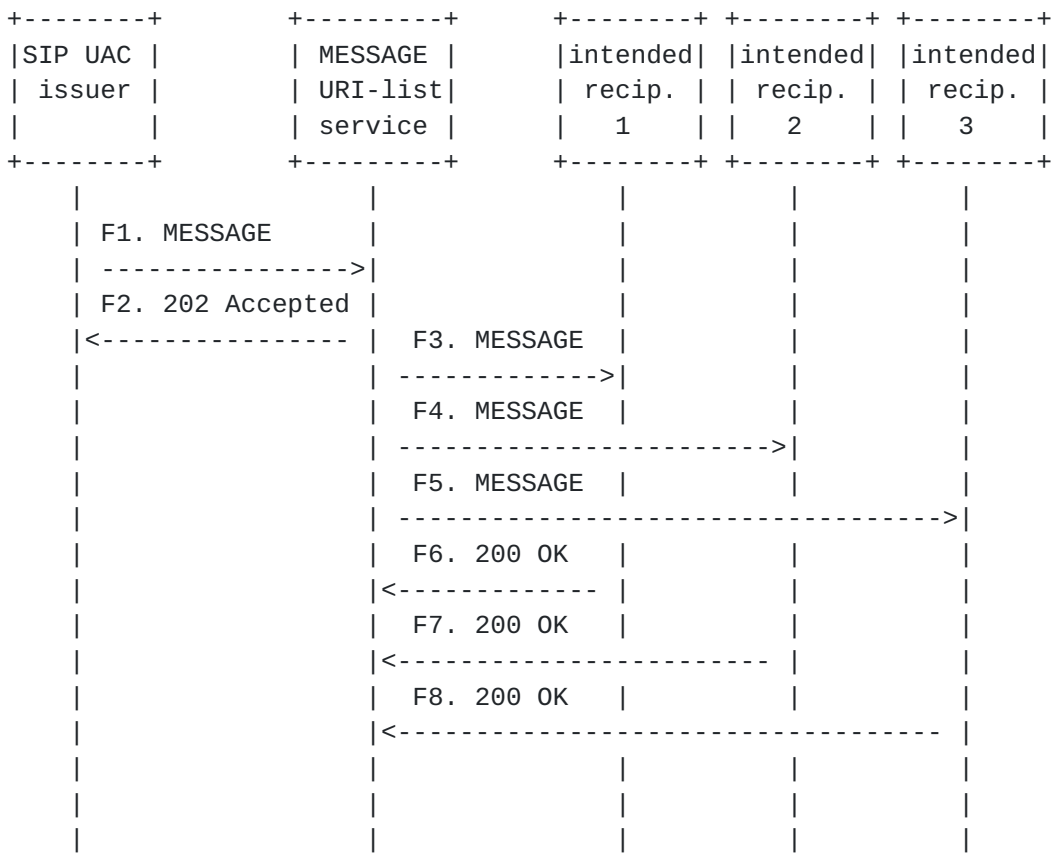

Figure 5: Example of operation

The MESSAGE request F1 is as follows:

```
MESSAGE sip:list-service.example.com SIP/2.0
Via: SIP/2.0/TCP uac.example.com
    ;branch=z9hG4bKhjhs8ass83
Max-Forwards: 70
To: MESSAGE URI-List Service <sip:list-service.example.com>
From: Carol <sip:carol@example.com>;tag=32331
Call-ID: d432fa84b4c76e66710
CSeq: 1 MESSAGE
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: 501

--boundary1
Content-Type: text/plain

Hello World!

--boundary1
Content-Type: application/resource-lists+xml
Content-Disposition: recipient-list

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
          xmlns:cp="urn:ietf:params:xml:ns:capacity"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@example.com">
       <cp:capacity>to</cp:capacity>
    </entry>
    <entry uri="sip:joe@example.org">
       <cp:capacity>cc</cp:capacity>
    </entry>
    <entry uri="sip:ted@example.net">
       <cp:capacity>bcc</cp:capacity>
    </entry>
  </list>
</resource-lists>
--boundary1--
```

Messages F4, F4 and F5 are similar in nature. Especially the bodies are exactly the same for all of them, since they include the instant message payload and a URI-list that contains the resources tagged with the "to" and "cc " capacity. We show an example of F3:

```
MESSAGE sip:bill@example.com SIP/2.0
Via: SIP/2.0/TCP list-service.example.com
    ;branch=z9hG4bKhjhs8as34sc
Max-Forwards: 70
To: <sip:bill@example.com>
From: Carol <sip:carol@uac.example.com>;tag=210342
Call-ID: 39s02sdsl20d9sj2l
CSeq: 1 MESSAGE
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: 501

--boundary1
Content-Type: text/plain

Hello World!

--boundary1
Content-Type: application/resource-lists+xml
Content-Disposition: recipient-list

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
          xmlns:cp="urn:ietf:params:xml:ns:capacity"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@example.com">
       <cp:capacity>to</cp:capacity>
    </entry>
    <entry uri="sip:joe@example.org">
       <cp:capacity>cc</cp:capacity>
    </entry>
  </list>
</resource-lists>
--boundary1--
```

9. Security Considerations

The Framework and Security Considerations for SIP URI-List Services [10] discusses issues related to SIP URI-list services. Implementations of MESSAGE URI-list services MUST follow the security-related rules in the framework for URI-list services [10]. These rules include mandatory authentication and authorization of clients, and opt-in lists.

If the contents of the instant message needs to be kept private, the user agent client SHOULD use S/MIME [9] to prevent a third party from viewing this information. In this case, the user agent client SHOULD

encrypt the instant message body with a content encryption key. Then, for each receiver in the list, the UAC SHOULD encrypt the content encryption key with the public key of the receiver, and attach it to the MESSAGE request.

10. Acknowledgements

Duncan Mills supported the idea of having 1 to n MESSAGEs. Ben Campbell, Paul Kyzivat, Cullen Jennings, and Jonathan Rosenberg provided helpful comments.

11. Change control

11.1 Changes from draft-ietf-sipping-uri-list-message-00.txt

Revision of the treatment of headers the MESSAGE URI-list service, on a header by header basis.

Added an overview section.

Added functionality that allows the sender of the incoming MESSAGE request to tag each of the intended recipients with the "to", "cc", or "bcc" capacity. If there are resources tagged as "to" or "cc", the URI-list service will include a URI-list in each of the outgoing MESSAGE request including those resources.

Procedures at the UAS included.

Better example including a flow.

11.2 Changes from draft-ietf-sipping-message-exploder-00.txt to draft-ietf-sipping-uri-list-message-00.txt

Clarified that the MESSAGE exploder should not distribute a body that has been encrypted with the public key of the exploder. The exception is the URI-list, which can be distributed by the exploder, providing that is encrypted with the public key of the receiver.

The security considerations section describes how to encrypt the list and how to encrypt the instant message payload.

Terminology aligned with the requirements and the framework for URI-list services (e.g., the term "exploder" has been deprecated).

11.3 Changes from draft-garcia-simple-message-exploder-00.txt to draft-garcia-sipping-message-exploder-00.txt

The MESSAGE exploder may or may not copy the URI-list body to the

outgoing MESSAGE request. This allows to extend the mechanism with a Reply-to-all feature.

It is clarified that the MESSAGE exploder must not include a list in the outgoing MESSAGE requests. This avoids loops or requires a MESSAGE exploder functionality in the next hop.

The MESSAGE exploder must remove the multipart/mixed wrapper if there is only one body left in the outgoing MESSAGE request.

Filename changed due to focus on the SIPPING WG.

12. References

12.1 Normative References

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

[2] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.

[3] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[4] 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.

[5] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[6] Jennings, C., Peterson, J. and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

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

[8] Rosenberg, J., "Extensible Markup Language (XML) Formats for Representing Resource Lists", draft-ietf-simple-xcap-list-usage-03 (work in progress), July 2004.

[9] Ramsdell, B., "S/MIME Version 3.1 Message Specification", draft-ietf-smime-rfc2633bis-09 (work in progress), April 2004.

[10] Camarillo, G., "Requirements and Framework for Session Initiation Protocol (SIP)Uniform Resource Identifier (URI)-List Services", draft-ietf-sipping-uri-services-00 (work in progress), July 2004.

12.2 Informational References

[11] Rosenberg, J., "Advanced Instant Messaging Requirements for the Session Initiation Protocol (SIP)", draft-rosenberg-simple-messaging-requirements-01 (work in progress), February 2004.

[12] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.

Authors' Addresses

Miguel A. Garcia-Martin
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland

EMail: miguel.an.garcia@nokia.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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