SIP WG
Internet-Draft
Expires: August 23, 2006

J. Elwell
Siemens plc
February 19, 2006

Connected Identity in the Session Initiation Protocol (SIP)
draft-elwell-sip-connected-identity-01.txt

Status of this Memo

Copyright Notice

Abstract

Because of retargeting of a dialog-forming request, the UAS can have a different identity from that in the To header. This document provides a means for that UA to supply its identity to the peer UA by means of a request in the reverse direction and for that identity to be signed by an authentication service. The same mechanism can be used to indicate a change of identity during a dialog, e.g., because of some action in a PSTN behind a gateway.

This work is being discussed on the sip@ietf.org mailing list.

Table of Contents

1. Conventions and Definitions

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 [2].

2. Introduction

SIP[1]initiates sessions but it also provides information on the identities of the parties at both ends of a session. Users need this information to help determine how to deal with communications initiated by SIP. As a call proceeds, these identities may change. This can happen for many reasons: calls are forwarded, calls are parked and picked up, calls are transferred, calls are queued to be picked up by a pool of agents, and so on. This can have impact on the identity of the party that answers a call. It can also cause the identity of a party to change during an established call.

This document extends the use of the From header field to allow it to convey "connected identity" information in either direction within the context of an existing INVITE-initiated dialog.

The provision of "response identity" for requests outside the context of an INVITE-initiated dialog is outside the scope of this document.

3. Existing mechanisms for conveying identity in the context of a call

When establishing a call and its session, the SIP From header field in the INVITE request provides a means for conveying the identity of the caller from the User Agent Client (UAC) to the User Agent Server (UAS), thereby allowing the caller's identity to be presented to the callee. There is no corresponding mechanism specified for conveying the identity of the callee from the UAS to the UAC, to allow the callee's identity to be presented to the caller. The identity of the callee is normally expected to be the identity placed in the To header field of the INVITE request, but often this expectation is not met because a different party answers the call, e.g., because of call forwarding.

> History information [5] gathered during the routing of a request and returned in the response can provide additional information to the UAC. However, this does not necessarily clearly indicate the AoR of the UAS. Also the methods described in Section 4 for authentication do not apply to history information, which relies instead on hop-by-hop security and transitive trust.

The Reply-To header field has its own meaning and cannot be relied on in all circumstances.
The Contact header field provides a contact URI, which may not reveal the identity (Address of Record) of the user on whose behalf the response is sent.

Parties involved in a call can change owing to actions such as call transfer. If such actions are achieved by issuing a new INVITE request (with a Replaces header field) between the two UAs that are to be involved in the re-arranged call, the SIP From header field in the INVITE request can provide identity information in one direction, but again there is no mechanism for conveying identity information in the reverse direction.

However, call re-arrangements are not always carried out using a new INVITE request. Sometimes a B2BUA performs call re-arrangements using third party call control (3PCC) techniques. With such techniques the UA involved in the original call and still involved in the re-arranged call receives only a re-INVITE or UPDATE request in the context of the original dialog between that UA and the B2BUA. This forces the UA to re-negotiate the session with the new remote party, but introduces a need to convey the identity of the new remote party to the UA. Because there is no new INVITE request (outside the context of the existing dialog), techniques applicable to new calls do not apply.

Another case where call re-arrangements are not carried out using a new INVITE request is where one of the UAs is a gateway to a PSTN and a call re-arrangement such as call transfer has occurred within the PSTN. The gateway then has a need to convey the identity of the new party within the PSTN to the remote UA. This needs to be done within the context of the existing dialog between the gateway and the remote UA. In this case there is probably not even any need to re-negotiate the session - the only requirement is to update the identity information.

4. Existing methods for providing authenticated identity information

Because the From header field in a request is generated by the UAC itself it can be subject to falsification. SIP has several means of providing cryptographic authentication of a request's source identity.

One such means for requests is HTTP-based digest authentication, as specified in [1]. Although a UAS can require digest authentication, it is not usually feasible between an arbitrary pair of UAs because of reliance on a shared secret. To achieve scalability, methods

based on public key cryptography are essential.

Another method is specified in [6], and is applicable to responses as well as requests. This requires a UA to have a private key and associated certificate in order to sign an Authenticated Identity Body (AIB) in the request or response. However, this has seen little deployment, since the public key infrastructures needed to support private keys and certificates in every UA are not generally available.

A third method is specified in [3]. For signature this uses a private key and certificate associated with the domain indicated in the From header URI. An authentication service, typically located at the outbound proxy, authenticates the UAC by some means, using digest authentication for example, and then inserts an Identity header and an Identity-Info header in the forwarded request. The Identity header contains a signature using the domain's private key and the Identity-Info header references the corresponding certificate.

5. Overview of solution

A mid-dialog request is used to provide connected identity. The UAC for that request inserts its identity in the From header field of the request and the Identity header can be used to provide authentication.

A request in the opposite direction to the INVITE request prior to or at the time the call is answered can indicate the identity of the alerting or answering party. A request in the same direction as the INVITE request prior to answer can indicate a change of calling party. A request in either direction after answer can indicate a change of party. In all cases a dialog (early or confirmed) must be established before such a request can be sent.

> Note that it might also be possible to provide a means of indicating the identity of the alerting or answering party in the response to the INVITE request. However, at present the problem of authenticating a response is still subject to study. In the absence of a solution to the response identity problem, the simple solution of using a request in the opposite direction to the INVITE request is sufficient.

This solution involves changing the URI (not the tags) in the To and From header fields of mid-dialog requests and their responses, compared with the corresponding values in the dialog forming request and response. Changing the To and From header field URIs was contemplated in Section 12.2.1.1 of RFC 3261, which says "Usage of

the URI from the To and From fields in the original request within subsequent requests is done for backwards compatibility with RFC 2543, which used the URI for dialog identification. In this specification, only the tags are used for dialog identification. It is expected that mandatory reflection of the original To and From URI in mid-dialog requests will be deprecated in a subsequent revision of this specification."

This document therefore deprecates mandatory reflection of the original To and From URIs in mid-dialog requests and their responses. It is assumed that deployed proxies will already be able to tolerate a change of URI, since this has been expected for a considerable time. To cater for any UAs that are not able to tolerate a change of URI, a new option tag "dialogUriChange" is introduced for providing a positive indication of support in the Supported header field.

> OPEN ISSUE. Should this be extended to allow a URI in the To header field of a response to change compared with the To header field in a request? This could convey a connected identity in a response to an INVITE request, but it would not be authenticated. Authentication would have to rely on transitive trust, which might be feasible in a closed environment where the sips URI scheme is used.

6. Behaviour

6.1. Behaviour of a UA that issues an INVITE request

When issuing an INVITE request, a UA that supports changes of URI in the From and To headers during a dialog SHOULD include the dialogUriChange option tag in the Supported header field.

After a dialog has been formed (after receipt of a reliable response to the INVITE request), if the dialogUriChange option tag has been received in a Supported header field and if the identity associated with the UA changes, the UA SHOULD issue a request on the same dialog containing the new identity in the URI of the From header field. Unless there is a need to invoke any other method, the UPDATE method [4] SHOULD be used if supported by the peer UA. If the UPDATE method is not supported by the peer UA, the re-INVITE method SHOULD be used. but this necessitates waiting until the dialog is confirmed.

> OPEN ISSUE. RFC 3311 talks about circumstances in which an UPDATE request cannot contain an SDP offer, yet does not explicitly talk about the use of UPDATE requests without SDP offers. It needs to be resolved whether an UPDATE request can be used in order to convey just a revised URI in the From header field even though no

SDP offer needs to be sent at the time. If the outcome is that an UPDATE request must contain an SDP offer, then SDP offer will need to be included in the UPDATE request.

After sending a request with a revised URI in the From header field, the UA SHALL send the same URI in the From header field of any future requests on the same dialog, unless the identity changes again.

If the UA has received a request from the peer UA in which the From header field URI differs from the To header field in the last request the UA sent on that dialog, the UA MAY indicate the revised identity to the user. In addition the UA SHOULD use this revised URI in the To header field of any future requests it sends on the same dialog.

OPEN ISSUE. Is this a good idea to change the content of the To header field URI on subsequent requests? How should a UAS for a subsequent request react it if receives the wrong value in the To header field URI, e.g., it receives the old URI when it has already sent an UPDATE request containing a new From header field URI?

6.2. Behaviour of a UA that receives an INVITE request

After receiving an INVITE request, a UA that supports changes of URI in the From and To headers during a dialog SHOULD include the dialogUriChange option tag in the Supported header field of any dialog-forming response.

After a dialog has been formed (after sending a reliable response to the INVITE request, i.e., a 2xx response or a reliable 1xx response), if the dialogUriChange option tag has been received in a Supported header field and if the identity associated with the UA differs from that received in the To header field of the INVITE request, the UA SHOULD issue a request on the same dialog containing the new identity in the URI of the From header field. Unless there is a need to invoke any other method, the UPDATE method SHOULD be used.

After sending a request with a revised URI in the From header field, the UA SHALL send the same URI in the From header field of any future requests on the same dialog, unless the identity changes again.

If the UA has received a request from the peer UA in which the From header field URI differs from that received in the previous request on that dialog, the UA MAY indicate the revised identity to the user. In addition the UA SHOULD use this revised URI in the To header field of any future requests it sends on the same dialog.

6.3. Behaviour of a UA during an established INVITE-initiated dialog

If the dialogUriChange option tag has been received in a Supported header field and if the identity associated with the UA differs from that received in the To header field of the INVITE request, the UA SHOULD issue a request on the same dialog containing the new identity in the URI of the From header field. Unless there is a need to invoke any other method, the UPDATE method SHOULD be used.

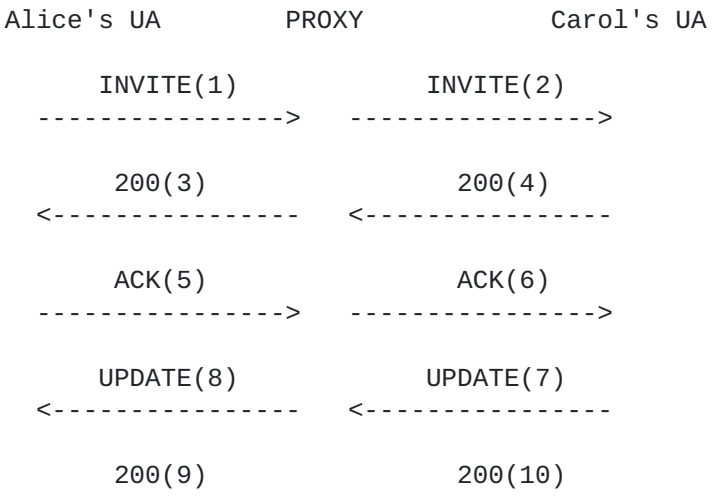
After sending a request with a revised URI in the From header field, the UA SHALL send the same URI in the From header field of any future requests on the same dialog, unless the identity changes again.

If the UA has received a request from the peer UA in which the From header field URI differs from that received in the previous request on that dialog, the UA MAY indicate the revised identity to the user. In addition the UA SHOULD use this revised URI in the To header field of any future requests it sends on the same dialog.

7. Examples

7.1. Sending connected identity after answering a call

In this example Carol's UA has been reached by retargeting at the proxy and thus her identity (AoR) is not equal to that in the To header field of the received INVITE request (Bob). Carol's UA therefore conveys it identity in the From header field of an UPDATE request. The proxy also provides an authentication service and therefore adds Identity and Identity-Info header field to the UPDATE request.



```
     ---------------->   ---------------->

   INVITE (1):

   INVITE sip:Bob@example.com SIP/2.0
   From: <sip:Alice@example.com>;tag=1
   To: <sip:Bob@example.com>
   Call-ID: 12345600@example.com
   CSeq: 1 INVITE
   Supported: dialogUriChange
   Contact: <sip:Alice@ua1.example.com>
   etc.

   INVITE(2):

   INVITE sip:Carol@ua2.example.com SIP/2.0
   From: <sip:Alice@example.com>;tag=1
   To: <sip:Bob@example.com>
   Call-ID: 12345600@example.com
   CSeq: 1 INVITE
   Supported: dialogUriChange
   Contact: <sip:Alice@ua1.example.com>
   Identity: "dKJ97..."
   Identity-Info: <https://example.com/cert>;alg=rsa-sha1
   etc.

   200(3):

   SIP/2.0 200 OK
   From: <sip:Alice@example.com>;tag=1
   To: <sip:Bob@example.com>;tag=2
   Call-ID: 12345600@example.com
   CSeq: 1 INVITE
   Supported: dialogUriChange
   Contact: <sip:Carol@ua2.example.com>
   etc.

   200(4):

   SIP/2.0 200 OK
   From: <sip:Alice@example.com>;tag=1
   To: <sip:Bob@example.com>;tag=2
   Call-ID: 12345600@example.com
   CSeq: 1 INVITE
   Supported: dialogUriChange
   Contact: <sip:Carol@ua2.example.com>
   etc.
```

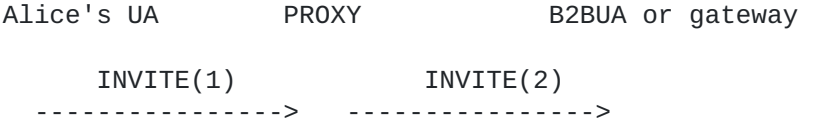
```
ACK (5):

ACK sip:Bob@example.com SIP/2.0
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>
Call-ID: 12345600@example.com
CSeq: 1 ACK
etc.

ACK (6):

ACK sip:Bob@example.com SIP/2.0
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>
Call-ID: 12345600@example.com
CSeq: 1 ACK
etc.

UPDATE (7):

UPDATE sip:Alice@ua1.example.com SIP/2.0
From: <sip:Carol@example.com>;tag=2
To: <sip:Alice@example.com>;tag=1
Call-ID: 12345600@example.com
CSeq: 2 UPDATE
Contact: <sip:Carol@ua2.example.com>
etc.
```

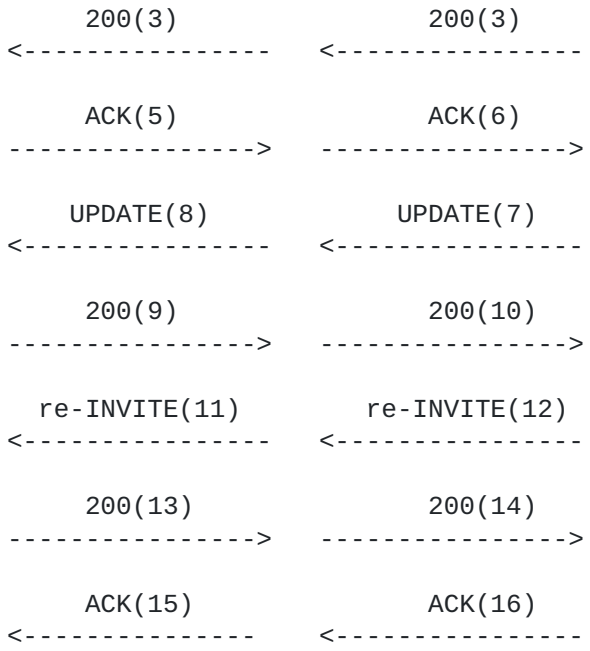
Note that the URI in the From header differs from that in the To header in the INVITE request/response. However, the tag is the same as that in the INVITE response.

```
UPDATE (8):

UPDATE sip:Alice@ua1.example.com SIP/2.0
From: <sip:Carol@example.com>;tag=2
To: <sip:Alice@example.com>;tag=1
Call-ID: 12345600@example.com
CSeq: 2 UPDATE
Contact: <sip:Carol@ua2.example.com>
Identity: "cdKJH43..."
Identity-Info: <https://example.com/cert>;alg=rsa-sha1
etc.

200(9):

SIP/2.0 200 OK
From: <sip:Carol@example.com>;tag=2
To: <sip:Alice@example.com>;tag=1
Call-ID: 12345600@example.com
CSeq: 2 UPDATE
Contact: <sip:Alice@ua1.example.com>
etc.

200(10):

SIP/2.0 200 OK
From: <sip:Carol@example.com>;tag=2
To: <sip:Alice@example.com>;tag=1
Call-ID: 12345600@example.com
CSeq: 2 UPDATE
Contact: <sip:Alice@ua1.example.com>
etc.
```

7.2. Sending revised connected identity during a call

In this example a call is established between Alice and Bob, where Bob lies behind a B2BUA or gateway to a PSTN. Then call transfer occurs in the B2BUA or PSTN, such that Alice becomes connected to Carol, and a re-INVITE request is issued allowing the session to be renegotiated. The B2BUA (or an entity behind it) or the gateway provides the authentication service and thus generates the Identity header in the re-INVITE request to provide authentication of Carol's identity.





INVITE (1):

```
INVITE sip:Bob@example.com SIP/2.0
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>
Call-ID: 12345600@example.com
CSeq: 1 INVITE
Supported: dialogUriChange
Contact: <sip:Alice@ua1.example.com>
etc.
```

INVITE(2):

```
INVITE sip:Bob@example.com SIP/2.0
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>
Call-ID: 12345600@example.com
CSeq: 1 INVITE
Supported: dialogUriChange
Contact: <sip:Alice@ua1.example.com>
Identity: "dKJ97..."
Identity-Info: <https://example.com/cert>;alg=rsa-sha1
etc.
```

200(3):

```
SIP/2.0 200 OK
```

```
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>;tag=2
Call-ID: 12345600@example.com
CSeq: 1 INVITE
Supported: dialogUriChange
Contact: <sip:Bob@ua2.example.com>
etc.

200(4):

SIP/2.0 200 OK
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>;tag=2
Call-ID: 12345600@example.com
CSeq: 1 INVITE
Supported: dialogUriChange
Contact: <sip:Bob@ua2.example.com>
etc.

ACK (5):

ACK sip:Bob@example.com SIP/2.0
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>
Call-ID: 12345600@example.com
CSeq: 1 ACK
etc.

ACK (6):

ACK sip:Bob@example.com SIP/2.0
From: <sip:Alice@example.com>;tag=1
To: <sip:Bob@example.com>
Call-ID: 12345600@example.com
CSeq: 1 ACK
etc.

UPDATE (7):

UPDATE sip:Alice@ua1.example.com SIP/2.0
From: <sip:Bob@example.com>;tag=2
To: <sip:Alice@example.com>;tag=1
Call-ID: 12345600@example.com
CSeq: 2 UPDATE
Contact: <sip:Bob@ua2.example.com>
Identity: "cdKJH43..."
Identity-Info: <https://example.com/cert>;alg=rsa-sha1
etc.
```

```
   UPDATE (8):

   UPDATE sip:Alice@ua1.example.com SIP/2.0
   From: <sip:Bob@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 2 UPDATE
   Contact: <sip:Bob@ua2.example.com>
   Identity: "cdKJH43..."
   Identity-Info: <https://example.com/cert>;alg=rsa-sha1
   etc.

   200(9):

   SIP/2.0 200 OK
   From: <sip:Bob@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 2 UPDATE
   Contact: <sip:Alice@ua1.example.com>
   etc.

   200(10):

   SIP/2.0 200 OK
   From: <sip:Bob@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 2 UPDATE
   Contact: <sip:Alice@ua1.example.com>
   etc.

   re-INVITE (11):

   INVITE sip:Alice@ua1.example.com SIP/2.0
   From: <sip:Carol@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 3 INVITE
   Contact: <sip:Carol@ua3.example.com>
   Identity: "ecdFG24..."
   Identity-Info: <https://example.com/cert>;alg=rsa-sha1
   etc.

   re-INVITE (12):

   INVITE sip:Alice@ua1.example.com SIP/2.0
   From: <sip:Carol@example.com>;tag=2
```

```
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 3 INVITE
   Contact: <sip:Carol@ua3.example.com>
   Identity: "ecdFG24..."
   Identity-Info: <https://example.com/cert>;alg=rsa-sha1
   etc.

   200(13):

   SIP/2.0 200 OK
   From: <sip:Carol@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 3 INVITE
   Contact: <sip:Alice@ua1.example.com>
   etc.

   200(14):

   SIP/2.0 200 OK
   From: <sip:Carol@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 3 INVITE
   Contact: <sip:Alice@ua1.example.com>
   etc.

   ACK (15):

   ACK sip:Alice@ua1.example.com SIP/2.0
   From: <sip:Carol@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 3 ACK
   etc.

   ACK (16):

   ACK sip:Alice@ua1.example.com SIP/2.0
   From: <sip:Carol@example.com>;tag=2
   To: <sip:Alice@example.com>;tag=1
   Call-ID: 12345600@example.com
   CSeq: 3 ACK
   etc.
```

8. IANA considerations

This specification registers a new SIP option tag, as per the guidelines in Section 27.1 of RFC 3261.

Name: dialogUriChange

Description: This option tag is used to indicate that a UA supports changes to URIs in From and To header fields during a dialog.

9. Security considerations

[3] discusses security considerations relating to the Identity header in some detail. Those same considerations apply when using the Identity header to authenticate a connected identity in the From header URI of a mid-dialog request.

A received From header field in a mid-dialog request that is not accompanied by a valid Identity header field (or other means of authentication) cannot be trusted (except in very closed environments) and should be treated in a similar way to a From header field in a dialog-initiating request that is not backed up by a valid Identity header field.

A signed connected identity in a mid-dialog request (URI in the From header field accompanied by a valid Identity header field) provides information about the peer UA in a dialog. In the case of the UA that was the UAS in the dialog-forming request, this identity is not necessarily the same as that in the To header field of the dialog-forming request. This is because of retargeting during the routing of the dialog-forming request. A signed connected identity says nothing about the legitimacy of such retargeting, but merely reflects the result of that retargeting.

Likewise, when a signed connected identity indicates a change of identity during a dialog, it conveys no information about the reason for such change of identity or its legitimacy.

Use of the sips URI scheme can minimise the chances of attacks in which inappropriate connected identity information is sent, either at call establishment time or during a call.

Privacy may be required by the user of a connected UA. To achieve privacy the UA MUST either decline to change the URI in the From header field of a mid-dialog request or populate it in the way described in [3] when anonymity is required.

10. Acknowledgments

Thanks to Francois Audet, Frank Derks, Steffen Fries, Cullen Jennings and Jon Peterson for providing valuable comments.

11. References

11.1. Normative References

[1] 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.

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

[3] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

[4] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3893, September 2002.

[5] Barnes, M., "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 3893, November 2005.

11.2. Informative References

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

[7] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

Author's Address

John Elwell
Siemens plc
Technology Drive
Beeston, Nottingham NG9 1LA
UK

Phone: +44 115 943 4989
Email: john.elwell@siemens.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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