Internet Engineering Task Force
Internet Draft

J. Elwell
Siemens

draft-elwell-sip-connected-identity-00.txt
Expires: April 2006 October 2005

Connected Identity in the Session Initiation Protocol (SIP)

Status of this Memo

Copyright Notice

Abstract

This provides a means of providing a signed connected identity in SIP. 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 TDM network behind a gateway.

Table of Contents

1 Introduction

An important aspect of the Session Initiation Protocol (SIP) [SIP] is the ability to convey to a User Agent Server (UAS) as part of a request an identity associated with the User Agent Client (UAC) that generated that request. Although the From header performs this function, this header is generated by the UAC client itself and therefore can be subject to falsification. SIP has several means of providing cryptographic authentication of a request's source identity.

One such means is digest authentication, as specified in [SIP]. 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 [AIB]. This requires a UAC to have a private key and associated certificate in order to sign an Authenticated Identity Body (AIB) in the request. 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 [Identity]. For signature this uses a private key and certificate associated with the domain indicated in the From header URI. 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 contains the corresponding certificate.

Some have argued that there is a need to provide the UAS's identity to the UAC in a response. This reflects the fact that a request can

be retargeted for various reasons before reaching the UAS, and the UAS identity may differ from that in the To header field of the request. Since the URI in the To header field of the response must equal that in the request, the To header field is not suitable for providing the UAS's identity. Furthermore, any such identity would need to be authenticated in some way. [AIB] provides for this, since the AIB contains a From field, the URI of which identifies the source of the response, not the source of the request (thus differing from the From header field of the message itself).

However, there is no equivalent of [Identity] for responses. This reflects the difficulty in the final proxy challenging the UAS to

provide digest authentication, in many circumstances a necessary step in adding an Identity header or equivalent.

For dialog-forming requests (such as INVITE), the identity of the UAS is particularly important, since that UA will remain as part of the dialog until the dialog terminates. The identity of that UA often differs from that in the To header of the dialog-forming request owing to retargeting. If the identity is made available to the UAC, the dialog can be terminated early if the UAS identity is not acceptable. For requests that are not part of a new or existing dialog, it can be argued that authenticated UAS identity is less important since any damage arising from reaching an unacceptable UAS has already been done.

Furthermore, the identity of a UA involved in a dialog can change during the course of that dialog. One example is where a UA is a PSTN/ISDN gateway and transfer occurs within the PSTN/ISDN. It is not only important to send the identity of the PSTN/ISDN party to the peer UA on dialog establishment, but also to send an updated identity if the party changes. A similar situation can arise with B2BUAs that perform third party call control operations.

This document therefore proposes a solution to the general problem of connected identity, which is the provision of the identity of the UAS in a dialog-forming request to the UAC of that request and the provision of a revised identity to the peer UA if identity changes during a dialog. In each case the UA whose identity is provided is known as the connected UA and that UA is known as the connected identity.

2 Overview of proposed solution

Because of difficulties providing authenticated identity in the response to a dialog forming request, a request in the reverse direction is used to provide authenticated identity of the UAS in the dialog forming request, i.e., the identity of the connected UA. Likewise, if the identity of either UA changes during the lifetime of the dialog, the new connected identity can be provided in a request issued by that UA. In either case the signalling path must pass through an authentication service acting on behalf of the connected UA, and therefore the proxy concerned must record-route. Note that this may involve different authentication services at the two ends of the dialog.

The URI in the From header field of a request in the backward direction (opposite direction to the dialog-forming request) is unsuitable for providing connected identity, since the URI in the From header field must always be the same as the URI in the To header field of the dialog-forming request (see 12.2.1.1/[SIP]). Because of

retargeting of the dialog-forming request, the connected identity can differ from the URI in the From header of a request in the reverse direction. Likewise the URI in the From header of a request in the forward direction (the same direction as the dialog-forming request) is unsuitable for providing a changed connected identity, because the From header field must not change.

Therefore a new Header known as Connected-URI is defined to convey the connected identity URI. A UA wishing to indicate its connected identity may include a Connected-URI header field in a request. This can be any request issued by a UA in the context of an early or established dialog. An UPDATE request [UPDATE] can be sent for this purpose if there is no other purpose for sending a request at this time.

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 Connected-URI 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 when sending Connected-URI is

sent.

An authentication service on the path of a request containing a Connected-URI header field may add a Connected-Identity header field to sign the request on behalf of the domain part of the URI indicated in the Connected-URI header field. This is similar to an Identity header but the signature covers also the Connected-URI header field and certifies that the UAC has credentials that allow it to register as a contact for that URI. To ascertain this the authentication service would normally challenge the UAC to provide digest authentication, unless TLS is used and the UA has already been authenticated on that connection, e.g., by means of a certificate during the TLS handshake or a shared secret used to respond to a challenge at the application layer. The authentication service also adds an Identity-Info header to provide information about the certificate needed to verify the signature.

A proxy that provides an authentication service may also add a Connected-URI header field if not already present in a request or replace an existing one.

A UAS receiving a request containing a Connected-URI header field and a valid Connected-Identity header field may use the connected identity for any purpose, such as passing to an application or displaying. A UAS receiving a request containing a Connected-URI header field but no Connected-Identity header field should either discard the connected identity or use it with caution.

OPEN ISSUE: Should we also consider the possibility of including a Connected-URI header in a response? An authentication service would be able to add a Connected-Identity header only if has some means of authenticating the UAS. It cannot challenge the UAS, so it would have to rely on other means, e.g., challenging an earlier request on the same TLS transport).

This document also specifies a new option tag, connectedID, to indicate support for or requirement for connected identity.

OPEN ISSUE. Is this option tag worthwhile? Use of it a Require header field to force the sending of a Connected-URI header field in a reverse request is not particularly secure, since it does not fall within the signature of the Identity header and could be removed by

an attacker. Also, even if the UAS provides a Connected-URI header field in a reverse request, there is no guarantee that an authentication service will be available or be prepared to add a signature in the form of a Connected-Identity header field.

3 Connected UA behaviour

3.1 Connected UA at dialog establishing time

When a dialog is established, the connected UA is the UA that acts as UAS for the dialog establishing request and returns a 1xx (not 100) or 2xx response.

The behaviour below relies on a connected UA knowing its connected URI, i.e., its AoR. Some UAs register as contacts for multiple AoRs. Provided a UA has registered a different contact URI for each AoR it registers for, then it can associate an incoming request with a particular AoR by examining the Request-URI.

After sending the first reliable response (1xx or 2xx) to a dialog forming request, a UA MAY send its identity (the connected identity) if the dialog-forming request contained a Supported header field with the connectedID option tag and MUST send its connected identity if the dialog-forming request contained a REQUIRED header field with the connectedID option tag.

To send its connected identity the UA MUST include a Connected-URI header field in a mid-dialog request, e.g., an UPDATE request or a (re-)INVITE request. If there is no other reason to send a mid-dialog request, the UA SHOULD send an UPDATE request for this specific purpose. The UA MUST accurately populate the Connected-URI header field with a value corresponding to an identity that it believes it is authorized to claim. It MUST set the URI portion of the header to

match a SIP, SIPS or TEL URI AoR which it is authorized to use in the domain (including anonymous URIs, as described in [Privacy]).

Note that [Identity] defines a number of new 4xx response codes, and these in general are applicable also to connected identity. If a UA supports these response codes, it will be able to respond intelligently to Identity-based error conditions.

3.2 Identity change during an established dialog

An identity change can occur at a gateway as a result of action in the legacy network beyond the gateway, e.g., call transfer. During an established dialog the gateway can receive updated identity information from the legacy network that prompts it to adopt a revised identity within the SIP network. This can apply to either the UAC or the UAS of the original dialog establishing request.

If during a dialog the identity of a UA changes from that which it indicated previously in the From header or Connected-URI header of a request, the UA MAY send its revised identity if it has received from the peer UA a Supported header field with the connectedID option tag and MUST send its revised identity if it has received from the peer UA a Required header field with the connectedID option tag. To send its revised identity the UA MUST included a Connected-URI header field in a mid-dialog request as specified in 3.1.

In this case it will frequently be the case that the UA provides its own authentication service and will therefore add a Connected-Identity header field too. Authentication is on the basis of trusting the identity received from the legacy network.

4 Authentication service behaviour

The authentication service described here is an extension to that described in [Identity] except that it authenticates the connected identity. As stated in [Identity], the authentication service role can be instantiated by a proxy server or a user agent. Any entity that instantiates the authentication service role MUST possess the private key of a domain certificate, and MUST be capable of authenticating one or more SIP users that can register in that domain. Commonly, this role will be instantiated by a proxy server, since these entities are more likely to have a static hostname, hold a corresponding certificate, and have access to SIP registrar capabilities that allow them to authenticate users in their domain.

A proxy wishing to perform the role of authentication service for a dialog must record-route during dialog establishment, so that mid-dialog requests pass through it.

The behaviour described below applies only to requests containing a

Connected-URI header field received in the context of an early or established dialog. Otherwise the authentication service behaviour of [Identity] is applicable (i.e., an Identity header will be added, if applicable).

A SIP entity that acts as an authentication service MUST add a Date header field to SIP requests if one is not already present. Similarly, an authentication service MUST add a Content-Length header field to SIP requests if one is not already present; this can help the verifier to double-check that it is hashing exactly as many bytes of message-body as the authentication service when it verifies the message.

An entity instantiating the authentication service role performs the following steps, in order, to generate a Connected-Identity header for a SIP request:

Step 1: The authentication service MUST extract the identity of the sender from the request. The authentication service takes this value from the Connected-URI header field; this AoR will be referred to here as the 'identity field'. If the identity field contains a SIP or SIPS URI, the authentication service MUST extract the hostname portion of the identity field and compare it to the domain(s) for which it is responsible. If the identity field uses the TEL URI scheme, the policy of the authentication service determines whether or not it is responsible for this identity; see Section 12 of [Identity] for more information. If the authentication service is not responsible for the identity in question, it SHOULD process and forward the request normally, but it MUST NOT add a Connected-Identity header; see below for more information on authentication service handling of an existing Connected-Identity header.

Step 2: The authentication service MUST determine whether or not the sender of the request is authorized to claim the identity given in the connected identity field. In order to do so, the authentication service MUST first authenticate the sender of the message. Authentication and authorization considerations are as described in authentication service behaviour step 2 in [Identity]. If the authentication service is unable to authorise the connected identity it MUST reject the request with a 403 response.

Step 3: The authentication service SHOULD ensure that any pre-existing Date header in the request is accurate, in accordance with authentication service behaviour step 3 in [Identity].

Step 4: The authentication service MUST form the connected identity signature and add a Connected-Identity header to the request containing this signature. After the Connected-Identity header has

been added to the request, the authentication service MUST also add an Identity-Info header. The Identity-Info header contains a URI from which its certificate can be acquired. Details on the generation of both of these headers are provided in section 6.

Finally, the authentication service MUST forward the message normally.

5 Verifier behaviour

This document extends the logical role for SIP entities called a 'verifier', as introduced in [Identity]. When a verifier receives a SIP request in the context of an early or established dialog and containing a Connected-Identity header, it may inspect the signature to verify the identity of the sender of the message. Typically, the results of a verification are provided as input to an authorization process which is outside the scope of this document. If neither an Identity header field nor a Connected-Identity header field is present in a request, and one is required by local policy (for example, based on a per-sending-domain policy, or a per-sending-user policy), then a 428 'Use Identity Header' response MUST be sent, which should be interpreted in this case as 'Use Identity header or Connected-Identity header, as appropriate'.

In order to verify the identity of the sender of a message containing a Connected-Identity header field together with Connected-URI and Identity-Info header fields, an entity acting as a verifier MUST perform the following steps, in the order here specified.

Step 1: The verifier MUST acquire the certificate for the signing domain as described in verifier behaviour step 1 in [Identity].

Step 2: The verifier MUST compare the identity of the signer with the domain portion of the URI in the Connected-URI header field using the method described in verifier behaviour step 2 of [Identity].

Step 3: The verifier MUST verify the signature in the Connected-Identity header field, following the procedures for generating the hashed digest- string described in Section 6. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 428 'Invalid Identity Header' response MUST be returned.

Step 4: The verifier MUST validate the Date, Contact and Call-ID headers the manner described in Section 14.1 of [IDENTITY]; recipients that wish to verify Connected-Identity signatures MUST

support all of the operations described there.

The verifier function is normally located at the UAS of a mid-dialog request. The UA SHOULD render the connected identity and its validity to the user through an appropriate user interface or to an application. A UA MAY also make use of a Connected-URI header field that is not accompanied by a Connected-Identity header field, i.e., an unsigned connected identity, in which case the UA SHOULD distinguish unsigned connected identities from signed connected identities when rendering the information to a user or application.

In order to have the possibility of receiving connected identity, a UA MUST include the connectedID option tag in a Supported or Required header field in a SIP message towards the peer UA, e.g., in the dialog-forming request. Use of this option tag in a Required header will cause the request to fail if connected identity is not supported and will cause the Connected-URI to be returned in a request in the reverse direction if connected identity is supported at the UAS. However, although the return of a Connected-URI header can be forced using this mechanism, this does not guarantee that it will be accompanied by a Connected-Identity header field, which will depend on the presence of an authentication service on the path and the ability of the authentication service to authenticate the sender.

6 Header syntax

This document specifies two new SIP headers: Connected-URI and Connected-Identity. Each of these headers can appear only once in a SIP message. The grammar for these two headers is:

```
Connected-URI = "Connected-URI" HCOLON ( name-addr / addr-spec )

Connected-Identity = "Identity" HCOLON signed-connected-id-digest

signed-connected-id-digest = LDQUOT 32LHEX RDQUOT
```

The signed-connected-id-digest is a signed hash of a canonical string generated from certain components of a SIP request. It is identical to signed-identity-digest in [Identity] except that in the canonical string instead of the addr-spec of the From header field the addr-spec of the Connected-URI header field MUST be used.

This document adds the following entries to Table 2 of [SIP]:

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Connected-URI	R	r	o	o	-	o	o	-

SUB	NOT	REF	INF	UPD	PRA
-	o	o	o	o	o

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Connected-Identity	R	a	o	o	-	o	o	-

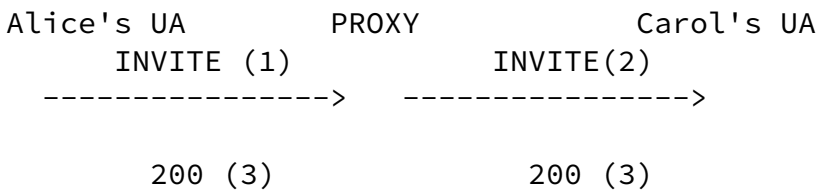
SUB	NOT	REF	INF	UPD	PRA
-	o	o	o	o	o

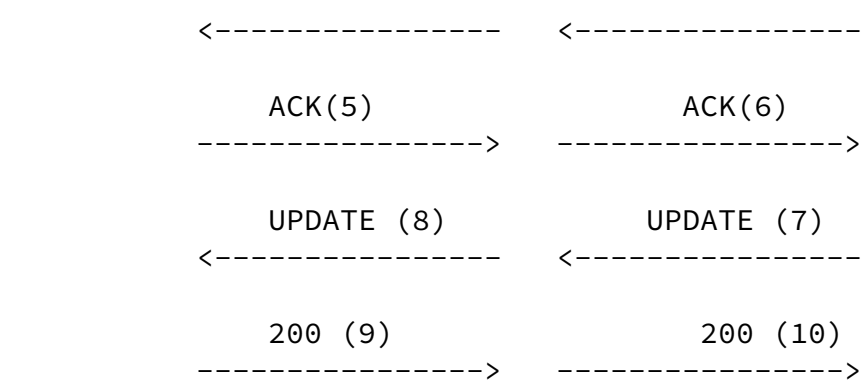
Note, in the table above, only methods that can be used in the context of an early or established dialog are applicable. The CANCEL method is excluded for reasons given in [Identity].

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 places a Connected-URI header field in an UPDATE request. The proxy also provides an authentication service and therefore adds a Connected-Identity header field and an Identity-Info header field to the UPDATE request.





INVITE (1) and INVITE (2)
These include either:
  Require: connectedID
or
  Supported: connectedID

UPDATE (7)

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

UPDATE (8)

```
  UPDATE sip:ua1@example.com SIP/2.0
  From: <sip:Bob@example.com>;tag=2
  To: <sip:Alice@example.com>;tag=3
  Call-ID: 12345600@example.com
  CSeq: 1 UPDATE
  Connected-URI: <sip:Carol@example.com>
  Conncected-Identity: "dKJ97..."
  Identity-Info: <https://example.com/cert>;alg=rsa-sha1
```

8 IANA considerations

This document requests changes to the header fields and option tags registries within the SIP parameters IANA registry.

8.1 Header field names

This document specifies two new SIP headers: Connected-URI and Connected-Identity. Their syntax is given in section 6. These headers are defined by the following information, which is to be added to the header sub-registry under http://www.iana.org/assignments/sip-parameters.

Header Name: Connected-URI
Compact Form: N/A
Header Name: Connected-Identity
Compact Form: N/A

8.2 SIP option tag

This specification registers a new SIP option tag, as per the guidelines in Section 27.1 of RFC 3261.
Name: connectedID
Description: This option tag is used to identify connected identity and the Connected-URI and Connected-Identity header fields. When used in a Supported header, it indicates support for receiving these header fields and acts as a request to provide this information. When used in a Required header it indicates that the request concerned should be rejected if connected identity information cannot be provided.

9 Security Considerations

[Identity] discusses security considerations relating to the Identity header in some detail. Essentially those same considerations apply to the Connected-Identity header.

A received Connected-URI header field that is not accompanied by a valid Connected-Identity header field cannot be trusted (except in very closed environments) and should be treated in a similar way to a From header field that is not backed up by a valid Identity header field.

A signed connected identity (Connected-URI header field accompanied by a valid Connected-Identity 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 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 of its legitimacy.

Privacy may be required by the user of a connected UA. To achieve privacy the UA MUST either decline to send the Connected-URI header field or populate it in the way described in [IDENTITY] for the From header field when anonymity is required. Note that if a Require header field has been received with the connectedID option tag, accepting the request but declining to send the Connected-URI header field is not an option, and therefore the UA MUST either decline the request or populate the Connected-URI header field anonymously.

10 Acknowledgements

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

11 Author's Addresses

John Elwell
Siemens Communications
Technology Drive
Beeston
Nottingham, UK, NG9 1LA
email: john.elwell@siemens.com

12 Normative References

[SIP] J. Rosenberg, H. Schulzrinne, et al., "SIP: Session initiation protocol", RFC 3261.

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

[Identity] J. Peterson, C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-

ietf-sip-identity-05 (work in progress).

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

[UPDATE] J. Rosenberg, "The Session Initiation Protocol (SIP) UPDATE method", RFC 3311.

Intellectual Property Statement

Disclaimer of Validity

Acknowledgement

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

13 Appendix - Rejected Alternatives (temporary 🙈 to be removed)

The following alternative solutions were considered and rejected.

13.1 Changing the From header to reflect connected identity

[SIP] disallows any change to the From and To header fields during the course of a dialog, since these header fields (along with the Call-Id header field) provide unique identification for a dialog. Although the tag parameters in the From and To header fields are in fact sufficient for dialog identification purposes, for backward compatibility with RFC 2543 changes to the URIs in these header fields are prohibited. It is probable that [SIP]-compliant implementations may break if a URI changes during a dialog. The community has already rejected this approach.

13.2 Conveying the connected identity URI in a body

This might have the advantage that the existing Identity header could be used, but this is contrary to the semantics of the Identity header.

13.3 Conveying the connected identity URI and the connected identity signature in the same header field.

This has the minor disadvantage that the syntax of the header field would differ from that of the Identity header field.

13.4 Reuse of the Identity header for signing connected identity

The signature in the Identity header basically authenticates the identity in the From header and would not be able to authenticate a different identity (e.g., in a Connected-URI header).

13.5 Response identity

Although the proposed solution can under some circumstances provide connected identity in a response, a general solution to response identity is not possible because of the inability to challenge a response to obtain authentication.

13.6 Establishment of a new dialog using Replaces

The UA wishing to convey its identity and being unable to do so in the From header field of a request on the existing INVITE-initiated dialog could send a new INVITE request containing a Replaces header field indicating replacement of the existing dialog at the UAS. The From header field of that request would contain the correct identity and could be signed by means of an Identity header in the normal way.

There is no normative specification at present covering the fate of the existing session when an existing INVITE-initiated dialog is replaced by a new dialog between the same two UAs. It would be undesirable to interrupt an ongoing session just because the dialog needs to be replaced, particularly if this means the calculation of new security keys for media.

The Replaces header is not defined for use in a SUBSCRIBE request, so this technique would not be applicable to SUBSCRIBE-initiated dialogs. It is not clear whether this would matter.

This technique would be unsuitable for use on an early dialog, since the replaced dialog might by-pass any proxy that was supervising the early dialog and might wish to cancel it under certain circumstances or take account of any non 2xx final response.

This technique would involve additional SIP messages (5, including the BYE request and response on the replaced dialog, rather than 2).