

The Session Initiation Protocol (SIP) Referred-By Mechanism

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2004).

Abstract

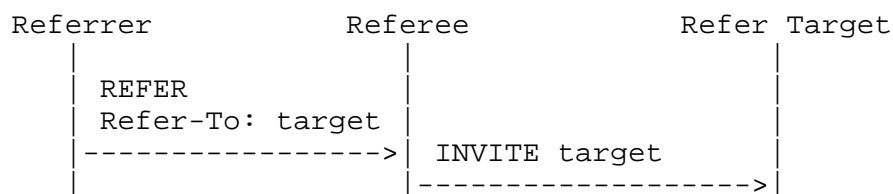
The Session Initiation Protocol (SIP) REFER method provides a mechanism where one party (the referrer) gives a second party (the referee) an arbitrary URI to reference. If that URI is a SIP URI, the referee will send a SIP request, often an INVITE, to that URI (the refer target). This document extends the REFER method, allowing the referrer to provide information about the REFER request to the refer target using the referee as an intermediary. This information includes the identity of the referrer and the URI to which the referrer referred. The mechanism utilizes S/MIME to help protect this information from a malicious intermediary. This protection is optional, but a recipient may refuse to accept a request unless it is present.

Table of Contents

1.	Overview	2
1.1.	Requirements Notation	3
2.	The Referred-By Mechanism	3
2.1.	Referrer Behavior	4
2.2.	Referee Behavior	4
2.3.	Refer Target Behavior	5
3.	The Referred-By Header Field	6
4.	The Referred-By Token	7
4.1.	Refer Target Inspection of a Referred-By Token	8
5.	The 429 Provide Referrer Identity Error Response	8
6.	Security Considerations	8
6.1.	Identifying the Referee in the Referred-by Token	10
7.	Examples	11
7.1.	Basic REFER	11
7.2.	Insecure REFER	14
7.3.	Requiring Referrer Identity	14
7.4.	Nested REFER	18
8.	IANA Considerations	23
9.	Contributors	23
10.	References	23
10.1.	Normative References	23
10.2.	Informative References	24
11.	Author's Address	24
12.	Full Copyright Statement	25

1. Overview

The SIP REFER method [2] provides a mechanism where one party (the referrer) provides a second party (the referee) with an arbitrary URI to reference. If that URI is a SIP URI, the referee will send a SIP request, often an INVITE, to that URI (the refer target). Nothing provided in [2] distinguishes this referenced request from any other request the referee might have sent to the refer target.



There are applications of REFER, such as call transfer [8], where it is desirable to provide the refer target with particular information about the referrer and the REFER request itself. This information may include, but is not limited to, the referrer's identity, the referred to URI, and the time of the referral. The refer target can use this information when deciding whether to admit the referenced request. This document defines one set of mechanisms to provide that information.

All of the mechanisms in this document involve placing information in the REFER request that the referee copies into the referenced request. This necessarily establishes the referee as an eavesdropper and places the referee in a position to launch man-in-the-middle attacks on that information.

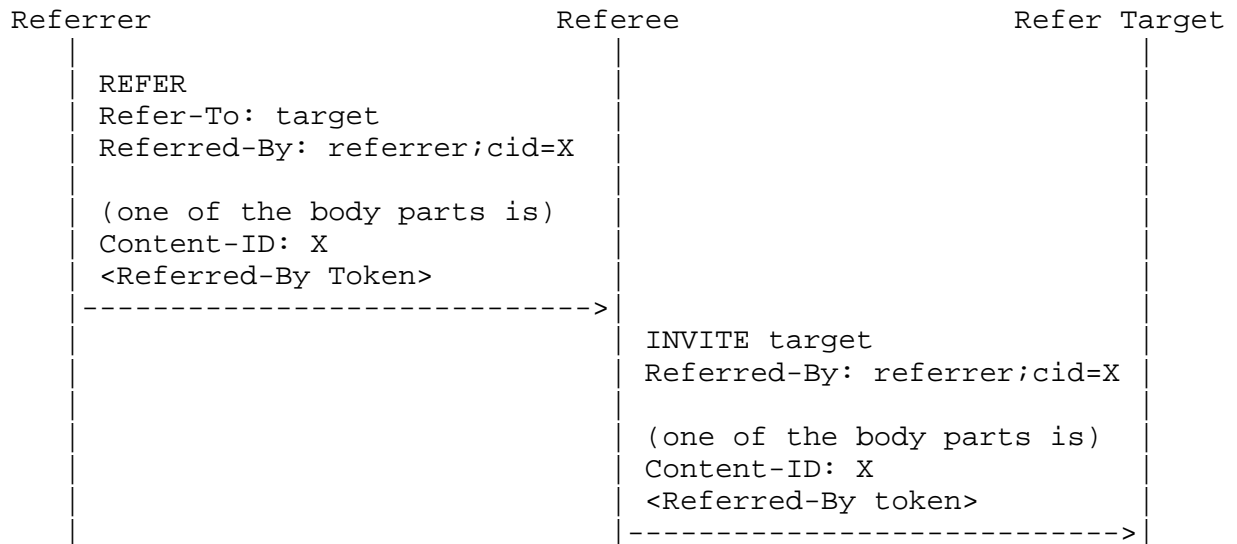
At the simplest level, this document defines a mechanism for carrying the referrer's identity, expressed as a SIP URI in a new header: Referred-By. The refer target can use that information, even if it has not been protected from the referee, at the perils and with the limitations documented here. The document proceeds to define an S/MIME based mechanism for expressing the identity of the referrer and capturing other information about the REFER request, allowing the refer target to detect tampering (and other undesirable behaviors) by the referee.

1.1. Requirements Notation

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 BCP 14, RFC 2119 [1].

2. The Referred-By Mechanism

The following figure summarizes how Referred-By information is carried to the Refer Target. The Referrer provides a Referred-By header with its SIP address-of-record, optionally associating an S/MIME protected token reflecting the identity of the referrer and the details of the REFER request. The Referee copies this header and the token, if provided, into the triggered request (shown here as an INVITE).



2.1. Referrer Behavior

A UA sending a REFER request (a referrer) MAY provide a Referred-By header field value in the request. A REFER request MUST NOT contain more than one Referred-By header field value.

A referrer MAY include a Referred-By token in a REFER request. A REFER request containing a Referred-By token MUST contain a Referred-By header field value with a cid parameter value equal to the Content-ID of the body part containing the token.

The referrer will receive a NOTIFY with a message/sipfrag [4] body indicating a final response of 429 "Provide Referrer Identity" to the referenced request if the refer target requires a valid Referred-By token to accept the request. This can occur when either no token is provided or a provided token is invalid.

The referrer will receive a 429 "Provide Referrer Identity" response to the REFER if the referee requires a Referred-By token to be present in order to accept the REFER.

If a referrer wishes to re-attempt to refer a referee after receiving a 429 response or a NOTIFY containing a 429, it MAY submit a new REFER request containing a Referred-By token.

2.2. Referee Behavior

A UA accepting a REFER request (a referee) to a SIP URI (using either the sip: or sips: scheme) MUST copy any Referred-By header field value and token into the referenced request without modification.

A referee MAY reject a REFER request that does not contain a Referred-By token with a 429 "Provide Referrer Identity" response. A referee SHOULD NOT reject a request that contains a Referred-By token encrypted to a key it does not possess simply because it cannot decrypt the token. (One scenario where such rejection would be appropriate is when the referee is attempting to remain anonymous (see Section 6.1).) Note that per [3], the referee should still be able to verify the signature of such an encrypted token.

A referee SHOULD present the same identity to the referrer and the refer target.

2.3. Refer Target Behavior

A UA receiving a non-REFER SIP request MAY inspect the request for a Referred-By header field and token.

If a Referred-By header field value is not present, this UA cannot distinguish this request from any other the UA acting as the referee might have sent. Thus, the UA would apply exactly the admissions policies and processing described in [5] to the request.

If a Referred-By header field value is present, the receiving UA can consider itself a refer target and MAY apply additional admission policies based on the contents of the Referred-By header field and token.

The referee is in a position to modify the contents of the Referred-By header field value, or falsely provide one even if no REFER actually exists. If such behavior could affect admission policy (including influencing the agent's user by rendering misleading content), the refer target SHOULD require that a valid Referred-By token be present.

The refer target MAY reject a request if no Referred-By token is present or if the token is stale using the 429 "Provide Referrer Identity" error response defined in Section 5. The 428 error response from [7] is not appropriate for this purpose - it is needed for the refer target to request an authentication token from the referee.

If no Referred-By token is present, the refer target MAY proceed with processing the request. If the agent provides any information from the Referred-By header to its user as part of processing the request, it MUST notify the user that the information is suspect.

The refer target MUST reject an otherwise well-formed request with an invalid Referred-By token (see Section 4) with a 429 error response.

3. The Referred-By Header Field

Referred-By is a request header field as defined by [5]. It can appear in any request. It carries a SIP URI representing the identity of the referrer and, optionally, the Content-ID of a body part (the Referred-By token) that provides a more secure statement of that identity.

```

Referred-By = ("Referred-By" / "b") HCOLON referrer-uri
              *( SEMI (referredby-id-param / generic-param) )

referrer-uri = ( name-addr / addr-spec )

referredby-id-param = "cid" EQUAL sip-clean-msg-id

sip-clean-msg-id = LDQUOTE dot-atom "@" (dot-atom / host) RDQUOTE

dot-atom = atom *( "." atom )

atom      = 1*( alphanum / "-" / "!" / "%" / "*" /
               "_" / "+" / "'" / "\"" / "~" )

```

Since the Content-ID appears as a SIP header parameter value which must conform to the expansion of the gen-value defined in [5], this grammar produces values in the intersection of the expansions of gen-value and msg-id from [9]. The double-quotes surrounding the sip-clean-msg-id MUST be replaced with left and right angle brackets to derive the Content-ID used in the message's MIME body. For example,

```

Referred-By: sip:r@ref.example;cid="2UWQFN309shb3@ref.example"
    indicates the token is in the body part containing

```

```

Content-ID: <2UWQFN309shb3@ref.example>

```

If the referrer-uri contains a comma, question mark, or semicolon, (for example, if it contains URI parameters) the URI MUST be enclosed in angle brackets (< and >). Any URI parameters are contained within these brackets. If the URI is not enclosed in angle brackets, any semicolon-delimited parameters are header-parameters, not URI parameters.

The Referred-By header field MAY appear in any SIP request, but is meaningless for ACK and CANCEL. Proxies do not need to be able to read Referred-By header field values and MUST NOT remove or modify them.

The following row should be interpreted as if it appeared in Table 3 of RFC 3261.

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Referred-By	R	-	o	-	o	o	o	

4. The Referred-By Token

The Referred-By token is an Authenticated Identity Body as defined by [3]. This body part MUST be identified with a MIME [6] Content-ID: field.

The sipfrag inside a Referred-By token MUST contain copies of the Refer-To, Referred-By, and Date header fields from the REFER request.

The token SHOULD NOT contain the Call-ID header field from the REFER request as that information is not useful to the refer target and may even be an information leak. The token SHOULD NOT contain the From header field from the REFER request since the identity being claimed is represented in the Referred-By header field.

The token MAY contain the To header field from the REFER request, but it SHOULD NOT be included unless the referrer has cryptographically identified the referee. Some ways this authentication can be achieved include inspecting the certificates used in a TLS association between the referrer and the referee or encrypting the Refer-To header in the REFER request using the S/MIME encryption techniques detailed in [5].

When inspecting the certificates used to establish TLS associations, the identity asserted in the token's To header field URI is compared to the subjectAltNames from the referee's certificate. The sip and sips URI schemes MUST be treated as equivalent for this comparison. If the URI is an exact match, confidence in the authentication is high and the To header field MAY be added to the token. If the certificate subjects contain only a hostname matching the hostname portion of the URI, an application level warning SHOULD be issued to the referrer agent's user seeking that user's consent before including the To header field in the token.

Including the To header field in the token significantly strengthens the claim being asserted by the token, but may have privacy implications as discussed in Section 6.1.

Additional header fields and body parts MAY be included in the token.

As described in [3], a Referred-By token MAY be encrypted as well as signed. The subjectAltName of the certificate used for these operations SHOULD exactly match the identity claimed in the referrer-uri in the Referred-By header field in the token.

4.1. Refer Target Inspection of a Referred-By Token

A refer target MUST treat a Referred-By token with an invalid signature as an invalid token. A target SHOULD treat a token with an aged Date header field value as invalid.

A target SHOULD verify that the request it receives matches the reference in the Refer-To header field in the token. This verification SHOULD include at least the request method and any indicated end-to-end header field values. Note that the URI in the Refer-To header field may not match the request URI in the received request due to request re-targeting between the referee and the refer target.

The target SHOULD verify that the identity in the Referred-By header field in the token exactly matches the SubjectAltName from the signing certificate, reporting discrepancies to its user as described in [3].

If the token contains a To header field, the target SHOULD verify that the identity it expresses matches the referrer. One way of verifying this is to exactly match the identity in the token's To header field with the subjectAltName of the certificate used by the referee to sign the aib protecting the request itself. The 428 response defined in [7] can be used to request such an aib if one is not already present.

5. The 429 Provide Referrer Identity Error Response

The 429 client error response code is used by a refer target to indicate that the referee must provide a valid Referred-By token. As discussed in the behavior section, the referee will forward this error response to the referrer in a NOTIFY as the result of the REFER. The suggested text phrase for the 429 error response is "Provide Referrer Identity".

6. Security Considerations

The mechanism defined in this specification relies on an intermediary (the referee) to forward information from the referrer to the refer target. This necessarily establishes the referee as an eavesdropper of that information and positions him perfectly to launch man-in-the-middle attacks using the mechanism.

A SIP proxy is similarly positioned. Protecting SIP messaging from malicious proxy implementations is discussed in [5]. In contrast to a proxy, the referee's agent is an endpoint. Proxies will typically be managed and monitored by service providers. Malicious behavior by a proxy is more likely to be noticed and result in negative repercussions for the provider than malicious behavior by an endpoint would be. The behavior of an endpoint can be entirely under the control of a single user. Thus, it is more feasible for an endpoint acting as referee to behave maliciously than it is for a proxy being operated by a service provider.

This specification uses an S/MIME based mechanism to enable the refer target to detect manipulation of the Referred-By information by the referee. Use of this protection is optional! The community has asserted that there are systems where trust in the validity of this information is either not important or can be established through other means. Any implementation choosing not to use this optional mechanism needs to provide its own defense to the following risks:

- o The Referred-By information is highly likely to influence request admission policy. For instance, it may be displayed to the user of the agent with a "This call was transferred to you by X. Accept?" prompt. A malicious referee can unduly influence that policy decision by providing falsified referred-by information. This includes falsely claiming to have been referred in the first place. (The S/MIME mechanism protects the information with a signature, hampering the referee's ability to inject or modify information without knowing the key used for that signature.)
- o A referee is by definition an eavesdropper of the referred-by information. Parts of that information may be sensitive. (The S/MIME mechanism allows encryption.)
- o The referee may store any referred-by information it sees and paste it into future unrelated requests. (The S/MIME mechanism allows detection of stale assertions by covering a timestamp with the signature and allows detection of use in unrelated requests by covering the Refer-To header field with the signature.)

The mechanisms in this specification do NOT prevent the referee from deleting ALL referred-by information from the referenced request. A refer target can not detect such deletion. This introduces no new problems since removing all referred-by information from a referenced request transforms it into an ordinary SIP request as described in [5]. Thus the referee gains no new influence over processing logic at the refer target by removing the referred-by information.

Refer targets can protect themselves from the possibility of a malicious referee removing a token (leaving an unsecured identity in the Referred-By header field) by using the 429 error response.

Applications using the mechanisms in this document may be able to take advantage of pre-existing relationships between the participants to mitigate the risks of its use. In some transfer scenarios, A has the choice of referring B to C or referring C to B. If A and B have a pre-existing trust relationship, leading A to have greater confidence that B will not behave maliciously (B is A's administrative assistant for example), referring B to C may make more sense.

This mechanism involves two SIP requests between three endpoints, the REFER and the referenced request. The content of those messages (including the referred-by information) is subject to the security considerations and protection mechanisms documented in [5].

Proxies between the participants may collect referred-by information and re-insert it in future requests or make it available to hostile endpoints. The end-to-end confidentiality capabilities discussed in [5] can help reduce the risk of exposing sensitive referred-by information to these proxies. The abuse possibilities in subsequent requests by proxies (or endpoints that they may leak information to) between the referee and the refer target are identical to the abuse by the referee, and the considerations discussed for a malicious referee applies. The abuse possibilities in subsequent requests by proxies (or endpoints that they may leak information to) between the referrer and the referee are similar to those discussed for the presentation of Authenticated Identity Bodies in [7].

6.1. Identifying the Referee in the Referred-by Token

To a refer target, a Referred-By token minimally asserts "The identity expressed by this Referred-By header field asked at the time indicated in this Date header field that the request indicated by this Refer-To header field be sent". This assertion makes no claims at all about who is being asked to send the request. This is sufficient to enable policies such as "Accept any requests referred by Alice", but not "Only accept requests from Bob if he can prove that Alice referred him to us". Thus, there is an opportunity for a cut-and-paste attack. If Mallory sees Alice refer Carol to us using a minimal token, he can copy that token into his own request (as long as it matches what is indicated in the embedded Refer-To header), and it will appear to us that Alice referred Mallory to us. This risk is best mitigated by protecting the REFER Alice sends to Carol from eavesdropping, using TLS or the S/MIME mechanisms detailed in [5].

Including the To header field from the REFER request in the Referred-by token enables the "Only accept requests from Bob if he can prove that Alice referred him to us". Alice is constrained to add this header to the token only if she is sure she is sending the REFER request to Bob. We, in turn, ensure it was Bob that sent the referenced request to us, in addition to validating Alice's signature of the token. Mallory's earlier attack is not effective with this token.

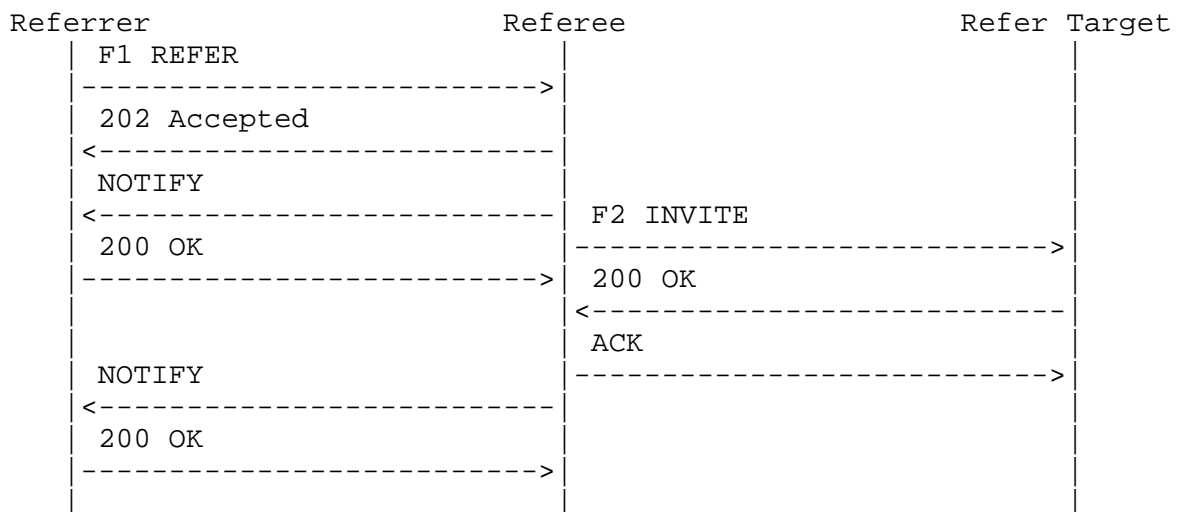
Including the To header field in the Referred-By token has privacy implications, however. Carol, above, might wish to contact us anonymously. That wish would be defeated if Carol's identity appeared in the token Alice created. If Alice encrypted the token to us, Carol will not even be aware of the information leak. To protect herself when she wishes anonymity, Carol will have to reject any REFER requests containing a Referred-By token she can not inspect.

7. Examples

7.1. Basic REFER

This example shows the secured Referred-By mechanism applied to a REFER to an SIP INVITE URI.

Details are shown only for those messages involved in exercising the mechanism defined in this document.



```
F1 REFER sip:referee@referee.example SIP/2.0
Via: SIP/2.0/UDP referrer.example;branch=z9hG4bK392039842
To: sip:referee@referee.example
From: sip:referrer@referrer.example;tag=39092342
Call-ID: 2203900ef0299349d9209f023a
CSeq: 1239930 REFER
Max-Forwards: 70
Contact: <sip:referrer.example>
Refer-To: <sip:refertarget@target.example>
Referred-By: <sip:referrer@referrer.example>
             ;cid="20398823.2UWQFN309shb3@referrer.example"
Content-Type: multipart/mixed; boundary=unique-boundary-1
Content-Length: (appropriate value)

--unique-boundary-1
Content-Type: multipart/signed;
             protocol="application/pkcs7-signature";
             micalg=sha1; boundary=dragons39
Content-ID: <20398823.2UWQFN309shb3@referrer.example>
Content-Length: (appropriate value)

--dragons39
Content-Type: message/sipfrag
Content-Disposition: aib; handling=optional

Date: Thu, 21 Feb 2002 13:02:03 GMT
Refer-To: <sip:refertarget@target.example>
Referred-By: <sip:referrer@referrer.example>
             ;cid="20398823.2UWQFN309shb3@referrer.example"

--dragons39
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
             handling=required

(appropriate signature goes here)

--dragons39--
--unique-boundary-1--

F2 INVITE sip:refertarget@target.example SIP/2.0
Via: SIP/2.0/UDP referee.example;branch=z9hG4bKffe209934aac
To: <sip:refertarget@target.example>
From: <sip:referee@referee.example>;tag=2909034023
Call-ID: fe9023940-a3465@referee.example
CSeq: 889823409 INVITE
Max-Forwards: 70
```

```
Contact: <sip:referee@referee.example>
Referred-By: <sip:referrer@referrer.example>
             ;cid="20398823.2UWQFN309shb3@referrer.example"
Content-Type: multipart/mixed; boundary=my-boundary-9
Content-Length: (appropriate value)

--my-boundary-9
Content-Type: application/sdp
Content-Length: (appropriate value)

v=0
o=referee 2890844526 2890844526 IN IP4 referee.example
s=Session SDP
c=IN IP4 referee.example
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

--my-boundary-9
Content-Type: multipart/signed;
             protocol="application/pkcs7-signature";
             micalg=sha1; boundary=dragons39
Content-ID: <20398823.2UWQFN309shb3@referrer.example>
Content-Length: (appropriate value)

--dragons39
Content-Type: message/sipfrag
Content-Disposition: aib; handling=optional

Date: Thu, 21 Feb 2002 13:02:03 GMT
Refer-To: <sip:refertarget@target.example>
Referred-By: <sip:referrer@referrer.example>
             ;cid="20398823.2UWQFN309shb3@referrer.example"

--dragons39
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
             handling=required

(appropriate signature goes here)

--dragons39--
--my-boundary-9--
```

7.2. Insecure REFER

The flow for this example is the same as that of Section 7.1. Here, the referrer has opted to not include a Referred-By token, and the refer target is willing to accept the referenced request without one.

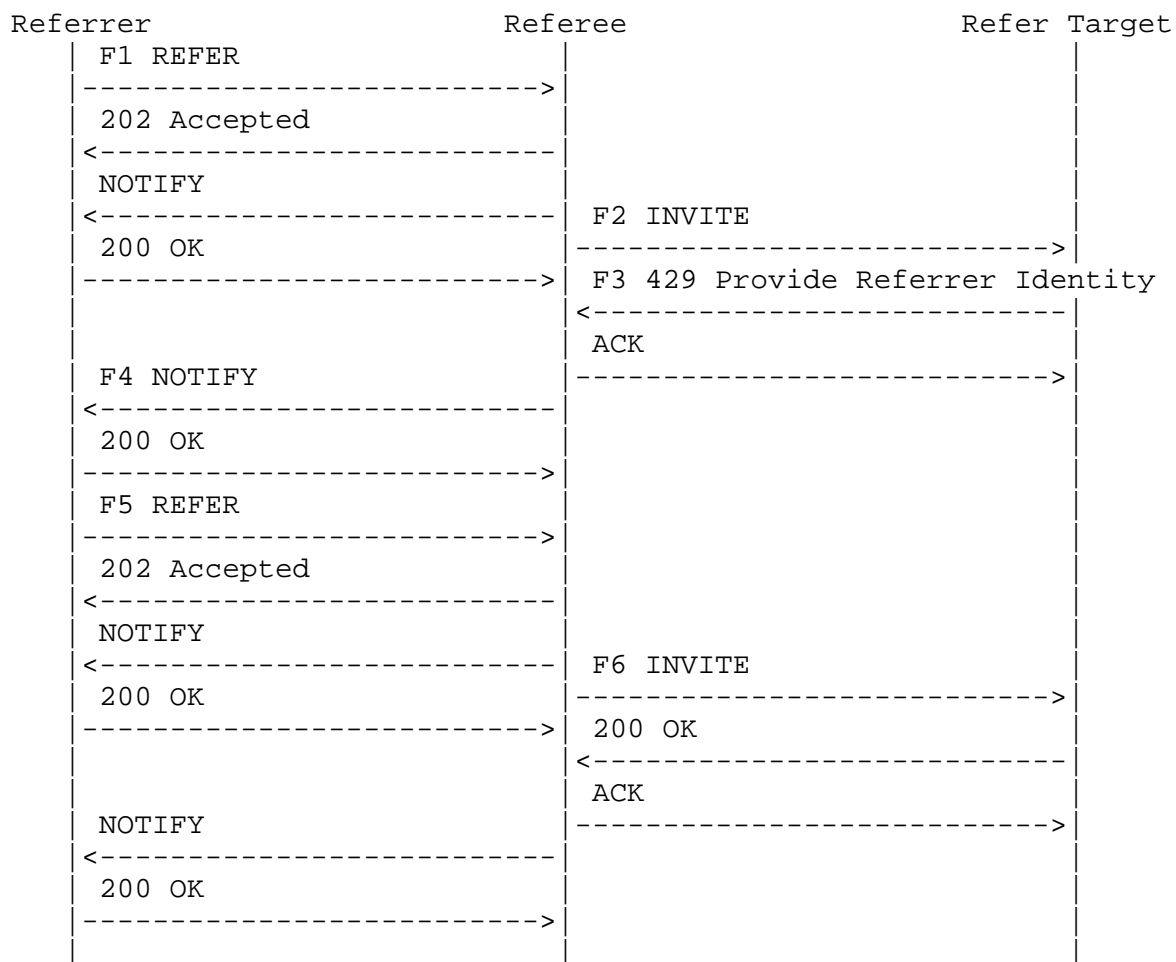
```
F1 REFER sip:referee@referee.example SIP/2.0
Via: SIP/2.0/UDP referrer.example;branch=z9hG4bK392039842
To: <sip:referee@referee.example>
From: <sip:referrer@referrer.example>;tag=39092342
Call-ID: 2203900ef0299349d9209f023a
CSeq: 1239930 REFER
Max-Forwards: 70
Contact: <sip:referrer.example>
Refer-To: <sip:refertarget@target.example>
Referred-By: <sip:referrer@referrer.example>
Content-Length: 0

F2 INVITE sip:refertarget@target.example SIP/2.0
Via: SIP/2.0/UDP referee.example;branch=z9hG4bKffe209934aac
To: <sip:refertarget@target.example>
From: <sip:referee@referee.example>;tag=2909034023
Call-ID: fe9023940-a3465@referee.example
CSeq: 889823409 INVITE
Max-Forwards: 70
Contact: <sip:referee@referee.example>
Referred-By: <sip:referrer@referrer.example>
Content-Type: application/sdp
Content-Length: (appropriate value)

v=0
o=referee 2890844526 2890844526 IN IP4 referee.example
s=Session SDP
c=IN IP4 referee.example
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

7.3. Requiring Referrer Identity

In contrast to the example in Section 7.2, the refer target requires a Referred-By token to accept the referenced request. The referrer chooses to provide an encrypted token (note that the block surrounded by asterisks represents encrypted content). F1 and F2 are identical to the messages detailed in Section 7.2.



F3 SIP/2.0 429 Provide Referrer Identity

Via: SIP/2.0/UDP referee.example;branch=z9hG4bKffe209934aac

To: <sip:refertarget@target.example>;tag=392093422302334

From: <sip:referee@referee.example>;tag=2909034023

Call-ID: fe9023940-a3465@referee.example

CSeq: 889823409 INVITE

Content-Length: 0

F4 NOTIFY sip:referrer@referrer.example SIP/2.0
Via: SIP/2.0/UDP refereee.example;branch=z9hG4bK2934209da390
To: <sip:referrer@referrer.example>;tag=39092342
From: <sip:referee@referee.example>;tag=199949923
Call-ID: 2203900ef0299349d9209f023a
CSeq: 3920390 NOTIFY
Event: refer;id=1239930
Subscription-State: terminated
Content-Type: message/sipfrag
Content-Length: (appropriate value)

SIP/2.0 429 Provide Referrer Identity

F5 REFER sip:referee@referee.example SIP/2.0
Via: SIP/2.0/UDP referrer.example;branch=z9hG4bK98823423
To: <sip:referee@referee.example>
From: <sip:referrer@referrer.example>;tag=39092342
Call-ID: 2203900ef0299349d9209f023a
CSeq: 1239931 REFER
Max-Forwards: 70
Contact: <sip:referrer.example>
Refer-To: <sip:refertarget@target.example>
Referred-By: <sip:referrer@referrer.example>
 ;cid="20342EFXEI.390sdefn2@referrer.example"
Content-Type: multipart/mixed; boundary=unique-boundary-1
Content-Length: (appropriate value)

--unique-boundary-1
Content-Type: multipart/signed;
 protocol="application/pkcs7-signature";
 micalg=sha1; boundary=boundary42
Content-ID: <20342EFXEI.390sdefn2@referrer.example>
Content-Length: (appropriate value)

--boundary42
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
 name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m;
 handling=required
Content-Length: (appropriate value)


```
*****
* Content-Type: message/sipfrag *
* Content-Disposition: aib; handling=optional *
* *
* Date: Thu, 21 Feb 2002 13:02:03 GMT *
* Refer-To: <sip:refertarget@target.example> *
* Referred-By: <sip:referrer@referrer.example> *
* ;cid="20342EFXEI.390sdefn2@referrer.example" *
*****
```

--boundary42

Content-Type: application/pkcs7-signature; name=smime.p7s
 Content-Transfer-Encoding: base64
 Content-Disposition: attachment; filename=smime.p7s;
 handling=required

(appropriate signature)

--boundary42--

F6 INVITE sip:refertarget@target.example SIP/2.0
 Via: SIP/2.0/UDP referee.example;branch=z9hG4bK3920390423
 To: <sip:refertarget@target.example>
 From: <sip:referee@referee.example>;tag=1342093482342
 Call-ID: 23499234-9239842993@referee.example
 CSeq: 19309423 INVITE
 Max-Forwards: 70
 Referred-By: <sip:referrer@referrer.example>
 ;cid="20342EFXEI.390sdefn2@referrer.example"
 Contact: <sip:referee@referee.example>
 Content-Type: multipart/mixed; boundary=my-boundary-9
 Content-Length: (appropriate value)

--my-boundary-9

Content-Type: application/sdp
 Content-Length: (appropriate value)

v=0
 o=referee 2890844526 2890844526 IN IP4 referee.example
 s=Session SDP
 c=IN IP4 referee.example
 t=0 0
 m=audio 49172 RTP/AVP 0
 a=rtpmap:0 PCMU/8000

```

--my-boundary-9
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature";
  micalg=sha1; boundary=boundary42
Content-ID: <20342EFXEI.390sdefn2@referrer.example>
Content-Length: (appropriate value)

--boundary42
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
  name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m;
  handling=required
Content-Length: (appropriate value)

*****
* Content-Type: message/sipfrag *
* Content-Disposition: aib; handling=optional *
* *
* Date: Thu, 21 Feb 2002 13:02:03 GMT *
* Refer-To: <sip:refertarget@target.example> *
* Referred-By: <sip:referrer@referrer.example> *
* ;cid="20342EFXEI.390sdefn2@referrer.example" *
*****

--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
  handling=required

(appropriate signature)

--boundary42--
--my-boundary-9--

```

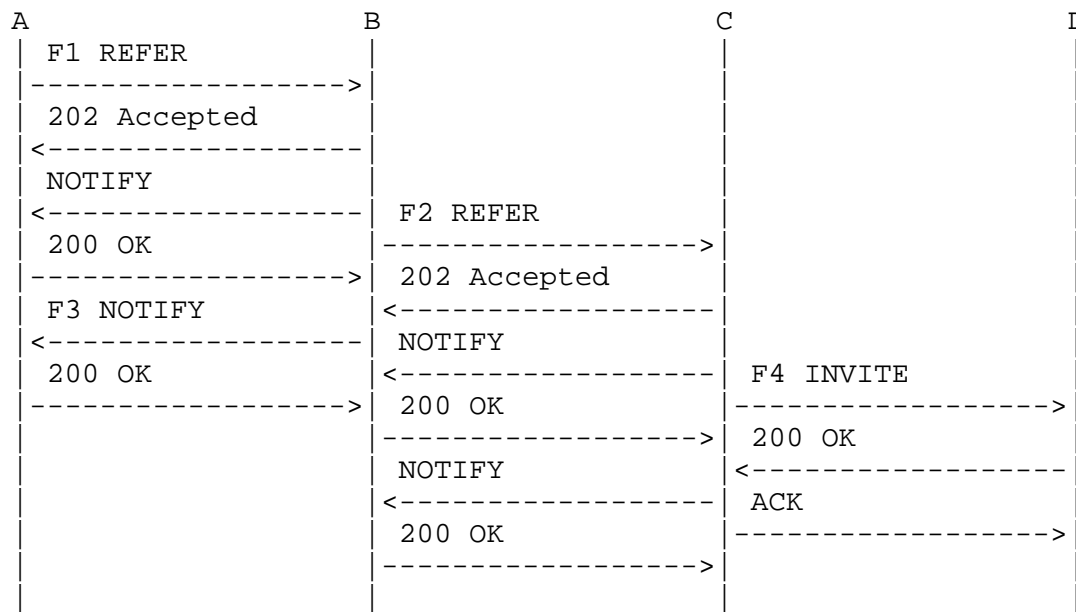
7.4. Nested REFER

The Refer-To URI may be a SIP URI indicating the REFER method. Consider The following URI which A uses to refer B to send a REFER request to C which refers C to send an INVITE to D.

Note that A provides a Referred-By token which gets passed through B and C to D. In particular, B does not provide its own Referred-By token to C. Also note that A is notified of the outcome of the request it triggered at B (the REFER), not at C (the INVITE).

```
Refer-To: <sip:C.example;method=REFER?Refer-To="<sip:D.example>">
```

This reference would result in the following flow:



```

F1 REFER sip:B SIP/2.0
Via: SIP/2.0/UDP A.example;branch=z9hG4bK3802394232
To: <sip:B.example>
From: <sip:A.example>;tag=23490234
Call-ID: 2304098023@A.example
CSeq: 2342093 REFER
Max-Forwards: 70
Contact: <sip:A.example>
Refer-To: <sip:C.example;method=REFER?Refer-To="<sip:D>.example">
Referred-By: <sip:A.example>;
    cid="23094202342.10123091233@A.example"
Content-Type: multipart/mixed; boundary=unique-boundary-1
Content-Length: (appropriate value)

--unique-boundary-1
Content-Type: multipart/signed;
    protocol="application/pkcs7-signature";
    micalg=sha1; boundary=dragons39
Content-ID: <23094202342.10123091233@A.example>
Content-Length: (appropriate value)

--dragons39
Content-Type: message/sipfrag
Content-Disposition: aib; handling=optional
  
```

```
Date: Thu, 21 Feb 2002 13:02:03 GMT
Refer-To: <sip:C.example;method=REFER?Refer-To=<sip:D.example>>
Referred-By: <sip:A.example>;
    cid="23094202342.10123091233@A.example"

--dragons39
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
    handling=required

(appropriate signature goes here)

--dragons39--
--unique-boundary-1--

F2 REFER sip:C.example SIP/2.0
Via: SIP/2.0/UDP B.example;branch=z9hG4bK00239842
To: <sip:C.example>
From: <sip:B.example>;tag=2934u23
Call-ID: 203942834@B.example
CSeq: 8321039 REFER
Max-Forwards: 70
Contact: <sip:B.example>
Refer-To: <sip:D.example>
Referred-By: <sip:A.example>;
    cid="23094202342.10123091233@A.example"
Content-Type: multipart/mixed; boundary=unique-boundary-1
Content-Length: (appropriate value)

--unique-boundary-1
Content-Type: multipart/signed;
    protocol="application/pkcs7-signature";
    micalg=sha1; boundary=dragons39
Content-ID: <23094202342.10123091233@A.example>
Content-Length: (appropriate value)

--dragons39
Content-Type: message/sipfrag
Content-Disposition: aib; handling=optional

Date: Thu, 21 Feb 2002 13:02:03 GMT
Refer-To: <sip:C.example;method=REFER?Refer-To=<sip:D.example>>
Referred-By: <sip:A.example>;cid="23094202342.1012309123@A.example"
```

```
--dragons39
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
    handling=required

(appropriate signature goes here)

--dragons39--
--unique-boundary-1--

F3 NOTIFY sip:A.example SIP/2.0
Via: SIP/2.0/UDP A.example;branch=z9hG4bK3802394232
To: <sip:A.example>;tag=23490234
From: <sip:B.example>;tag=5923020
Call-ID: 2304098023@A.example
CSeq: 29420342 NOTIFY
Event: refer;id=2342093
Subscription-State: terminated
Max-Forwards: 70
Contact: <sip:B.example>
Content-Type: message/sipfrag
Content-Length: (appropriate value)

SIP/2.0 202 Accepted

F4 INVITE sip:D.example SIP/2.0
Via: SIP/2.0/UDP C.example;branch=z9hG4bK29348234
To: <sip:D.example>
From: <sip:C.example>;tag=023942334
Call-ID: 23489020352@C.example
CSeq: 1230934 INVITE
Max-Forwards: 70
Contact: <sip:C.example>
Referred-By: <sip:A.example>;
    cid="23094202342.10123091233@A.example"
Content-Type: multipart/mixed; boundary=unique-boundary-1
Content-Length: (appropriate value)

--unique-boundary-1
Content-Type: application/sdp
Content-Length: (appropriate value)
```

```
v=0
o=C 2890844526 2890844526 IN IP4 C.example
s=Session SDP
c=IN IP4 C.example
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

--unique-boundary-1
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature";
  micalg=sha1; boundary=dragons39
Content-ID: <23094202342.10123091233@A.example>
Content-Length: (appropriate value)

--dragons39
Content-Type: message/sipfrag
Content-Disposition: aib; handling=optional

Date: Thu, 21 Feb 2002 13:02:03 GMT
Refer-To: <sip:C.example;method=REFER?Refer-To="<sip:D.example>">
Referred-By: <sip:A.example>;
  cid="23094202342.1012309123@A.example"

--dragons39
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
  handling=required

(appropriate signature goes here)

--dragons39--
--unique-boundary-1--
```

8. IANA Considerations

This document defines a new SIP header field name with a compact form (Referred-By and b respectively). It also defines a new SIP client error response code (429).

The following changes are reflected at:

<http://www.iana.org/assignments/sip-parameters>

The following row has been added to the header field section (replacing any existing row for Referred-By).

Header Name	Compact Form	Reference
Referred-By	b	[RFC3892]

The following row has been added to the response code section under the Request Failure 4xx heading.

429 Provide Referrer Identity	[RFC3892]
-------------------------------	-----------

9. Contributors

Rohan Mahy distilled RFC2822's msg-id into this document's definition of sip-clean-msg-id.

10. References

10.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.
- [3] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.
- [4] Sparks, R., "Internet Media Type message/sipfrag", RFC 3420, November 2002.
- [5] 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.

- [6] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

10.2. Informative References

- [7] Peterson, J., "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", Work in Progress, March 2003.
- [8] Sparks, R. and A. Johnston, "Session Initiation Protocol Call Control - Transfer", Work in Progress, February 2003.
- [9] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

11. Author's Address

Robert J. Sparks
Xten
5100 Tennyson Parkway
Suite 1000
Plano, TX 75024

EMail: RjS@xten.com

12. Full Copyright Statement

Copyright (C) The Internet Society (2004).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/S HE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the IETF's procedures with respect to rights in IETF Documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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

