SIP Working Group
Internet Draft
Document: <draft-ietf-sip-privacy-04.txt>
Category: Standards Track

W. Marshall
AT&T

K. Ramakrishnan
TeraOptic Networks

E. Miller
Terayon

G. Russell
CableLabs

B. Beser
Juniper Networks

M. Mannette
K. Steinbrenner
3Com

D. Oran
F. Andreasen
Cisco

J. Pickens
Com21

P. Lalwaney
Nokia

J. Fellows
Copper Mountain Networks

D. Evans
D. R. Evans Consulting

K. Kelly
NetSpeak

M. Watson
Nortel Networks

February 27, 2002

SIP Extensions for Network-Asserted Caller Identity and Privacy within Trusted Networks

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

1. Abstract

This document describes extensions to SIP that enable a network of trusted SIP servers to assert the identity of end users or end systems, and to convey indications of end-user requested privacy. The use of these extensions are only applicable inside an administrative domain, or among federations of administrative domains with previously agreed-upon policies for usage of such information. This document does NOT offer a general privacy or identity model suitable for inter-domain use or use in the Internet at large.

2. Scope of Applicability

This document describes extensions to SIP that enable a network of trusted SIP servers to assert the identity of end users or end systems, and to convey indications of end-user requested privacy. The use of these extensions are only applicable inside an administrative domain, or among federations of administrative domains with previously agreed-upon policies for usage of such information. Such a "network" is explicitly trusted by its users and end-systems to either publicly assert the identity of each party, or be responsible for withholding that identity outside of the trusted domain or federation of domains if privacy is requested. The means by which the network determines the identity to assert is outside the scope of this document.

This document does NOT offer a general privacy or identity model suitable for inter-domain use or use in the Internet at large. Its assumptions about the trust relationship between the user and the network may not apply in many applications. For example, these extensions do not accommodate a model whereby end users can independently assert their identity by use of the extensions defined here. Furthermore, since the asserted identities are not cryptographically certified, they are subject to forgery, replay,

and falsification in any architecture that does not provide full transitive trust. The asserted identities also lack an indication of who is asserting the identity, and therefore the assertions are not useful outside of the federation of domains, where such information

would be crucial in order to determine the validity or value of the assertion.

Despite these limitations, there are sufficiently useful specialized deployments that meet the assumptions described above, and can accept the limitations that result, to warrant publication of this mechanism. An example deployment would be a closed network which emulates a traditional circuit switched telephone network.

It should be noted, that the mechanisms described in this draft are not intended to be used for user-asserted identity. As described above, the mechanisms are merely intended to enable trusted intermediaries to assert an identity for users.

3. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [2].

4. Introduction

Various providers which attempt to offer a telephony service over IP networks have selected SIP as a base protocol. These environments require a way for trusted network elements (for example SIP proxy servers) to communicate the identity of users using such a service, yet also need to withhold this information from untrusted entities under certain circumstances. Such networks typically assume some level of transitive trust.

These networks must support certain popular telephony services as well as some regulatory and public safety requirements. These include Calling Identity Delivery services, Calling Identity Delivery Blocking, and the ability to trace the originator of a call. While baseline SIP can support each of these services independently, certain combinations cannot be supported. For example, a caller that wants to maintain privacy and consequently provides unintelligible information in the SIP From header field will not be identifiable by intermediaries. However, since SIP does not allow the contents of the From header field to be modified by intermediaries, this will prevent certain services, e.g., call trace, from being performed by intermediaries which do not directly perform SIP authentication. Furthermore, the issue of privacy in an IP environment is more complicated than in the PSTN. The caller and callee will normally exchange IP traffic directly, and IP address information itself may reveal some privacy. The issue of IP address privacy for both the caller and callee consequently needs to be addressed as well. Although we recognize and discuss the IP address privacy problem, we do not provide a solution to it in this

document.

In order to solve the network asserted caller identity and privacy problem we assume an architecture where the caller initiates a

session to the callee via a trusted entity in its network. The callee in turn receives the session initiation via a trusted entity in its network. A trusted entity is here defined as a SIP proxy or SIP UA, that belongs to and is controlled by the "network". The trusted entities provide the services of determining the identity of the calling or called party, and furthermore add identity information for that party to the SIP messages. Trusted entities in the network thus serve as intermediaries that provide the caller and callee with network asserted identity information about the remote party. It should be noted, that the means by which the trusted entities determine these asserted identities are outside the scope of this document.

The trusted entities do not trust the end users or end systems they serve. Furthermore, trusted entities may not trust their next or previous hop, e.g., if that hop represents a different untrusted domain. This leads our architecture to have the concept of trust boundaries. Trust and trust boundaries are in the eye of the beholder. For example, if A trusts B, but B does not trust A, then A does not see a trust boundary between the two, but B does. When an entity receives a message across a trust boundary, it does not trust it. From a network asserted caller identity point of view, this means the identity information cannot be trusted. Similarly, if a message is to be sent across a trust boundary, the sending entity does not trust the next hop to handle the message as desired. This implies, that the entity must ensure, that any privacy needed is provided before the message is forwarded across a trust boundary, while still being able to trace the originating party if needed.

The architecture is illustrated in the following figures.

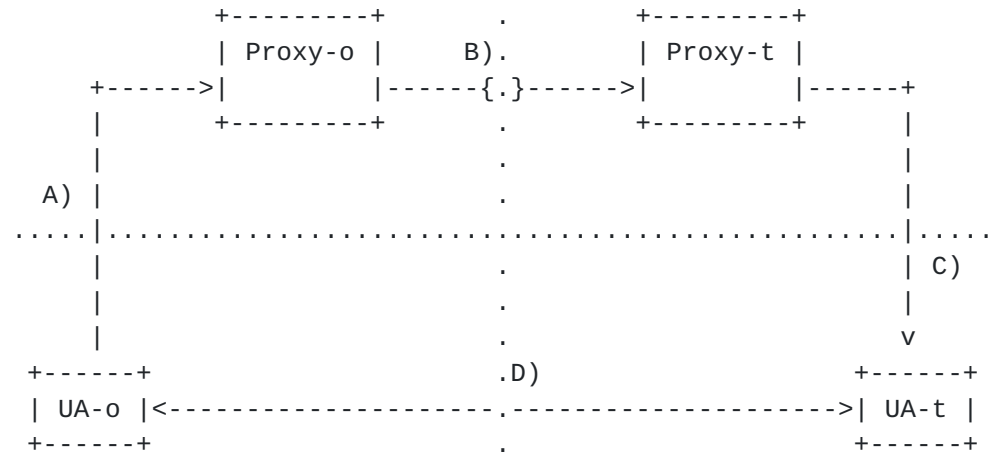

Figure 1 - Basic Architecture with Trust Boundaries (1)

In Figure 1, we show the basic architecture which includes two user agents, two proxies (trusted entities), and four trust boundaries. The trust relationships associated with these are:

A) UA-o trusts Proxy-o, however Proxy-o does not trust UA-o.
B) Proxy-o may or may not trust Proxy-t, and Proxy-t may or may not trust Proxy-o.
C) Proxy-t does not trust UA-t, however UA-t trusts Proxy-t.
D) UA-o may or may not trust UA-t, and UA-t may or may not trust UA-o.

In the above, Proxy-o serves as the trusted intermediary for UA-o, whereas Proxy-t serves as the trusted intermediary for UA-t. Proxy-o determines and asserts the identity information for UA-o, whereas Proxy-t determines and asserts the identity information for UA-t. Both UA-o and UA-t are referred to as untrusted user agents in the above.

In Figure 2, we consider another example, this time introducing the concept of a trusted user agent. A trusted user agent is a UA operated by the network, for example a PSTN gateway or a voicemail system, as opposed to an end user or customer:

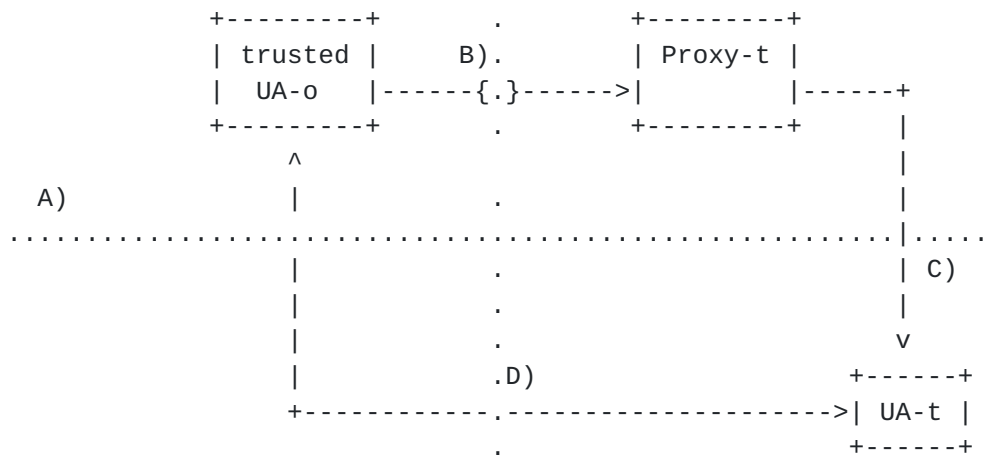

Figure 2 - Basic Architecture with Trust Boundaries (2)

The trust relationships associated with the trust boundaries are:

A) UA-o does not trust UA-t. UA-t may or may not trust UA-o.
B) UA-o may or may not trust Proxy-t, and Proxy-t may or may not trust UA-o.
C) Proxy-t does not trust UA-t, however UA-t trusts Proxy-t.
D) Same as A.

In this case, UA-o is a trusted user agent and hence does not need a trusted intermediary; UA-o simply provides the asserted identity information itself. However, UA-t is an untrusted user agent, and hence Proxy-t, which serves as the trusted intermediary for UA-t, determines and asserts the identity information for UA-t.

Finally, in Figure 3 we consider the case of two trusted user agents:

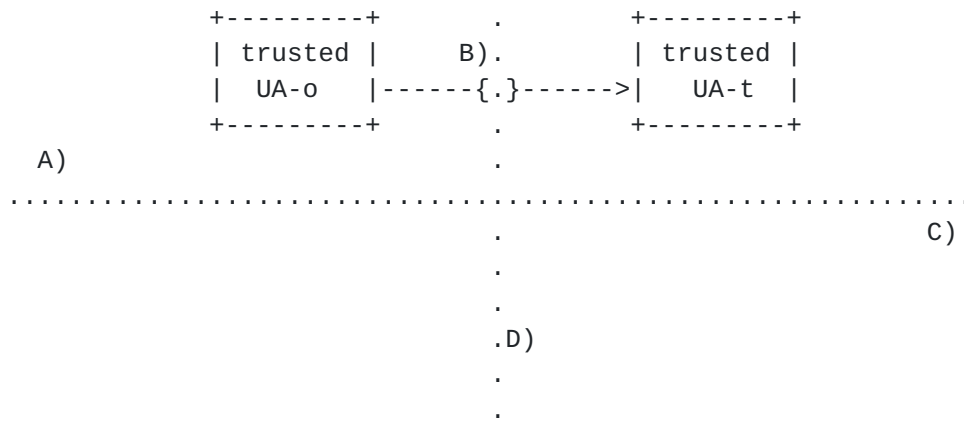

Figure 3 - Basic Architecture with Trust Boundaries (3)

The trust relationships associated with the trust boundaries are:

A) N/A.
B) UA-o may or may not trust UA-t, and UA-t may or may not trust UA-o.
C) N/A.
D) N/A.

In this case, both UA-o and UA-t are trusted user agent and hence do not need a trusted intermediary; the user agents simply provide the asserted identity information themselves.

In this document we define two extensions to SIP that allow the calling and called parties to have their identity asserted by a trusted intermediary while still being able to maintain their identity privacy with respect to one another.

The first extension is a new general header, Remote-Party-ID, which identifies a party and is added by the trusted network entities. Different types of party information can be provided, e.g., calling, or called party, and for each type of party, different types of identity information, e.g. subscriber, or terminal, can be provided. Since a party may not wish to reveal some or all of this information to an untrusted entity, the party can request a specific level of privacy for each. The intermediary also has the ability to specify a required level of privacy.

The second extension is a new general header, RPID-Privacy, which specifies the privacy handling desired by the user for any Remote-Party-ID headers added by intermediaries. This enables an entity to control the desired level of privacy when intermediaries add Remote-Party-ID headers that assert the identity of the entity.

When a trusted intermediary receives a message from an untrusted entity, the trusted intermediary attempts to determine the identity

of the originator, by means outside the scope of this document. When the identity has been determined, the trusted intermediary ensures that corresponding Remote-Party-ID information is included in the message sent. Also, the trusted intermediary ensures that any privacy requested (with respect to the other party) is provided prior to forwarding a message across a trust boundary to an untrusted entity. Any Remote-Party-ID information received from an untrusted entity is either verified successfully using an authentication mechanism or an explicit policy of trust, or tagged with an indication that it could not be verified, so the receiver knows that it should not necessarily trust the information.

This document defines a set of party types and identity information. New types of party and identity information as well as other attributes may be introduced, thereby allowing new services to make use of the generic network asserted identity information and privacy handling defined here.

5. Protocol Overview

When an untrusted UAC sends an INVITE, OPTIONS, REGISTER or extension method request through a trusted intermediary, i.e., proxy, the proxy MAY be adding one or more Remote-Party-ID headers that identify the calling party. The UAC can indicate the level of privacy that should be afforded to such Remote-Party-ID headers by including one or more RPID-Privacy headers with the request. The RPID-Privacy header allows the UAC to control the privacy down to the party-type and identity-type level.

When a trusted UAC sends an INVITE, OPTIONS, REGISTER or extension method request, the trusted UAC includes a calling subscriber Remote-Party-ID header field in the request in order to identify the originator of the call. The calling subscriber Remote-Party-ID MUST contain an addr-spec identifying the caller and MAY contain a "display-name" for the caller as well. Other types of network asserted Remote-Party-ID MAY be included as well. If privacy is desired for a given Remote-Party-ID header, the UAC MUST include a privacy token set to one or more of "uri", "name" or "full". Furthermore, if the UAC wants to control privacy for any Remote-Party-ID headers added by downstream proxies, the UAC MUST include one or more RPID-Privacy headers specifying the desired privacy.

When a proxy supporting this extension receives an INVITE, OPTIONS, REGISTER or extension method request from an untrusted entity (UA or proxy), the proxy first examines the request for the presence of any Remote-Party-ID headers. The value of these headers cannot be trusted and hence the proxy will either have to validate them (by means outside the scope of this document) or make sure they are not marked as trusted. If the proxy wants to ensure, that the calling

party can be identified by the called party, the proxy MUST authenticate the calling party (by means outside the scope of this document) and insert a calling party Remote-Party-ID header that is marked as being trusted. If the proxy is unable to authenticate the

calling party, it MAY reject the request, e.g., with a 403 or 407. If the proxy can determine other types of identity information for the calling party, it MAY insert those as trusted Remote-Party-ID headers as well. If the request contained one or more RPID-Privacy header fields, any Remote-Party-ID header fields added by the proxy MUST have their privacy indication set accordingly.

Prior to a trusted entity (UA or proxy) forwarding the INVITE, OPTIONS, REGISTER or extension method request to an untrusted entity, the trusted entity MUST look for the presence of a privacy request indication in each Remote-Party-ID header field. If one is found, the privacy requested MUST be provided for that Remote-Party-ID header field prior to forwarding the request to the untrusted entity. For "uri" and "name" privacy, this typically involves encrypting and possibly removing information provided in the Remote-Party-ID.

Once a UAS supporting this extension receives the INVITE, OPTIONS, REGISTER or extension method request via a trusted entity, the UAS can use the calling subscriber Remote-Party-ID information provided to identify the originator of the call, unless the originator had requested privacy. Note that if the UAS did not receive the call via a trusted entity, any Remote-Party-ID information provided may be false.

The subsequent behavior now depends on whether the UAS is trusted or not:

* If the UAS is trusted, it SHOULD include a called subscriber Remote-Party-ID identifying the called party in the first non-100 response. The party information SHOULD be set to "called" and the identity information SHOULD be set to "subscriber". Additional Remote-Party-ID header fields MAY be provided as well. If the UAS desires privacy for a Remote-Party-ID, it MUST include a privacy request indication in that Remote-Party-ID header. Note that the privacy request is only guaranteed to be satisfied if the previous hop is trusted and it furthermore supports the extensions defined here. If the UAS can not guarantee both, then any privacy desired MUST be provided before the response is forwarded upstream.

* If the UAS instead is untrusted, the session simply continues as described below.

If the UAS (trusted or untrusted) wants to control the level of privacy afforded to any Remote-Party-ID headers that may be inserted by upstream proxies in its response, the UAS MUST include one or more RPID-Privacy header fields with the relevant privacy indication(s) in that response.

A trusted UAS MAY also include Remote-Party-ID headers in subsequent provisional and final responses to the request. The trusted UAS SHOULD include a called party Remote-Party-ID header if the contents are different than sent in a previous response. The party

information SHOULD be set to "called" and the identity information SHOULD be set to "subscriber". Additional Remote-Party-ID header fields may be provided as well.

When a proxy supporting this extension receives a non-100 response to an INVITE, OPTIONS, REGISTER or extension method request from an untrusted entity, the proxy first examines the response for the presence of any Remote-Party-ID headers. By definition, the value of these headers cannot be trusted and hence the proxy will either have to validate them (by means outside the scope of this document) or make sure they are not marked as trusted. If the proxy wants to make sure that the called party can be identified, the proxy MUST authenticate the called party (by means outside the scope of this document) and insert a called party Remote-Party-ID header that is marked as being trusted. If the proxy is unable to identify the called party, the proxy SHOULD simply mark any Remote-Party-ID headers in the message as untrusted. If the proxy can determine other types of identity information for the calling party, the proxy MAY insert those as trusted Remote-Party-ID headers as well. If the response contained one or more RPID-Privacy parameters, any Remote-Party-ID header fields added by the proxy MUST have their privacy indication set accordingly.

The proxy MUST also ensure that any privacy requested in the response is provided prior to forwarding it to an untrusted entity.

Finally, when a UAC (trusted or not) receives the first non-100 response to an INVITE, OPTIONS, or REGISTER request from a trusted entity, the UAC can use the called subscriber Remote-Party-ID information (if present) to identify the called party, unless the terminator had requested privacy. Subsequent non-100 responses MAY contain Remote-Party-ID information as well. When the UAC receives the final 200 response, it MAY contain a called subscriber Remote-Party-ID header identifying the party the UAC was connected to. Again, this information SHOULD NOT be trusted if it was not received via a trusted entity.

6. Header Field Definitions

Table 1 below is an extension of tables 2 and 3 in [3] for the new header fields defined here:

	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Remote-Party-ID		amd	-	-	-	o	o	o
RPID-Privacy		r	-	-	-	o	o	o

Table 1: Summary of header fields.

The Remote-Party-ID and the RPID-Privacy headers can be used with the INVITE, OPTIONS, and REGISTER methods as well as any response to these. Extension methods MAY utilize these headers to achieve remote

party identification and privacy. The procedures at User Agents MAY be specific to particular new methods, however, the generic handling at proxies MUST be as specified in this document.

Note that any privacy requested may not be honored unless the request or response is sent through a trusted entity that supports the extensions defined here. Similarly, Remote-Party-ID information may not be trustworthy if it was received in a request or response from a non-trusted entity or an entity that does not support the extensions defined here.

6.1 Remote-Party-ID Header Field Definitions

The Remote-Party-ID header field provides information about the remote party. Different types of party information can be provided, e.g. calling and called, and for each, different types of identity information can be provided as well. A request or response MAY contain more than one Remote-Party-ID header field, with privacy requested independently for each. Remote-Party-ID is defined by the following ABNF [4] (productions for the undefined nonterminals can be found in [3]):

```
  Remote-Party-ID    = "Remote-Party-ID" HCOLON rpid *(COMMA rpid)

  rpid               =  [display-name] LAQUOT addr-spec RAQUOT
                                              *(SEMI rpi-token)

  rpi-token          = rpi-screen / rpi-pty-type /
                        rpi-id-type / rpi-privacy / other-rpi-token

  rpi-screen         = "screen" EQUAL ("no" / "yes")

  rpi-pty-type       = "party" EQUAL ("calling" / "called" / token)

  rpi-id-type        = "id-type" EQUAL ("subscriber" / "user" /
                                         "term"  / token)

  rpi-privacy        = "privacy" EQUAL
                          (rpi-priv-element
                           /(LDQUOT rpi-priv-element
                              *(COMMA rpi-priv-element) RDQUOT)
                           )

  rpi-priv-element   = ("full" / "name" / "uri" / "off" / token)
                                    ["-" ( "network" / token )]

  other-rpi-token    = ["-"] token [EQUAL (token / quoted-string)]
```

Furthermore, we define the value "private" for "other-user" in an "addr-spec", to indicate that the userinfo part of an "addr-spec" is

in a non-intelligible form. The syntax for "other-user" is therefore refined to:

other-user = token / "private"

Comparisons follow the case-sensitivity rules defined by SIP [3].

The "display-name" in Remote-Party-ID is a text string that identifies the name of the party. The "addr-spec" contains information identifying the party either in clear-text or encrypted form. In the latter case, the addr-spec contains a SIP-URI or a SIPS-URI where the "userinfo" part of the "addr-spec" typically contains the encrypted party information, whereas the "hostport" identifies the entity that can decrypt the information. Furthermore, an "other-user" value of "private" will then be present to indicate that the "addr-spec" is non-intelligible. Depending on the rpi-pty-type, the "addr-spec" can be used as the Request-URI by the UA to initiate certain call control functions or subsequent calls that are required to reference the party.

The rpi-screen parameter describes what level of trust network associates with the Remote-Party-ID information. The value "yes" indicates the Remote-Party-ID was asserted by the previous hop or the previous hop received the message from a trusted entity with this indication. The value "no" (assumed by default) indicates the Remote-Party-ID was either not asserted by the previous hop or the previous hop received the message from an untrusted entity. Multiple rpi-screen parameters MAY be present in a Remote-Party-ID - if both "yes" and "no" are present, "no" will take precedence. It should be noted, that the rpi-screen parameter provides a somewhat weak form of trustworthiness. In particular, it depends on transitive trust as well as correct implementation, configuration and support for the extensions defined here in the associated chain of trust. Should any of these dependencies not hold, the value "yes" may actually not be trustworthy. Future extensions to SIP may define a general and more robust mechanism that can be used here.

The rpi-pty-type describes the type of party to which this header refers. There MUST NOT be more than one rpi-pty-type present in a Remote-Party-ID. If the rpi-pty-type parameter is absent, the "display-name" and "addr-spec" describe the party from which the request or response was received, i.e., "calling" party in the case of requests and "called" in the case of responses. Additional values MAY be defined as extensions. Such extensions SHALL be documented in an RFC and registered with IANA subject to the considerations given in Section 10.2.

The rpi-id-type describes the nature of the identity provided. Several types of identity can be provided for each party, however there MUST NOT be more than one rpi-id-type present in a given Remote-Party-ID. If the rpi-id-type parameter is absent, the Remote-Party-ID contains the subscriber identity.

This document defines three identity types that can be asserted by a trusted intermediary - additional values MAY be defined as extensions in which case they SHALL be documented in an RFC and

registered with IANA subject to the considerations given in Section 10.2. The types are defined based on the nature of the information they represent, as opposed to a particular application. Individual applications requiring network asserted identity information will specify which identities should be used for that application:

Subscriber identity (rpi-id-type="subscriber"):

  This identifies the owner of the subscription which is being used for the session.

User identity (rpi-id-type="user"):

  This identifies the individual participating in the session. This can be used when multiple individuals are able to originate sessions under the same subscription. If absent, this is assumed to be the same as the subscriber identity.

Terminal identity (rpi-id-type="term")

  This identifies the terminal being used for the session. For example several users may be able to 'log in' to a single terminal, in which case the identity of the terminal will differ from that of the user, subscriber, etc. If absent, this is assumed to be the same as the subscriber identity.

Entities SHOULD NOT include multiple Remote-Party-ID headers containing the same identity information marked with different rpi-id-types.

The rpi-privacy parameter describes whether the identity information must be hidden from untrusted entities. There MAY be multiple rpi-privacy parameters in a Remote-Party-ID. If privacy is requested, it MUST be one or more of "full", "uri", or "name". The value "full" means that both the "display-name" and the "addr-spec" MUST be hidden. The values "name" and "uri" mean that the "display-name" or the "addr-spec" MUST be hidden respectively. The value "off" indicates that lack of privacy is explicitly requested, and MUST be the only value if present. The values may be postfixed with a string indicating that the privacy request was made by an entity other than the party itself. Postfixing with the value "-network" indicates that intermediaries ("the network") have requested that the information be hidden, despite the user not making such a request. Additional values MAY be defined as extensions. Such extensions SHALL be documented in an RFC and registered with IANA subject to the considerations given in Section 10.2.

It should be noted, that an entity requesting only Remote-Party-ID privacy will not receive complete privacy. The values "uri" and "name" merely affect information that may be displayed as opposed to

truly hiding the identity of the requesting entity, since the identity of the host, e.g., IP address, is not hidden. For full

privacy, the entity will need IP address privacy as well - this is discussed further in Section 8.2.

Finally, the "other-rpi-token" parameter allows the Remote-Party-ID header field to be extended with other types of parameters which SHALL be documented in an RFC and registered with IANA subject to the considerations given in Section 10.2. By default, such extensions will be assumed to contain network asserted identity information, and hence MUST be supported for identity verification to pass successfully. The prefix "-" is used to indicate that a parameter extension does not need to be supported by a given entity in order for that Remote-Party-ID to be verified successfully. Consequently, such extensions MUST NOT begin with the character "-".

6.2 RPID-Privacy Header Field Definition

The RPID-Privacy header field allows an entity (typically an untrusted user agent) to indicate a desired level of privacy for any Remote-Party-ID header that may be added by subsequent entities, e.g., a downstream proxy. Any Remote-Party-ID header added of the party-type and identity-type indicated, shall have the privacy specified applied to it. If the party-type is omitted, the privacy specified applies to all party-types. If the identity-type is omitted, the privacy specified applies to all identity-types. A request or response MAY contain zero, one or more RPID-Privacy header fields. The RPID-Privacy header field is defined by the following ABNF [4]:

```
RPID-Privacy       = "RPID-Privacy" HCOLON rpid-priv
                                          *(COMMA rpid-priv)

rpid-priv          = rpid-privacy-token *(SEMI rpid-privacy-token)
                                    ; rpi-privacy MUST be present

rpid-privacy-token = rpi-pty-type / rpi-id-type / rpi-privacy
```

Comparisons follow the case-sensitivity rules defined by SIP [3].

When multiple RPID-Privacy headers are present, the following precedence rules MUST be used:

* RPID-Privacy with both rpi-id-type and rpi-pty-type takes precedence over
* RPID-Privacy with only rpi-id-type, which takes precedence over
* RPID-Privacy with only rpi-pty-type, which takes precedence over
* RPID-Privacy with neither rpi-id-type nor rpi-pty-type.

Any remaining overlaps or conflicts are resolved by order: a later RPID-Privacy indication in a message will take precedence over an earlier RPID-Privacy indication in that message. The following

example illustrates the above:

RPID-Privacy: privacy=full;party=calling;id-type=subscriber

```
RPID-Privacy: party=calling;rpi-privacy=off
RPID-Privacy: party=calling;rpi-privacy=uri
```

Per the rules above, a new calling subscriber Remote-Party-ID will get full privacy, and any other calling party Remote-Party-ID will get uri privacy.

7. Protocol Semantics

Below, we provide the protocol semantics for an untrusted UAC, a trusted UAC, an untrusted UAS, a trusted UAS, and a proxy.

7.1 Untrusted UAC Behavior

When an untrusted UAC supporting this extension sends an INVITE, OPTIONS, REGISTER or extension method request, and the UAC wants to control the privacy for any Remote-Party-ID header that might be added by a downstream proxy, the UAC MUST include one or more RPID-Privacy headers. Each of these RPID-Privacy headers MUST include an rpi-privacy parameter specifying the desired level of privacy, e.g. "uri", to maintain privacy of the "addr-spec".

If the UAC desires "name" or "full" privacy, the UAC MUST NOT reveal the originating subscriber's name in the "display-name" portion of any header. This can be achieved by either not providing a "display-name" or by setting the "display-name" to "Anonymous" in such fields, e.g., From and Contact.

If the UAC desires "uri" or "full" privacy, the UAC MUST NOT reveal the subscriber's identity in any header field. In particular, the contents of header fields need to be considered as described below:

* From: The UAC SHOULD supply a cryptographically random identifier for the userinfo, and a non-identifying hostname, e.g., "localhost", in the host name. The cryptographically random identifier ensures a globally unique dialog identification (despite the use of "localhost") while still providing privacy.

* To: If a telephone number is used in the addr-spec, the telephone number SHOULD be a full E.164 number (including the country code) that is different from the From header field. If a host name is included, it SHOULD be a fully qualified domain name.

* Contact: The same cryptographically random identifier used in the From header field SHOULD be supplied for the userinfo, and an IP-address SHOULD be used in the host name.

* All other headers that may contain either an IP address or a domain name, e.g., Call-ID, and Via, SHOULD use the IP-address form. It should however be noted, that this simple privacy step

may be overcome fairly easily in many cases.

The UAC may also explicitly request that privacy is not to be provided by setting the rpi-privacy parameter in the corresponding RPID-Privacy header to "off". This is also the default value, unless provisioned otherwise.

The first non-100 response to the INVITE, OPTIONS, REGISTER or extension method request received by the UAC through its trusted proxy MAY contain one or more Remote-Party-ID header fields. A Remote-Party-ID with party type "called" will identify the called party. If such a Remote-Party-ID header field either does not contain an rpi-screen parameter, or it contains an rpi-screen parameter with the value "no" (this includes the case where both "yes" and "no" is provided), the UAC SHOULD NOT trust the identity information provided. An end-to-end encrypted Remote-Party-ID header field can of course also not be trusted, regardless of the value of the rpi-screen parameter. It should be noted, that the rpi-screen parameter provides a somewhat weak form of trustworthiness. In particular, it depends on transitive trust as well as correct implementation, configuration and support for the extensions defined here in the associated chain of trust. Should any of these dependencies not hold, the value "yes" may actually not be trustworthy. Future extensions to SIP may define a general and more robust mechanism that can be used here.

Subsequent responses to the requests MAY also contain Remote-Party-ID header fields. Such Remote-Party-ID header fields with party type "called" identify other parties to which the session has been directed, for whatever reason.

Remote-Party-ID headers contained in the final response, with rpi-pty-type set to "called" identify the party which provided the final answer. In the case of an INVITE response, this identifies the answering party. Again, end-to-end encrypted Remote-Party-ID header fields can not be trusted.

7.2 Trusted UAC Behavior

When a trusted UAC supporting this extension sends an INVITE, OPTIONS, REGISTER or extension method request, and it knows the identity of the calling party, the UAC SHOULD include a calling subscriber Remote-Party-ID header in the request in order to identify the originator of the call. However, if the request is part of an existing dialog, and the request is sent directly to the UAS, then the UAC MAY omit the calling subscriber Remote-Party-ID header. The Remote-Party-ID header MUST at a minimum contain an "addr-spec" to uniquely identify the calling subscriber. The "addr-spec" SHOULD be the same string as appears in the Request-URI for incoming call attempts. The Remote-Party-ID SHOULD include an rpi-pty-type set to "calling" and an rpi-id-type set to "subscriber" - we refer to this

as a calling subscriber Remote-Party-ID. The rpi-screen parameter SHOULD be set to "yes". The Remote-Party-ID MAY optionally include a "display-name" which SHOULD be set to a name that the trusted UAC has associated with the calling subscriber, e.g. the subscriber's

full name. The UAC MAY include other Remote-Party-ID information as well.

If the UAC desires privacy for the Remote-Party-ID header fields it added, it MUST include an rpi-privacy parameter with each relevant Remote-Party-ID. The rpi-privacy parameter MUST specify the desired level of privacy, e.g. "uri", to maintain privacy of the "addr-spec".

If the UAC wants to control the privacy for any Remote-Party-ID header that might be added by a downstream proxy, the UAC MUST furthermore include one or more RPID-Privacy headers indicating the desired level of privacy. Each such RPID-Privacy header MUST include an rpi-privacy parameter specifying the desired level of privacy, e.g. "uri", to maintain privacy of the "addr-spec".

If the UAC indicates "name" or "full" privacy (in either Remote-Party-ID or RPID-Privacy), the UAC MUST NOT reveal the originating subscriber's name in the "display-name" portion of any other header than Remote-Party-ID. This can be achieved by either not providing a "display-name" or setting the "display-name" to "Anonymous" in such fields, e.g. From and Contact.

If the UAC desires "uri" or "full" privacy, the UAC MUST NOT reveal the subscriber's identity in any other header field than Remote-Party-ID. In particular, the contents of header fields needs to be considered as described for untrusted UACs (Section 7.1).

The UAC may also explicitly request that privacy is not to be provided for a Remote-Party-ID by setting the rpi-privacy parameter to "off". This is also the default value, unless provisioned otherwise.

When privacy is requested for one or more Remote-Party-ID headers, the UAC MUST ensure that such privacy is provided prior to forwarding the message to an untrusted entity. Two different options for achieving this are defined here:

1) Do not provide the privacy until the message is forwarded to an untrusted entity.
2) Provide the privacy before forwarding the message, irrespective of whether the next hop is trusted or not.

We first describe option 1, which has the benefit of leaving the Remote-Party-ID in clear as long as possible at the expense of introducing a Proxy-Require "privacy":

If privacy was requested, and the next hop is trusted, the UA MUST ensure that a Proxy-Require header with an option-tag of "privacy" is present. This will ensure that a downstream proxy will apply the

necessary privacy prior to forwarding the message to an untrusted entity. Should a 420 response listing "privacy" as an unsupported option be returned, then privacy can not be provided for this call.

The UA MUST then either initiate a new session without requiring privacy, or the session initiation attempt MUST be abandoned. Furthermore, the UA MUST take precautions to protect the identity information from eavesdropping and interception, e.g., by use of IPSec.

If the UA forwards the request to an untrusted entity, and privacy was requested, the UA MUST ensure the privacy requested will be honored. For each Remote-Party-ID requesting privacy, the UAC MUST do the following:

* If rpi-privacy contains the value "full" or "uri", the UA MUST replace the "addr-spec" in that Remote-Party-ID header field with a private "addr-spec" containing a SIP-URI or a SIPS-URI. The private "addr-spec" MUST list the UA itself in the hostport and include a "user=private" user-param.

* If rpi-privacy contains the value "full" or "name", the UA MUST delete the "display-name" in that Remote-Party-ID header field.

Generation of the userinfo part of a private "addr-spec" is a UA internal issue, as long as the requested privacy is honored and the ability to trace the originator is preserved. However, it is RECOMMENDED to construct the user part by including:

* the initial "addr-spec",

* the value of rpi-privacy, and

* sufficient checksum information to prevent tampering by an untrusted entity.

All of this information MUST then be encoded or encrypted such that the next hop is unable to discern the original Remote-Party-ID. It is RECOMMENDED that the string be encrypted with a symmetric private key, and converted to a printable string using Base64 encoding. The UA MAY include other information in the userinfo part as well.

Prior to forwarding the request to an untrusted entity, the UA SHOULD remove any "privacy" option tag present in a Proxy-Require header field to prevent unnecessary failure of the request if downstream proxies do not support this extension.

We now describe the second option for providing Remote-Party-ID privacy. With this option, the UA applies the same processing for each Remote-Party-ID as in option 1, however it does so regardless of whether the next hop is trusted or not. Since the privacy has now been applied, there is no need to insert a Proxy-Require "privacy". However, there is also no well-defined way for a downstream (trusted) entity to determine the identity of the calling party,

without that entity knowing both the details of how the private "addr-spec" was constructed (crypto algorithm, MAC, encoding, etc.) as well as which key to use for decrypting the information. The

solutions to these problems are left as an exercise to the reader, and hence interoperability should not be expected.

The first non-100 response received by the UAC MAY contain one or more Remote-Party-ID header fields. A Remote-Party-ID with party type "called" will identify the called party. If the response was received via a trusted entity, and the Remote-Party-ID header field either does not contain an rpi-screen parameter, or it contains an rpi-screen parameter with the value "no" (this includes the case where both "yes" and "no" is provided), the UAC SHOULD NOT trust the identity information provided. An end-to-end encrypted Remote-Party-ID header field can of course also not be trusted, regardless of the value of the rpi-screen parameter.

Subsequent responses received by the UAC MAY also contain Remote-Party-ID header fields. Such Remote-Party-ID header fields with party type "called" identify other parties to which the request has been directed, for whatever reason.

Remote-Party-ID headers contained in the final response, with rpi-pty-type set to "called" identify the party which provided the final answer. In the case of an INVITE response, this identifies the answering party. Again, end-to-end encrypted Remote-Party-ID header fields can not be trusted.

7.3 Untrusted UAS Behavior

An untrusted UAS supporting this extension and receiving an INVITE, OPTIONS, REGISTER or extension method request via its trusted proxy looks for a Remote-Party-ID header field with rpi-pty-type "calling" and rpi-id-type "subscriber", i.e., a calling subscriber Remote-Party-ID, to identify the originator of the request. If rpi-pty-type is omitted from a Remote-Party-ID in the request, "calling" is assumed, and if rpi-id-type is omitted, "subscriber" is assumed. If a calling subscriber Remote-Party-ID either does not contain an rpi-screen parameter or it contains an rpi-screen parameter with a value of "no" (this includes the case where both "yes" and "no" is provided), the UAS SHOULD NOT trust the identity information provided. An end-to-end encrypted Remote-Party-ID header field can of course also not be trusted, regardless of the value of the rpi-screen parameter. Otherwise, the UAS SHOULD use the information provided to identify the calling party rather than any information provided in the From or any other header field. Note that the request MAY contain other Remote-Party-ID header fields.

If the UAS wants to control the privacy for any Remote-Party-ID header that might be added to its response by an upstream proxy, the UAS MUST include one or more RPID-Privacy headers indicating the desired level of privacy. Each such RPID-Privacy header MUST include

an rpi-privacy parameter specifying the desired level of privacy, e.g., "uri" to maintain privacy of the "addr-spec".

7.4 Trusted UAS Behavior

A trusted UAS supporting this extension and receiving an INVITE, OPTIONS, REGISTER or extension method request from a trusted entity looks for a Remote-Party-ID header field with rpi-pty-type "calling" and rpi-id-type "subscriber", i.e. a calling subscriber Remote-Party-ID, to identify the originator of the request. If rpi-pty-type is omitted from a Remote-Party-ID in the request, "calling" is assumed, and if rpi-id-type is omitted, "subscriber" is assumed. If a calling subscriber Remote-Party-ID either does not contain an rpi-screen parameter or it contains an rpi-screen parameter with a value of "no" (this includes the case where both "yes" and "no" is provided), the UAS SHOULD NOT trust the identity information provided. An end-to-end encrypted Remote-Party-ID header field can of course also not be trusted, regardless of the value of the rpi-screen parameter. Otherwise, the UAS SHOULD use the information provided to identify the calling party rather than any information provided in the From or any other header field. Note that the request MAY contain other Remote-Party-ID header fields.

If the trusted UAS knows the identity of the party that was reached, it SHOULD include a called subscriber Remote-Party-ID identifying the called party in the first non-100 response. However, if the request was part of an existing dialog, and the request was sent directly to the UAS, then the UAS MAY omit the called subscriber Remote-Party-ID header from the response. In addition, the UAS MAY insert Remote-Party-ID headers in any further non-100 responses. The UAS SHOULD insert a new called subscriber Remote-Party-ID header if the called party information changed from the called party information sent in the previous response. For each of these, the party information SHOULD be set to "called" and the identity information SHOULD be set to "subscriber". Otherwise, the rules for the Remote-Party-ID are similar to those for the INVITE, OPTIONS, REGISTER or extension method request sent by a trusted UAC. Additional Remote-Party-ID header fields MAY be provided as well.

If the UAS desires privacy for a Remote-Party-ID header field it added, it MUST include a privacy request indication in that Remote-Party-ID header. Note that the privacy request is only guaranteed to be satisfied if the previous hop is trusted and it furthermore supports the extensions defined here. If the UAS cannot guarantee both, then any privacy desired MUST be provided before the response is forwarded upstream. Alternatively, the UAS MAY simply omit Remote-Party-ID's requiring privacy from the response.

7.5 Proxy Behavior

When a proxy supporting this extension receives an INVITE, OPTIONS, REGISTER or extension method request from a trusted entity, it does not apply any special processing until the message is forwarded to the next hop. If the message instead came from an untrusted entity, the proxy MUST do the following:

First, the proxy MUST examine the message for the presence of any Remote-Party-ID headers. Since the request was received from an untrusted entity, each of these MUST either be verified by the proxy or have their rpi-screen parameter set to "no". If the proxy is able to successfully verify the information in a Remote-Party-ID header field (by means outside the scope of this document), the proxy MUST add an rpi-screen parameter set to "yes" for that Remote-Party-ID. Furthermore, this MUST be the only rpi-screen parameter for that Remote-Party-ID. If verification fails however, further processing depends on the reason for the failure. Two different failure reasons are defined here:

* The information provided could not be verified because the proxy does not support verification of the identity information for this particular Remote-Party-ID.

* The proxy supports verification of this particular Remote-Party-ID, however the identity information provided is incorrect and the proxy detected that, or the identity information could not be verified.

In the first case, the proxy MUST add an rpi-screen parameter set to "no". The proxy SHOULD furthermore ensure this is the only rpi-screen parameter. In the second case, the proxy MUST by default add an rpi-screen parameter set to "no" and ensure this is the only rpi-screen parameter, however individual extensions and local procedures MAY specify a different behavior, for example rewrite or removal of the offending Remote-Party-ID header field.

Second, if the proxy knows the identity of the calling party (by means outside the scope of this document), and there is no corresponding calling subscriber Remote-Party-ID header field present in the request, the proxy SHOULD include a calling subscriber Remote-Party-ID with the request in order to identify the originator of the request. The Remote-Party-ID header MUST at a minimum contain an "addr-spec" to uniquely identify the calling subscriber. The "addr-spec" SHOULD be the same as appears in the Request-URI for incoming call attempts to that party. The Remote-Party-ID SHOULD include an rpi-pty-type set to "calling" and an rpi-id-type set to "subscriber". The rpi-screen parameter SHOULD be set to "yes". The Remote-Party-ID MAY optionally include a "display-

name" which SHOULD be set to a name that the proxy has associated with the calling subscriber, e.g., the subscriber's full name. The proxy MAY include other Remote-Party-ID information as well.

If the proxy is unable to determine the identity of the calling party, it MAY alternatively reject the request, e.g. with a 403 or 407. The details of this is outside the scope of this document.

If the proxy added one or more Remote-Party-ID headers to the request, the proxy MUST look for the presence of any RPID-Privacy header fields and set the rpi-privacy parameter on the Remote-Party-ID headers the proxy added accordingly (see Section 6.2). If there were no RPID-Privacy headers present, but the From header field contained the value "Anonymous" as the display-name, the proxy MUST apply "full" privacy to all Remote-Party-ID headers it added - this ensures backwards compatibility with current SIP. Note however, that the proxy does not check the validity of a display-name provided in the From header field.

The proxy is now ready to forward the message. If there are no Remote-Party-ID headers requesting privacy, the message is simply forwarded. However, if there is a request for some kind of privacy, the proxy MUST apply the same processing as a trusted UAC would (see Section 7.2). In particular, the proxy MUST ensure that any privacy requested is provided prior to forwarding the message to an untrusted entity - refer to Section 7.2 for details.

When the proxy receives a response to the INVITE, OPTIONS, REGISTER or extension method request from a trusted entity, it does not apply any special processing until the message is forwarded to the next hop. If the response instead came from an untrusted entity, and it was a non-100 response, the proxy MUST do the following:

First, the proxy examines the response for the presence of any Remote-Party-ID headers and applies similar processing as it did for the request.

Second, if the proxy knows the identity of the party that was reached (by means outside the scope of this document), and there is no corresponding called subscriber Remote-Party-ID header field present in the response, the proxy SHOULD add a called subscriber Remote-Party-ID to the response in order to identify the party reached. The Remote-Party-ID header MUST at a minimum contain an "addr-spec" to uniquely identify the subscriber reached. The "addr-spec" SHOULD be the same string as appears in the Request-URI for incoming call attempts to that party. The Remote-Party-ID SHOULD include an rpi-pty-type set to "called" and an rpi-id-type set to "subscriber". The rpi-screen parameter SHOULD be set to "yes". The Remote-Party-ID MAY optionally include a "display-name" which SHOULD be set to a name that the proxy has associated with the called subscriber, e.g. the subscriber's full name. The proxy MAY include other Remote-Party-ID information as well.

If the proxy is unable to determine the identity of the party reached, it SHOULD continue normal processing, and simply omit adding a called party Remote-Party-ID to the response.

If the proxy added one or more Remote-Party-ID header fields to the response, the proxy MUST look for the presence of any RPID-Privacy header fields in the response and set the rpi-privacy parameter on the Remote-Party-ID headers the proxy added accordingly (see Section 6.2).

The proxy is now ready to forward the response. If there are no Remote-Party-ID headers requesting privacy, the response is simply forwarded upstream. However, if there is a request for some kind of privacy, the proxy MUST apply the same processing as a trusted UAS would (see Section 7.4). In particular, the proxy MUST ensure that any privacy requested is provided prior to forwarding the response to an untrusted entity - refer to Section 7.4 for details. Again, it should be noted, that either type of privacy request is only guaranteed to be satisfied if the previous hop is trusted and it furthermore supports the extensions defined here. If the proxy cannot guarantee both, then any privacy desired MUST be provided before the response is forwarded upstream. Alternatively, the proxy MAY simply omit Remote-Party-ID's requiring privacy from the response.

7.6 Additional Proxy and Trusted User Agent Behavior

A proxy or trusted UA supporting this extension SHOULD be prepared to receive a request containing a SIP-URI or SIPS-URI with a user-param set to "private". If the "hostport" part of the URI identifies the entity handling the request, the entity MUST recover the private information. For entities that use the encryption recommendation provided earlier, this implies decrypting the "userinfo" portion of the URI and replacing it with the decrypted addr-spec that was contained in the "userinfo" portion as well as any other SIP information included. Note that the decrypted addr-spec may itself contain a "private" URI.

If the entity is unable to recover a "private" SIP-URI or SIPS-URI, it MUST fail the request with a 4xx error code.

8. Examples of Use

In this Section, we illustrate how the request for privacy may work in practice. It should be noted that the privacy service described can be implemented in a number of ways; we merely describe one possible solution in this section.

8.1 Basic Privacy Example

The Figure below illustrates a basic privacy example scenario:

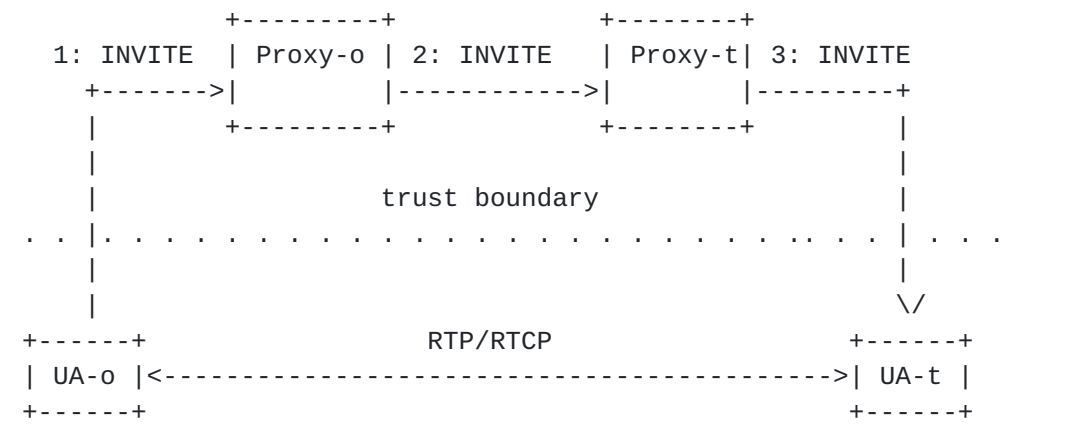

Figure 4 - Basic Privacy Example

The originating user agent (UA-o) sends an INVITE (1) to Proxy-o where it requests uri and name, i.e. full, privacy for any Remote-Party-ID headers that might be added. Since the From header field contains calling identity information, UA-o supplies a cryptographically random identifier for the userinfo, and the non-identifying hostname "localhost" rather than its true identity:

```
INVITE
From:           sip:xyz@localhost
RPID-Privacy:   full
```

Proxy-o determines the calling subscriber identity, and adds a corresponding Remote-Party-ID header to the request. The privacy setting on this header is derived from the RPID-Privacy header present in the INVITE (1) received from the UA. Since proxy-o knows that the identity information in the Remote-Party-ID is correct, Proxy-o also includes an rpi-screen parameter set to "yes". Proxy-o trusts Proxy-t, and hence the Remote-Party-ID can be passed in clear. However, to ensure proper privacy processing, Proxy-o adds a Proxy-Require "privacy" to the request before it sends INVITE(2) to Proxy-t:

```
INVITE
From:           sip:xyz@localhost
Remote-Party-ID: "John Doe" <sip:jdoe@foo.com>;party=calling;
                   id-type=subscriber;privacy=full;screen=yes
Proxy-Require:  privacy
```

When Proxy-t receives the INVITE, it examines the privacy request included in the INVITE and sees that uri and name privacy is requested. Since the next hop is untrusted, Proxy-t therefore removes the "display-name" from the calling subscriber Remote-Party-ID, encrypts the "addr-spec" and rpi-privacy, puts the result in the

"userinfo" part, inserts itself as the "hostport" and adds a "user=private" user-param. Also, Proxy-t removes the Proxy-Require "privacy" before sending the INVITE(3) to UA-t:

```
INVITE
From:            sip:xyz@localhost
Remote-Party-ID: <sip:e(<sip:jdoe@foo.com>;privacy=full
                       )@proxy-t.foo.com;user=private
                  >;party=calling;id-type=subscriber;
                                 privacy=full;screen=yes
```

UA-t notes the presence of the Remote-Party-ID, but since it indicates full privacy, UA-t can only identify the calling subscriber as private, however it knows that the subscriber's identity has been asserted by somebody its proxy trusts, since the rpi-screen parameter is set to "yes". UA-t decides to accept the call setup, and responds with a 180 Ringing. In this case, there is no request for privacy for any Remote-Party-ID headers by upstream proxies, so a normal 180 response is sent back.

Proxy-t determines the identity of UA-t and adds a corresponding Remote-Party-ID as well as an rpi-screen parameter set to "yes". Since no privacy was requested, proxy-t can provide the Remote-Party-ID information to proxy-o in clear:

```
180
Remote-Party-ID: "Mary Doe" <sip:mdoe@foo.com>;party=called;
                                 id-type=subscriber;screen=yes
```

Proxy-o forwards the response to UA-o as is.

While this illustrates the basic operation of the service, there are additional issues that need to be considered. In SIP, there are several fields that can reveal the identity of the calling party, either in part or completely. Other protocols used, e.g. SDP and RTP may reveal identity information as well. A user agent wishing to not reveal its identity should consider each of these. The next example looks more closely at this.

8.2 Complete Privacy Example

The second example we look at is one where IP-address privacy is requested. Note that we do not actually define an IP address privacy mechanism in this document, however the example below illustrates how IP address privacy could be provided and the underlying considerations behind providing it.

The Figure below illustrates how IP address privacy can be achieved by inserting a trusted intermediary, an anonymizer, for the signaling and the media streams between UA-o and UA-t. The interface between the proxies and the media anonymizer is purposely not defined here:

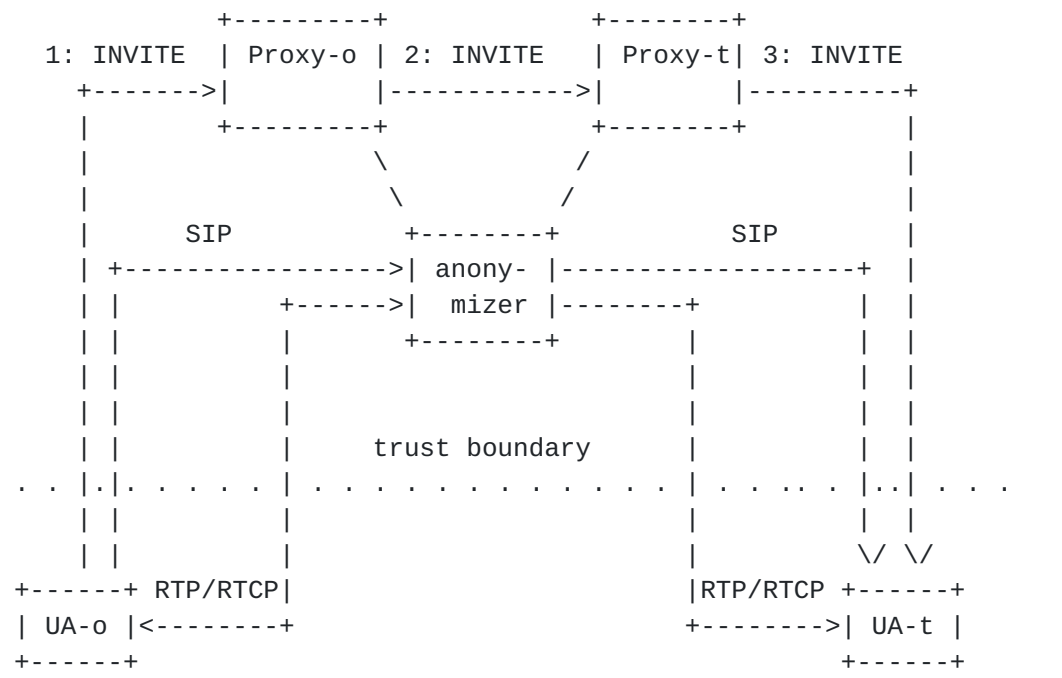

Figure 5 - Full Privacy Example

For all signaling and media exchange purposes, the anonymizer adds a level of indirection thereby hiding the IP address(es) of UA-o from UA-t. This indirection is used both for the media streams and SIP signaling, beyond the initial INVITE, exchanged directly between UA-o and UA-t. A further refinement into IP address privacy just for signaling and IP address privacy just for media streams could be provided as well.

In order to provide IP address privacy for the SIP signaling, we need to consider the header fields which may reveal IP address information. These header fields and their IP address privacy considerations are:

* The From header field must use the non-identifying host name "localhost".

* The Call-ID UAC must not be based on the originator's IP address.

* A Contact header field must be set to point to the anonymizer to prevent any direct signaling between UA-o and UA-t.

* An Alert-Info, Call-Info, Error-Info, or Reply-To header field should either be omitted or not point to any location that can help identify UA-o.

* Via, Record-Route, Route, and any other header fields identifying either UA-o or Proxy-o must be hidden, e.g., by encryption or

simple stateful removal and re-insertion by Proxy-t and/or the anonymizer.

Note that in addition to the above SIP header fields, extensions may define additional header fields which compromise IP address privacy.

An alternative to the media anonymizer function shown above is to implement the anonymizer as a back to back User Agent thereby trivially hiding IP address information in the SIP signaling itself.

Furthermore, when SDP or another session description protocol is used to describe the media in the session, the session descriptions exchanged by the user agents need to be modified to direct the media streams to the anonymizer. The use of SDP fields revealing calling identity information needs to be considered as well. Similar concerns apply to the use of RTCP.

9. Security Considerations

The mechanism provided in this document is a partial consideration of the problem of identity and privacy in SIP. For example, these mechanisms provide no means by which end users can conceal their identities from the network. Additionally, information which the user designates as 'private' can be inspected by any intermediaries participating in the trusted network.

Also, this document does not include means by which a network will authenticate a user, nor means by which intermediaries can authenticate one another, although both of these functions are required for a complete implementation of this system. If the means chosen for this are inherently insecure (for example, blindly using an originating IP address as an assertion that a request came from a certain entity or user), then it will significantly degrade the services provided by this extension.

When a trusted entity has determined the identity information for a given party that wishes to have its identity remain private, and the trusted entity then sends a message to any destination with that party's identity in a Remote-Party-ID header, the entity MUST take precautions to protect the identity information from eavesdropping and interception to protect the confidentiality and integrity of that identity information. The use of transport or network layer hop-by-hop security mechanisms, such as TLS or IPSec, can satisfy this requirement.

As noted above, Remote-Party-ID information that is received can only be trusted if it is received from a trusted entity and it was not encrypted end-to-end. The reason is, that end-to-end encryption of a Remote-Party-Id will prevent the trusted intermediaries from setting the rpi-screen parameter correctly.

Finally, a user agent or proxy can only assume that a privacy

request will be honored, if it is sent to a trusted entity. Thus, if a user agent or proxy does not know if its SIP message (request or response) is sent to a trusted entity (proxy or UA), it should assume that a privacy request for that message will not be honored.

10. IANA Considerations

10.1 New SIP Header Fields and Option Tag

This document defines two new SIP header fields (without the double-quotes):

* "Remote-Party-ID", and

* "RPID-Privacy"

This document also defines a new option-tag "privacy".

10.2 Remote-Party-ID IANA Registry

Upon publication of this document as a standards track RFC, IANA is instructed to establish a new sub-registry under http://www.iana.org/assignments/sip-parameters and enter the Remote-Party-ID extensions defined below in it. Before any such extension can be registered, the extension MUST be documented in an RFC and be reviewed by a designated expert [RFC2434] for applicability to SIP and the Remote-Party-ID as defined in this document. As part of this review, the expert will verify that the extension complies with the applicability defined for this document (see Section 2). The expert reviewer will send email to the IESG on the overall review determination.

Any reviewed and approved Remote-Party-ID extensions defined SHALL be registered with IANA as follows:

* rpi-id-type: A literal name MUST be provided.

* rpi-pty-type: A literal name MUST be provided.

* rpi-privacy: A new privacy indication or a new privacy postfix can be defined - each of these have a separate name space:

  * privacy indication: A literal name MUST be provided.

  * privacy postfix: A literal name MUST be provided.

* other-rpi-token: A literal name, which MUST NOT start with the dash character ("-"), MUST be provided. If the extension is on the form "type = value", then a description of the permissible values SHOULD furthermore be provided.

* privacy-tag: A literal name MUST be provided.

10.3 New Remote-Party-ID "other-rpi-token"

This document defines a new Remote-Party-ID "other-rpi-token" (without the double-quotes):

* "np" for Nature of Party.

The definition of this "other-rpi-token" is provided in Appendix A.

11. Notice Regarding Intellectual Property Rights

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

12. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

3 Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., E. Schooler, "SIP: Session Initiation Protocol", Work in Progress, February 21, 2002.

4 Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and

Demon Internet Ltd., November 1997.

13. Acknowledgments

The basis of this document is the Distributed Call Signaling work in the PacketCable project, which is the work of a large number of people, representing many different companies. The authors would like to recognize and thank the following for their assistance: John Wheeler, Motorola; David Boardman, Daniel Paul, Arris Interactive; Bill Blum, Jon Fellows, Jay Strater, Jeff Ollis, Clive Holborow, Motorola; Doug Newlin, Guido Schuster, Ikhlaq Sidhu, 3Com; Jiri Matousek, Bay Networks; Farzi Khazai, Nortel; John Chapman, Bill Guckel, Michael Ramalho, Cisco; Chuck Kalmanek, Doug Nortz, John Lawser, James Cheng, Tung-Hai Hsiao, Partho Mishra, AT&T; Telcordia Technologies; and Lucent Cable Communications. Additionally, the authors would like to thank the SIP working group, and in particular the following individuals who all made significant contributions to this document: Jonathan Rosenberg, Igor Slepchin, Michael Thomas, Dean Willis, and Rohan Mahy. Alan Johnston provided the "nature of party" extension in Appendix A.

14. Authors' Addresses

Bill Marshall
AT&T
Florham Park, NJ 07932
Email: wtm@research.att.com

K. K. Ramakrishnan
TeraOptic Networks
Sunnyvale, CA
Email: kk@teraoptic.com

Ed Miller
Terayon
Louisville, CO 80027
Email: E.Miller@terayon.com

Glenn Russell
CableLabs
Louisville, CO 80027
Email: G.Russell@Cablelabs.com

Burcak Beser
Juniper Networks
Sunnyvale, CA
Email: burcak@juniper.net

Mike Mannette
3Com
Rolling Meadows, IL 60008

Email: Michael-Mannette@3com.com

Kurt Steinbrenner
3Com
Rolling Meadows, IL 60008
Email: Kurt-Steinbrenner@3com.com

Dave Oran
Cisco
Acton, MA 01720
Email: oran@cisco.com

Flemming Andreasen
Cisco
Edison, NJ
Email: fandreas@cisco.com

John Pickens
Com21
San Jose, CA
Email: jpickens@com21.com

Poornima Lalwaney
Nokia
San Diego, CA 92121
Email: poornima.lalwaney@nokia.com

Jon Fellows
Motorola
San Diego, CA 92121
Email: jfellows@gi.com

Doc Evans
D. R. Evans Consulting
Boulder, CO 80303
Email: n7dr@arrl.net

Keith Kelly
NetSpeak
Boca Raton, FL 33587
Email: keith@netspeak.com

Mark Watson
Nortel Networks
Maidenhead, UK
Email: mwatson@nortelnetworks.com

Appendix A: Nature of Party

This document defines a new "other-rpi-token" to identity the nature of the party in the Remote-Party-id. The Remote-Party-ID Nature of Party information (rpi-np) is supplied on a "token = value" form as defined by the following grammar:

```
rpi-np = "np" "=" ("ordinary" | "residential" | "business" |
                   "priority" | "hotel" | "failure" | "hospital" |
                   "prison" | "police" | "test" | "payphone" |
                   "coin" | "payphone-public" | "payphone-private" |
                   "coinless" | "restrict" | "coin-restrict" |
                   "coinless-restrict" | "reserved" | "operator" |
                   "trans-freephone" | "isdn-res" | "isdn-bus" |
                   "unknown" | "emergency" | "not-applicable" |
                   "cellular-ordinary" | "cellular-roaming" | token )
```

The rpi-np describes the nature of the party identified - additional values can be defined as extensions. Typically, this information will come from ANI information digits (II) which are used in the Public switched network in the US. Information digits are two digit codes that precede the Called Party Number and provide information to the Exchange carriers and IECs about the type of line that originated the call or any special characteristics of the Billing number. In the non-US environments, a parameter called the Calling Party Category (CPC) usually plays the role of information digits in the US and provides similar information. The rpi-np is meant to be primarily used in the PSTN to SIP direction. It is not intended to be used in the SIP to PSTN direction. Mapping to information digits towards the PSTN might invoke unintended results in the PSTN.

The following example illustrates the use of the Nature of Party:

```
Remote-Party-ID: "Mary Doe" <sip:mdoe@foo.com>;party=called;
                 id-type=subscriber;np=ordinary;screen=yes
```

Below is the recommended mapping from II to rpi-np:

00	ordinary
01	not-applicable
02	failure
06	not-applicable
07	not-applicable
20	not-applicable
23	not-applicable
24	trans-freephone
25	trans-freephone and payphone
27	payphone

29	prison
30-32	not-applicable
34	operator
40-49	depends on the carrier's implementation

52	not-applicable
60	not-applicable
61	cellular-ordinary
62	cellular-ordinary
63	cellular-roaming
66	not-applicable
67	not-applicable
70	payphone
80-89	reserved
93	not-applicable

Note that a particular II value could map to two different values of the rpi-np. For example, II value of 25 can map to rpi-np=trans-freephone and rpi-np=payphone.

Full Copyright Statement