Network Working Group
Request for Comments: 3323
Category: Standards Track

J. Peterson
Neustar

November 2002

A Privacy Mechanism for the Session Initiation Protocol (SIP)

Status of this Memo

Copyright Notice

Abstract

This document defines new mechanisms for the Session Initiation Protocol (SIP) in support of privacy. Specifically, guidelines are provided for the creation of messages that do not divulge personal identity information. A new "privacy service" logical role for intermediaries is defined to answer some privacy requirements that user agents cannot satisfy themselves. Finally, means are presented by which a user can request particular functions from a privacy service.

Table of Contents

1. Introduction

This document provides privacy requirements and mechanisms for the Session Initiation Protocol (SIP).

Privacy is defined in this document as the withholding of the identity of a person (and related personal information) from one or more parties in an exchange of communications, specifically a SIP dialog. These parties potentially include the intended destination(s) of messages and/or any intermediaries handling these messages. As identity is defined in this document, withholding the identity of a user will, among other things, render the other parties in the dialog unable to send new SIP requests to the user outside of the context of the current dialog.

In SIP, identity is most commonly carried in the form of a SIP URI and an optional display-name. A SIP address-of-record has a form similar to an email address with a SIP URI scheme (for example, sip:alice@atlanta.com). A display-name is a string containing a name for the identified user (for example, "Alice"). SIP identities of this form commonly appear in the To and From header fields of SIP requests and responses. A user may have many identities that they use in different contexts.

There are numerous other places in SIP messages in which identity-related information can be revealed. For example, the Contact header field contains a SIP URI, one that is commonly as revealing as the address-of-record in the From. In some headers, the originating user agent can conceal identity information as a matter of local policy without affecting the operation of the SIP protocol. However, certain headers are used in the routing of subsequent messages in a dialog, and must therefore be populated with functional data.

The privacy problem is further complicated by proxy servers (also referred to in this document as "intermediaries" or "the network") that add headers of their own, such as the Record-Route and Via headers. Information in these headers could inadvertently reveal something about the originator of a message; for example, a Via header might reveal the service provider through whom the user sends requests, which might in turn strongly hint at the user's identity to some recipients. For these reasons, the participation of intermediaries is also crucial to providing privacy in SIP.

Two complimentary principles have guided the design of this privacy mechanism: Users are empowered to hide their identity and related personal information when they issue requests, but intermediaries and designated recipients of requests are entitled to reject requests whose originator cannot be identified.

The privacy properties of only those specific headers enumerated in the core SIP specification ([1]), as opposed to headers defined by any existing or planned extension, are discussed in this document - however, the privacy mechanisms described in this document can be extended to support extensions.

There are other aspects of the general privacy problem for SIP that are not addressed by this document. Most significantly, the mechanisms for managing the confidentiality of SIP headers and bodies, as well the security of session traffic, are not reconsidered here. These problems are sufficiently well addressed in the baseline SIP specification and related documents, and that no new mechanisms are required.

This document begins with a section that provides a general framework and architecture for privacy in SIP (Section 3), followed by sections that detail user agent behavior (Section 4) and privacy service behavior (Section 5).

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [2] and indicate requirement levels for compliant SIP implementations.

3. Varieties of Privacy

A user may possess many identities that are used in various contexts; generally, identities are addresses-of-record which are bound to particular registrars (operated by the administrators of a domain) with whom SIP user agents register. The operators of these domains may be employers, service providers, or unaffiliated users themselves.

When a user voluntarily asserts an identity in a request, they are claiming that they can receive requests sent to that identity in that domain. Strictly speaking, privacy entails the restriction of the distribution of a specific identity and related personal information from some particular party or parties that are potentially recipients of the message. In particular, there are scenarios in which a party desiring anonymity may:

- send a message and withhold an identity from the final destination(s) while still communicating an identity to one or more intermediaries
- send a message and withhold their identity from some or all intermediaries, but still communicate an identity end-to-end to the final destination(s)
- withhold identity from both intermediaries and final destination(s)

The result of withholding an identity is that the parties in question would be unable, for example, to attempt to initiate a new dialog with the anonymous party at a later time. However, the anonymous party still must be capable of receiving responses and new requests during the dialog in which it is participating.

It may be desirable to restrict identity information on both requests and responses. Initially, it might seem unusual to suggest that a response has privacy concerns - presumably, the originator of the request knows who they were attempting to contact, so the identity of the respondent can hardly be confidential. However, some personal information in responses (such as the contact address at which the respondent is currently registered) is subject to privacy concerns and can be addressed by these mechanisms.

3.1 When is Privacy Necessary?

Users may wish for identity information to be withheld from a given party for any number of reasons, for example:

- Users might want to contact a particular party without revealing their identity in order to impart information with which they would not like to be associated
- Users might fear that the exposure of their identity or personal information to some networks or destinations will make them a target for unsolicited advertising, legal censure or other undesirable consequences
- Users might want to withhold from participants in a session the identity by which they are known to network intermediaries for the purposes of billing and accounting

When a user agent decides to send a request through a proxy server, it may be difficult for the originator to anticipate the final destination of that message. For that reason, users are advised not to base their estimation of their privacy needs on where they expect a message will go. For example, if a user sends a request to telephone number, they may believe that the final destination of the request will be a station in the public switched telephone network (PSTN) that is unable to inspect, say, SIP Contact headers, and therefore assume that it is safe to leave such headers in the clear; however, such a request might very well end up being retargeted by the network to a native SIP endpoint to which Contact headers are quite legible.

This document describes three degrees of privacy - one level of user-provided privacy, and two levels of network-provided privacy (header privacy and session privacy). How much privacy does a user need for any given session? Generally, if a user is seeking privacy, they're going to need as much of it as they can get. However, if a user knows of no privacy service, they must be content with user-provided privacy alone. Similarly, if a user knows of an anonymization service that can provide session privacy, but is unable to secure session traffic to prevent the anonymizer from possibly eavesdropping on the session, they might judge the loss of session privacy to be the lesser evil. The user might also be aware of exceptional conditions about the architecture in which the user agent is deployed that may obviate one or more privacy concerns.

A user may not always be the best judge of when privacy is required even under ideal circumstances, and thus privacy may in some architectures by applied by intermediaries without the user's explicit per-message request. By sending a request through intermediaries that can provide a privacy role, the user tacitly permits privacy functions to be invoked as needed.

It is also important that users understand that intermediaries may be unable to provide privacy functions requested by users. Requests for privacy may not be honored due to legal constraints, unimplemented or misconfigured features, or other exceptional conditions.

Note that just as it is the prerogative of a user to conceal their identity, so it must also be the prerogative of proxy servers and other users to refuse to process requests from users whom they cannot identify. Therefore users should not just automatically withhold their identity for all requests and responses - inability to ascertain the identity of the originator of the request will frequently be grounds for rejection. Privacy should only be requested when the user has a need for it.

Further to this point, withholding some information in signaling might not be necessary for all user agents to ensure privacy. For example, user agents may acquire their IP addresses and hostnames dynamically, and these dynamic addresses may not reveal any information about the user whatsoever. In these cases, restricting access to hostnames (as described in Section 4.1.1.3) is unnecessary.

3.2 User-Provided Privacy

There is a certain amount of privacy that a user agent can provide itself. For example, the baseline SIP specification permits a user agent to populate the From header field of a request with an anonymous value. Users can take similar steps to avoid revealing any other unnecessarily identity information in related SIP headers (this is discussed further in Section 4.1.1).

A user may have different privacy needs for a message if it traverses intermediaries rather than going directly end-to-end. A user may attempt to conceal things from intermediaries which are not concealed from the final destination, and vice versa. For example, using baseline SIP mechanisms, a user agent can encrypt SIP bodies end-to-end in order to prevent intermediaries from inspecting them. If a SIP message will not pass through intermediaries, however, this step might not be necessary (i.e., lower-layer security, without the addition of security for SIP bodies, could be sufficient).

Also note that if a dialog goes directly end-to-end between participants, however, it will not be possible to conceal the network addresses of the participants.

3.3 Network-Provided Privacy

If a user is sending a request through intermediaries, a user agent can conceal its identity to only a limited extent without the intermediaries' cooperation. Also, some information can only be concealed from destination endpoints if an intermediary is entrusted to remove it.

For these reasons a user must have a way to request privacy from intermediaries, a means that allows users both to signal some indications of the desired privacy services, and to ensure that their call is routed to an intermediary that is capable of providing these services. A user may be aware of a specific third-party anonymizing host, one with which they have a pre-existing relationship, or a user may request that their local administrative domain provide privacy services.

Intermediaries may also be empowered to apply privacy to a message without any explicit signaling from the originating user, since user agents may not always be cognizant or capable of requesting privacy when it is necessary.

4. User Agent Behavior

There are three different ways that a user agent can contribute to the privacy of a request - by populating headers with values that reflect privacy requirements, by requesting further privacy services from the network, and by using cryptographic confidentiality to secure headers and bodies. Note that the last of these is outside the scope of this document.

The mechanisms provided in this section assume that a user agent is sufficiently configurable that a user can select header values and provision privacy preferences (ideally on a per-call basis). If this isn't the case, it is possible that a user can route their call through a privacy service that is configured to groom signaling from this user agent in order to provide some of the function described below (see Section 5).

4.1 Constructing Private Messages

Privacy starts with the user agent. The bulk of the steps that are required to conceal private information about the sender of a message are, appropriately enough, the sender's responsibility.

The following SIP headers, when generated by a user agent, can directly or indirectly reveal identity information about the originator of a message: From, Contact, Reply-To, Via, Call-Info, User-Agent, Organization, Server, Subject, Call-ID, In-Reply-To and Warning. Note that the use of an authentication system (such as the SIP Digest authentication method described in [1]) also usually entails revealing identity to one or more parties; for more information see Section 6.

The first and most obvious step is that user agents SHOULD not include any optional headers that might divulge personal information; there's certainly no reason for a user seeking privacy to include a Call-Info. Secondly, the user SHOULD populate URIs throughout the message in accordance with the guidelines given in Section 4.1.1. For example, users SHOULD create an anonymous From header field for the request. Finally, users MAY also need to request certain privacy functions from the network, as described in Section 4.2.

The Call-ID header, which is frequently constructed in a manner that reveals the IP address or hostname of the originating client, requires special mention. User agents SHOULD substitute for the IP address or hostname that is frequently appended to the Call-ID value a suitably long random value (the value used as the 'tag' for the From header of the request might even be reused).

Note that if the user wants to conceal any of the above headers from intermediaries alone, without withholding them from the final destination of the message, users MAY also place legitimate values for these headers in encapsulated 'message/sip' S/MIME bodies as described in Section 23 of [1].

4.1.1 URIs, Display-Names and Privacy

A certain amount of privacy can be afforded by choosing to populate SIP headers with URIs and display-names that do not reveal any identity information. In some of the header fields (for example, the Reply-To and From headers), URIs are not used in further signaling within the current dialog. In others, like the Contact header, an inaccurate URI will result in a failure to route subsequent requests within the dialog.

4.1.1.1 Display-Names

It is a relatively common practice in email and other applications to use an assumed name in the display-name component of the From header field. Outside of a business context (especially in applications such as instant messaging or Internet gaming) the use of such aliases is unlikely to provide a cause for distrust.

It is RECOMMENDED that user agents seeking anonymity use a display-name of "Anonymous".

4.1.1.2 URI Usernames

The structure of a URI itself can reveal or conceal a considerable amount of personal information. Consider the difference between:

sip:jon.peterson@neustar.biz

and

sip:a0017@anonymous-sip.com

From the former, the full name and employer of the party in question can easily be guessed. From the latter, you learn nothing other than that the party desires anonymity. In some cases, sufficient anonymity can be achieved by selecting an oblique URI. Today, the SIP specification recommends a URI with "anonymous" in the user portion of the From header.

In some URIs, such as those that appear in Contact headers, it MAY also make sense to omit the username altogether, and provide only a hostname, like: sip:anonymous-sip.com

4.1.1.3 URI Hostnames and IP Addresses

It is assumed by this document that the user that requests privacy wishes to receive future requests and responses within this dialog, but does not wish to reveal an identity that could be used to send new requests to him outside the scope of this dialog. For that reason, different treatment must be recommended for URIs that are used in the context of routing further requests in the dialog, as opposed to routing new requests outside the context of the dialog.

For headers indicating how the user would like to be contacted for future sessions (such as the From header), it might not immediately be obvious why changing the hostname would be necessary - if the username is 'anonymous', requests will not be routable to the anonymous user.

Sometimes, merely changing the username will not be enough to conceal a user's identity. A user's SIP service provider might decisively reveal a user's identity (if it reflected something like a small company or a personal domain). So in this case, even though the URI in the From header would not dereference to the anonymous user, humans might easily guess the user's identity and know the proper form of their address-of-record.

For these reasons, the hostname value 'anonymous.invalid' SHOULD be used for anonymous URIs (see [3] for more information about the reserved 'invalid' DNS TLD). The full recommended form of the From header for anonymity is (note that this From header, like all others, MUST contain a valid and unique 'tag=' parameter):

From: "Anonymous" <sip:anonymous@anonymous.invalid>;tag=1928301774

For headers indicating how further requests in the current dialog should be routed (namely the Contact header, Via header, and session information in the SDP), there seems to be little that a user can do to disguise the existing URI, because users MUST provide a value that will allow them to receive further requests. In some cases, disguising or failing to provide the username, as described above, may create some level of privacy, but the hostname provides a more significant obstacle.

Is there much additional privacy in using an IP address rather than a hostname? It does prevent someone who casually inspects a message from gathering information that they might see otherwise. However, reverse-resolving such addresses is generally trivial, and substituting an IP address for a hostname could introduce some complications, for example due to NAT and firewall traversal concerns. Headers used in routing may also rely on certain DNS practices to provide services that would be lost if an IP address is used in place of a hostname.

This document thus recommends that the host portion of URIs that are used in the routing of subsequent requests, such as URIs appearing in the Contact header, SHOULD NOT be altered by the user agent due to privacy considerations. If these headers require anonymization, the user requests that service from an intermediary, namely a privacy service.

Note that many of the considerations regarding the Contact header above apply equal well to SIP headers in which a hostname, rather than a URI, is used for some routing purpose (namely the Via header).

4.2 Expressing Privacy Preferences

There are some headers that a user agent cannot conceal itself, because they are used in routing, that could be concealed by an intermediary that subsequently takes responsibility for directing messages to and from the anonymous user. The user agent must have some way to request such privacy services from the network. For that purpose, this document defines a new SIP header, Privacy, that can be used to specify privacy handling for requests and responses.

```
Privacy-hdr  =  "Privacy" HCOLON priv-value *(";" priv-value)
priv-value   =   "header" / "session" / "user" / "none" / "critical"
                 / token
```

User agents SHOULD include a Privacy header when network-provided privacy (as described in Section 3.3) is required. Note that some intermediaries may also add the Privacy header to messages, including privacy services. However, such intermediaries SHOULD only do so if they are operating at a user's behest, for example if a user has an administrative arrangement with the operator of the intermediary that it will add such a Privacy header. An intermediary MUST NOT modify the Privacy header in any way if the 'none' priv-value is already specified.

The values of priv-value today are restricted to the above options, although further options can be defined as appropriate (see Section 7). Each legitimate priv-value can appear zero or one times in a Privacy header. The current values are:

header: The user requests that a privacy service obscure those headers which cannot be completely expunged of identifying information without the assistance of intermediaries (such as Via and Contact). Also, no unnecessary headers should be added by the service that might reveal personal information about the originator of the request.

session: The user requests that a privacy service provide anonymization for the session(s) (described, for example, in a Session Description Protocol [5] body) initiated by this message. This will mask the IP address from which the session traffic would ordinarily appear to originate. When session privacy is requested, user agents MUST NOT encrypt SDP bodies in messages. Note that requesting session privacy in the absence of any end-to-end session encryption raises some serious security concerns (see Section 5.2).

user: This privacy level is usually set only by intermediaries, in order to communicate that user level privacy functions (as discussed in Section 5.3) must be provided by the network, presumably because the user agent is unable to provide them. User agents MAY however set this privacy level for REGISTER requests, but SHOULD NOT set 'user' level privacy for other requests.

none: The user requests that a privacy service apply no privacy functions to this message, regardless of any pre-provisioned profile for the user or default behavior of the service. User agents can specify this option when they are forced to route a message through a privacy service which will, if no Privacy header is present, apply some privacy functions which the user does not desire for this message. Intermediaries MUST NOT remove or alter a Privacy header whose priv-value is 'none'. User agents MUST NOT populate any other priv-values (including 'critical') in a Privacy header that contains a value of 'none'.

critical: The user asserts that the privacy services requested for this message are critical, and that therefore, if these privacy services cannot be provided by the network, this request should be rejected. Criticality cannot be managed appropriately for responses.

When a Privacy header is constructed, it MUST consist of either the value 'none', or one or more of the values 'user', 'header' and 'session' (each of which MUST appear at most once) which MAY in turn be followed by the 'critical' indicator.

The following table specifies extensions to Table 2 in [1].

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Privacy		amrd	o	o	o	o	o	o

Header field	SUB	NOT	PRK	IFO	UPD	MSG
Privacy	o	o	o	o	o	o

4.3 Routing Requests to Privacy Services

The most obvious way for a user agent to invoke the privacy function is to direct a request through an intermediary known to act as a privacy service. Doing so traditionally entails the configuration of pre-loaded Route headers that designate the privacy service.

It is RECOMMENDED that service providers couple the privacy service function with a local outbound proxy. Users can thereby send their messages that request privacy through their usual outbound route. Users should not assume, however, that the administrative domain that is the destination of the request would be willing and able to perform the privacy service function on their behalf. If the originating user wishes to keep their local administrative domain a secret, then they must use a third-party anonymization service outside of any of the principal administrative domains associated with the session.

It is highly RECOMMENDED that user agents use network or transport layer security, such as TLS, when contacting a privacy service. Ideally, users SHOULD establish a direct (i.e., single pre-loaded Route header) connection to a privacy service; this will both allow the user to inspect a certificate presented by the privacy service, and it will provide confidentiality for requests that will reduce the chances that the information that the privacy service will obscures is revealed before a message arrives at the privacy service. By establishing a direct connection to a privacy service, the user also eliminates the possibility that intermediaries could remove requests for privacy. If a direct connection is impossible, users SHOULD use a mechanism like SIPS to guarantee the use of lower-layer security all the way to the privacy service.

If a user agent believes that it is sending a request directly to a privacy service, it SHOULD include a Proxy-Require header containing a new option-tag, 'privacy', especially when the 'critical' priv-value is present in the Privacy header. That way, in the unlikely event that the user agent sends a request to an intermediary that does not support the extensions described in this document, the request will fail. Note that because of special privacy service

behavior (described in Section 5), no subsequent intermediaries in the signaling path of the request will also need to the support the 'privacy' option-tag - once the privacy service has fulfilled all the required privacy functions, the 'privacy' option-tag is removed from the Proxy-Require header.

4.4 Routing Responses to Privacy Services

Making sure that responses will go through a privacy service is a little bit trickier. The path traversed by SIP responses is the same as the path over which the request traveled. Thus, the responding user agent, for example, cannot force a privacy service to be injected in the response path after it has received a request.

What a responding user agent can do, however, is ensure that the path by which requests reach them traverses their privacy service. In some architectures, the privacy service function will be fulfilled by the same server to which requests for the local administrative domain are sent, and hence it will automatically be in the path of incoming requests. However, if this is not the case, the user will have to ensure that requests are directed through a third-party privacy service.

One way to accomplish this is to procure an 'anonymous callback' URI from the third-party service and to distribute this as an address-of-record. A privacy service provider might offer these anonymous callback URIs to users in the same way that an ordinary SIP service provider grants addresses-of-record. The user would then register their normal address-of-record as a contact address with the third-party service.

Alternatively, a user agent could send REGISTER requests through a privacy service with a request for 'user' level privacy. This will allow the privacy service to insert anonymous Contact header URIs. Requests sent to the user's conventional address-of-record would then reach the user's devices without revealing any usable contact addresses.

Finally, a user might generate a CPL ([7]) script that will direct requests to an anonymization service.

Users are also advised to use transport or network layer security in the response path. This may involve registering a SIPS URI and/or maintaining persistent TLS connections over which their user agent receives requests.

Privacy services MAY in turn route requests through other privacy services. This may be necessary if a privacy service does not support a particular privacy function, but it knows of a peer that does. Privacy services may also cluster themselves into networks that exchange session traffic between one another in order to further disguise the participants in a session, although no specific architecture or method for doing so is described in this document.

5. Privacy Service Behavior

This document defines a new SIP logical role called a "privacy service". The privacy service role is instantiated by a network intermediary, frequently by entities that can act as SIP proxy servers. The function of a privacy service is to supply privacy functions for SIP messages that cannot be provided by user agents themselves.

When a message arrives at a server that can act as a privacy service, the service SHOULD evaluate the level of privacy requested in a Privacy header. Usually, only the services explicitly requested should be applied. However, privacy services MAY have some means outside SIP of ascertaining the preferences of the user (such as a pre-arranged user profile) and therefore they MAY perform such other privacy functions without an explicit Privacy header. Performing even a user-level privacy function in a privacy service could be useful, for example, when a user is sending messages from a legacy client that does support the Privacy header, or a user agent that does not allow the user to configure the values of headers that could reveal personal information. However, if the Privacy header value of 'none' is specified in a message, privacy services MUST NOT perform any privacy function and MUST NOT remove or modify the Privacy header.

Privacy services MUST implement support for the 'none' and 'critical' privacy tokens, and MAY implement any of other privacy levels described in Section 4.2 as well as any extensions that are not detailed in this document. In some cases, the privacy service will not be capable of fulfilling the requested level of privacy. If the 'critical' privacy level is present in the Privacy header of a request, then if the privacy service is incapable of performing all of the levels of privacy specified in the Privacy header then it MUST fail the request with a 500 (Server Error) response code. The reason phrase of the status line of the response SHOULD contain appropriate text indicating that there has been a privacy failure as well as an enumeration of the priv-value(s) which were not supported by the privacy service (the reason phrase SHOULD also respect any Accept-Language header in the request if possible).

When a privacy service performs one of the functions corresponding to a privacy level listed in the Privacy header, it SHOULD remove the corresponding priv-value from the Privacy header - otherwise, any other privacy service involved with routing this message might unnecessarily apply the same function, which in many cases would be undesirable. When the last priv-value (not counting 'critical') has been removed from the Privacy header, the entire Privacy header MUST be removed from a message.

When the privacy service removes the entire Privacy header, if the message is a request, the privacy service MUST also remove any 'privacy' option-tag from the Proxy-Require header field of the request.

5.1 Header Privacy

If a privacy level of 'header' is requested, then the originating user has asked the privacy service to help to obscure headers that might otherwise reveal information about the originator of the request. However, the values that have been so obscured must be recoverable when further messages in the dialog need to be routed to the originating user agent. In order to provide these functions the privacy service must frequently act as a transparent back-to-back user agent (B2BUA).

Firstly, a request for header privacy entails that the server SHOULD NOT add any headers to the message that reveal any identity or personal information, including the following: Call-Info, Server, and Organization. All of these provide optional information that could reveal facts about the user that has request anonymity.

Privacy services operating on requests SHOULD remove all Via headers that have been added to the request prior to its arrival at the privacy service (a practice referred to as "Via stripping") and then SHOULD add a single Via header representing themselves. Note that the bottommost such Via header field value in a request contains an IP address or hostname that designates the originating client, and subsequent Via header field values may indicate hosts in the same administrative domain as the client. No Via stripping is required when handling responses.

Contact headers are added by user agents to both requests and responses. A privacy service SHOULD replace the value of the Contact header in a message with a URI that does not dereference to the originator of the message (such as the anonymous URI described in Section 4.1.1.3). The URI that replaces the existing Contact header field value MUST dereference to the privacy service.

In a manner similar to Via stripping, a privacy service SHOULD also strip any Record-Route headers that have been added to a request before it reaches the privacy service - though note that no such headers will be present if there is only one hop between the originating user agent and the privacy service, as is recommended above. Such Record-Route headers might also divulge information about the administrative domain of the client.

For the purposes of this document, it is assumed that the privacy service has locally persisted the values of any of the above headers that are so removed, which requires the privacy service to keep a pretty significant amount of state on a per-dialog basis. When further requests or responses associated with the dialog reach the privacy service, it MUST restore values for the Via, Record-

Route/Route or Contact headers that it has previously removed in the interests of privacy. There may be alternative ways (outside the scope of this document) to perform this function that do not require keeping state in the privacy service (usually means that involve encrypting and persisting the values in the signaling somehow).

The following procedures are RECOMMENDED for handling the Record-Route header field of requests and responses, which provides specialchallenges to a privacy service:

When a privacy service is processing (on behalf of the originator) a request that contains one or more Record-Route header field values, the privacy service must strip these values from the request and remember both the dialog identifiers and the ordered Record-Route header field values. As described above, it must also replace the Contact header field with a URI indicating itself. When a response with the same dialog identifiers arrives at the privacy service, the privacy service must reapply any Record-Route header field values to the response in the same order, and it must then add a URI representing itself to the Record-Route header field of the response. If the response contains Record-Route header field values of its own, these must also be included (in order) in the Record-Route header field after the URI representing the privacy service.

Note that when a privacy service is handling a request and providing privacy on behalf of the destination of the request, providing privacy for Record-Route headers downstream of the privacy service is significantly more complicated. This document recommends no way of statefully restoring those headers if they are stripped.

5.2 Session Privacy

If a privacy level of 'session' is requested, then the user has requested that the privacy service anonymize the session traffic (e.g., for SIP telephony calls, the audio media) associated with this dialog.

The SIP specification dictates that intermediaries such as proxy servers cannot inspect and modify message bodies. The privacy service logical role MUST therefore act as a back-to-back user agent in order to provide media privacy, effectively terminating and re-originating the messages that initiate a session (although in support of session privacy the privacy service does not need to alter headers characterizing the originator or destination when the request is re-originated). In order to introduce an anonymizer for session traffic, the privacy service needs to control a middlebox [8] that can provide an apparent source and sink for session traffic. The details of the implementation of an anonymizer, and the modifications

that must be made to the Session Description Protocol (SDP [5]) bodies in the messages that initiate a session are outside the scope of this document.

The risk, of course, of using such an anonymizer is that the anonymizer itself is party to your communications. For that reason, requesting session-level privacy without resort to some sort of end-to-end security for the session traffic (with RTP [6] media, for example, SRTP [4]) is NOT RECOMMENDED.

5.3 Applying User-Level Privacy Functions at a Privacy Service

If a privacy level of 'user' is requested, then the originating user has requested that privacy services perform the user-level privacy functions described in Section 4.1.

Note that the privacy service MUST remove any non-essential informational headers that have been added by the user agent, including the Subject, Call-Info, Organization, User-Agent, Reply-To and In-Reply-To.

Significantly, user-level privacy could entail the modification of the From header, changing it from its original value to an anonymous value. Prior to the current issue of the SIP specification, the modification of the values of the To and From headers by intermediaries was not permitted, and would result in improper dialog matching by the endpoints. Currently, dialog matching uses only the tags in the To and From headers, rather than the whole header fields. Thus, under the new rules the URI values in the To and From headers themselves could be altered by intermediaries. However, some legacy clients might consider it an error condition if the value of the URI in the From header altered between the request and the response.

Also, performing user-level privacy functions MAY entail the modification of the Call-ID header, since the Call-ID commonly contains a hostname or IP address corresponding to the originating client. This field is essential to dialog matching, and it cannot be altered by intermediaries.

Therefore, any time that a privacy service needs to modify any dialog-matching headers for privacy reasons, it SHOULD act as a transparent back-to-back user agent, and it MUST persist the former values of the dialog-matching headers. These values MUST be restored in any messages that are sent to the originating user agent.

6. Security Considerations

Messages that request privacy require confidentiality and integrity. Without integrity, the requested privacy functions could be downgraded or eliminated, potentially exposing identity information. Without confidentiality, eavesdroppers on the network (or any intermediaries between the user and the privacy service) could see the very personal information that the user has asked the privacy service to obscure.

All of the network-provided privacy functions in this document entail a good deal of trust for the privacy service. Users should only trust privacy services that are somehow accountable to them.

Operators of privacy services should be aware that in the eyes of downstream entities, a privacy service will be the only source to which anonymous messages can be traced.

Note that authentication mechanisms, including the Digest authentication method described in the SIP specification, are outside the scope of the privacy considerations in this document. Revealing identity through authentication is highly selective, and may not result in the compromise of any private information. Obviously, users that do not wish to reveal their identity to servers that issue authentication challenges MAY elect not to respond to such challenges.

7. IANA Considerations

This document defines a new SIP header field called "Privacy" that allows a user agent to request a certain degree of privacy for a message. This behavior associated with this header is specified in Section 4.2. This header has been added to the header sub-registry under http://www.iana.org/assignments/sip-parameters.

```
 Header name: Privacy
Compact form: none defined
```

This document also creates an IANA registry for values that populate the Privacy header. This registry should be indexed by priv-value tokens and should contain a short semantic description of the new value. The current values of the "Privacy" header are as follows:

- user: Request that privacy services provide a user-level privacy function
- header: Request that privacy services modify headers that cannot be set arbitrarily by the user (Contact/Via).

- session: Request that privacy services provide privacy for session media
- none: Privacy services must not perform any privacy function
- critical: Privacy service must perform the specified services or fail the request

New values for the "Privacy" header can only be defined by IETF Consensus including RFC publication (RFC 2434). IANA registration for the "Privacy" header field values is required along with the RFC publication.

Authors of extensions to the SIP protocol that expose personal information about the participants in sessions are advised against extending the "Privacy" header - rather, it is preferable to create new identity mechanisms whose privacy can be managed by the user agent without the agency of intermediaries.

This document also defines a new SIP option-tag, 'privacy', that represents support for the extension defined in this document.

Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[2] Bradner, S., "Key words for use in RFCs to indicate requirement levels", BCP 14, RFC 2119, March 1997.

[3] Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names", RFC 2606, June 1999.

Informative References

[4] Baugher, M., McGrew, D., Oran, D., Blom, R., Carrara, E., Naslund, M. and K. Normann, "The Secure Real Time Transport Protocol", Work in Progress.

[5] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[6] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[7] Lennox, J. and H. Schulzrinne, "CPL: A Language for User Control of Internet Telephony Services", Work in Progress

[8] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A. and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

Author's Address

Jon Peterson
NeuStar, Inc.
1800 Sutter St
Suite 570
Concord, CA 94520 US

Phone: +1 925/363-8720
EMail: jon.peterson@neustar.biz
URI: http://www.neustar.biz/

Acknowledgments

The author would like to thank Allison Mankin, Rohan Mahy, Eric Rescorla, Mark Watson, Cullen Jennings, Robert Sparks, Jonathan Rosenberg, Ben Campbell, Tom Gray and John Elwell for their comments.

Full Copyright Statement

Acknowledgement

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