SIP Working Group
Internet Draft
Document: <draft-dcsgroup-sip-privacy-02.txt>

W. Marshall
K. Ramakrishnan
AT&T

E. Miller
G. Russell
CableLabs

B. Beser
M. Mannette
K. Steinbrenner
3Com

D. Oran
F. Andreasen
Cisco

J. Pickens
Com21

P. Lalwaney
Nokia

J. Fellows
Motorola

D. Evans
Secure Cable Solutions

K. Kelly
NetSpeak

June, 2000

SIP Extensions for Caller Identity and Privacy

Status of this Memo

1. Abstract

This document describes two extensions to the Session Initiation Protocol (SIP) [4]. The extensions allow callers and callees to maintain their privacy in an environment where one or more proxies serve as intermediaries which can provide the identity of the parties either directly or indirectly. The extensions allow the parties to be identified either by name or by type the latter of which can be used to identify some group of callers and callees.

2. 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].

3. Introduction

In order for SIP to be a viable alternative to the current PSTN, SIP 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, as well as the ability to trace the originator of a call. While 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 From header field will not be identifiable, e.g. for a return call or call trace, by entities more than a single hop away, since the contents of the From header cannot be modified. We note that this problem is not telephony specific but applies to other forms of session initiation as well. Furthermore, the issue of privacy in an IP environment is more complicated than in the PSTN, as 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.

In order to solve the above we assume an architecture as described in [5] , where a SIP User Agent is associated with a trusted proxy, and proxies in turn communicate with other proxies and user agents which may or may not be trusted. Calls utilizing the services of

this architecture must both be placed and received through the trusted proxy.

This document defines two extensions to SIP that allow the calling and called party to be identified by a trusted intermediary while still being able to maintain their privacy. A new general header, Remote-Party-ID, identifies each party, and another new general header, Anonymity, defines the level of privacy requested by the party. The trusted intermediary verifies the Remote-Party-ID information supplied and ensures the privacy requested is provided when forwarding a message across an untrusted boundary.

4. Protocol Overview

UACs that wish to use the extensions defined here MUST include a Proxy-Require header in the initial INVITE request containing the option tag "privacy". When such a UAC makes a call, it SHOULD include a Remote-Party-ID header in the initial INVITE request in order to identify the originator of the call. The Remote-Party-ID MUST contain a SIP-URL identifying the UAC and MAY contain a "display-name" for the UAC as well. Additionally, if privacy is desired, the UAC MUST include an Anonymity header, which can request one or more of URI, Name, and IP address privacy.

When a proxy supporting this extension receives an INVITE from an untrusted entity, it looks for the presence of a Remote-Party-ID header. If one is found, the proxy determines if the previous hop was a UA the proxy serves. If so, the Remote-Party-ID information is verified and modified if needed. If the request instead came from another untrusted entity, the proxy either removes the Remote-Party-ID information or marks it as being untrusted. Alternatively, the proxy MAY reject the request, e.g. with a 403 or 407.

Prior to forwarding the request to an untrusted entity, the proxy MUST look for the presence of an Anonymity header requesting privacy. If one is found, the privacy requested MUST be provided prior to forwarding the request. For URI and Name privacy, this involves encrypting and possibly removing information provided in the Remote-Party-ID. For IP Address privacy, it involves providing a level of indirection for signaling and media through an entity we refer to as an Anonymizer. The Anonymity header is removed as well.

Once a UAS supporting this extension receives the INVITE, it can use the Remote-Party-ID information provided to identify the originator of the call, unless the originator had requested privacy. If the INVITE contained a Proxy-Require with an option tag of "privacy", the UAS SHOULD include a Remote-Party-ID identifying it in the first non-100 response. Irrespective, the UAS MUST include an Anonymity header if it desires any privacy.

When a proxy supporting this extension receives a non-100 response to the initial INVITE, it looks for a Remote-Party-ID header field and applies similar processing as for the initial INVITE with one difference. If the INVITE did not contain a Proxy-Require with an option tag of "privacy", the proxy MUST ensure that any privacy requested in the response is provided prior to forwarding it, irrespective of whether the previous hop is trusted or not.

Finally, when the UAC receives the first non-100 response from the UAS, it can use Remote-Party-ID information provided to identify the terminating party, unless the terminator had requested privacy.

5. Header Field Definitions

Table 1 below is an extension of tables 4 and 5 in [4] for the new headers defined here:

	where	enc.	e-e	ACK	BYE	CAN	INV	OPT	REG
Anonymity	g	n	h	-	-	-	o	-	-
Remote-Party-ID	g	n	h	-	-	-	o	-	-

Table 1: Summary of header fields.

The headers can be used in an INVITE as well as any response to an INVITE.

5.1 Remote-Party-ID Header Field Definitions

The Remote-Party-ID header field provides the identity of the remote party. At the called party it contains the identity of the caller, and at the calling party, it contains the translated identity of the called party. Remote-Party-ID is defined by the following ABNF [3]:

```
Remote-Party-ID    = "Remote-Party-ID" ":" [display-name]
                          "<" addr-spec ">" *(";" rpi-token)

rpi-token          = rpi-screen | rpi-type | other-rpi-token

rpi-screen         = "rpi-screen" "=" ("no" | "yes" )

rpi-type           = "rpi-type" "=" 1#token

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

Furthermore, we define the value "private" for "other-user" in an "addr-spec", to indicate that the user 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 [4].

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 "user" part of the "addr-spec" 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 encrypted.

The "rpi-screen" describes what verification the Remote-Party-ID information has undergone. The value "yes" (assumed by default) indicates the Remote-Party-ID was verified successfully by the proxy itself or the proxy received the INVITE from a trusted proxy with this indication. The value "no" indicates the Remote-Party-ID was either not verified successfully by the proxy or the proxy received the message from an untrusted entity.

The "rpi-type" allows a group of users to be identified by some common denominator. The denominator(s) used as well as the semantics associated with these are a local issue and hence outside the scope of this document. One example use might be to define an "rpi-type" of "operator". An "operator" caller type might request special privileges, e.g. performing an emergency interrupt on a voice call, that the UA might not normally allow. Again, we purposely do not define any particular rpi-types or semantics here.

Finally, the "other-rpi-token" parameter allows Remote-Party-ID to be extended.

5.2 Anonymity Header Field Definition

The Anonymity header field allows an originating SIP user agent to indicate the degree of privacy that should be provided to its session.

The ABNF for the proposed header field follows:

```
Anonymity       = "Anonymity" ":"  1#privacy-tag
privacy-tag     = "full" | "uri" | "name" | "ipaddr" | "off"
```

Comparisons follow the case-sensitivity rules defined by SIP [4].

If privacy is requested, it MUST be one or more of "full", "uri", "name", or "ipaddr". The value "off" indicates that no privacy is requested, and MUST be the only value if present.

The value "uri" requests the party's identity not be provided to the destination. The value "name" requests the party's name not be provided. The value "ipaddr" requests IP privacy such that the

other party does not learn the IP address of this party. The value "full" requests both URI, Name, and IP address privacy.

It should be noted, that the header field allows both the originating and terminating user agent to indicate its desire for privacy.

6. Protocol Semantics

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

6.1 UAC Behavior

When a UAC supporting this extension initiates a call through its trusted proxy, it SHOULD include a Remote-Party-ID header in the initial INVITE request in order to identify the originator of the call. The Remote-Party-ID header MUST at a minimum contain an "addr-spec" to uniquely identify the calling party. The "addr-spec" SHOULD be the same string as appears in the Request-URI for incoming call attempts. The Remote-Party-ID may optionally include a "display-name" and an "rpi-type". The "display-name" SHOULD be a name that the proxy has associated with the calling party, e.g. the subscribers full name. The "rpi-type" can be used as a convenience to identify some group of users.

If the UAC desires privacy for the call, it MUST include an Anonymity header specifying the desired level of privacy, e.g. "uri" to maintain privacy of the "addr-spec". As honoring the privacy requested depends on the proxy, the UAC MUST furthermore include a Proxy-Require header with an option-tag of "privacy".

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

If the UAC desires "uri" or "full" privacy, the UAC MUST NOT reveal the originating subscriber's identity in the SIP-URL in the From header field. The UAC SHOULD instead supply a cryptographically random identifier for the userinfo, and a non-identifying hostname, e.g. "localhost", for the hostport.

If the UAC desires "ipaddr" or "full" privacy, the UAC MUST NOT base the Call-ID on the originator's IP address.

The first non-100 response received by the UAC MAY also contain a Remote-Party-ID identifying the called party. If the Remote-Party-ID contains an "rpi-screen" parameter with a value of "no", the UAC SHOULD NOT trust the validity of the information provided. Otherwise, the UAC SHOULD use the information provided to identify the called-party rather than any information originally put in the

To header field. The "addr-spec" contained in this Remote-Party-ID can be used as the Request-URI by the UAC to initiate certain call control functions or subsequent calls that are required to reference the party reached. Examples of these include call transfer and repeat call.

6.2 UAS Behavior

A UAS supporting this extension and receiving an INVITE from its trusted proxy looks for a Remote-Party-ID header field to identify the originator of the request. If the Remote-Party-ID contains an "rpi-screen" parameter with a value of "no", the UAS SHOULD NOT trust the validity of the information provided. Otherwise, the UAS SHOULD use the information provided to identify the caller rather than any information provided in the From header field.

The "addr-spec" contained in the Remote-Party-ID received can be used as the Request-URI by the UAS to initiate certain call control functions or subsequent calls that are required to reference the party reached. Examples of these include call transfer and return call.

If the initial INVITE contained a Proxy-Require header field with an option tag of "privacy", the UAS SHOULD insert a Remote-Party-ID header field identifying itself into the first non-100 response it sends. The rules for the Remote-Party-ID are similar to those for the initial INVITE for a UAC.

Regardless of whether the UAS provides a Remote-Party-ID in the first non-100 response, the UAS MAY insert an Anonymity header in that response to request any desired called party privacy. It should be noted though, that the UAS can not depend on this privacy being honored, if the original INVITE did not contain a Proxy-Require with an option tag of "privacy".

6.3 Proxy Behavior

When a proxy supporting this extension receives an INVITE from an untrusted entity, the proxy first determines if the request came from a UAC that it serves. If so, it examines the INVITE for the presence of a Remote-Party-ID header field. If a Remote-Party-ID header field is present, the information supplied is verified and, if needed, rewritten. The proxy MUST verify that the "addr-spec" provided is a valid "addr-spec" for that UAC; if not, the proxy MUST rewrite the "addr-spec" with a valid "addr-spec" for that UAC. If "display-name" is provided in Remote-Party-ID, the proxy MUST verify that the "display-name" is a valid string for the UAC; if not or if the "display-name" is omitted, the proxy MUST rewrite the "display-name" with a valid string for the UAC or remove the "display-name". Note, that the proxy does not check a "display-name" provided in the From header field. If "rpi-type" is provided, the proxy MUST verify

that the UAC is of the indicated "rpi-type"(s); if not, the proxy MUST remove the offending "rpi-type"(s) - this includes removing unrecognized "rpi-type"(s).

If a Remote-Party-ID header was not present in the INVITE, but the proxy is able to identify the originating UAC anyway, the proxy inserts a Remote-Party-ID header with the correct information.

If the request instead came from an untrusted entity, and it was not a UAC the proxy serves or the proxy is unable to identify the entity, the proxy MUST either remove any Remote-Party-ID header or add "rpi-screen=no" before the request is forwarded. Alternatively, the proxy MAY reject the request, e.g. with a 403 or 407.

The proxy furthermore looks for the presence of an Anonymity header. If an Anonymity header is present and the next hop is trusted, the proxy MUST ensure that a Proxy-Require header with an option-tag of "privacy" is present.

If the proxy forwards the request to an untrusted entity, and the Anonymity header is present, the proxy MUST remove the Anonymity header and ensure the privacy requested will be honored.

For non IP-address privacy, the proxy MUST do the following: If the Anonymity header contains the value "full" or "uri", the proxy MUST replace the "addr-spec" in the Remote-Party-ID header in the initial INVITE with a private "addr-spec" and add a "user=private" parameter. If the Anonymity header contains the value "full" or "name", the proxy MUST delete the "display-name" in the Remote-Party-ID header field in the initial INVITE. To generate the user part of a private "addr-spec", the proxy MUST include (1) the initial "addr-spec", (2) the value of Anonymity, and (3) sufficient checksum information to prevent tampering by the untrusted party. It MAY contain any other information the proxy desires as well. This information MUST be encoded or encrypted such that the next hop is unable to discern the initial "addr-spec". It is RECOMMENDED that the string be encrypted with a symmetric privately-held key, and converted to a printable string using Base64 encoding. The proxy MUST identify itself in the hostname of the private "addr-spec".

For IP-address privacy, the proxy MUST rewrite the request to ensure that the IP-address of the originating UAC will not be revealed. This implies that neither SIP signaling nor IP media streams are exchanged directly between the UAC and UAS. A level of indirection which we call an Anonymizer MUST be provided.

Prior to forwarding the request, the proxy 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.

When receiving the first non-100 response to the initial INVITE from an untrusted entity, the proxy first determines if the response came from a UAS that it serves.

If it did, the proxy examines the response for the presence of a Remote-Party-ID and Anonymity header and applies similar processing as for an INVITE received from a UAC served by the proxy. Furthermore, if the original INVITE did not contain a Proxy-Require header field with an option tag of "privacy", the proxy can not determine if the previous hop supports the extension or not. Consequently, if the response contains a request for privacy, the privacy MUST be applied by this proxy, irrespective of whether the upstream hop is trusted or not.

If the response came from an untrusted entity, and it was not a UAS the proxy serves, the proxy MUST either remove any Remote-Party-ID header provided or set "rpi-screen=no" before the response is forwarded upstream. The same action MUST be taken when the initial INVITE did not contain a Proxy-Require with an option tag of "private", irrespective of whether the downstream hop was trusted or not.

6.4 Additional proxy behavior

A proxy supporting this extension SHOULD be prepared to receive a request containing a SIP-URL with a user parameter of "private". If the "hostport" part of the SIP-URL identifies the proxy handling the request, the proxy MUST decrypt the "user" portion of the SIP-URL and replace it with the decrypted SIP-URL that was contained in the "user" portion as well as any other information included, e.g. Anonymity. Note that the decrypted SIP-URL may itself contain a "private" SIP-URL. If the proxy is unable to decrypt and recover such a "private" SIP-URL, it MUST fail the request with a 4xx error code.

7 Example of Use

In this Section, we will 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.

7.1 Basic Privacy Example

The Figure below illustrates a basic privacy example scenario

```
                +---------+             +--------+
     1: INVITE  | Proxy-o | 2: INVITE   | Proxy-t| 3: INVITE
       +------->|         |------------>|        |---------+
       |        +---------+             +--------+         |
       |                                                   |
       |                 trust boundary                    |
   . . |. . . . . . . . . . . . . . . . . . . . . . . .. . | . . .
       |                                                   |
       |                                                   \/
   +------+                  RTP/RTCP                   +------+
   | UA-o |<------------------------------------------->| UA-t |
   +------+                                             +------+
```

Figure 1 - Basic Privacy Example

The originating user agent (UA-o) sends an INVITE (1) to Proxy-o where it identifies itself and requests URI and Name privacy. Since the From header field contains calling identity information, UA-o supplies a cryptographically random identifier for the user info, and a non-identifying hostname, e.g. "localhost" rather than its true identity:

```
INVITE
From:              sip:xyz@localhost
Remote-Party-ID: "John Doe" <sip:jdoe@foo.com>
Anonymity:         uri, name
Proxy-Require:     privacy
```

Proxy-o verifies the calling identity information before it sends INVITE (2) to Proxy-t, which in this case is trusted. Proxy-t examines the Anonymity header field included in the INVITE and sees that URI and Name privacy is requested. Proxy-t therefore removes the "display-name" from Remote-Party-ID, encrypts the "addr-spec" ID, puts the result in the "user" part, inserts itself as the "host" and adds a "user=private" parameter. Also, Proxy-t removes the Anonymity header:

```
INVITE
From:             sip:xyz@localhost
Remote-Party-ID: <sip:e(<sip:jdoe@foo.com>)@proxy-t.foo.com;
                                                   user=private>
Proxy-Require:   privacy
```

UA-o notes the presence of the Remote-Party-ID, but since a "user=private" parameter is provided, it can only identify the calling party as private. UA-o decides to accept the call, and responds with a 180 Ringing. In this case, there is no request for Anonymity, so only the Remote-Party-ID of the called party is added:

```
180
Remote-Party-ID: <sip:mdoe@foo.com>
```

Proxy-t verifies the information provided and adds the omitted "display-name" to the Remote-Party-ID. Since no Anonymity 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>
```

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. Our next example looks more closely at this.

7.2 Full Privacy Example

The second example we look at is one where full privacy is requested, i.e. both calling name and number privacy, as well as IP address privacy. The Figure below illustrates how IP address privacy can be achieved by inserting a trusted intermediary, an anonymizer, for the media streams between UA-o and UA-t.

```
                +---------+             +--------+
     1: INVITE  | Proxy-o | 2: INVITE   | Proxy-t| 3: INVITE
       +------->|         |------------>|        |----------+
       |        +---------+             +--------+          |
       |                  \           /                     |
       |                   \         /                      |
       |      SIP           +--------+          SIP         |
       | +----------------->| anony- |-------------------+  |
       | |          +------>|  mizer |--------+          |  |
       | |          |       +--------+        |          |  |
       | |          |                         |          |  |
       | |          |                         |          |  |
       | |          |     trust boundary      |          |  |
   . . |.|. . . . . | . . . . . . . . . . . . | . . . .. |..| . . .
       | |          |                         |          |  |
       | |          |                         |         \/ \/
   +------+ RTP/RTCP|                         |RTP/RTCP +------+
   | UA-o |<--------+                         +-------->| UA-t |
   +------+                                             +------+
```

Figure 2 - 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.

Also, the following commonly used header fields may reveal privacy information, which can be remedied as described:

@ The From header field must not reveal any calling identity information in the SIP-URL. This can be remedied, e.g. by supplying a cryptographically random identifier for the userinfo, and a non-identifying hostname, e.g. "localhost". The "display-name" can either be omitted or provided as "anonymous".
@ A Contact header field must be set to point to the anonymizer to prevent any direct signaling between UA-o and UA-t
@ Via 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.
@ Call-ID should not be based on UA-o's IP-address

Furthermore, when SDP 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.

8. Security Considerations

A user requesting complete privacy must still authenticate himself to the proxy, and therefore the SIP messages between the UA and the proxy MUST be protected. Likewise, it is necessary that the proxies take precautions to protect the user identification information from eavesdropping and interception. Use of IPSec between the UA and proxy as well as between proxies is recommended.

9. Notice Regarding Intellectual Property Rights

AT&T may seek patent or other intellectual property protection for some or all of the technologies disclosed in the document. If any standards arising from this disclosure are or become protected by one or more patents assigned to AT&T, AT&T intends to disclose those patents and license them on reasonable and non-discriminatory terms. Future revisions of this draft may contain additional information regarding specific intellectual property protection sought or received.

3COM may seek patent or other intellectual property protection for some or all of the technologies disclosed in the document. If any

standards arising from this disclosure are or become protected by one or more patents assigned to 3COM, 3COM intends to disclose those patents and license them on reasonable and non-discriminatory terms. Future revisions of this draft may contain additional information regarding specific intellectual property protection sought or received.

10. 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 Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax

Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997

4 M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg,"SIP: session initiation protocol," Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

5 Marshall, W. et. al, "Architectural Considerations for Providing Carrier Class Telephony Services Utilizing SIP-based Distributed Call Control Mechanisms", Internet Draft, Internet Engineering Task Force, draft-dcsgroup-sip-arch-02, June 2000, Work In Progress

11. Acknowledgments

The Distributed Call Signaling work in the PacketCable project 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.

12. Author's Addresses

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

K. K. Ramakrishnan
AT&T
Florham Park, NJ 07932
Email: kkrama@research.att.com

Ed Miller
CableLabs
Louisville, CO 80027
Email: E.Miller@Cablelabs.com

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

Burcak Beser
3Com
Rolling Meadows, IL 60008
Email: Burcak_Beser@3com.com

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
Secure Cable Solutions
Westminster, CO 30120
Email: drevans@securecable.com

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

Full Copyright Statement

Expiration Date This memo is filed as <draft-dcsgroup-sip-privacy-02.txt>, and expires December 31, 2000.