

SIPREC
Internet-Draft
Intended status: Informational
Expires: March 13, 2013

A. Hutton, Ed.
Siemens Enterprise
Communications
L. Portman, Ed.
NICE Systems
R. Jain
IPC Systems
K. Rehor
Cisco Systems, Inc.
September 9, 2012

**An Architecture for Media Recording using the Session Initiation
Protocol
draft-ietf-siprec-architecture-06**

Abstract

Session recording is a critical requirement in many communications environments such as call centers and financial trading. In some of these environments, all calls must be recorded for regulatory, compliance, and consumer protection reasons. Recording of a session is typically performed by sending a copy of a media stream to a recording device. This document describes architectures for deploying session recording solutions in an environment which is based on the Session Initiation Protocol (SIP).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 13, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Definitions	4
3.	Session Recording Architecture	5
3.1.	Location of the Session Recording Client	5
3.1.1.	B2BUA acts as a Session Recording Client	5
3.1.2.	Endpoint acts as Session Recording Client	7
3.1.3.	A SIP Proxy cannot be a Session Recording Client	7
3.1.4.	Interaction with MEDIACTRL	8
3.1.5.	Interaction with Conference Focus	9
3.2.	Establishing the Recording Session	10
3.2.1.	Session Recording Client Initiated Recording	11
3.2.2.	Session Recording Server Initiated Recording	11
3.2.3.	Pause/Resume Recording Session	12
3.2.4.	Media Stream Mixing	12
3.2.5.	Media Transcoding	12
3.3.	Recording Metadata	12
3.3.1.	Contents of recording metadata	13
3.3.2.	Mechanisms for delivery of metadata to Session Recording Server	13
3.4.	Notifications to the Recorded User Agents	13
3.5.	Preventing the recording of a SIP session	13
4.	IANA considerations	13
5.	Security considerations	14
6.	Acknowledgements	14
7.	Informative References	15
	Authors' Addresses	16

1. Introduction

Session recording is a critical requirement in many communications environments such as call centers and financial trading. In some of these environments, all calls must be recorded for regulatory, compliance, and consumer protection reasons. Recording of a session is typically performed by sending a copy of a media stream to a recording device. This document describes architectures for deploying session recording solutions in an environment which is based on the Session Initiation Protocol (SIP) the requirements for which are described in [[RFC6341](#)].

This document focuses on how sessions are established between a Session Recording Client (SRC) and the Session Recording Server (SRS) for the purpose of conveying the Replicated Media and Recording Metadata (e.g. Identity of parties involved) relating to the Communication Session.

Once the Replicated Media and Recording Metadata have been received by the Session Recording Server they will typically be archived for retrieval at a later time. The procedures relating to the archiving and retrieval of this information is outside the scope of this document.

This document only considers active recording, where the Session Recording Client purposefully streams media to a Session Recording Server. Passive recording, where a recording device detects media directly from the network (E.g. using port mirroring techniques), is outside the scope of this document. In addition, lawful intercept is outside the scope of this document which takes account of the IETF policy on wiretapping [[RFC2804](#)].

The Recording Session that is established between the Session Recording Client and the Session Recording Server uses the normal procedures for establishing INVITE initiated dialogs as specified in [[RFC3261](#)] and uses SDP for describing the media to be used during the session as specified in [[RFC4566](#)]. However it is intended that some extensions to SIP (E.g. Headers, Option Tags, Etc.) will be defined to support the requirements for media recording. The Replicated Media is required to be sent in real-time to the Session Recording Server and is not buffered by the Session Recording Client to allow for real-time analysis of the media by the Session Recording Server.

2. Definitions

Session Recording Server (SRS): A Session Recording Server (SRS) is a SIP User Agent (UA) that is a specialized media server or collector that acts as the sink of the recorded media. An SRS is typically implemented as a multi-port device that is capable of receiving media from multiple sources simultaneously. An SRS is the sink of the recorded session metadata.

Session Recording Client (SRC): A Session Recording Client (SRC) is a SIP User Agent (UA) that acts as the source of the recorded media, sending it to the SRS. An SRC is a logical function. Its capabilities may be implemented across one or more physical devices. In practice, an SRC could be a personal device (such as a SIP phone), a SIP Media Gateway (MG), a Session Border Controller (SBC) or a SIP Media Server (MS) integrated with an Application Server (AS). This specification defines the term SRC such that all such SIP entities can be generically addressed under one definition. The SRC provides metadata to the SRS.

Communication Session (CS): A session created between two or more SIP User Agents (UAs) that is the subject of recording.

Recording Session (RS): The SIP session created between an SRC and SRS for the purpose of recording a Communication Session.

Recording aware User Agent (UA): A SIP User Agent that is aware of SIP extensions associated with the Communication Session. Such extensions may be used to notify the Recording aware UA that a session is being recorded, or by a Recording aware UA to express preferences as to whether a recording should be started, paused, resumed or stopped.

Recording unaware User Agent (UA): A SIP User Agent that is unaware of SIP extensions associated with the Communication Session. Such Recording unaware UA will be notified that a session is being recorded or express preferences as to whether a recording should be started, paused, resumed or stopped via some other means that is out of scope of SIPREC.

Recording Metadata: The metadata describing the communication session that is required by the Session Recording Server. This will include for example the identity of users that participate in the Communication Session and dialog state. Typically this metadata is archived with the replicated media at the Session Recording Server. The recording metadata is delivered in real-time to the Session Recording Server.

Replicated Media: A copy of the media associated with the Communication Session created by the Session Recording Client and sent to the Session Recording Server. It may contain all the media associated with the communication session (E.g. Audio and Video) or just a subset (E.g. Audio). Replicated Media is part of Recording Session.

3. Session Recording Architecture

3.1. Location of the Session Recording Client

This section contains some example session recording architectures showing how the Session Recording Client is a logical function that can be located in or split between various physical components.

3.1.1. B2BUA acts as a Session Recording Client

A SIP Back to Back User Agent (B2BUA) which has access to the media that is to be recorded may act as a Session Recording Client. The B2BUA may already be aware that a session needs to be recorded before the initial establishment of the communication session or the decision to record the session may occur after the session has been established.

If the B2BUA/SRC makes the decision to initiate the Recording Session (RS) then it will initiate the establishment of a SIP Session by sending an INVITE to the Session Recording Server.

If the Session Recording Server makes the decision to initiate the recording session then it will initiate the establishment of a SIP Session by sending an INVITE to the B2BUA/Session Recording Client.

The RS INVITE contains information which identifies the session as being established for the purposes of recording and prevents the session from being accidentally rerouted to a UA which is not a SRS.

The B2BUA/SRC is responsible for notifying the UAs involved in the communication session that the session is being recorded.

The B2BUA/SRC is responsible for complying with requests from recording aware UAs or through some configured policies indicating that the communication session should not be recorded.

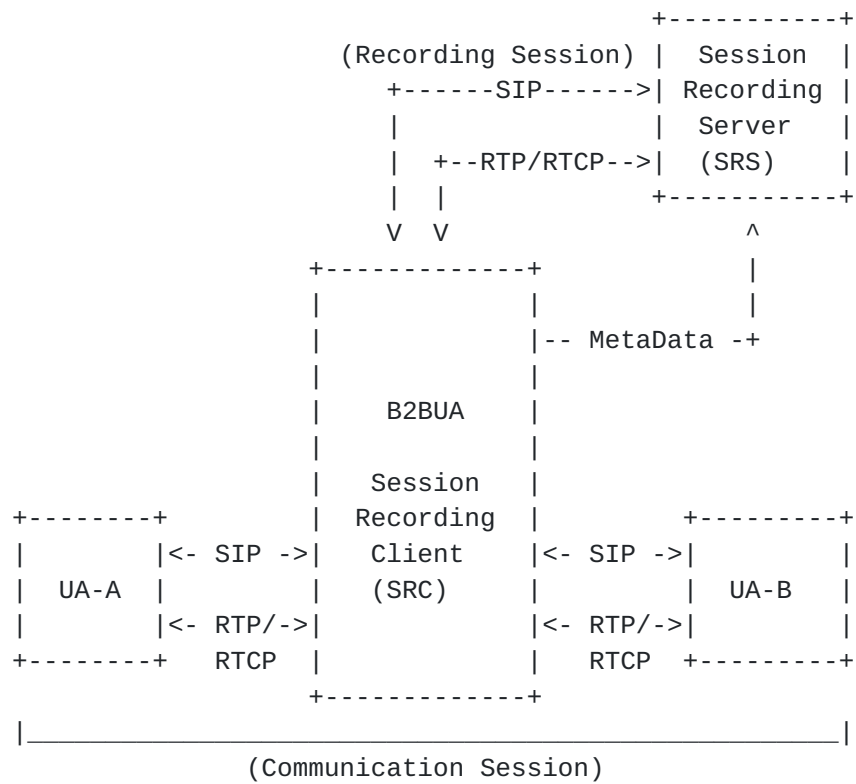


Figure 1: B2BUA Acts as the Session Recording Client.

3.1.2. Endpoint acts as Session Recording Client

A SIP Endpoint / UA may act as a Session Recording Client in which case the endpoint sends the Replicated Media to the Session Recording Server

If the endpoint makes the decision to initiate the Recording Session then it will initiate the establishment of a SIP Session by sending an INVITE to the Session Recording Server.

If the Session Recording Server makes the decision to initiate the Recording Session then it will initiate the establishment of a SIP Session by sending an INVITE to the endpoint. The actual decision mechanism is out of scope of SIPREC.

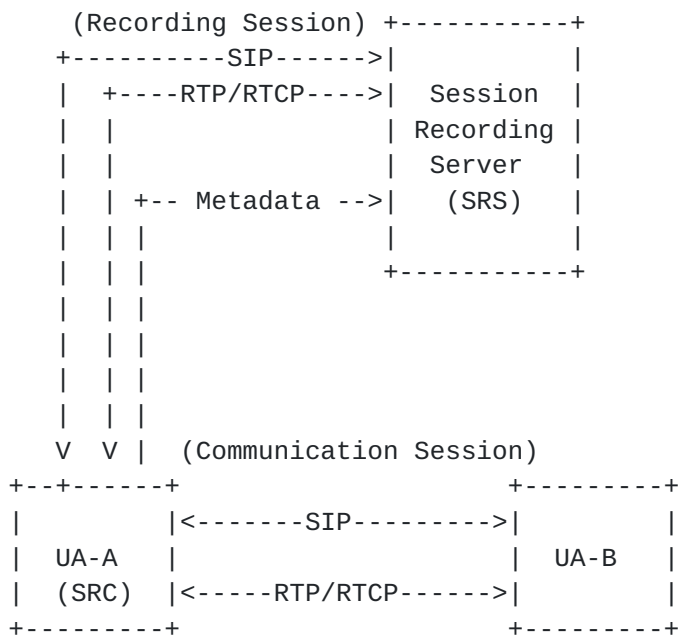


Figure 2: SIP Endpoint acts as the Session Recording Client

3.1.3. A SIP Proxy cannot be a Session Recording Client

A SIP Proxy is unable to act as an SRC because it does not have access to the media and therefore has no way of enabling the delivery of the replicated media to the SRS

3.1.4. Interaction with MEDIACTRL

The mediactrl architecture [RFC5567] describes an architecture in which an Application Server (AS) controls a Media Server (MS) which may be used for purposes such as conferencing and recording media streams. In the [RFC5567] architecture the AS typically uses SIP Third Party Call Control (3PCC) to instruct the SIP UAs to direct their media to the Media Server.

The Session Recording Client and Session Recording Server described in this document may act as an application server as described in [RFC5567]; and therefore, when further decomposed, may be made up of an application server which uses a mediactrl interface to control a media server for the purpose of recording the media streams. However, this interface is considered outside the scope of this document.

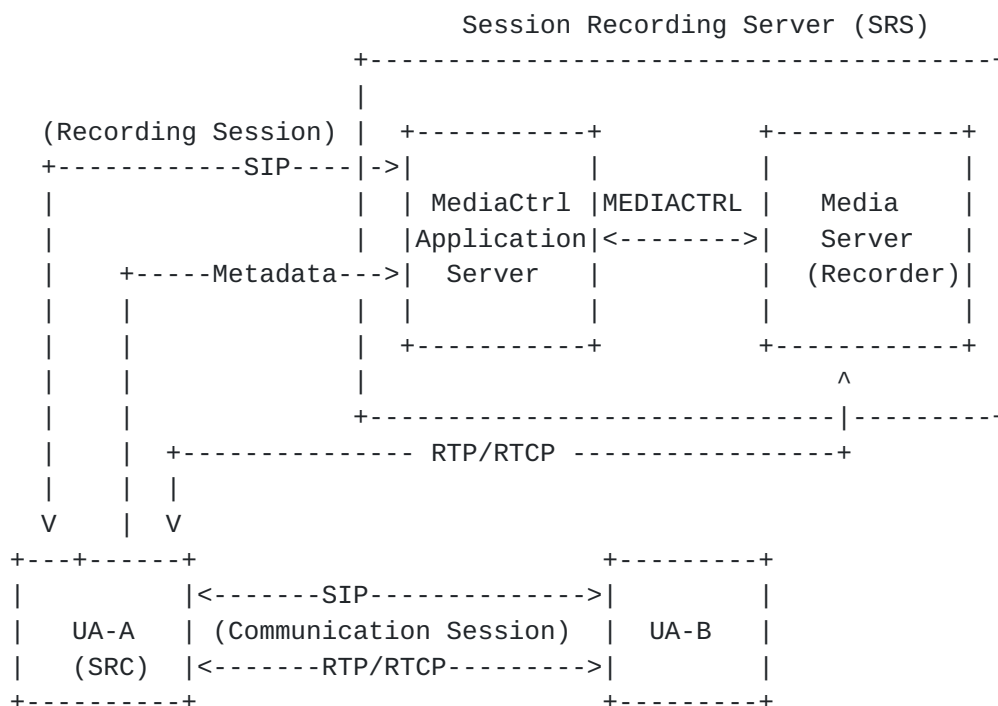


Figure 3: Example of Session Recording Server using MEDIACTRL

The SRC may be involved in the conference from the very beginning or may join at some later point of time.

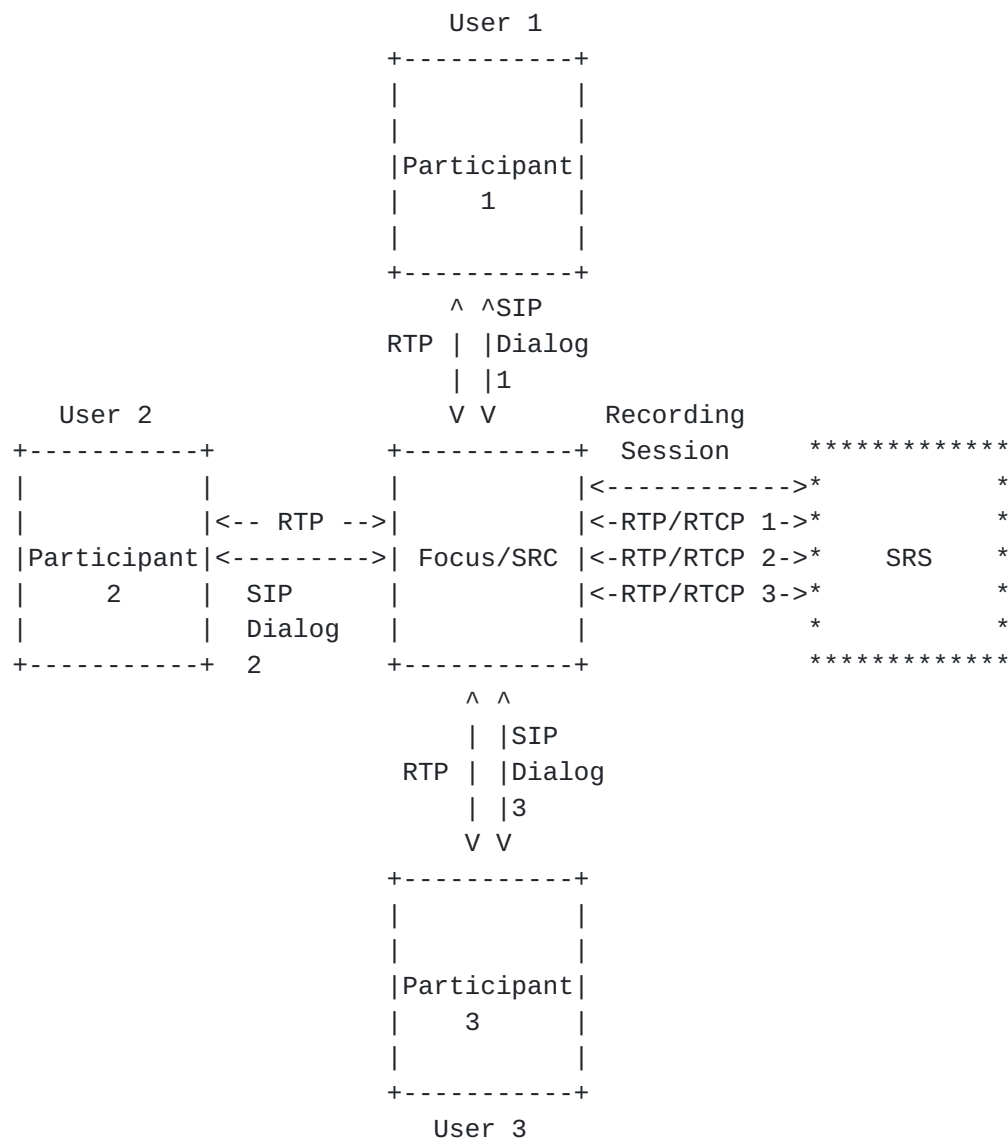


Figure 5: Conference Focus acting as an SRC.

3.2. Establishing the Recording Session

The Session Recording Client or the Session Recording Server may initiate the Recording Session.

It should be noted that the Recording Session is independent from the Communication Session that is being recorded at both the SIP dialog level and at the session level.

Concerning media negotiation, regular SIP/SDP capabilities should be used, and existing transcoding capabilities and media encryption should not be precluded.

3.2.1. Session Recording Client Initiated Recording

When the Session Recording Client initiates the Recording Session for the purpose of conveying media to the Session Recording Server it performs the following actions.

- o The SRC is provisioned with a Unified Resource Identifier (URI) for the SRS, which is resolved through normal [[RFC3263](#)] procedures.
- o Initiates the dialog by sending an INVITE request to the Session Recording Server. The dialog is established according to the normal procedures for establishing an INVITE initiated dialog as specified in [[RFC3261](#)].
- o Include in the INVITE an indication that the session is established for the purpose of recording the associated media.
- o If the Replicated Media is to be started immediately then the Session Recording Client will include an SDP attribute of "a=sendonly" for each media line or "a=inactive" if it is not ready to transmit the media.
- o The Recording Session may replicate all media associated with the Communication Session or only a subset.
- o Replicate the media streams that are to be recorded and transmit the media to the Session Recording Server.

3.2.2. Session Recording Server Initiated Recording

When the Session Recording Server initiates the media recording session with the Session Recording Client it performs the following actions.

- o The SRS is provisioned with a Unified Resource Identifier (URI) for the SRC, which is resolved through normal [[RFC3263](#)] procedures.
- o Send an INVITE request to the Session Recording Client
- o Include in the INVITE an indication that the session is established for the purpose of recording the associated media. Possible mechanisms for this include using the Require header or a media feature tag as defined in [[RFC3840](#)].
- o Identify the sessions that is to be recorded. The actual mechanism of the identification depends on SRC policy.

- o If the Recording Session is to be started immediately then the Session Recording Client will include an SDP attribute of "a=recvonly" for each media line or "a=inactive" if it is not ready to receive the media

If the Session Recording Server does not have prior knowledge of what media streams are available to be recorded it can make use of an offerless INVITE which allows the Session Recording Client to make the initial Session Description Protocol (SDP) offer.

3.2.3. Pause/Resume Recording Session

The Session Recording Server or the Session Recording Client may pause the recording by changing the SDP direction attribute to "inactive" and resume the recording by changing the direction back to "recvonly" or "sendonly".

3.2.4. Media Stream Mixing

In a basic session involving only audio there are typically two audio/RTP streams between the two UAs involved transporting media in each direction. When recording this media the two streams may be mixed at the SRC before being transmitted to the SRS or it may be a requirement of the recording server that the media streams are not mixed and are sent to the SRS as two separate streams. The case when media is mixed at the SRC is simple as only a single media stream is required to be sent to the SRS. However in the case when the media streams are not mixed then the SDP offer sent to the SRS must describe two separate media streams.

3.2.5. Media Transcoding

The communication session (CS) and the recording session (RS) are negotiated separately using a standard SDP offer/answer exchange which may result in the SRC having to perform media transcoding between the two sessions. If the SRC is not capable of performing media transcoding it may limit the media formats in the offer to the SRS depending on what media is negotiated on the CS or may limit what it includes in the offer on the CS if it has prior knowledge of the media formats supported by the SRS. However typically the SRS will be a more capable device which can provide a wide range of media format options to the SRC and may also be able to make use of a media transcoder as detailed in [[RFC5369](#)].

3.3. Recording Metadata

3.3.1. Contents of recording metadata

The metadata model is defined in [[I-D.ietf-siprec-metadata](#)].

3.3.2. Mechanisms for delivery of metadata to Session Recording Server

The SRS obtains session recording metadata from the SRC. The metadata is transported via SIP based mechanisms as specified in [[I-D.ietf-siprec-protocol](#)]

It is also possible that metadata is transported via non SIP based mechanisms but these are considered out of scope.

It is also possible to have RS session without the metadata, in such case SRS will be receiving it by some other means or not at all.

3.4. Notifications to the Recorded User Agents

Typically a user that is involved in a session that is to be recorded is notified by an announcement at the beginning of the session or may receive some warning tones within the media. However the standardization of media recording protocols when using SIP enable an indication that the call is being recorded to be included in the SIP requests and responses associated with that communication session.

It is the Session Recording Client that provides the notification to all SIP UAs for which it is replicating received media for the purpose of recording including the local user if the Session Recording Client is a SIP endpoint.

3.5. Preventing the recording of a SIP session

A Recording Aware UA may during the initial session establishment or during an established session provide an indication of their preference with regard to recording the media in the communication session. The mechanism for this are specified in [[I-D.ietf-siprec-protocol](#)]

4. IANA considerations

This draft mentions SIP/SDP extensions. The associated IANA considerations are addressed in [[I-D.ietf-siprec-protocol](#)] that defines them.

5. Security considerations

The Recording Session is fundamentally a standard SIP dialog and media session and therefore makes use of existing SIP security mechanisms for securing the Recording Session and Recording Metadata.

The intended use of this architecture is only for the case where the users are aware that they are being recorded, and the architecture provides the means for the Session Recording Client to notify users that they are being recorded.

This architectural solution is not intended to support lawful intercept which in contrast requires that users are not informed.

It is the responsibility of the Session Recording Server to protect the Replicated Media and Recording Metadata once it has been received and archived. The mechanism for protecting the storage and retrieval from the SRS is out of scope of this work.

Protection of the RS should be not weaker than protection of the CS, and may need to be stronger because the media is retransmitted (allowing more possibility for interception). This applies to both the signaling and media paths.

It is essential that the SRC will authenticate the SRS because the client must be certain that it is recording on the right recording system. It is less important that the SRS authenticate the SRC, but implementations must have the ability to perform mutual authentication.

In some environments, it is desirable to not decrypt and re-encrypt the media. This means the same key is used on the CS and RS. This only works when the media is encrypted, not the transport. When the media must be decrypted on the CS, and is re-encrypted on the RS, a new key must be used.

The retrieval mechanism for media recorded by this protocol is out of scope. Implementations of retrieval mechanisms should consider the security implications carefully as the retriever is not usually a party to the call that was recorded. Retrievers should be authenticated carefully. The crypto suites on the retrieval should be no less strong than used on the RS, and may need to be stronger.

6. Acknowledgements

Thanks to John Elwell, Brian Rosen, Alan Johnson, Cullen Jennings, Hadriel Kaplan, Henry Lum, Paul Kyzivat, Parthasarathi R, Ram Mohan

R, Charles Eckel and Friso Feenstra for their significant contributions and assistance with this document and Working Group, and to all the members of SIPREC WG mailing list for providing valuable input to this work.

7. Informative References

- [RFC3261] 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.
- [RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", [RFC 3263](#), June 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC6341] Rehor, K., Portman, L., Hutton, A., and R. Jain, "Use Cases and Requirements for SIP-Based Media Recording (SIPREC)", [RFC 6341](#), August 2011.
- [I-D.ietf-siprec-metadata] R, R., Ravindran, P., and P. Kyzivat, "Session Initiation Protocol (SIP) Recording Metadata", [draft-ietf-siprec-metadata-06](#) (work in progress), March 2012.
- [I-D.ietf-siprec-protocol] Portman, L., Lum, H., Eckel, C., Johnston, A., and A. Hutton, "Session Recording Protocol", [draft-ietf-siprec-protocol-04](#) (work in progress), May 2012.
- [RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [RFC 3840](#), August 2004.
- [RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", [RFC 4353](#), February 2006.
- [RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session

Initiation Protocol (SIP) Event Package for Conference State", [RFC 4575](#), August 2006.

[RFC5567] Melanchuk, T., "An Architectural Framework for Media Server Control", [RFC 5567](#), June 2009.

[RFC5369] Camarillo, G., "Framework for Transcoding with the Session Initiation Protocol (SIP)", [RFC 5369](#), October 2008.

[RFC2804] IAB and IESG, "IETF Policy on Wiretapping", [RFC 2804](#), May 2000.

Authors' Addresses

Andrew Hutton (editor)
Siemens Enterprise Communications
Hofmannstrasse 51
Munich D-81379
Germany

Email: andrew.hutton@siemens-enterprise.com

Leon Portman (editor)
NICE Systems
8 Hapnina
Ra'anana 43017
Israel

Email: leon.portman@nice.com

Rajnish Jain
IPC Systems
777 Commerce Drive
Fairfield, CT 06825
USA

Email: rajnish.jain@ipc.com

Ken Rehor
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Email: krehor@cisco.com