

Internet Engineering Task Force (IETF)
Request for Comments: 7245
Category: Informational
ISSN: 2070-1721

A. Hutton, Ed.
Unify
L. Portman, Ed.
NICE Systems
R. Jain
IPC Systems
K. Rehor
Cisco Systems, Inc.
May 2014

An Architecture for Media Recording Using the Session Initiation Protocol

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 that is based on the Session Initiation Protocol (SIP).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7245>.

Copyright Notice

Copyright (c) 2014 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 SRC	5
3.1.1. B2BUA Acts as a SRC	5
3.1.2. Endpoint Acts as SRC	6
3.1.3. A SIP Proxy Cannot Be a SRC	7
3.1.4. Interaction with MEDIACTRL	7
3.1.5. Interaction with Conference Focus	9
3.2. Establishing the Recording Session	10
3.2.1. SRC-Initiated Recording	11
3.2.2. SRS-Initiated Recording	11
3.2.3. Pause/Resume Recording Session	12
3.2.4. Media Stream Mixing	12
3.2.5. Media Transcoding	12
3.2.6. Lossless Recording	12
3.3. Recording Metadata	13
3.3.1. Contents of Recording Metadata	13
3.3.2. Mechanisms for Delivery of Metadata to SRS	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	15
7. Informative References	15

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 as defined in "Use Cases and Requirements for SIP-Based Media Recording (SIPREC)" [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 the parties involved) relating to the Communication Session.

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

This document only considers active recording, where the SRC purposefully streams media to a SRS. 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 SRC and the SRS uses the normal procedures for establishing INVITE-initiated dialogs as specified in [RFC3261] and uses the Session Description Protocol (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 SRS and is not buffered by the SRC to allow for real-time analysis of the media by the SRS.

2. Definitions

The first four definitions are quoted from RFC 6341.

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 CS.

The following terms are defined by this document.

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

Recording Metadata: The metadata describing the CS that is required by the SRS. This will include, for example, the identities of users that participate in the CS and dialog state. Typically, this metadata is archived with the Replicated Media at the SRS. The recording metadata is delivered in real-time to the SRS.

Replicated Media: A copy of the media that is associated with the CS, was created by the SRC, and was sent to the SRS. It may contain all the media associated with the CS (e.g., audio and video) or just a subset (e.g., audio). Replicated Media is part of the Recording Session.

3. Session Recording Architecture

3.1. Location of the SRC

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

3.1.1. B2BUA Acts as a SRC

A SIP Back-to-Back User Agent (B2BUA) that has access to the media to be recorded may act as an SRC. The B2BUA may already be aware that a session needs to be recorded before the initial establishment of the CS, or the decision to record the session may occur after the session has been established.

If the SRC makes the decision to initiate the RS, then it will do so by sending a SIP INVITE request to the SRS.

If the SRS makes the decision to initiate the Recording Session, then it will initiate the establishment of a SIP RS by sending an INVITE to the SRC.

The RS INVITE contains information that identifies the session as being established for the purposes of recording and prevents the session from being accidentally rerouted to a UA that is not an SRS if the RS was initiated by the SRC or vice versa.

The B2BUA/SRC is responsible for notifying the UAs involved in the CS 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 CS should not be recorded.

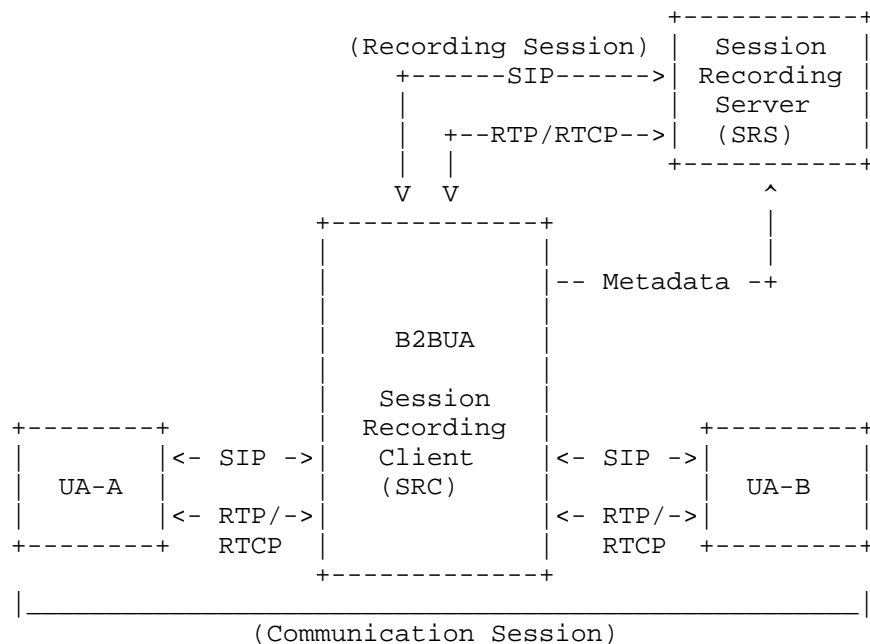


Figure 1: B2BUA Acts as the Session Recording Client

3.1.2. Endpoint Acts as SRC

A SIP endpoint / UA may act as a SRC. In that case, the endpoint sends the Replicated Media to the SRS.

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 SRS.

If the SRS 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 for the SIP media recording architecture.

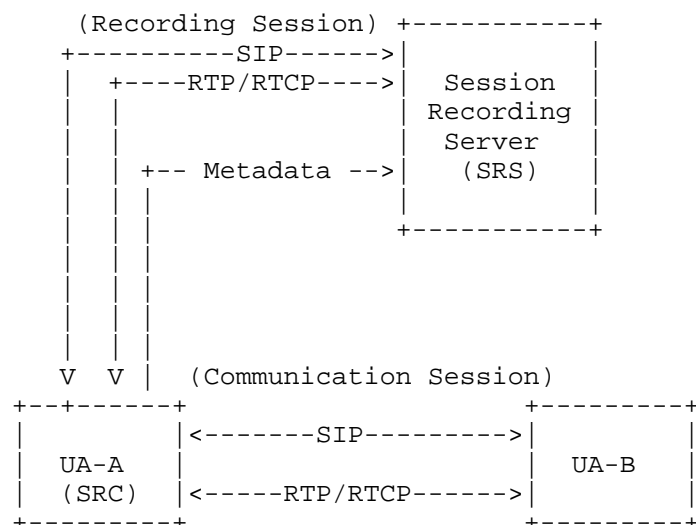


Figure 2: SIP Endpoint Acts as the Session Recording Client

3.1.3. A SIP Proxy Cannot Be a SRC

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 architecture described in [RFC5567], the AS typically uses SIP Third Party Call Control (3PCC) to instruct the SIP UAs to direct their media to the Media Server.

The SRC or the SRS described in this document may be architected according to [RFC5567]; therefore, when further decomposed, they may be made up of an AS that uses a MEDIACTRL interface to control an MS.

As shown in Figure 3, when the SRS is architected according to [RFC5567], the MS acts as a sink of the recording media, and the AS acts as a sink of the metadata and the termination point for RS SIP signaling. As shown in Figure 4, when the SRC is architected according to [RFC5567], the MS acts as a source of recording media, and the AS acts as a source of the metadata and the termination point for RS SIP signaling.

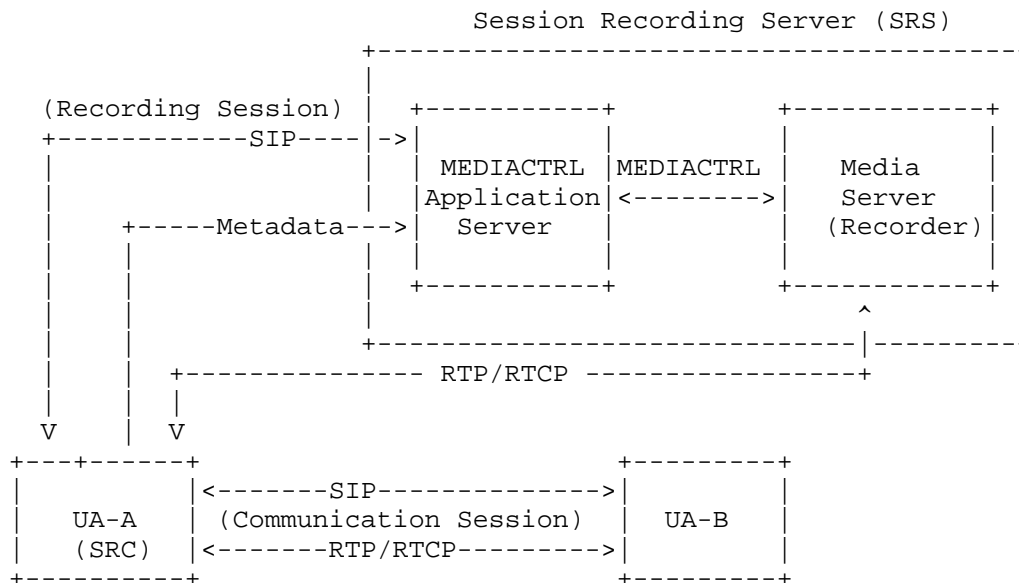


Figure 3: Example of Session Recording Server using MEDIACTRL

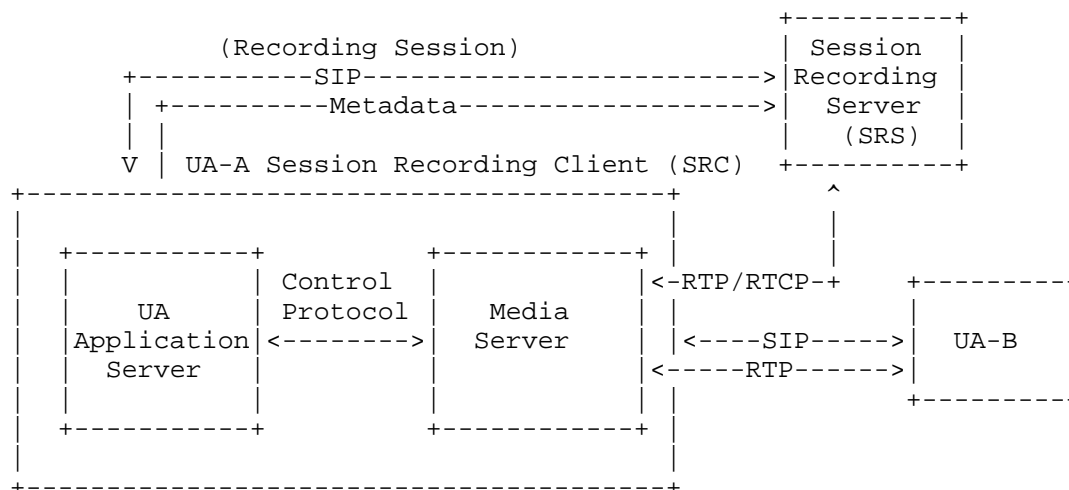


Figure 4: Example of Session Recording Client Decomposition

3.1.5. Interaction with Conference Focus

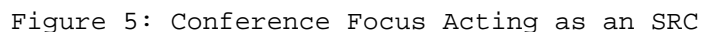
In the case of a centralized conference, a combination of the conference focus and mixer [RFC4353] may act as a SRC and therefore provide the SRS with the Replicated Media and associated recording metadata. In this arrangement, the SRC is able to provide media and metadata relating to each of the participants, including, for example, any side conversations where the media passes through the mixer.

The conference focus can either provide mixed Replicated Media or separate streams per conference participant (as depicted in Figure 5).

The conference focus may also act as a recording-aware UA in the case when one of the participants acts as a SRC.

In an alternative arrangement, a SIP endpoint that is a conference participant can act as an SRC. The SRC will in this case have access to the media and metadata relating to that particular participant and may be able to obtain additional metadata from the conference focus. The SRC may, for example, use the conference event package as described in [RFC4575] to obtain information about other participants that it provides to the SRS within the recording metadata.

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



3.2. Establishing the Recording Session

The SRC or the SRS may initiate the Recording Session.

It should be noted that the Recording Session is independent from the CS 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. SRC-Initiated Recording

When the SRC initiates the Recording Session for the purpose of conveying media to the SRS, it performs the following actions:

- o Is provisioned with a Unified Resource Identifier (URI) for the SRS; the URI is resolved through normal [RFC3263] procedures.
- o Initiates the dialog by sending an INVITE request to the SRS. The dialog is established according to the normal procedures for establishing an INVITE-initiated dialog as specified in [RFC3261].
- o Includes in the INVITE an indication that the session is established for the purpose of recording the associated media.
- o Includes an SDP attribute of "a=sendonly" for each media line if the Replicated Media is to be started immediately, or includes "a=inactive" if it is not ready to transmit the media.
- o Replicates the media streams that are to be recorded and transmits the media to the SRS.

The Recording Session may replicate all media associated with the CS or only a subset.

3.2.2. SRS-Initiated Recording

When the SRS initiates the media Recording Session with the SRC, it performs the following actions:

- o Is provisioned with a Unified Resource Identifier (URI) for the SRC; the URI is resolved through normal [RFC3263] procedures.
- o Sends an INVITE request to the SRC.
- o Includes in the INVITE an indication that the session is established for the purpose of recording the associated media.
- o Identifies the sessions that are to be recorded. The actual mechanism of the identification depends on SRC policy.
- o Includes an SDP attribute of "a=recvonly" for each media line if the Recording Session is to be started immediately, or includes "a=inactive" if it is not ready to receive the media.

If the SRS 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 SRC to make the initial SDP offer.

3.2.3. Pause/Resume Recording Session

The SRS or the SRC 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 in transporting media in each direction. When recording this media, the two streams may be mixed or not mixed at the SRC before being transmitted to the SRS. In the case when they are not mixed, two separate streams are sent to the SRS, and the SDP offer sent to the SRS must describe two separate media streams. In the mixed case, a single mixed media stream is sent to the SRS.

3.2.5. Media Transcoding

The CS and the RS are negotiated separately using the 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].

3.2.6. Lossless Recording

Session recording may be a regulatory requirement in certain communication environments. Such environments may impose a requirement generally known as "lossless recording". An overall solution for lossless recording may involve multiple layers of solutions. Individual aspects of the solutions may range from administering networks for appropriate QoS, reliable transmission of recorded media, and perhaps certain SIPREC protocol-level capabilities in SRC and SRS.

3.3. Recording Metadata

3.3.1. Contents of Recording Metadata

The metadata model is defined in [REC-METADATA].

3.3.2. Mechanisms for Delivery of Metadata to SRS

The SRS obtains session recording metadata from the SRC. The metadata is transported via SIP-based mechanisms as specified in [REC-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 an RS session without the metadata; in that case, the SRS will be receiving the metadata 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, SIPREC enables an indication that the call is being recorded to be included in the SIP requests and responses associated with that CS.

The SRC provides the notification to all SIP UAs for which it is replicating received media for the purpose of recording. If the SRC is acting as a SIP endpoint, as described in Section 3.1.2, then it also provides a notification to the local user.

3.5. Preventing the Recording of a SIP Session

During the initial session establishment or during an established session, a recording-aware UA may provide an indication of its preference with regard to recording the media in the CS. The mechanisms for this are specified in [REC-PROTOCOL]

4. IANA Considerations

This document has no actions for IANA. This document mentions SIP/SDP extensions. The associated IANA considerations are addressed in [REC-PROTOCOL], which 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 SRC 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 SRS to protect the Replicated Media and Recording Metadata once it has been received and archived. The stored content must be protected using a cipher at least as strong (or stronger) than the original content; however, the mechanism for protecting the storage and retrieval from the SRS is out of scope of this work. The keys used to store the data must also be securely maintained by the SRS and should only be released, securely, to authorized parties. How to secure these keys, properly authorize a receiving party, or securely distribute the keying material is also out of scope of this work.

Protection of the RS should not be 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 media encryption key is negotiated and used within the CS and RS. If for any reason the media are decrypted on the CS and are 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 cryptosuites on the retrieval should be no less strong than those 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, Friso Feenstra, and Dave Higton for their significant contributions and assistance with this document and working group. Also, thanks to all the members of the 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.
- [REC-METADATA] Ravindranath, R., Ravindran, P., and P. Kyzivat, "Session Initiation Protocol (SIP) Recording Metadata", Work in Progress, February 2014.
- [REC-PROTOCOL] Portman, L., Lum, H., Eckel, C., Johnston, A., and A. Hutton, "Session Recording Protocol", Work in Progress, February 2014.
- [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)
Unify
Hofmannstrasse 51
81359 Munich
Germany

EMail: andrew.hutton@unify.com

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

EMail: leon.portman@gmail.com

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

EMail: rajnish.jain@outlook.com

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

EMail: krehor@cisco.com

