SIPREC
Internet-Draft
Intended status: Standards Track
Expires: February 26, 2015

L. Portman
NICE Systems
H. Lum, Ed.
Genesys
C. Eckel
Cisco
A. Johnston
Avaya
A. Hutton
Unify

August 25, 2014

Session Recording Protocol
draft-ietf-siprec-protocol-14

Abstract

This document specifies the use of the Session Initiation Protocol (SIP), the Session Description Protocol (SDP), and the Real Time Protocol (RTP) for delivering real-time media and metadata from a Communication Session (CS) to a recording device. The Session Recording Protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) between the Session Recording Client (SRC), which is on the path of the CS, and a Session Recording Server (SRS) at the recording device.

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 February 26, 2015.

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 4
2. Terminology 4
3. Definitions 4
4. Scope 4
5. Overview of operations 5
5.1. Delivering recorded media 5
5.2. Delivering recording metadata 7
5.3. Receiving recording indications and providing recording preferences 8
6. SIP Handling 9
6.1. Procedures at the SRC 10
6.1.1. Initiating a Recording Session 10
6.1.2. SIP extensions for recording indication and preference 10
6.2. Procedures at the SRS 11
6.3. Procedures for Recording-aware User Agents 11
7. SDP Handling 12
7.1. Procedures at the SRC 12
7.1.1. SDP handling in RS 12
7.1.1.1. Handling media stream updates 13
7.1.2. Recording indication in CS 14
7.1.3. Recording preference in CS 15
7.2. Procedures at the SRS 15
7.3. Procedures for Recording-aware User Agents 17
7.3.1. Recording indication 17
7.3.2. Recording preference 18
8. RTP Handling 19
8.1. RTP Mechanisms 19
8.1.1. RTCP 19
8.1.2. RTP Profile 20
8.1.3. SSRC 20

8.1.4. CSRC 21
8.1.5. SDES 21
8.1.5.1. CNAME 21
8.1.6. Keepalive 21
8.1.7. RTCP Feedback Messages 22
8.1.7.1. Full Intra Request 22
8.1.7.2. Picture Loss Indicator 22
8.1.7.3. Temporary Maximum Media Stream Bit Rate Request . 23
8.1.8. Symmetric RTP/RTCP for Sending and Receiving 23
8.2. Roles 23
8.2.1. SRC acting as an RTP Translator 24
8.2.1.1. Forwarding Translator 25
8.2.1.2. Transcoding Translator 25
8.2.2. SRC acting as an RTP Mixer 26
8.2.3. SRC acting as an RTP Endpoint 26
8.3. Packet Loss 27
8.4. RTP Session Usage by SRC 27
8.4.1. SRC Using Multiple m-lines 27
8.4.2. SRC Using Mixing 28
8.5. RTP Session Usage by SRS 29
9. Metadata 30
9.1. Procedures at the SRC 30
9.2. Procedures at the SRS 32
9.2.1. Formal Syntax 34
10. Persistent Recording 34
11. IANA Considerations 34
11.1. Registration of Option Tags 35
11.1.1. siprec Option Tag 35
11.1.2. record-aware Option Tag 35
11.2. Registration of media feature tags 35
11.2.1. src feature tag 35
11.2.2. srs feature tag 36
11.3. New Content-Disposition Parameter Registrations 36
11.4. Media Type Registration 36
11.4.1. Registration of MIME Type application/rs-metadata-request 36
11.5. SDP Attributes 37
11.5.1. 'record' SDP Attribute 37
11.5.2. 'recordpref' SDP Attribute 37
12. Security Considerations 38
12.1. Authentication and Authorization 38
12.2. RTP handling 39
12.3. Metadata 40
12.4. Storage and playback 40
13. Acknowledgements 40
14. References 40
14.1. Normative References 40
14.2. Informative References 41

Authors' Addresses 43

1. Introduction

This document specifies the mechanism to record a Communication Session (CS) by delivering real-time media and metadata from the CS to a recording device. In accordance to the architecture [RFC7245], the Session Recording Protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) between the Session Recording Client (SRC), which is on the path of the CS, and a Session Recording Server (SRS) at the recording device.

SIP is also used to deliver metadata to the recording device, as specified in [I-D.ietf-siprec-metadata]. Metadata is information that describes recorded media and the CS to which they relate.

The Session Recording Protocol intends to satisfy the SIP-based Media Recording requirements listed in [RFC6341].

In addition to the Session Recording Protocol, this document specifies extensions for user agents that are participants in a CS to receive recording indications and to provide preferences for recording.

2. Terminology

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 [RFC2119].

3. Definitions

This document refers to the core definitions provided in the architecture document [RFC7245].

The RTP Handling section uses the definitions provided in "RTP: A Transport Protocol for Real-Time Application" [RFC3550].

4. Scope

The scope of the Session Recording Protocol includes the establishment of the recording sessions and the reporting of the metadata. The scope also includes extensions supported by User Agents participating in the CS such as indication of recording. The user agents need not be recording-aware in order to participate in a CS being recorded.

The following items, which are not an exhaustive list, do not represent the protocol itself and are considered out of the scope of the Session Recording Protocol:

- Delivering recorded media in real-time as the CS media
- Specifications of criteria to select a specific CS to be recorded or triggers to record a certain CS in the future
- Recording policies that determine whether the CS should be recorded and whether parts of the CS are to be recorded
- Retention policies that determine how long a recording is stored
- Searching and accessing the recorded media and metadata
- Policies governing how CS users are made aware of recording
- Delivering additional recording session metadata through non-SIP mechanism

5. Overview of operations

This section is informative and provides a description of recording operations.

Section 6 describes the SIP communication in a recording session between a SRC and a SRS, and the procedures for recording-aware user agents participating in a CS. Section 7 describes the SDP in a recording session, and the procedures for recording indications and recording preferences. Section 8 describes the RTP handling in a recording session. Section 9 describes the mechanism to deliver recording metadata from the SRC to the SRS.

As mentioned in the architecture document [RFC7245], there are a number of types of call flows based on the location of the Session Recording Client. The following sample call flows provide a quick overview of the operations between the SRC and the SRS.

5.1. Delivering recorded media

When a SIP Back-to-Back User Agent (B2BUA) with SRC functionality routes a call from UA(A) to UA(B), the SRC has access to the media path between the user agents. When the SRC is aware that it should be recording the conversation, the SRC can cause the B2BUA to bridge the media between UA(A) and UA(B). The SRC then establishes the Recording Session with the SRS and sends replicated media towards the SRS.

An endpoint may also have SRC functionality, where the endpoint itself establishes the Recording Session to the SRS. Since the endpoint has access to the media in the Communication Session, the endpoint can send replicated media towards the SRS.

The following is a sample call flow that shows the SRC establishing a recording session towards the SRS. The call flow is essentially identical when the SRC is a B2BUA or as the endpoint itself. Note that the SRC can choose when to establish the Recording Session independent of the Communication Session, even though the following call flow suggests that the SRC is establishing the Recording Session (message #5) after the Communication Session is established.

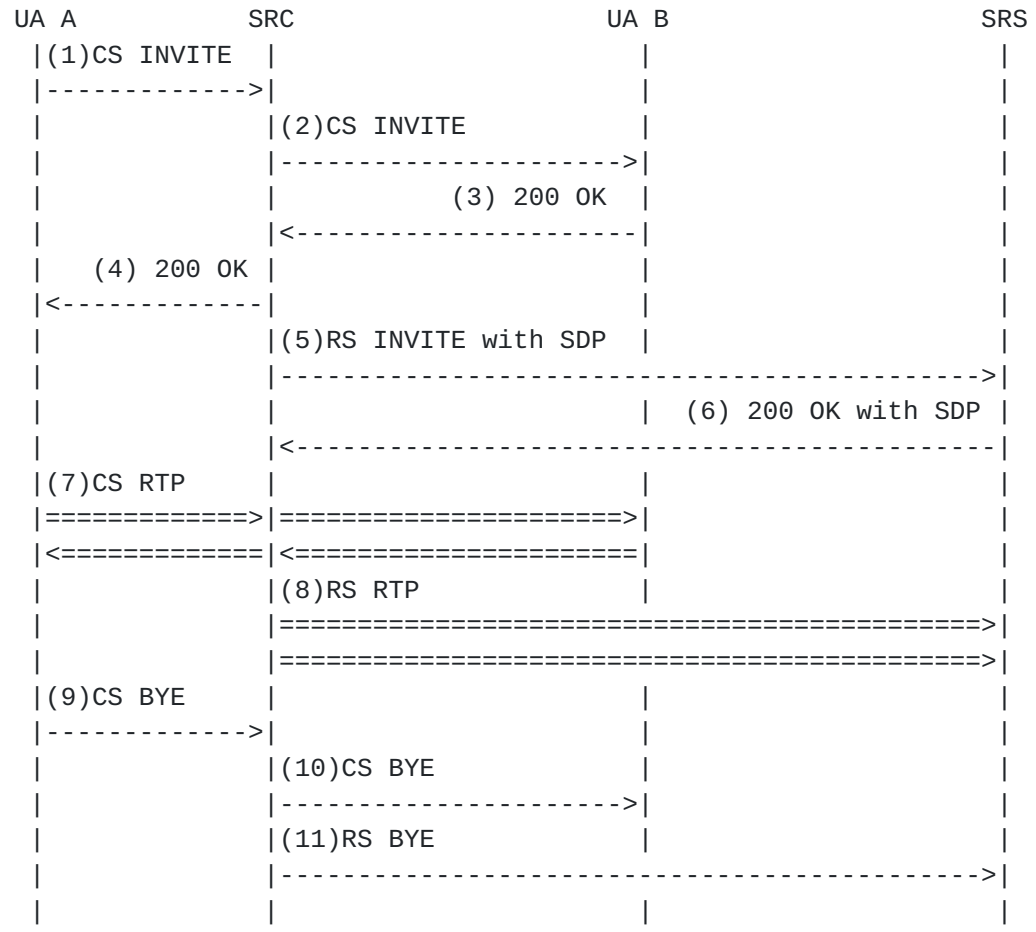


Figure 1: Basic recording call flow

The above call flow can also apply to the case of a centralized conference with a mixer. For clarity, ACKs to INVITEs and 200 OKs to BYEs are not shown. The conference focus can provide the SRC functionality since the conference focus has access to all the media

from each conference participant. When a recording is requested, the SRC delivers the metadata and the media streams to the SRS. Since the conference focus has access to a mixer, the SRC may choose to mix the media streams from all participants as a single mixed media stream towards the SRS.

An SRC can use a single recording session to record multiple communication sessions. Every time the SRC wants to record a new call, the SRC updates the recording session with a new SDP offer to add new recorded streams to the recording session, and correspondingly also update the metadata for the new call.

An SRS can also establish a recording session to an SRC, although it is beyond the scope of this document to define how an SRS would specify which calls to record.

5.2. Delivering recording metadata

The SRC is responsible for the delivery of metadata to the SRS. The SRC may provide an initial metadata snapshot about recorded media streams in the initial INVITE content in the recording session. Subsequent metadata updates can be represented as a stream of events in UPDATE or reINVITE requests sent by the SRC. These metadata updates are normally incremental updates to the initial metadata snapshot to optimize on the size of updates, however, the SRC may also decide to send a new metadata snapshot anytime.

Metadata is transported in the body of INVITE or UPDATE messages. Certain metadata, such as the attributes of the recorded media stream are located in the SDP of the recording session.

The SRS has the ability to send a request to the SRC to request for a new metadata snapshot update from the SRC. This can happen when the SRS fails to understand the current stream of incremental updates for whatever reason, for example, when SRS loses the current state due to internal failure. The SRS may optionally attach a reason along with the snapshot request. This request allows both SRC and SRS to synchronize the states with a new metadata snapshot so that further metadata incremental updates will be based on the latest metadata snapshot. Similar to the metadata content, the metadata snapshot request is transported as content in UPDATE or INVITE sent by the SRS in the recording session.

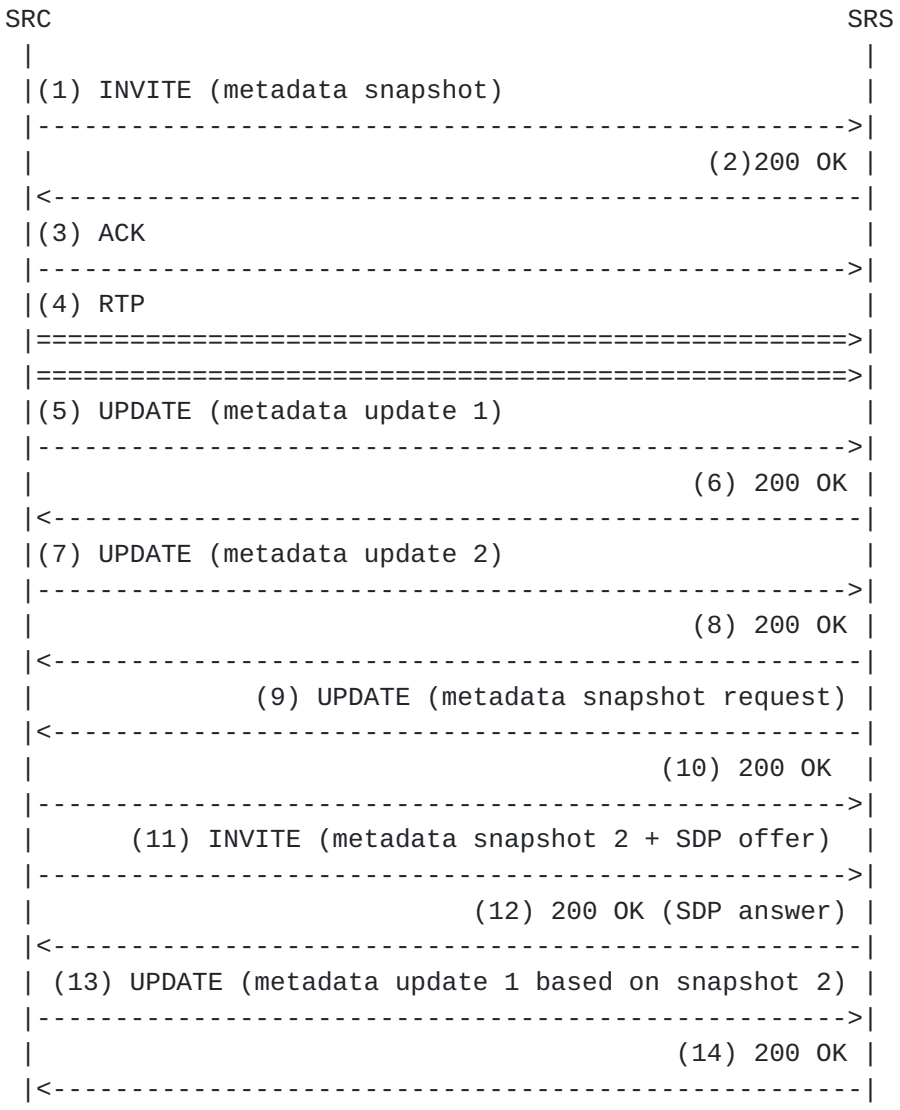


Figure 2: Delivering metadata via SIP UPDATE

5.3. Receiving recording indications and providing recording preferences

The SRC is responsible to provide recording indications to the participants in the CS. A recording-aware UA supports receiving recording indications via the SDP attribute a=record, and it can specify a recording preference in the CS by including the SDP attribute a=recordpref. The recording attribute is a declaration by the SRC in the CS to indicate whether recording is taking place. The recording preference attribute is a declaration by the recording-aware UA in the CS to indicate the recording preference.

To illustrate how the attributes are used, if a UA (A) is initiating a call to UA (B) and UA (A) is also an SRC that is performing the recording, then UA (A) provides the recording indication in the SDP offer with a=record:on. Since UA (A) is the SRC, UA (A) receives the recording indication from the SRC directly. When UA (B) receives the SDP offer, UA (B) will see that recording is happening on the other endpoint of this session. Since UA (B) is not an SRC and does not provide any recording preference, the SDP answer does not contain a=record nor a=recordpref.

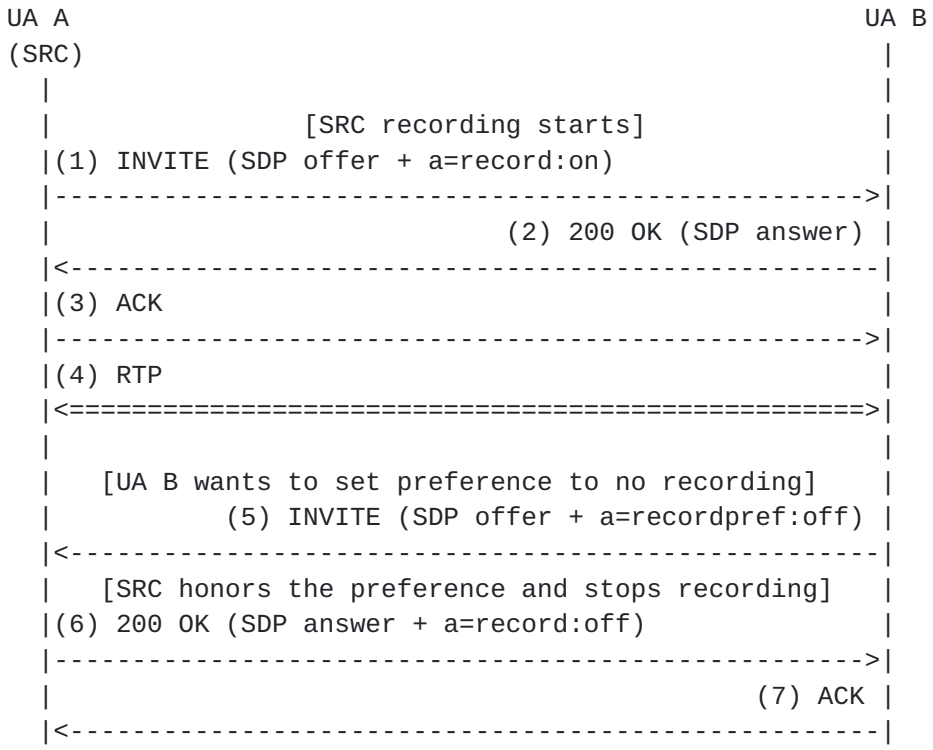


Figure 3: Recording indication and recording preference

After the call is established and recording is in progress, UA (B) later decides to change the recording preference to no recording and sends a reINVITE with the a=recordpref attribute. It is up to the SRC to honor the preference, and in this case SRC decides to stop the recording and updates the recording indication in the SDP answer.

6. SIP Handling

6.1. Procedures at the SRC

6.1.1. Initiating a Recording Session

A recording session is a SIP session with specific extensions applied, and these extensions are listed in the procedures for SRC and SRS below. When an SRC or an SRS receives a SIP session that is not a recording session, it is up to the SRC or the SRS to determine what to do with the SIP session.

The SRC can initiate a recording session by sending a SIP INVITE request to the SRS. The SRC and the SRS are identified in the From and To headers, respectively.

The SRC MUST include the '+sip.src' feature tag in the Contact URI, defined in this specification as an extension to [RFC3840], for all recording sessions. An SRS uses the presence of the '+sip.src' feature tag in dialog creating and modifying requests and responses to confirm that the dialog being created is for the purpose of a Recording Session. In addition, when an SRC sends a REGISTER request to a registrar, the SRC MAY include the '+sip.src' feature tag to indicate the that it is a SRC.

Since SIP Caller Preferences extensions are optional to implement for routing proxies, there is no guarantee that a recording session will be routed to an SRC or SRS. A new options tag is introduced: "siprec". As per [RFC3261], only an SRC or an SRS can accept this option tag in a recording session. An SRC MUST include the "siprec" option tag in the Require header when initiating a Recording Session so that UA's which do not support the session recording protocol extensions will simply reject the INVITE request with a 420 Bad Extension.

When an SRC receives a new INVITE, the SRC MUST only consider the SIP session as a recording session when both the '+sip.srs' feature tag and 'siprec' option tag are included in the INVITE request.

6.1.2. SIP extensions for recording indication and preference

For the communication session, the SRC MUST provide recording indication to all participants in the CS. A participant UA in a CS can indicate that it is recording-aware by providing the "record-aware" option tag, and the SRC MUST provide recording indications in the new SDP a=record attribute described in the SDP Handling section. In the absence of the "record-aware" option tag, meaning that the participant UA is not recording-aware, an SRC MUST provide recording indications through other means such as playing a tone in-band, if the SRC is required to do so (e.g. based on policies).

An SRC in the CS may also indicate itself as a session recording client by including the '+sip.src' feature tag. A recording-aware participant can learn that a SRC is in the CS, and can set the recording preference for the CS with the new SDP a=recordpref attribute described in the SDP Handling section below.

6.2. Procedures at the SRS

When an SRS receives a new INVITE, the SRS MUST only consider the SIP session as a recording session when both the '+sip.src' feature tag and 'siprec' option tag are included in the INVITE request.

The SRS can initiate a recording session by sending a SIP INVITE request to the SRC. The SRS and the SRC are identified in the From and To headers, respectively.

The SRS MUST include the '+sip.srs' feature tag in the Contact URI, as per [RFC3840], for all recording sessions. An SRC uses the presence of this feature tag in dialog creating and modifying requests and responses to confirm that the dialog being created is for the purpose of a Recording Session (REQ-30). In addition, when an SRS sends a REGISTER request to a registrar, the SRS SHOULD include the '+sip.srs' feature tag to indicate that it is a SRS.

An SRS MUST include the "siprec" option tag in the Require header as per [RFC3261] when initiating a Recording Session so that UA's which do not support the session recording protocol extensions will simply reject the INVITE request with a 420 Bad Extension.

6.3. Procedures for Recording-aware User Agents

A recording-aware user agent is a participant in the CS that supports the SIP and SDP extensions for receiving recording indication and for requesting recording preferences for the call. A recording-aware UA MUST indicate that it can accept reporting of recording indication provided by the SRC with a new option tag "record-aware" when initiating or establishing a CS, meaning including the "record-aware" tag in the Supported header in the initial INVITE request or response.

A recording-aware UA MUST be prepared to provide a recording indication to the end user through an appropriate user interface, indicating whether recording is on, off, or paused for each medium. Some user agents that are automatons (e.g. IVR, media server, PSTN gateway) may not have a user interface to render recording indication. When such user agent indicates recording awareness, the UA SHOULD render recording indication through other means, such as passing an in-band tone on the PSTN gateway, putting the recording

indication in a log file, or raising an application event in a VoiceXML dialog. These user agents MAY also choose not to indicate recording awareness, thereby relying on whatever mechanism an SRC chooses to indicate recording, such as playing a tone in-band.

7. SDP Handling

7.1. Procedures at the SRC

The SRC and SRS follows the SDP offer/answer model in [RFC3264]. The procedures for SRC and SRS describe the conventions used in a recording session.

7.1.1. SDP handling in RS

Since the SRC does not expect to receive media from the SRS, the SRC typically sets each media stream of the SDP offer to only send media, by qualifying them with the a=sendonly attribute, according to the procedures in [RFC3264].

The SRC sends recorded streams of participants to the SRS, and the SRC MUST provide a label attribute (a=label), as per [RFC4574], on each media stream in order to identify the recorded stream with the rest of the metadata. The a=label attribute identifies each recorded media stream, and the label name is mapped to the Media Stream Reference in the metadata as per [I-D.ietf-siprec-metadata]. The scope of the a=label attribute only applies to the SDP and Metadata conveyed in the bodies of the SIP request or response that the label appeared in. Note that a recorded stream is distinct from a CS stream; the metadata provides a list of participants that contributes to each recorded stream.

The following is an example SDP offer from SRC with both audio and video recorded streams. Note that the following example contains unfolded lines longer than 72 characters. These are captured between <allOneLine> tags.

```
v=0
o=SRC 2890844526 2890844526 IN IP4 198.51.100.1
s=-
c=IN IP4 198.51.100.1
t=0 0
m=audio 12240 RTP/AVP 0 4 8
a=sendonly
a=label:1
m=video 22456 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=sendonly
a=label:2
m=audio 12242 RTP/AVP 0 4 8
a=sendonly
a=label:3
m=video 22458 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=sendonly
a=label:4
```

Figure 4: Sample SDP offer from SRC with audio and video streams

7.1.1.1. Handling media stream updates

Over the lifetime of a recording session, the SRC can add and remove recorded streams from the recording session for various reasons. For example, when a CS stream is added or removed from the CS, or when a CS is created or terminated if a recording session handles multiple CSes. To remove a recorded stream from the recording session, the SRC sends a new SDP offer where the port of the media stream to be removed is set to zero, according to the procedures in [RFC3264]. To add a recorded stream to the recording session, the SRC sends a new SDP offer by adding a new media stream description or by reusing an old media stream which had been previously disabled, according to the procedures in [RFC3264].

The SRC can temporarily discontinue streaming and collection of recorded media from the SRC to the SRS for reasons such as masking

the recording. In this case, the SRC sends a new SDP offer and sets the media stream to inactive (a=inactive) for each recorded stream to be paused, as per the procedures in [RFC3264]. To resume streaming and collection of recorded media, the SRC sends a new SDP offer and sets the media stream to sendonly (a=sendonly). Note that a CS itself may change the media stream direction by updating the SDP, for example, by setting a=inactive for SDP hold. Media stream direction changes in CS are conveyed in the metadata by the SRC. When a CS media stream is changed to/from inactive, the effect on the corresponding RS media stream is governed by SRC policy. The SRC MAY have a local policy to pause an RS media stream when the corresponding CS media stream is inactive, or it MAY leave the RS media stream as sendonly.

7.1.2. Recording indication in CS

While there are existing mechanisms for providing an indication that a CS is being recorded, these mechanisms are usually delivered on the CS media streams such as playing an in-band tone or an announcement to the participants. A new 'record' SDP attribute is introduced to allow the SRC to indicate recording state to a recording-aware UA in CS.

The 'record' SDP attribute appears at the media level or session level in either SDP offer or answer. When the attribute is applied at the session level, the indication applies to all media streams in the SDP. When the attribute is applied at the media level, the indication applies to the media stream only, and that overrides the indication if also set at the session level. Whenever the recording indication needs to change, such as termination of recording, then the SRC MUST initiate a reINVITE or UPDATE to update the SDP a=record attribute.

The following is the ABNF of the 'record' attribute:

```
attribute /= record-attr
; attribute defined in RFC 4566

record-attr = "record:" indication
indication = "on" / "off" / "paused"
```

on: Recording is in progress.

off: No recording is in progress.

paused: Recording is in progress but media is paused.

7.1.3. Recording preference in CS

When the SRC receives the a=recordpref SDP in an SDP offer or answer, the SRC chooses to honor the preference to record based on local policy at the SRC. If the SRC makes a change in recording state, the SRC MUST report the new recording state in the a=record attribute in the SDP answer or in a subsequent SDP offer.

7.2. Procedures at the SRS

Typically the SRS only receives RTP streams from the SRC; therefore, the SDP offer/answer from the SRS normally sets each media stream to receive media, by setting them with the a=recvonly attribute, according to the procedures of [RFC3264]. When the SRS is not ready to receive a recorded stream, the SRS sets the media stream as inactive in the SDP offer or answer by setting it with a=inactive attribute, according to the procedures of [RFC3264]. When the SRS is ready to receive recorded streams, the SRS sends a new SDP offer and sets the media streams with a=recvonly attribute.

The following is an example of SDP answer from SRS for the SDP offer from the above sample. Note that the following example contain unfolded lines longer than 72 characters. These are captured between <allOneLine> tags.

```
v=0
o=SRS 0 0 IN IP4 198.51.100.20
s=-
c=IN IP4 198.51.100.20
t=0 0
m=audio 10000 RTP/AVP 0
a=recvonly
a=label:1
m=video 10002 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=recvonly
a=label:2
m=audio 10004 RTP/AVP 0
a=recvonly
a=label:3
m=video 10006 RTP/AVP 98
a=rtpmap:98 H264/90000
<allOneLine>
a=fmtp:98 profile-level-id=42A01E;
          sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
</allOneLine>
a=recvonly
a=label:4
```

Figure 5: Sample SDP answer from SRS with audio and video streams

Over the lifetime of a recording session, the SRS can remove recorded streams from the recording session for various reasons. To remove a recorded stream from the recording session, the SRS sends a new SDP offer where the port of the media stream to be removed is set to zero, according to the procedures in [RFC3264].

The SRS SHOULD NOT add recorded streams in the recording session when SRS sends a new SDP offer. Similarly, when the SRS starts a recording session, the SRS SHOULD initiate the INVITE without an SDP offer to let the SRC generate the SDP offer with recorded streams.

The following sequence diagram shows an example where the SRS is initially not ready to receive recorded streams, and later updates the recording session when the SRS is ready to record.

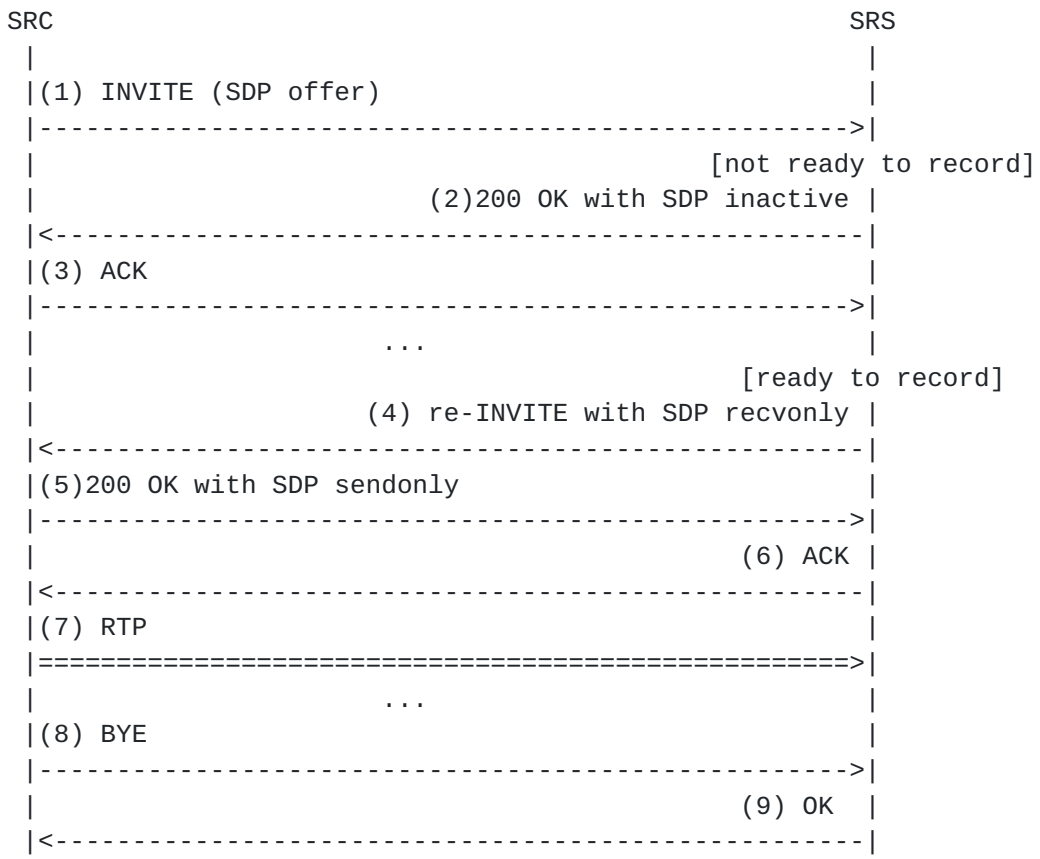


Figure 6: SRS responding to offer with a=inactive

7.3. Procedures for Recording-aware User Agents

7.3.1. Recording indication

When a recording-aware UA receives an SDP offer or answer that includes the a=record attribute, the UA MUST provide the recording indication to the end user whether the recording is on, off, or paused for each medium based on the most recently received a=record SDP attribute for that medium.

When a CS is traversed through multiple UAs such as a B2BUA or a conference focus, each UA involved in the CS that is aware that the CS is being recorded MUST provide the recording indication through the a=record attribute to all other parties in the CS.

It is possible that more than one SRC is in the call path of the same CS, but the recording indication attribute does not provide any hint as to which SRC or how many SRCs are recording. An endpoint knows

only that the call is being recorded. Furthermore, this attribute is not used as a request for a specific SRC to start/stop recording.

7.3.2. Recording preference

A participant in a CS MAY set the recording preference in the CS to be recorded or not recorded at session establishment or during the session. A new 'recordpref' SDP attribute is introduced, and the participant in CS may set this recording preference attribute in any SDP offer/answer at session establishment time or during the session. The SRC is not required to honor the recording preference from a participant based on local policies at the SRC, and the participant can learn the recording indication through the a=record SDP attribute as described in the above section.

The SDP a=recordpref attribute can appear at the media level or session level and can appear in an SDP offer or answer. When the attribute is applied at the session level, the recording preference applies to all media stream in the SDP. When the attribute is applied at the media level, the recording preference applies to the media stream only, and that overrides the recording preference if also set at the session level. The user agent can change the recording preference by changing the a=recordpref attribute in subsequent SDP offer or answer. The absence of the a=recordpref attribute in the SDP indicates that the UA has no recording preference.

The following is the ABNF of the recordpref attribute:

```
attribute /= recordpref-attr
; attribute defined in RFC 4566

recordpref-attr = "a=recordpref:" pref
pref = "on" / "off" / "pause" / "nopreference"
```

on: Sets the preference to record if it has not already been started. If the recording is currently paused, the preference is to resume recording.

off: Sets the preference for no recording. If recording has already been started, then the preference is to stop the recording.

pause: If the recording is currently in progress, sets the preference to pause the recording.

nopreference: To indicate that the UA has no preference on recording.

8. RTP Handling

This section provides recommendations and guidelines for RTP and RTCP in the context of SIPREC. In order to communicate most effectively, the Session Recording Client (SRC), the Session Recording Server (SRS), and any Recording aware User Agents (UAs) should utilize the mechanisms provided by RTP in a well-defined and predicable manner. It is the goal of this document to make the reader aware of these mechanisms and provide recommendations and guidelines.

8.1. RTP Mechanisms

This section briefly describes important RTP/RTCP constructs and mechanisms that are particularly useful within the content of SIPREC.

8.1.1. RTCP

The RTP data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery. RTCP, as defined in [RFC3550], is based on the periodic transmission of control packets to all participants in the RTP session, using the same distribution mechanism as the data packets. Support for RTCP is REQUIRED, per [RFC3550], and it provides, among other things, the following important functionality in relation to SIPREC:

1) Feedback on the quality of the data distribution

This feedback from the receivers may be used to diagnose faults in the distribution. As such, RTCP is a well-defined and efficient mechanism for the SRS to inform the SRC, and for the SRC to inform Recording aware UAs, of issues that arise with respect to the reception of media that is to be recorded.

2) Carries a persistent transport-level identifier for an RTP source called the canonical name or CNAME

The SSRC identifier may change if a conflict is discovered or a program is restarted; in which case receivers can use the CNAME to keep track of each participant. Receivers may also use the CNAME to associate multiple data streams from a given participant in a set of related RTP sessions, for example to synchronize audio and video. Synchronization of media streams is also facilitated by the NTP and RTP timestamps included in RTCP packets by data senders.

8.1.2. RTP Profile

The RECOMMENDED RTP profiles for the SRC, SRS, and Recording aware UAs are "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", [RFC5124] when using encrypted RTP streams, and "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", [RFC4585] when using non encrypted media streams. However, as this is not a requirement, some implementations may use "The Secure Real-time Transport Protocol (SRTP)", [RFC3711] and "RTP Profile for Audio and Video Conferences with Minimal Control", AVP [RFC3551]. Therefore, it is RECOMMENDED that the SRC, SRS, and Recording aware UAs not rely entirely on SAVPF or AVPF for core functionality that may be at least partially achievable using SAVP and AVP.

AVPF and SAVPF provide an improved RTCP timer model that allows more flexible transmission of RTCP packets in response to events, rather than strictly according to bandwidth. AVPF based codec control messages provide efficient mechanisms for an SRC, SRS, and Recording aware UAs to handle events such as scene changes, error recovery, and dynamic bandwidth adjustments. These messages are discussed in more detail later in this document.

SAVP and SAVPF provide media encryption, integrity protection, replay protection, and a limited form of source authentication. They do not contain or require a specific keying mechanism.

8.1.3. SSRC

The synchronization source (SSRC), as defined in [RFC3550] is carried in the RTP header and in various fields of RTCP packets. It is a random 32-bit number that is required to be globally unique within an RTP session. It is crucial that the number be chosen with care in order that participants on the same network or starting at the same time are not likely to choose the same number. Guidelines regarding SSRC value selection and conflict resolution are provided in [RFC3550].

The SSRC may also be used to separate different sources of media within a single RTP session. For this reason as well as for conflict resolution, it is important that the SRC, SRS, and Recording aware UAs handle changes in SSRC values and properly identify the reason of the change. The CNAME values carried in RTCP facilitate this identification.

8.1.4. CSRC

The contributing source (CSRC), as defined in [RFC3550], identifies the source of a stream of RTP packets that has contributed to the combined stream produced by an RTP mixer. The mixer inserts a list of the SSRC identifiers of the sources that contributed to the generation of a particular packet into the RTP header of that packet. This list is called the CSRC list. It is RECOMMENDED that a SRC or Recording aware UA, when acting a mixer, sets the CSRC list accordingly, and that the SRC and SRS interpret the CSRC list appropriately when received.

8.1.5. SDES

The Source Description (SDES), as defined in [RFC3550], contains an SSRC/CSRC identifier followed by a list of zero or more items, which carry information about the SSRC/CSRC. End systems send one SDES packet containing their own source identifier (the same as the SSRC in the fixed RTP header). A mixer sends one SDES packet containing a chunk for each contributing source from which it is receiving SDES information, or multiple complete SDES packets if there are more than 31 such sources.

8.1.5.1. CNAME

The Canonical End-Point Identifier (CNAME), as defined in [RFC3550], provides the binding from the SSRC identifier to an identifier for the source (sender or receiver) that remains constant. It is important the SRC and Recording aware UAs generate CNAMEs appropriately and that the SRC and SRS interpret and use them for this purpose. Guidelines for generating CNAME values are provided in "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)" [RFC7022].

8.1.6. Keepalive

It is anticipated that media streams in SIPREC may exist in an inactive state for extended periods of times for any of a number of valid reasons. In order for the bindings and any pinholes in NATs/ firewalls to remain active during such intervals, it is RECOMMENDED that the SRC, SRS, and Recording aware UAs follow the keep-alive procedure recommended in "Application Mechanism for Keeping Alive the NAT Mappings Associated to RTP/RTP Control Protocol (RTCP) Flows" [RFC6263] for all RTP media streams.

8.1.7. RTCP Feedback Messages

"Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)" [RFC5104] specifies extensions to the messages defined in AVPF [RFC4585]. Support for and proper usage of these messages is important to SRC, SRS, and Recording aware UA implementations. Note that these messages are applicable only when using the AVFP or SAVPF RTP profiles

8.1.7.1. Full Intra Request

A Full Intra Request (FIR) Command, when received by the designated media sender, requires that the media sender sends a Decoder Refresh Point at the earliest opportunity. Using a decoder refresh point implies refraining from using any picture sent prior to that point as a reference for the encoding process of any subsequent picture sent in the stream.

Decoder refresh points, especially Intra or IDR pictures for H.264 video codecs, are in general several times larger in size than predicted pictures. Thus, in scenarios in which the available bit rate is small, the use of a decoder refresh point implies a delay that is significantly longer than the typical picture duration.

8.1.7.1.1. SIP INFO for FIR

"XML Schema for Media Control" [RFC5168] defines an Extensible Markup Language (XML) Schema for video fast update. Implementations are discouraged from using the method described except for backward compatibility purposes. Implementations SHOULD use FIR messages instead.

To make sure a common mechanism exists between the SRC and SRS, the SRS MUST support both mechanisms (FIR and SIP INFO), using FIR when negotiated successfully with the SRC, and using SIP INFO otherwise.

8.1.7.2. Picture Loss Indicator

Picture Loss Indication (PLI), as defined in [RFC4585], informs the encoder of the loss of an undefined amount of coded video data belonging to one or more pictures. [RFC4585] recommends using PLI instead of FIR to recover from errors. FIR is appropriate only in situations where not sending a decoder refresh point would render the video unusable for the users. Examples where sending FIR is appropriate include a multipoint conference when a new user joins the conference and no regular decoder refresh point interval is established, and a video switching MCU that changes streams.

8.1.7.3. Temporary Maximum Media Stream Bit Rate Request

A receiver, translator, or mixer uses the Temporary Maximum Media Stream Bit Rate Request (TMMBR) to request a sender to limit the maximum bit rate for a media stream to the provided value. Appropriate use of TMMBR facilitates rapid adaptation to changes in available bandwidth.

8.1.7.3.1. Renegotiation of SDP bandwidth attribute

If it is likely that the new value indicated by TMMBR will be valid for the remainder of the session, the TMMBR sender is expected to perform a renegotiation of the session upper limit using the session signaling protocol. Therefore for SIPREC, implementations are RECOMMENDED to use TMMBR for temporary changes, and renegotiation of bandwidth via SDP offer/answer for more permanent changes.

8.1.8. Symmetric RTP/RTCP for Sending and Receiving

Within an SDP offer/answer exchange, RTP entities choose the RTP and RTCP transport addresses (i.e., IP addresses and port numbers) on which to receive packets. When sending packets, the RTP entities may use the same source port or a different source port as those signaled for receiving packets. When the transport address used to send and receive RTP is the same, it is termed "symmetric RTP" [RFC4961]. Likewise, when the transport address used to send and receive RTCP is the same, it is termed "symmetric RTCP" [RFC4961].

When sending RTP, it is REQUIRED to use symmetric RTP. When sending RTCP, it is REQUIRED to use symmetric RTCP. Although an SRS will not normally send RTP, it will send RTCP as well as receive RTP and RTCP. Likewise, although an SRC will not normally receive RTP from the SRS, it will receive RTCP as well as send RTP and RTCP.

> Note: Symmetric RTP and symmetric RTCP are different from RTP/RTCP multiplexing [RFC5761].

8.2. Roles

An SRC has the task of gathering media from the various UAs in one or more Communication Sessions (CSs) and forwarding the information to the SRS within the context of a corresponding Recording Session (RS). There are numerous ways in which an SRC may do this, including but not limited to, appearing as a UA within a CS, or as a B2BUA between UAs within a CS.

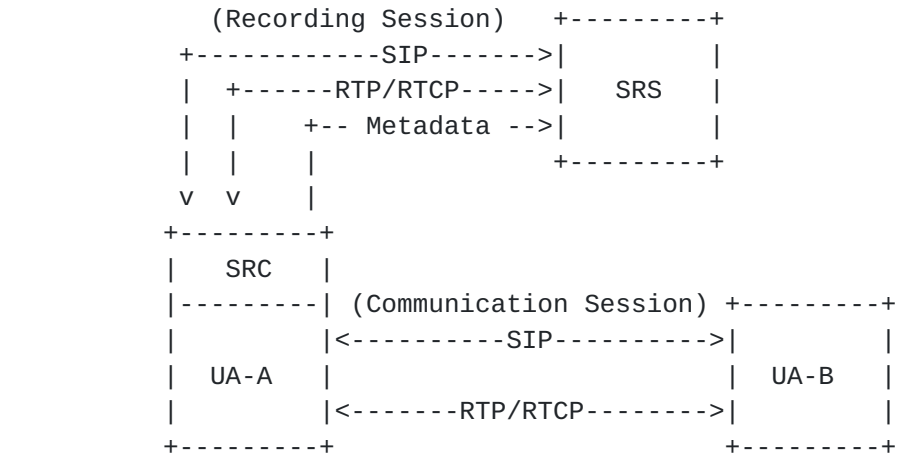


Figure 7: UA as SRC

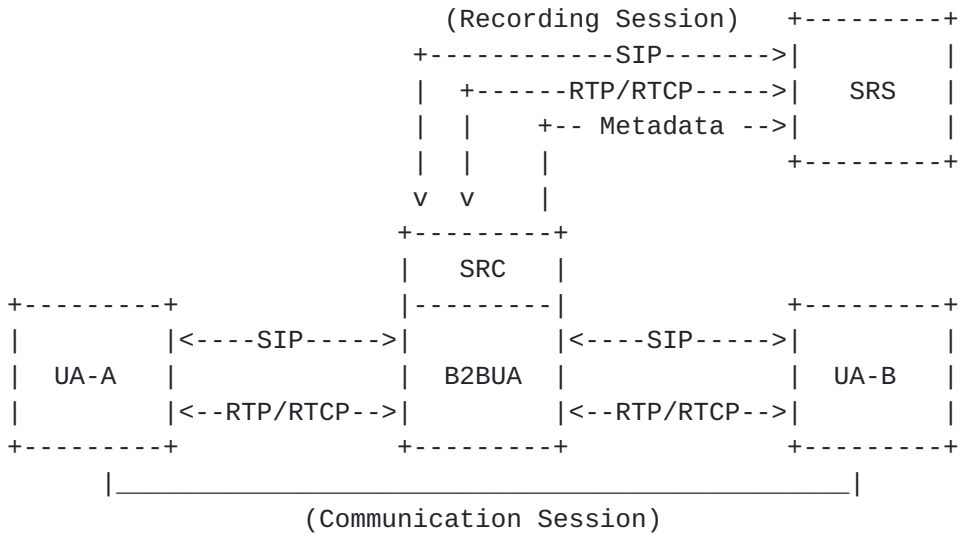


Figure 8: B2BUA as SRC

The following subsections define a set of roles an SRC may choose to play based on its position with respect to a UA within a CS, and an SRS within an RS. A CS and a corresponding RS are independent sessions; therefore, an SRC may play a different role within a CS than it does within the corresponding RS.

8.2.1. SRC acting as an RTP Translator

The SRC may act as a translator, as defined in [RFC3550]. A defining characteristic of a translator is that it forwards RTP packets with their SSRC identifier intact. There are two types of translators, one that simply forwards, and another that performs transcoding (e.g., from one codec to another) in addition to forwarding.

8.2.1.1. Forwarding Translator

When acting as a forwarding translator, RTP received as separate streams from different sources (e.g., from different UAs with different SSRCs) cannot be mixed by the SRC and MUST be sent separately to the SRS. All RTCP reports MUST be passed by the SRC between the UAs and the SRS, such that the UAs and SRS are able to detect any SSRC collisions.

RTCP Sender Reports generated by a UA sending a stream MUST be forwarded to the SRS. RTCP Receiver Reports generated by the SRS MUST be forwarded to the relevant UA.

UAs may receive multiple sets of RTCP Receiver Reports, one or more from other UAs participating in the CS, and one from the SRS participating in the RS. A Recording aware UA SHOULD be prepared to process the RTCP Receiver Reports from the SRS, whereas a recording unaware UA may discard such RTCP packets as not of relevance.

If SRTP is used on both the CS and the RS, decryption and/or re-encryption may occur. For example, if different keys are used, it will occur. If the same keys are used, it need not occur. Section 12 provides additional information on SRTP and keying mechanisms.

If packet loss occurs, either from the UA to the SRC or from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss. The SRC does not play a role in this other than forwarding the associated RTP and RTCP packets.

8.2.1.2. Transcoding Translator

When acting as a transcoding translator, an SRC MAY perform transcoding (e.g., from one codec to another), and this may result in a different rate of packets between what the SRC receives on the CS and what the SRC sends on the RS. As when acting as a forwarding translator, RTP received as separate streams from different sources (e.g., from different UAs with different SSRCs) cannot be mixed by the SRC and MUST be sent separately to the SRS. All RTCP reports MUST be passed by the SRC between the UAs and the SRS, such that the UAs and SRS are able to detect any SSRC collisions.

RTCP Sender Reports generated by a UA sending a stream MUST be forwarded to the SRS. RTCP Receiver Reports generated by the SRS MUST be forwarded to the relevant UA. The SRC may need to manipulate the RTCP Receiver Reports to take account of any transcoding that has taken place.

UAs may receive multiple sets of RTCP Receiver Reports, one or more from other UAs participating in the CS, and one from the SRS participating in the RS. A Recording aware UA SHOULD be prepared to process the RTCP Receiver Reports from the SRS, whereas a recording unaware UA may discard such RTCP packets as not of relevance.

If SRTP is used on both the CS and the RS, decryption and/or re-encryption may occur. For example, if different keys are used, it will occur. If the same keys are used, it need not occur. Section 12 provides additional information on SRTP and keying mechanisms.

If packet loss occurs, either from the UA to the SRC or from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss. The SRC does not play a role in this other than forwarding the associated RTP and RTCP packets.

8.2.2. SRC acting as an RTP Mixer

In the case of the SRC acting as a RTP mixer, as defined in [RFC3550], the SRC combines RTP streams from different UA and sends them towards the SRS using its own SSRC. The SSRCs from the contributing UA SHOULD be conveyed as CSRCs identifiers within this stream. The SRC may make timing adjustments among the received streams and generate its own timing on the stream sent to the SRS. Optionally an SRC acting as a mixer can perform transcoding, and can even cope with different codings received from different UAs. RTCP Sender Reports and Receiver Reports are not forwarded by an SRC acting as mixer, but there are requirements for forwarding RTCP Source Description (SDES) packets. The SRC generates its own RTCP Sender and Receiver reports toward the associated UAs and SRS.

The use of SRTP between the SRC and the SRS for the RS is independent of the use of SRTP between the UAs and SRC for the CS. Section 12 provides additional information on SRTP and keying mechanisms.

If packet loss occurs from the UA to the SRC, the SRC SHOULD detect and attempt to recover from the loss. If packet loss occurs from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss.

8.2.3. SRC acting as an RTP Endpoint

The case of the SRC acting as an RTP endpoint, as defined in [RFC3550], is similar to the mixer case, except that the RTP session between the SRC and the SRS is considered completely independent from the RTP session that is part of the CS. The SRC can, but need not, mix RTP streams from different participants prior to sending to the

SRS. RTCP between the SRC and the SRS is completely independent of RTCP on the CS.

The use of SRTP between the SRC and the SRS for the RS is independent of the use of SRTP between the UAs and SRC for the CS. Section 12 provides additional information on SRTP and keying mechanisms.

If packet loss occurs from the UA to the SRC, the SRC SHOULD detect and attempt to recover from the loss. If packet loss occurs from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss.

8.3. Packet Loss

If packet loss occurs from the UA to the SRC, the SRC, if acting as an RTP mixer or RTP endpoint, SHOULD detect and attempt to recover from the loss. If acting as an RTP translator (either forwarding or transcoding), the SRC does not detect or attempt to recover from the loss. It simply forwards the associated RTP and RTCP packets. In such cases, as well as if packet loss occurs from the SRC to the SRS, the SRS SHOULD detect and attempt to recover from the loss.

8.4. RTP Session Usage by SRC

There are multiple ways that an SRC may choose to deliver recorded media to an SRS. In some cases, it may use a single RTP session for all media within the RS, whereas in others it may use multiple RTP sessions. The following subsections provide examples of basic RTP session usage by the SRC, including a discussion of how the RTP constructs and mechanisms covered previously are used. An SRC may choose to use one or more of the RTP session usages within a single RS. For the purpose of base interoperability between SRC and SRS, an SRC MUST support separate m-lines in SDP, one per CS media direction. The set of RTP session usages described is not meant to be exhaustive.

8.4.1. SRC Using Multiple m-lines

When using multiple m-lines, an SRC includes each m-line in an SDP offer to the SRS. The SDP answer from the SRS MUST include all m-lines, with any rejected m-lines indicated with a zero port, per [RFC3264]. Having received the answer, the SRC starts sending media to the SRS as indicated in the answer. Alternatively, if the SRC deems the level of support indicated in the answer to be unacceptable, it may initiate another SDP offer/answer exchange in which an alternative RTP session usage is negotiated.

In order to preserve the mapping of media to participant within the CSs in the RS, the SRC SHOULD map each unique CNAME within the CSs to a unique CNAME within the RS. Additionally, the SRC SHOULD map each unique combination of CNAME/SSRC within the CSs to a unique CNAME/SSRC within the RS. In doing so, the SRC may act as an RTP translator or as an RTP endpoint.

The following figure illustrates a case in which each UA represents a participant contributing two RTP sessions (e.g. one for audio and one for video), each with a single SSRC. The SRC acts as an RTP translator and delivers the media to the SRS using four RTP sessions, each with a single SSRC. The CNAME and SSRC values used by the UAs within their media streams are preserved in the media streams from the SRC to the SRS.

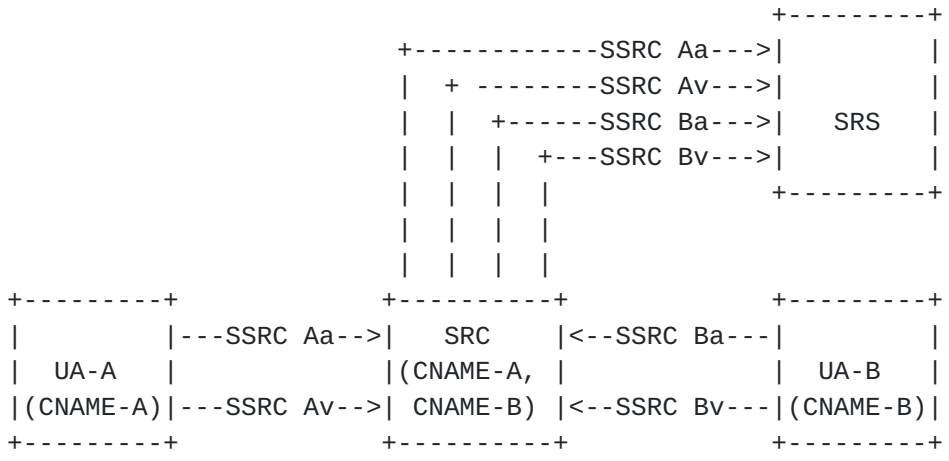


Figure 9: SRC Using Multiple m-lines

8.4.2. SRC Using Mixing

When using mixing, the SRC combines RTP streams from different participants and sends them towards the SRS using its own SSRC. The SSRCs from the contributing participants SHOULD be conveyed as CSRCs identifiers. The SRC includes one m-line for each RTP session in an SDP offer to the SRS. The SDP answer from the SRS MUST include all m-lines, with any rejected m-lines indicated with the zero port, per [RFC3264]. Having received the answer, the SRC starts sending media to the SRS as indicated in the answer.

In order to preserve the mapping of media to participant within the CSs in the RS, the SRC SHOULD map each unique CNAME within the CSs to a unique CNAME within the RS. Additionally, the SRC SHOULD map each unique combination of CNAME/SSRC within the CSs to a unique CNAME/SSRC within the RS. The SRC MUST avoid SSRC collisions, rewriting

SSRCs if necessary when used as CSRCs in the RS. In doing to, the SRC acts as an RTP mixer.

In the event the SRS does not support this usage of CSRC values, it relies entirely on the SIPREC metadata to determine the participants included within each mixed stream.

The following figure illustrates a case in which each UA represents a participant contributing two RTP sessions (e.g. one for audio and one for video), each with a single SSRC. The SRC acts as an RTP mixer and delivers the media to the SRS using two RTP sessions, mixing media from each participant into a single RTP session containing a single SSRC and two CSRCs.

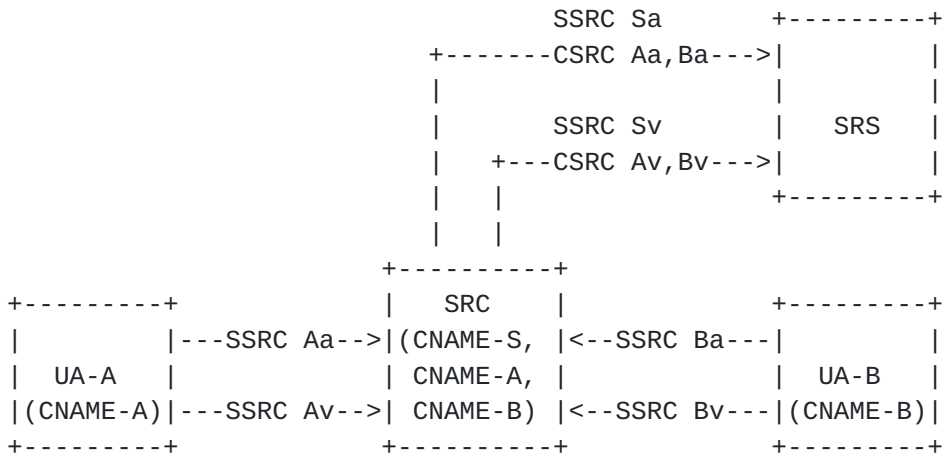


Figure 10: SRC Using Mixing

8.5. RTP Session Usage by SRS

An SRS that supports recording an audio CS MUST support SRC usage of separate audio m-lines in SDP, one per CS media direction. An SRS that supports recording an video CS MUST support SRC usage of separate video m-lines in SDP, one per CS media direction. Therefore, for an SRS supporting a typical audio call, the SRS has to support receiving at least two audio m-lines. For an SRS supporting a typical audio and video call, the SRS has to support receiving at least four total m-lines in the SDP, two audio m-lines and two video m-lines.

These requirements allow an SRS to be implemented that supports video only, without requiring support for audio recording. They also allow an SRS to be implemented that supports recording only one direction of one stream in a CS; for example, an SRS designed to record security monitoring cameras that only send (not receive) video without any audio. These requirements were not written to prevent

other modes being implemented and used, such as using a single m-line and mixing the separate audio streams together. Rather, the requirements were written to provide a common base mode to implement for the sake of interoperability. It is important to note that an SRS implementation supporting the common base may not record all media streams in a CS if a participant supports more than one m-line in a video call, such as one for camera and one for presentation. SRS implementations may support other modes as well, but have to at least support the ones above such that they interoperate in the common base mode for basic interoperability.

9. Metadata

Some metadata attributes are contained in SDP, and others are contained in a new content type "application/rs-metadata". The format of the metadata is described as part of the mechanism in [I-D.ietf-siprec-metadata]. A new "disposition-type" of Content-Disposition is defined for the purpose of carrying metadata. The value is "recording-session", which indicates the "application/rs-metadata" content contains metadata to be handled by the SRS.

9.1. Procedures at the SRC

The SRC MUST send metadata to the SRS in an RS. The SRC SHOULD send metadata as soon as it becomes available and whenever it changes. Cases in which an SRC may be justified in waiting temporarily before sending metadata include:

- waiting for previous metadata exchange to complete (i.e. cannot send another SDP offer until previous offer/answer completes, and may prefer not to send an UPDATE during this time either).
- constraining the signaling rate on the RS.
- sending metadata when key events occur rather than for every event that has any impact on metadata.
- desire to suppress certain metadata out of concern for privacy or perceived lack of need for it to be included in the recording.

Metadata sent by the SRC is categorized as either a full metadata snapshot or a partial update. A full metadata snapshot describes all metadata associated with the RS. The SRC MAY send a full metadata snapshot at any time. The SRC MAY send a partial update only if a full metadata snapshot has been sent previously.

The SRC MAY send metadata (either a full metadata snapshot or a partial update) in an INVITE request, an UPDATE request [RFC3311], or

an 200 response to an offerless INVITE from the SRS. If the metadata contains a reference to any SDP labels, the request containing the metadata MUST also contain an SDP offer that defines those labels.

When a SIP message contains both an SDP offer and metadata, the request body MUST have content type "multipart/mixed", with one subordinate body part containing the SDP offer and another containing the metadata. When a SIP message contains only an SDP offer or metadata, the "multipart/mixed" container is optional.

The SRC SHOULD include a full metadata snapshot in the initial INVITE request establishing the RS. If metadata is not yet available (e.g an RS established in absence of a CS), the SRC SHOULD send a full metadata snapshot as soon as metadata becomes available.

If the SRC receives a snapshot request from the SRS, it MUST immediately send a full metadata snapshot.

The following is an example of a full metadata snapshot sent by the SRC in the initial INVITE request:

```
INVITE sip:recorder@example.com SIP/2.0
Via: SIP/2.0/TCP src.example.com;branch=z9hG4bKdf6b622b648d9
From: <sip:2000@example.com>;tag=35e195d2-947d-4585-946f-09839247
To: <sip:recorder@example.com>
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
CSeq: 101 INVITE
Max-Forwards: 70
Require: siprec
Accept: application/sdp, application/rs-metadata-request
Contact: <sip:2000@src.example.com>;+sip.src
Content-Type: multipart/mixed;boundary=foobar
Content-Length: [length]

--foobar
Content-Type: application/sdp

v=0
o=SRS 2890844526 2890844526 IN IP4 198.51.100.1
s=-
c=IN IP4 198.51.100.1
t=0 0
m=audio 12240 RTP/AVP 0 4 8
a=sendonly
a=label:1

--foobar
Content-Type: application/rs-metadata
Content-Disposition: recording-session

[metadata content]
```

Figure 11: Sample INVITE request for the recording session

9.2. Procedures at the SRS

The SRS receives metadata updates from the SRC in INVITE and UPDATE requests. Since the SRC can send partial updates based on the previous update, the SRS needs to keep track of the sequence of updates from the SRC.

In the case of an internal failure at the SRS, the SRS may fail to recognize a partial update from the SRC. The SRS may be able to recover from the internal failure by requesting for a full metadata snapshot from the SRC. Certain errors, such as syntax errors or semantic errors in the metadata information, are likely caused by an error on the SRC side, and it is likely the same error will occur

again even when a full metadata snapshot is requested. In order to avoid repeating the same error, the SRS can simply terminate the recording session when a syntax error or semantic error is detected in the metadata.

The SRS MAY explicitly request a full metadata snapshot by sending an UPDATE request. This request MUST contain a body with content disposition type "recording-session", and MUST NOT contain an SDP body. The SRS MUST NOT request a full metadata snapshot in an UPDATE response or in any other SIP transaction. The format of the content is "application/rs-metadata-request", and the body format is a simple text-based format. The following shows an example:

```
UPDATE sip:2000@src.exmaple.com SIP/2.0
Via: SIP/2.0/UDP srs.example.com;branch=z9hG4bKdf6b622b648d9
To: <sip:2000@exmaple.com>;tag=35e195d2-947d-4585-946f-098392474
From: <sip:recorder@example.com>;tag=1234567890
Call-ID: d253c800-b0d1ea39-4a7dd-3f0e20a
CSeq: 1 UPDATE
Max-Forwards: 70
Require: siprec
Contact: <sip:recorder@srs.example.com>;+sip.srs
Accept: application/sdp, application/rs-metadata
Content-Disposition: recording-session
Content-Type: application/rs-metadata-request
Content-Length: [length]

SRS internal error
```

Figure 12: Metadata Request

The SRS MAY include the reason why a metadata snapshot request is being made to the SRC in the reason line. This reason line is free form text, mainly designed for logging purposes on the SRC side. The processing of the content by the SRC is entirely optional since the content is for logging only, and the snapshot request itself is indicated by the use of the application/rs-metadata-request content type.

When the SRC receives a request for a metadata snapshot, it MUST immediately provide a full metadata snapshot in a separate INVITE or UPDATE transaction. Any subsequent partial updates will not be dependent on any metadata sent prior to this full metadata snapshot.

The metadata received by the SRS can contain ID elements used to cross reference one element to another. An element containing the definition of an ID, and an element containing a reference to that ID will often be received from the same SRC. It is also valid for those elements to be received from different SRCs, for example, when each endpoint in the same CS act as an SRC to record the call and a common ID refers to the same CS. The SRS MUST NOT consider this an error.

9.2.1. Formal Syntax

The formal syntax for the application/rs-metadata-request MIME is described below using the augmented Backus-Naur Form (BNF) as described in [RFC5234].

```
snapshot-request = srs-reason-line CRLF
srs-reason-line  = [TEXT-UTF8-TRIM]
; TEXT-UTF8-TRIM defined in RFC3261
```

10. Persistent Recording

Persistent recording is a specific use case outlined in REQ-005 or Use Case 4 in [RFC6341], where a recording session can be established in the absence of a communication session. The SRC continuously records media in a recording session to the SRS even in the absence of a CS for all user agents that are part of persistent recording. By allocating recorded streams and continuously sending recorded media to the SRS, the SRC does not have to prepare new recorded streams with new SDP offer when a new communication session is created and also does not impact the timing of the CS. The SRC only needs to update the metadata when new communication sessions are created.

When there is no communication sessions running on the devices with persistent recording, there is no recorded media to stream from the SRC to the SRS. In certain environments where Network Address Translator (NAT) is used, typically a minimum of flow activity is required to maintain the NAT binding for each port opened. Agents that support Interactive Connectivity Establishment (ICE) solves this problem. For non-ICE agents, in order not to lose the NAT bindings for the RTP/RTCP ports opened for the recorded streams, the SRC and SRS SHOULD follow the recommendations provided in [RFC6263] to maintain the NAT bindings.

11. IANA Considerations

11.1. Registration of Option Tags

This specification registers two option tags. The required information for this registration, as specified in [RFC3261], is as follows.

11.1.1. siprec Option Tag

Name: siprec

Description: This option tag is for identifying the SIP session for the purpose of recording session only. This is typically not used in a Supported header. When present in a Require header in a request, it indicates that the UAS MUST be either a SRC or SRS capable of handling the contexts of a recording session.

11.1.2. record-aware Option Tag

Name: record-aware

Description: This option tag is to indicate the ability for the user agent to receive recording indicators in media level or session level SDP. When present in a Supported header, it indicates that the UA can receive recording indicators in media level or session level SDP.

11.2. Registration of media feature tags

This document registers two new media feature tags in the SIP tree per the process defined in [RFC2506] and [RFC3840]

11.2.1. src feature tag

Media feature tag name: sip.src

ASN.1 Identifier: 25

Summary of the media feature indicated by this tag: This feature tag indicates that the user agent is a Session Recording Client for the purpose for Recording Session.

Values appropriate for use with this feature tag: boolean

The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is only useful for a Recording Session.

Examples of typical use: Routing the request to a Session Recording Server.

Security Considerations: Security considerations for this media feature tag are discussed in Section 11.1 of RFC 3840.

11.2.2. srs feature tag

Media feature tag name: sip.srs

ASN.1 Identifier: 26

Summary of the media feature indicated by this tag: This feature tag indicates that the user agent is a Session Recording Server for the purpose for Recording Session.

Values appropriate for use with this feature tag: boolean

The feature tag is intended primarily for use in the following applications, protocols, services, or negotiation mechanisms: This feature tag is only useful for a Recording Session.

Examples of typical use: Routing the request to a Session Recording Client.

Security Considerations: Security considerations for this media feature tag are discussed in Section 11.1 of RFC 3840.

11.3. New Content-Disposition Parameter Registrations

This document registers a new "disposition-type" value in Content-Disposition header: recording-session.

recording-session: The body describes either:

* metadata about the recording session

* reason for metadata snapshot request

as determined by the MIME value indicated in the Content-Type.

11.4. Media Type Registration

11.4.1. Registration of MIME Type application/rs-metadata-request

This document registers the application/rs-metadata-request MIME media type in order to describe a recording session metadata snapshot request. This media type is defined by the following information:

Media type name: application

Media subtype name: rs-metadata-request

Required parameters: none

Options parameters: none

11.5. SDP Attributes

This document registers the following new SDP attributes.

11.5.1. 'record' SDP Attribute

Contact names: Leon Portman leon.portman@gmail.com, Henry Lum henry.lum@genesyslab.com

Attribute name: record

Long form attribute name: Recording Indication

Type of attribute: session or media level

Subject to charset: no

This attribute provides the recording indication for the session or media stream.

Allowed attribute values: on, off, paused

11.5.2. 'recordpref' SDP Attribute

Contact names: Leon Portman leon.portman@nice.com, Henry Lum henry.lum@genesyslab.com

Attribute name: recordpref

Long form attribute name: Recording Preference

Type of attribute: session or media level

Subject to charset: no

This attribute provides the recording preference for the session or media stream.

Allowed attribute values: on, off, pause, nopreference

12. Security Considerations

The recording session is fundamentally a standard SIP dialog [RFC3261], therefore, the recording session can reuse any of the existing SIP security mechanisms available for securing the session signaling, the recorded media, and the metadata. The use cases and requirements document [RFC6341] outlines the general security considerations, and this document describes specific security recommendations.

The SRC and SRS MUST support SIP with TLS and MAY support SIPS with TLS as per [RFC5630]. The Recording Session SHOULD be at least as secure as the Communication Session, meaning using at least the same strength of cipher suite as the CS if the CS is secured. For example, if the CS uses SIPS for signaling and RTP/SAVP for media, then the RS should not downgrade the level of security in the RS to SIP or plain RTP since doing so will mean an automatic security downgrade for the CS. In deployments where the SRC and the SRS are in the same administrative domain and the same physical switch that prevents outside user access, some SRCs may choose to lower the level of security when establishing a recording session. While physically securing the SRC and SRS may prevent an outside attacker from accessing important call recordings, this still does not prevent an inside attacker from accessing the internal network to gain access to the call recordings.

12.1. Authentication and Authorization

At the transport level, the recording session uses TLS authentication to validate the authenticity of the SRC and SRS. The SRC and SRS MUST implement TLS mutual authentication for establishing the recording session. Whether the SRC/SRS chooses to use TLS mutual authentication is a deployment decision. In deployments where the SRC and the SRS are in the same administrative domain, the SRC and SRS may choose not to authenticate each other, or to have the SRC authenticate the SRS only, as there is an inherent trust relation between the SRC and the SRS when they are hosted in the same administrative domain. In deployments where the SRS can be hosted on a different administrative domain, it is important to perform mutual authentication to ensure the authenticity of both the SRC and the SRS before transmitting any recorded media. The risk of not authenticating the SRS is that the recording may be sent to a compromised SRS and that a sensitive call recording will be obtained by an attacker. On the other hand, the risk of not authenticating the SRC is that an SRS will accept calls from an unknown SRC and allow potential forgery of call recordings.

There may be scenarios in which the signaling between the SRC and SRS is not direct, e.g. a SIP proxy exists between the SRC and the SRS. In such scenarios, each hop is subject to the TLS mutual authentication constraint and transitive trust at each hop is utilized. Additionally, an SRC or SRS may use other existing SIP mechanisms available, including but not limited to, Digest Authentication [RFC3261], Asserted Identity [RFC3325], and Connected Identity [RFC4916].

The SRS may have its own set of recording policies to authorize recording requests from the SRC. The use of recording policies is outside the scope of the Session Recording Protocol.

12.2. RTP handling

In many scenarios it will be critical for the media transported between the SRC and the SRS to be protected. Media encryption is an important element in the overall SIPREC solution; therefore the SRC and the SRS MUST support RTP/SAVP [RFC3711] and RTP/SAVPF [RFC5124]. RTP/SAVP and RTP/SAVPF provide media encryption, integrity protection, replay protection, and a limited form of source authentication. They do not contain or require a specific keying mechanism. At a minimum, the SRC and SRS MUST support the SDP Security Descriptions (SDES) key negotiation mechanism [RFC4568]. For cases in which DTLS-SRTP is used to encrypt a CS media stream, an SRC may use SRTP Encrypted Key Transport (EKT) [I-D.ietf-avt-srtp-ekt] in order to use SRTP-SDES in the RS without needing to re-encrypt the media.

When RTP/SAVP or RTP/SAVPF is used, an SRC can choose to use the same or different keys in the RS than the ones used in the CS. Some SRCs are designed to simply replicate RTP packets from a CS media stream to the SRS, in which case the SRC will use the same key in the RS as used in the CS. In this case, the SRC MUST secure the SDP containing the keying material in the RS with at least the same level of security as in the CS. The risk of lowering the level of security in the RS is that it will effectively become a downgrade attack on the CS since the same key is used for both CS and RS.

SRCs that decrypt an encrypted CS media stream and re-encrypt it when sending it to the SRS MUST use a different key for the RS media stream than what is used for the CS media stream, to ensure that it is not possible for someone who has the key for the CS media stream to access recorded data they are not authorized to access.

12.3. Metadata

Metadata contains sensitive information such as the address of record of the participants and other extension data placed by the SRC. It is essential to protect the content of the metadata in the RS. Since metadata is a content type transmitted in SIP signaling, metadata SHOULD be protected at the transport level by SIPS/TLS.

12.4. Storage and playback

While storage and playback of the call recording is beyond the scope of this document, it is worthwhile to mention here that it is also important for the recording storage and playback to provide a level of security that is comparable to the communication session. It would defeat the purpose of securing both the communication session and the recording session mentioned in the previous sections if the recording can be easily played back with a simple unsecured HTTP interface without any form of authentication or authorization.

13. Acknowledgements

We want to thank John Elwell, Paul Kyzivat, Partharsarathi R, Ram Mohan R, Hadriel Kaplan, Adam Roach, Miguel Garcia, Thomas Stach, Muthu Perumal, Dan Wing, and Magnus Westerlund for their valuable comments and inputs to this document.

14. References

14.1. Normative References

[I-D.ietf-siprec-metadata]
R, R., Ravindran, P., and P. Kyzivat, "Session Initiation Protocol (SIP) Recording Metadata", draft-ietf-siprec-metadata-16 (work in progress), August 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2506] Holtman, K., Mutz, A., and T. Hardie, "Media Feature Tag Registration Procedure", BCP 31, RFC 2506, March 1999.

[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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

14.2. Informative References

[I-D.ietf-avt-srtp-ekt] Wing, D., McGrew, D., and K. Fischer, "Encrypted Key Transport for Secure RTP", draft-ietf-avt-srtp-ekt-03 (work in progress), October 2011.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, July 2006.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[RFC4916] Elwell, J., "Connected Identity in the Session Initiation Protocol (SIP)", RFC 4916, June 2007.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.

[RFC5168] Levin, O., Even, R., and P. Hagendorf, "XML Schema for Media Control", RFC 5168, March 2008.

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, October 2009.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.

[RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, June 2011.

[RFC6341] Rehor, K., Portman, L., Hutton, A., and R. Jain, "Use Cases and Requirements for SIP-Based Media Recording (SIPREC)", RFC 6341, August 2011.

[RFC7022] Begen, A., Perkins, C., Wing, D., and E. Rescorla, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 7022, September 2013.

[RFC7245] Hutton, A., Portman, L., Jain, R., and K. Rehor, "An Architecture for Media Recording Using the Session Initiation Protocol", RFC 7245, May 2014.

Authors' Addresses

Leon Portman
NICE Systems
22 Zarhin Street
P.O. Box 690
Ra'anana 4310602
Israel

Email: leon.portman@gmail.com

Henry Lum (editor)
Genesys
1380 Rodick Road, Suite 201
Markham, Ontario L3R4G5
Canada

Email: henry.lum@genesyslab.com

Charles Eckel
Cisco
170 West Tasman Drive
San Jose, CA 95134
United States

Email: eckelcu@cisco.com

Alan Johnston
Avaya
St. Louis, MO 63124

Email: alan.b.johnston@gmail.com

Andrew Hutton
Unify
Brickhill Street
Milton Keynes MK15 0DJ
United Kingdom

Email: andrew.hutton@unify.com