SIPPING Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 27, 2008

D. Wing
Cisco
F. Audet
Nortel
S. Fries
Siemens AG
H. Tschofenig
Nokia Siemens Networks
A. Johnston
Avaya

February 24, 2008

Secure Media Recording and Transcoding with the Session Initiation Protocol
draft-wing-sipping-srtp-key-03

Status of this Memo

Abstract

Call recording is an important feature in enterprise telephony applications. Some industries such as financial traders have requirements to record all calls in which customers give trading orders. This poses a particular problem for Secure RTP systems as

many SRTP key exchange mechanisms do not disclose the SRTP session keys to intermediate SIP proxies. As a result, these key exchange mechanisms cannot be used in environments where call recording is needed.

This document specifies a secure mechanism for a cooperating endpoint to disclose its SRTP master keys to an authorized party to allow secure call recording.

Table of Contents

1. Introduction

Call recording is an important feature in enterprise telephony applications. Some industries such as financial traders have requirements to record all calls in which customers give trading orders. In others, calls are recorded, as the near ubiquitous announcement says, "for training and quality control purposes".

Note that the services and examples in this document are not wiretapping as defined in Raven [RFC2804]. Specifically, all recording done by enterprises is always announced to both parties. Also, in most circumstances, the intent of the recording is to protect both parties from later disagreements about what was said during the conversation or to remedy mistakes made.

First, four different recording modes are discussed. Then example call flows for how this can be accomplished using standard SIP primitives. Finally, the impact of encrypted media, SRTP, is discussed.

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] and indicate requirement levels for compliant mechanisms.

The following terminology is taken directly from SIP Event State

Publication Extension [RFC3903]:

Event Publication Agent (EPA): The User Agent Client (UAC) that issues PUBLISH requests to publish event state.

Event State Compositor (ESC): The User Agent Server (UAS) that processes PUBLISH requests, and is responsible for compositing event state into a complete, composite event state of a resource.

Publication: The act of an EPA sending a PUBLISH request to an ESC to publish event state.

3. Introduction to SRTP Call Recording

This document addresses two difficulties with End-to-end encryption of RTP (SRTP [RFC3711]): transcoding and media recording. When peering with other networks, different codecs are sometimes necessary (e.g., transcoding a surround-sound codec for transmission over a

highly-compressed bandwidth-constrained network). In some environments (e.g., stock brokerages and banks) regulations and business needs require recording calls with coworkers or with customers. In many environments, quality problems such as echo can only be diagnosed by listening to the call (analyzing SRTP headers is not sufficient).

With an RTP stream, transcoding is accomplished by modifying SDP to offer a different codec through a transcoding device [RFC4117], and call recording or monitoring can be accomplished with an Ethernet sniffer listening for SIP and its associated RTP, with a media relay, or with a Session Border Controller. However, when media is encrypted end-to-end [I-D.ietf-sip-media-security-requirements], these existing techniques fail because they are unable to decrypt the media packets.

When a media session is encrypted with SRTP, there are three techniques to decrypt the media for monitoring or call recording:

1. the endpoint establishes a separate media stream to the recording device, with a separate SRTP key, and sends the (mixed) media to the recording device. This techniques is often called 'active

recording'. The disadvantages of this technique include doubling bandwidth requirements in the network and additionally the processing power on the client side. Moreover, the loss of media recording facility doesn't cause loss of call (as is required in some environments). Depending on the application requirements it may be necessary to establish a reliable connection to the recording device to cope with possible packet loss on the unreliable link, typically used for media transport. Because the endpoint maintains its own key with the connected party, this technique is more secure: a malicious media recording device cannot inject media to the connected party on behalf of the endpoint.

2. the endpoint relays media through a device which forks a separate media stream to the recording device. This technique is often employed by Session Border Controllers. This relay does not, itself, have access to the SRTP key.

3. Network monitoring devices are used to listen to the SRTP traffic and correlate SRTP with SIP. This correlation requires cooperation of call signaling devices if the call signaling is encrypted (e.g., with TLS).

This document describes cases (2) and (3) where a cooperating endpoint publishes its SRTP master keys to an authorized party using the SIP Event State Publication Extension [RFC3903]. The mechanism

can be described as passive recording, as the client is not directly involved with the media recording. The client merely provides the key information to a recording device. The mechanism described in this paper allows secure disclosure of SRTP session keys to authorized parties so that an endpoints media stream can be transcoded or decrypted, as needed by that environment. Technique (1) stated above is not considered further in this document, as it does not require the disclosure of the key used for the communication between the two endpoints.

4. Recording Modes

There are four common modes of call recording which are described in the following sections.

4.1. Always On Recording

In the Always On recording mode, for an identified endpoint, phone number, user or agent, all calls both incoming and outgoing are recorded. For example, a toll free call to a helpline could utilize this mode to record the entire text of calls.

4.2. Recording On Demand

In the Recording On Demand recording mode, only certain calls are recorded. For example, in a call center application, personal or non-call center calls by an agent might not be recorded.

4.3. Required Recording

In the Required Recording mode, the requirement for recording is so strong that if call recording resources are unavailable, the call must not be setup or an existing call must be disconnected.

4.4. Pause and Resume Recording

In the Pause and Resume Recording Mode, only parts of a given call may be recorded. For example, when the call is placed on hold, recording may be paused and resumed when the call is resumed. Or, IVR interactions in which a user enters account numbers and pin numbers should not be recorded, as the DTMF tones convey private or secure information. Pausing can be unidirectional or bi-directional.

5. Recording Call Flows

This section will show how these four recording modes can be

implemented .

In SIP call recording, the two-way RTP or SRTP media session between two UAs is sent to a UA referred to as a Recording UA. While it is possible for recording to be done locally in a UA, this has no impact on the SIP call flows.

While it is also possible for the recording policy and decision

making to be included in an endpoint, it is more common to have a third party control recording and cause the RTP or SRTP to be sent to the Recording UA. In these call flows, this third party will be called the Controller.

If the Controller acts as a third party call controller (3PCC) [RFC3725], it is possible for the Controller to cause each UA to send an extra media stream to the Recorder. However, for this call flow to work:

1. Both UAs must support multiple media lines and streams sent to different addresses (e.g., Section 2.4 of SDP Examples [RFC4317]).

2. Both UAs must have twice the normal bandwidth available.

3. Both UAs must know to send the same media on both media streams.

While 1 and 2 are possible, 3 is the most difficult. Without additional information in the SDP, each media stream is considered a separate media stream.

Alternatively, the Controller could be a combination of a SIP Proxy and a media relay (e.g., a Session Border Controller). This media relay would copy media streams to a second location. The protocol and coordination between these two elements is outside the scope of this specification. In another model discussed in Section 5, the Controller could be a SIP Focus and a Media Server with some special logic. Finally, the Controller could be realized as a B2BUA.

Using this model, there are no SIP, SDP, or bandwidth requirements on either UA. The Controller then can cause the media received at the Media Relay to be copied to the Recorder. An example is shown in Figure 1, below where the Recorder records a call between Alice and Bob.

Alice Controller Bob Recorder

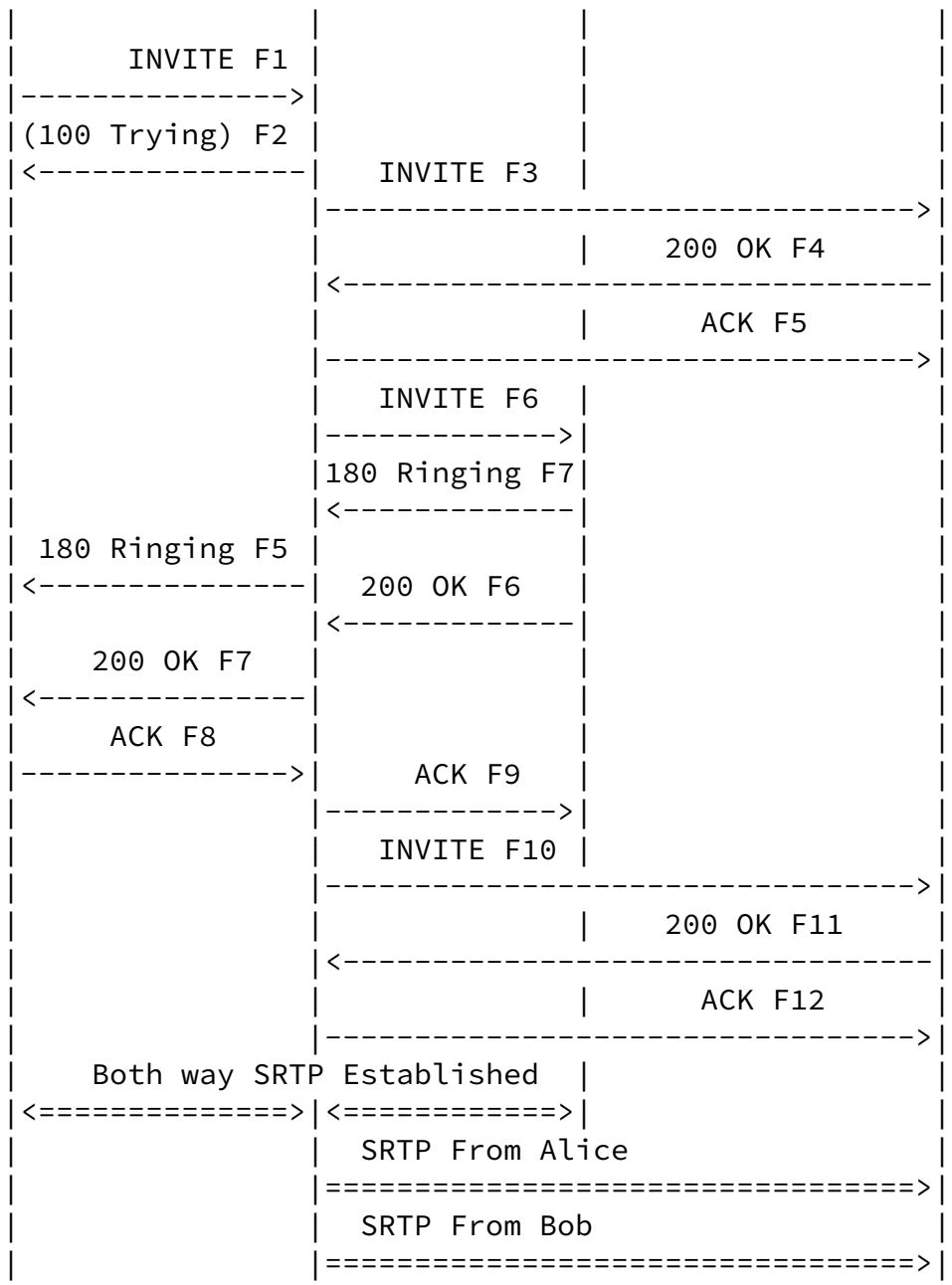


Figure 1: Controller Proxy or B2BUA

The following sections will discuss and extend this basic call flow for the four recording modes.

5.1. Always On Recording

The Always On recording mode for the user Bob can be implemented using the call flow of Figure 1 if every call made to Bob is handled in this way.

5.2. Recording On Demand

In the Recording On Demand recording mode, the call flow of Figure 1 is used selectively - only for the calls that need to be recorded. For the non-recorded flows, the Controller could act as a Proxy Server and make no changes to the signaling or media flows. By not inserting a Record-Route, the Controller could even drop out of the SIP dialog for calls where recording is not of interest.

5.3. Required Recording

Required recording could also be implemented using Figure 1, as the INVITE is sent first to the Recorder before being sent to Bob. As a result, if the INVITE is refused (i.e., the Recorder is unable to record the call), the INVITE will not be forwarded to Bob and the call refused. Also, if the Recorder disconnects during the call or is unable to provide recording resources (i.e., disks full, etc.), the BYE from the Recorder can be used to terminate the call to Bob. This is show in Figure 2, below.

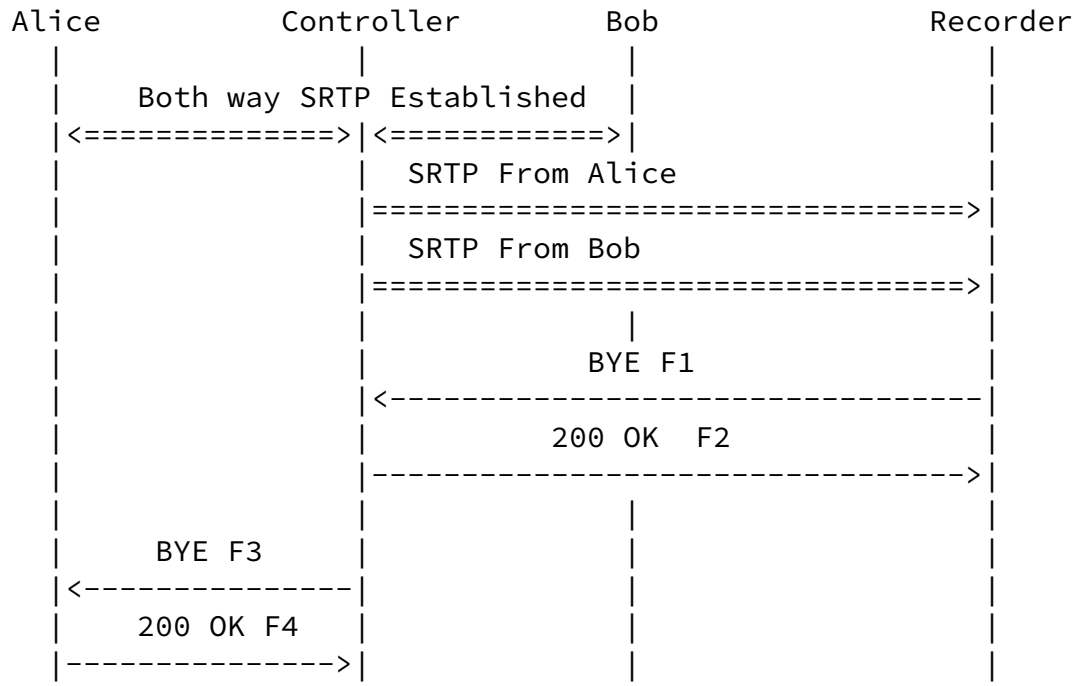


Figure 2: Required Recording Call Flow

5.4. Pause and Resume Recording Call Flow

The Pause and Resume recording mode can be initiated by the call flow of Figure 2. When the recording is to be paused, for example, when the caller Alice places the call on hold, the hold re-INVITE from

Alice causes the Controller to place the call to the Recorder on hold as well. No media is sent to the Recorder until a re-INVITE starts

the recording again, as shown in Figure 3, below.

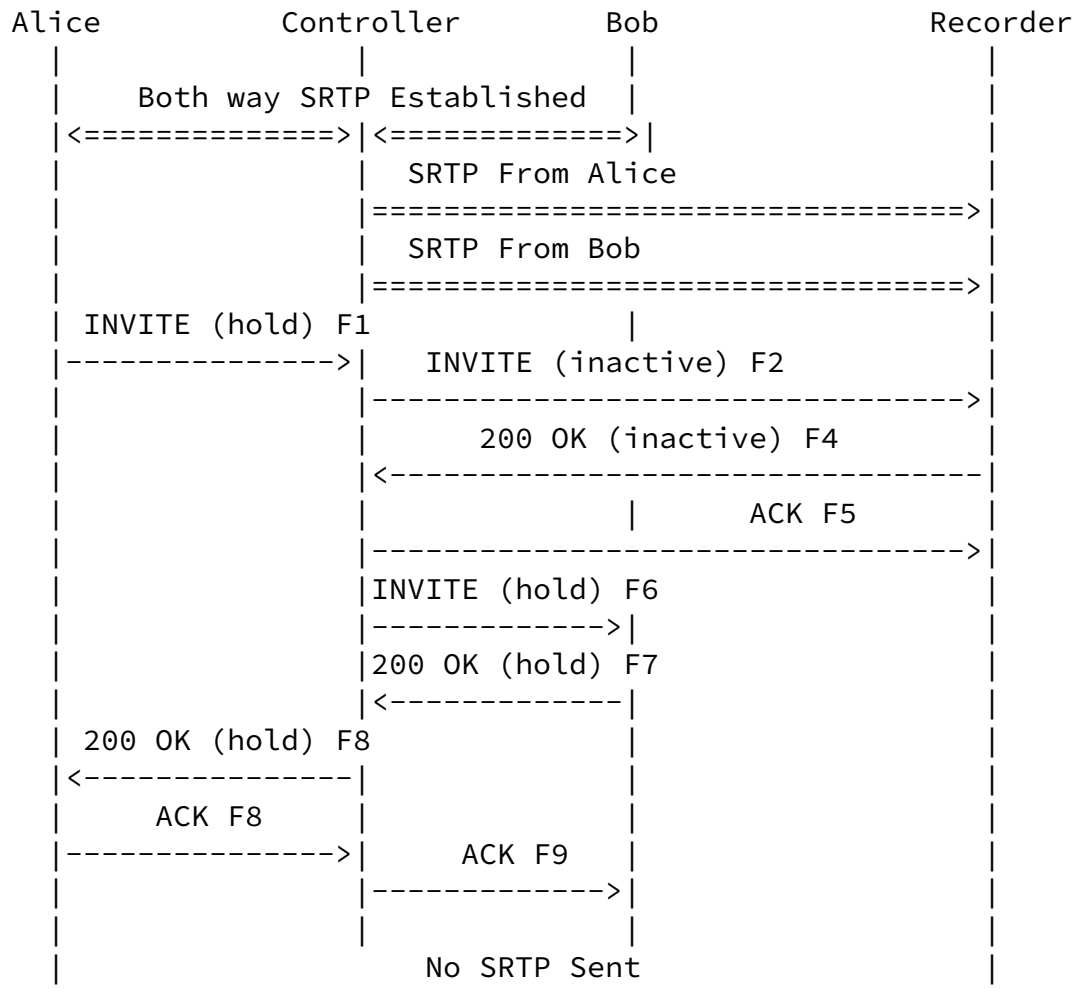


Figure 3: Pause and Resume Call Flow

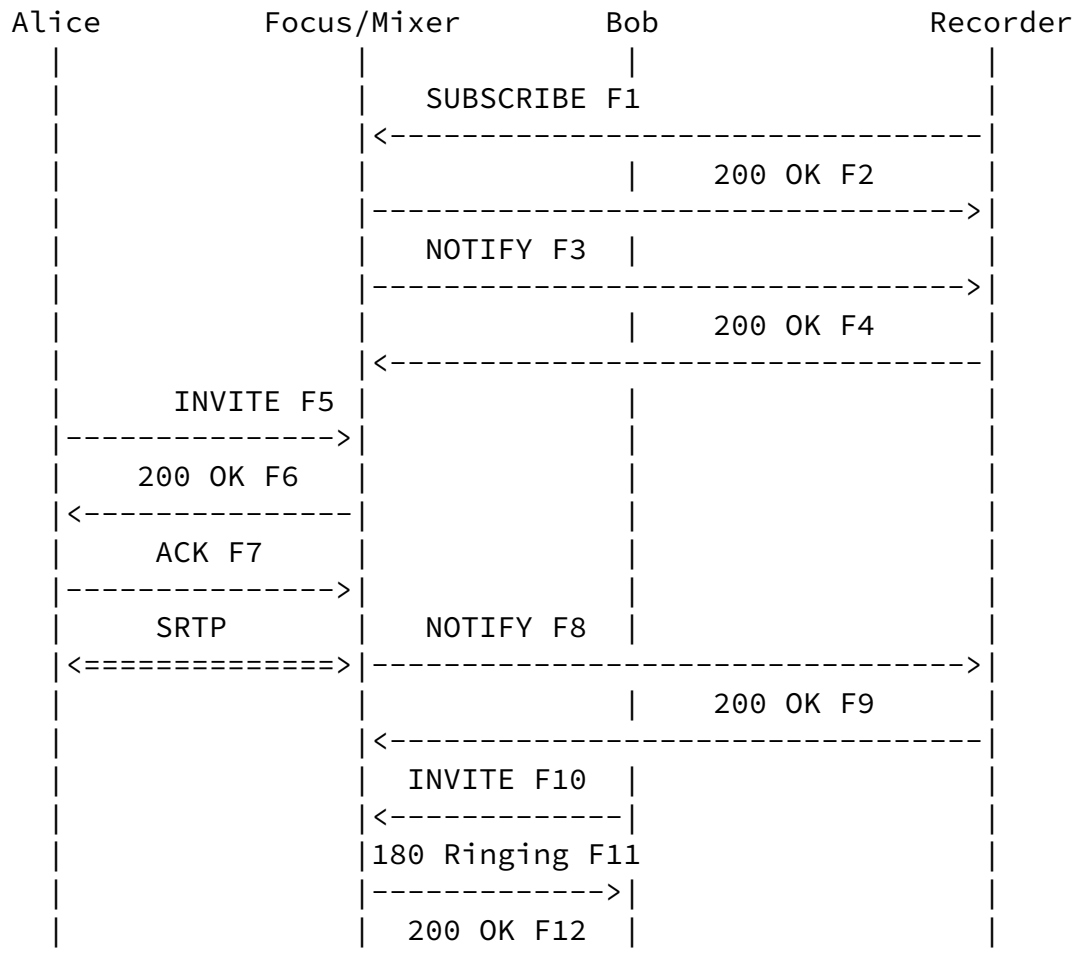
5.5. Conference Recording

A call flow for conference recording is shown in Figure 4, below. This call flow is similar to the previous ones except with a focus instead of the Controller. The recorder SUBSCRIBEs to the focus using the conference event package to learn of call recording events of interest to the Recorder.

With the subscription established by the SUBSCRIBE, the Recorder receives NOTIFYs whenever recording events of interest occur from the Controller. For example, the Recorder is informed when Alice joins the conference, but recording is not initiated. When notification that Bob has joined the conference is received in a NOTIFY, F7, is sent. In this example, the Recorder decides to record the call and sends a INVITE with Join to the Controller, F16. The dialog information used to construct the Join header field is obtained using the NOTIFY, F13. The Focus/Mixer then begins to stream the media to

the Recorder for the duration of the conference.

This model could be used for other recording modes. In this case, the event package would be a new event package specifically tailored to the recording application, containing all the information needed by a Recorder to make a decision on whether or not to record a call. The details of this event package may be defined in a future draft. Note that presently, CTI (Computer Telephone Integration) protocols are used for this purpose today.


Alice
Focus/Mixer
Bob
Recorder
SUBSCRIBE F1
200 OK F2
NOTIFY F3
200 OK F4
INVITE F5
200 OK F6
ACK F7
SRTP
NOTIFY F8
200 OK F9
INVITE F10
180 Ringing F11
200 OK F12

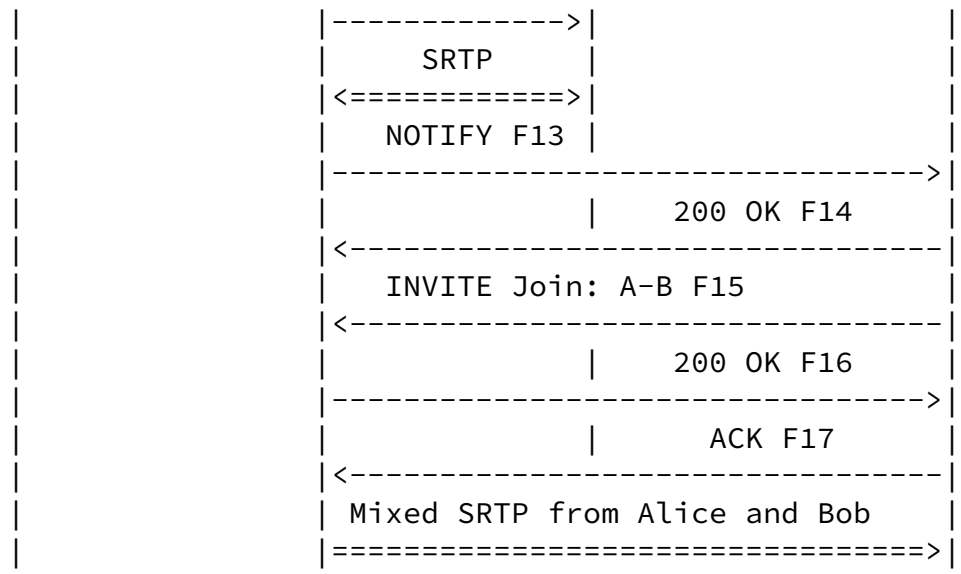


Figure 4: Conference Recording Call Flow

6. Transcoding

There are similarities between transcoding and call recording, especially technique 2 described in Section 3. An endpoint that

desires transcoding can provide its SRTP key to a transcoder and request its services.

[[This section is a placeholder, and will be expanded in a later version of this document.]]

7. Media Considerations

The following sections will discuss considerations relating to the media streams.

7.1. Offer/Answer Considerations

For the call flows in this document, it is assumed that a single bi-directional media stream is to be recorded. Normally, this would be negotiated using a single media line (m= line) in the SDP with a default direction attribute (a=sendrcv). The media stream sent from

the Controller to the Recorder could be done in two different ways, depending on the media handling in the Controller. In the simplest case, each direction of the media stream between Alice and Bob could be converted to a separate uni-directional media stream sent to the Controller. In the INVITE from the Controller to the Recorder, for a single recording session, there would be two media lines (m=) with each marked as send only (a=sendonly). This has the advantage that the Controller does not have to perform any processing on the RTP packets - they are simply forwarded without changing SSRC or sequence numbers. The Recording device will then mix the packets together or possibly record the two sides of the conversation separately, if desired.

In the other model, the Controller can function as an RTP mixer, in which case a single uni-directional media stream will be used with a single media line. The Controller will need to process the RTP packets by mixing them and including its own SSRC and sequence number in the resulting RTP packets. The Recorder will then not have to mix them and will not have the option of recording the two sides separately.

The approach of using two separate media lines is the recommended one as it allows for simple RTP packet processing at the Controller and also provides recording flexibility at the Recorder. However, a Recorder should also be able to handle the case where the Controller performs the mixing as well.

7.2. Operation

For transcoding, RTP packets must be sent from and received by a device which performs the transcoding. When the media is encrypted, this device must be capable of decrypting the media, performing the transcoding function, and re-encrypting the media.

> ISSUE-1: should we consider providing some or all of the SIP headers, as well? Some recording functions will need to know the identity of the remote party. This information could be gleaned from the SIP proxies, though, and starts to fall outside the

intended scope of this document.

ISSUE-2: The authors have been considering use of MIKEY [RFC3830], but MIKEY may not be used off the shelf. Certain changes to the state machine may have to be made (MIKEY [RFC3830] describes the TGK transport rather than SRTP master key transport).

7.2.1. Learning Name and Certificate of ESC

The endpoint will be configured with the AOR of its ESC (e.g., "transcoder@example.com"). If S/MIME is used to send the SRTP master key to the ESC, the endpoint is additionally configured with the certificate of its ESC.

The name and public key of the ESC is configured into the endpoint. It is vital that the public key of the ESC is not changed by an unauthorized user. Changes to change that public key will cause SRTP key disclosure to be encrypted with that key. It is RECOMMENDED that endpoints restrict changing the public key of the disclosure device using protections similar to changes to the endpoint's SIP username and SIP password.

7.2.2. Authorization of ESC

Depending on the application, authorization of the key disclosure and distribution to the ESC may be necessary besides the pure transport security of the key distribution itself. This may be the case when the configuration framework [I-D.ietf-sipping-config-framework] is not applied and thus the information about the ESC is not known to the client.

This can be done by providing a SAML extension [I-D.ietf-sip-saml] in the header of the SUBSCRIBE message. The SAML assertion shall at least contain the information about the ESC, call related information to associate the call with the assertion (editors note: we may also define wildcards here to allow for recordings of all phone calls for

a day, independent of the call) and a reference to the certificate for the ESC. The latter information is needed to transport the SRTP Session Key to the ESC in a protected manner, as described in the section below.

The signature of the SAML assertion should be produced using the private key of the domain certificate. This certificate MUST have a SubjAltName which matches the domain of user agent's SIP proxy (that is, if the SIP proxy is sip.example.com, the SubjAltName of the domain certificate signing this SAML assertion MUST also be example.com). Here, the main focus is placed on communication of clients with the ESC, which belongs to the client's home domain.

7.2.3. Sending SRTP Session Keys to ESC

SDP is used to describe the media session to the ESC. However, the existing Security Descriptions [RFC4568] only describes the master key and parameters of the SRTP packets being sent -- it does not describe the master key (and parameters) of the SRTP being received, or the SSRC being transmitted. For transcoding and media recording, both the sending key and receiving key are needed and in some cases the SSRC is needed.

Thus, we hereby extend the existing crypto attribute to indicate the SSRC. We also create a new SDP attribute, "rcrypto", which is identical to the existing "crypto" attribute, except that it describes the receiving keys and their SSRCs. For example:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
  inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
  SSRC=1899
a=rcrypto:1 AES_CM_128_HMAC_SHA1_80
  inline:AmO4q1OVAHNiYRj6HmS3JFWNCFqSpTqHWKKIN1Mw|2^20|1:32
  SSRC=3289
a=rcrypto:1 AES_CM_128_HMAC_SHA1_80
  inline:Hw3JFWNCFqSpTqNiYRj6HmSWKMHAmO4q1KIN1OVA|2^20|1:32
  SSRC=4893
```

Figure 5: Example SDP

The full SDP, including the keying information, is then sent to the ESC. The keying information MUST be encrypted and integrity protected. Existing mechanisms such as S/MIME [RFC3261] and SIPS [I-D.ietf-sip-sips] or SIP over TLS (on all hops per administrative means) MAY be used to achieve this goal, or other mechanisms may be defined.

[[ISSUE-3: if a endpoint is receiving multiple incoming streams from multiple endpoints, it will have negotiated different keys with each of them, and all of that traffic is coming to the same transport address on the endpoint. Thus, we need a way to describe the different keys we're using to/from different transport addresses. One solution is to indicate the remote transport address. Indicating the remote SSRC is insufficient for this task, as several SRTP keying mechanisms do not include SSRC in their signaling (DTLS-SRTP, ZRTP, Security Descriptions).

For example, if there were two remote peers with different keys, we could signal it like this:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
  inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
  192.0.2.1:5678 SSRC=1899 SSRC=3892
a=rcrypto:1 AES_CM_128_HMAC_SHA1_80
  inline:AmO4q1OVAHNiYRj6HmS3JFWNCFqSpTqHWKKIN1Mw|2^20|1:32
  192.0.2.1:5678 SSRC=3289 SSRC=2813
a=crypto:1 AES_CM_128_HMAC_SHA1_80
  inline:GdUJShpX1ZLEw6UzF3WSJjNzB4d1BINUAv+PSdFc|2^20|1:32
  192.0.2.222:2893
a=rcrypto:1 AES_CM_128_HMAC_SHA1_80
  inline:6UzF3IN1ZLEwAv+PSdFcWUGdUJShpXSJjNzB4d1B|2^20|1:32
  192.0.2.222:2893
```

Figure 6: Strawman solution

]]

7.2.4. Scenarios and Call Flows

The following scenarios and call flows depict the assumptions for the provision of media key disclosure. Figure 7 shows the general setup within the home domain of the client. Note that the authors assume that the client only discloses media keys only to an entity in the client's home network rather than to an arbitrary entity in the visited network.

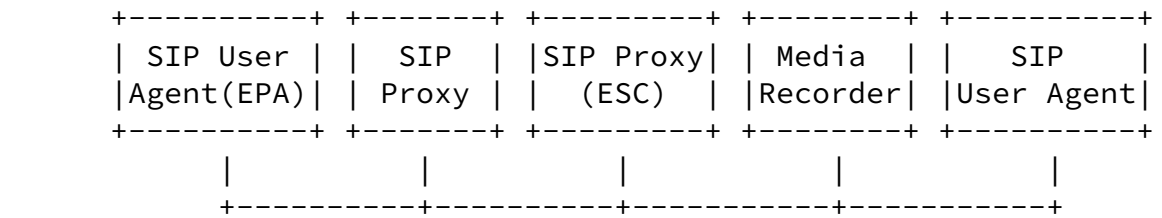


Figure 7: Network Topology

Based on this setup there are different options to realize the key disclosure, depending on the environment. In the following two approaches are distinguished.

Publishing media keys to the ESC

This requires that the configuration management provides the ESC configuration data (e.g., certificate, policy) in a secure way to the client. As stated above, this configuration is outside the scope of this document, but an example can be found in [I-D.ietf-sipping-config-framework]. The key disclosure in this approach uses the PUBLISH method to disclose the key to the ESC according to a given policy.

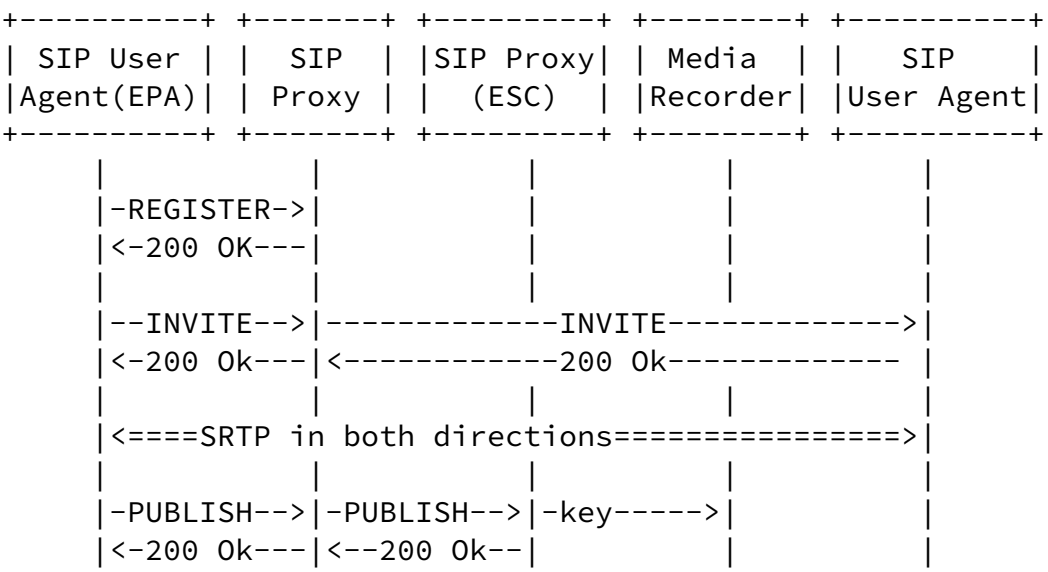


Figure 8: Message Flow showing Publishing of Media Keys to ESC

Note that the protocol between the ESC and the recorder is out of scope of this document.

Using SAML assertions for ESC contact

In this approach authorization is provided via a SAML assertion,

see [I-D.ietf-sip-saml], indicating which ESC is allowed to perform call recording of a single or a set of calls, depending on the content of the assertion. Here a SAML assertion is provided as part of the SUBSCRIBE message, send from the ESC to the client. The assertion needs to provide at least the call relation, or a time interval for which media recoding is going to be performed. The SAML assertion is signed with the private key associated with the domain certificate, which is in possession of the

authentication service. The call flow would look like following:

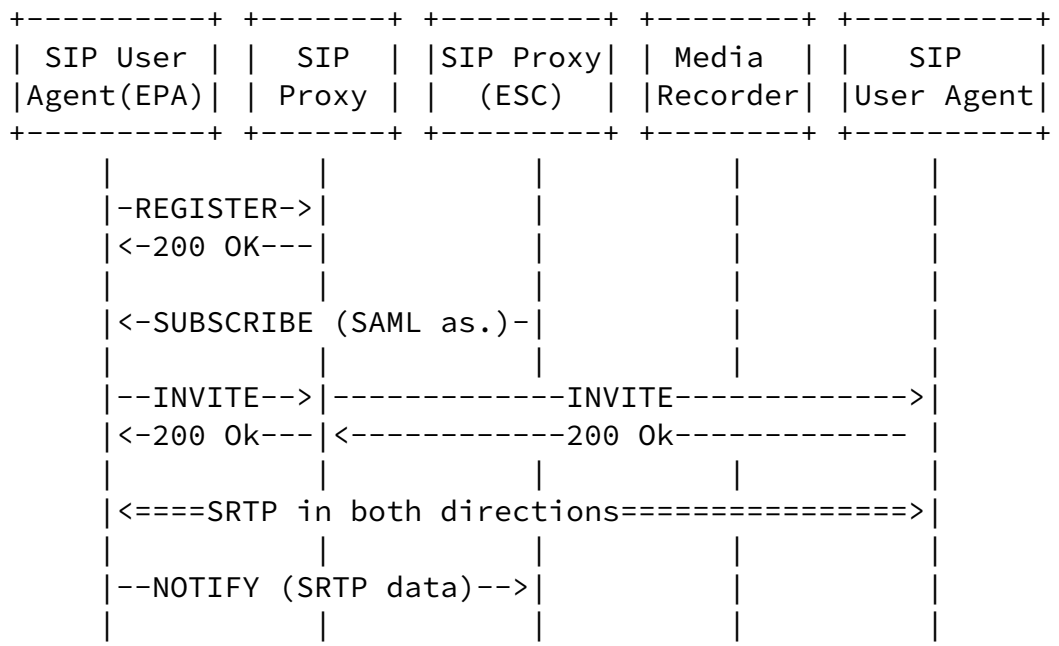


Figure 9: Message Flow Showing Publication using SAML

8. Grammar

[[Grammar will be provided in a subsequent version of this document.]]

9. Security Considerations

9.1. Incorrect ESC

Insertion of the incorrect public key of the SRTP ESC will result in disclosure of the SRTP session key to an unauthorized party. Thus, the UA's configuration MUST be protected to prevent such misconfiguration. To avoid changes to the configuration in the end device, the configuration access MUST be suitably protected.

9.2. Risks of Sharing SRTP Session Key

A party authorized to obtain the SRTP session key can listen to the media stream and could inject data into the media stream as if it were either party. The alternatives are worse: disclose the device's private key to the transcoder or media recording device, or abandon using secure SRTP key exchange in environments that require media transcoding or media recording. As we wish to promote the use

of secure SRTP key exchange mechanisms, disclosure of the SRTP session key appears the least of these evils.

9.3. Disclosure of Call Recording

Secure SRTP key exchange techniques which implement this specification SHOULD provide a "disclosure flag", similar to that first proposed in Appendix B of [I-D.zimmermann-avt-zrtp]. In this way, both endpoints can be made aware of such recording and provide appropriate alerting to their users (via an audible, visual, or other indicator).

9.4. Integrity and encryption of keying information

The mechanism describe in this specification relies on protecting and encrypting the keying information. There are well known mechanism to achieve that goal.

Using SIPS to convey the SRTP key exposes the SRTP master key to all SIP proxies between the Event Publication Agent (ESC, the SIP User Agent) and the Event State Compositor (ESC). S/MIME allows disclosing the SRTP master key to only the ESC.

10. IANA Considerations

New SSRC extension of the "crypto" attribute, and the new "rcrypto" attribute will be registered here.

11. Examples

This is an example showing a SIPS AOR for the ESC. This relies on the SIP network providing TLS encryption of the SRTP master keys to the ESC.

```
PUBLISH sips:recorder@example.com SIP/2.0
Via: SIP/2.0/TLS pua.example.com;branch=z9hG4bK652hsge
To: <sips:recorder@example.com>
From: <sips:dan@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
Expires: 3600
Event: srtp
Content-Type: application/sdp
Content-Length: ...

v=0
```

```
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
c=IN IP4 192.0.2.101
t=0 0
m=audio 49172 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_80
  inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
a=rcrypto:1 AES_CM_128_HMAC_SHA1_80
  inline:AmO4q1OVAHNiYRj6HmS3JFWNCFqSpTqHWKI8K1Mw|2^20|1:32
a=rtpmap:0 PCMU/8000
```

Figure 10: Example with "SIPS:" AOR

This is an example showing an S/MIME-encrypted transmission to the recorder's AOR, recorder@example.com. The data enclosed in "*" is encrypted with recorder@example.com's public key.

```
PUBLISH sip:recorder@example.com SIP/2.0
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
To: <sip:recorder@example.com>
From: <sip:dan@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
Expires: 3600
```

```
Event: srtp
Content-Type: application/pkcs7-mime;smime-type=enveloped-data;
              name=smime.p7m
Content-Transfer-Encoding: binary
Content-ID: 1234@atlanta.example.com
Content-Disposition: attachment;filename=smime.p7m;
                     handling=required
Content-Length: ...

 ********************************************************************
 * (encryptedContentInfo)                                           *
 * Content-Type: application/sdp                                    *
 * Content-Length: ...                                              *
 *                                                                  *
 * v=0                                                              *
 * o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com*
 * s=-                                                              *
 * c=IN IP4 192.0.2.101                                             *
 * t=0 0                                                            *
 * m=audio 49172 RTP/SAVP 0                                         *
 * a=crypto:1 AES_CM_128_HMAC_SHA1_80                               *
 *   inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32      *
 * a=rcrypto:1 AES_CM_128_HMAC_SHA1_80                              *
 *   inline:AmO4q1OVAHNiYRj6HmS3JFWNCFqSpTqHWKI8K1Mw|2^20|1:32      *
 * a=rtpmap:0 PCMU/8000                                             *
 *                                                                  *
 ********************************************************************
```

Figure 11: Example with S/MIME-encrypted SDP

12. Acknowledgements

Thanks to Sheldon Davis and Val Matula for suggesting improvements to the document.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

[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.

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

[RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.

13.2. Informational References

[I-D.ietf-sip-media-security-requirements]
Wing, D., Fries, S., Tschofenig, H., and F. Audet, "Requirements and Analysis of Media Security Management Protocols", draft-ietf-sip-media-security-requirements-02 (work in progress), January 2008.

[I-D.ietf-sip-saml]
Tschofenig, H., Hodges, J., Peterson, J., Polk, J., and D. Sicker, "SIP SAML Profile and Binding", draft-ietf-sip-saml-03 (work in progress), November 2007.

[I-D.ietf-sip-sips]
Audet, F., "The use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", draft-ietf-sip-sips-08 (work in progress), February 2008.

[I-D.ietf-sipping-config-framework]
Channabasappa, S., "A Framework for Session Initiation Protocol User Agent Profile Delivery", draft-ietf-sipping-config-framework-15 (work in progress), February 2008.

[I-D.zimmermann-avt-zrtp]
Zimmermann, P., "ZRTP: Media Path Key Agreement for Secure RTP", draft-zimmermann-avt-zrtp-04 (work in progress), July 2007.

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

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4117] Camarillo, G., Burger, E., Schulzrinne, H., and A. van Wijk, "Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc)", RFC 4117, June 2005.

[RFC4317] Johnston, A. and R. Sparks, "Session Description Protocol (SDP) Offer/Answer Examples", RFC 4317, December 2005.

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

Appendix A. Outstanding Issues

Authors' to-do list:

- Separate B2BUA function from media relay function in the call flows and in the text.

Authors' Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: dwing@cisco.com

Francois Audet
Nortel
4655 Great America Parkway
Santa Clara, CA 95054
USA

Email: audet@nortel.com

Steffen Fries
Siemens AG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: steffen.fries@siemens.com

Hannes Tschofenig
Nokia Siemens Networks
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@nsn.com
URI: http://www.tschofenig.com

Alan Johnston
Avaya
St. Louis, MO
USA

Email: alan@sipstation.com

Full Copyright Statement

Intellectual Property

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

This document was produced using xml2rfc v1.33pre8 (of http://xml.resource.org/) from a source in RFC-2629 XML format.