Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 18, 2008

D. Wing
Cisco Systems
F. Audet
Nortel
S. Fries
Siemens AG
H. Tschofenig
Nokia Siemens Networks

November 15, 2007

Disclosing Secure RTP (SRTP) Session Keys with a SIP Event Package
draft-wing-sipping-srtp-key-02

Status of this Memo

Copyright Notice

Abstract

Many Secure RTP (SRTP) key exchange mechanisms do not disclose the SRTP session keys to intermediate SIP proxies. However, these key exchange mechanisms cannot be used in environments where transcoding,

monitoring, or call recording are needed. This document specifies a secure mechanism for a cooperating endpoint to disclose its SRTP master keys to an authorized party.

Table of Contents

1. Introduction

This document addresses 2 difficulties with End-to-end encryption of RTP (SRTP [RFC3711]): transcoding and media recording. When peering with other networks, different codecs are sometimes necessary (transcoding a surround-sound codec for transmission over a highly-compressed bandwidth-constrained network, for example). 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.wing-rtpsec-keying-eval], 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 referenced as active recording here. 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). A significant advantage of this technique, however, is that it's secure: a malicious media recording device cannot inject media to the connected party on behalf of the endpoint. 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.

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, and could also be employed by TURN servers.

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

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 into the media recording. It 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.

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].

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. 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 (RFC3830 describes the TGK transport rather than SRTP master key transport).

3.1. 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.

3.2. 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 config 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, according to [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.

3.3. 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 1: 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 2: Strawman solution

]]

3.4. Scenarios and Call Flows

The following scenarios and call flows depict the assumptions for the provision of media key disclosure. Figure 3 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.

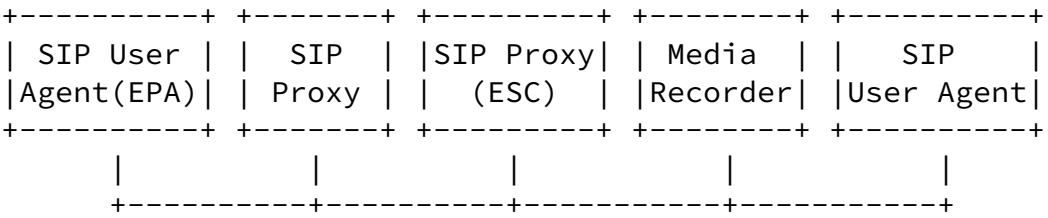


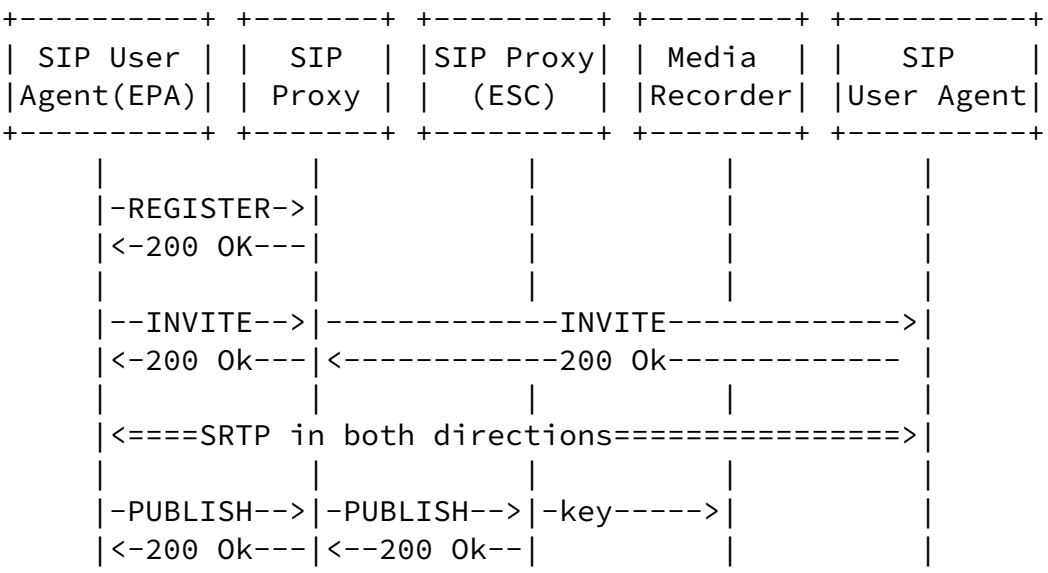
Figure 3: Network Topology

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

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

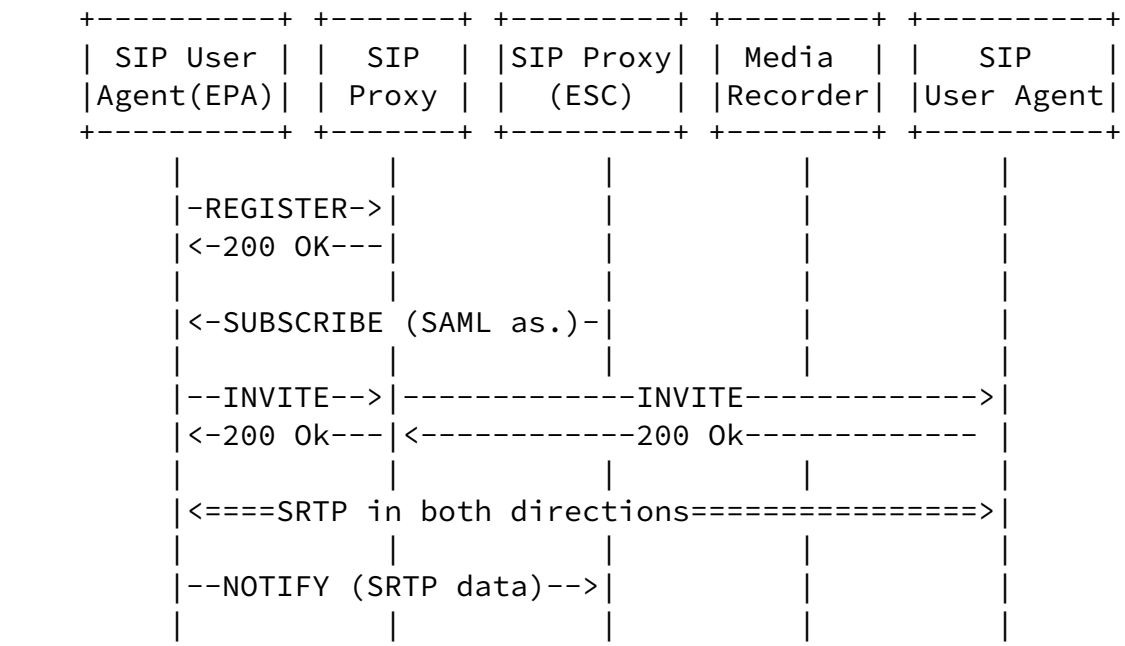
according to a given policy.



Note that the protocol between the ESC and the media 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 intervall 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:



4. Grammar

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

5. Security Considerations

5.1. 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.

5.2. 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.

5.3. Disclosure Flag

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].

5.4. Integrity and encryption of keying information

The mechanism describe in this specification relies on protecting and encrypting the keying infomation. 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.

6. IANA Considerations

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

7. 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 6: Example with "SIPS:" AOR

This is an example showing an S/MIME-encrypted transmission to the media 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 7: Example with S/MIME-encrypted SDP

8. References

8.1. Normative References

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

[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.

[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.

8.2. Informational References

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

[I-D.wing-rtpsec-keying-eval] Audet, F. and D. Wing, "Evaluation of SRTP Keying with SIP", draft-wing-rtpsec-keying-eval-02 (work in progress), February 2007.

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

[I-D.ietf-sipping-config-framework] Channabasappa, S., "A Framework for Session Initiation Protocol User Agent Profile Delivery", draft-ietf-sipping-config-framework-13 (work in progress), October 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-06 (work in progress), August 2007.

[I-D.ietf-sip-saml] Tschofenig, H., "SIP SAML Profile and Binding", draft-ietf-sip-saml-02 (work in progress), May 2007.

[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.

[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.

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

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).