Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 05, 2015

N. Ismail
Cisco
R. Barnes
Mozilla
D. Benham
N. Buckles
Cisco

July 04, 2014

Requirements for Secure RTP Media Switching
draft-ismail-avtcore-media-req-00

Abstract

This draft outlines the requirements for enabling media switches to form a multimedia multi-user conferences without needing to have the keys used to provide confidentiality and integrity for the media in the conference.

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 January 05, 2015.

Table of Contents

1. Introduction

Modern audio and video conferencing systems include RTP middleboxes that can often "switch" video and audio streams without mixing them. When receivers have homogenous coding capabilities and can receive multiple streams each, such media switchers avoid the need to decode and re-encode media for the purpose of compositing video or mixing audio. Instead they can forward encoded media as it was sent by the transmitter. In this case, a media switching device can behave more like a media switching RTP Translator [I-D.ietf-avtcore-rtp-topologies-update], which we will label an RTP Translator Forwarding Switch (RTFS).

Modern audio and video conferencing systems have also decomposed switching infrastructure into a) a controller that deals with the signaling and keeps track of who is in the conference and b) one or more media switching devices that receive, rewrite headers and transmit streams to receivers. In scalable systems, media switching devices may be deployed in many distributed locations to optimize bandwidth or latency and may be rented on demand from third-parties to meet peak loading needs. Therefore, there is a need to locate switching devices in data centers and/or be operated by third-parties not otherwise trusted with decryption or encryption of audio and video media.

This draft outlines the requirements for enabling media switching/ RTFS devices to perform only the functions they need to, including header rewites and authenticating transmitters and receivers, without

having to acquire or use the keys to provide confidentiality and integrity for the media in SRTP. This enables deployments where the privacy of the media can be assured even when a third-party service is used for switching media.

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. Media Switching/RTFS Architecture

In traditional conferencing systems, the conferencing media infrastructure fully decrypts, decodes and processes RTP media streams received from one or more transmitters prior to forwarding the newly encoded (transcoded, composited and/or mixed) and encrypted RTP media streams to the rest of receivers. Media Switching Mixers, which may need to composite or mix media, maintain independent and persistent SRTP sessions with each endpoint [I-D.ietf-avtcore-rtp-topologies-update]. More specifically, each endpoint establishes a point-to-point SRTP session with conferencing media infrastructure, which has its own persistent SSRCs, SRTP keys and SRTP contexts (reference the figure below) [RFC7201].

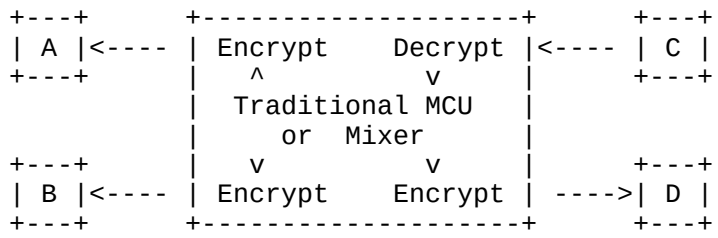


Figure 1: Traditional MCU or Mixer

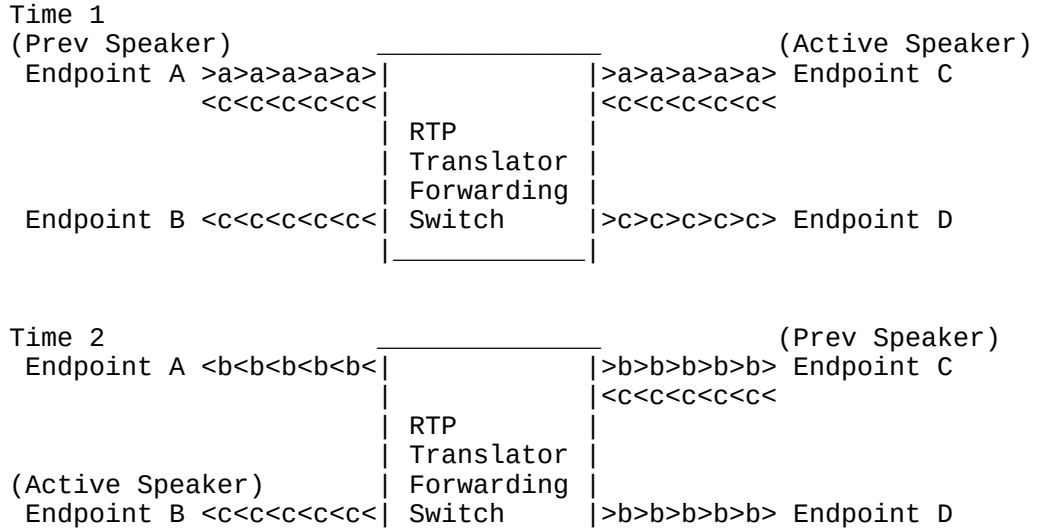
When receivers have homogenous coding capabilities and can receive multiple streams each, a media switcher can avoid processing media and (selectively) forward streams while manipulating only the necessary parts of the RTP headers prior to forwarding to receivers. The RTP payload part of streams from transmitters is forwarded without any processing or changes.

In this case, a media switching device can behave more like a scalable RTP Translator Forwarding Switch (RTFS), maintaining the SSRCs of the transmitting endpoints rather than generating their own persistent SSRCs towards every receiving endpoint (reference the figure below). Though this is not the only viable embodiment of a

media switching architecture, this is the most relevant for the requirements discussed in this document.

Figure 2: Scalable RTP Translator Forwarding Switch (RTFS)

These media switching/RTFS devices may selectively forward only certain transmitted stream(s) at any given time, such as the video and audio stream from the currently active speaker. In this case, endpoints receive different RTP video streams that are generated by different transmitters, each with its own SSRC, SRTP key and SRTP context. All these streams are rendered to the end user as a single video source representing the most active speaker. Moreover, endpoints do not receive the same RTP streams all the times. For example, in the figure below, endpoints A, B and D receive the video streams from endpoint C, the currently active speaker, which is actually receiving video from endpoint A, the previous active speaker. Later, when endpoint B becomes the active speaker, then endpoints A, C and D will start to receive video from B, which continues to receive video from endpoint C. In the final time slot, when Endpoint A becomes the active speaker, the process continues.



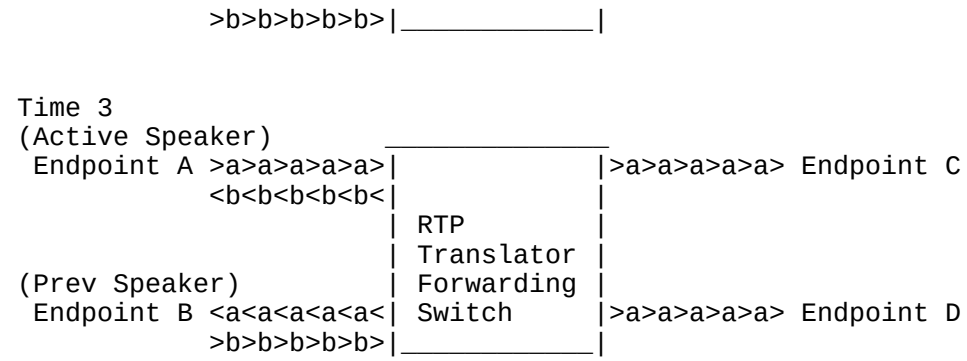


Figure 3: RTFS Media Flow for Active Speakers

Meeting the objective of scalability and simplicity in this media switching architecture starts with minimizing/eliminating the media processing performed by the media switching device, but can also to be extended to cryptography, where crypto processing and crypto state maintained by the media switching/RTFS devices are minimized. With the advent of cloud-based services, it is essential to enable deployments where the privacy of the media can be assured even when a third-party service is used for conference switching. Then enterprises can use cloud-based, third-party conferencing services while restricting such from accessing and manipulation of their media content. The ability to eliminate the need of media switching/RTFS devices to decrypt and re-encrypt packets is not merely a scalability and simplicity requirement, but is also a core security requirement in cloud-based conferencing services.

4. RTP header manipulation

A media switching/RTFS device might need to modify some of the RTP header fields to map between different values picked by different endpoints prior to switching. An example is the RTP payload type values which for SIP endpoints calling into the conference are picked by the endpoints. Different endpoints are likely to pick different values for the same media format. The media switching device is responsible for mapping between such different values. In the case of RTP payload types, the conference system might be able to send a SIP reinvite to renegotiate the RTP payload type value down to a shared value hence avoiding the remapping. This mechanism does not always work as endpoints can choose to use asymmetric payload types. Renegotiation also adds complexity and delays to the conferencing system. Other RTP header fields such as RTP extension headers can also be modified, deleted or added as they are negotiated separately with each participants.

On the other hand, two of the RTP fields must not be modified by media switches that do not have access to the media encryption keys. These two fields are the SSRC and the RTP sequence number. Both fields are used in the calculation of the SRTP cipher's IV, thus requiring a total re-encryption upon modification.

Below is the set of RTP header fields along with whether a media switching/RTFS device might modify them, unlikely to modify them or must not modify them.

- Version (V): This field is unlikely to be modified by the media switching device
- Padding marker (P): This field is unlikely to be modified by the media switching device
- Extension (X): The media switching device might modify this field when it needs to add RTP extension headers where none existed or if it needs to delete existing RTP extension headers
- Contributing sources count (CC): The media switching device is unlikely to modify this field
- Marker bit (M): This field is unlikely to be modified by the media switching device
- Payload Type (PT): The media switching device might modify this field to map between different RTP type values picked by different endpoints
- Sequence Number (SEQ): The media switching device must not modify this field
- Timestamp (TS): This field is unlikely to be modified by the media switching device
- Synchronization Source (SSRC): This field must not be modified by the media switching device
- Extension Header (ExtHDR): The media switching device is likely modify this field either to change its value or to delete it completely

5. Requirements

The following are the security solution requirements for media switching/RTFS device that enable media privacy to be maintained across participant endpoints.

1. Solution needs to maintain all current SRTP security properties.

2. Solution need to extend replay attacks protection to cover cross-participants replay prevention. Packets sent between the media switching device and participant A cannot be retransmitted to participant B undetected.

3. Keys used for encryption and authentication of RTP payloads and other information deemed unsuitable for accessibility by the media switching device must not be generated by or accessible to any of the media switching devices.

4. The media switching devices must be capable, if authorized, of changing any part of an RTP header except for the RTP sequence number and SSRC. This in turn mandates that the media switching devices must have access to the keys used for the authentication of RTP header fields other than SSRC and RTP sequence number when a proper authorization is in place.

5. The SRTP master keys must not be generated by the media switching devices

6. The media switching devices must not be involved in the distribution of the SRTP master keys to participants nor in the authentication of the participants identities for the purpose of key distribution

7. The media switching devices must be able to switch an already active SRTP stream to a new receiver while guaranteeing the timely synchronization between the SRTP context of the transmitter and its old and new receivers. Of special interest is the RoC part pf the SRTP context due to its dynamic nature. It is important to note that media switching devices can not change RTP sequence numbers as that would require packet re-encryption.

6. Example Scenario

The above requirements (especially 3 and 4) imply that there is a need for SRTP ciphersuites that allow a split key and split authentication model. Instead of the current single SRTP master key, this document requires two independent SRTP master keys. The first

is an end to end key that is used for the encryption of the RTP payload and other information requiring end-to-end encryption. The end to end key is also used for the authentication of the RTP payload, the RTP sequence number, RoC and SSRC as well as any other information requiring end-to-end authentication. The second key is hop-by-hop key used for the authentication of the RTP packet as well as any other information requiring hop by hop authentication (e.g. RTCP packet authentication). The hop-by-hop key can also be used for encryption of information that the switch is authorized to access and modify, such as encrypted RTCP packets.

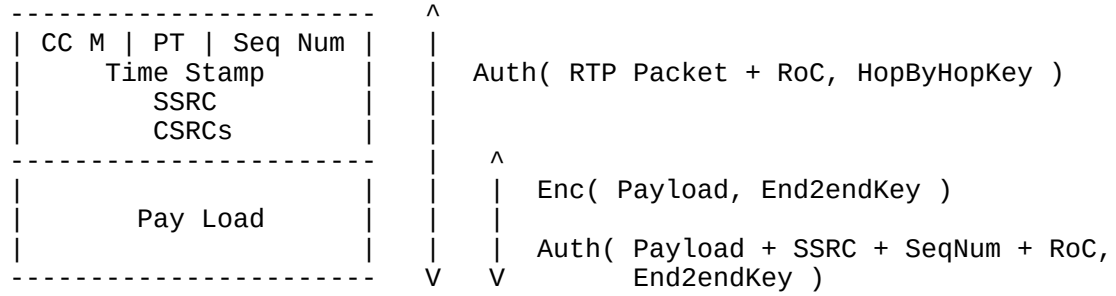
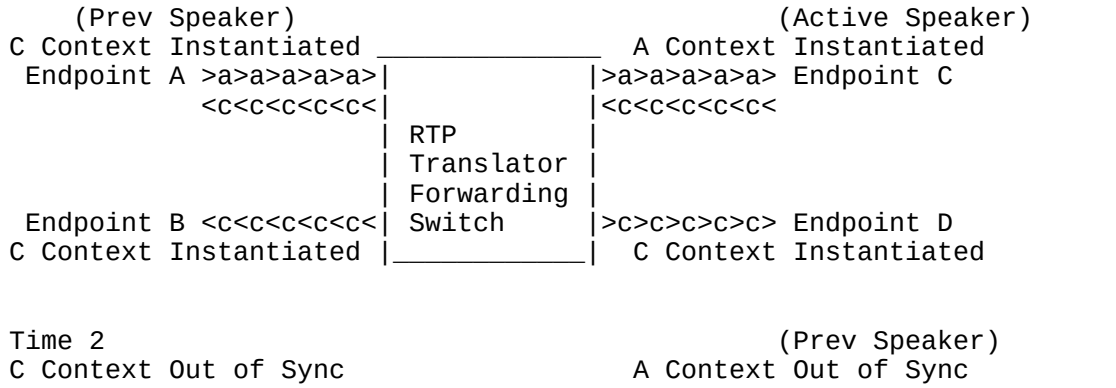


Figure 4: SRTP Split key-authentication model

The following figures illustrate how this split-context system could be used to accomplish the RTP forwarding objectives above. We do not show the control interactions that would be necessary to distribute the requisit keys among the participants.

TODO: Flesh out this example case further

Note that media from endpoints are flowing in direction of the arrows.



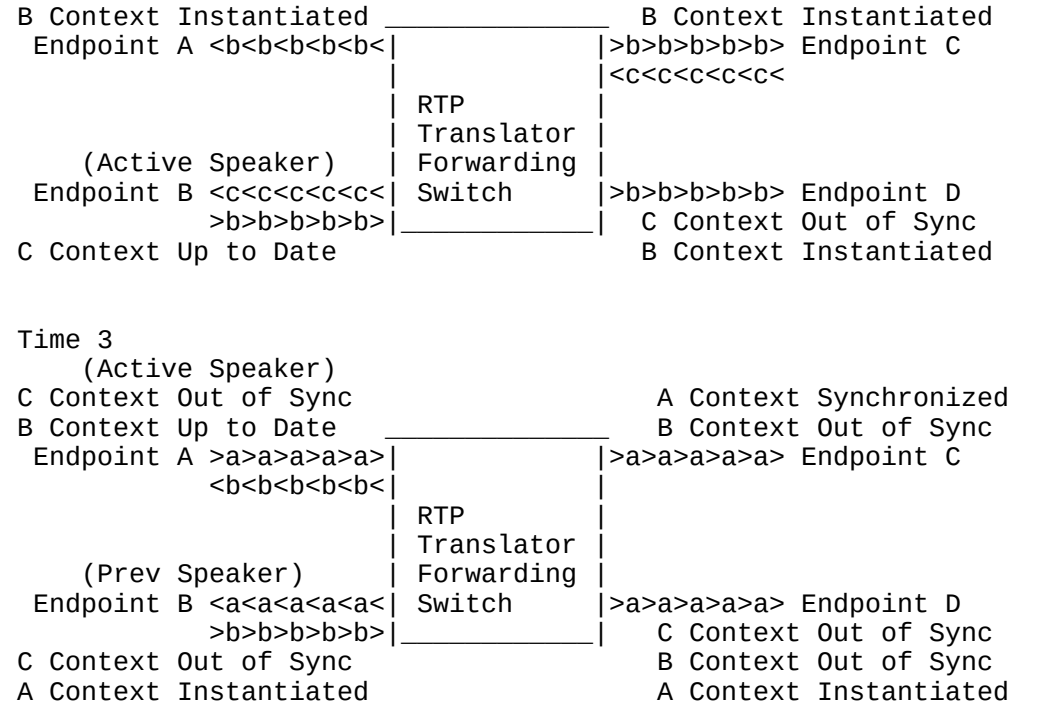


Figure 5: SRTP context synchronization

7. Security Considerations

This specification is all about new requirements for a system for securing RTP headers separately from the RTP body.

The requirements discussed above lead to a need for new SRTP cipher suites that split protection between hop-by-hop and end-to-end protections. This split may require new models for managing SRTP keys, e.g., extensions to DTLS-SRTP or EKT. We do not address requirements for key management in this document, since they would be accomplished at the control layer, rather than the RTP forwarding layer.

8. IANA Considerations

This document requires no actions from IANA.

9. Acknowledgements

The authors would like to thank Eric Rescorla and Cullen Jennings for their inputs. <GET YOUR NAME HERE - PLEASE SEND COMMENTS>.

10. References

10.1. Normative References

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

10.2. Informative References

[I-D.ietf-avtcore-rtp-topologies-update] Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-02 (work in progress), May 2014.

[I-D.ietf-rtcweb-security-arch] Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-09 (work in progress), February 2014.

[I-D.ietf-rtcweb-security] Rescorla, E., "Security Considerations for WebRTC", draft-ietf-rtcweb-security-06 (work in progress), January 2014.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, April 2014.

Authors' Addresses

Nermeen Ismail
Cisco
170 W Tasman Dr.
San Jose
US

Email: nermeen@cisco.com

Richard Barnes
Mozilla
331 E Evelyn Ave.
Mountain View
US

Email: rlb@ipv.sx

David Benham
Cisco
170 W Tasman Dr.
San Jose
US

Email: dbenham@cisco.com

Nathan Buckles
Cisco
170 W Tasman Dr.
San Jose
US

Email: nbuckles@cisco.com