Network Working Group
Internet Draft
Intended status: Standards Track
Expires: September 7, 2015

P. Jones (Ed.)
N. Ismail
D. Benham
Cisco Systems

March 7, 2015

A Solution Framework for Private Media in a Switched Conferencing draft-jones-avtcore-private-media-framework-01

Abstract

This document describes a solution framework for ensuring that media confidentiality and integrity are maintained end-to-end within the context of a switched conferencing environment where the switching conference server is not entrusted with the media encryption keys. The solution aims to build upon existing security mechanisms defined for the real-time transport protocol (RTP).

Status of this Memo

This Internet-Draft is submitted to IETF 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 September 7, 2015.

Table of Contents

1. Introduction

Switched conferencing is an increasingly popular model for multimedia conferences with multiple participants using a combination of audio, video, text, and other media types. With this model, real-time media flows from conference participants are not mixed, transcoded, transrated, recomposed, or otherwise manipulated on the conference server, as might be the case with a traditional multipoint control unit (MCU). Instead, media flows transmitted by conference participants are simply forwarded by the switching conference server to each of the other participants, selectively forwarding flows based on voice activity detection or other criteria. In some instances, the switching conference server may make limited modifications to RTP [RFC3550] headers, for example, but the actual media content (e.g., voice or video data) is unaltered.

An advantage of switched conferencing is that conference servers can be deployed on general-purpose computing hardware, as there is no need for the specialized hardware required to manipulate media flows

that one finds on a traditional hardware MCU. This, in turn, means that it is possible to deploy switching conference servers in virtualized environments, including private and public clouds.

However, deploying conference resource in a cloud environment may introduce a higher security risk. Whereas traditional conference servers were usually deployed in private networks that were protected from public access by firewalls, cloud-based conference resources might be viewed as less secure since they are not always physically controlled by those who use the hardware. Additionally, there are usually several ports open to the public in cloud deployments, most significantly being ports where the administrator can log in to make configuration changes, install software updates, and so on.

Recognizing the need to improve the way in which media confidentiality is ensured, requirements for private media were specified in [I.D-draft-jones-avtcore-private-media-reqts]. Attempting to meet those requirements, this document defines a solution framework wherein privacy is ensured by making it impossible for a switching conference server to gain access to keys needed to decrypt or authenticate the actual media content sent between conference participants. At the same time, the framework allows for the switching conference server to modify certain RTP headers; add, remove, encrypt, or decrypt RTP header extensions; and encrypt and decrypt RTCP packets. The framework also prevents replay attacks by authenticating each packet transmitted between a given participant and the switching conference server by using a key that is independent from the media encryption and authentication key(s) and is unique to the participating endpoint and the switching conference server.

A goal of this framework is to meet the referenced requirements and stated objectives by utilizing existing security procedures defined for RTP with minimal extensions.

2. Requirements Language

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 RFC 2119 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

3. Private Media Trust Model

The private media trust model is specified in [I.D-draft-jones-avtcore-private-media-reqts].

4. Solution Framework Overview

The purpose for this framework is to define a means through which media privacy can be ensured when communicating within a switched

conferencing environment. This framework specifies the re-use of

several technologies, including SRTP [RFC3711], EKT [I.D-draft-ietf-avtcore-srtp-ekt], and DTLS-SRTP [RFC5764].

4.1. Switching Conference Server Behavior

Before going into the specifics of how media privacy is ensured, first consider Figure 1 below depicting the behavior of a switching conferencing server forwarding media between participants.

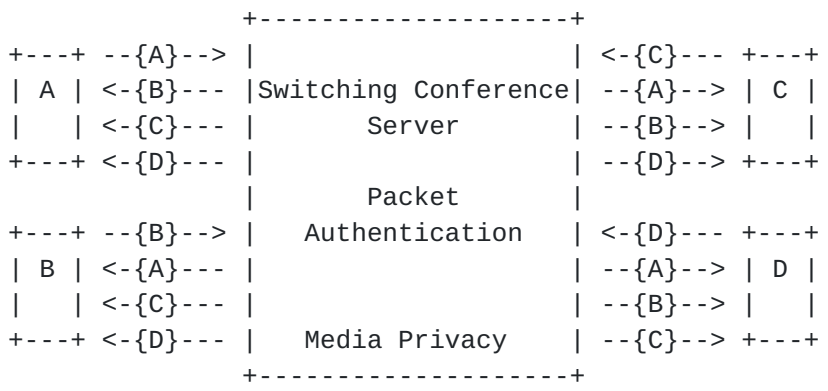


Figure 1 - Switching Conference Server

In the above figure, each of the participating endpoints sends media to the switching conference server where each flow is protected using the security mechanisms that will be discussed later. Each endpoint then receives media from each of the other participants in the conference. Importantly, the switching conference server is unable to decrypt the media content or modify media content without being detected by receiving endpoints.

The framework does not require, however, for each of the participant flows to be transmitted to every other endpoint in the conference. In many situations, the switching conference server will transmit only a subset of the media flows to each participant. This might be to restrict the bandwidth usage, provide a primary video flow and thumbnail flows to single-screen video endpoints, etc.

The following two diagrams and corresponding explanatory text are for illustrative purposes to describe one possible operational mode.

Figure 2 - Endpoint "C" is the Active Speaker

As depicted in Figure 2, each of the endpoints in the conference is receiving a single flow. In particular, all but one endpoints are receiving media flows from endpoint "C", the current active speaker. Endpoint "C" is receiving media from endpoint "A", the former active speaker.

Figure 3 - Endpoint "B" is the Active Speaker

When the active speaker transitions, so do the video flows. As depicted in Figure 3, the active speaker transitions from "C" to "B". Now, each of the endpoints receives a copy of the media flows from "B", while "B" receives the media flow from "C", the former active speaker.

How many flows and what type of flows a switching conference server transmits to a receiving endpoint are outside the scope of this document.

4.2. End-to-End Media Privacy

To ensure the confidentiality of RTP media packets, endpoints utilize EKT keys known to conference participants to encrypt the media content of the RTP packet (i.e., the RTP payload) using authenticated encryption. These keys may change from time-to-time for various reasons, such as when a new conference participant joins a conference or when a conference participant leaves a conference. When it is decided that a conference is to be re-keyed is outside the scope of

this document, but it is important that an unstrusted switching conference server is never given access to those keys.

This framework does not attempt to hide the fact that communication between parties takes place. Rather, it only addresses the end-to-end confidentiality and integrity of the actual media content.

4.3. Hop-by-Hop Operations

To ensure the integrity of transmitted media packets, this framework requires that every packet be authenticated. While media is both encrypted and authenticated end-to-end, RTP packets are also authenticated hop-by-hop. The authentication key used for hop-by-hop authentication is derived from the SRTP master key shared only on the respective hop. If conference servers are cascaded, then there will also be SRTP master keys and derived authentication keys shared between the cascaded servers. Importantly, each of these keys is distinct per hop and no two hops ever intentionally use the same SRTP master key.

It is expected that the conference servers may find it necessary to change certain parts of the RTP packet header, add or remove RTP header extensions, etc. By using hop-by-hop authentication, the switching media server is given liberty to change certain values present in the RTP header, such as the payload type value.

If there is a desire to encrypt RTP header extensions, an encryption key is derived from the hop-by-hop SRTP master key to encrypt header extensions as per [RFC6904]. This will give the switching conference server visibility into header extensions, such as the one used to determine audio level [RFC6464] of conference participants. Note that allowing RTP header extensions to be encrypted requires that all hops decrypt and re-encrypt any encrypted header extensions.

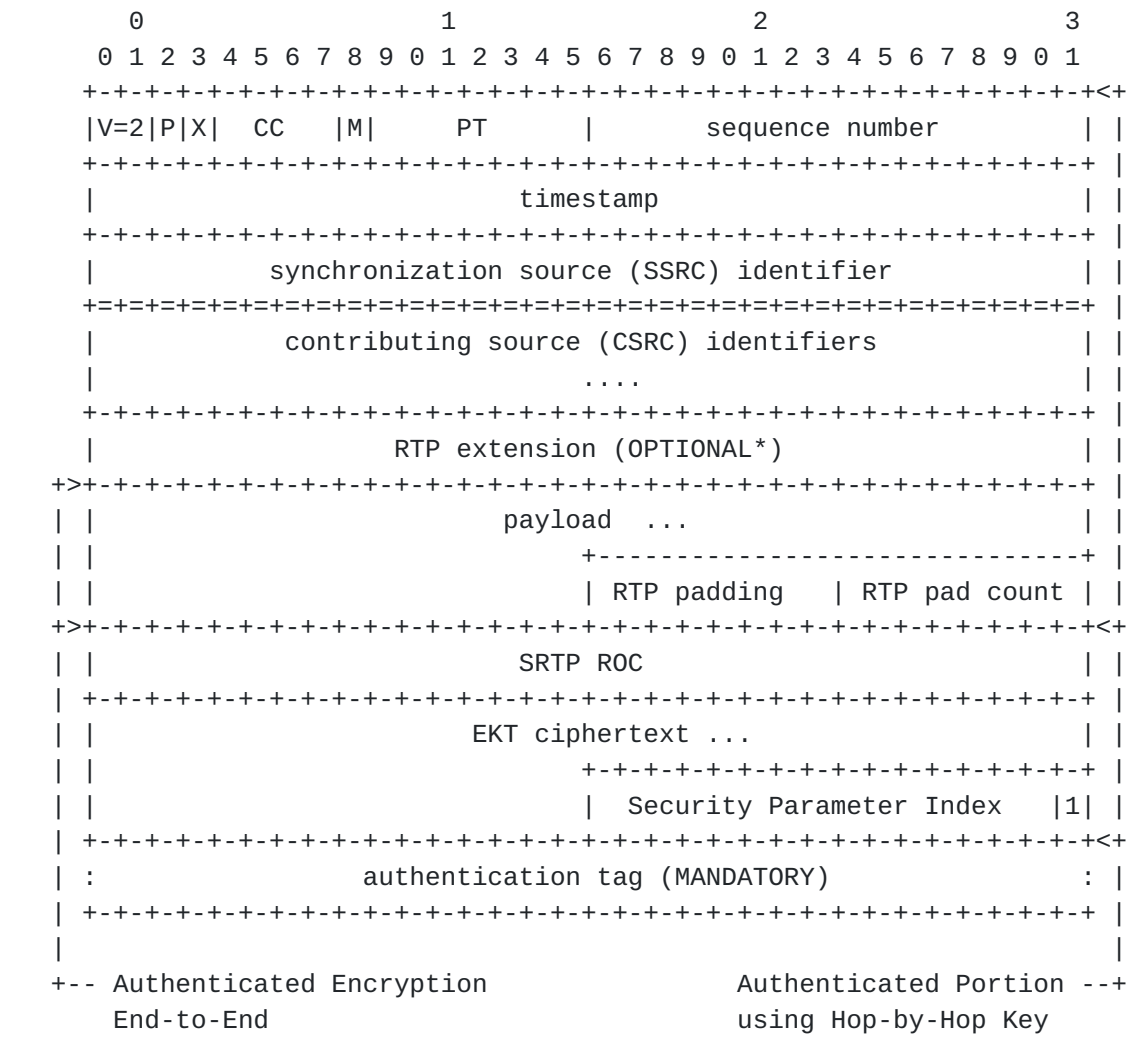
RTCP is optionally encrypted and mandatorily authenticated hop-by-hop using the encryption and authentication keys derived from the SRTP master key for the hop. This gives the switching conference server the flexibility of either forwarding RTCP packets unchanged, transmit compound RTCP packets, or to create RTCP packets to report statistics or for conference control.

One of the reasons for performing hop-by-hop authentication is to provide replay protection. If a media packet is replayed to the switching conference server, it will be detected. Likewise, the endpoint can detect replayed packets originally sent by the media server. Packets received by an endpoint that were originally sent to a different endpoint will fail to pass authentication checks.

5. Media Packet Format

Since the RTP packet payload is encrypted and authenticated end-to-end, extensions optionally encrypted hop-by-hop, and the entire RTP

packet is authenticated hop-by-hop, it may be useful to see the entire RTP packet similarly to what is shown in [RFC3711].



* Header extensions are optionally Encrypted Hop-by-Hop

Figure 4 - Private Media SRTP Packet

The rollover counter value is shown and transmitted as plaintext. This is necessary since a switching conference server may not transmit media from one "silent" participant to another participant in the conference for a long period of time. When media from that "silent" participant is later sent to that other participant, the receiving participant would not otherwise know the value of the rollover counter. Further, this value is needed so that the correct authentication tag can be generated hop-by-hop.

The EKT field shown in Figure 4 is the "Full EKT Field". The "Short EKT Field" may also be present in its place.

6. SRTP Cryptographic Context

For any given media source identified by its SSRC, there is a single SRTP cryptographic context as described in Section 3.2 of [RFC3711] used in this framework. However, this framework extends the

parameter set of the cryptographic context by adding an identifier for the end-to-end authenticated encryption algorithm. That parameter has associated with it an SRTP master key, and as outlined in Section 3.2.1, other associated values that relate to the master key (e.g., master salt and key length values). For AES-CCM, there will also be an associated "Tag_Size_Flag" value (see [I.D-draft-ietf-avtcore-srtp-aes-gcm]).

The existing parameters in the SRTP cryptographic context are used for hop-by-hop operations, including the optional encryption of RTP header extensions, authentication tag generation, etc.

7. Cryptographic Operations

7.1. Hop-by-Hop Authentication and Optional Encryption

For operations that occur hop-by-hop, the cryptographic transforms defined in SRTP [RFC3711] (or other standardized transforms) may be used in order optionally encrypt RTP header extensions, authenticate the RTP packet, optionally encrypt the RTCP packet, and to authenticate the RTCP packet.

The encryption and authentication of the RTP payload (media content) itself is not a hop-by-hop operation, as explained in the next section.

The procedures for optionally encrypting RTP header extensions is define in [RFC6904] and MUST be used when encrypting header extensions using the hop-by-hop SRTP master key to derive the k_he and k_hs values.

The procedures for authenticating the RTP packet, optionally encrypting the RTCP packet, and for authenticating the RTCP packet shall follow the procedures defined in [RFC3711] using the hop-by-hop SRTP master key and master salt to derive additional keys as specified in that specification.

7.2. End-to-End Media Payload Encryption and Authentication

This section covers the encryption and authentication of the RTP payload (i.e., media content) using the SRTP master key(s) derived from the EKT Key(s) by the endpoints communicating in a switched conferencing environment.

This framework requires that the end-to-end cryptographic transforms use authenticated encryption with associated data (AEAD) algorithms. Specifically, the transforms defined in [I.D-draft-ietf-avtcore-srtp-aes-gcm] are used as the default transforms in this framework.

The procedures followed to encrypt the payload are those described in [I.D-draft-ietf-avtcore-srtp-aes-gcm], except that the associated

data used with those algorithms specified in Section 9.2 is redefined as follows:

Associated Data: The version V (2 bits), padding flag P (1 bit), the sequence number (16 bits), timestamp (32 bits), and SSRC (32 bits).

The authentication tag for the end-to-end encrypted payload immediately follows the encrypted payload in the packet format.

Note that RTP header extensions are not encrypted as a part of the end-to-end function. Rather, they are encrypted as a hop-by-hop operation as explained in the previous section.

Only a part of the RTP packet is authenticated with the above definition of "Associated Data" since packets are authenticated hop-by-hop and there is a desire to allow switching conference servers to make changes to certain parts of the RTP header. For these reasons, there is a need for an authentication tag as defined in [RFC3711] to be placed at the end of the RTP packet. This authentication tag is provided via the hop-by-hop authentication operation as discussed in the previous section. Note that this is also a deviation from what [I.D-draft-ietf-avtcore-srtp-aes-gcm] recommends, but is necessary to allow the switching conference server to make changes to certain fields that would otherwise be protected.

8. Key Exchange

Within this framework, there are various keys each endpoint needs: those for end-to-end encryption/authentication and those for hop-by-hop authentication, optional encryption of RTP header encryptions, SRTCP authentication, and optional SRTCP encryption. Likewise, the switching conference server needs a hop-by-hop key when communicating with an endpoint or cascaded conference server. The challenge is in securely exchanging these keys to the appropriate entities.

To facilitate key exchange, we utilize DTLS-SRTP and procedures defined in EKT. We will elaborate on this further in the following sub-sections.

8.1. Session Signaling

The session signaling protocol is not significant to this specification, since the call processing functions are untrusted. Signaling might be via SIP [RFC3261] or a proprietary signaling between a browser and a server, as examples. What is important is that the signaling convey, in some manner, the fingerprint of the endpoint's certificate that will be used with DTLS-SRTP. For the sake of providing a more concrete discussion, we will assume SIP is

used and SDP [RFC4566] conveys the fingerprint information as per [RFC5763].

The endpoint ("User Agent" in SIP terminology) will send an INVITE message containing SDP for the media session along with fingerprints. This message or part thereof MUST be cryptographically signed so as to prevent unauthorized, undetectable modification of the fingerprint value, or the message MUST be sent to a trusted element over a secure connection.

For this example, we will assume the endpoint sends a message to a call processing function (e.g., a B2BUA) over a TLS connection. The B2BUA might sign the message using the procedures described in [RFC4474] for the benefit of forwarding the message to other entities, including the switching conference server. It's important to note, however, that this does not lend to the security of media, as the call processing function is not trusted.

The Key Management Function (KMF) needs to receive information about the call. This might be performed via an interface between the endpoint and the KMF, the call processing function and the KMF, or it might be via a signaling interface between the switching conference server and the KMF (see Error! Reference source not found.). Regardless, it is important that the endpoint's certificate fingerprint and a participant identifier (a random value created by the endpoint and provided to the KMF for each RTP session) are securely conveyed to the KMF. The client certificate and participant identifier will allow the KMF to associate the DTLS connection to the specific endpoint and RTP session for the conference. The endpoint to KMF information exchange is outside the scope of this document.

Ultimately, a call is established on the switching conference server and the endpoint receives address information to which it may establish one or more RTP sessions.

Call signaling going back to the endpoint might contain the certificate fingerprint of the KMF that will process DTLS-SRTP messages. Alternatively, the endpoint might already know the certificate fingerprint. Whatever mechanism is employed, it is extremely vital that the endpoint be able to fully trust the validity of the fingerprint information for the KMF.

[Editor's Note: How would an endpoint that is outside an enterprise domain (e.g., an associate at another company) be able to interact with the enterprise KMF? It might be necessary to have a trusted call processing entity that signs messages that the foreign endpoint can validate so that it knows that it can trust the certificate fingerprint of the KMF.]

8.2. Negotiating SRTP Protection Profiles and Key Exchange

8.2.1. Endpoint and KMF

There is a need for an SRTP master key and STRP master salt for hop-by-hop authentication and optional encryption known to the endpoint and the conference server. Additionally, there is a need to exchange an EKT master key and EKT master salt for the end-to-end encryption of the media content that is known to all participants in the conference, but not known to the switching conference servers.

To convey keys, the endpoint uses the procedures defined in [I.D-draft-ietf-avtcore-srtp-ekt] for DTLS-SRTP over the media ports for the RTP session. However, the switching conference server does not terminate the DTLS signaling. Rather, DTLS packets received by the conference server are forwarded to the KMF and vice versa. The figure below depicts this.

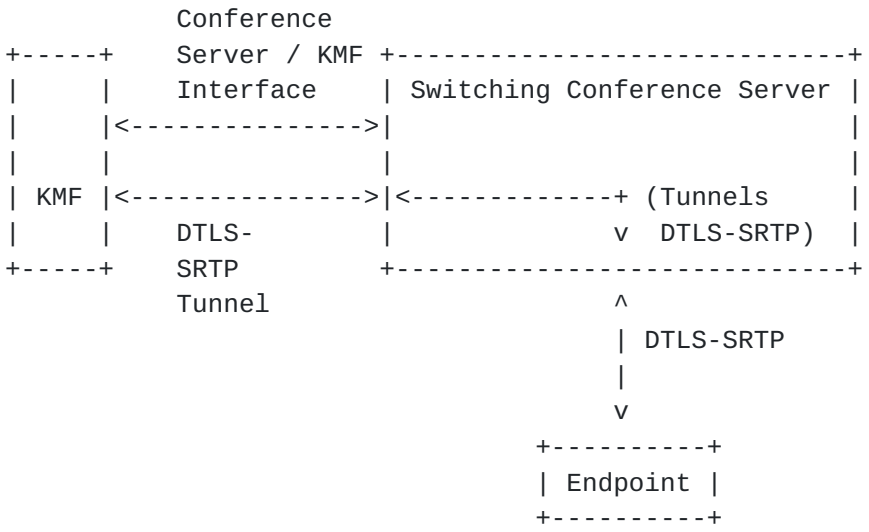


Figure 5 - DTLS-SRTP Tunneled to KMF

Through this tunneled DTLS-SRTP exchange, an EKT master key and EKT master salt are conveyed from the KMF to the endpoint, which the endpoint will use when deriving SRTP keys and encrypt and authenticate the media content in SRTP packets. The endpoint does not transmit media encryption keys to the KMF. The endpoint will follow the procedures specified in the EKT specification to generate an SRTP master key and convey this information to conference participants periodically (and anytime an I-Frame is explicitly requested) via the "Full EKT Field". [Editor's note: we are proposing changes to the EKT draft that will include the ROC separated from the EKT Ciphertext. Additionally, we need a mechanism to negotiate SRTP Protection Profiles for the end-to-end encryption/authentication.

This might be an extension to EKT, a new extension, or even an application-layer exchange over the DTLS connection to the KMF.]

This framework also calls for the extension of EKT in order to negotiate the SRTP Protection Profile used for end-to-end encryption and authentication. The RECOMMENDED default protection profile is AEAD_AES_128_GCM [I.D-draft-ietf-avtcore-srtp-aes-gcm].

The DTLS-SRTP procedures will result in the determination of an SRTP master key and master salt, along with an SRTP Protection Profile. This information is used for the hop-by-hop operations. [Editor's note: We could use DTLS-SRTP only to negotiate the SRTP Protection Profiles and then introduce a new extension to allow the KMF to send out the hop-by-hop key and salt to both the endpoint and conference server. Open to alternative suggestions from the workgroup.]

During the lifetime of the conference, conference participants may come and go. During those events, the KMF will send a new EKT message to clients providing a new EKT key to use from that point forward.

If a new participant does not support the same SRTP Protection Profile in use by the conference, the KMF must initiate a new DTLS-SRTP handshake with all conference participants to negotiate a new security profile and to re-key the conference. This may cause some disruption to conference. Therefore, it is recommended that we select a small number of protection profiles that must be implemented by all endpoints.

Summary of what we need to realize this framework:

- Endpoint must securely convey its certificate information to the KMF and negotiate a participant identifier (e.g., a UUID securely conveyed, but need not be encrypted) before a connection to the conference server is attempted.

- A means through EKT or another extension to negotiate the SRTP security profiles for end-to-end encryption/authentication

- A means through EKT or another extension of sending the participant identifier (the participant identifier could implicitly identify the conference)

- A change to EKT such that the ROC is transmitted in the clear, with integrity check performed by XORing the ROC with the IV used in AES Key Wrap

- A means of conveying per-hop SRTP master key and salt information to the switching conference server

To help in understanding better the sequence of messages, consider the following figure:

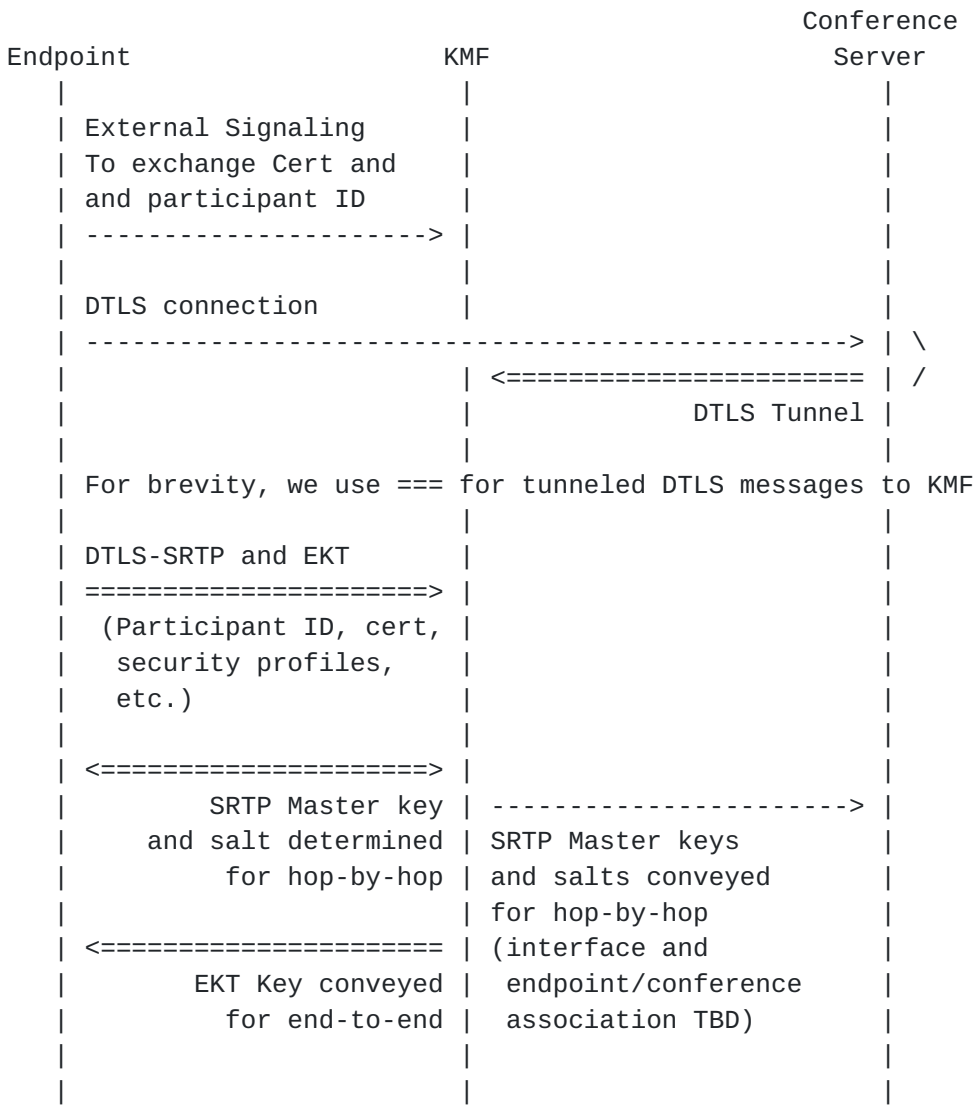


Figure 6 - Key Exchange Procedure

Following the key exchange, the endpoint will be able to encrypt media end-to-end and authenticate packets hop-by-hop. Likewise, the conference server will be able to authenticate the received packet at the hop, but will have no visibility into the encrypted media content.

8.2.2. Switching Conference Server and KMF

[Editor's Note: there must be an interface between the switching conference server and the KMF so that cipher suites and key information can be conveyed for each participant in each conference for hop-by-hop operations. This interface is out of scope for this document.]

9. Changing Media Forwarded and EKT Field

Endpoints transmit media to the switching conference server as they would in a traditional conference, except that media is encrypted and authenticated with different keys as outlined in this framework. Each media source within an RTP session has a distinct SSRC and endpoints work to address SSRC collisions when they occur. From the endpoint's perspective, what is particularly unique about the model described in this document is how the RTP payload (media content) is encrypted and authenticated end-to-end, while other security procedures are performed hop-by-hop.

To ensure a speedy decoder synchronization in receivers when transitioning from forwarding one active speaker's media to the next, a switching conference server will send a request for Full Intra-frame Request (FIR) [RFC5104] (also known as a "video fast update" in [H.323] systems) when a decision is made to switch active video flows. When the endpoint receives this request, it would transmit the video frame as requested and include with that initial packet the current "Full EKT Field" so that recipients will be able to decrypt the media flow. Additionally, a "Full EKT Field" should be transmitted about every 100ms to ensure that conference participants can decrypt the media transmitted.

It is not possible to request a "Full EKT Field" for audio flows. For this reason, it is RECOMMENDED that a "Full EKT Field" be included in audio packets about every 100ms to smooth the transition of the active speaker's audio forwarded by the server.

Endpoints SHOULD NOT include the "Full EKT Field" more frequently than specified herein, rather opting for the "Short EKT Field" when sending most packets to reduce the bandwidth consumed on the wire.

A switching conference server may forward a single audio and video flow to a receiver, or it may forward multiple flows. The number of media flows very much depends on the capabilities of the receiving device. How the number of media flows to forward is determined or negotiated is outside the scope of this document.

To aid in determining when to transition the active speaker's audio or video, endpoints MUST implement [RFC6464] in order to provide a hint to the switching media server as to which endpoint should be designated as the one of the active speaker(s).

10. IANA Considerations

There are no IANA considerations for this document.

11. Security Considerations

[TBD]

12. References

12.1. Normative References

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

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

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

[I.D-draft-ietf-avtcore-srtp-ekt] Mattson, J., McGrew, D., Wing, D., and F. Andreasen, "Encrypted Key Transport for Secure RTP", Work in Progress, October 2014.

[RFC6904] J. Lennox, "Encryption of Header Extensions in the Secure Real-time Transport Protocol (SRTP)", RFC 6904, December 2013.

[I.D-draft-ietf-avtcore-srtp-aes-gcm] McGrew, D. and K. Igoe, "AES-GCM and AES-CCM Authenticated Encryption in Secure RTP (SRTP)", Work in Progress, July 2014.

[RFC5763] Fischl, J. Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, May 2010.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[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.

[RFC6464] Lennox, J., Ivov, E., and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", RFC 6464, December 2011.

12.2. Informative References

[I.D-draft-jones-avtcore-private-media-reqts] Jones, P. et al., "Requirements for Private Media in a Switched Conferencing Environment", Work in Progress, March 2015.

[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.

[H.323] Recommendation ITU-T H.323, "Packet-based multimedia communications systems", December 2009.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

13. Acknowledgments

The authors would like to thank Christian Oien for invaluable input on this document.

Authors' Addresses

Paul E. Jones
Cisco Systems, Inc.
7025 Kit Creek Rd.
Research Triangle Park, NC 27709
USA

Phone: +1 919 476 2048
Email: paulej@packetizer.com

Nermeen Ismail
Cisco Systems, Inc.
170 W Tasman Dr.
San Jose
USA

Email: nermeen@cisco.com

David Benham
Cisco Systems, Inc.
170 W Tasman Dr.
San Jose
USA

Email: dbenham@cisco.com