Network Working Group
Internet Draft
Intended status: Informational
Expires: January 6, 2016

P. Jones (Ed.)
N. Ismail
D. Benham
N. Buckles
Cisco Systems
J. Mattsson
Ericsson
R. Barnes
Mozilla

July 6, 2015

Private Media Requirements in Privacy Enhanced RTP Conferencing
draft-jones-perc-private-media-reqts-00

Abstract

This document specifies the requirements for ensuring the privacy and integrity of real-time transport protocol (RTP) media flows between two or more endpoints communicating through one or more centrally located media distribution devices (MDDs).

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 January 6, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Users of multimedia communication products and services have privacy expectations that are largely satisfied with the use of SRTP [RFC3711] and related technologies when communicating point-to-point over the Internet. When two or more endpoints communicate through a traditional media server, it is necessary for those endpoints to share the SRTP master key and salt information with the traditional media server so that it can authenticate and decrypt received RTP and RTCP packets. The key material is needed so that a traditional media server can perform various operations on the media, such as mixing,

transcoding, and transrating. The traditional media server also needs the master key and salt in order to transmit media packets to other endpoints in the conference. The need for a traditional media server to have the master key represents a security risk.

Within a corporate or other isolated environment where all conferencing resources, including both call control and media processing functions, are tightly controlled, this security risk can be effectively managed. However, managing this risk is becoming increasing difficult as conferencing resources are deployed in networks that are not so strictly managed or controlled, including resources on virtualized servers deployed in third-party cloud environments.

There are also existing public voice and video conferencing service providers in which users must place full trust by sharing media encryption keys in order to use those services. This exposes corporations, for example, to a higher risk of being subjected to corporate espionage. While it is not the intent of this draft to suggest that any existing service provider would permit or condone any illicit use of its service, the fact is that security threats can come from either internal or external sources and remain undiscovered for long periods of time.

It is possible to ensure real-time transport protocol (RTP) media privacy in deployments using one or more centrally located media distribution devices (MDDs) with limited changes in the security mechanisms used today. This document discusses this possibility in more detail and presents a set of requirements that are neutral with respect to session signaling protocols.

This document is focused on ensuring the privacy of RTP media in centralized MDD models only. Other types of media are out of scope. Other, non-centralized media distribution models are also out of scope.

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. Terminology

Adversary - An unauthorized entity that may attempt to compromise the performance of a media distribution device through various means, including, but not limited to, the transmission of bogus media packets or attempt to gain access to the plaintext of the media.

Media content - The portion of the RTP (i.e., the encrypted RTP payload) or other packet containing the actual audio, video, or other multimedia information that is considered confidential and is subject to end-to-end encryption. This does not include, for example, RTP headers, RTP header extensions, or RTCP packets.

Switching media distribution device - A media distribution device that does not decrypt RTP media flows or perform processing on the media payload, but instead simply forwards the received media from a sender to the other endpoints in a multimedia conference. A switching media distribution device may modify some portion of the RTP header and may often consume and create RTCP messages for efficient media handling.

4. Background

Traditional media servers used for multimedia conferencing would mix, transcode, transrate, and/or recompose media flows from one or more conference participants' endpoints, sending out a different audio and video flow to each endpoint. For audio, this might entail mixing some number of input flows that appear to contain audio intended to be heard by the other participants, with each endpoint receiving a flow that does not contain that participant's own audio. For video, the traditional media server may elect to send only video showing the current active speaker, a tiled composition of all participants or the most recent active speakers, a video flow with the active speaker presented prominently with other participants presented as thumbnail images, or some other composite arrangement. It is also common for audio or video to be transcoded. A typical traditional media server is depicted in Figure 1.

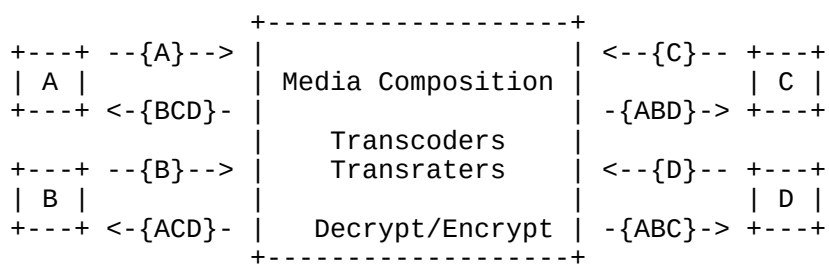

Figure 1 - Traditional Media Server

Traditional media servers require a significant amount of processing power, which in turn translates into a high cost for conferencing hardware manufacturers. Significantly, too, it is very difficult to deploy these servers in a cloud environment due to the high processing demands, as the specialized hardware found in the traditional media server does not exist in a cloud environment.

To enable the traditional media server to perform its job, the server establishes one or more SRTP sessions with each of the conference endpoints wherein it is given access to the keys required to decrypt and encrypt media flows from and to each endpoint. This means that the traditional media server is necessarily a fully trusted entity in the communication path. Any time these servers are deployed in a network that is not secured, it increases the risk that an adversary might gain access to cryptographic key material, allowing the adversary to be able to see and listen to ongoing conferences. In some instances, depending on how the hardware is designed and how keys and certificates are managed, it might be possible for an adversary to see and listen to previously recorded conferences or future conferences.

The Secure Real-time Transport Protocol (SRTP) [RFC3711] is a profile of RTP, which can provide confidentiality, message authentication, and replay protection to the RTP traffic and to the RTP Control Protocol (RTCP). Encryption of header extension in SRTP [RFC6904] provides a mechanism extending the mechanisms of [RFC3711], to selectively encrypt RTP header extensions in SRTP. [RFC3711] and [RFC6904] solves end-to-end use cases between two endpoints, and does not consider use cases where a sender delivers media to a receiver via a cloud-based conferencing service.

5. Motivation for Private Media using switching MDDs

5.1. Switching Media in Cloud Services

There is a trend in the industry for enterprises to use cloud services to host multi-party conferences and meet-me services, either exclusively or to meet peak loads on-demand. At the same time, there is shift toward using lightweight, cost-effective switching MDDs in cloud services that do not necessarily need to mix audio or composite/transcode video. Also fueling the use of such lightweight MDDs is the desire to fully exploit virtualized computing resources and dynamic scalability potential available in cloud computing environments.

The increased use of cloud services has exposed a problem. There are two different trust domains from a media perspective: endpoints and other devices in a trusted domain, and MDDs controlled by the cloud service in an untrusted domain. Other examples of conference devices spread across trusted and untrusted domains are likely, but the cloud service trend is triggering the urgency to address the need to allow for lightweight media conference while enabling media privacy at the same time.

With a switching MDD, each endpoint transmits media as it would with a traditional media server. However, the switching MDD merely forwards all or a subset of the media to the other endpoints in the conference (where at least one other endpoint may be associated with

a cascaded media distribution device), leaving composition to the receiving endpoint. It is also worth noting that, for a switching MDD model to work successfully, each endpoint in the conference must support the media formats transmitted by all other endpoints in the conference. More modern endpoints support multiple codecs and formats, making this commercially practical.

Figure 2 depicts an example of a switching MDD wherein each endpoint is receiving the media flows transmitted by each of the other endpoints in the conference.

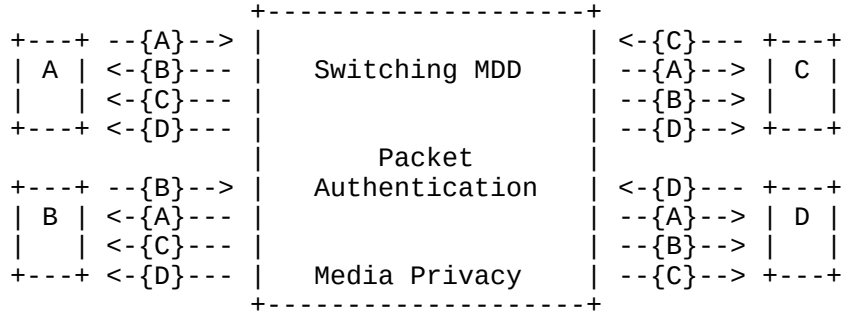

Figure 2 - Switching Media Distribution Device

Note - The use of multiple arrows directed toward each endpoint is not intended to suggest the use of separate RTP sessions.

By using methods such as those described in [RFC6464], it is possible for the switching MDD to transmit the appropriate audio and video flows to endpoints without having knowledge of the content of the encrypted media. The following "Active Speaker Switching" examples help illustrate this point.

In Figure 3, endpoints A, B and D receive the video streams from endpoint C, the currently active speaker, which is receiving video from endpoint A, the previous active speaker. Later when endpoint B becomes the active speaker (Figure 4), endpoints A, C and D will start to receive video from B, while endpoint B continues to receive video from endpoint C. Finally in Figure 5, endpoint A becomes the active speaker.

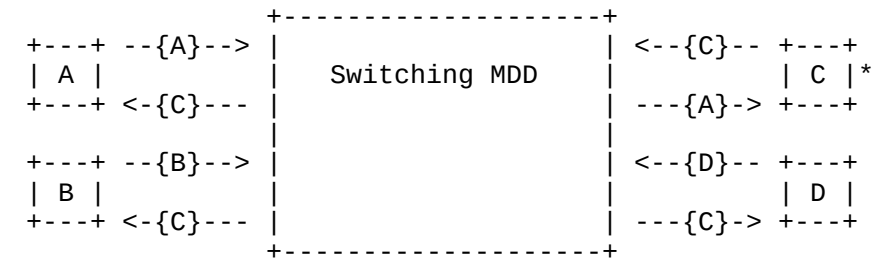

Figure 3 - Endpoint "C" is the Active Speaker

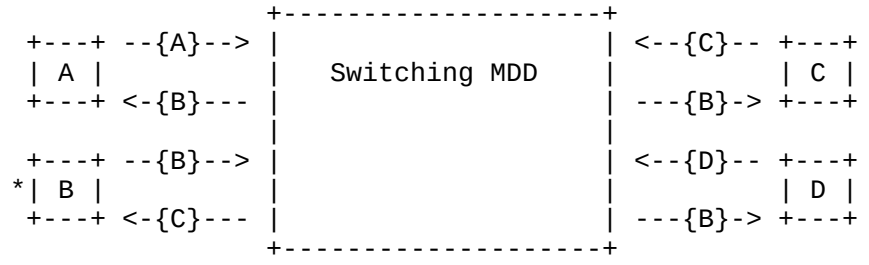

Figure 4 - Endpoint "B" is the Active Speaker

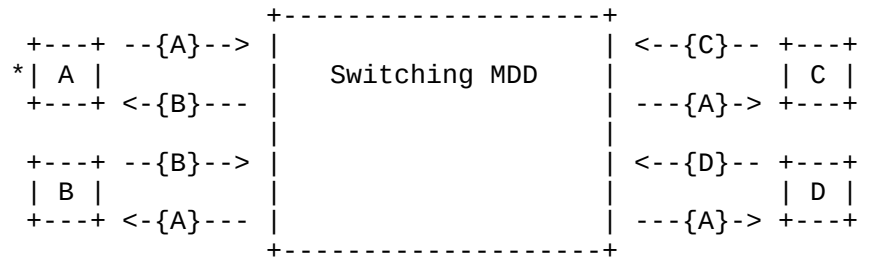

Figure 5 - Endpoint "A" is the Active Speaker

Switched media can also enable conferences to scale to include many more endpoints simultaneously than would be possible with a traditional media server. Like traditional media servers, switching MDDs can also be cascaded or interconnected in a meshed topology to increase the size of the conference without putting undue burden on any particular server.

5.2. Private Media Security through Switching

A traditional media server, or MCU, establishes an SRTP session with each endpoint separately, and needs to decrypt packets containing media for presentation to other endpoints. By using a switching MDD, it is possible to keep the media encryption keys private to the endpoints such that the MDD does not have access to the keys used for

media encryption. The switching MDD just forwards media received to each of the other endpoints in the conference.

This provides for a significantly improved security model, as one can, for example, utilize conferencing resources in the cloud that do not have to be trusted. That said, there may be situations where the switching MDD needs to modify the RTP packet received from an endpoint, such as by adding or removing an RTP header extension, modifying the payload type value, etc. It would be the responsibility of the switching MDD to ensure that media of the expected type and containing the correct information is received by a recipient.

Thus, there is a need to utilize an end-to-end encryption and authentication key (or pair of keys) and a hop-by-hop encryption and authentication key (or pair of keys). The end-to-end encryption and authentication key(s) is to ensure that media remains private to the trusted endpoints. The hop-by-hop authentication key allows the switching MDD to authenticate RTP and RTCP packets and to optionally modify certain elements of those packet. The hop-by-hop encryption key is to optionally encrypt RTP header extensions and optionally encrypt RTCP packets. The current SRTP and related specifications do not define use of a dual-key (hop-by-hop and end-to-end) approach. However, such an approach is possible and would result in ensuring the privacy of media while also enabling the more scalable switched conferencing model.

This dual-key model does necessitate a change in the way that keys are managed. However, the topic of key management is outside the scope of this requirements document. High-level assumptions, such as if the end-to-end context uses a group key as SRTP master key or if individual SRTP master keys (that may be derived/negotiated from another group key), are likely to influence the solution derived from this document.

6. Private Media Trust Model

The architectural model suggested in this document enables switching MDDs to be hosted in domains in which the network elements may have low trust, or where the trustworthiness is uncertain. This does not mean that the service provider is completely untrusted; it simply means that high enough trust with media decryption is not required. This has the benefit of protecting the endpoint’s media in the case of external attacks against the MDD.

In this model, certain elements are considered trusted and others are considered untrusted. Trust in the context of this document means that the element can be in possession of the media encryption key(s) for a past, current, or potentially future conference (or portion thereof) used to protect media content.

In the general case, only the endpoint and an associated key management function, which may be integrated with the endpoint or in a separate stand-alone entity, needs to be trusted. However, it is recognized that in certain deployments, some elements that are classified as untrusted in this document might be placed into the trusted domain and thus be considered trusted. One example might be a gateway, traditional media server or other MDD in a trusted environment connecting endpoints to the same private media conference. This document does not preclude such deployment combinations, but does not rely on them in order to keep the examples and model definitions focused on the simple, most general case.

Each of the elements discussed below has a direct or indirect relationship with each other. The following diagram depicts the trust relationships described in the following sub-sections and the media or signaling interfaces that exist between them, showing the trusted elements on the left and untrusted elements on the right. Note that this is a functional diagram and elements may be co-located or further divided into multiple separate physical entities. Further, it is not necessary that every interface exist between all elements, such as both an interface from the endpoint and call processing function to a key management function, though both are possible options.

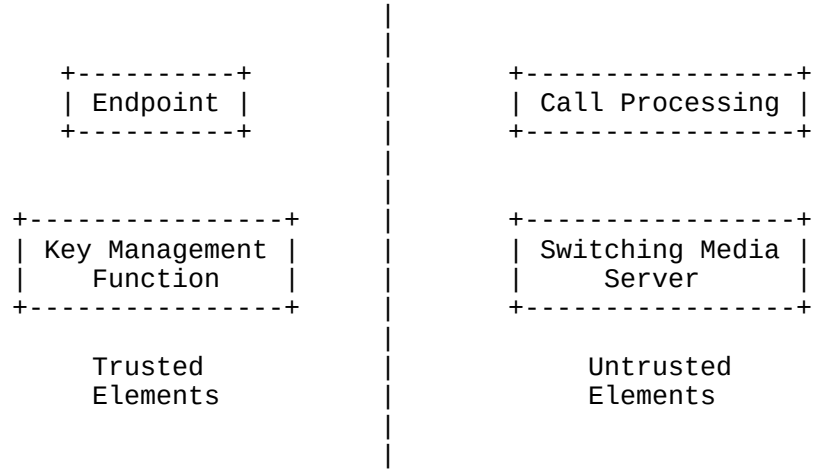

Figure 6 - Relationship of Trusted and Untrusted Elements

6.1. Trusted Elements

The endpoint is considered a trusted element, as it will be sourcing media flows transmitted to other endpoints and will be receiving media for rendering. While it is possible for an endpoint to be compromised and perform in unexpected ways, such as transmitting a decrypted copy of media content to an adversary, such security issues and defenses are outside the scope of this document.

The other trusted element is a key management function (KMF), which may be integrated with the endpoints or exist standalone. This function is responsible for providing cryptographic keys to the endpoints for encrypting and authenticating media content. The KMF is also responsible for providing cryptographic keys to the conferencing resources, such as the MDD, to enable authentication of media packets received by an endpoint. Interaction between the KMF and untrusted call processing functions may be necessary to ensure endpoints are delivered the appropriate keys. The KMF needs to be tightly controlled and managed to prevent exploitation by an adversary, as any kind of security compromise of the KMF puts the security of the conference at risk.

6.2. Untrusted Elements

The call processing function is responsible for such things as authenticating the user or endpoint for the purpose of joining a conference, signing messages, and processing call signaling messages. This element is responsible for ensuring the integrity, and optionally the confidentiality, of call signaling messages between itself, the endpoint, and other network elements. However, it is considered an untrusted element for the purposes of this document, as it cannot be trusted to have access to or be able to gain access to cryptographic key material that provides privacy and integrity of media packets.

There might be several independent call processing functions within an enterprise, service provider network, or the Internet that are classified as untrusted. Any signaling information that passes through these untrusted entities is subject to inspection by that element and might be altered by an adversary.

Likewise, there may be certain deployment models where the call processing function is considered trusted. In such cases, trusted call processing functions MUST take responsibility for ensuring the integrity of received messages before delivering those to the endpoint. How signaling message integrity is ensured is outside the scope of this document, but might use such methods as defined in [RFC4474].

The final element is the switching MDD, which is responsible for forwarding encrypted media packets and conference control information to endpoints in the conference. It is also responsible for conveying secured signaling between the endpoints and the key management function, acquiring per-hop authentication keys from the KMF, and performing per-hop authentication operations for media packets. This function might also aggregate conference control information and initiate various conference control requests. Forwarding of media packets requires that the switching MDD have access to RTP headers or header extensions and potentially modify those message elements, but

the actual media content MUST not be decipherable by the switching MDD.

Further, the switching MDD does not have the ability to determine whether an endpoint is authorized to have access to media encryption keys. Merely joining a conference MUST NOT be interpreted as having authority. Media encryption keys are conveyed to the endpoint by the KMF in such a way as to prevent the switching MDD from having access to those keys.

It is assumed that an adversary might have access to the switching MDD and have the ability to read any of the contents that pass through. For this reason, it is untrusted to have access to the media encryption keys.

As with the call processing functions, it is appreciated that there may be some deployments wherein the switching MDD is trusted. However, for the purposes of this document, the switching MDD is considered untrusted so that we can be ensure to develop a solution that will work even in the most hostile environments.

It is expected that a switching MDD performs its role in properly forwarding media packets, taking measures to safeguard against replay attacks, etc. If a MDD is exploited, an adversary may do such things as discard packets, replay packets, or introduce unacceptable delay in packet delivery.

7. Goals and Non-Goals

7.1. Goals

7.1.1. Ensure End-To-End Confidentiality

The content of the communication and all media needs to be confidential within the group of entities explicitly invited into the conference. An external monitoring adversary should not be able to deduce the human-to-human communication that actually occurred from capturing the media packets.

At the same time, it is necessary to allow switching MDDs to manipulate certain RTP header fields like the payload type value.

7.1.2. Ensure End-To-End Source Authentication of Media

In a conference system with multiple endpoints it is vital that the media content presented to any of the human participants is from the stated endpoint, and not an adversary that attempts to inject misleading content. Nor should an adversary be able to fool the system into becoming a trusted party in the conference. Only explicitly invited parties shall be able to contribute content.

7.1.3. Provide a More Efficient Service than "Full-Mesh"

A multi-party conference that has the goals of confidentiality and source authentication can be established as a "full mesh" (i.e., each participating endpoint directly addresses each of the other endpoints). However, this has a significant issue with the amount of consumed resources in both the uplink and the downlink from each endpoint.

A switched conferencing model would yield the efficiencies desired.

7.1.4. Support Cloud-Based Conferencing

To achieve cost-effective and scalable conferencing, it must be possible to run the MDD instances in a cloud-based virtualized environment.

From a security standpoint, this is a significant issue since the virtualized server instance and the underlying hardware and software upon which it runs might not be secure from an adversary.

7.1.5. Limiting an Endpoint’s Access to Content

Since an invited endpoint will be provided with the content protection keys, the endpoint can decrypt content from time periods before and after the endpoint joined the conference. However, this is not always desirable. It should be possible to re-key the content protection keys every time a participant joins or leaves the conference so each particular set of endpoints uses a unique key.

This also changes the trust level required on the conference roster handling at any point and how to keep that accurate and secured.

It should be noted that timely completion of the re-keying operations become an obstacle in system design and operation. Thus, it is a goal to allow for this possibility when it is deemed essential, but it should not be a requirement on a system to re-key each time the participant list changes.

7.1.6. Compatibility with the WebRTC Security Architecture

It is a goal of this work to ensure compatibility with the WebRTC security architecture as described in [I.D-rtcweb-security-arch]. As an example, local resources that are considered a part of the trusted computing base (TCB), such as keying material derived using DTLS-SRTP, will remain within the TCB and not exposed to untrusted entities.

The browser is reliant on an external calling service to convey signaling information that may open the door for a man-in-the-middle attack, such as the conveyance of certificate fingerprints over the

interface between the browser and the calling service. However, as described in [I.D-rtcweb-security-arch], the browser may utilize additional services, such as a trusted identify provider, to mitigate such risks.

Having said the foregoing, this document does not aim to define requirements for end-to-end security for the WebRTC data channel.

7.2. Non-Goals

7.2.1. Securing the Endpoints

The security of a communication session requires that the endpoints are not compromised and that the users are trustworthy. If not, credentials and decrypted content may be shared with third parties. However, this is hard to prevent through system design. Thus, it should be assumed that the endpoint is secure and the user is trustworthy; how to achieve this is out of scope this document.

7.2.2. Concealing that Communication Occurs

A non-goal is to attempt to prevent a pervasive monitoring adversary from knowing that the communication session has occurred. The reason for excluding this as a goal is that it is extremely difficult to achieve, as a pervasive monitoring adversary can be expected to be able to have knowledge of all IP flows that enter or exit local ISPs, across links that straddle national borders or internet exchange points. To hide the fact communication occurred, the flows required to achieve the communication session need to be highly difficult to correlate between different legs of the communication.

At this stage this is deemed too difficult to attempt and will need to be a subject for further study. Existing attempts include The Onion Router (TOR), against which it has been claimed to be possible to monitor, at least partially, by an adversary with sufficient reach.

Also of consideration is that trying to conceal the fact that communication occurred actually makes it more difficult for network administrators to effectively manage and troubleshoot issues with conference calls.

7.2.3. Individual Media Source Authentication

Although the endpoints in the conference are authenticated, it is not a goal to provide source authentication of the media at the individual user level, instead being satisfied with being able to authenticate media as coming from an invited endpoint or not.

There exist solutions that can provide individual media source authentication (e.g., TESLA). However, they impact the performance

or security properties they provide. Thus, further study is required to determine impact and resulting security properties if desired to have individual source authentication.

7.2.4. Multicast -based Conferencing

Using multicast to construct a non-centralized media distribution model is out of scope. This document is focused only on models where endpoints, or other devices, participating in a conference unicast media to a centrally located media distribution device.

8. Requirements

The following are the security solution requirements for switched conferencing that enable end-to-end media privacy between all endpoints.

Note that while some switching MDDs might be fully trusted entities, the intent of this solution and purpose for these requirements is to address those servers that are not trusted.

PM-01: Switching media distribution device MUST be able to switch the media between endpoints in a conference without having access to unencrypted media content.

PM-02: Solution MUST maintain all current SRTP security goals, namely the ability to provide for end-to-end confidentiality, provide for hop-by-hop replay protection, and ensure hop-by-hop and end-to-end message integrity.

PM-03: Solution MUST extend replay protection to cover each hop in the media path, both ensuring that any received packet is destined for the recipient and not a duplicate.

PM-04: Keys used for end-to-end encryption and authentication of RTP payloads and other information deemed unsuitable for access by the switching media distribution device MUST NOT be generated by or accessible to any component that is not trusted.

PM-05: The switching media distribution device MUST be allowed to make changes to the RTP header and the RTP header extensions.

PM-06: A cryptographic context suitable for enabling end-to-end authenticated encryption MUST be defined.

PM-07: The switching media distribution device, or any entity that is not fully trusted, MUST NOT be involved in the user or endpoint authentication for the purpose of media key distribution.

PM-08: The switching media distribution device MUST be able to switch an already active RTP stream to a new receiver, while guaranteeing the timely synchronization between the RTP security context of the transmitter and its current and new receivers.

PM-09: It MUST be possible for the switching media distribution device to determine if a received media packet was transmitted by an endpoint in possession of a valid hop-by-hop key for that conference.

PM-10: It MUST be possible for a conference to be optionally re-keyed as desired, such as each time a participant joins or leaves the conference.

PM-11: Any solution satisfying this requirements document MUST provide for a means through which WebRTC-compliant endpoints can participate in a switched conference using private media as outlined herein.

PM-12: All RTP senders, including the switching media distribution device, MUST adhere to all congestion control requirements that are required by the RTP profile and topology in use, including RTP circuit breakers [I.D-ietf-avtcore-rtp-circuit-breakers]. Since the switching media distribution device is unable to perform transcoding or transrating that requires access to the unencrypted media, its reaction to congestion signals is often limited to dropping packets that would otherwise be forwarded in the absence of congestion, and signaling congestion to the RTP source. This is similar to the congestion control behavior of the Media Switching Mixer and Selective Forwarding Middlebox/Unit in [I.D-ietf-avtcore-rtp-topologies-update].

PM-13: It MUST be possible for a media distribution device or an endpoint to authenticate a received RTCP packet.

9. IANA Considerations

There are no IANA considerations for this document.

10. Security Considerations

[TBD]

11. References

11.1. 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.

[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.

[I.D-rtcweb-security-arch] E. Rescorla, "WebRTC Security Architecture", Work in Progress, March 2015.

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

[I.D-ietf-avtcore-rtp-topologies-update] Westerlund, M., and S. Wenger, "RTP Topologies", Work in Progress, March 2015.

[I.D-ietf-avtcore-rtp-circuit-breakers] Perkins, C. S., and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", Work in Progress, March 2015.

11.2. Informative References

[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.

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

12. Acknowledgments

The authors would like to thank Marcello Caramma, Matthew Miller, Christian Oien, Magnus Westerlund, Cullen Jennings, Christer Holmberg, Bo Burman, Jonathan Lennox, Suhas Nandakumar, Dan Wing, Roni Even, and Mo Zanaty for their invaluable input.

13. Contributors

Yi Cheng
Ericsson
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 17 589
Email: yi.cheng@ericsson.com

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

Nathan Buckles
Cisco Systems, Inc.
170 W Tasman Dr.
San Jose
USA

Email: nbuckles@cisco.com

John Mattsson
Ericsson AB
SE-164 80 Stockholm
Sweden

Phone: +46 10 71 43 501
Email: john.mattsson@ericsson.com

Richard Barnes
Mozilla
331 E Evelyn Ave.

Mountain View
USA

Email: rlb@ipv.sx