Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 4, 2018

V. Singh
callstats.io
Inamdar
Ravindranath
Cisco Systems, Inc.

December 31, 2017

Multipath RTP (MPRTP) with Secure Real-Time Transport (SRTP)
draft-kaustubh-mprtp-dtls-srtp-02

Abstract

This document describes the considerations of using Multipath RTP (MPRTP) with Datagram Transport Layer Security (DTLS) protocol.

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 https://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 July 4, 2018.

Table of Contents

1. Introduction

Multi path RTP(MPRTP) [1] is an extension to RTP that allows multi homed endpoints to use a plurality of transmission paths to send/ receive media.

MPRTP functions as a layer of abstraction between the RTP stack and the multiplicity of transport paths available for media transmission by splitting and recombining media streams.

Datagram Transport Layer Security (DTLS) [RFC4347] is a channel security protocol that offers integrated key management, parameter negotiation, and secure data transfer. Because DTLS data transfer protocol is generic, it is less highly optimized for use with RTP than is SRTP, which has been specifically tuned for that purpose, DTLS SRTP [RFC5764] is an extension to DTLS that is optimized to work with Secure Real Time transport protocol [RFC3711] to provide integrated key management, SRTP algorithm negotiation and SRTP parameter negotiation.

[I-D.ietf-avtcore-mprtp] [2]conceptually introduces the possibility of transmitting RTP over a plurality of sub flows using an extension to RTP, however in real world deployments there is a need to secure

transmission paths whether singular or multiple. The motivation of this draft is to highlight the operating principles of MPRTP with DTLS SRTP.

2. Motivation

When DTLS SRTP is used with MPRTP there are several cryptographic considerations that arise from the perspective of SRTP and DTLS-SRTP. The following sections of this draft highlight these considerations.

3. SRTP Cryptographic Context considerations

A SRTP cryptographic context is maintained by key management and provides several parameters that are vital to the proper operation of the SRTP framework (E.g. ROC, index, master keys, session keys etc.) In the case of a single RTP stream that is secured via DTLS-SRTP, there is tight synchronization between different cryptographic parameters on sending and receiving application.

In the case of MPRTP, due to the presence of sub-flows, there can be two possible approaches with regards to securing media traffic using DTLS-SRTP:

1. Re-use the same DTLS-SRTP association as traffic fails over from interface to another

2. Use multiple DTLS-SRTP associations if traffic uses several sub-flows concurrently.

The implications of using either approach are detailed in the sub-sections below.

3.1. DTLS association re-use

One of major use cases of MPRTP is that of mobility, wherein a currently active interface experiences a severe degradation of transmission quality or disappears altogether as the device moves between networks. In such cases the MPRTP stack may offload all traffic to a secondary preference interface to continue media transmission. This change in the media address would require an updated offer/answer exchange to be sent when ICE is used, in other cases in-band RTCP based advertisements may be used. For such scenarios instead of setting up an entirely new DTLS SRTP association, the existing association can be reused by retaining the same 'dtls-id' as defined by <https://tools.ietf.org/html/draft-ietf-mmusic-dtls-sdp-22>

In cases where there is a need to gradually offload traffic from an currently active interface to an another/additional interfaces, a new DTLS SRTP association must be setup or alternatively, there might be a future specification/extension to DTLS-SRTP that defines such behavior.

3.2. Cryptographic index

When using multiple sub-flows with MPRTP, each sub-flow sets up a different DTLS SRTP association, the cryptographic context for each DTLS SRTP association is unique and referenced using a distinct triplet identifier.

<SSRC, destination network address, destination transport port number>

While de-queuing packets from the application, the MPRTP stack may choose to distribute these packets across several active sub-flows after packet encryption and optional authentication, such that each active sub-flow would most likely not service packets with monotonously increasing global RTP sequence numbers (the per flow sequence numbers however, would be monotonously increasing)

The encryption process specified in section 3.3 of RFC3711 when applied to the MPRTP scenario would cause a huge skew when indices for successive packets in a given sub-flow are calculated as the index value uses the global RTP sequence number as per the following definition

i = 2^16 * ROC + SEQ

When packets arrive out-of-order, this skew in packet indices can cause the replay protection algorithm on the receiving application to misfire and incorrectly drop packets as the replay window would accept packet indices that slightly lag behind the current cryptographic index value. Setting the window size to be large enough to accept packets whose indices drastically lag behind the current cryptographic context index value, renders the replay protection algorithm ineffective and incapable of identifying replay attacks. This particular problem is exacerbated when a certain sub-flow(s) are scantily used for packet transmission with majority of the packets transmitted on other sub-flows with better path characteristics.

3.3. Cryptographic keys

RFC3711 allows for sharing of keys across different RTP streams in a media session. From the perspective of MPRTP, there are two major problems that could arise with key sharing across different sub-flows.

3.3.1. Two-time keypad

MPRTP makes use of the same SSRC value across all sub-flows of a given media type (e.g. audio), in general scenarios, unique keystreams are generated per packet regardless of the sub-flow over which they are transmitted as the index of the packet being encrypted in unique. However in scenarios where certain critical packets are transmitted over all or some of the sub-flows for the sake of redundancy and reliability (e.g. I-Frame, named telephony events), the very same keystream value is generated and leads to "two-time key pad" rendering the secure framework open to attacks. The chances of a two-type key pad issue is exacerbated if key re-use is allowed among different MPRTP media types (e.g. audio and video) in a given media session as the chance of an attacker detecting duplicate keystreams increases at least by a factor of two.

3.3.2. Authentication key re-use

The SSRC field in an SRTP packet is an authentication-protected field and if the same authentication key is used, an attacker can substitute one stream into another causing media playout issues at the receiver application.

3.4. Coordination between a plurality of cryptographic contexts and MPRTP

MPRTP involves the splitting of a single RTP stream into a number of subflows that appear as distinct streams from the perspective of the network. However the MPRTP stack at the receiver side is responsible for re-combining these streams and presenting a single flow of RTP packets to the application. Due to the presence of multiple sub-flows (with distinct network addresses and ports), a separate DTLS-SRTP association is required per sub-flow, with each association maintaining a distinct set of cryptographic parameters as per section 3.2.1 of RFC3711.

With each encrypt/decrypt cycle occurring across sub-flows, the MPRTP stack on the sender and receiver side has to ensure that various parameters of the cryptographic context are updated across each sub-flow, followed by correct sequencing of packets before it is presented to the application. The computational costs of maintaining

multiple sub-flows, running several encrypt/decrypt cycles per sub-flow and sequencing packets correctly is significantly higher in comparison to a single RTP stream.

4. Re-keying considerations

To avoid the "two-time key pad" problem, it is necessary for an SRTP/SRTCP stream to re-key (master key) every time the 48 bit index space is exhausted, this ensures that duplicate key-streams aren't generated. As discussed in section 2.1, MPRTP may setup sub-flows that are scantily used in packet transmission, in which case a given sub-flow would quickly exhaust it's index space, roll over and possibly produce duplicate keystreams leading to a potential breakdown of the SRTP framework. If the master key is shared across several sub-flows this would certainly lead to frequent re-keying across several sub-flows adding to the cryptographic load.

5. Encrypting MPRTP header extensions

MPRTP uses header extensions in RTP and RTCP packets for the following use-cases:

1. To communicate the sub-flow ID and sub-flow specific sequence numbers

2. To report per sub-flow RTCP reports

3. For interface advertisement (RTCP)

The transforms and constructs of RFC3711 encrypt only the payload of the SRTP packets, without considering the header extensions. Given that the MPRTP header extension could be visible to an attacker, fields like the sub-flow specific ID or sub-flow specific sequence numbers can easily be manipulated, causing issues on the receiving application. For example, an adversary can change the sub-flow specific sequence number to indicate a drastic change causing the receiving application to drop the packet. The sub-flow specific ID could be also be changed to reflect a stream ID that is non existent causing the receiving application to completely drop all packets corresponding to the rogue sub-flow ID.

NOTE:As per my our internal discussion, creation of an authentication tag would ensure that RTP header extensions are unaltered, however in the case where encryption proceeds without authentication, it may be better to encrypt the MPRTP header extensions (a counter argument could be that even normal RTP headers like SSRC, global sequence numbers etc. could be altered without authentication)

6. DTLS associations

As discussed in earlier sections of this draft, multiple DTLS SRTP associations must be established per sub-flow in an MPRTP setup, maintaining multiple sub-flows with DTLS SRTP does bring up some additional considerations that are discussed below:

6.1. DTLS Session Resumption

For related media streams within a RTP session, it is advised to use DTLS session resumption to reduce the cost of cryptographic operations. Using DTLS session resumption leads to the re-use of the master key across all the sub-flows, which could lead to the problems highlighted in section 2.2.1 and 3. It is advisable to use parallel, distinct DTLS associations to protect the sub-flows such that the keys are unique across all sub-flows.

6.2. Keeping Sub-flows Alive

Certain MPRTP sub-flows that are secure via DTLS SRTP, might be used sparingly for packet transmission, with the majority of traffic being sent over other high priority sub-flows (as determined via ICE or a local algorithm at the sender side), in order to ensure that sparingly used sub-flows at the DTLS layer, the DTLS heartbeat extension as defined in RFC6520 may be used. This ensures that the costly operation of a DTLS re-negotiation is avoided and also ensures that TURN or STUN bindings are refreshed if media traverses through NAT's or relays.

6.3. Late Binding of Cryptographic Contexts

As DTLS SRTP associations are agnostic to the SSRC of media streams, DTLS-SRTP uses a "late binding" mechanism as far as cryptographic contexts are concerned. A MPRTP endpoint can have multiple DTLS SRTP associations, in which case on receiving the SRTP packet, an assertion needs to be made on what association that SSRC corresponds to, so, initially the cost of the algorithm to determine this will be equal to the number of sub-flows and the algorithm might require additional passes as sub-flows are added.

7. Security Considerations

TBD

8. Acknowledgements

9. References

9.1. Normative References

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <https://www.rfc-editor.org/info/rfc4145>.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, DOI 10.17487/RFC4347, April 2006, <https://www.rfc-editor.org/info/rfc4347>.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, DOI 10.17487/RFC4572, July 2006, <https://www.rfc-editor.org/info/rfc4572>.

[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, DOI 10.17487/RFC5764, May 2010, <https://www.rfc-editor.org/info/rfc5764>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <https://www.rfc-editor.org/info/rfc5761>.

[I-D.ietf-avtcore-mprtp] Singh, V., Karkkainen, T., Ott, J., Ahsan, S., and L. Eggert, "Multipath RTP (MPRTP)", draft-ietf-avtcore-mprtp-03 (work in progress), July 2016.

9.2. URIs

[1] https://tools.ietf.org/html/draft-singh-avtcore-mprtp-10

[2] https://tools.ietf.org/html/draft-singh-avtcore-mprtp-10

Authors' Addresses

Varun Singh
callstats.io
Annankatu 31-33 C 42,
Helsinki 00100
Finland

Email: varun@callstats.io
URI: http://www.callstats.io/

Kaustubh Inamdar
Cisco Systems, Inc.
Cessna Business Park ,
Kadabeesanahalli Village, Varthur Hobli,
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: kinamdar@cisco.com

Ram Mohan Ravindranath
Cisco Systems, Inc.
Cessna Business Park ,
Kadabeesanahalli Village, Varthur Hobli,
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: rmohanr@cisco.com