ICE V. Singh
Internet-Draft callstats.io
Intended status: Informational K. Inamdar
Expires: December 31, 2017 R. Ravindranath
Cisco Systems, Inc.
June 29, 2017

Multipath RTP (MPRTP) with Secure Real-Time Transport (SRTP)
draft-kaustubh-mprtp-dtls-srtp-00

Abstract

This document describes the considerations when using Multipath RTP (MPRTP) with Secure Real-time Transport (SRTP) security context set up with the Datagram Transport Layer Security (DTLS) protocol.

Table of Contents

1. Introduction

[I-D.ietf-avtcore-mprtp] is an extension to RTP that allows multi homed endpoints to use 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) [RFC6347] 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.

MPRTP [I-D.ietf-avtcore-mprtp]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 [I-D.ietf-avtcore-mprtp] with DTLS-SRTP [RFC5764].

2. Motivation

MPRTP [I-D.ietf-avtcore-mprtp] introduces the concept of transmitting RTP over multiple subflows. When DTLS-SRTP [RFC5764] is used with MPRTP [I-D.ietf-avtcore-mprtp]there are different design considerations possible. The following sections of this draft highlight some of these considerations.

3. SRTP Cryptographic Context considerations

Each SRTP stream requires the sender and receiver to maintain cryptographic state information which is called "SRTP Cryptographic Contexts" as defined in Section 3.2 of [RFC3711]. A SRTP cryptographic context is maintained for each SRTP session 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 subflows, 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 subflows concurrently.

The implications of using either approach are detailed in the sub-seections 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 [I-D.ietf-ice-rfc5245bis] 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 'tls-id' as defined by [I-D.ietf-mmusic-dtls-sdp]

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 subflows with MPRTP, each subflow 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 subflows after packet encryption and optional authentication, such that each active subflow 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 subflow 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 subflow(s) are scantily used for packet transmission with majority of the packets transmitted on other subflows 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 subflows.

3.3.1. Two-time keypad

MPRTP makes use of the same SSRC value across all subflows of a given media type (e.g. audio), in general scenarios, unique keystreams are generated per packet regardless of the subflow 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 subflows 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 subflows (with distinct network addresses and ports), a separate DTLS-SRTP association is required per subflow, 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 subflows, the MPRTP stack on the sender and receiver side has to ensure that various parameters of the cryptographic context are updated across each subflow, followed by correct sequencing of packets before it is presented to the application. The computational costs of maintaining multiple subflows, running several encrypt/decrypt cycles per subflow

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 arent generated. MPRTP may setup subflows that are scantily used in packet transmission, in which case a given subflow would quickly exhaust its 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 subflows this would certainly lead to frequent re-keying across several subflows 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 subflow ID and subflow specific sequence numbers

2. To report per subflow 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 subflow specific ID or subflow specific sequence numbers can easily be manipulated, causing issues on the receiving application. For example, an adversary can change the subflow specific sequence number to indicate a drastic change causing the receiving application to drop the packet. The subflow 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 subflow ID.

SRTP authentication tag would ensure that RTP header extensions are unaltered, however in the case where encryption proceeds without authentication, it may be desirable to encrypt the MPRTP header extensions.

6. DTLS associations

As discussed in earlier sections of this draft, multiple DTLS-SRTP associations must be established per subflow in an MPRTP setup, maintaining multiple subflows 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 subflows, which could lead to the problems highlighted in Section 3.1. It is advisable to use parallel, distinct DTLS-SRTP associations to protect the subflows such that the keys are unique across all subflows.

6.2. Keeping subflows Alive

Certain MPRTP subflows that are secure via DTLS SRTP, might be used sparingly for packet transmission, with the majority of traffic being sent over other high priority subflows (as determined via ICE [I-D.ietf-ice-rfc5245bis] or a local algorithm at the sender side), in order to ensure that sparingly used subflows 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 NATs 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 subflows and the algorithm might require additional passes as subflows are added.

7. Security Considerations

TBD

8. IANA Considerations

This document does not add any new extensions. No updates needed to IANA registry.

9. Acknowledgements

10. References

10.1. Normative References

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[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, <http://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, <http://www.rfc-editor.org/info/rfc3711>.

[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.

10.2. Informative References

[I-D.ietf-ice-rfc5245bis] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-10 (work in progress), May 2017.

[I-D.ietf-mmusic-dtls-sdp] Holmberg, C. and R. Shpount, "Using the SDP Offer/Answer Mechanism for DTLS", draft-ietf-mmusic-dtls-sdp-26 (work in progress), June 2017.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, DOI 10.17487/RFC6520, February 2012, <http://www.rfc-editor.org/info/rfc6520>.

Authors' Addresses

Varun Singh
CALLSTATS I/O Oy
Annankatu 31-33 C 42,
Helsinki 00100
Finland

Email: kinamdar@cisco.com
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