mmusic
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2016

C. Jennings
P. Jones
Cisco
A. Roach
Mozilla

October 18, 2015

# SRTP Double Encryption Procedures
## draft-jennings-perc-double-00

Abstract

In some conferencing scenarios, it is desirable for an intermediary to be able to manipulate some RTP parameters, while still providing strong end-to-end security guarantees. This document defines a SRTP procedures that uses two separate but related cryptographic contexts to provide "hop by hop" and "end to end" security guarantees. Both the end-to-end and hop-by-hop cryptographic transforms can utilizes an authenticated encryption with associated data scheme or take advantage of future SRTP transforms with different properties. SRTCP is encrypted hop-by-hop using an already-defined SRTCP cryptographic transform.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 20, 2016.

## 1. Introduction

Cloud conferencing systems that are based on switched conferencing have a central media distribution device (MDD) that receives media from clients and distributes it to other clients, but does not need to interpret or change the media content. For these systems, it is desirable to have one security association from the sending client to the receiving client that can encrypt and authenticated the media end-to-end while still allowing certain RTP header information to be changed by the MDD. At the same time, a separate security association provides integrity and optional confidentiality for the RTP and media flowing between the MDD and the clients. More information about the requirements can be found in [I-D.jones-perc-private-media-reqts].

This specification RECOMMENDS the SRTP AES-GCM transform [I-D.ietf-avtcore-srtp-aes-gcm] to encrypt an RTP packet to form the end-to-end security association. The output of this is treated as an RTP packet and (optionally) again encrypted with an SRTP transform to form the hop-by-hop security association between the client and the MDD. The MDD decrypts and checks integrity of the hop-by-hop security. At this point the MDD may change some of the RTP header information that would impact the end-to-end integrity. For any values that are changed, the original values before changing are included in a new RTP header extension called the Original Header Block. The new RTP packet is encrypted with the hop-by-hop security association for the destination client before being sent. The receiving client decrypts and checks integrity for the hop-by-hop association from the MDD then replaces any parameters the MDD changes using the information in the Original Header Block before decrypting and checking the end-to-end integrity.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Terms:

- MDD: media distribution device that routes media from one client to other clients
- E2E: end-to-end meaning the link from one client through the MDD to the client at the other end.
- HBH: hop-by-hop meaning the link from the client to or from the MDD.
- OHB: Original Header Block containing a TLVs for each value that the MDD Changed in the RTP header.

## 3. Cryptographic Contexts

This specification uses two cryptographic contexts: An "end-to-end" context that is used by endpoints that originate and consume media, and a "hop-by-hop" context" that is used by an MDD that wishes to make modifications to some RTP header fields. The RECOMMENDED cipher for the hop-by-hop and end-to-end context is AES-GCM but as new SRTP ciphers are defined, new combination of the double encryption version of them can be added to the IANA registry.

The keys and salt for these contexts are generated with the following steps:

- Generate key and salt values of twice the length required by the E2E and HBH transforms
- Assign the first part of each value to be the key and salt, respectively, for the inner transform.
- Assign the second part of each value to be the key and salt, respectively, for the outer transform.

Obviously, if the MDD is to be able to modify header fields but not decrypt the payload, then it must have cryptographic context for the outer transform, but not the inner transform. This document does not define how the MDD should be provisioned with this information.

## 4. Original Header Block

Any SRTP packet processed following these procedures MAY contain an Original Header Block (OHB) extension.

This RTP header extension contains the original values of any modified header fields, in the following form:

(type || value) || (type || value) || ...

In each type/value pair, the "type" field indicates the type of parameter that was changed, and the "value" field carries the original value of the parameter. The mapping from RTP header parameters to type values, and the length of the value field is as follows

| Field | Type | Value length |
|---|---|---|
| X | 1 | 1 |
| CC | 2 | 1 |
| M | 3 | 1 |
| PT | 4 | 1 |
| Seq Num | 5 | 2 |
| Timestamp | 6 | 4 |
| SSRC | 7 | 4 |
| Ext Len | 8 | 2 |

Open Issue: We could make a efficient coding by packing the above values as bits in bit field and perhaps packing some of the single values into the same byte.

## 5. Operations

### 5.1. Encrypting a Packet

To encrypt a packet, the endpoint encrypts the packet with the inner transform, may add an OHB, then applies the outer transform.

- Form an RTP packet. If there are any header extensions, they MUST use [RFC5285].

- Apply the transform to the RTP packet

- Optionally add an OHB header extension. The endpoint MAY include any header fields that are signaled to be modified by the MDD, to reduce processing burden on the MDD. Open Issue: do we want the sending client to be able to add an OHB?

- Apply the SRTP cryptographic transform with the outer parameters (outer transform)

## 5.2. Modifying a Packet

In order to modify a packet, the MDD undoes the outer transform, modifies the packet, updates the OHB with any new modifications, and re-applies the outer tranform.

- Apply the (outer) decryption transform to the packet
- Separate the OHB from the (encrypted) original payload
- Change any required parameters
- If a changed parameter is not already in the OHB, add it with its original value to the OHB. Note that in the case of cascaded MDDs, the first MDD may have already added an OHB.
- If the MDD resets a parameter to its original value, it MAY drop it from the OHB.
- The MDD MUST NOT delete any header extensions, but MAY add them.
  - If the MDD adds any header extensions, it must append them and it must maintain the order of the original headers in the [RFC5285] block.
  - If the MDD appends headers, then it MUST add the value of the original [RFC5285] length field to the OHB, or update it if it is already there. The original [RFC5285] length is counted in words and stored in the Ext Len field of the OHB.
- Recombine the new OHB and the (encrypted) original payload
- Apply the (outer) encryption transform to the packet

## 5.3. Decrypting a Packet

To decrypt a packet, the endpoint first decrypts and verifies using the outer transform, then uses the OHB to reconstruct the original packet, which it decrypts and verifies with the inner transform.

- Apply the (outer) decryption transform to the packet
- Separate the OHB from the (encrypted) original payload
- Form a new SRTP packet with:

    * Header = Received header, with header fields replaced with values from OHB

    * Header extensions truncated to the [RFC5285] length in OHB

    * Payload = (encrypted) original payload

  o Apply the (inner) decryption transform to this synthetic SRTP packet

### 5.4. Recommended Inner and Outer Cryptographic Transforms

This specification recommends and defines values for AES-GCM as both the inner and outer cryptographic transforms (DOUBLE_SRTP_AEAD_AES_128_GCM and DOUBLE_SRTP_AEAD_AES_256_GCM). This transform provides for authenticated encryption and will consume additional processing time double-encrypting for HBH. However, the approach is secure and simple, and is thus viewed as an acceptable tradeoff in processing efficiency.

If a new SRTP transform was defined that encrypted some of all of the RTP header, it would be reasonable for systems to have the option of using that for the outer transform. Similarly if a new transform was defined that provided only integrity, that would also be reasonable to use for the HBH as the payload data is already encrypted by the E2E.

## 6. Security Considerations

It is obviously critical that the intermediary have only the outer transform parameters, and not the inner. We rely on an external key management protocol to assure this property.

Modifications by the intermediary result in the recipient getting two values for changed parameters (original and modified). The recipient will have to choose which to use; there is risk in using either that depends on the session setup.

The security properties for both the inner and outer key holders are the same as the security properties of classic SRTP

## 7. IANA Considerations

### 7.1. RTP Header Extension

TODO - Define RTP header extension for the OBP block.

### 7.2. DTLS-SRTP

We request IANA to add the following values to defines a DTLS-SRTP "SRTP Protection Profile" defined in [RFC5764].

```
DOUBLE_SRTP_AEAD_AES_128_GCM    = {TBD, TBD }
DOUBLE_SRTP_AEAD_AES_256_GCM    = {TBD, TBD }
```

The SRTP transform parameters for each of these protection are:

```
DOUBLE_SRTP_AEAD_AES_128_GCM
     cipher:                 AES_128_GCM
     cipher_key_length:      256 bits
     cipher_salt_length:     192 bits
     aead_auth_tag_length:   32 octets
     auth_function:          NULL
     auth_key_length:        N/A
     auth_tag_length:        N/A
     maximum lifetime:       at most 2^31 SRTCP packets and
                                        at most 2^48 SRTP packets

DOUBLE_SRTP_AEAD_AES_256_GCM
     cipher:                 AES_256_GCM
     cipher_key_length:      512 bits
     cipher_salt_length:     192 bits
     aead_auth_tag_length:   32 octets
     auth_function:          NULL
     auth_key_length:        N/A
     auth_tag_length:        N/A
     maximum lifetime:       at most 2^31 SRTCP packets and
                                        at most 2^48 SRTP packets
```

The first half of the key and salt is used for the inner (E2E) transform and the second half is used for the outer (HBH) transform.

## 8. Acknowledgements

Many thanks to review from GET YOUR NAME HERE. Send comments.

## 9. References

### 9.1. Normative References

[I-D.ietf-avtcore-srtp-aes-gcm]
McGrew, D. and K. Igoe, "AES-GCM Authenticated Encryption in Secure RTP (SRTP)", draft-ietf-avtcore-srtp-aes-gcm-17 (work in progress), June 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <http://www.rfc-editor.org/info/rfc5285>.

[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>.

## 9.2. Informative References

[I-D.jones-perc-private-media-reqts] Jones, P., Ismail, N., Benham, D., Buckles, N., Mattsson, J., and R. Barnes, "Private Media Requirements in Privacy Enhanced RTP Conferencing", draft-jones-perc-private-media-reqts-00 (work in progress), July 2015.

## Authors' Addresses

Cullen Jennings
Cisco

Email: fluffy@iii.ca

Paul E. Jones
Cisco

Email: paulej@packetizer.com

Adam Roach
Mozilla

Email: adam@nostrum.com