Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

P. Jones
Cisco Systems
P. Ellenbogen
Princeton University
N. Ohlmeier
Mozilla

October 31, 2016

DTLS Tunnel between a Media Distributor and Key Distributor to Facilitate Key Exchange
draft-jones-perc-dtls-tunnel-04

Abstract

This document defines a DTLS tunneling protocol for use in multimedia conferences that enables a Media Distributor to facilitate key exchange between an endpoint in a conference and the Key Distributor. The protocol is designed to ensure that the keying material used for hop-by-hop encryption and authentication is accessible to the media distributor, while the keying material used for end-to-end encryption and authentication is inaccessible to the media distributor.

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 May 4, 2017.

Copyright Notice

Table of Contents

1. Introduction

An objective of the work in the Privacy-Enhanced RTP Conferencing (PERC) working group is to ensure that endpoints in a multimedia conference have access to the end-to-end (E2E) and hop-by-hop (HBH) keying material used to encrypt and authenticate Real-time Transport Protocol (RTP) [RFC3550] packets, while the Media Distributor has access only to the hop-by-hop (HBH) keying material for encryption and authentication.

This specification defines a tunneling protocol that enables the media distributor to tunnel DTLS [RFC6347] messages between an endpoint and the key distributor, thus allowing an endpoint to use DTLS-SRTP [RFC5764] for establishing encryption and authentication keys with the key distributor.

The tunnel established between the media distributor and key distributor is a TLS connection that is established before any

messages are forwarded by the media distributor on behalf of the endpoint. DTLS packets received from the endpoint are encapsulated by the media distributor inside this tunnel as data to be sent to the key distributor. Likewise, when the media distributor receives data from the key distributor over the tunnel, it extracts the DTLS message inside and forwards the DTLS message to the endpoint. In this way, the DTLS association for the DTLS-SRTP procedures is established between the endpoint and the key distributor, with the media distributor simply forwarding packets between the two entities and having no visibility into the confidential information exchanged.

Following the existing DTLS-SRTP procedures, the endpoint and key distributor will arrive at a selected cipher and keying material, which are used for HBH encryption and authentication by both the endpoint and the media distributor. However, since the media distributor would not have direct access to this information, the key distributor explicitly shares the HBH key information with the media distributor via the tunneling protocol defined in this document. Additionally, the endpoint and key distributor will agree on a cipher for E2E encryption and authentication. The key distributor will transmit keying material to the endpoint for E2E operations, but will not share that information with the media distributor.

By establishing this TLS tunnel between the media distributor and key distributor and implementing the protocol defined in this document, it is possible for the media distributor to facilitate the establishment of a secure DTLS association between an endpoint and the key distributor in order for the endpoint to receive E2E and HBH keying material. At the same time, the key distributor can securely provide the HBH keying material to the media distributor.

2. Conventions Used In This Document

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] 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. Tunneling Concept

A TLS connection (tunnel) is established between the media distributor and the key distributor. This tunnel is used to relay DTLS messages between the endpoint and key distributor, as depicted in Figure 1:

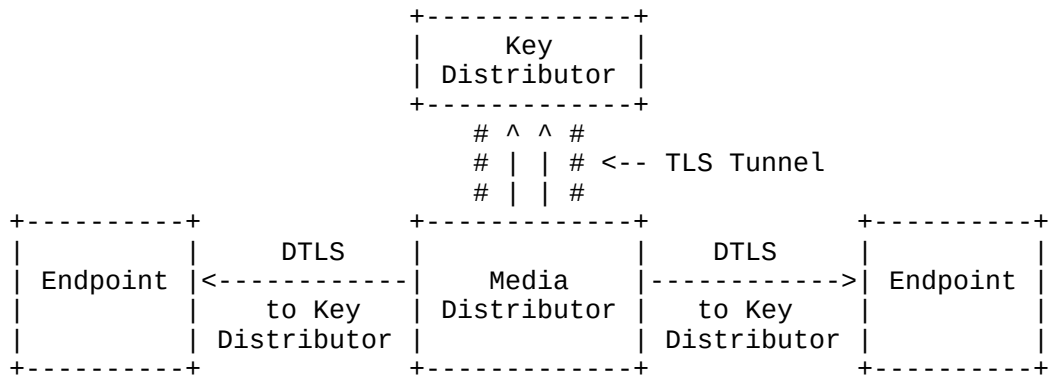

Figure 1: TLS Tunnel to Key Distributor

The three entities involved in this communication flow are the endpoint, the media distributor, and the key distributor. The behavior of each entity is described in Section 5.

The key distributor is a logical function that might might be co-resident with a key management server operated by an enterprise, reside in one of the endpoints participating in the conference, or elsewhere that is trusted with E2E keying material.

4. Example Message Flows

This section provides an example message flow to help clarify the procedures described later in this document. It is necessary that the key distributor and media distributor establish a mutually authenticated TLS connection for the purpose of sending tunneled messages, though the complete TLS handshake for the tunnel is not shown in Figure 2 since there is nothing new this document introduces with regard to those procedures.

Once the tunnel is established, it is possible for the media distributor to relay the DTLS messages between the endpoint and the key distributor. Figure 2 shows a message flow wherein the endpoint uses DTLS-SRTP to establish an association with the key distributor. In the process, the media distributor shares its supported SRTP protection profile information (see [RFC5764]) and the key distributor shares HBH keying material and selected cipher with the media distributor.

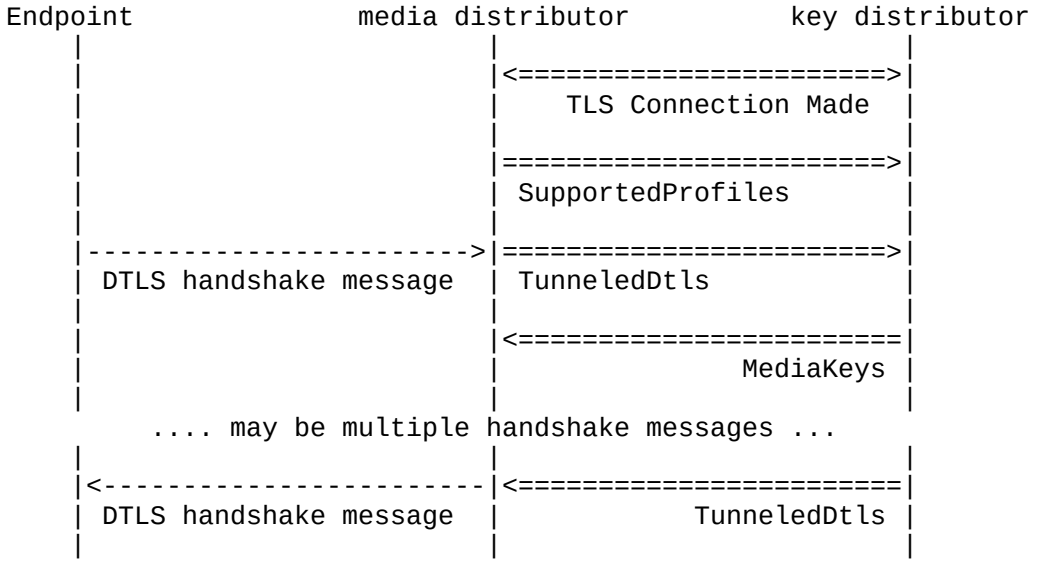

Figure 2: Sample DTLS-SRTP Exchange via the Tunnel

After the initial TLS connection has been established each of the messages on the right-hand side of Figure 2 is a tunneling protocol message as defined in Section Section 6.

SRTP protection profiles supported by the media distributor will be sent in a "SupportedProfiles" message when the TLS tunnel is initially established. The key distributor will use that information to select a common profile supported by both the endpoint and the media distributor to ensure that hop-by-hop operations can be successfully performed.

As DTLS messages are received from the endpoint by the media distributor, they are forwarded to the key distributor encapsulated inside abbrev "TunneledDtls" message. Likewise, as "TunneledDtls" messages are received by the media distributor from the key distributor, the encapsulated DTLS packet is forwarded to the endpoint.

The key distributor will provide the SRTP [RFC3711] keying material to the media distributor for HBH operations via the "MediaKeys" message. The media distributor will extract this keying material from the "MediaKeys" message when received and use it for hop-by-hop encryption and authentication.

5. Tunneling Procedures

The following sub-sections explain in detail the expected behavior of the endpoint, the media distributor, and the key distributor.

It is important to note that the tunneling protocol described in this document is not an extension to TLS [RFC5246] or DTLS [RFC6347]. Rather, it is a protocol that transports DTLS messages generated by an endpoint or key distributor as data inside of the TLS connection established between the media distributor and key distributor.

5.1. Endpoint Procedures

The endpoint follows the procedures outlined for DTLS-SRTP [RFC5764] in order to establish the cipher and keys used for encryption and authentication, with the endpoint acting as the client and the key distributor acting as the server. The endpoint does not need to be aware of the fact that DTLS messages it transmits toward the media distributor are being tunneled to the key distributor.

5.2. Tunnel Establishment Procedures

Either the media distributor or key distributor initiates the establishment of a TLS tunnel. Which entity acts as the TLS client when establishing the tunnel and what event triggers the establishment of the tunnel are outside the scope of this document. Further, how the trust relationships are established between the key distributor and media distributor are also outside the scope of this document.

A tunnel MUST be a mutually authenticated TLS connection.

The media distributor or key distributor MUST establish a tunnel prior to forwarding tunneled DTLS messages. Given the time-sensitive nature of DTLS-SRTP procedures, a tunnel SHOULD be established prior to the media distributor receiving a DTLS message from an endpoint.

A single tunnel MAY be used to relay DTLS messages between any number of endpoints and the key distributor.

A media distributor MAY have more than one tunnel established between itself and one or more key distributors. When multiple tunnels are established, which tunnel or tunnels to use to send messages for a given conference is outside the scope of this document.

5.3. Versioning Considerations

All messages for an established tunnel MUST utilize the same version value. If the version of any subsequent message differs from that of the initial message, that message MUST be discarded and the tunnel connection closed.

Since the media distributor sends the first message over the tunnel, it effectively establishes the version of the protocol to be used. If that version is not supported by the key distributor, it MUST discard the message, transmit an "UnsupportedVersion" message, and close the TLS connection.

The media distributor MUST take note of the version received in an "UnsupportedVersion" message and use that version when attempting to re-establish a failed tunnel connection. Note that it is not necessary for the media distributor to understand the newer version of the protocol to understand that the first message received is "UnsupportedVersion". The media distributor can determine from the first two octets received what the version number is and that the message is "UnsupportedVersion". The rest of the data received, if any, would be discarded and the connection closed (if not already closed).

5.4. Media Distributor Tunneling Procedures

The first message transmitted over the tunnel is the "SupportedProfiles" (see Section 6). This message informs the key distributor about which DTLS-SRTP profiles the media distributor supports. This message MUST be sent each time a new tunnel connection is established or, in the case of connection loss, when a connection is re-established.

The media distributor MUST forward all messages received from an endpoint for a given DTLS association through the same tunnel if more than one tunnel has been established between it and a key distributor.

> Editor’s Note: Do we want to have the above requirement or would we prefer to allow the media distributor to send messages over more than one tunnel to more than one key distributor? The latter would provide for higher availability, but at the cost of key distributor complexity. The former would allow the usage of a load distributor in front of the key distributor.

The media distributor MUST assign a unique association identifier for each endpoint-initiated DTLS association and include it in all messages forwarded to the key distributor. The key distributor will

subsequently include this identifier in all messages it sends so that the media distributor can map messages received via a tunnel and forward those messages to the correct endpoint. The association identifier SHOULD be randomly assigned and values not be re-used for a short period of time (e.g., five minutes) to ensure any residual state in the key distributor is clear and to ensure any packets already transmitted from the key distributor are not directed to the wrong endpoint.

The tunnel protocol enables the key distributor to separately provide HBH keying material to the media distributor for each of the individual endpoint DTLS associations, though the media distributor cannot decrypt messages between the key distributor and endpoints.

When a DTLS message is received by the media distributor from an endpoint, it forwards the UDP payload portion of that message to the key distributor encapsulated in a "TuneledDtls" message. If the media distributor knows which conference to which a given DTLS association belongs, it can pass the conference identifier to the key distributor using the "conf_id" field of the "TunneledDtls" message.

The media distributor MUST support the same list of protection profiles for the life of a given endpoint’s DTLS association, which is represented by the association identifier.

When a "MediaKeys" message is received, the media distributor MUST extract the cipher and keying material conveyed in order to subsequently perform HBH encryption and authentication operations for RTP and RTCP packets sent between it and an endpoint. Since the HBH keying material will be different for each endpoint, the media distributor uses the association identifier included by the key distributor to ensure that the HBH keying material is used with the correct endpoint.

The media distributor MUST forward all DTLS messages received from either the endpoint or the key distributor (via the "TunneledDtls" message) to ensure proper communication between those two entities.

When the media distributor detects an endpoint has disconnected or when it receives conference control messages indicating the endpoint is to be disconnected, the media distributors MUST send an "EndpointDisconnect" message with the association identifier assigned to the endpoint to the key distributor. The media distributor SHOULD take a loss of all RTP and RTCP packets as an indicator that the endpoint has disconnected. The particulars of how RTP and RTCP are to be used to detect an endpoint disconnect, such as timeout period, is not specified. The media distributor MAY use additional indicators to determine when an endpoint has disconnected.

5.5. Key Distributor Tunneling Procedures

When the media distributor relays a DTLS message from an endpoint, the media distributor will include an association identifier that is unique per endpoint-originated DTLS association. The association identifier remains constant for the life of the DTLS association. The key distributor identifies each distinct endpoint-originated DTLS association by the association identifier.

The key distributor MUST encapsulate any DTLS message it sends to an endpoint inside a "TunneledDtls" message (see Section 6).

The key distributor MUST use the same association identifier in messages sent to an endpoint as was received in messages from that endpoint. This ensures the media distributor can forward the messages to the correct endpoint.

The key distributor extracts tunneled DTLS messages from an endpoint and acts on those messages as if that endpoint had established the DTLS association directly with the key distributor. The key distributor is acting as the DTLS server and the endpoint is acting as the DTLS client. The handling of the messages and certificates is exactly the same as normal DTLS-SRTP procedures between endpoints.

The key distributor MUST send a "MediaKeys" message to the media distributor as soon as the HBH encryption key is computed and before it sends a DTLS "Finished" message to the endpoint. The "MediaKeys" message includes the selected cipher (i.e. protection profile), MKI [RFC3711] value (if any), SRTP master keys, and SRTP master salt values. The key distributor MUST use the same association identifier in the "MediaKeys" message as is used in the "TunneledDtls" messages for the given endpoint.

The key distributor, can use the certificate of the endpoint and correlate that with signaling information to know which conference this session is associated with. The key distributor informs the media distributor of which conference this session is associated by sending a globally unique conference identifier in the "conf_id" attribute of the "MediaKeys".

The key distributor MUST select a cipher that is supported by both the endpoint and the media distributor to ensure proper HBH operations.

6. Tunneling Protocol

Tunneled messages are transported via the TLS tunnel as application data between the media distributor and the key distributor. Tunnel messages are specified using the format described in [RFC5246] section 4. As in [RFC5246], all values are stored in network byte (big endian) order; the uint32 represented by the hex bytes 01 02 03 04 is equivalent to the decimal value 16909060.

The protocol defines several different messages, each of which containing the the following information:

- Protocol version
- Message type identifier
- The message body

Each of these messages is a "TunnelMessage" in the syntax, with a message type indicating the actual content of the message body.

6.1. Tunnel Message Format

The syntax of the protocol is defined below. "TunnelMessage" defines the structure of all messages sent via the tunnel protocol. That structure includes a field called "msg_type" that identifies the specific type of message contained within "TunnelMessage".

```
enum {
    unsupported_version(1),
    supported_profiles(2),
    media_keys(3),
    tunneled_dtls(4),
    endpoint_disconnect(5),
    (255)
} MsgType;

struct {
    uint8 version;
    MsgType msg_type;
    select (MsgType) {
        case unsupported_version: UnsupportedVersion;
        case supported_profiles:  SupportedProfiles;
        case media_keys:          MediaKeys;
        case tunneled_dtls:       TunneledDtls;
        case endpoint_disconnect: EndpointDisconnect;
  } body;
} TunnelMessage;
```

The elements of "TunnelMessage" include:

- version: indicates the version of this protocol (0x00).
- msg_type: the type of message contained within the structure "body".

The "UnsupportedVersion" message is defined as follows:

```
struct { } UnsupportedVersion;
```

The "UnsupportedVersion" message does not convey any additional information in the body.

The "SupportedProfiles" message is defined as:

```
uint8 SRTPProtectionProfile[2]; // from RFC5764

struct {
  SRTPProtectionProfile protection_profiles<0..2^16-1>;
} SupportedProfiles;
```

This message contains this single element: * protection_profiles: The list of two-octet SRTP protection profile values as per [RFC5764] supported by the media distributor.

The "MediaKeys" message is defined as:

```
struct {
    uint32 association_id;
    SRTPProtectionProfile protection_profile;
    opaque mki<0..255>;
    opaque client_write_SRTP_master_key<1..255>;
    opaque server_write_SRTP_master_key<1..255>;
    opaque client_write_SRTP_master_salt<1..255>;
    opaque server_write_SRTP_master_salt<1..255>;
    opaque conf_id<0..255>;
} MediaKeys;
```

The fields are described as follows:

- association_id: A value that identifies a distinct DTLS association between an endpoint and the key distributor.
- protection_profiles: The value of the two-octet SRTP protection profile value as per [RFC5764] used for this DTLS association.
- mki: Master key identifier [RFC3711].
- client_write_SRTP_master_key: The value of the SRTP master key used by the client (endpoint).
- server_write_SRTP_master_key: The value of the SRTP master key used by the server (media distributor).

- client_write_SRTP_master_salt: The value of the SRTP master salt used by the client (endpoint).
- server_write_SRTP_master_salt: The value of the SRTP master salt used by the server (media distributor).
- conf_id: Identifier that uniquely specifies which conference the media distributor should place this media flow in.

The "TunneledDtls" message is defined as:

```
struct {
    uint32 association_id;
    opaque conf_id<0..255>;
    opaque dtls_message<0..2^16-1>;
} TunneledDtls;
```

The fields are described as follows:

- association_id: An value that identifies a distinct DTLS association between an endpoint and the key distributor.
- conf_id: Optional identifier that uniquely specifies which conference this media flow is in.
- dtls_message: the content of the DTLS message received by the endpoint or to be sent to the endpoint.

The "EndpointDisconect" message is defined as:

```
struct {
    uint32 association_id;
} EndpointDisconnect;
```

The fields are described as follows:

- association_id: An value that identifies a distinct DTLS association between an endpoint and the key distributor.

7. Example Binary Encoding

The "TunnelMessage" is encoded in binary following the procedures specified in [![RFC5246]]. This section provides an example of what the bits on the wire would look like for the "SupportedProfiles" message that advertises support for both SRTP_AEAD_AES_128_GCM and SRTP_AEAD_AES_256_GCM [RFC7714].

```
TunnelMessage:
              version: 0x00
         message_type: 0x01
    SupportedProfiles:
       protection_profiles:  0x0004 (length)
                             0x00070008 (value)
```

Thus, the encoding on the wire presented here in network bytes order would be this stream of octets:

0x0001000400070008

8. IANA Considerations

This document establishes a new registry to contain message type values used in the DTLS Tunnel protocol. These data type values are a single octet in length. This document defines the values shown in Table 1 below, leaving the balance of possible values reserved for future specifications:

MsgType	Description
0x01	Unsupported Version
0x02	Supported SRTP Protection Profiles
0x03	Media Keys
0x04	Tunneled DTLS
0x05	Endpoint Disconnect

Table 1: Data Type Values for the DTLS Tunnel Protocol

The value 0x00 and all values in the range 0x06 to 0xFF are reserved.

The name for this registry is "Datagram Transport Layer Security (DTLS) Tunnel Protocol Data Types for Privacy Enhanced Conferencing".

9. Security Considerations

The encapsulated data is protected by the TLS connection from the endpoint to key distributor, and the media distributor is merely an on path entity. The media distributor does not have access to the end-to-end keying material This does not introduce any additional security concerns beyond a normal DTLS-SRTP association.

The HBH keying material is protected by the mutual authenticated TLS connection between the media distributor and key distributor. The key distributor MUST ensure that it only forms associations with

authorized media distributors or it could hand HBH keying material to untrusted parties.

The supported profiles information sent from the media distributor to the key distributor is not particularly sensitive as it only provides the cryptographic algorithms supported by the media distributor. Further, it is still protected by the TLS connection between the media distributor and the key distributor.

10. Acknowledgments

The author would like to thank David Benham and Cullen Jennings for reviewing this document and providing constructive comments.

11. References

11.1. Normative References

[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>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[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>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[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>.

[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>.

11.2. Informative References

[RFC7714] McGrew, D. and K. Igoe, "AES-GCM Authenticated Encryption in the Secure Real-time Transport Protocol (SRTP)", RFC 7714, DOI 10.17487/RFC7714, December 2015, <http://www.rfc-editor.org/info/rfc7714>.

Authors' Addresses

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

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

Paul M. Ellenbogen
Princeton University

Phone: +1 206 851 2069
Email: pe5@cs.princeton.edu

Nils H. Ohlmeier
Mozilla

Phone: +1 408 659 6457
Email: nils@ohlmeier.org