draft-jones-perc-dtls-tunnel-03

DTLS Tunnel Protocol

Facilitating SRTP and EKT Key Exchange via the Media Distributor

Paul E. Jones

IETF 96 ● Berlin

July 22, 2016

What is New in -03

- Modified the protocol to reduce the message size
- Added text to discuss PMTU issues as requested
- IANA considerations for the “data type” values carried in the protocol
- Changed terminology
  - KMF => Key Distributor
  - MDD => Media Distributor

Concerns with Current Draft

- Using DTLS between Key Distributor and Media Distributor presents NAT/FW traversal challenges
- Current protocol is very limited in terms of extensibility
- Current protocol is tightly coupled to specific DTLS messages that will change in DTLS 1.3

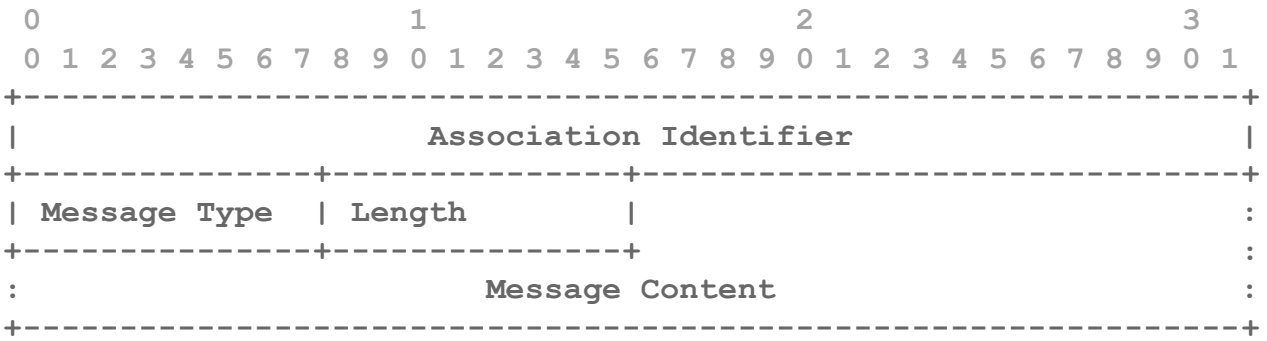
A **TLS** or **WebRTC Data Channel** could address the above concerns.

Let’s decide as a group which direction we should take. See the next few slides outlining some alternatives.

Alternative 1: TLS from Key Distributor

- Simple interface
- Allows for more flexibility in the future (new messages, etc.)
- Protocol can be a trivial TLV protocol (next slide)
  - Note that “AssociationID == 0” could be reserved to indicate information that applies to all DTLS associations (e.g., supported protection profiles)

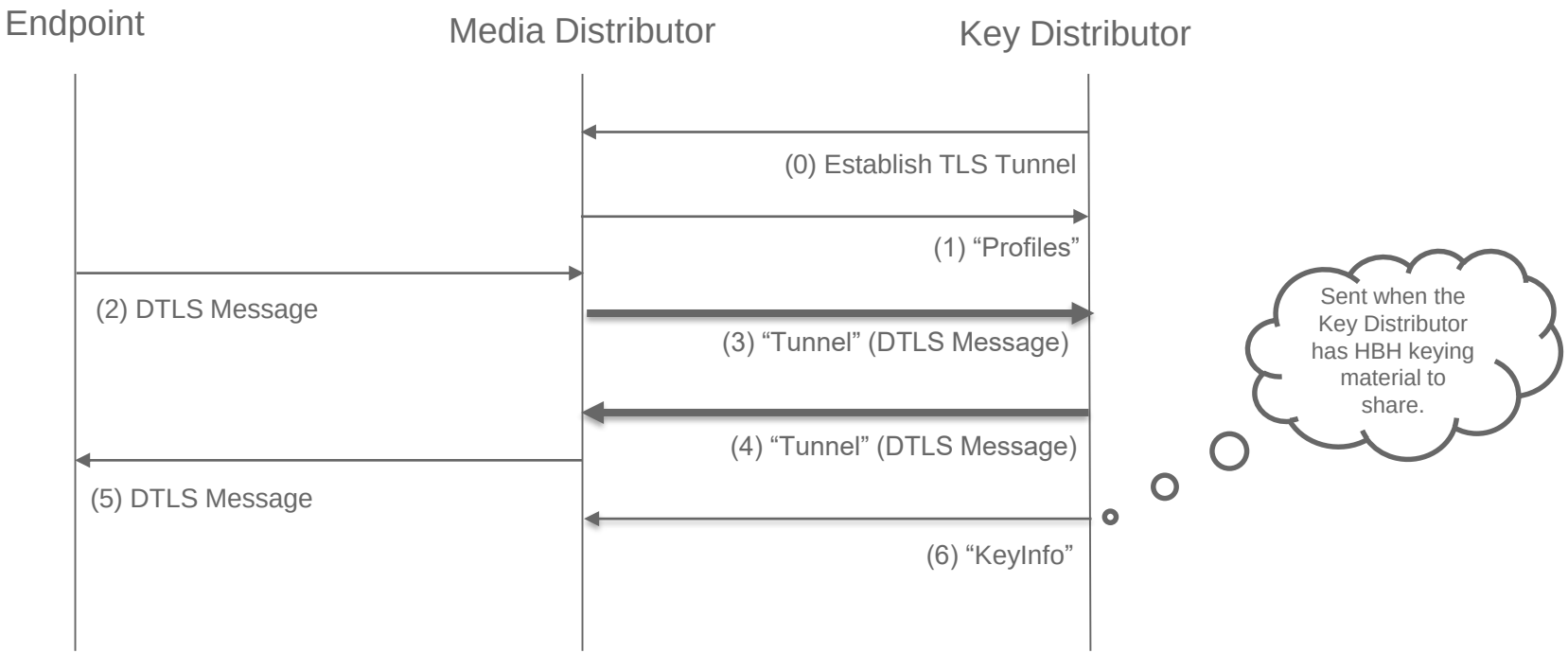
Possible Protocol Design over TLS



Simple Tag/Length/Value Protocol

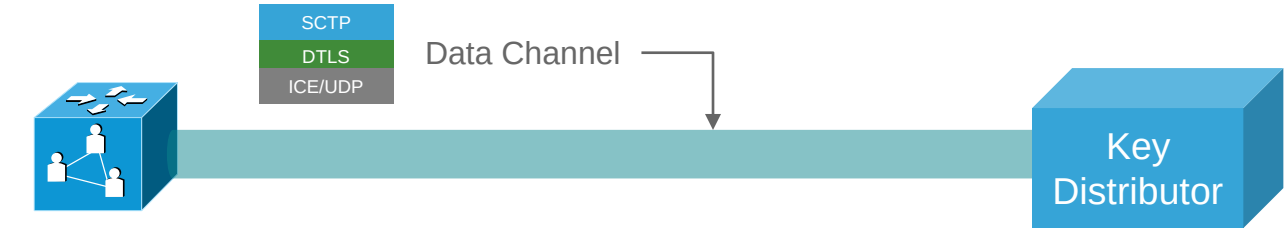
Message Type	Message Content
0x01	Tunneled DTLS message
0x02	Supported DTLS-SRTP profiles (from Media Distributor)
0x03	Key Information (from Key Distributor)

Sample Message Flow



Alternative 2: WebRTC Data Channel

- Endpoint procedures remain unchanged: standard DTLS-SRTP
- Interface from the Key Distributor to the Media Distributor changes



- Browsers already have data channels
- Single SCTP association established
- SCTP Stream IDs can replace the “association identifier”
- Allows for more flexibility in the future (new messages, etc.)
- Protocol can be a trivial TLV protocol

Several Sub-Options for Data Channels

2A) *n* Reliable SCTP steams

- Reliable, in-order message delivery (ensure keys arrive on time)
- “Tunnel” packets sent as “WebRTC Binary” data (PPID 53)
- Association ID is replaced with the SCTP stream ID (see next slide)

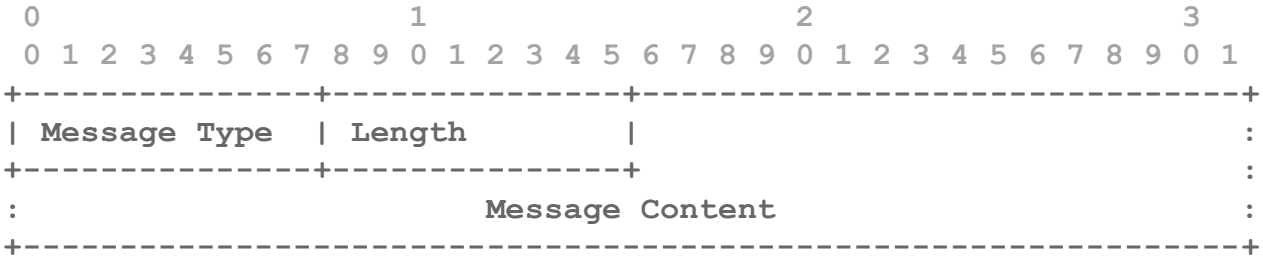
2B) *n* Unreliable Steams + 1 reliable “control” stream

- Relayed DTLS packets sent over unreliable stream with no additional encapsulation and using PPID 53
- Supported profiles and key information sent over a reliable stream, with the session ID in the control protocol packets and using PPID 53
- Association ID replaced with Stream ID, with the control channel protocol looking like the TLS example

2C) One Reliable Stream

- Reliable, in-order message delivery (ensure keys arrive on time)
- Messages encapsulated in a protocol similar the example shown for TLS

Possible Protocol Design with Data Channel (2A)



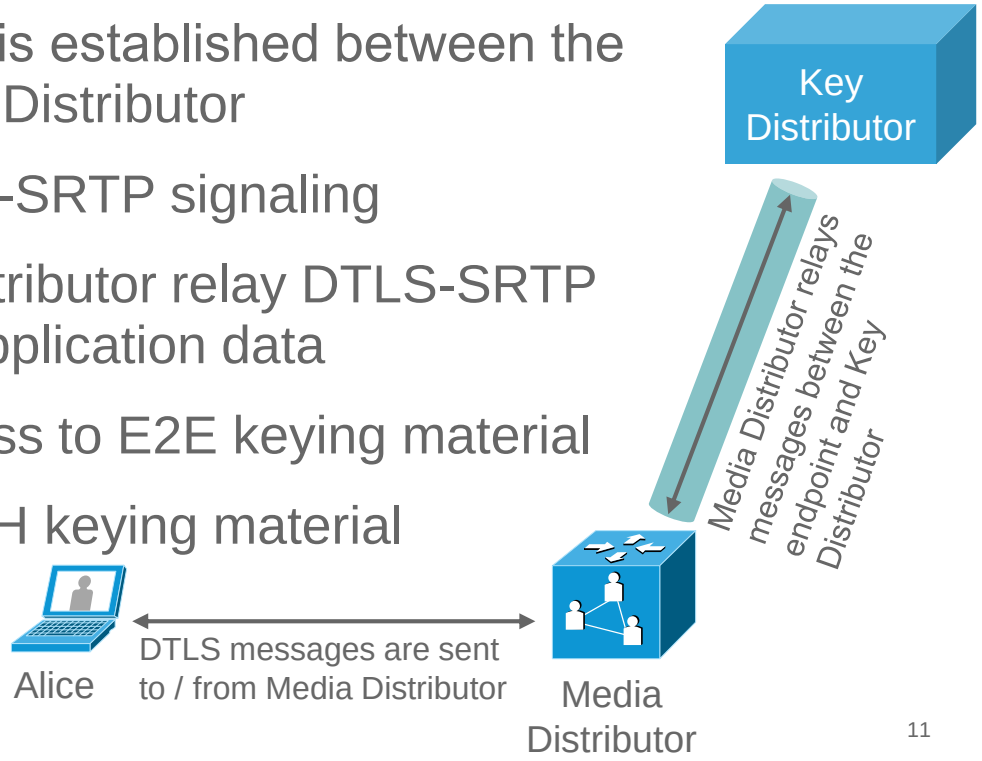
Simple Tag/Length/Value Protocol

Message Type	Message Content
0x01	Tunneled DTLS message
0x02	Supported DTLS-SRTP profiles (from Media Distributor)
0x03	Key Information (from Key Distributor)

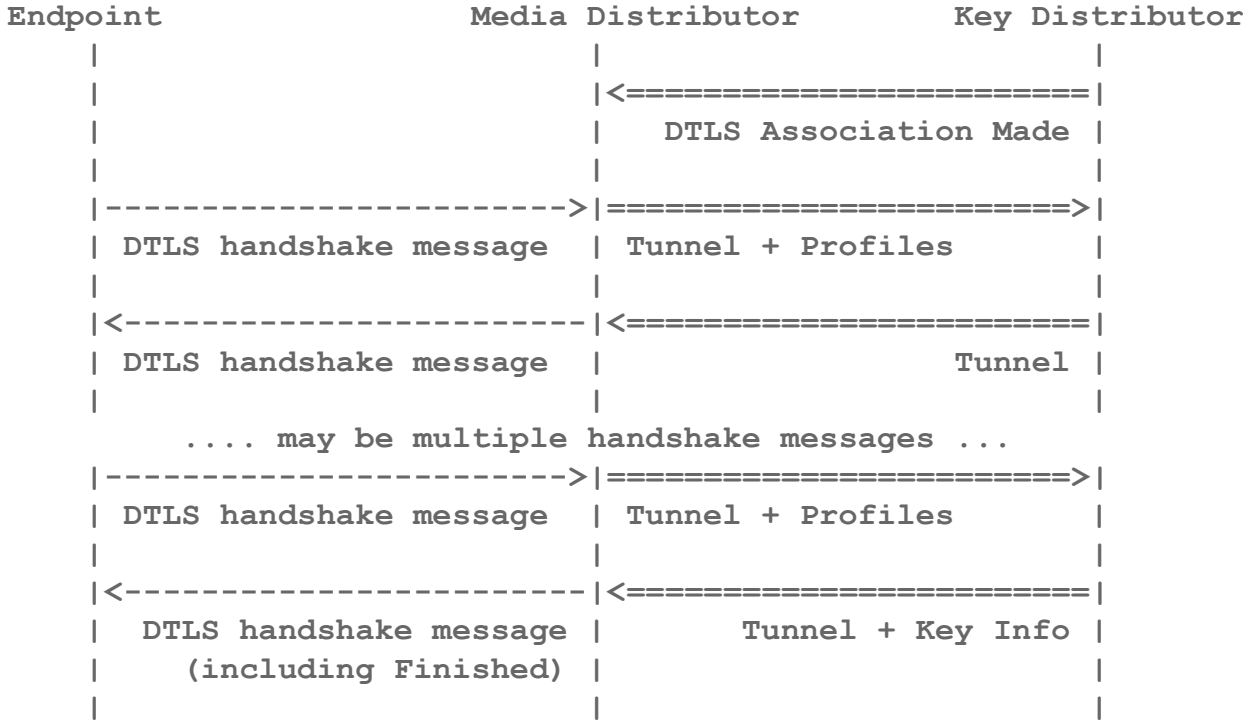
Backup Slides

DTLS Tunnel Overview

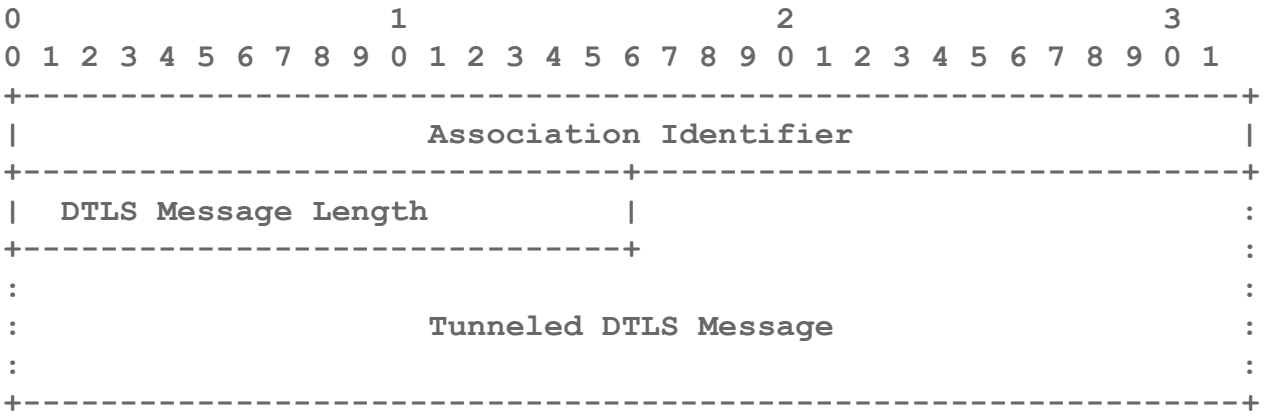
- A DTLS association (“tunnel”) is established between the Key Distributor and the Media Distributor
- Endpoints use standard DTLS-SRTP signaling
- Media Distributor and Key Distributor relay DTLS-SRTP through the DTLS tunnel as application data
- Media Distributor has no access to E2E keying material
- Media Distributor receives HBH keying material



Message Exchange via the Tunnel

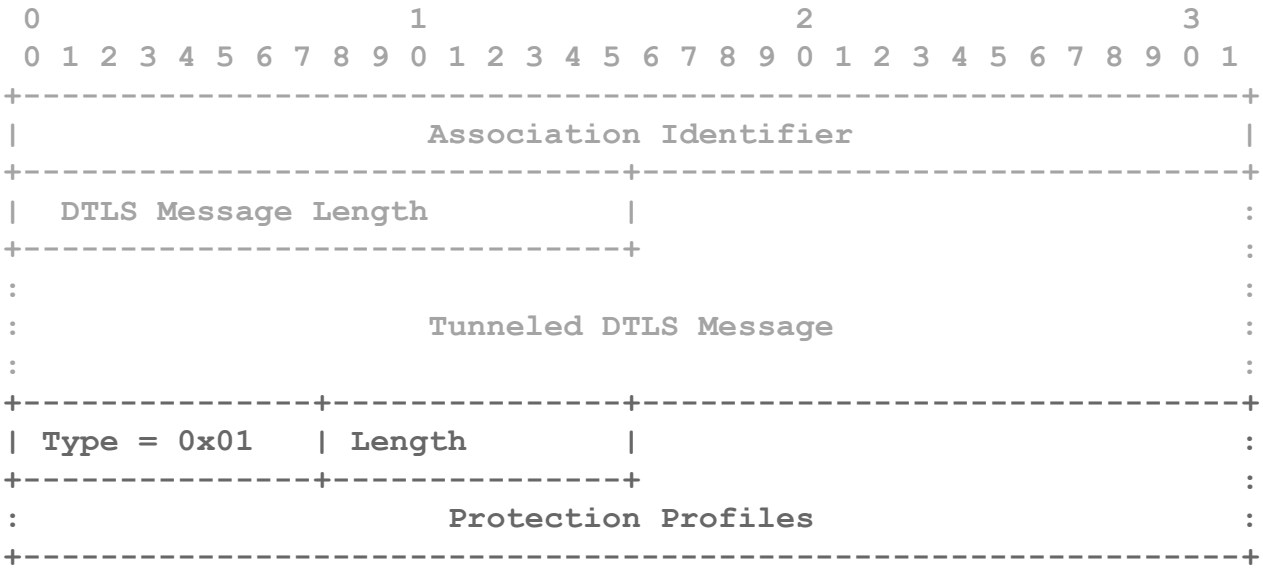


Tunnel Message Format



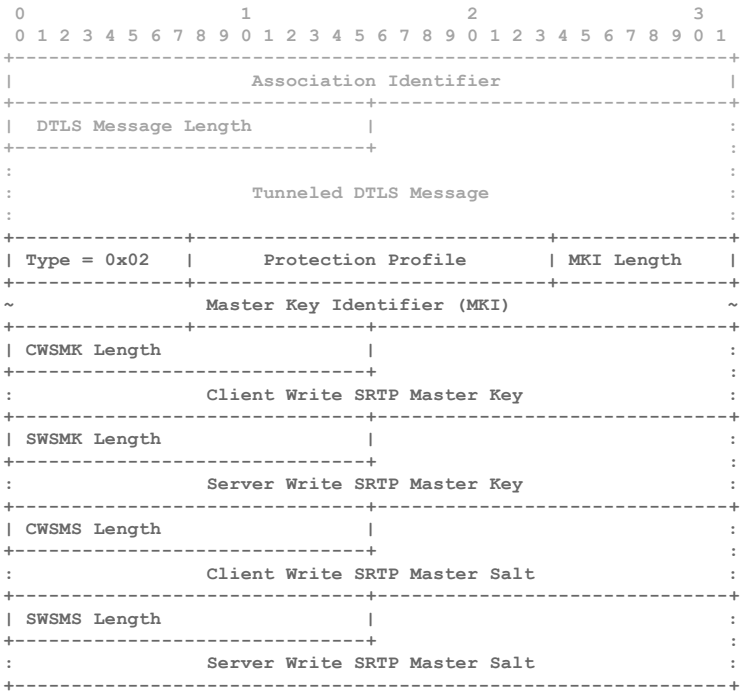
This is usually what is used between the Key Distributor and the Media Distributor and this forms the basic header for all messages

Messages from Media to Key Distributor



Media Distributor always advertises its list of supported DTLS-SRTP protection profiles

Tunnel + Key Info Message



This message is sent to from the Key Distributor to the Media Distributor when the DTLS Finished message is sent to the endpoint. This provided the Media Distributor with the cipher and key information selected for HBH operations.


CISCO