PERC EKT Diet
draft-jennings-perc-srtp-ekt-diet

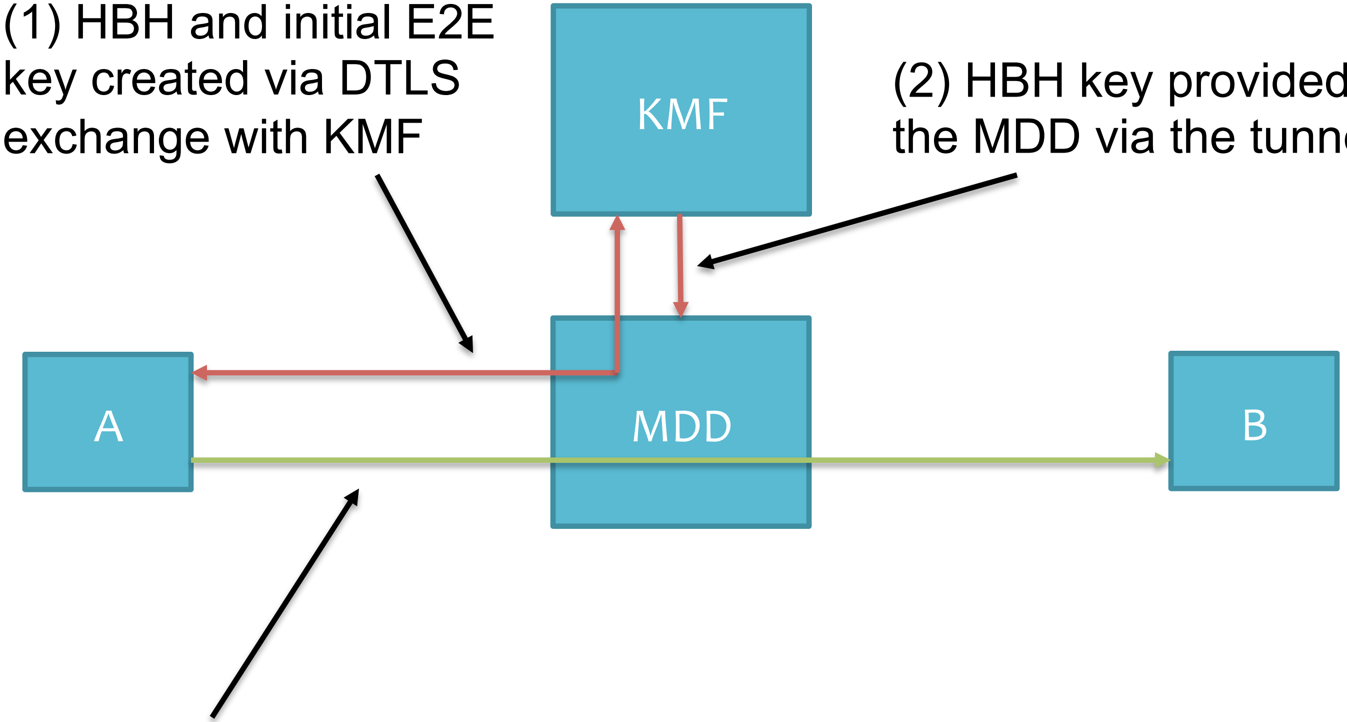
Cullen Jennings <fluffy@iii.ca>

April 4, 2016

This is the short version

- This draft is just a cut down version of draft-ietf-avtcore-srtp-ekt-03 to help discussion about the key parts of EKT for the PERC
- EKT as specified by various IETF drafts has been implemented and shipping for years but we need to finish this

Keys used in PERC



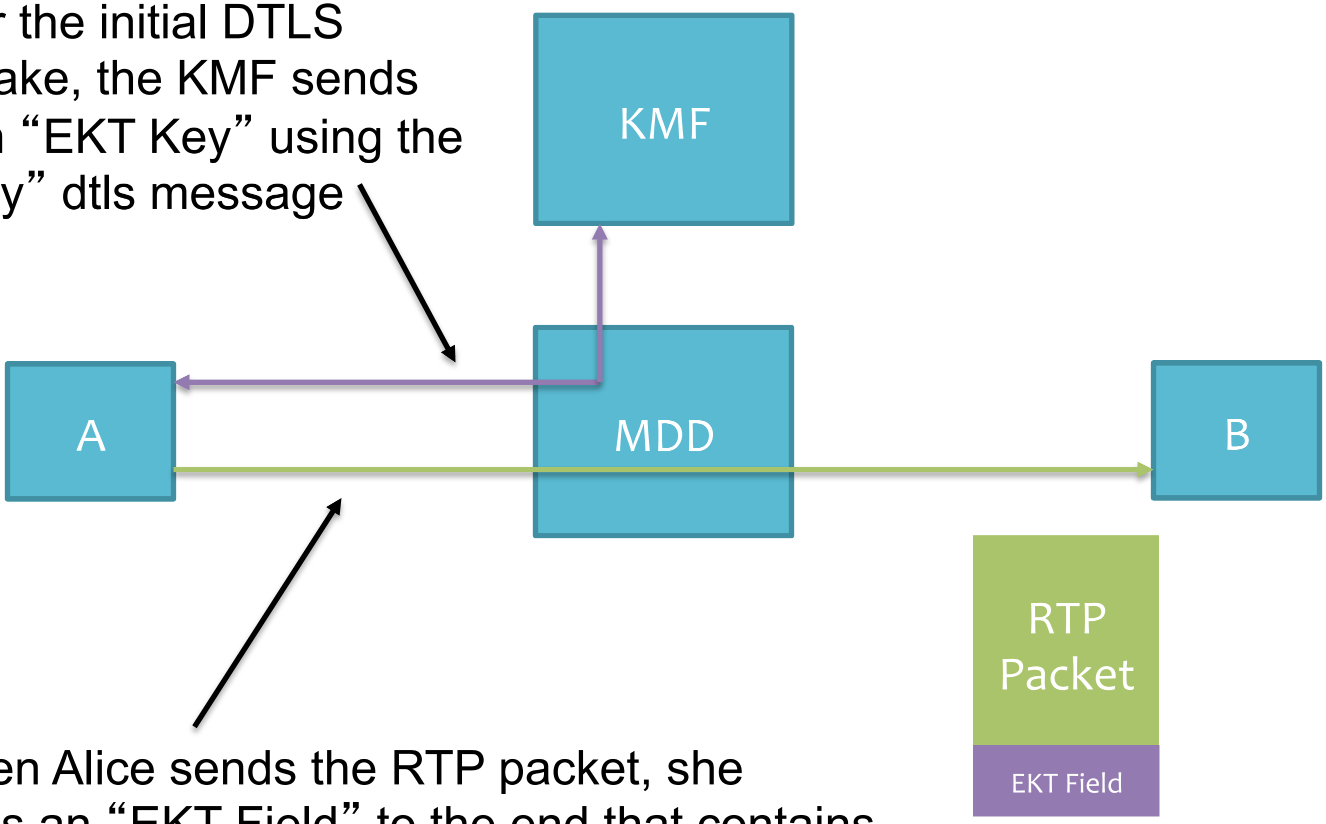
(1) HBH and initial E2E key created via DTLS exchange with KMF

(2) HBH key provided to the MDD via the tunnel

(3) A encrypts a packet using the E2E key and sends it to B, but how is he able to decrypt it? This is where EKT comes into play.

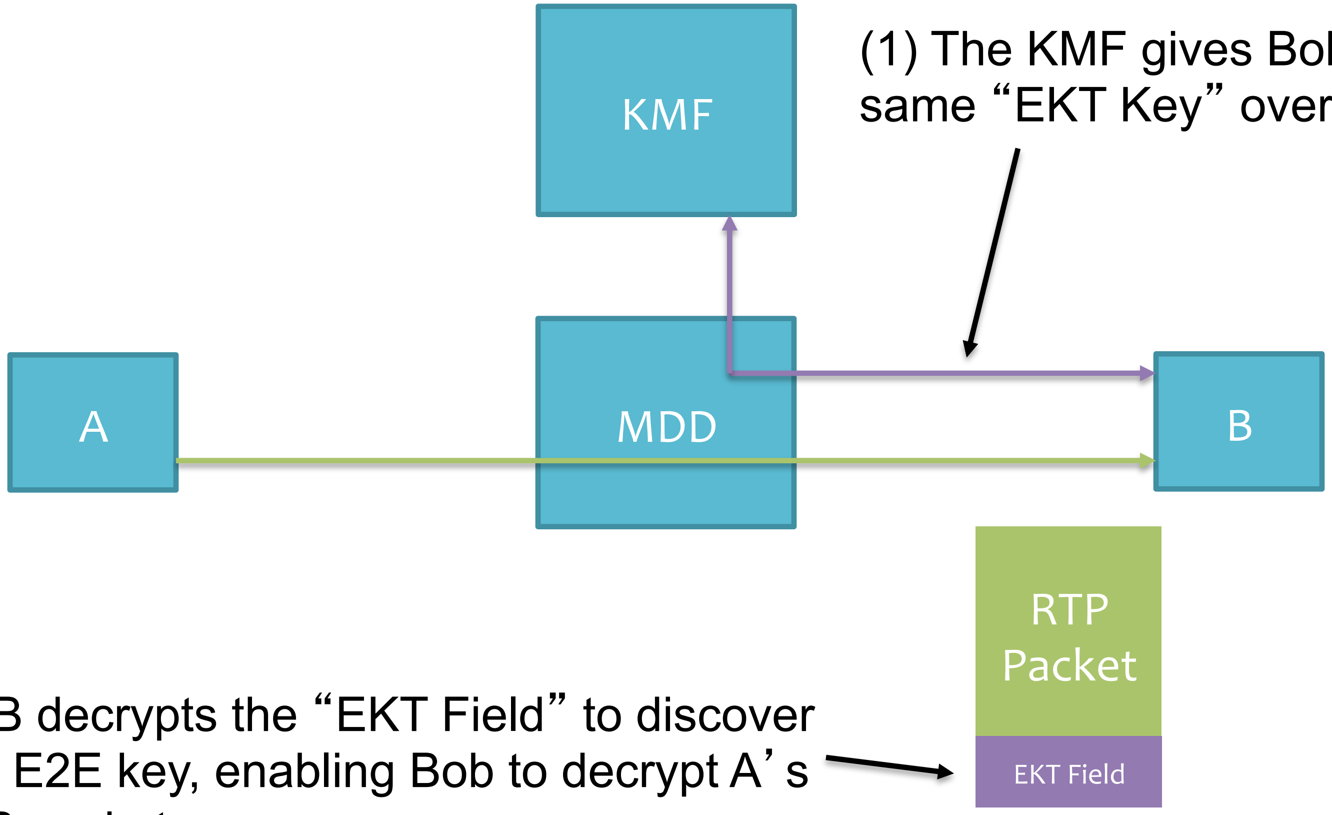
Media Sender

(1) After the initial DTLS handshake, the KMF sends Alice an “EKT Key” using the “ekt_key” dtls message



(2) When Alice sends the RTP packet, she attaches an “EKT Field” to the end that contains the E2E key, encrypted using the “EKT Key”

Media Receiver Processing



(1) The KMF gives Bob the same “EKT Key” over DTLS

(2) B decrypts the “EKT Field” to discover A’s E2E key, enabling Bob to decrypt A’s RTP packet.

Why EKT?

- Simple solution for an endpoint to securely convey its E2E key to all conference participants
- KMF can re-key the conference by sending a new EKT key to all participants
- An endpoint can autonomously change its E2E key and transmit it to everyone else

Obtaining EKT Key

- KMF sends following data to the endpoint (in the “ekt_key” dtls message):
  - EKT Cipher
  - EKT Key Value
  - EKT Master Salt
  - SPI to uniquely identify the EKT Key

EKT Field added to SRTP Packet

- The sender encrypts the SSRC, ROC, and SRTP Master Key ands sends this along with unencrypted SPI
- Receiver uses the SPI to know which EKT Key to use then decrypts the rest of the information
- The SRTP Master Salt that was received with the EKT Key along with ROC, SSRC, and SRTP Master Key are then used for decryption of SRTP from that SSRC

  – Note mistake editing the draft accidentally removed how to compute the EKT Ciphertext. This will be added back.

Issue - Extensibility

- Right now we have 1 bit to tell the types of EKT message in SRTP.
- Over time we have used multiple different data formats in deployed versions
- Proposal:
  - Move to 1 byte message type with IANA registry
  - Include a length of EKT Field

Issue – Draft Organization

- Split EKT draft into:
  - EKT Base Spec
  - EKT in MIKEY
  - EKT in Security Descriptions
- Move EKT Base Spec draft to PERC WG?

Issue: Crypto sizes

- Have AES-128 and AES-256 and ability to extend in future
- Do we need AES-192 now?

Issue: Key Transition Timing

- Problem
  - Imagine a participants leaves the conference and the KMF rekeys. One of the DTLS message is lost and DTLS need to retransmitted. It can be a few hundred milliseconds before all participants have the new EKT Key
- Proposal
  - say don't use new EKT Key until 250ms after receiving it

Open Issues – Names

- Changes names across PERC docs
  - EKT Key -> ???
  - EKT Field -> ???
  - SPI -> ???

Issue: SPI (Aka EKT Key ID)

- Currently just unique 15 bit ID
- Propose that we say it monotonically increases so that it is very clear which key is older or newer

Issue: Special EKT Key for announcements?

- Marking a EKT Key as used by send only announcement server
- Allow this EKT Key to be used for duration of session
- Add a new short version of EKT Field that provides just the ID of the EKT Key to use