A Solution Framework for Private Media in Privacy Enhanced RTP Conferencing

(draft-jones-perc-private-media-framework-00)

IETF 93 / July 2015

Paul E. Jones • Nermeen Ismail • David Benham

Cisco

Agenda

- Security Objectives
  - Hop-By-Hop
  - End-to-End
- Key Exchange
- Sending a Packet
- Entities with Keys
- Design Considerations
  - Changes to EKT
  - KMF Trust and Congestion Management
  - Cryptographic Context & RTP Header Values

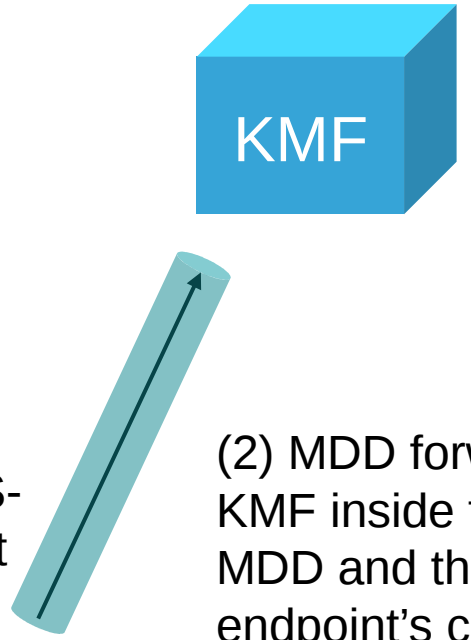
Security Objectives – Hop-by-Hop

- Use SRTP procedures and key shared between only two adjacent entities to perform all hop-by-hop operations:
  - Authentication of the RTP and RTCP packets
  - Optional hop-by-hop encryption of RTP header extensions
  - Optional hop-by-hop encryption of RTCP packets

Security Objectives – End-to-End

- Use an EKT key known to all conference participants to facilitate end-to-end authenticated encryption of media content (i.e., audio & video)
- Why EKT?
  - Provides a single, shared EKT Key known to all conference participants.
  - Provides quick cryptographic context synchronization (avoid ROC-guessing) when a new media flow is forwarded which had not been seen for a long time by receiver(s), such as when participant was previously silent.
  - Avoids need to re-key a conference when an SSRC collision occurs, addressing the “two time pad” issue described in Section 9.1/RFC 3711 (every sender will have a distinct SRTP master key for end-to-end encryption).

PERC Key Exchange



(0) DTLS tunnel for encapsulating DTLS-SRTP from endpoint

(2) MDD forwards DTLS messages to the KMF inside the DTLS tunnel between the MDD and the KMF. KMF validates the endpoint's certificate.

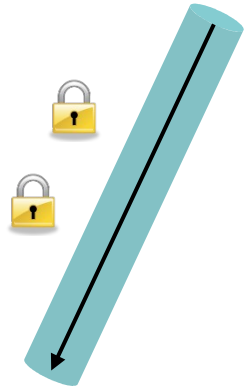
Alice

(1) Establish DTLS connection with KMF

MDD

Bob

PERC Key Exchange (continued)



(3) KMF sends the EKT Key to the endpoint via the tunnel. The tunneled DTLS info cannot be decrypted by the MDD. As a part of the tunneling protocol, the hop-by-hop SRTP key is given to the MDD.

(5) Hop-by-hop key computed. Media key created.

Alice

(4) MDD forwards DTLS traffic. KMF certificate verified, EKT Key delivered.

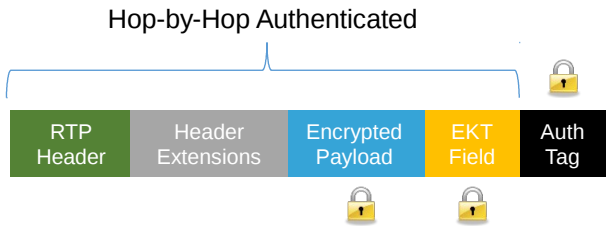
MDD

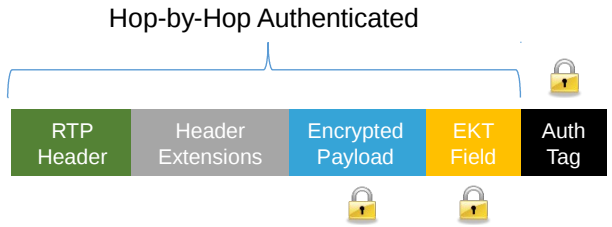
The MDD holds the hop-by-hop key

Bob

Sending a Packet

(7) Packets are authenticated and replay protection enforced. The MDD is permitted to alter RTP header extensions, payload type values, and other limited info. When forwarding the packet, the next per-hop key is used to authenticate the packet.





Alice

(6) Media is transmitted into the conference, encrypted with end-to-end key, authenticated with hop-by-hop key

MDD

(8) Media is forwarded to other participants, which also authenticate received packets. Media is then decrypted and rendered.

Bob

What Entity Has Access to What Key Material?

	Endpoint A	MDD A	MDD B	Endpoint B
End-to-End EKT Key and Salt	Yes	No	No	Yes
Hop-by-Hop Key (A⇔ MDD A)	Yes	Yes	No	No
Hop-by-Hop Key (MDD A ⇔ MDD B)	No	Yes	Yes	No
Hop-by-Hop Key (B⇔ MDD B)	No	No	Yes	Yes

An “endpoint” might be an end-user terminal device, gateway, or other entity that is trusted to join the conference and participate in end-to-end media

Design Considerations to Discuss – Changes to EKT Proposed

- ROC is transmitted as plaintext in the EKT Tag
- We need a mechanism to negotiate SRTP Protection Profiles for the end-to-end encryption/authentication
  - DTLS-SRTP is proposed for hop-by-hop, but need something for end-to-end and adding something to the EKT message exchanges might be best approach
- A solution to the ISN / MKI security issue John Mattsson raised during IETF 92
  - Propose we do away with MKI
- Do we have the flexibility in PERC to do this work?

Design Considerations to Discuss –
KMF Trust and Congestion Management Out of Scope

- This Framework document focuses only on media security, leaving KMF-to-endpoint trust establishment to be dealt with separately

- This Framework document leaves any PERC impacts on congestion management in RTP middleboxes to be dealt with separately
  - An AVT design team is exploring this area and making contributions around frame markings in extension headers, etc.

Design Considerations to Discuss – Cryptographic Context & RTP Header Values

- This Framework document adds an end-to-end key, salt, and AEAD cipher suite to the SRTP cryptographic context at transmitter
  - SRTP cryptographic context uses SSRC, sequence number, and ROC
  - Thus, all of the above must be conveyed to receiver in order to decrypt

- Options to convey cryptographic context to receiver:
  - Forward with those values in the RTP header unchanged, the default in this doc
  - If RTP middle box must re-write any of those values in the RTP header, it must copy the original values to elsewhere in packet, which we would need to specify (e.g., RTP header extension or payload appendage)

…form a design team to discuss further?

THANK YOU!