PERC: Double + EKT

IETF 99, July 2017, Prague

- Cullen & Sergio
Agenda

One broad open issue on how to deal with repair like packets

  Specifically, RTX, FlexFEC, and RED

Presentation will look at range of options for each

  Options looked at includes proposal from the lite draft

Exciting and awesome joint proposal from Sergio, Cullen, Emil, & Alex that none of us like and all of us can live with (ietf bumpy consensus)

But first ... talk about the facts everyone agrees on about what these proposal are ..
What is “lite”? 

- Lite is an approach where SRTP is tunneled inside SRTP
- The E2E SRTP is keyed by the JavaScript in the same way SRTP with SDES would be keyed
- The HBH DTLS-SRTP is keyed by normal DTLS without use of DTLS EKT

Repair operations are done before HBH DTLS-SRTP but after the E2E SRTP (details on later slides)
FlexFEC
FlexFEC Outside of PERC

Endpoint (RTP Sender)
FlexFEC Proposals for PERC

A

Unencrypted RTP packet → SRTP-Double → Cache → Repair → SRTP (HBH)

B

Unencrypted RTP packet → SRTP-Double → Cache → Repair → SRTP Double

Lite

Unencrypted RTP packet → SRTP E2E → SRTP HBH1 → Cache → Repair → SRTP (HBH2)

Media Distributor can not do repair
**Endpoint (RTP Sender)**

**RTX A (Same as FEC A)**

- Unencrypted RTP packet
- SRTP-Double(1)
- Cache
- RTX
- Single HBH2

**RTX B**

- Unencrypted RTP packet
- SRTP-Double(1)
- Insert new Header Extension with OSN
- Cache
- RTX + OSN Ext
- SRTP-Double(2)

**RTX - Lite**

- Unencrypted RTP packet
- SRTP E2E
- SRTP HBH1
- Cache
- RTX
- SRTP-HBH2

**RTX**

- Media Distributor can not do repair
RED
RED - Overview

- Each packet contains an alternative version of the previous packet:

Above from IETF 37 in 1996, Thank you Colin
Redundant Encoding (RED) - Use-cases

1. Redundant Encoding (RED) at EndPoint

2. Primary Stream (H0, then H1, ...)

3. Redundant Encoding (RED) (H1, L0)
Redundant Encoding (RED)

**RED A (Same as FEC A)**

RTP Packet @ T1 → SRTP-Double(1) → Cache-T1 → RED → SRTP-HBH

- F=1 |PT0 | TimeStamp-0 | double(Payload-0...) |
- F=0 |PT1 | double(Payload-1 .........) |

**RED B**

RTP Packet @ T1 → Cache-T1 → RED → SRTP-Double

- F=1 |PT0 | TimeStamp-0 | Payload-0... |
- F=0 |PT1 | Payload-1 ......... |

**RED Lite**

RTP Packet @ T1 → SRTP-E2E → Cache-T1 → RED → SRTP-HBH

- F=1 |PT0 | TimeStamp-0 | E2E(Payload-0...) |
- F=0 |PT1 | E2E(Payload-1 .........) |

Media Distributor can not read things inside RED packet.
Proposal
Sergio Proposal to move OHB to Payload

- Take the content of OHB and move it to a block at the start of the payload
- Replace length with single byte bitfield indicating which “original” fields follow in payload.
  - Bit 0: PT
  - Bit 1: Seq No
  - Bit 2: M flag
  - Bit 3..6: future extensions
  - Bit 7: reserved for getting more bits
- Add a bit that is set for a field with counter of number of E2E protected header fields
- Have the EKT Message in the DTLS from the Key Distributor tell the client the value of the bitfield (as conveyed by the Media Distributor)
Proposal (part 2 of 2)

- Move the OHB information from header extension to payload (see previous slide)
- RTX, RED, and FlexFEC ordering: use the ordering described as “A” in this draft
- DTMF: Do not support the Media Distributor being able to receive DTMF (No change to current drafts)
EKT
EKT Issues

No open issues.

Add in to DTLS EKT message, the value of the bitfield value for the new Double Payload if we make the OHB -> Payload change
Backup
Before Enc

WIRE

pt

Repair

HBH

E2E

MEDIA

After Enc

WIRE

pt

Repair

HBH

E2E

MEDIA

Lite

WIRE

pt

Repair

HBH

E2E

MEDIA

- State of the art [Jennings]
- s-flag for E2E vs. HBH
- Keeps unitary transform, with different output depending on s-flag
- “Triple” encryption

- Breaks apart transform, since repair operation has to operate on intermediate product

- State of the art [FlexFEC?]?
- Keeps unitary transform, no need for s-flag

Plaintext

SRTP Transform

Repair transform (RTX / FEC)