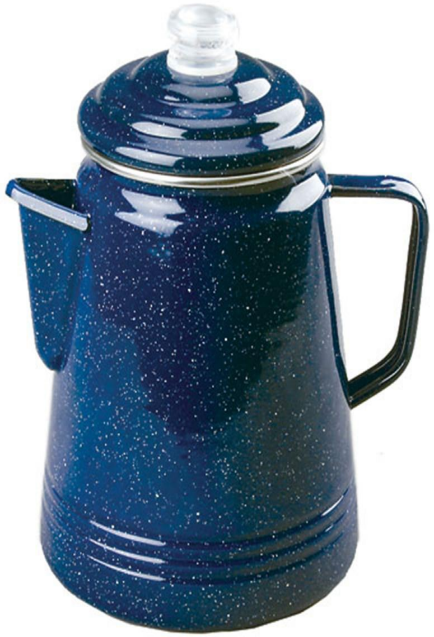

PERC Interim
5/24/2017

Note well!

Any submission to the IETF intended by the Contributor for publication as all or part of an IETF Internet-Draft or RFC and any statement made within the context of an IETF activity is considered an "IETF Contribution". Such statements include oral statements in IETF sessions, as well as written and electronic communications made at any time or place, which are addressed to:

- The IETF plenary session
- The IESG, or any member thereof on behalf of the IESG
- Any IETF mailing list, including the IETF list itself, any working group or design team list, or any other list functioning under IETF auspices
- Any IETF working group or portion thereof
- The IAB or any member thereof on behalf of the IAB
- The RFC Editor or the Internet-Drafts function

All IETF Contributions are subject to the rules of RFC 5378 and RFC 3979 (updated by RFC 4879).

Statements made outside of an IETF session, mailing list or other function, that are clearly not intended to be input to an IETF activity, group or function, are not IETF Contributions in the context of this notice.

Please consult RFC 5378 and RFC 3979 for details.

A participant in any IETF activity is deemed to accept all IETF rules of process, as documented in Best Current Practices RFCs and IESG Statements.

A participant in any IETF activity acknowledges that written, audio and video records of meetings may be made and may be available to the public.

Adopted Drafts - Status Quo

Draft	Status
Double	One pending technical issue, **discussed today**
EKT	No major updates ***since April 2016***
Tunnel	No major updates ***since October 2016***
Framework	No major updates **since May 2016**

Milestones

Jun 2017 - Submit architecture or framework specification to IESG

Jun 2017 - Submit SRTP protocol extension specification to IESG

Jun 2017 - Submit Key-management protocol specification to IESG

Almost there !! Let’s see if we can make this happen

Jun 2017 - Submit documentation of how to integrate solution in SIP, WebRTC and CLUE to IESG

Recap .. since ietf 98

New versions submitted - double, ekt, tunnel

PERC Keying Documentation - Cullen's video and slides. (Framework to adopt text)

Open issue on RTX/FEC/DTMF support in double - topic for today (next slide)

Today

Problem Statement - HBH Only Crypto Scope in Double

MDD needs HBH only control for certain RTP usages

- RTX/FEC/RTCP/DTMF ...

Clients needs to specify HBH and E2E crypto scopes explicitly

Off to Cullen...

PERC Double for RTX, FEC, DTMF, & RTCP

May 2017

Cullen Jennings

Problem to Solve

Some RTP/RTCP packets should not have E2E security:

1. Draft currently sends RTCP only HbH protected
    - Media Distributor needs to be able to read and write RTCP payloads
2. Packet repair such as RTX and FEC can be handled by Media Distributor
    - Media Distributor needs to be able to read RTX and FEC payload
3. Some systems might want Media Distributor to see DTMF for things like mute
    - Media Distributor needs to be able to read DTMF payload

The RTX Problem

Repairing streams at each hop is generally better than end to end repair as the errors don’t accumulate reducing the probability of repair.

Not all Media Distributor will want to bother with repair so it can also be done end to end.

The Media Distributor can notice a packet is missing and request a retransmission. The repair packet will have the original sequence number in the payload which will be E2E encrypted. Very sad that it was not in the RTP headers instead.

How RTX Works

Consider RTC with SRTP not using double

> RTX forms the payload by taking the encrypted payload of the original packet and prepending the the original sequence number to form a new payload
>
> This new payload is now encrypted with the crypto for the RTX stream

Double/PERC can not change the above without changing RTX

There seemed to be some confusion on the list about if RTX without double caused the packet to be encrypted twice. It does.

Solutions considered to RTX

1. Send RTX/FEC non-authenticated, do the repair, and if the repair packet validated, then consider it OK
2. Provide clean way for Double to have only HbH security for identified packets
3. Provide a way to *also* put the original sequence number in RTP headers
4. Like we did for RTCP, add some text in the draft to detect RTX packets and handle them specially
5. Split the SRTP stack in half, redo the PERC Framework, double, and EKT, and then do RTX on an intermediate result inside the stack
6. Send the RTX on a separate DTLS-SRTP flow that is not using double
7. Don’t support RTX or only support RTX in and E2E mode
8. Do a new version of RTX that has the original sequence number in header not payload

Abbreviated Analysis

A bunch of these solutions might work as one off solution for RTX but they are undesirable because they result in a framework that is limited in what it can do for future things that have similar problems as RTX

Just doing RTX and FEC non authenticated, then doing the authentication on repaired packet may be best for “least work for DoS” and current FEC. Unfortunately, for future FEC like reed-solomon, it would not protect FEC state from being corrupted

The DTMF example, where it is not clear if it is desirable for the Media Distributor to read DTMF or not, provides the best example of why we would want the framework to be flexible for this

Proposed Solutions

Add a flag to the OHB (Original Header Block) that indicates if this packet is both HbH and E2E protected or just HbH protected. Call this the s-flag.

- The s-flag would be set for RTCP, RTX, and FEC indicating that the E2E protection would be skipped and may or may not be set for things like DTMF
- The flag needs to be sent in the OHB so that the receiver knows how to process/decrypt the packet before it has enough information to know if it is a type where s-flag would be set

The receiver does not accept media that comes in a packet with the s-flag set

Properties of s-flag solution

- One consistent approach for RTCP, FEC, RTX, DTMF and future extensions
- Minimal changes to double, EKT, and framework as this is very close to how RTCP already works.
  - PR at https://github.com/ietf/perc-wg/pull/112/files is small
- Very easy to implement - basically adds if statement to skip E2E code in double if s-flag is set
  - Receiver needs clear guidance on which packets allow s-flag to be set. For media, this needs to only be things where the payload is already encrypted in E2E form.
- Most flexible solution that allows the framework to support widest range of solutions for future extensions
- The sender or Media Distributor implementation can cache encrypted payloads that might get used later or can recompute them when needed as it wishes

back to chairs ..

Consensus ?

Does the presented solution (PR) address the pending issue on

“Specifying HBH only scope where needed “

Does the proposed solution provides a way to

“Support RTX/FEC/RTCP/DTMF for HBH only use-cases”