Privacy Enhanced RTP Conferencing
1st Virtual Interim 2017
Session 2017-05-24 8:00 - 9:45 (PST)
Chairs:- Suhas Nandakumar, Nils Ohlmeier

Summary

- draft-ietf-perc-double
    - Cullen presented on topic of how to solve hop-by-hop repair in double
    - WG agrees to able to do HBH repair the payload first need get encrypted before applying repairs like RTC or FEC
    - Several participants pointed out that RTX is implemented with the original sequence number plus the unencrypted payload
    - Even though FEC has an non-default mode to first encrypt and then apply FEC several participants have doubts if any existing implementation supports this mode
    - Out of a list of 8 proposals two possible candidates got identified
    - Cullen will write up a proposal how to clearly mark HBH vs. E2E packets
    - Emil will write up or collect existing document for the proposal to have distinct crypto operations in double
    - Sergio will write up the concerns regarding being able to do E2E packet repair
    - Another virtual interim meeting in about two weeks time

Raw notes from Paul Jones:

- Suhas opened the meeting presenting the "Note Well" statement, draft status, review of milestones
- Recap since IETF 98:
    - New drafts of double, ekt, and tunnel
    - Cullen prepared documentation on PERC keying
    - Open issues on RTC/FEC/DTMF are open topics that prompted this interim (made a challenge due to how PERC is presently specified)
- Cullen presented slides on the challenges with RTX, FEC, DTMF, & RTCP
    - Repair HBH is better than repairing E2E, though not all MDs will want to do that
    - Some MDs might want to handle DTMF
    - We had a discussion on RTX with some possible solutions
        - The slides presented said that the RTX packets are formed by taking the SRTP packet as input, but participants pointed out that RTX starts with the original RTP packet
        - There was some discussion at a high level how to produce a solution, including discussion on what can change in terms of existing RFCs and current drafts
        - A key issue (shared with default FEC implementation) is that RTX acts on the RTP packet before encryption
        - Orthogonal question: do we need to do encryption HBH (noting we do agree that authentication is needed)?

        - We need to perform E2E encryption first, then perform any RTX/FEC procedures
        - There was a desire to optimize the solution further, so there was concern with one option that would perhaps be "triple" encryption
        - Given that one pass of encryption is required before RTX/FEC, then RTX/FEC, the order can be determined by the fact that double is employed (concern this might be a layer violation)
        - Jonathan had an "offset" proposal, but he expressed concern himself since it is easy to get wrong and leak confidential information -- in short, he feels it is a bad idea
        - Of the solutions proposed (thus far), there seemed to be support for #2 and #5, which are:
            - (2) Provide clean way for Double to have only HBH security for identified packets
            - (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
    - We had some discussion on MD-initiated media and handling DTMF
        - Do we want a mechanism to indicate that media is encrypted only HBH?
        - There was concern with allowing the MD initiate media, as the MD could inject media into a flow that contain participants' voice and that would not be good
- Areas we need to do additional work:
    - How do we identify E2E and HBH encrypted packets (to allow the MD send/receive packets) (e.g., payload type or other)

Raw notes from Nils Ohlmeier:

- Suhas: Introduction slides
    - Note well
    - Adopted drafts status
    - Milestones
    - Since IETF 98
- Emil: will we have agenda bashing
- Suhas: No. Agenda was clear.
    - Problem statement
- Cullen:
    - Problem to Solve
    - The RTX Problem
    - Jonathan Lenox: This is not how RTX works
    - Emil Ivov: The whole payload including the sequence number is encrypted
    - Cullen: The actual payload is encrypted, then add the original sequence number, and encrypt again
    - Jonathan: No
    - How RTX works
    - Pulling up RFC 4588 section 12

- Cullen: does someone know where in RFC this is
- Jonathan: No, because we did not believe it exists
- EKR: What is the exact problem here?
- Jonathan: You are running AES once
- Sergio: I have implemented twice with one round of AES
- Jonathan: me too and it interpose
- Emil: Change RTX?
- Jonathan: Define a new transform for PERC
- ABR: How about saying don't use RTX with double, use FEC instead
- Sergio: we are jumping into conclusion prematurely.
- Jonathan: something has to change
- Cullen: if RTX works as Jonathan says its crappy
- Solutions considered to RTX
- 1 current FEC would work unauthenticated, but future stateful FEC would not work
- 3 and 8 are duplicates
- Jonathan: a missing option the sender defines how many bytes into the payload the double encryption starts
- Sergio: 5 is essentially my PERC Lite proposal
- Cullen: your solution is the equivalent to do SDES via JavaScript
- Abbreviated Analysis
- Emil: For RTX this would mean the payload is not encrypted?
- Cullen: RTX needs to change in some way to make it work
- Solutions considered to RTX
- Emil: 5 is our proposal, 3 seems reasonable, any other solutions viable
- Cullen: number 2
- Sergio: Does FEC have the same problem as RTX?
- Cullen: encrypted FEC repairs encrypted payload
- Sergio: No
- Mo: the default is to do encrypt first followed by FEC, there is another mode to do it the other way around
- Jonathan: I think nobody implemented the second non-default mode
- Mo: for double you need to do one crypto operation first, before doing any other repair. I think everyone can agree on that
- Jonathan: which do we need to HBH crypto?
- Richard: we need authentication for HBH
- Emil: because you wanted it to look as close as possible to original SRTP
- Ekr: I believe you could do it without encrypting
- Russ: not doing crypto on the HBH make key management harder
- Jonathan: this discussion is orthogonal
- Emil: everyone except Cullen agrees we can do 5
- Emil: we yet have to hear objections to Sergios
- Mo: everybody agrees encrypted payload before repair

- 2nd what people call RTX and FEC is different from what needs to be implemented for double
- Emil: these are details, but I think in general we agree
- Suhas: Mo's proposal do we agree on encrypting before repair
- Emil: yes
- Mo: the point of disagreement how many rounds of crypto with which keys
- and we are talking about any HBH repair operation
- Sergio: red can not be used for any of this, because it only repairs payload
- Jonathan: yes red can only be used end to end
- Cullen: how to make progress
- write up the solution for number 5 (from the slides) or point to the already written up
- Suhas: another call in two weeks?
- Mo: make some progress on this call?
- 2 and 5 are not as different as people think
- Emil: 2 is tripple?
- Mo: yes 2 is tripple
- Emil: I would love to see a proposal for 2 with tripple
- Cullen: lets compare options 2 and 5 how much crypto
- Mo: by allowing Null cipher on HBH you would get 5
- Sergio: the difference between 2 and 5 require different changes on the implementations
- Mo: how do you signal which crypto operation to do
- Emil: I don't think that's accurate
- option number 3 id not bad either
- Cullen: I'll write it up
- Mo: OHB already allows you to overwrite the sequence number
- Cullen: you utilize the seq number in OHB to communicate the seq number
- Emil: that would kill the crypto
- Jonathan: that would only work for RTX, but not for FEC
- Cullen: does specifying the encrypt offset would that work for flex fec?
- Jonathan: I would have to check - you can't to it fixed it depends on the FEC payload
- Emil: the client needs to understand that RTX works different in double, then without - so we don't need a new payload type
- having another payload allows you to reject either flavor of RTX
- Richard: you can only do end-to-end repair with a different payload type
- Cullen: a design goal is to keep the changes for client minimal
- does the client need to aware of the media type it operates on?
- Emil: I think people are okay with layer violations
- Mo: an indicator if something is HBH or E2E might in general be helpful
- Richard: yes we need that
- Emil: I don't understand
- Mo: how about the tones for people joining and dropping
- Emil: but that would allow the MD to replay voice of participants

- Richard: the MD should only be allowed to inject certain types of packets
- Ekr: I don't think that is right.
- Emil: isn't enough to use two different payload types to differentiate between HBH and E2E?
- Cullen: does have anyone strong objections against the offset option?
- Jonathan: yes I do because it goes really wrong if you get the offset wrong
- Mo: only do it on repair packets to prevent it going horribly wrong
- Cullen: if we do it only for repair packets its very similar to the flag
- Emil: we agree that we can make repair work on encrypted
- Jonathan, Mo: yes
- Suhas: Emil and Cullen work on options 2 and 5
- for header extensions not encrypted, but authenticated end-to-end
- for SSRC re-write Ericson pointed out the security risk
- Sergio: verify if E2E RTX and FEC would work
- Jonathan: if the MD changes something and covers it in the OHB for regular media probably doesn't work with E2E repair packets
- Emil: other options then 5 have high implementation costs
- Jonathan: what are the MD operations needed for the options on the table?
- have clients and MD's have the same lib operations, rather then using their own distinct operations
- Suhas: Emil and Cullen work on proposing their solutions, and Sergio write up the concern regarding E2E repairs and a call in two weeks