IETF 92 AVTCORE

DRAFT-IETF-AVTCORE-SRTP-EKT

JOHN MATTSSON (EDITOR)

ERICSSON RESEARCH

SCOPE OF DRAFT-IETF-AVTCORE-EKT?

- draft-jones-avtcore-private-media-framework proposes to use EKT for the e2e secure conferencing use case.

- The following changes is proposed:
  - Changed position of the EKT field.
  - ROC is transmitted outside of EKT.
  - Extension of EKT to negotiate e2e Protection Profile
  - Extension of EKT in order to send the participant identifier.

- Anyone against broadening the scope of EKT?

EKT (ISN) PROBLEMS WITH RTCP

- EKT uses a the ISN field (Initial Sequence Number) to signal the timing for SRTP re-keying.
  - `EKT_Plaintext = SRTP_Master_Key || SSRC || ROC || ISN`

    where ISN is the initial SEQ for the new SRTP master key.

- Problem is that SRTCP have 31 bit SRTCP Index instead of SEQ.
  - Problematic for Sender Reports.
  - Even more problematic for Receiver Reports

REPLAY ATTACKS CAUSED BY ISN

- Assume a sender sends the following two EKT fields during ROC = 17
  - `EKT_Plaintext = Key = XX || SSRC = 1234 || ROC = 17 || ISN = 0`
  - `EKT_Plaintext = Key = YY || SSRC = 1234 || ROC = 17 || ISN = 1337`

- An ISN = 0 has a special meaning, an attacker can replay the first EKT
  - `EKT_Plaintext = Key = XX || SSRC = 1234 || ROC = 17 || ISN = 0`
  - The receiver will accept the replayed EKT and set the key to XX
  - As the sender uses Key = YY for SEQ >= 1337, the receiver will discard all SRTP packets until the sender sends a new EKT field.

- One replayed EKT packet causes long term DoS…

REPLAY ATTACKS CAUSED BY ISN

- Original source is using SSRC = 8128 and ROC = 2^31, sends
  - `EKT_Plaintext = Key = XX || SSRC = 8128 || ROC = 2^31 || ISN = 0`
- New source starts using SSRC = 8128 and ROC = 0, sends
  - `EKT_Plaintext = Key = ZZ || SSRC = 8128 || ROC = 0 || ISN = 0`
- The original source detects the collision and moves to a new SSRC
- An attacker can replay EKT from the Original source
  - The receiver will accept the EKT and use Key = XX and ROC = 2^31 for the new source.
  - The receiver will discard all SRTP and EKT packets until the new source reaches ROC = 2^31

- One replayed EKT packet causes very long term DoS…

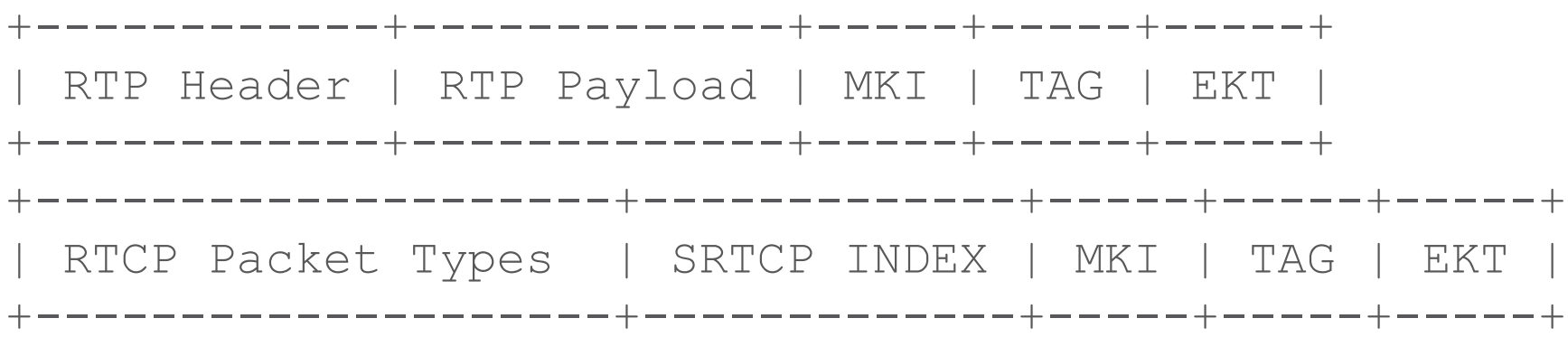
SUGGESTIONS

Do we really need several SRTP Master keys per SSRC and EKT SPI?

- If no, simply remove ISN. Problems solved. EKT_Plaintext would be
  - `EKT_Plaintext = SRTP_Master_Key || SSRC || ROC`

- If yes, remove ISN and use MKI. Problems solved. EKT_Plaintext would be:
  - `EKT_Plaintext = SRTP_Master_Key || SSRC || ROC || MKI`

  - The SRTP and SRTCP packets would look like:



ANALYSIS: MKI INSTEAED OF ISN

Pros

- Solves the problems with SRTCP and ISN
  - EKT can be used in RTP and and/or RTCP.
  - SRTP and SRTCP may use different keys.
- Solves the problems with replay of Full EKT fields.
  - If the receiver has the key with MKI = X, the replay has no effect.
  - If the receiver does not have the key with MKI = X, the effect is just that the receiver stores the key.
  - In the SSRC collision case the replay attack is only effective if both sources use the same MKI value and the DoS is then only until the sender sends a new EKT field.
  - ROC calculation can continue to be modulo 2^31

Cons

- MKI could by default be one byte. Packets with Short EKT field would then be 1 byte longer.

Could specify that MKI is optional, a source is then limited to one SRTP master key per EKT SPI.