DTLS OVER SCTP BIS

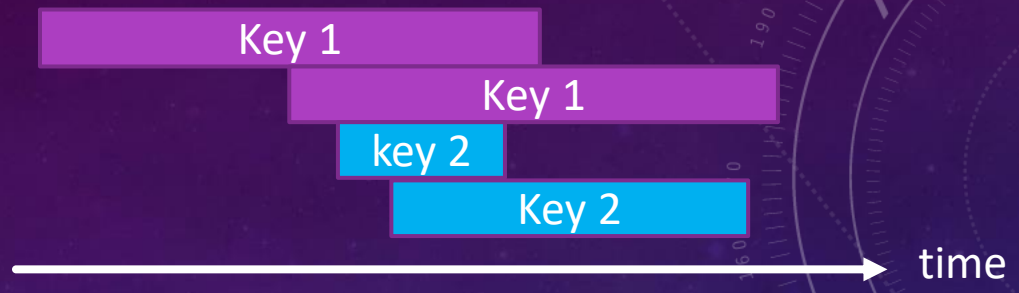
DRAFT-IETF-TSVWG-DTLS-OVER-SCTP-BIS-01

MAGNUS WESTERLUND
CLAUDIO PORFIRI
JOHN MATTSSON
MICHAEL TÜXEN

-01 UPDATE

- Switched to use SCTP Adaptation Layer Indication for the initial negotiation
- Clarified Replay Protection
- Clarified some behavior in failure scenarios
  - Receiver side resource exhaustion
  - SCTP user message and DTLS record mismatch
- Defined Socket API extensions for SCTP-AUTH
  - Allow the secure usage of non-mandatory algorithms
- Receiver dropping old DTLS keys after one full epoch based on SCTP-AUTH key
- Clarified Differences between DTLS 1.2 and 1.3
  - DTLS 1.3 removed renegotiation
    - No Perfect Forward Secrecy rekeying
    - No Server re-authentication
  - Current draft does not have feature parity
- Potential key-epoch limitation (16-bit field)
  - DTLS 1.3 may address this going to 32/64 bits
- DTLS requires dropping old keys in the time frame of one Maximum Segment Lifetime
  - SCTP can’t ensure processing of all user message using old Key in that time frame

REKEYING HEADACHES



Issue 1: DTLS 1.3 and semi-permanent sessions

- No PFS rekeying
- No mutual re-authentication
- No updated TLS Exporter secret for SCTP-AUTH
- Significant issue for long lived sessions
- Uncertain if DTLS 1.3 extensions will be defined
- Some Applications can not restart their SCTP association without significant cost

Issue 2: Knowing when old key is no longer needed

- DTLS Sender side can track when all DTLS records protected by old key in all streams have been received by the SCTP stack of the peer.
- DTLS Receiver struggles due to multi-streaming knowing for certainty that it has received all DTLS records using the old key
- Current Solution based on SCTP-AUTH Key-ID
  - Limits a single user message to one key epoch
  - Application impact?

DESIGN DIRECTIONS

Issue 1: DTLS 1.3

A. Instead of rekeying: Create a new DTLS connection and seamlessly switch over to it
   - DTLS have DTLS Connection ID
   - Issue #1 still needs to be handled on Connection level

B. Work towards DTLS extension to solve issue before DTLS 1.2 needs to be replaced

Solution for DTLS 1.3 may require separate draft for timely conclusion

Issue 2: Knowing when rekeying is completed

1. Use current mechanism with SCTP-Auth Key-ID
   - Accept limitations
2. Require SCTP API changes to enable SCTP-AUTH rekeying at any point
3. Have DTLS use multiple SCTP user messages per ULP User Message
4. Add other explicit signal that key change is completed in SCTP or DTLS layer?

CONCLUSION

- Authors will continue work on solutions
  - Target solution proposal before end of September
- Appreciate any input
- Want to avoid significant delay