draft-ietf-avt-dtls-srtp

David McGrew, Eric Rescorla

AVT WG 69th IETF

Draft Background

- Based on draft-mcgrew-tls-srtp (6/06)
- Discussed extensively in RTPsec
- Protects point-to-point RTP
  - DTLS handshake establishes keys
  - SRTP packet processing for RTP/RTCP

Recent Changes

1. Clarification: in the Symmetric RTP case, only one DTLS handshake is needed
2. Duplicate list of “srtp profiles” eliminated
3. Editorial nits

Open Issue #1

- Use the "TLS Extractor” instead of purpose-built extension to TLS KDF?
  - If so, Section 3.3 should be rewritten accordingly

Open Issue #2

- Do we need a "symmetry breaking" rule (Section 3.6.2.1)
  - Defines what should happen when a devices that sent a clientHello receives a clientHello
- Would handle cases in which the signaling system can't tell a device which should act as client and which should act as server
- **Opportunistic probing**

Open Issue #3

- Use “single DTLS session per SRTP session" (Appendix B)
- Pro
  - Lower computation and latency
  - Better match for SRTP policy model
- Con
  - Deviates from TLS practice

“Single DTLS”

```
Client                Server
(Sender)            (Receiver)
   <--- DTLS ----->    src/dst = a/b and b/a
   ---- SRTP ----->    src/dst = a/b, clientWriteKeys
   ---- SRTCP --->    src/dst = c/d, clientWriteKeys
   <--- SRTCP ----    src/dst = d/c, serverWriteKeys
```

Keys on ports c/d derived from handshake on ports a/b

DTLS per SRTP & SRTCP

```
Client                  Server
(Sender)              (Receiver)
   <--- DTLS ----->     src/dst = a/b and b/a
   ---- SRTP ---->     src/dst = a/b clientWriteKeys

   <--- DTLS ----->     src/dst = c/d and d/c
   ---- SRTCP --->     src/dst = c/d clientWriteKeys
   <--- SRTCP ----     src/dst = d/c serverWriteKeys
```

Keys derived from handshake on same ports pair