CID and/or DTLS 1.3

Eric Rescorla
ekr@rtfm.com
General Idea

- **We did CID for DTLS 1.2 and DTLS 1.3**
  - Lot of demand for DTLS 1.2 CID now.
  - A little more time for DTLS 1.3 CID
- **Some question about whether the DTLS 1.3 headers are really ideal**
  - And maybe we want to harmonize with QUIC
- **Can we unlock DTLS 1.2 CID while we think about DTLS 1.3 CID**
Header formats

- **CID draft needs to accommodate mixed CID/non-CID flows**
  - Some clients may not support CID even if the server wants it
  - Proposals to have an indicator bit in the header;
  - this is not necessary, but it is convenient

- **Indicator bit is straightforward in DTLS 1.2 and DTLS 1.3 long header**
  - Spare bits in the CT and the length

- **... but it’s not straightforward in DTLS 1.3**
  - Because the header is carefully packed
  - So this is also important to resolve for DTLS 1.3
Option One: Implicit CIDs

- No explicit “CID present bit”
  - Remember: receiver controls CID
- In the TLS 1.2 and TLS 1.3 headers, CID goes right before length so you need to demux CID versus length
  - Lengths > 2^16 are forbidden
  - If all CIDs have the high bit set, demux is easy
- Short header is harder (no length)
  - Fix the first $n$ bits of CID
  - The first $n$ bits of ciphertext are random
  - If the prefix matches, assume CID present and try to decrypt
    - Error rate $2^{-n}$
    - If decryption fails, you can try without CID or just discard packet
Option 2: Explicit header

- Easy with DTLS 1.2 and DTLS 1.3 long header
  - Either CT or length available
  - Reasons to believe CT is better

- Harder with DTLS 1.3 header
  - We’d need to redesign
  - Thomas suggested expanding by one byte and use some of the bits for flags
  - DTLS 1.2 sequence number might also be too short (12 bits)
Potential Unified Header Design

```
+------------------------+
| 0 | 0 | 1 | C | L | X | X | X |
+------------------------+
|                      E | Sequence Number   |
+------------------------+
```

- C: CID present
- L: Length present (2 bytes afterwards)
- E: Epoch (2 bits)
- Sequence number: 14 bits
Arguments for New Header

- One header format, not two
- **Gives us 2 more bits for the sequence number (14 bits)**
  - We could actually have 2 more if we use two of the bits in the first byte for epoch
- Plus we have some room for other flags (1-3 bits)

- We don’t need to do exactly this
  - Might rip off the QUIC headers
  - The question is if we think this kind of thing makes a CID indicator bit look better
Proposed Way Forward

- Decide if we want an implicit or explicit CID
- If implicit, we’re done-ish
- If explicit, can define for DTLS 1.2 right away
  - Modulo CT versus length bikeshed
- Work a bit on the best DTLS 1.3 format
Sequential Sequence Numbers

- Sequential sequence numbers leak CID linkage
  - Need to do something
- DTLS 1.2
  - Just use the skipping trick from QUIC
- DTLS 1.3
  - Probably we should just encrypt the sequence numbers
  - This may take some time to work out
CID Update

- Draft currently uses a DTLS 1.3 post-handshake message for CID update
  - No answer for TLS 1.2
- Some options for DTLS 1.2
  - Do nothing
  - Require rehandshake
  - Port the post-handshake messages into DTLS 1.2

Discuss