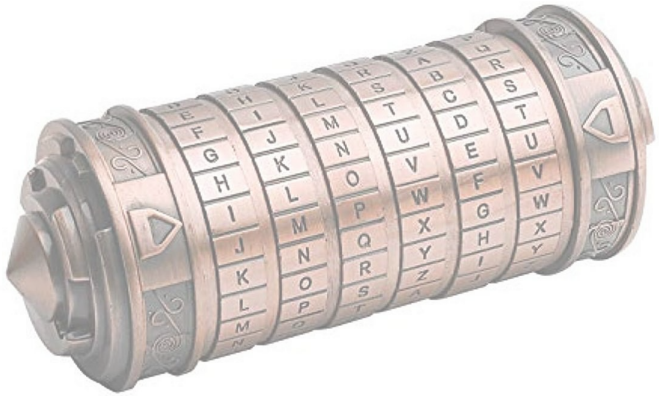
Cryptex

Encrypted Extensions (and CSRCs)

draft-uberti-avtcore-cryptex-01

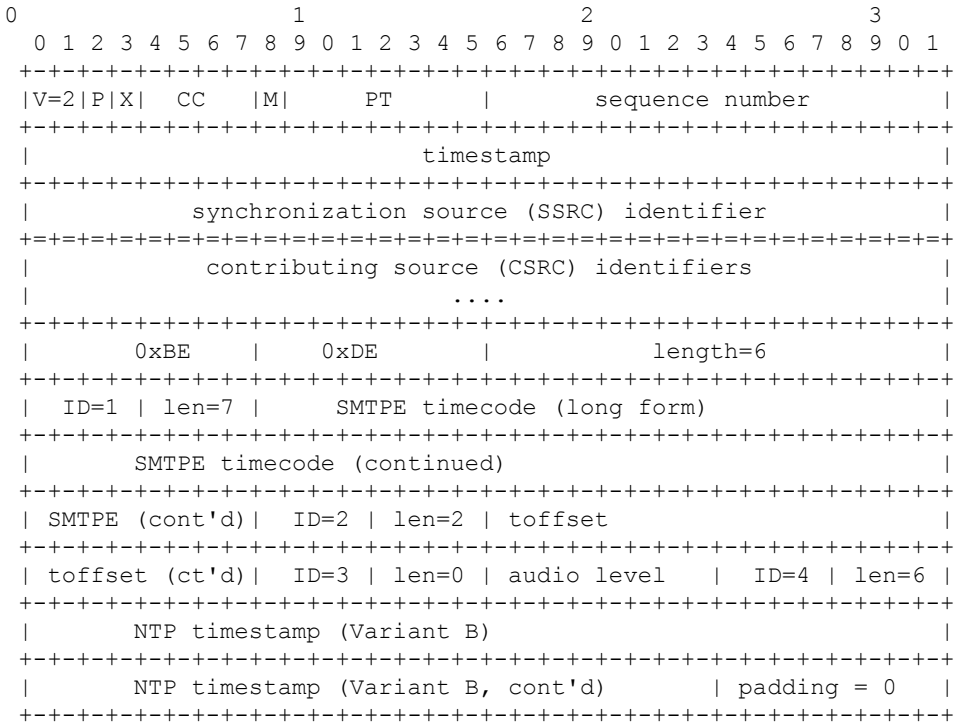


Justin Uberti, Google

IETF 109

Problem

None of the RTP header is encrypted, including extensions and CSRCs



More Problems

WebRTC implementations use a lot of extensions

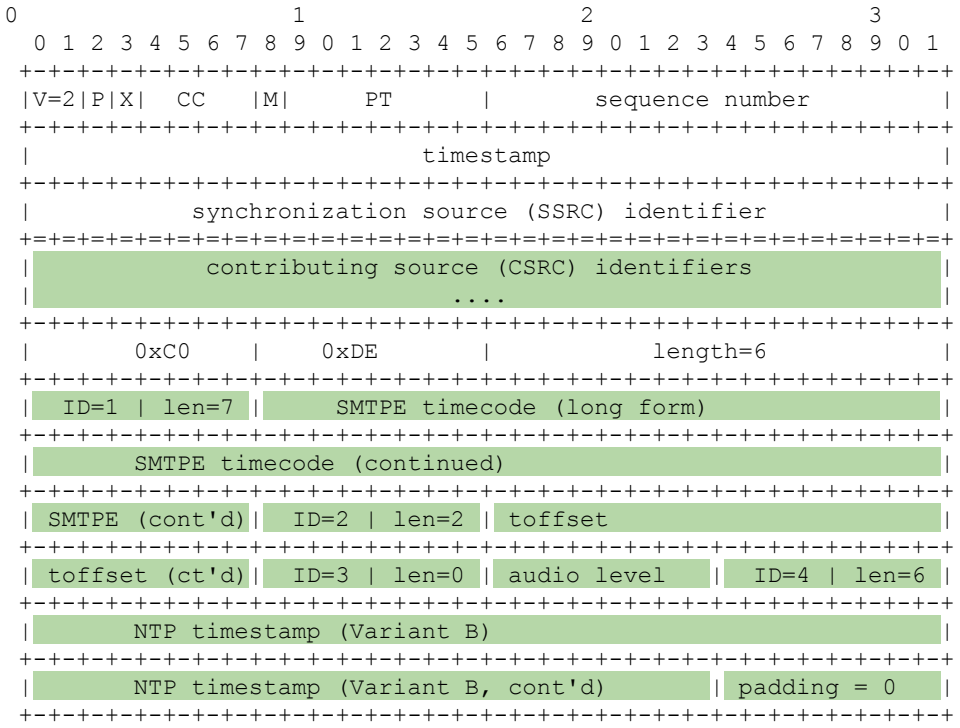
```
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=extmap:14 urn:ietf:params:rtp-hdrext:toffset
a=extmap:2 http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time
a=extmap:13 urn:3gpp:video-orientation
a=extmap:3 http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
a=extmap:12 http://www.webrtc.org/experiments/rtp-hdrext/playout-delay
a=extmap:11 http://www.webrtc.org/experiments/rtp-hdrext/video-content-type
a=extmap:7 http://www.webrtc.org/experiments/rtp-hdrext/video-timing
a=extmap:8 http://tools.ietf.org/html/draft-ietf-avtext-framemarking-07
a=extmap:9 http://www.webrtc.org/experiments/rtp-hdrext/color-space
a=extmap:4 urn:ietf:params:rtp-hdrext:sdes:mid
a=extmap:5 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
a=extmap:6 urn:ietf:params:rtp-hdrext:sdes:repaired-rtp-stream-id
```

Some of these are at least somewhat sensitive
(e.g., ssrc-audio-level, video-content-type)

All of these leak some amount of metadata
(e.g., application type, HDR support, HW/SW encoder)

Solution

Encrypt CSRCs + extension block

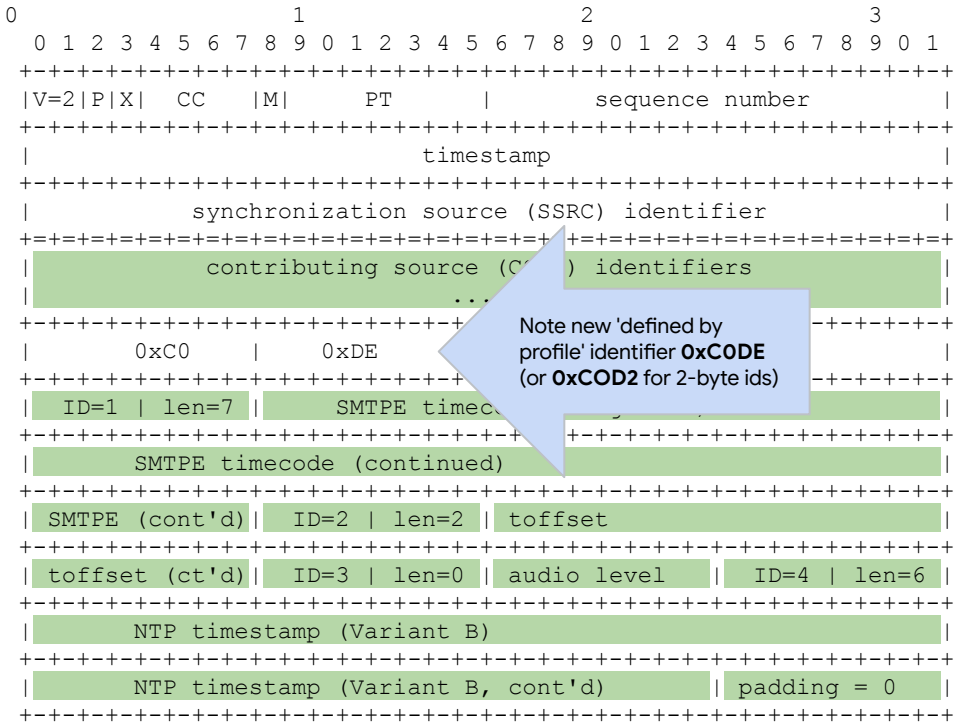


Uses existing SRTP encryption mechanism

Compatible with existing RTP parsing code

Solution

Encrypt CSRCs + extension block



Details

- Presence of encryption indicated via `0xC0DE` or `0xCOD2`
  - If packet has only CSRCs but no extensions, an empty `0xC0DE` block MUST be added
- Capability signaled via new SDP attribute
  - Allowed at session-level or media-level
  - `a=cryptex`?
- In WebRTC, enabled via new RTCConfiguration API
  - 'negotiate' or 'require'

Thank You