Null-Cipher Mode for DTLS-SRTP in WebRTC

Giri Mandyam

Qualcomm Innovation Center

Introduction

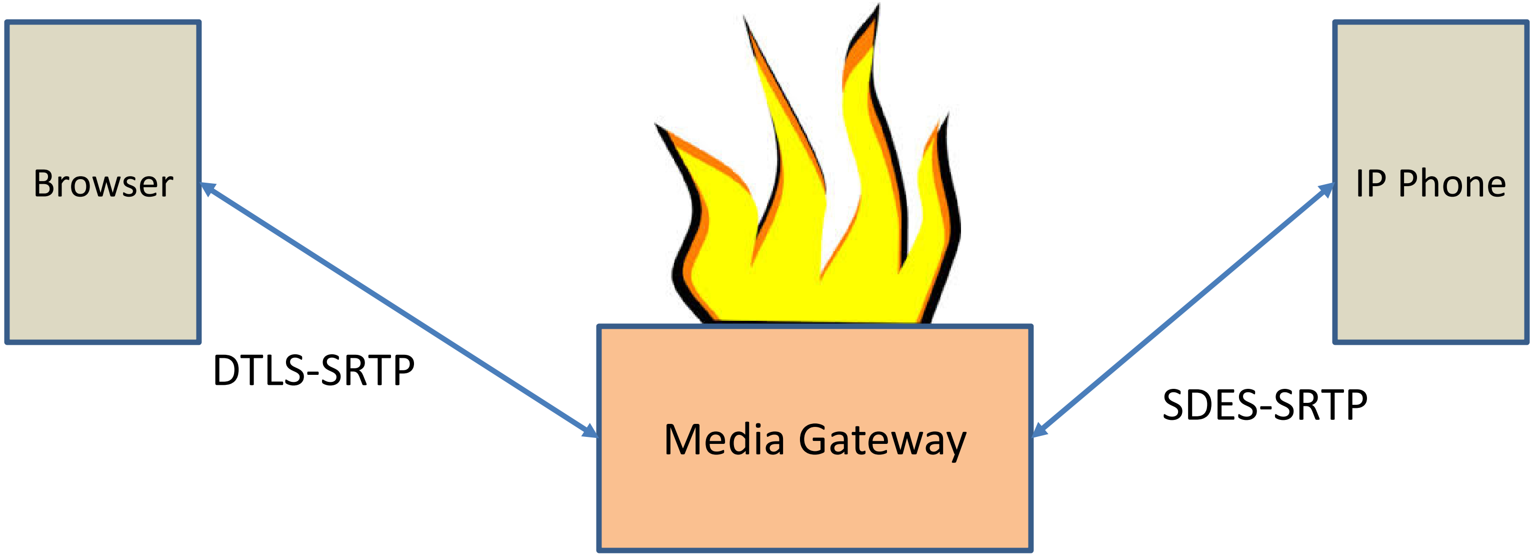
- RFC 5764 calls out two protection profiles involving null ciphers
  - SRTP_NULL_HMAC_SHA1_80
  - SRTP_NULL_HMAC_SHA1_32
- It is not required for implementations to support the null cipher profiles, but it is not prohibited either
- Recommendation is that valid WebRTC implementations be allowed to support null cipher

Testing Purposes

- Developers have requested the ability to disable media encryption
  - https://code.google.com/p/webrtc/issues/detail?id=491

Interop Between Different Security Domains

- It has already been discussed in the RTCWeb WG the pitfalls of having to decrypt-re-encrypt for SDES-SRTP and DTLS-SRTP interworking (without EKT)

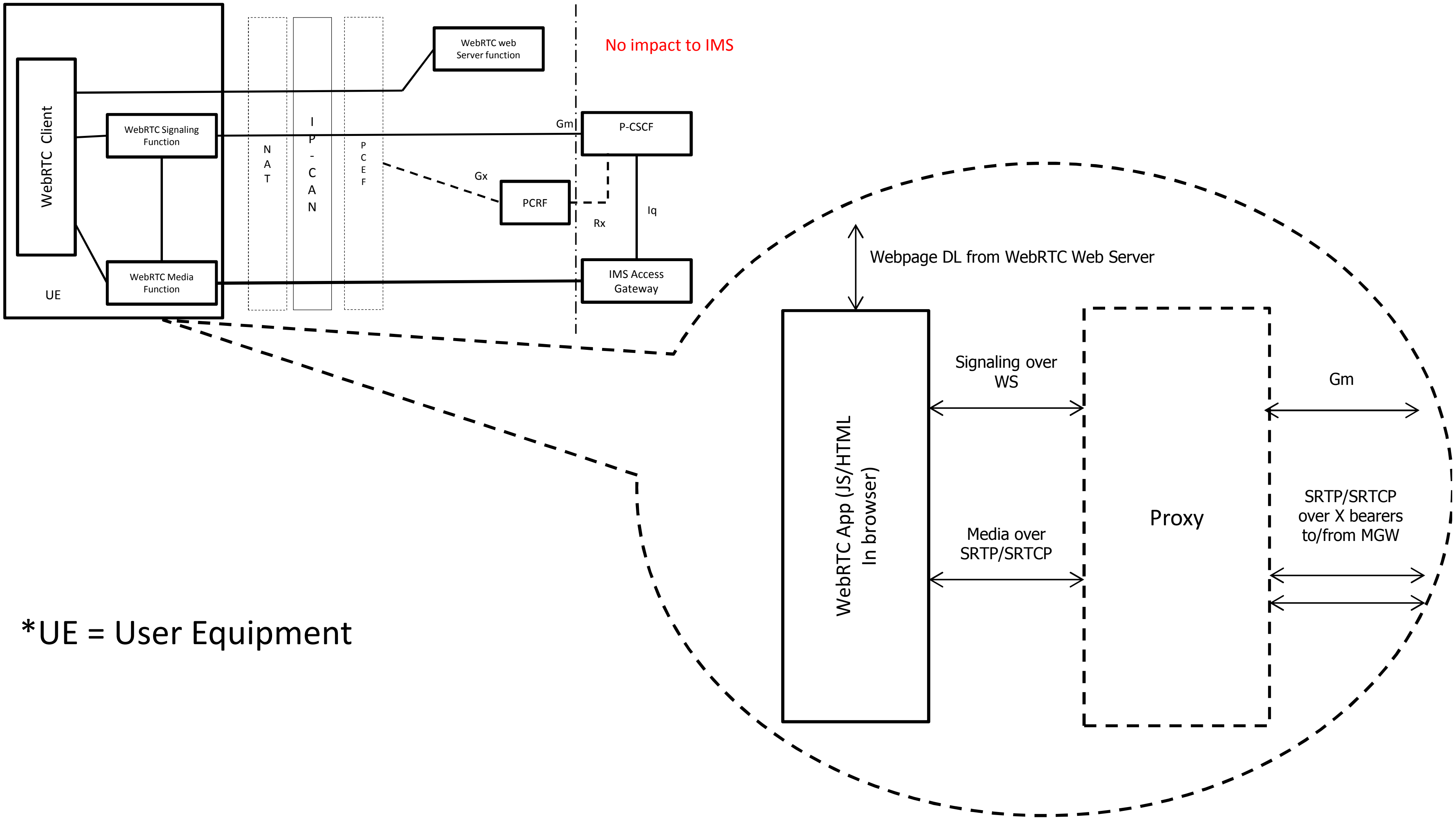


*Adapted from http://tools.ietf.org/agenda/83/slides/slides-83-rtcweb-3.pdf

Other example: Interop with IMS MMTel (VoLTE)

- IMS clients and core network do not currently support DTLS
  - – 3GPP has proposed core network enhancements that would terminate DTLS-SRTP in the network for adaptation to RTP/SRTP (e.g. SDES-based)
  - – “MGW on fire” scenario from previous slide
- What if IMS interop was on device side?

Mobile (UE) Proxy for WebRTC for IMS Interop



*UE = User Equipment

Where does Null Cipher fit in?

- Do not want “MGW on fire” scenario inside of handheld devices
- Null cipher negotiation results in unencrypted media in this case, but only within mobile device
  - Eavesdropping possible but unlikely
  - Ease of implementation; media originating from browser does not have to be unencrypted in absence of EKT

Recommendations

- For most scenarios, unencrypted media is not desirable
  - Valid WebRTC implementations must support encryption
- WebRTC implementations that support null cipher should be allowed
  - Useful for narrow purposes, e.g. testing and device-based proxy for interop w/other domains
- Consistent with RFC 5764