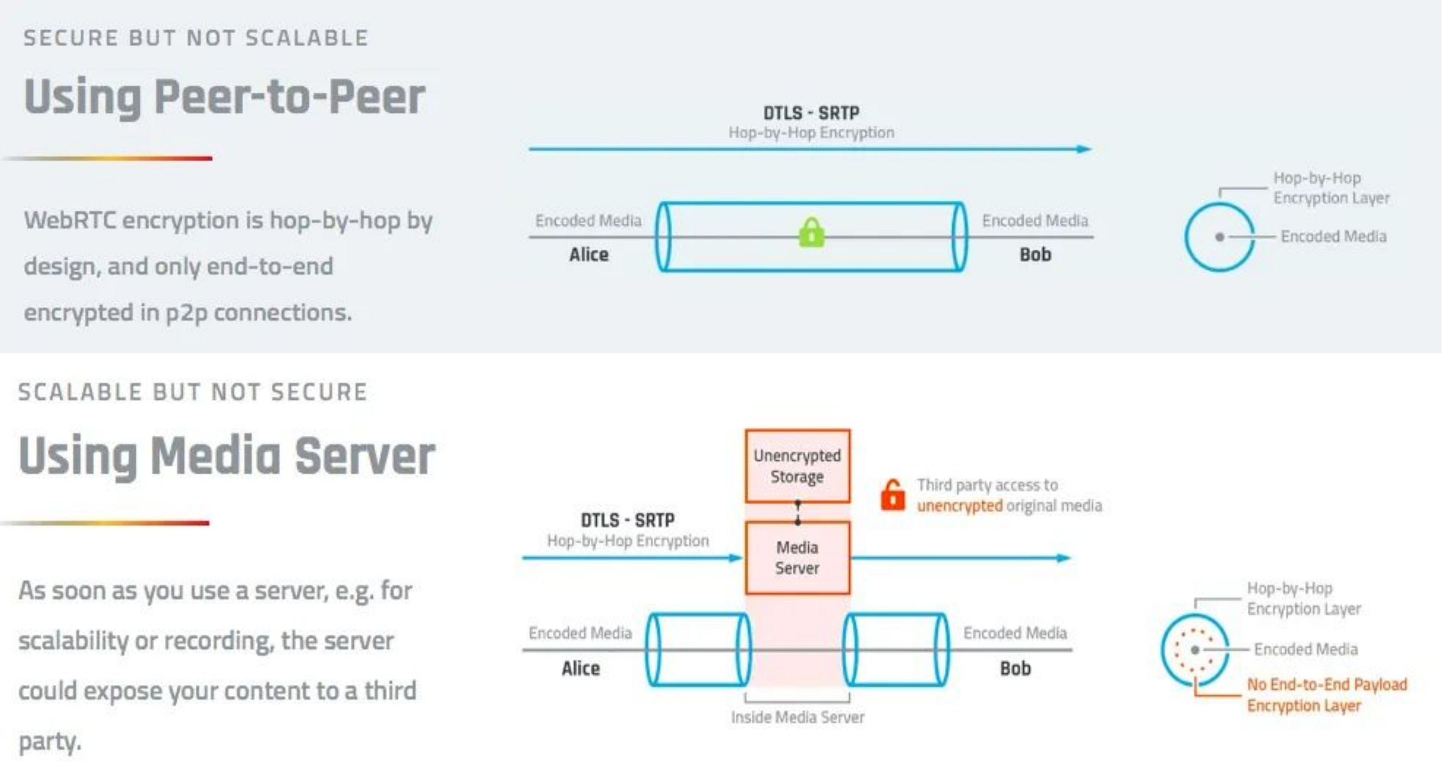
SFrame

RTP Encapsulation

WebRTC P2P encryption



SECURE BUT NOT SCALABLE

Using Peer-to-Peer

WebRTC encryption is hop-by-hop by design, and only end-to-end encrypted in p2p connections.

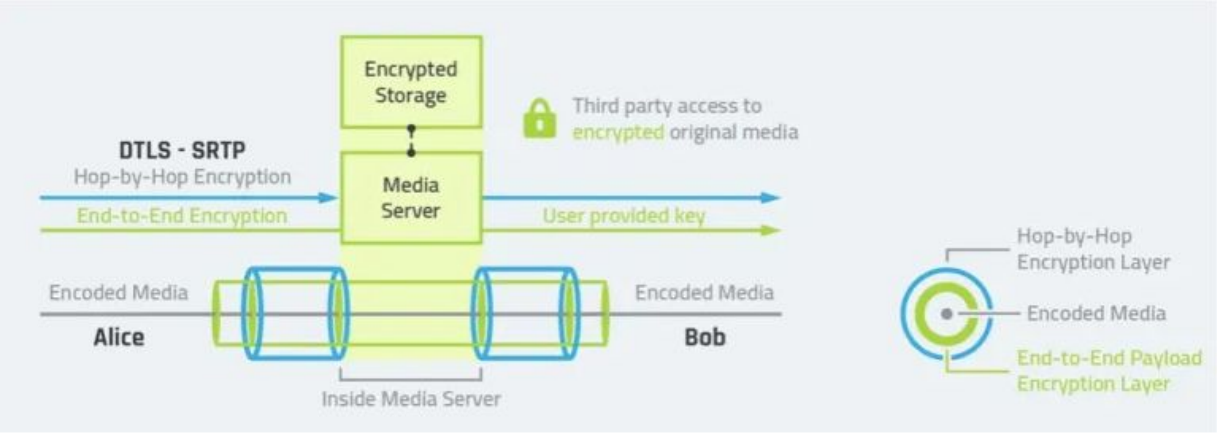
SCALABLE BUT NOT SECURE

Using Media Server

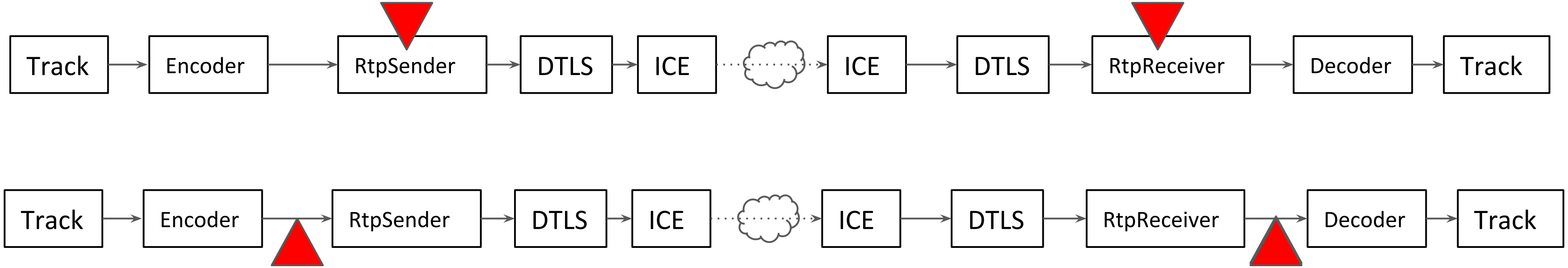
As soon as you use a server, e.g. for scalability or recording, the server could expose your content to a third party.

SFrame goals

- Provide a secure E2EE mechanism for audio and video in conference calls.
- Decouple media encryption from key management to allow SFrame to be used with an arbitrary KMS.
- Minimize overhead.
- Independence from the underlying transport.
- When used with RTP, require no special handling for RTX and FEC.
- Minimize the changes needed in SFU servers and in endpoints.
- Work with the most popular audio and video codecs used in conferencing scenarios.



Per packet vs per frame encryption

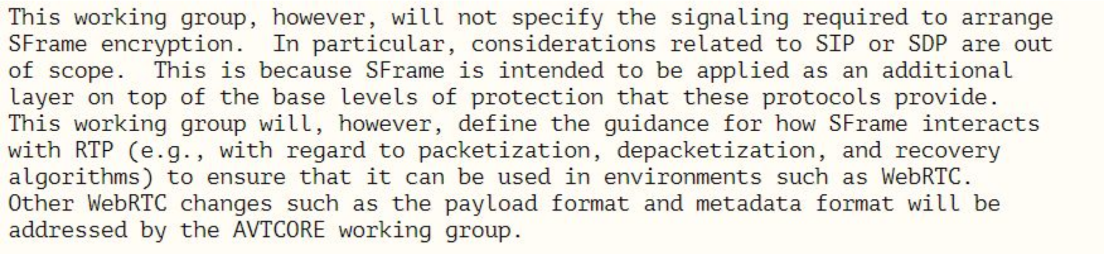


- Implemented in web browsers via Insertable Streams API which allows not only e2ee but any transformation of the media frames in JS. This allows other use cases like inserting app-defined metadata in the video frame.

Video Payloads usage within SFrame

- Not all video codecs support SFrame easily.
- Each video codec requires different processing by SFrame if used with their standard RTP packetization.
- Having a different solution per video codec will require extra specification effort, will make implementation harder and will create huge problems in interoperability.
- SFUs requires access to frame metadata for detecting frame type and performing layer selection.
- If RTP, Frame metadata is better carried on an header extension than inside the payload.
- It would be preferably to have a transport protocol agnostic solution, as SFrame.

https://datatracker.ietf.org/meeting/109/materials/slides-109-sframe-video-payloads-usage-with-sframe-00



This working group, however, will not specify the signaling required to arrange SFrame encryption. In particular, considerations related to SIP or SDP are out of scope. This is because SFrame is intended to be applied as an additional layer on top of the base levels of protection that these protocols provide. This working group will, however, define the guidance for how SFrame interacts with RTP (e.g., with regard to packetization, depacketization, and recovery algorithms) to ensure that it can be used in environments such as WebRTC. Other WebRTC changes such as the payload format and metadata format will be addressed by the AVTCORE working group.

Next steps

- Video packetization which is codec agnostic and allows to transport a raw binary blob.
- Metadata RTP header extension for SFU operations.
  - *Framemarking.*
  - *AV1 Dependency Descriptor.*
- SDP negotiation.
  - Define negotiation of encrypted and non-encrypted formats and how are they related.
  - *RTP Payload Format Restrictions (draft-ietf-mmusic-rid).*