SFrame
E2EE for Video Conferencing

IETF 109

emadomara@google.com

Goals

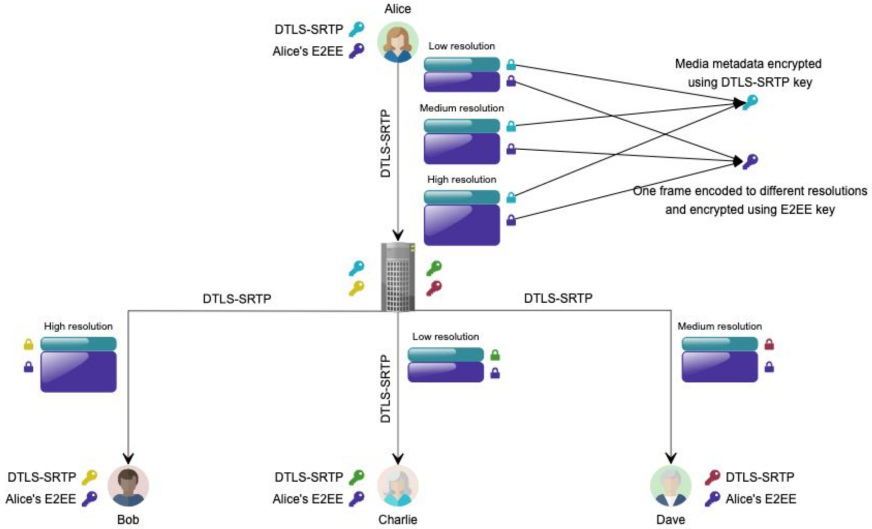
- Goals
  - Security
  - Simplicity
  - Efficiency
  - Transport agnostic
- Non Goals
  - Signaling
  - Metadata payload format
  - Key exchange

Secure Frame

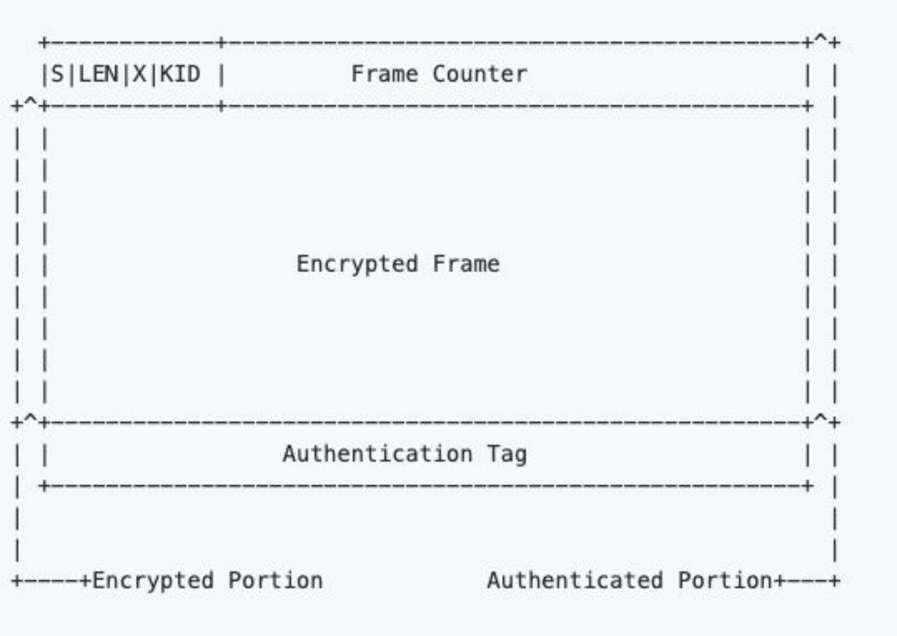
- A new protocol to end-to-end encrypt video conferences
- Encrypt the entire media frame instead of per packet encryption to reduce the overhead
- Transport agnostic as the encryption happens before packetization
- Simple to implement by the client and easy to adopt by existing media backends
- Compatible with existing packets fixing schemas like FEC

SFrame

- Mechanism to efficiently encrypt RTC traffic end to end
  - Encrypts the entire media frame rather than individual packets to minimize the overhead
  - Exposes only the metadata needed by the server to route the streams
  - Individual packets are still HBH encrypted
- SFrame keys are exchanged securely out of band between the endpoints
  - Each user has their own key to encrypt their outgoing traffic
  - Can be used with any KMS like Signal or MLS
  - Keys are exchanged via the signaling channel at the call setup and when the call participants changes
- The server can only access the media metadata but **can not access** the media contents

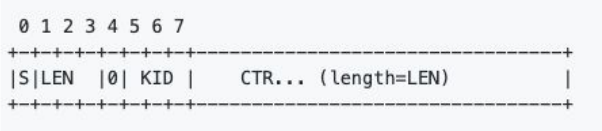


Wire Format



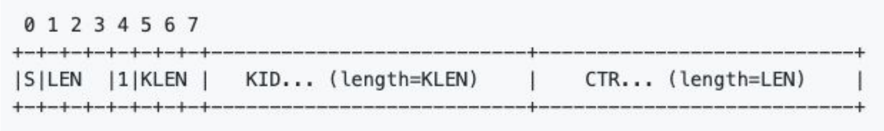
```
   +------------+------------------------------------------+^+
   |S|LEN|X|KID |         Frame Counter                    | |
 +^+------------+------------------------------------------+ |
 | |                                                       | |
 | |                                                       | |
 | |                                                       | |
 | |                                                       | |
 | |                 Encrypted Frame                       | |
 | |                                                       | |
 | |                                                       | |
 | |                                                       | |
 | |                                                       | |
 +^+-------------------------------------------------------+^+
 | |                 Authentication Tag                    | |
 | +-------------------------------------------------------+ |
 |                                                           |
 |                                                           |
 +----+Encrypted Portion            Authenticated Portion+---+
```

SFrame payload



```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+---------------------------------+
|S|LEN  |0| KID |    CTR... (length=LEN)          |
+-+-+-+-+-+-+-+-+---------------------------------+
```

SFrame short header

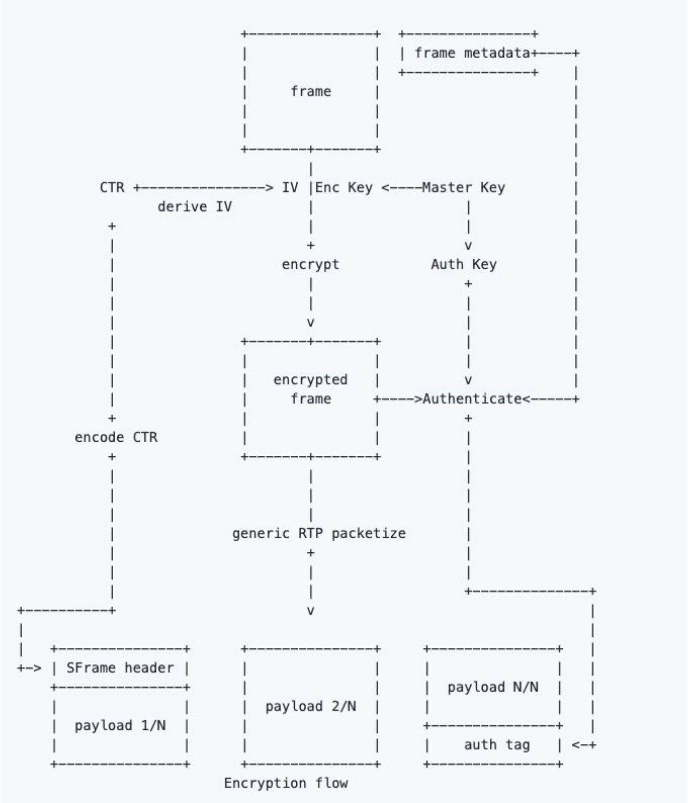


```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+---------------------------+---------------------------+
|S|LEN  |1|KLEN |   KID... (length=KLEN)    |    CTR... (length=LEN)    |
+-+-+-+-+-+-+-+-+---------------------------+---------------------------+
```

SFrame long header

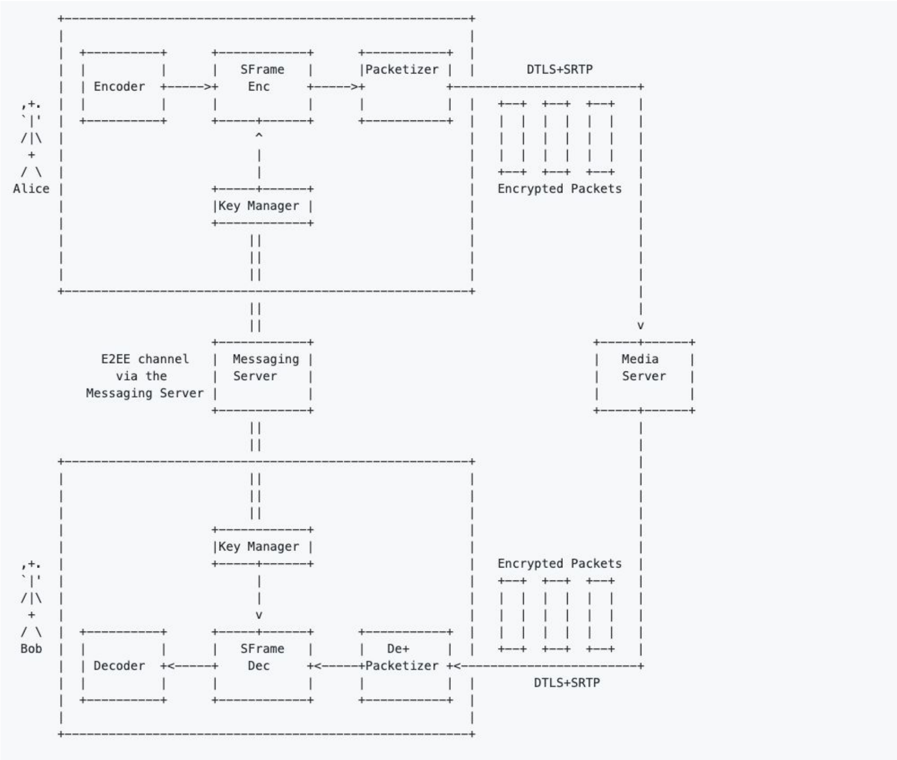
Encryption Schema

- Each endpoint creates and securely exchange their master key
- From the master key, SFrame derives 3 keys
    - Encryption key to encrypt the media frame
    - Authentication key to authenticate the encrypted frame. SFrame header and the media metadata
    - Salt key to derive the IV
- The entire payload is then split into smaller packets



SFrame in WebRTC

- SFrame works with existing RTC frameworks like WebRTC
- The encryptor in injected after the frame is encoded and before it is packetized
- Media metadata are passed to the server using a special RTP header extension
- The server can construct the encrypted frame without access the contents



Open Issues

WebRTC Changes

- Changes needed from other WebRTC WG
  - Signaling SFrame
    Signaling the use of SFrame in the SDP

  - RTP payload type
    New RTP payload type for SFrame packets

  - Frame metadata RTP header extension
    New RTP header extension to pass the frame metadata

Signature: Sign or not to Sign?

- To avoid impersonation by a malicious user, the frame needs to be signed
- Signature overhead is significant
- Proposals
  - Sign every N frame (Currently in the document)
    - Every N frame sends a signature over all hashes of the last N Frames
    - Sends the N hashes along the signature
    - Very complex
  - No Signature
    - Prefered
    - Update the document to remove the current signature schema

Partial Frames

- Some codecs like H264 uses smaller decodable units (NAL Units)
- The current specs supports only full frame
- Recipients won’t be able to decode the smaller unit until the entire frame is delivered and decrypted
- Proposal
    - Add support to encrypt partial frames
    - Increase the overhead but adds more flexibility

Thank You!