Encrypted Key Transport for Secure RTP

`draft-mcgrew-srtp-ekt-00.txt`

David McGrew

Flemming Andreasen

Lakshminath Dondeti

Overview

- In-band keying, protected by separate RTP session-level key
- Conveys SRTP master key and ROC
- Contains ‘Offer’ correlator
  - Security Parameter Index (SPI)
- Indicates key scope
  - Initial Sequence Number (ISN)
- Uses SRTCP Authentication Tag for transport
  - Could use SRTP Auth Tag or Header Extension

How it works

SRTP source

SRTCP Packet

EKT Data

SRTCP Processing

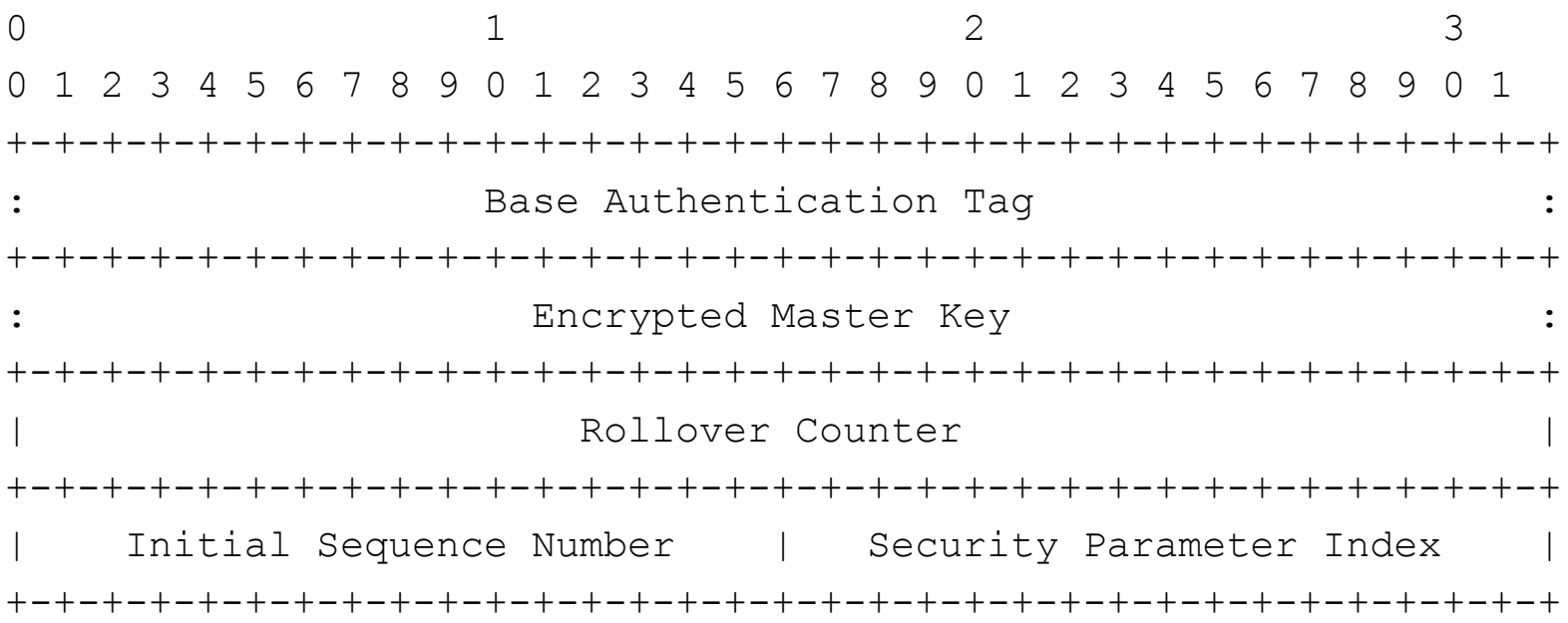
Session Organizer

EKT key

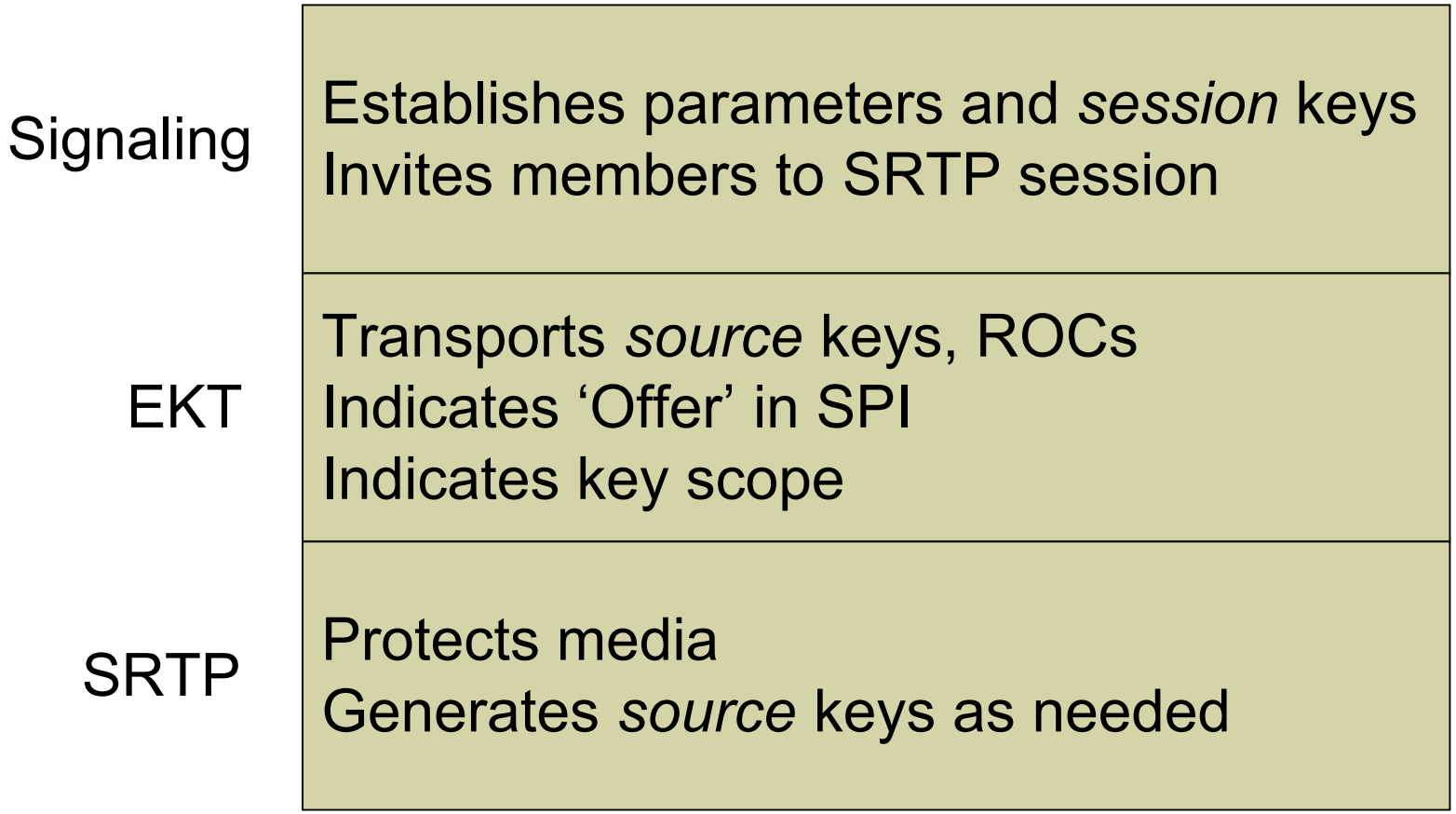
Decrypt

Master key
ROC
Initial SEQ

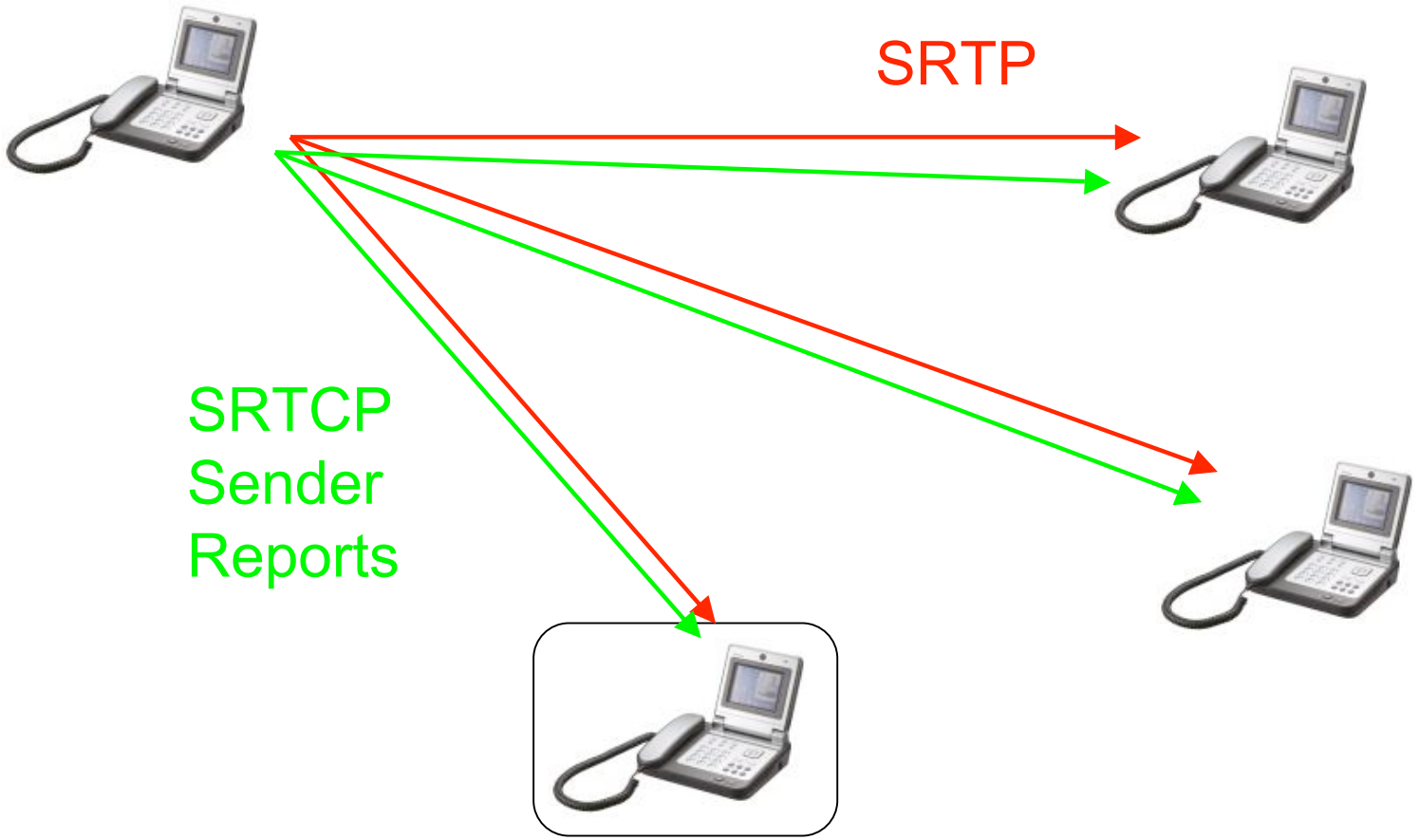
Authentication Tag Format



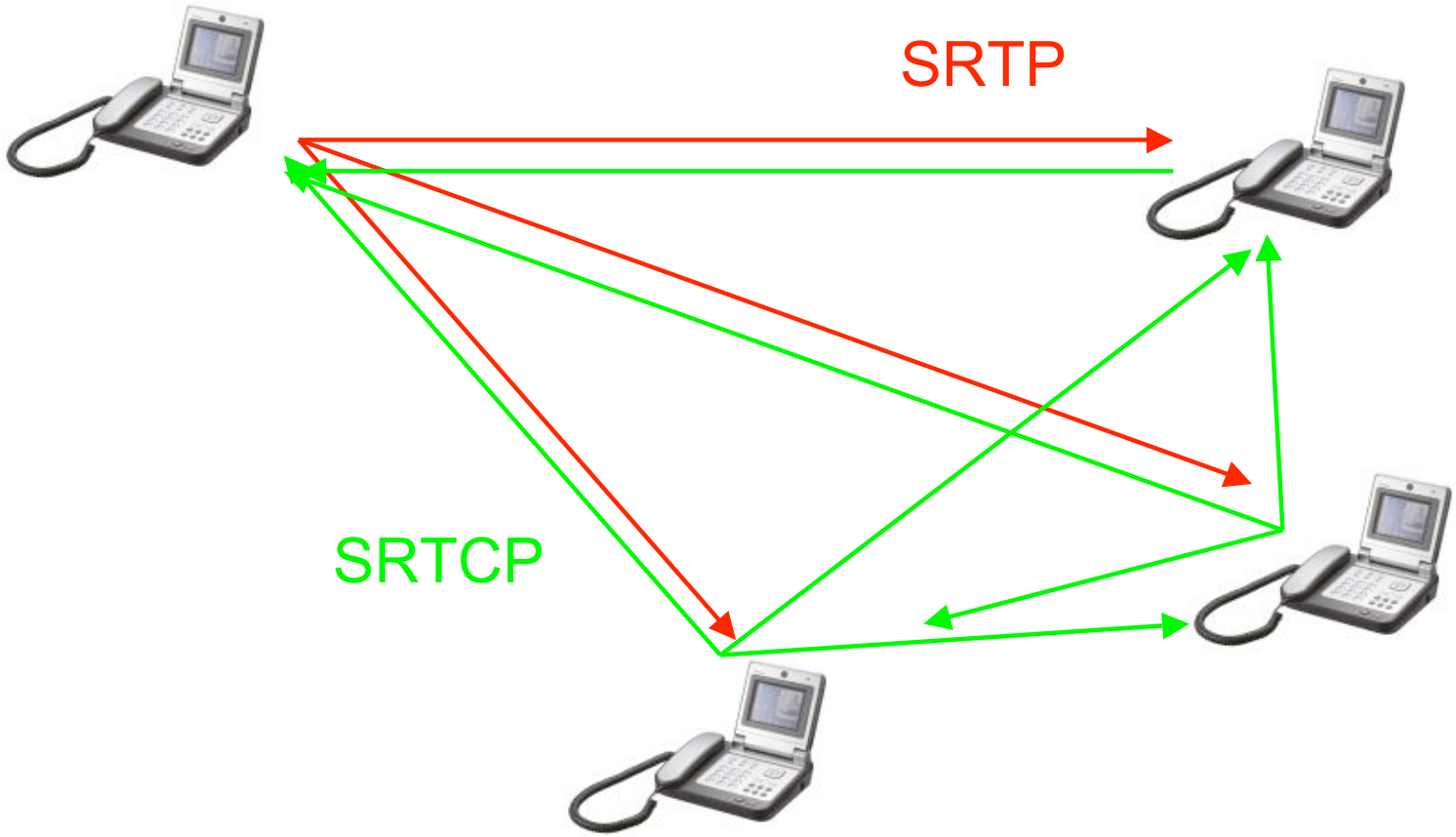
Architectural View



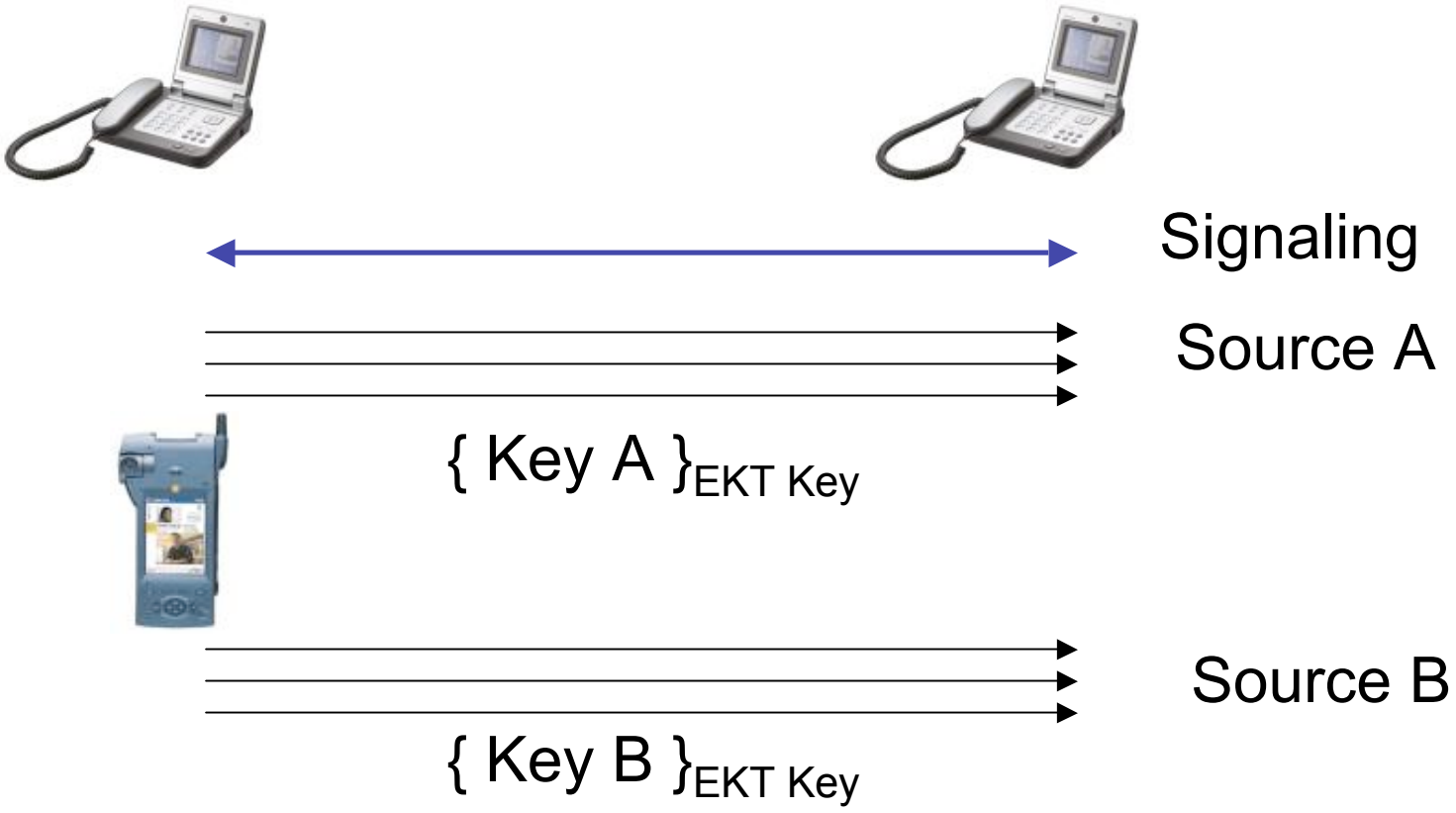
Late Joiners



Scalability



New RTP Sources



Benefits

- EKT decouples SRTP from signaling
  - Allows endpoints to start up (or rekey) SRTP sources at will
  - Allows participants to join sessions that are already in progress
  - No central coordination of ROCs, SSRCs, or SRTP's per-source master keys required
  - Allows SRTP to indicate cryptosuite
    - Solves SIP Early Media problem without Preconditions

Benefits (continued)

- High scalability
  - SRTCP receiver reports ‘carry own keys’
- Can work with any SRTP keying system
  - Transports keys rather than setting them
- No extra round trips
- Benefits multiparty RTP
  - SIP parallel forking

EKT Limitations

- Requires SRTCP
  - Could be extended to use SRTP
- Requires EKT secret keys established through out-of-band means
  - Could be extended to work with Diffie-Hellman
- Provides *group* security after SIP parallel fork
  - But meets all SRTP security requirements
- Adds ~ 24 bytes to each SRTCP packet
- No parameter negotiation

Future Work

- Standards track?
- Implement in `libsrtp`
  - `mpeg4ip` integration
- Incorporate feedback
- Extend MIKEY bindings
- Define bindings to DTLS-RTP and/or SDP DH
- SRTP transport method