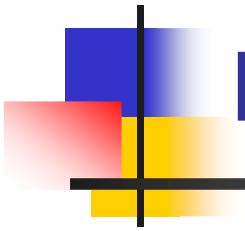
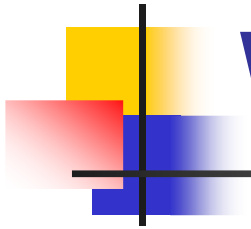


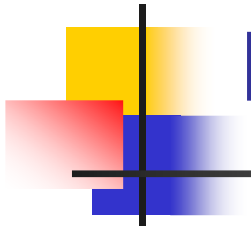
MIKEYv2

ldondeti@qualcomm.com



Why MIKEYv2?

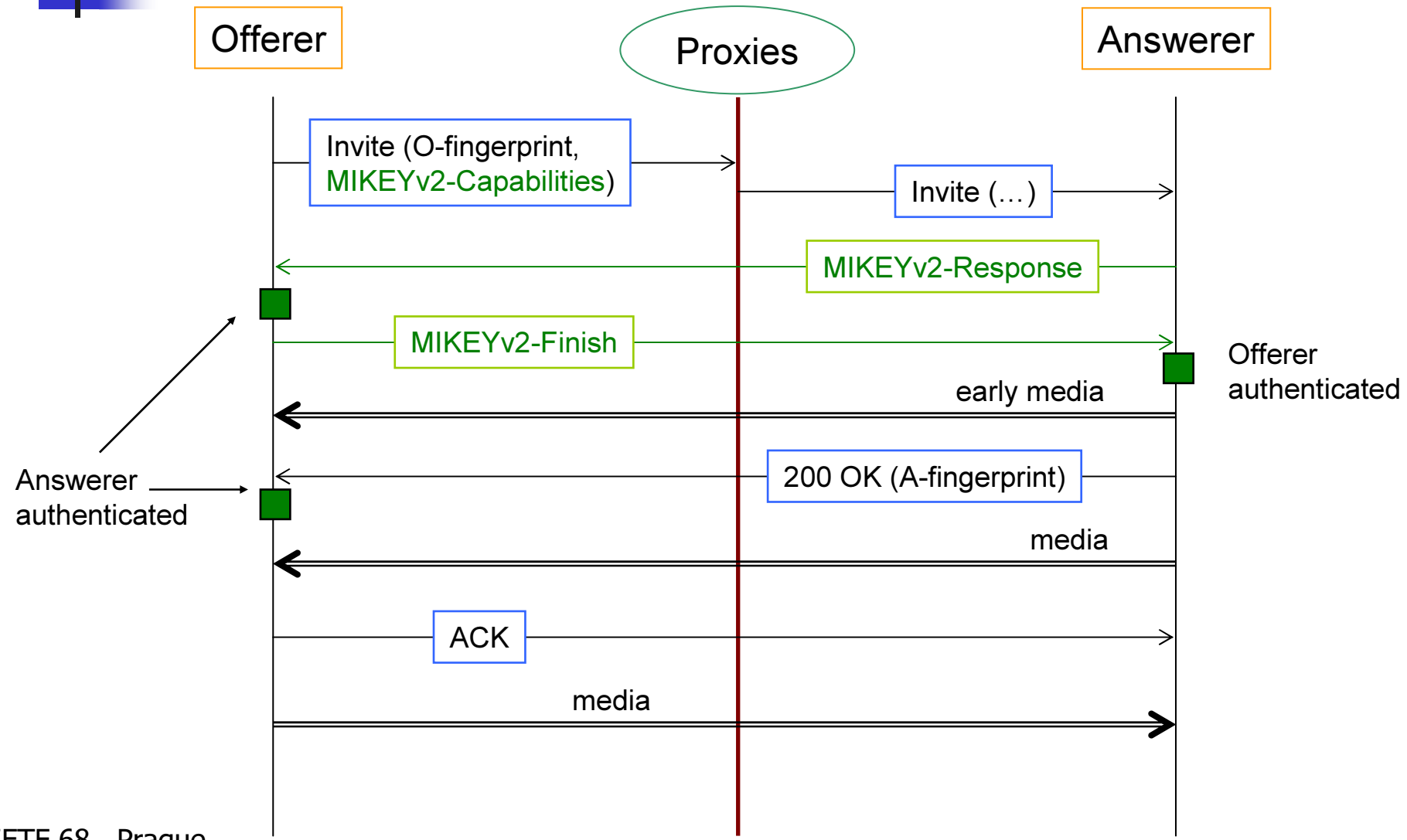
- MIKEY is an efficient key management protocol designed for SRTP keying
- Used for broadcast keying in 3GPP and OMA
- Easy to add crypto algorithm and mode negotiation
- MIKEYv2 finishes in 1 RT in the media path
- The code and design reuse argument applies to MIKEY as much as it applies to TLS
  - If some devices end up needing to implement MIKEYv2 in the smartcard, code reuse may become an important consideration



Properties of MIKEYv2

- Peer-to-peer key management protocol
- May be run in the signaling or media path
- Finishes in 1 RT in the media path
- Runs over UDP or multiplexed with RTP/RTCP
- Supports crypto algorithm and mode negotiation
- Supports unicast and broadcast keying
- Establishes SRTP crypto context for multiple media streams
- Can amortize the cost of public key operations over multiple sessions
- Solves hunger, poverty, cancer, AIDS and other assorted problems of the world

MIKEYv2 Message Flow

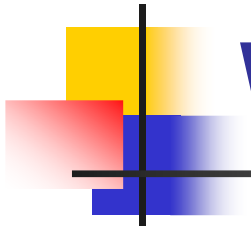


MIKEYv2 Capabilities Message

- O → A: HDR, RANDi, CAP, IDi, CERTi, [IDr]
- The CAP payload is new
- The Initiator lists the crypto algorithms and MIKEY modes it supports
- The Responder and the Initiator are expected to include this message in MAC calculation

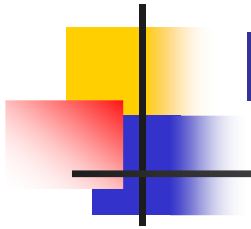
MIKEYv2 Messages in Media Path

- MIKEYv2-PSK
  - O ← A: HDR, RANDi, RANDr, IDr, {SP}, KEMAC
  - O → A: HDR, RANDi, RANDr, [SP], V
- MIKEYv2-RSA
  - O ← A: HDR, RANDi, RANDr, IDr, [CERTr], {SP}, KEMAC, PKE, SIGNr
  - O → A: HDR, RANDi, RANDr, [SP], V
- MIKEYv2-DH
  - O ← A: HDR, RANDi, RANDr, [IDr|CERTr], {SP}, DHr, SIGNr
  - O → A: HDR, RANDi, RANDr, [SP], DHi, SIGNi
- MIKEYv2-DHHMAC
  - O ← A: HDR, RANDi, RANDr, IDr, {SP}, DHr, KEMACi
  - O → A: HDR, RANDi, RANDr, [SP], DHi, KEMACi



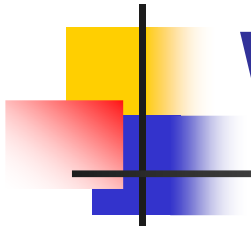
Why Can't We Use DTLS-SRTP/ZRTP?

- Have I already said we should build on MIKEY? ☺
- Desirable to use a single protocol for unicast and group keying?
- Moving forward, not quite comfortable with some of the notions in the other choices
  - Does separation of keying and data encapsulation with TLS work well?
  - Is the client-server paradigm of TLS not an issue for peer-to-peer operation?
  - ZRTP is too chatty!
  - It is not clear when ZRTP finishes
    - Too many messages for SRTP establishment?



If We Must Use One of the Others

- As few RTs as possible please
- Support initiation via SDP
- Make the protocol more peer-to-peer
- Support for multiple sessions
- I do think MIKEYv2 is the best option
  - But I can always pitch it to the WHO! ☺
    - Solves hunger, poverty, cancer, AIDS and other assorted problems of the world



What do I Mean Peer-to-Peer?

- DTLS-SRTP is a client-server protocol
  - Q: If new media sessions are initiated by the original answerer/client, would a new DTLS session be needed?
- Many TLS extensions take this mode of operation into account
  - e.g., TLS session resumption