# EAP Efficient Re-authentication

Lakshminath Dondeti, ldondeti@qualcomm.com
Vidya Narayanan, vidyan@qualcomm.com

IETF68; March 2007

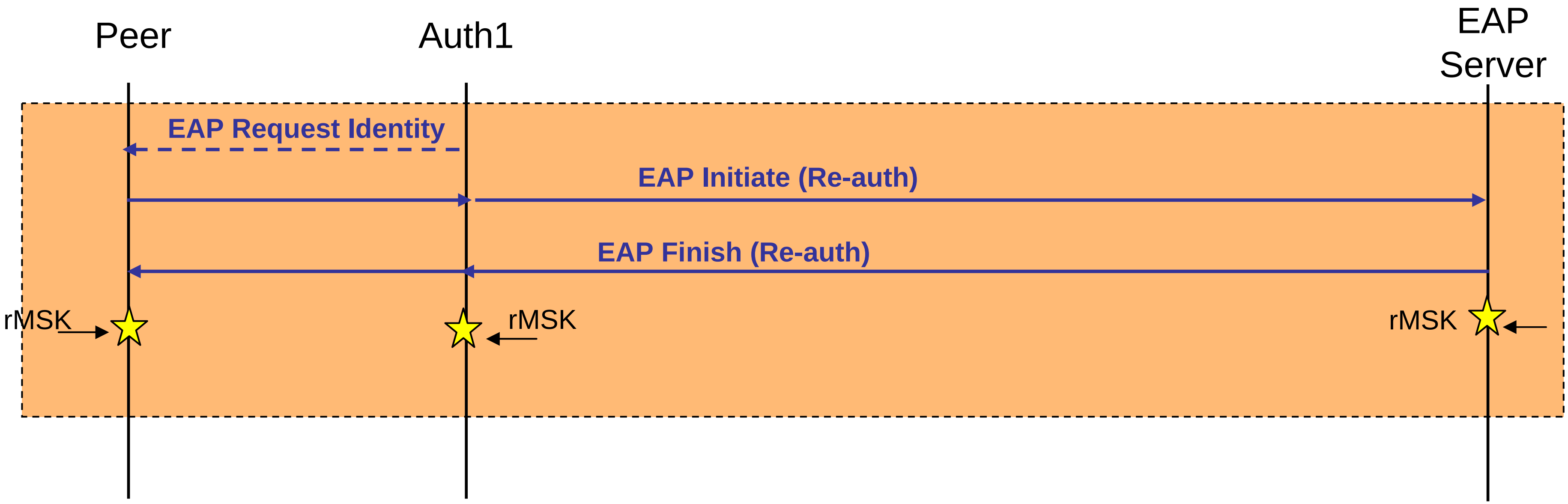
# Re-auth Goals

- MUST be better than full EAP authentication
  - "The protocol MUST be responsive to handover and re-authentication latency performance within a mobile access network"
- EAP lower layer independence
- EAP method independence
- AAA protocol compatibility and keying
- Co-existence with current EAP operation

# Re-authentication – Consensus so far

- The root of the HOKEY key hierarchy comes from the EMSK hierarchy
- The re-authentication protocol will be carried in native EAP
  - No support for EAP method-based transport
- Local domain support for HOKEY?

# EAP-ER Operation

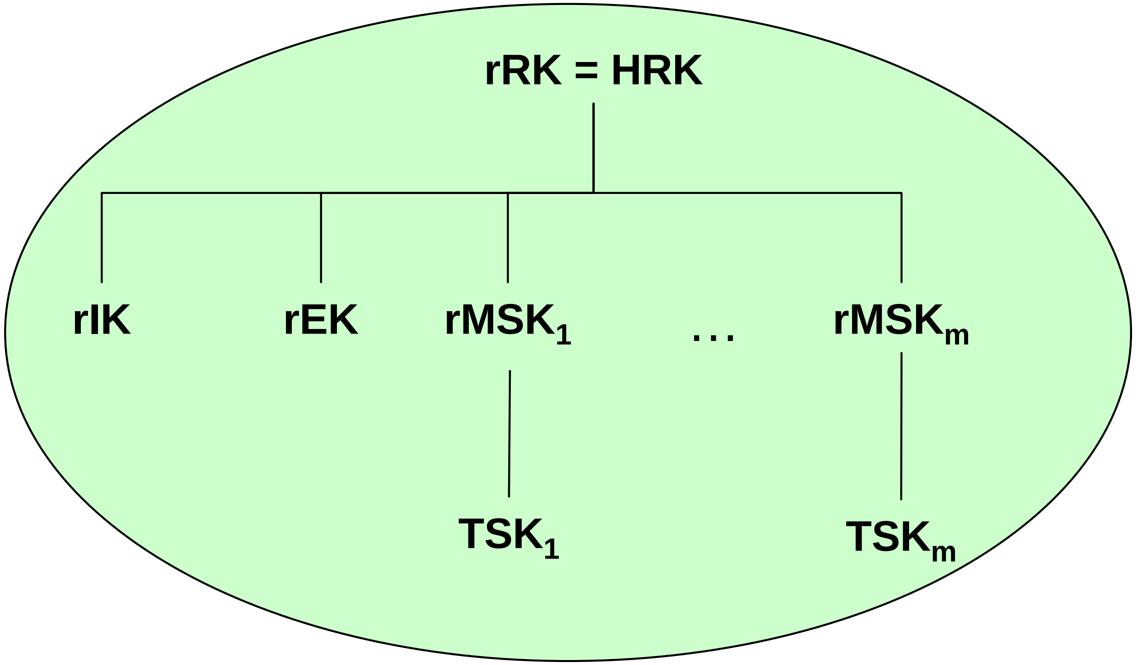



- The most optimal method of re-authentication is the peer-initiated model
- Optional server-initiated model
  - EAP Request Identity from the Authenticator to the peer serves a trigger for Re-Auth
- The Peer authenticates first
  - Uses temporary identity or a key identity for identity protection
- The Finish message contains Server's authentication and also serves the same purpose as EAP Success
- To support peer-initiated operation, changes to peer's state machine are needed
  - Peer must be able to maintain retransmission timers

# Local Re-auth Server

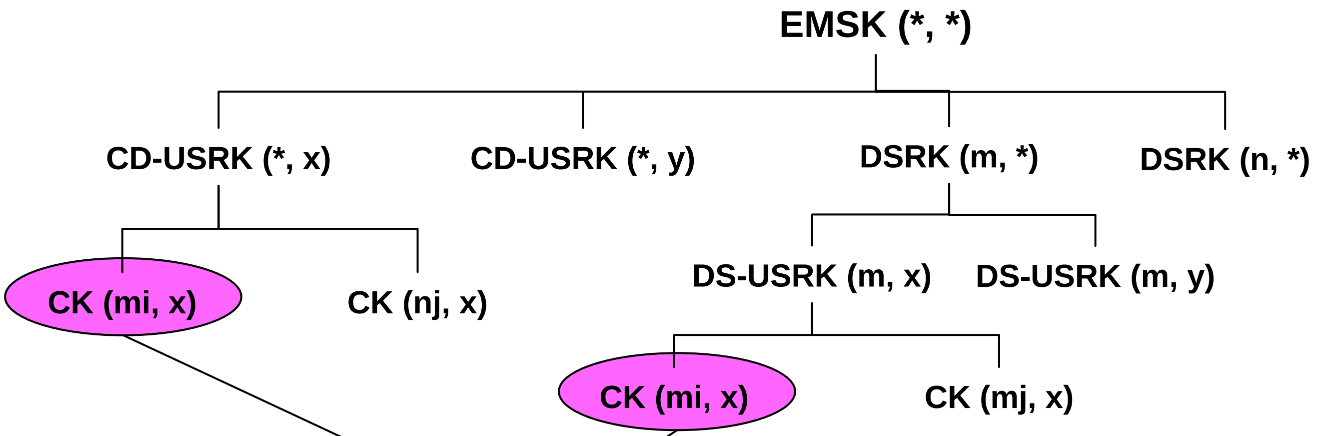
- Re-auth may still take too long if the AS is too many hops away
- Must be able to perform re-auth with a local server when handing off within a local area
- Key hierarchy must support both models
- The re-auth protocol must support some bootstrapping capability
  - Local server must be provided a key
  - Peer may need to be provided a server ID

# Re-authentication Key Hierarchy




- rRK is the Re-authentication Root Key
- rIK is the Re-auth Integrity Key and used to provide proof of possession of Re-auth keys
- rEK is the Encryption Key used to encrypt any confidential data exchanged between the peer and the EAP-ER server
- rMSK is the MSK equivalent key
  - Derived based on the run of the EAP-ER protocol
  - Each Authenticator change, whether or not an Authenticator is revisited, is treated the same

## Relation to EMSK Key Hierarchy

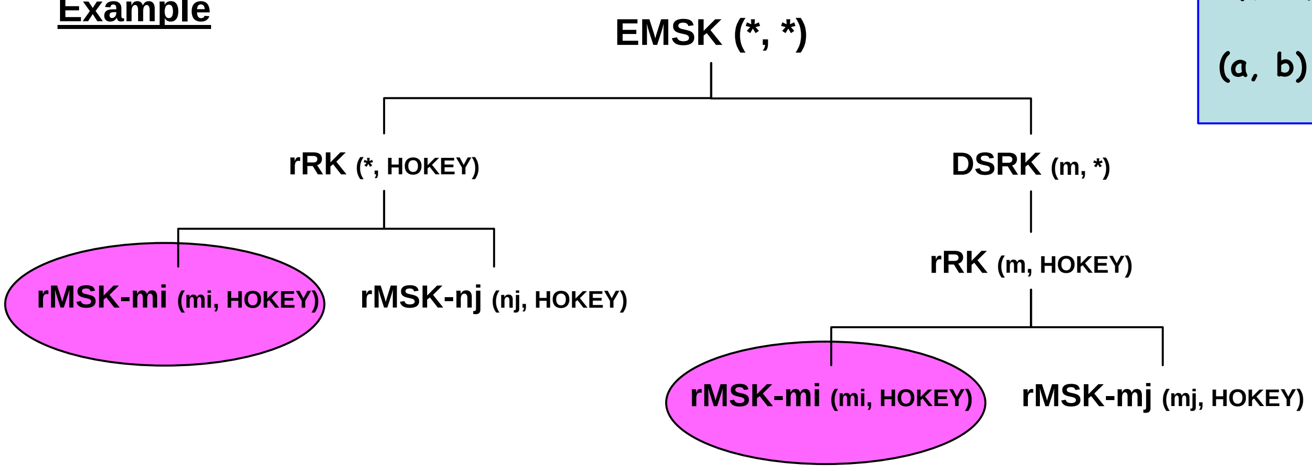



CKs for a given entity (mi – entity 'i' in domain 'm') can be derived either from CD-USRK or DSRK hierarchy

DRK: Domain Root Key
DSRK: Domain-Specific Root Key
USRK: Usage-Specific Root Key
CD-USRK: Cross-Domain USRK
DS-USRK: Domain-Specific USRK
CK: Cryptographic Usage Key

(a, b) → Scope = a; Context = b

## Example



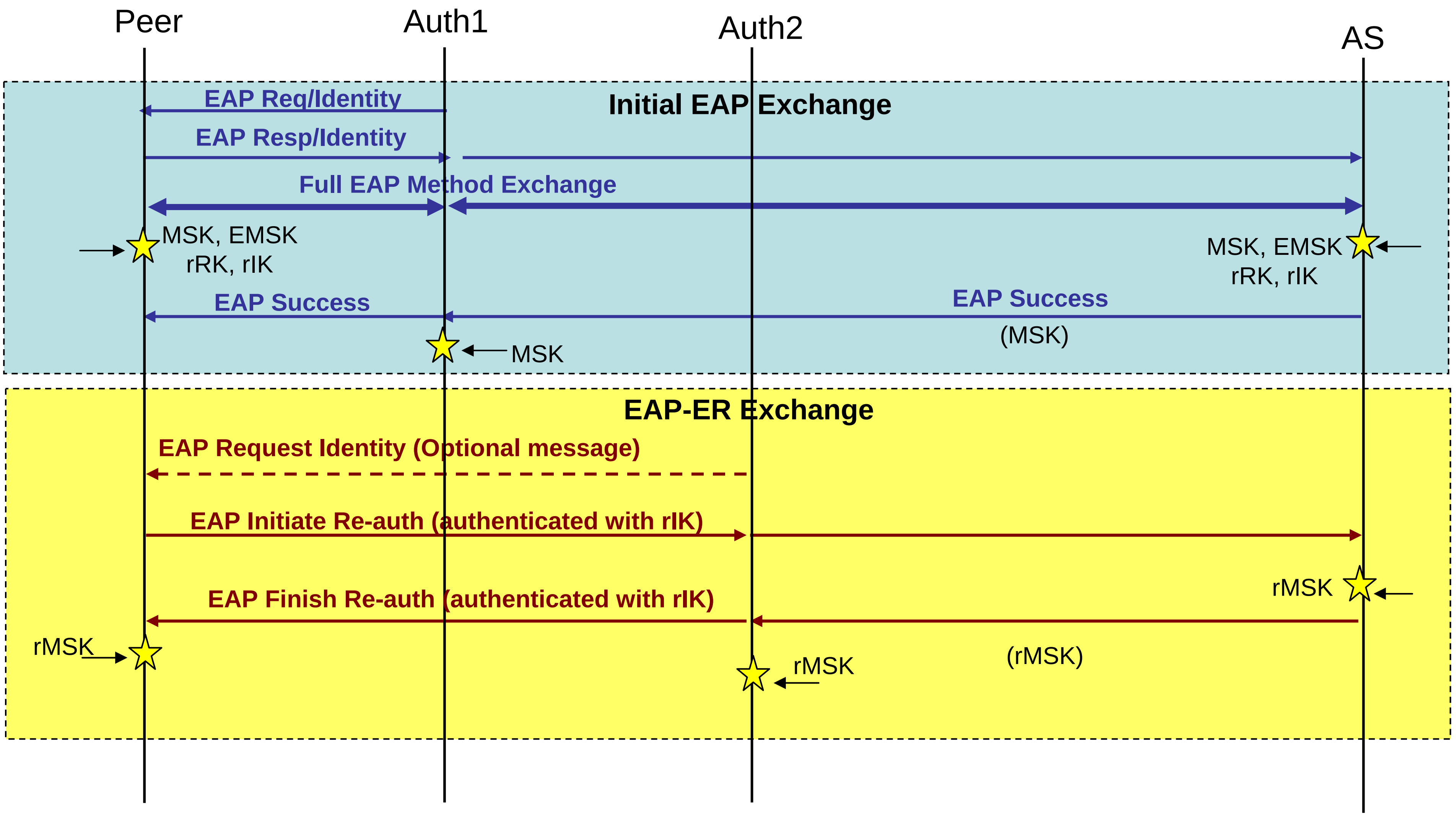
# Lower-layer Support

- For optimal operation, the lower layer may
  - advertise re-auth capability
    - Alternatively, peer may fail re-auth and attempt full EAP
  - advertise a local re-auth server
    - Server ID may be obtained from the lower layer at the peer
      - Peer may not need to be “bootstrapped” at the EAP layer
- Key for the local server may be delivered along with the full EAP exchange
  - Alternatively, key may be bootstrapped by an explicit EAP-ER bootstrap exchange

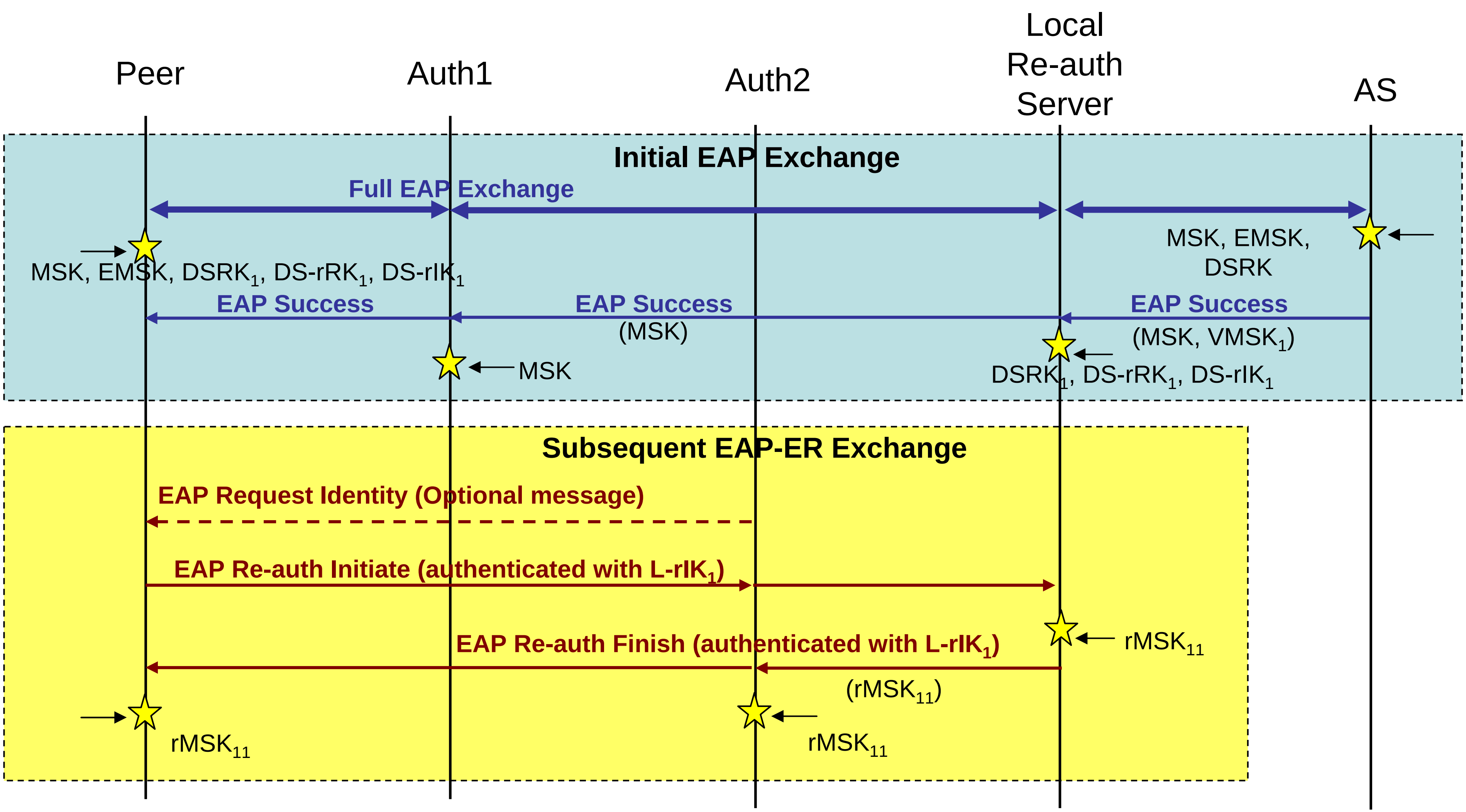
# EAP-ER Summary

- Method-independent protocol for efficient re-authentication
  - EAP-ER is a single roundtrip re-authentication protocol
  - Access agnostic; can be used for inter-technology handoffs
  - Proof of possession of key material of an earlier authentication
  - EAP-ER execution with a local server
- Key Generation in EAP-ER
  - rRK is the root of the hierarchy
    - May be generated from the EMSK or DSRK
  - Re-authentication MSKs (rMSK)
    - Serves the same purpose as an MSK

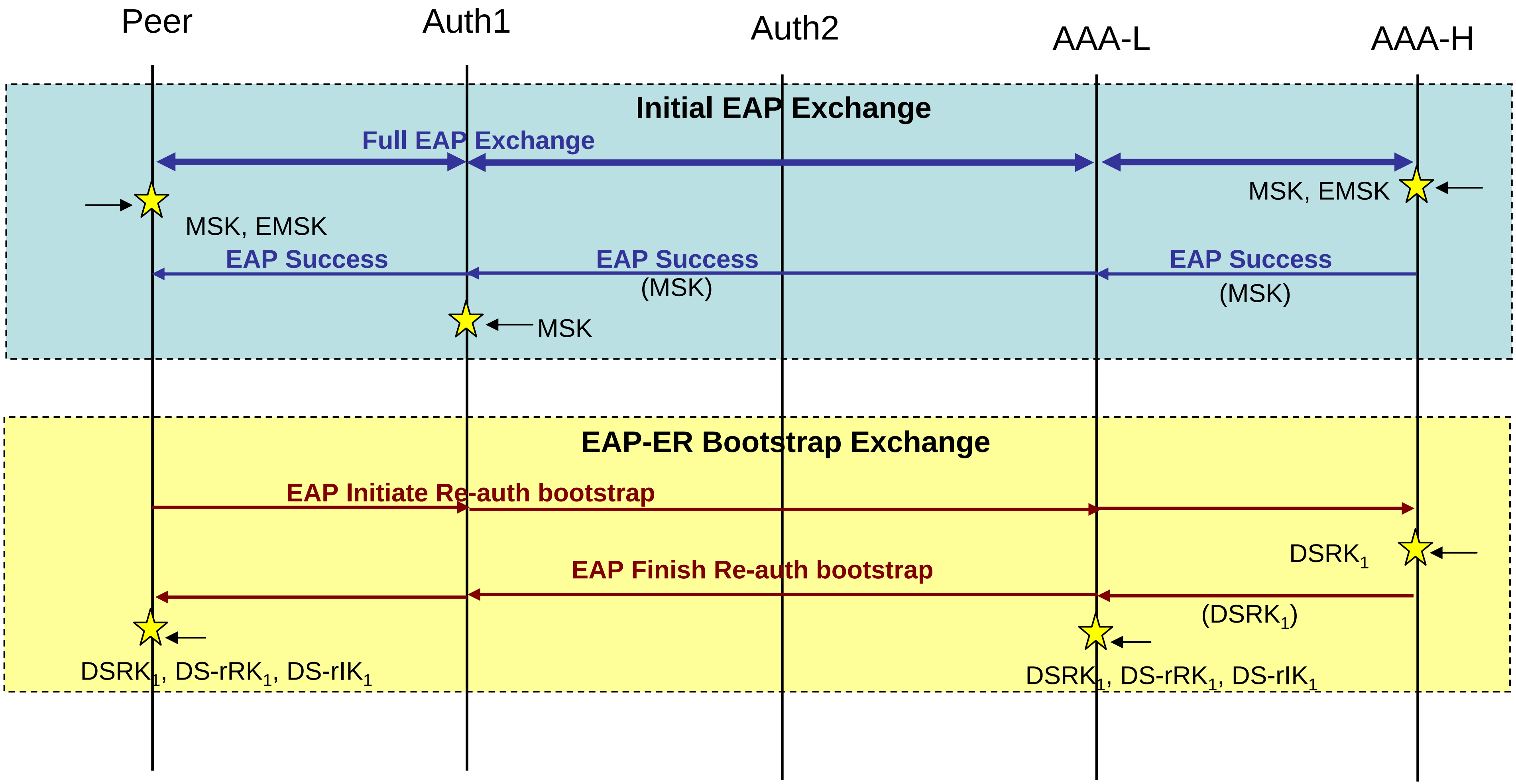
# EAP-ER Exchange with AS (EAP Server)



# EAP-ER Exchange with Local Re-auth Server



# EAP-ER Bootstrap Exchange



# Backup Slides

# EAP Re-auth Packet format

| | | |
|---|---|---|
| Code | Identifier | Length |
| Type | Flags | SEQ |
| 1 or more TVs or TLVs containing identities | | |
| Crypto-Suite | Authentication Tag (variable) | |
| Authentication Tag (contd) | | |

| | | |
|---|---|---|
| Type | Length | Value (variable length) |
| Value (contd) | | |

# EAP-ER attributes

- Peer sends an EAP Initiate Re-auth message with
  - rIKname for key lookup and Proof of possession verification
  - server-id (optional)
  - Peer-id, NAI (optional)
    - If neither peer-id nor server-id are present, rIKname must be in the form of an NAI
  - Server/Peer Nonce (optional)
- Code indicates Initiate/Finish
- Flags indicate bootstrap or not
- SEQ for replay protection
- Crypto-suite indicates the algorithm used for integrity protection
- Authentication tag is the proof of possession of the rIK

# Key derivation

- rRK = prf+ (K, S), where,
  - K = EMSK and
  - S = rRK Label
    - (“EAP Re-authentication Root Key”)

- rRK_name = NDF-64( EAP Session-ID, rRK Label )

- rIK = prf+ (rRK, "Re-authentication Integrity Key")

- rIK_name = prf-64 (rRK, "rIK Name")

- rMSK = prf+(rRK, SEQ)

# What is Low Latency?

- Security becomes a burden when any latency or overhead is added to the critical handoff path ☺
  - Mobile access networks resort to insecure practices when security adds latency to handoffs

- Two aspects of latency
  - Number of roundtrips
  - Distance to the AS

- Ideally, the protocol should be executable in parallel with connection establishment
  - I.e., add 0 incremental time to L2 handoffs

- It may also be unacceptable to have to go back to the AS (EAP Server) upon every handoff
  - EAP Server may be too many hops away!