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?
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

EMSK (*, *)

- CD-USRK (*, x)
  - CK (mi, x)
  - CK (nj, x)
- CD-USRK (*, y)
- DSRK (m, *)
  - DS-USRK (m, x)
  - DS-USRK (m, y)
- DSRK (n, *)
  - CK (mj, x)

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

Example

EMSK (*, *)

- rRK (*, HOKEY)
  - rMSK-mi (mi, HOKEY)
  - rMSK-nj (nj, HOKEY)
- DSRK (m, *)
  - rRK (m, HOKEY)
  - rMSK-mi (mi, HOKEY)
  - rMSK-mj (mj, HOKEY)

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
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)

Peer — Auth1 — Auth2 — AS

EAP Reg/Identity
EAP Resp/Identity
Full EAP Method Exchange
EAP Success

EAP Request Identity (Optional message)
EAP Initiate Re-auth (authenticated with rIK)
EAP Finish Re-auth (authenticated with rIK)

EAP Success (MSK)

MSK, EMSK rRK, rIK

rMSK (rMSK)

MSK, EMSK rRK, rIK

rMSK
EAP-ER Exchange with Local Re-auth Server

Peer

Auth1

Auth2

Local
Re-auth
Server

AS

Full EAP Exchange

MSK, EMSK, DSRK₁, DS-rRK₁, DS-rIK₁

EAP Success

EAP Success

EAP Success

MSK, EMSK, DSRK

EAP Request Identity (Optional message)

EAP Re-auth Initiate (authenticated with L-rIK₁)

EAP Re-auth Finish (authenticated with L-rIK₁)

EAP Success

(rMSK₁₁)

rMSK₁₁

rMSK₁₁

(rMSK₁₁)
EAP-ER Bootstrap Exchange

Initial EAP Exchange

Full EAP Exchange

MSK, EMSK

EAP Success

EAP Success (MSK)

MSK

EAP Initiate Re-auth bootstrap

EAP Finish Re-auth bootstrap

DSRK₁, DS-rRK₁, DS-rIK₁

DSRK₁, DS-rRK₁, DS-rIK₁

DSRK₁

(DSRK₁)
Backup Slides
# EAP Re-auth Packet format

<table>
<thead>
<tr>
<th>Code</th>
<th>Identifier</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Flags</td>
<td>SEQ</td>
</tr>
</tbody>
</table>

### 1 or more TVs or TLVs containing identities

<table>
<thead>
<tr>
<th>Crypto-Suite</th>
<th>Authentication Tag (variable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Tag (contd)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Value (variable length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value (contd)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

• \( rRK = \text{prf}^+ (K, S) \), where,
  – \( K = \text{EMSK} \) and
  – \( S = rRK \text{ Label} \)
    • ("EAP Re-authentication Root Key")

• \( rRK\_name = \text{NDF-64}(\text{EAP Session-ID, rRK Label}) \)

• \( rIK = \text{prf}^+ (rRK, "\text{Re-authentication Integrity Key}") \)

• \( rIK\_name = \text{prf-64}(rRK, "rIK Name") \)

• \( rMSK = \text{prf}^+(rRK, \text{SEQ}) \)
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!