Internet Engineering Task Force (IETF)
Request for Comments: 6630
Category: Standards Track
ISSN: 2070-1721

Z. Cao
H. Deng
China Mobile
Q. Wu
Huawei
G. Zorn, Ed.
Network Zen

June 2012

EAP Re-authentication Protocol Extensions for Authenticated Anticipatory Keying (ERP/AAK)

Abstract

The Extensible Authentication Protocol (EAP) is a generic framework supporting multiple types of authentication methods.

The EAP Re-authentication Protocol (ERP) specifies extensions to EAP and the EAP keying hierarchy to support an EAP method-independent protocol for efficient re-authentication between the peer and an EAP re-authentication server through any authenticator.

Authenticated Anticipatory Keying (AAK) is a method by which cryptographic keying material may be established upon one or more Candidate Attachment Points (CAPs) prior to handover. AAK uses the AAA infrastructure for key transport.

This document specifies the extensions necessary to enable AAK support in ERP.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Extensible Authentication Protocol (EAP) [RFC3748] is a generic framework supporting multiple types of authentication methods. In systems where EAP is used for authentication, it is desirable not to repeat the entire EAP exchange with another authenticator. The EAP Re-authentication Protocol (ERP) [RFC5296] specifies extensions to EAP and the EAP keying hierarchy to support an EAP method-independent protocol for efficient re-authentication between the EAP re-authentication peer and an EAP re-authentication server through any authenticator. The re-authentication server may be in the home network or in the local network to which the mobile host (i.e., the EAP re-authentication peer) is connecting.

Authenticated Anticipatory Keying (AAK) [RFC5836] is a method by which cryptographic keying material may be established upon one or more Candidate Attachment Points (CAPs) prior to handover. AAK utilizes the AAA infrastructure for key transport.

This document specifies the extensions necessary to enable AAK support in ERP.

2. Terminology

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Acronyms

The following acronyms are used in this document; see the references for more details.

AAA
    Authentication, Authorization, and Accounting [RFC3588]

CAP
    Candidate Attachment Point [RFC5836]

DSRK
    Domain-Specific Root Key [RFC5295]

EA
    Abbreviation for "ERP/AAK"

EA Peer
An EAP peer that supports the ERP/AAK. Note that all references to "peer" in this document imply an EA peer, unless specifically noted otherwise.

NAI
Network Access Identifier [RFC4282]

pMSK
pre-established Master Session Key

pRK
pre-established Root Key

rIK
re-authentication Integrity Key [RFC5296]

rRK
re-authentication Root Key [RFC5296]

SAP
Serving Attachment Point [RFC5836]

3. ERP/AAK Description

ERP/AAK is intended to allow (upon request by the peer) the establishment of cryptographic keying materials on a single Candidate Attachment Point prior to the arrival of the peer at the Candidate Access Network (CAN).

In this document, ERP/AAK support by the peer is assumed. Also, it is assumed that the peer has previously completed full EAP authentication and that either the peer or the SAP knows the identities of neighboring attachment points. Note that the behavior of a peer that does not support the ERP-AAK scheme defined in this specification is out of the scope of this document. Figure 1 shows the general protocol exchange by which the keying material is established on the CAP.

```
     +------+         +-----+        +-----+          +-----------+
     | Peer |         | SAP |        | CAP |          | EA Server |
     +--+---+         +--+--+        +--+--+          +-----+-----+
        |                |              |                   |
     a. | [EAP-Initiate/ |              |                   |
        | Re-auth-start  |              |                   |
        | (E flag)]      |              |                   |
        |<---------------|              |                   |
        |                |              |                   |
     b. | EAP-Initiate/  |              |                   |
        | Re-auth        |              |                   |
        | (E flag)       |              |                   |
        |--------------->|              |                   |
     c. |                | AAA(EAP-Initiate/Re-auth(E flag))|
        |                |--------------------------------->|
        |                |              |         +---------+---------+
        |                |              |         | CA authorized &   |
     d. |                |              |         |  and EA Keying    |
        |                |              |         |   Distribution    |
        |                |              |         +---------+---------+
        |                |              |                   |
        |                |              |                   |
     f. |                | AAA (EAP-Finish/Re-auth(E flag)) |
        |                |<---------------------------------|
     g. | EAP-Finish/    |              |                   |
        | Re-auth(E flag)|              |                   |
        |<---------------|              |                   |
        |                |              |                   |
```

Figure 1: ERP/AAK Exchange

```
                  +-----------+               +---------+
                  |           |               |         |
                  | EA Server |               |   CAP   |
                  |           |               |         |
                  +-----|-----+               +----|----+
                        |                          |
                        |                          |
                        |    AAA Request (pMSK)    |
                     e.1|------------------------->|
                        |                          |
                        |                          |
                        |                          |
                        |  AAA Response (Success)  |
                     e.2|<-------------------------|
                        |                          |
                        |                          |
                        |                          |
```

Figure 2: Key Distribution for ERP/AAK

ERP/AAK reuses the packet format defined by ERP, but specifies a new flag to differentiate EAP early authentication from EAP re-authentication. The peer initiates ERP/AAK without an external trigger, or initiates ERP/AAK in response to an EAP-Initiate/Re-Auth-Start message from the SAP.

In the latter case, the SAP MAY send the identity of one or more Candidate Attachment Points to which the SAP is adjacent to the peer in the EAP-Initiate/Re-auth-Start message (see step a in Figure 1). The peer SHOULD override the identity of CAP(s) carried in the EAP-Initiate/Re-auth-Start message by sending EAP-Initiate/Re-auth with the E flag set if it knows to which CAP it will move. If the EAP-Initiate/Re-auth-Start packet is not supported by the peer, it MUST be silently discarded.

If the peer initiates ERP/AAK, the peer MAY send an early-authentication request message (EAP-Initiate/Re-auth with the E flag set) containing the keyName-NAI, the CAP-Identifier, rIK, and sequence number (see step b in Figure 1). The realm in the keyName-NAI field is used to locate the peer’s ERP/AAK server. The CAP-Identifier is used to identify the CAP. The re-authentication Integrity Key (rIK) is defined by Narayanan & Dondeti in [RFC5296] and is used to protect the integrity of the message. The sequence number is used for replay protection.

The SAP SHOULD verify the integrity of this message at step b. If this verification fails, the SAP MUST send an EAP-Finish/Re-auth message with the Result flag set to ’1’ (Failure). If the verification succeeds, the SAP SHOULD encapsulate the early-authentication message into a AAA message and send it to the peer’s ERP/AAK server in the realm indicated in the keyName-NAI field (see step c in Figure 1).

Upon receiving the message, the ERP/AAK server MUST first use the keyName indicated in the keyName-NAI to look up the rIK and check the integrity and freshness of the message. Then, the ERP/AAK server MUST verify the identity of the peer by checking the username portion of the KeyName-NAI. If any of the checks fail, the server MUST send an early-authentication finish message (EAP-Finish/Re-auth with E flag set) with the Result flag set to ’1’. Next, the server MUST authorize the CAP specified in the CAP-Identifier TLV. In the success case, the server MUST derive a pMSK from the pRK for the CAP carried in the CAP-Identifier field using the sequence number associated with CAP-Identifier as an input to the key derivation. (see step d in Figure 1).

Then, the ERP/AAK server MUST transport the pMSK to the authorized CAP via AAA (see Section 7) as illustrated above (see steps e.1 and e.2 in Figure 2). Note that key distribution in Figure 2 is one part of step d in Figure 1.

Finally, in response to the EAP-Initiate/Re-auth message, the ERP/AAK server SHOULD send the early-authentication finish message (EAP--Finish/Re-auth with E flag set) containing the identity of the authorized CAP to the peer via the SAP along with the lifetime of the pMSK. If the peer also requests the rRK Lifetime, the ERP/AAK server SHOULD send the rRK Lifetime in the EAP-Finish/Re-auth message (see steps f and g in Figure 1).

4. ERP/AAK Key Hierarchy

ERP/AAK uses a key hierarchy similar to that of ERP. The ERP/AAK pre-established Root Key (pRK) is derived from either the EMSK or the DSRK as specified below (see Section 4.1). In general, the pRK is derived from the EMSK if the peer is located in the home AAA realm and derived from the DSRK if the peer is in a visited realm. The DSRK is delivered from the EAP server to the ERP/AAK server as specified in [KEYTRAN]. If the peer has previously been authenticated by means of ERP or ERP/AAK, the DSRK SHOULD be directly reused.

```
                  DSRK    EMSK
                    |       |
                +---+---+---+---+
                |
               pRK            ...
```

Figure 3: ERP/AAK Root Key Derivation

Similarly, the pre-established Master Session Key (pMSK) is derived from the pRK. The pMSK is established for the CAP when the peer early authenticates to the network. The hierarchy relationship is illustrated Figure 4, below.

```
                     pRK
                      |
              +--------+--------+
              |
              pMSK             ...
```

Figure 4: ERP/AAK Key Hierarchy

4.1. Derivation of the pRK and pMSK

The rRK is derived as specified in [RFC5295].

pRK = KDF (K, S), where

K = EMSK or K = DSRK and

S = pRK Label | "\0" | length

The pRK Label is an IANA-assigned 8-bit ASCII string:

EAP Early-Authentication Root Key@ietf.org

assigned from the "User Specific Root Keys (USRK) Key Labels" name space in accordance with Salowey, et al. [RFC5295]. The KDF and algorithm agility for the KDF are also defined in RFC 5295. The KDF algorithm is indicated in the cryptosuite field or list of cryptosuites TLV payload as specified in Sections 5.2 and 5.3.

The pMSK uses the same KDF as pRK and is derived as follows:

pMSK = KDF (K, S), where

K = pRK and

S = pMSK label | "\0" | SEQ | length

The pMSK label is the 8-bit ASCII string:

EAP Early-Authentication Master Session Key@ietf.org

The length field refers to the length of the pMSK in octets encoded as specified in RFC 5295. SEQ is sent by either the peer or the server in the ERP/AAK message using the SEQ field or the Sequence number TLV. It is encoded as a 16-bit number as specified in Sections 5.2 and 5.3.

5. Packet and TLV Extension

This section describes the packet and TLV extensions for the ERP/AAK exchange.

5.1. EAP-Initiate/Re-auth-Start Packet and TLV Extension

Figure 5 shows the new parameters contained in the EAP-Initiate/Re-auth-Start packet defined in [RFC5296].

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Code      |  Identifier   |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |E| Reserved    |     1 or more TVs or TLVs     ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: EAP-Initiate/Re-auth-Start Extension

Flags

'E' - The E flag is used to indicate early authentication. This field MUST be set to '1' if early authentication is in use, and it MUST be set to '0' otherwise.

The rest of the 7 bits (Reserved) MUST be set to 0 and ignored on reception.

Type/Values (TVs) and TLVs

CAP-Identifier: Carried in a TLV payload. The format is identical to that of a DiameterIdentity [RFC3588]. It is used by the SAP to advertise the identity of the CAP to the peer. Exactly one CAP-Identifier TLV MAY be included in the EAP-Initiate/Re-auth-Start packet if the SAP has performed CAP discovery.

If the EAP-Initiate/Re-auth-Start packet is not supported by the peer, it SHOULD be discarded silently.

5.2. EAP-Initiate/Re-auth Packet and TLV Extension

Figure 6 illustrates the new parameters contained in the EAP-Initiate/Re-auth packet defined in [RFC5296].

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Code      |  Identifier   |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |R|x|L|E|Resved |             SEQ               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 1 or more TVs or TLVs                         ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Cryptosuite  |        Authentication Tag                      ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: EAP-Initiate/Re-auth Extension

Flags

'x' - The x flag is reserved. It MUST be ignored on receipt.

'L' - As defined in Section 5.3.2 of [RFC5296], this bit is used to request the key lifetimes from the server.

'E' - The E flag is used to indicate early authentication.

The first bit(R) and final 4 bits (Resved) MUST be set to 0 and ignored on reception.

SEQ

As defined in Section 5.3.2 of [RFC5296], this field is 16-bit sequence number and used for replay protection.

TVs and TLVs

keyName-NAI: As defined in [RFC5296], this is carried in a TLV payload. The Type is 1. The NAI is variable in length, not exceeding 253 octets. The username part of the NAI is the EMSKname used to identify the peer. The realm part of the NAI is the peer's home domain name if the peer communicates with the home EA server or the domain to which the peer is currently attached (i.e., local domain name) if the peer communicates with a local EA server. The

SAP knows whether the KeyName-NAI carries the local domain name by comparing the domain name carried in the KeyName-NAI with the local domain name that is associated with the SAP. Exactly one keyName-NAI attribute SHALL be present in an EAP-Initiate/Re-auth packet and the realm part of it SHOULD follow the use of internationalized domain names defined in [RFC5890].

CAP-Identifier: Carried in a TLV payload. The Type is 11. This field is used to indicate the Fully Qualified Domain Name (FQDN) of a CAP. The value field MUST be encoded as specified in Section 8 of [RFC3315]. Exactly one instance of the CAP-Identifier TLV MUST be present in the ERP/AAK-Key TLV.

Sequence number: The Type is 7. The value field is a 16-bit field and used in the derivation of the pMSK for a CAP.

Cryptosuite

This field indicates the integrity algorithm used for ERP/AAK. Key lengths and output lengths are either indicated or obvious from the cryptosuite name, e.g., HMAC-SHA256-128 denotes Hashed Message Authentication Code (HMAC) computed using the SHA-256 function [RFC4868] with 256-bit key length and the output truncated to 128 bits [RFC2104]. We specify some cryptosuites below:

0-1	RESERVED
2	HMAC-SHA256-128
3	HMAC-SHA256-256

HMAC-SHA256-128 is REQUIRED to implement, and it SHOULD be enabled in the default configuration.

Authentication Tag

This field contains an integrity checksum over the ERP/AAK packet from the first bit of the Code field to the last bit of the Cryptosuite field, excluding the Authentication Tag field itself. The value field is calculated using the integrity algorithm indicated in the Cryptosuite field and rIK specified in [RFC5296] as the secret key. The length of the field is indicated by the Cryptosuite.

The peer uses the Authentication Tag to determine the validity of the EAP-Finish/Re-auth message from the server.

If the message doesn't pass verification or the Authentication Tag is not included in the message, the message SHOULD be discarded silently.

If the EAP-Initiate/Re-auth packet is not supported by the SAP, it SHOULD be discarded silently. The peer MUST maintain retransmission timers for reliable transport of the EAP-Initiate/Re-auth message. If there is no response to the EAP-Initiate/Re-auth message from the server after the necessary number of retransmissions (see Section 6), the peer MUST assume that ERP/AAK is not supported by the SAP.

5.3. EAP-Finish/Re-auth Packet and TLV Extension

Figure 7 shows the new parameters contained in the EAP-Finish/Re-auth packet defined in [RFC5296].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |R|x|L|E|Resved |             SEQ               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 1 or more TVs or TLVs                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cryptosuite  |         Authentication Tag                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: EAP-Finish/Re-auth Extension

Flags

'R' - As defined in Section 5.3.3 of [RFC5296], this bit is used as the Result flag. This field MUST be set to '1' to indicate success, and it MUST be set to '0' otherwise.

'x' - The x flag is reserved. It MUST be ignored on receipt.

'L' - As defined in Section 5.3.3 of [RFC5296], this bit is used to request the key lifetimes from the server.

'E' - The E flag is used to indicate early authentication.

The final 4 bits (Resved) MUST be set to 0 and ignored on reception.

SEQ

As defined in Section 5.3.3 of [RFC5296], this field is a 16-bit sequence number and is used for replay protection.

TVs and TLVs

keyName-NAI: As defined in [RFC5296], this is carried in a TLV payload. The Type is 1. The NAI is variable in length, not exceeding 253 octets. Exactly one keyName-NAI attribute SHALL be present in an EAP-Finish/Re-auth packet.

ERP/AAK-Key: Carried in a TLV payload for the key container. The Type is 8. Exactly one ERP/AAK-key SHALL be present in an EAP-Finish/Re-auth packet.

```
ERP/AAK-Key ::=
     { sub-TLV: CAP-Identifier }
     { sub-TLV: pMSK Lifetime }
     { sub-TLV: pRK Lifetime }
     { sub-TLV: Cryptosuites }
```

CAP-Identifier
Carried in a sub-TLV payload. The Type is 11 (less than 128). This field is used to indicate the identifier of the candidate authenticator. The value field MUST be encoded as specified in Section 8 of [RFC3315]. At least one instance of the CAP-Identifier TLV MUST be present in the ERP/AAK-Key TLV.

pMSK Lifetime
Carried in a sub-TLV payload of the EAP-Finish/Re-auth message. The Type is 10. The value field is an unsigned 32-bit field and contains the lifetime of the pMSK in seconds. This value is calculated by the server after performing the pRK Lifetime computation upon receiving the EAP-Initiate/Re-auth message. The rIK SHOULD share the same lifetime as the pMSK. If the 'L' flag is set, the pMSK Lifetime attribute MUST be present.

pRK Lifetime
Carried in a sub-TLV payload of EAP-Finish/Re-auth message. The Type is 9. The value field is an unsigned 32-bit field and contains the lifetime of the pRK in seconds. This value is calculated by the server before performing the pMSK Lifetime computation upon receiving a EAP-Initiate/Re-auth message. If the 'L' flag is set, the pRK Lifetime attribute MUST be present.

List of Cryptosuites

Carried in a sub-TLV payload. The Type is 5 [RFC5296]. The value field contains a list of cryptosuites (at least one cryptosuite SHOULD be included), each 1 octet in length. The allowed cryptosuite values are as specified in Section 5.2. The server SHOULD include this attribute if the cryptosuite used in the EAP-Initiate/Re-auth message was not acceptable and the message is being rejected. The server MAY include this attribute in other cases. The server MAY use this attribute to signal its cryptographic algorithm capabilities to the peer.

Cryptosuite

This field indicates the integrity algorithm and PRF used for ERP/AAK. HMAC-SHA256-128 is REQUIRED to implement, and it SHOULD be enabled in the default configuration. Key lengths and output lengths are either indicated or obvious from the cryptosuite name.

Authentication Tag

This field contains the integrity checksum over the ERP/AAK packet from the first bit of the Code field to the last bit of the Cryptosuite field, excluding the Authentication Tag field itself. The value field is calculated using the integrity algorithm indicated in the Cryptosuite field and the rIK [RFC5296] as the integrity key. The length of the field is indicated by the corresponding Cryptosuite.

The peer uses the authentication tag to determine the validity of the EAP-Finish/Re-auth message from a server.

If the message doesn’t pass verification or the authentication tag is not included in the message, the message SHOULD be discarded silently.

If the EAP-Initiate/Re-auth packet is not supported by the SAP, it is discarded silently. The peer MUST maintain retransmission timers for reliable transport of the EAP-Initiate/Re-auth message. If there is no response to the EAP-Initiate/Re-auth message from the server after the necessary number of retransmissions (see Section 6), the peer MUST assume that ERP/AAK is not supported by the SAP.

5.4. TV and TLV Attributes

With the exception of the rRK Lifetime and rMSK Lifetime TV payloads, the attributes specified in Section 5.3.4 of [RFC5296] also apply to this document. In this document, new attributes that may be present in the EAP-Initiate and EAP-Finish messages are defined as below:

- Sequence number: This is a TV payload. The Type is 7.

- ERP/AAK-Key: This is a TLV payload. The Type is 8.

- pRK Lifetime: This is a TV payload. The Type is 9.

- pMSK Lifetime: This is a TV payload. The Type is 10.

- CAP-Identifier: This is a TLV payload. The Type is 11.

6. Lower-Layer Considerations

Similar to ERP, some lower-layer specifications may need to be revised to support ERP/AAK; refer to Section 6 of [RFC5296] for additional guidance.

7. AAA Transport Considerations

The AAA transport of ERP/AAK messages is the same as that of the ERP message [RFC5296]. In addition, this document requires AAA transport of the ERP/AAK keying materials delivered by the ERP/AAK server to the CAP. Hence, a new AAA message for the ERP/AAK application should be specified to transport the keying materials.

8. Security Considerations

This section provides an analysis of the protocol in accordance with the AAA key management requirements specified in [RFC4962].

- Cryptographic algorithm independence: ERP-AAK satisfies this requirement. The algorithm chosen by the peer for calculating the authentication tag is indicated in the EAP-Initiate/Re-auth message. If the chosen algorithm is unacceptable, the EAP server returns an EAP-Finish/Re-auth message with a Failure indication.

- Strong, fresh session keys: ERP-AAK results in the derivation of strong, fresh keys that are unique for the given CAP. A pMSK is always derived on demand when the peer requires a key with a new CAP. The derivation ensures that the compromise of one pMSK does not result in the compromise of a different pMSK at any time.

- Limit key scope: The scope of all the keys derived by ERP-AAK is well defined. The pRK is used to derive the pMSK for the CAP. Different sequence numbers for each CAP MUST be used to derive a unique pMSK.

- Replay detection mechanism: For replay protection, a sequence number associated with the pMSK is used. The peer increments the sequence number by one after it sends an ERP/AAK message. The server sets the expected sequence number to the received sequence number plus one after verifying the validity of the received message, and it responds to the message.

- Authenticate all parties: The EAP Re-authentication Protocol provides mutual authentication of the peer and the server. The peer and SAP are authenticated via ERP. The CAP is authenticated and trusted by the SAP.

- Peer and authenticator authorization: The peer and authenticator demonstrate possession of the same keying material without disclosing it, as part of the lower-layer secure authentication protocol.

- Keying material confidentiality: The peer and the server derive the keys independently using parameters known to each entity.

- Uniquely named keys: All keys produced within the ERP context can be referred to uniquely as specified in this document.

- Prevent the domino effect: Different sequence numbers for each CAP MUST be used to derive the unique pMSK so that the compromise of one pMSK does not hurt any other CAP.

- Bind key to its context: The pMSKs are bound to the context in which the sequence numbers are transmitted.

- Confidentiality of identity: This is the same as with ERP [RFC5296].

- Authorization restriction: All the keys derived are limited in lifetime by that of the parent key or by server policy. Any domain-specific keys are further restricted to be used only in the domain for which the keys are derived. Any other restrictions of session keys may be imposed by the specific lower layer and are out of scope for this specification.

9. IANA Considerations

IANA has assigned five TLVs from the registry of EAP Initiate and Finish Attributes maintained at http://www.iana.org/assignments/eap-numbers/ with the following numbers:

- o Sequence number: This is a TV payload. The Type is 7.
- o ERP/AAK-Key: This is a TLV payload. The Type is 8.
- o pRK Lifetime: This is a TLV payload. The Type is 9.
- o pMSK Lifetime: This is a TLV payload. The Type is 10.
- o CAP-Identifier: This is a TLV payload. The Type is 11.

This document reuses the cryptosuites that were created for "Re-authentication Cryptosuites" in [RFC5296].

Further, IANA has added a new label in the "User Specific Root Keys (USRK) Key Labels" sub-registry of the "Extended Master Session Key (EMSK) Parameters" registry, as follows:

EAP Early-Authentication Root Key@ietf.org

A new registry for the flags in the EAP Initiate/Re-auth-Start message called the "EAP Initiate/Re-auth-Start Flags" has been created and a new flag (E) has been assigned as follows:

(E) 0x80

The rest of the values in the 8-bit field are reserved. New values can be assigned by Standards Action or IESG Approval [RFC5226].

A new registry for the flags in the EAP Initiate/Re-auth message called the "EAP Initiate/Re-auth Flags" has also been created. The following flags are reserved:

(R) 0x80 [RFC5296]

(B) 0x40 [RFC5296]

(L) 0x20 [RFC5296]

This document assigns a new flag (E) as follows:

(E) 0x10

The rest of the values in the 8-bit field are reserved. New values can be assigned by Standards Action or IESG Approval.

Further, this document creates a new registry for the flags in the EAP Finish/Re-auth message called the "EAP Finish/Re-auth Flags". The following values are assigned.

(R) 0x80 [RFC5296]

(B) 0x40 [RFC5296]

(L) 0x20 [RFC5296]

This document assigns a new flag (E) as follows:

(E) 0x10

The rest of the values in the 8-bit field are reserved. New values can be assigned by Standards Action or IESG approval.

10. Acknowledgements

In writing this document, Yungui Wang contributed to early versions of this document and we have received reviews from many experts in the IETF, including Tom Taylor, Tena Zou, Tim Polk, Tan Zhang, Semyon Mizikovsky, Stephen Farrell, Radia Perlman, Miguel A. Garcia, and Sujing Zhou. We apologize if we miss some of those who have helped us.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5295] Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)", RFC 5295, August 2008.

[RFC5296] Narayanan, V. and L. Dondeti, "EAP Extensions for EAP Re-authentication Protocol (ERP)", RFC 5296, August 2008.

11.2. Informative References

[KEYTRAN] Zorn, G., Wu, W., and V. Cakulev, "Diameter Attribute-Value Pairs for Cryptographic Key Transport", Work in Progress, August 2011.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, May 2007.

[RFC4962] Housley, R. and B. Aboba, "Guidance for Authentication, Authorization, and Accounting (AAA) Key Management", BCP 132, RFC 4962, July 2007.

[RFC5836] Ohba, Y., Wu, Q., and G. Zorn, "Extensible Authentication Protocol (EAP) Early Authentication Problem Statement", RFC 5836, April 2010.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

Authors' Addresses

Zhen Cao
China Mobile
53A Xibianmennei Ave., Xuanwu District
Beijing, Beijing 100053
P.R. China

EMail: zehn.cao@gmail.com

Hui Deng
China Mobile
53A Xibianmennei Ave., Xuanwu District
Beijing, Beijing 100053
P.R. China

EMail: denghui02@gmail.com

Qin Wu
Huawei
Floor 12, HuiHong Mansion, No. 91 BaiXia Rd.
Nanjing, Jiangsu 210001
P.R. China

Phone: +86 25 56623633
EMail: sunseawq@huawei.com

Glen Zorn (editor)
Network Zen
227/358 Thanon Sanphawut
Bang Na, Bangkok 10260
Thailand

Phone: +66 (0) 87-040-4617
EMail: glenzorn@gmail.com