Internet Draft

Paal Engelstad
Telenor R&D, California

Expires August 2002
February 2002

EAP over UDP (EAPoUDP)

<draft-engelstad-pana-eap-over-udp-00.txt>

Status of this Memo

This document is an individual submission for the PANA Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the mailing list pana@research.telcordia.com.

Abstract

This document specifies the Extensible Authentication Protocol over UDP (EAPoUDP) to be used for network access authentication. An access domain is represented by one or many PANA Authentication Agents (PAAs). Before a PANA Client (PaC) is granted access to the domain, a PAA and a PaC MAY use EAPoUDP to authenticate each other. EAPoUDP is a variation of the Extensible Authentication Protocol (PPP EAP) [2], but runs instead over IP - either IPv4 or IPv6. Unlike PPP EAP, EAPoUDP allows authentication over any link layer technology. Furthermore, the PAA and the PaC need not be on the same link. EAPoUDP uses UDP as its transport protocol.

Table of Contents

1 Introduction

This document specifies the Extensible Authentication Protocol over UDP (EAPoUDP) to be used for network access authentication.

An access domain is represented by one or many PANA Authentication Agents (PAAs). Before a PANA Client (PaC) is granted access to the domain, a PAA and a PaC MAY use EAPoUDP to authenticate each other.

EAPoUDP calls for methods for Initial Authentication (I-A), Re-Authentication (R-A) and Disconnect Notification (D-N). I-A is for mutual authentication, which a PaC and a PAA are expected to perform before the PaC is granted access to an access domain. A product of the I-A method is a session key established between PaC and PAA. This session key can be used for R-A, when a PAA or a PaC wants to re-authenticate the other party and validate that it is still present and alive. After successful authentication, either the PAA or the PaC MAY want to terminate the authentication relationship by sending a (D-N) to the other party.

EAPoUDP is a variation of the Extensible Authentication Protocol (PPP EAP) [2]. Unlike PPP EAP, EAPoUDP runs over IP - either IPv4 or IPv6. Thus, it allows PaCs and PAAs to authenticate each other over any link layer technology, and they do not need to be on the same link. For a lightweight solution, UDP is chosen as transport protocol.

EAPoUDP assumes that prior to authentication the PaC has configured a valid IPv4 or IPv6 address for itself. It MAY also have discovered an IP-address for at least one PAA in the access domain. PAA Discovery mechanisms are proposed and detailed in [1].

Where to locate PAAs (e.g. with a PAA located on each access router or with a pool of PAAs located anywhere in the access domain) represents an architectural tradeoff. The PANA WG may leave to implementers and operators to decide which architecture best fits their needs. Alternatively, the PANA WG may mandate that PAAs are located on access routers. The scheme presented in this document should accommodate all alternative PAA configurations.

2 Terminology

This document uses the following terminology same as in [10]:

Device Identifier (DI)

This is the identifier used by the network as a handle to control and police the network access of a client. Depending on the access technology, identifier might contain any of IP address, link-layer address, switch port number, etc. of a device. PANA authentication agent keeps a table for binding device identifiers to the PANA clients.

PANA Client (PaC)

This is the entity wishing to obtain network access from a PANA authentication agent within a network. A PANA client is associated with a network device and a set of credentials to prove its identity within the scope of PANA.

PANA Authentication Agent (PAA)

This is the entity whose responsibility is to authenticate the credentials provided by a PANA client and grant network access service to the device associated with the client and identified by a DI.

In addition, the following terms are introduced:

Initiator

The Initiator (i.e. like a PPP EAP Authenticator [2], [3]) of an EAPoUDP authentication method is the entity (i.e. a PaC or a PAA) that sends the EAPoUDP Request(s) to a Peer.

Peer

The Peer of an EAP-based authentication method is the entity (i.e. a PaC or a PAA) that sends the EAPoUDP Response(s) back to an Initiator.

Initial Authentication (I-A)

Initial Authentication is the method for mutual authentication, that a PaC and a PAA are expected to perform before the PaC is granted access to an access domain. A product of I-A is a session key is established between PaC and PAA, which is used for Re-Authentication (below).

Re-Authentication (R-A)

After Initial Authentication (I-A), a PAA or a PaC MAY want to re-authenticate the other party and validate that it is still present and alive. A common method for R-A is that the Initiator sends a challenge to the Peer. The Peer computes a hash over the challenge, keyed by a session key, and returns the result to the Initiator. The Peer and Initiator use the session key established during the Initial Authentication to key the hash. This document specifies a method for re-authentication (R-A).

Disconnect Notification (D-N)

After successful authentication, either the PAA or the PaC MAY want to explicitly terminate the authentication relationship by sending a Disconnect-Notification (D-N) to the other party. D-Ns alone cannot guarantee disconnect. Due to Denial-of-Service (DoS) threats, D-N cannot be guaranteed to reach the other party. Disconnect can only be guaranteed by mandatory timeout mechanisms implemented in I-A and R-A. Thus, D-N is a function to optimize EAPoUDP. D-Ns MUST be integrity protected to avoid being a tool for DoS attacks.

3 UDP as transport protocol

This document suggests that EAPoUDP uses UDP as its transport protocol.

For a lightweight solution, UDP and ICMP are both attractive alternatives. UDP is chosen here to allow for application layer implementations.

EAPoUDP SHOULD use IANA-assigned port numbers (TBD).

4 EAPoUDP reuses PPP EAP message formats

4.1 EAPoUDP message format

The EAPoUDP specification follows that of PPP EAP ([2], [3]) unless otherwise specified in this document.

The EAPoUDP packet reuses the PPP EAP format ([2], [3]). An EAPoUDP packet will be sent as follows:

```
          +-----------+-----------+-------------+
          | IP header | UDP header | EAP message |
          +-----------+-----------+-------------+
```

All messages begins with a 32-bit header following the UDP header:

```
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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Data ...
+-+-+-+-+-+
```

Code

The Code field is one octet and identifies the type of EAPoUDP message. EAPoUDP Codes reuse the following EAP Codes:

1 Request
2 Response
3 Success
4 Failure

Identifier

The Identifier field is one octet and is used - together with source and destination IP-addresses (i.e. IP-addresses of Initiator and Peer) - to match responses with requests.

Length

The Length field is two octets and indicates the length (in octets) of the EAPoIP message including the Code, Identifier, Length and Data fields.

Data

The Data field is zero or more octets. The Code field determines the format of the Data field.

The Data field of Request and Response messages consists of an additional Type field of 1 octet followed by Type-Data:

```
    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      |  Type-Data ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type (Request/Response messages)

This Type field of Request/Response messages indicates which authentication method is carried in Type-Data.

Type-Data

The Type-Data field is zero or more octets and carries information associated with the authentication method. The Type field determines the format of the Type-Data field.

4.2 Types for Request/Response messages

EAPoUDP reuses some selected PPP EAP methods. However, PPP EAP methods cannot blindly be ported into EAPoUDP without taking security threats into account. On multi-access links, PPP EAP methods that are vulnerable to attacks (including eavesdropping, address spoofing, replay attacks and man-in-the-middle attacks), MUST NOT be used with EAPoUDP.

EAPoUDP will explicitly specify which PPP EAP methods to be used, and assign a type value to the selected method. Other PPP EAP methods MUST NOT be used with EAPoUDP.

The following EAPoUDP Types are supported in EAPoUDP:

Type	
1	Identity
2	Notification
3	NAK
4	MD-5 Challenge for Re-Authentication
TBD	Selected method for Initial Authentication
TBD	Selected method for Disconnect-Notification

Other Types MUST NOT be used.

To ensure correct and secure operation in a multi-access environment, EAPoUDP imposes additional requirements on the operation of selected PPP EAP methods. Next sub-section summarizes the additional requirements imposed on the MD-5 Challenge method selected for EAPoUDP re-authentication.

4.3 MD-5 Challenge for Re-Authentication

4.3.1 MD-5 Challenge/Response format

This document proposes to reuse the PPP EAP MD-5 Challenge/Response authentication method for EAPoUDP re-authentication ([2], [3]). However, we have modified the Type-Data format of challenges and responses to incorporate an optional Device Identifier. The content of the Type-Data field is summarized below.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Value-Length  |  Value ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
   |  Name-Length  |  Name ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
   |   DI-Length   |  Device Identifier of Peer ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Value-Length

This is the length, in octets, of the Value field, which MUST be at least one octet.

Value

The Value field contains a challenge in Request messages, and a calculated MD-5 hash (Section 4.3.2) in Response messages.

Name-Length

This is the length, in octets, of the Name field, which SHOULD be at least one octet.

Name

The Name field contains the Network Access Identifier (NAI) of the sender of the message. A request message, for example, would contain a NAI of the Initiator, and a response message would contain a NAI of the Peer. This field MAY contain a temporary NAI, which MAY have been derived during Initial Authentication.

DI-Length

This is the length, in octets, of the Device Identifier field. If a Device Identifier is not present in the message, the value is set to zero.

Device Identifier

This field contains a Device Identifier of the Peer. The Initiator MAY include a Device Identifier in a challenge request to confirm that the IP (or MAC) source addresses of packets received from the Peer is correct. The Initiator incorporates the address(es) into the Device Identifier, generates a random challenge, and sends the challenge request message to the Peer. The Peer MUST copy the Device Identifier field from the Challenge message into the same field of the response message. (Later versions of this document MAY open for more extensive negotiations of Device Identifier values.) The Peer MUST validate that the information in the Device Identifier is correct. The Peer MUST NOT return a valid Response message if the information is not correct.

The format of the Device Identifier will be specified in a follow-on document.

4.3.2 MD-5 Hash Calculation

To ensure correct and secure operation in a multi-access environment, EAPoUDP imposes requirements on how the MD-5 hash is calculated:

The MD-5 hash MUST be calculated over a stream of octets in sequence consisting of the Network Access Identifier (NAI) of Peer, followed by (concatenated with) Device Identifier of Peer (if present), followed by (concatenated with) the Identifier octet, followed by (concatenated with) the session key for re-authentication, and followed by (concatenated with) the Challenge Value. The Device Identifier is copied from the Device Identifier field in the MD-5 Response message.

Since the MD-5 hash is calculated over the NAI of the Peer, it will protect against reflection attacks, even when Initiator and Peer use the same session key for re-authentication in both directions. In comparison, the original PPP EAP MD-5 hash is only calculated over the Identifier, session key, and Challenge, and requires different session keys in each direction.

An Initiator can protect against address spoofing attacks of a Peer's IP-address (or MAC-address) by sending a challenge to the Peer with the Peer's addresses incorporated into the Device Identifier field. The Peer confirms the validity of the addresses by returning the hash calculated over both challenge and device identifier.

4.3.3 Validation of Device Identifier of a Peer

The Initial Authentication method sets up a cache consisting of the other party's identifier, session keys and IP-address (and/or MAC address). Upon receiving an EAPoUDP packet, a PAA or PaC checks the source address, and consults the cache to find the sender's identity and session keys.

However, the selected Initial Authentication method may not be capable of ensuring that the addresses in the cache are correct, and have not been subject to a IP-spoofing attack by a malicious Man-In-The-Middle (MITM). In that case, the Initial Authentication MAY be followed by Re-authentication where the IP- (and/or MAC-) address(es) are incorporated in the Device Identifier. Thus, the re-authentication method can be used as a means to verify the correctness of addresses in the cache. After one successful re-authentication, PAA can safely grant PaC access to the domain.

5 EAPoUDP authentication schemes

5.1 Starting the authentication session

The specification of EAPoUDP should determine the ways in which Initial Authentication (I-A) can be started. There are a number of possibilities, and the following three sub-sections describe some alternatives.

Without loss of generality, we assume in the following discussion that the EAPoUDP I-A method is carried out with PAA as the Initiator.

5.1.1 Alternative 1: I-A triggered by the access network

The following diagram shows a model, but not details, describing the message exchange where the access network triggers PAA to start Initial Authentication:

```
   PaC         AP/AR/DHCP/DAD/etc.       PAA
    |                  |                  |
    |                  |                  |
    |                  | 1a) A trigger    |
    |                  |----------------->|
    |                  | (PaC IP-address) |
    |                                     |
    |                                     |
    | 1b) Identity Request                |
    |<------------------------------------|
    | 1c) Idenity Response (PaC-ID)       |
    |------------------------------------>|
    |                                     |
    |                                     |
    | 2a) Initial-Auth.: First Request    |
    |<------------------------------------|
```

In this alternative, PAA should be co-located with the entity that sends the trigger, e.g. with the Access Router or DHCP server.

The messages are described as follows:

PaC Discovery:

1a) Trigger

Initially, PAA receives a trigger indicating the arrival of a new and un-authenticated PaC. The trigger may come from DHCP (e.g. the DHCP server sends a signal to PAA after having assigned an IP-address to a new PaC), or from another protocol entity. The trigger SHOULD provide PAA with the IP-address of the PaC.

1b) Identity Request

PAA sends an EAPoUDP Identity Request to the given IP-address of the PaC to find out the identity of the PaC.

1c) Identity Response

PaC returns its identity in an EAPoUDP Identity Response.

Initial Authentication:

2a) After having obtained PaC's identity, the PAA starts Initial Authentication (Section 5.2).

5.1.2 Alternative 2: I-A triggered by unsolicited Identity Response

The following diagram shows a model describing the message exchange where the I-A is triggered by a PaC. After having discovered a PAA, the PaC sends it an unsolicited Identity Request, which triggers the PAA to start Initial Authentication:

```
   PaC            AR/DHCP server         PAA
    |                  |                  |
    |                  |                  |
    |1a) PAA Discovery |                  |
    |<---------------->|                  |
    |                  |                  |
    |                                     |
    |                                     |
    |1b) (Unsolicited) Identity Response  |
    |------------------------------------>|
    |                                     |
    |                                     |
    | 2a) Initial-Auth.: First Request    |
    |<------------------------------------|
```

The messages are described as follows:

1a) PAA discovery [1]:

A PaC discovers the IP-address and identity of a PAA (e.g. in a DHCP option). PAA Discovery mechanisms are proposed and detailed in [1].

1b) (Unsolicited) Identity Response:

If the client can positively determine that it has to authenticate, e.g. through successful PAA discovery, it MAY send an unsolicited Identity Response to the PAA, containing the PaC's Identifier. The PaC is free to pick the Identifier octet value. The client MUST NOT send an unsolicited Identity Response if it has already received an Identity Request. (The same method has been proposed in [7].)

Initial Authentication:

2a) The unsolicited Identity Request triggers the PAA to start Initial Authentication (Section 5.2).

5.1.3 Alternative 3: I-A triggered by anycasted PAA discovery

The following diagram shows the message exchange where a PaC uses anycast to discover PAA. This triggers PAA to start Initial Authentication:

```
PaC                                  PAA
 |                                    |
 |                                    |
 |                                    |
 | 1a) (Anycasted) Identity Request   |
 |----------------------------------->|
 | 1b) (Unicasted) Identity Response  |
 |<-----------------------------------|
 |                                    |
 | 1c) Identity Request               |
 |<-----------------------------------|
 | 1d) Identity Response (PaC-ID)     |
 |----------------------------------->|
 |                                    |
 |                                    |
 | 2a) Initial-Auth.: First Request   |
 |<-----------------------------------|
```

The anycasted Identity Request triggers PAA to discover PaC's Identity (message 1c and 1d), before starting Initial Authentication (Section 5.2).

5.2 Initial Authentication

PAA is assumed to be the Initiator of Initial Authentication.

```
      PaC                                     PAA
       |                                     |
       | 2a) Initial-Auth.: First Request    |
       |<------------------------------------|
       |                                     |
       |  Possible additional Requests and   |               +--------+
       |             Responses:              |               |Local   |
       |------------------------------------>|lookup credent.| storage|
       |<------------------------------------|-------------->|   or   |
       |------------------------------------>|return credent.|AAA-    |
       |<------------------------------------|<--------------| infra- |
       |                                     |               | struct.|
       |                                     |               +--------+
       | 2n) Initial-Auth.: Last Response    |
       |------------------------------------>|
       |                                     |
```

The messages are described as follows:

2a) The PAA initiates the Initial Authentication (I-A) by sending an I-A-Request to the PaC.

The I-A method eventually selected by PANA WG may call for additional Request/Response exchanges.

2n) PaC returns the last I-A-Response.

For Initial Authentication, PAA MAY use a local storage, a back-end AAA infrastructure, a Certificate Authority or some other kind of Trusted Third Party (TTP) to verify credentials of a PaC, and to obtain credentials that can be verified by the PaC. The actual process for obtaining and verifying credentials is out of scope for the EAPoUDP specification.

EAPoUDP Success and Failure messages, which parallel those of PPP EAP ([2], [3]), have been omitted here for simplicity.

5.3 Re-authentication

A PAA or PaC MAY re-authenticate the other party at any time after Initial Authentication.

```
    Initiator                               Peer
   (PAA/PaC)                             (PaC/PAA)
       |                                     |
       | 3a) Re-authentication Challenge     |
       |------------------------------------>|
       |                                     |
       | 3b) Re-authentication Response      |
       |<-------------------------------------|
       |                                      |
```

PAA MAY act as an Initiator when re-authenticating PaC as a Peer, or PaC MAY act as an Initiator when re-authenticating the PAA as a Peer.

5.4 Disconnect

A PAA or PaC MAY terminate the authentication relationship by sending a Disconnect Notification to the other party any time after Initial Authentication.

```
   Initiator                               Peer
   (PAA/PaC)                             (PaC/PAA)
       |                                     |
       |  5) Disconnect Notification         |
       |------------------------------------>|
       |                                     |
```

One way of ensuring the integrity of a Disconnect Notification is to require the Initial Authentication method generate a separate Disconnect One-time Password (D-OTP) to integrity-protect the Disconnect Notification message.

5.5 Back-end communication

There are a number of different ways that a PAA may interact with the back-end for authentication to verify credentials of PaCs and to obtain credentials that can be used by PaC to authenticate PAA. The examples above provide one possible scenario:

```
      PaC            PAA
       |              |                               +--------+
       | EAPoUDP msg. |                               |Local   |
       |------------->|lookup credentials & verifiers | storage|
       |              |------------------------------>|   or   |
       |              |return credentials & verifiers |AAA-    |
       |              |<------------------------------| infra- |
       | EAPoUDP msg. |                               | struct.|
       |<-------------|                               +--------+
       |              |
```

Another scenario may call for the use of PAA as a pass-through as follows:

```
      PaC            PAA
       |              |
       | EAPoUDP msg. |                             +--------+
       |------------->|  Forwarded EAPoUDP message  |        |
       |              |---------------------------->| Auth.- |
       |              |  Returned EAPoUDP message   |        |
       |              |<----------------------------| server |
       | EAPoUDP msg. |   (+ master session keys)   |        |
       |<-------------|                             +--------+
       |              |
```

There might be other possible scenarios. This issue is implementation dependent and is out of scope for EAPoUDP.

6 Further work

6.1 Selection of remaining methods

PANA WG MUST select the specific methods used for Initial Authentication and Disconnect Notification. Both EAP AKA [7] and EAP SRP-SHA1 [8] are methods that may be considered for Initial Authentication.

Algorithms to derive session keys from Initial Authentication should also be specified. EAP-independent key-derivation algorithms are under development [9].

6.2 Retransmission and timeout mechanisms

The EAPoUDP protocol may require specific retransmission and timeout mechanisms being used as default for all messages. Specific (i.e. non-default) time-out and re-transmission mechanisms MAY be specified for selected EAPoUDP message types where user input (e.g. a password) is expected.

7 Security Considerations

EAPoUDP reuses existing EAP methods, but the multi-access, multi-hop environment it MAY operate in raises additional security threats. The final EAPoUDP specification MUST therefore further ensure that each EAPoUDP method can be used securely in this environment [10].

IANA Considerations

IANA need to assign a UDP port number for EAPoUDP.

Acknowledgements

...

References

[1] Engelstad, P., "Discovery Mechanism for PANA Authentication Agents (PAA-discovery)", <draft-engelstad-pana-paa-discovery-00.txt>, January 2002, Work in Progress.

[2] Blunk, L. and Vollbrecht, J., "PPP Extensible Authentication Protocol", RFC 2284, March 1998.

[3] Blunk, L., Vollbrecht, J., and Aboba, B., "Extensible Authentication Protocol (EAP)", <draft-ietf-pppext-rfc2284bis-01.txt> (RFC2284bis), November 2001, Work in Progress.

[4] Aboba, B., Beadles, M. "The Network Access Identifier", RFC 2486, January 1999.

[5] Narten, T., and Draves, R., "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.

[6] Tsirtis, G., "EAP over ICMP", <draft-tsirtis-eap-over-icmp-00.txt>, January 2002, Work in Progress.

[7] Arkko, J., Haverinen, H., "EAP AKA Authentication", <draft-arkko-pppext-eap-aka-01.txt>, November 2001, Work in Progress.

[8] Carlson, J., Aboba, B., Haverinen, H.,"EAP SRP-SHA1 Authentication Protocol", <draft-ietf-pppext-eap-srp-03.txt>, July 2001, Work in Progress.

[9] Aboba, B., Simon, D. "The EAP Keying Problem", <draft-aboba-pppext-key-problem-01.txt>, February 2002, Work in Progress.

[10] Yegin (ed.) et al., "Protocol for Carrying Authentication for Network Access (PANA) Requirements and Terminology", <draft-ietf-pana-requirements-00.txt>, February 2002, Work in Progress.

Author's Address

Paal E. Engelstad
Telenor R&D (California)
399 Sherman Ave. Suite #12
Palo Alto, CA 94306, USA

Tel.: + 1-650- 714 7537
e-mail: paal@telenorisv.com