Internet Engineering Task Force (IETF)
Request for Comments: 7652
Updates: 6887
Category: Standards Track
ISSN: 2070-1721

M. Cullen
S. Hartman
Painless Security
D. Zhang

T. Reddy
Cisco

September 2015

Port Control Protocol (PCP) Authentication Mechanism

Abstract

An IPv4 or IPv6 host can use the Port Control Protocol (PCP) to flexibly manage the IP address-mapping and port-mapping information on Network Address Translators (NATs) or firewalls to facilitate communication with remote hosts. However, the uncontrolled generation or deletion of IP address mappings on such network devices may cause security risks and should be avoided. In some cases, the client may need to prove that it is authorized to modify, create, or delete PCP mappings. This document describes an in-band authentication mechanism for PCP that can be used in those cases. The Extensible Authentication Protocol (EAP) is used to perform authentication between PCP devices.

This document updates RFC 6887.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Using the Port Control Protocol (PCP) [RFC6887], an application can flexibly manage the IP address-mapping information on its network address translators (NATs) and firewalls and can control their policies in processing incoming and outgoing IP packets. Because NATs and firewalls both play important roles in network security architectures, there are many situations in which authentication and access control are required to prevent unauthorized users from accessing such devices. This document defines a PCP security extension that enables PCP servers to authenticate their clients with the Extensible Authentication Protocol (EAP). The EAP messages are encapsulated within PCP messages during transmission.

The following issues are considered in the design of this extension:

- Loss of EAP messages during transmission.
- Reordered delivery of EAP messages.
- Generation of transport keys.
- Integrity protection and data origin authentication for PCP messages.
- Algorithm agility.

The mechanism described in this document meets the security requirements to address the Advanced Threat Model described in the base PCP specification [RFC6887]. This mechanism can be used to secure PCP in the following situations:

- On security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings for specific traffic.
- On equipment (such as Carrier-Grade NATs (CGNs) or service provider firewalls) that serves multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- For any implementation that wants to be more permissive in authorizing applications to create mappings for successful inbound communications destined to machines located behind a NAT or a firewall.

2. Terminology

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 RFC 2119 [RFC2119].

Most of the terms used in this document are introduced in [RFC6887].

PCP client: A PCP software instance that is responsible for issuing PCP requests to a PCP server. In this document, a PCP client is also an EAP peer [RFC3748], and it is the responsibility of a PCP client to provide the credentials when authentication is required.

PCP server: A PCP software instance that resides on the PCP-controlled device that receives PCP requests from the PCP client and creates appropriate state in response to that request. In this document, a PCP server is integrated with an EAP authenticator [RFC3748]. Therefore, when necessary, a PCP server can verify the credentials provided by a PCP client and make an access control decision based on the authentication result.

PCP-Authentication (PA) session: A series of PCP message exchanges transferred between a PCP client and a PCP server. The PCP messages that are part of a given session include the PA messages used to perform EAP authentication, key distribution, and session management, as well as the common PCP messages secured with the keys distributed during authentication. Each PA session is assigned a distinctive Session ID.

Session partner: A PCP implementation involved in a PA session. Each PA session has two session partners (a PCP server and a PCP client).

PCP device: A PCP client or a PCP server.

Session lifetime: The lifetime associated with a PA session. The session lifetime of the PA session decides the lifetime of the current authorization given to the PCP client.

PA Security Association (PCP SA): An association formed between a PCP client and a PCP server by sharing cryptographic keying material and associated context. The formed duplex security association is used to protect the bidirectional PCP signaling traffic between the PCP client and PCP server.

Master Session Key (MSK): A key derived by the partners of a PA session, using an EAP key-generating method (e.g., the method defined in [RFC5448]).

PCP-Authentication (PA) message: A PCP message containing an AUTHENTICATION Opcode. Specifically, a PA message sent from a PCP server to a PCP client is referred to as a PA-Server message, while a PA message sent from a PCP client to a PCP server is referred to as a PA-Client message. Therefore, a PA-Server message is actually a PCP response message as specified in [RFC6887], and a PA-Client message is a PCP request message. This document specifies an option -- the PA_AUTHENTICATION_TAG option defined in Section 5.5 for PCP authentication -- to provide integrity protection and message origin authentication for PA messages.

Common PCP message: A PCP message that does not contain an AUTHENTICATION Opcode. This document specifies an AUTHENTICATION_TAG option to provide integrity protection and message origin authentication for the common PCP messages.

3. Protocol Details

3.1. Session Initiation

At the beginning of a PA session, a PCP client and a PCP server need to exchange a series of PA messages in order to perform an EAP authentication process. Each PA message MUST contain an AUTHENTICATION Opcode and may optionally contain a set of options for various purposes (e.g., transporting authentication messages and session management). The Opcode-specific information in an AUTHENTICATION Opcode consists of two fields: Session ID and Sequence Number. The Session ID field is used to identify the PA session to which the message belongs. The Sequence Number field is used to detect whether reordering or duplication occurred during message delivery.

3.1.1. Authentication Triggered by the Client

When a PCP client intends to proactively initiate a PA session with a PCP server, it sends a PA-Initiation message (a PA-Client message with the result code INITIATION) to the PCP server. Section 5.1 updates the PCP request message format with result codes for the PCP authentication mechanism. In the Opcode-specific information of the message, the Session ID and Sequence Number fields are set to zero. The PA-Client message MUST also contain a NONCE option (defined in Section 5.3) that consists of a random nonce.

After receiving the PA-Initiation message, if the PCP server agrees to initiate a PA session with the PCP client, it will reply with a PA-Server message that contains an EAP request, and the Result Code field of this PA-Server message is set to AUTHENTICATION_REQUEST. In addition, the server MUST assign a unique session identifier to distinctly identify this session and insert the identifier into the Session ID field in the Opcode-specific information of the PA-Server message. The Sequence Number field of the message is set to zero. The PA-Server message MUST contain a NONCE option so as to send the nonce value back. The nonce will then be used by the PCP client to check the freshness of this message. Subsequent PCP messages within this PA session MUST contain this session identifier.

```
         PCP                                                 PCP
        client                                              server
          |-- PA-Initiation --------------------------------->|
          |    (Seq=0, rc=INITIATION, Session ID=0)           |
          |                                                   |
          |<-- PA-Server -------------------------------------|
          |     (Seq=0, Session ID=X, EAP request,            |
          |      rc=AUTHENTICATION_REQUEST)                   |
          |                                                   |
          |-- PA-Client ------------------------------------->|
          |    (Seq=1, Session ID=X, EAP response,            |
          |     rc=AUTHENTICATION_REPLY)                      |
          |                                                   |
          |<-- PA-Server -------------------------------------|
          |     (Seq=1, Session ID=X, EAP request,            |
          |      rc=AUTHENTICATION_REQUEST)                   |
```

3.1.2. Authentication Triggered by the Server

In the scenario where a PCP server receives a common PCP request message from a PCP client that needs to be authenticated, the PCP server rejects the request with an AUTHENTICATION_REQUIRED error code and can reply with an unsolicited PA-Server message to initiate a PA session. The Result Code field of this PA-Server message is set to AUTHENTICATION_REQUEST. In addition, the PCP server MUST assign a Session ID for the session and transfer it within the PA-Server message. The Sequence Number field in the PA-Server message is set to zero. If the PCP client retries the common request before EAP authentication is successful, then it will receive an AUTHENTICATION_REQUIRED error code from the PCP server. In subsequent PA messages exchanged during this session, the Session ID will be used in order to help session partners distinguish the messages within this session from those not within it. When the PCP client receives this initial PA-Server message from the PCP server, it can reply with a PA-Client message or silently discard the request message, according to its local policies. In the PA-Client message, a NONCE option that consists of a random nonce MAY be appended. If so, in the next PA-Server message, the PCP server MUST forward the nonce back within a NONCE option.

```
          PCP                                               PCP
         client                                            server
           |-- Common PCP request -------------------------->|
           |                                                 |
           |<- Common PCP response --------------------------|
           |    (rc=AUTHENTICATION_REQUIRED)                 |
           |                                                 |
           |<-- PA-Server -----------------------------------|
           |     (Seq=0, Session ID=X, EAP request,          |
           |      rc=AUTHENTICATION_REQUEST)                 |
           |                                                 |
           |-- PA-Client ---------------------------------->|
           |    (Seq=0, Session ID=X, EAP response,          |
           |     rc=AUTHENTICATION_REPLY)                    |
           |                                                 |
           |<-- PA-Server -----------------------------------|
           |     (Seq=1, Session ID=X, EAP request,          |
           |      rc=AUTHENTICATION_REQUEST)                 |
```

3.1.3. Authentication Using EAP

In a PA session, an EAP request message is transported within a PA-Server message and an EAP response message is transported within a PA-Client message. EAP relies on the underlying protocol to provide reliable transmission; any reordered delivery or loss of packets occurring during transmission must be detected and addressed. Therefore, after sending out a PA-Server message, the PCP server will not send a new PA-Server message in the same PA session until it receives a PA-Client message with a proper sequence number from the PCP client, and vice versa. If a PCP client receives a PA message containing an EAP request and for some reason cannot generate an EAP response immediately (e.g., waiting for human input in order to construct an EAP message, or waiting for the additional PA messages in order to assemble a complete EAP message from fragmented packets), the PCP device MUST reply with a PA-Acknowledgement message (a PA message with a RECEIVED_PAK option) to indicate that the message has been received. This approach not only can avoid unnecessary retransmission of the PA message but also can guarantee reliable message delivery in conditions where a PCP device needs to receive multiple PA messages carrying the fragmented EAP request before generating an EAP response. The number of EAP messages exchanged between the PCP client and PCP server depends on the EAP method used for authentication.

In this approach, a PCP client and a PCP server MUST perform a key-generating EAP method in authentication. Specifically, a PCP authentication implementation MUST support Extensible Authentication Protocol Tunneled Transport Layer Security (EAP-TTLS) [RFC5281] and

SHOULD support the Tunnel Extensible Authentication Protocol (TEAP) [RFC7170]. Therefore, after a successful authentication procedure, a Master Session Key (MSK) will be generated. If the PCP client and the PCP server want to generate a transport key using the MSK, they need to agree upon a Pseudorandom Function (PRF) for the transport key derivation and a Message Authentication Code (MAC) algorithm to provide data origin authentication for subsequent PCP messages. In order to do this, the PCP server needs to append a set of PRF options and MAC_ALGORITHM options to the initial PA-Server message. Each PRF option contains a PRF that the PCP server supports, and each MAC_ALGORITHM option contains a MAC algorithm that the PCP server supports. Moreover, in the first PA-Server message, the server MAY also attach an ID_INDICATOR option (defined in Section 5.11) to direct the client to choose correct credentials. After receiving the options, the PCP client MUST select the PRF and the MAC algorithm that it would like to use; it then MUST add the associated PRF and MAC Algorithm options to the next PA-Client message.

After the EAP authentication, the PCP server sends out a PA-Server message to indicate the EAP authentication and PCP authorization results. If the EAP authentication succeeds, the result code of the PA-Server message is AUTHENTICATION_SUCCEEDED. In this case, before sending out the PA-Server message, the PCP server MUST update the PCP SA with the MSK and transport key and MUST use the derived transport key to generate a digest for the message. The digest is transported within a PA_AUTHENTICATION_TAG option for PCP Auth. A more detailed description of generating the authentication data can be found in Section 6.1. In addition, the PA-Server message MUST also contain a SESSION_LIFETIME option (defined in Section 5.9) that indicates the lifetime of the PA session (i.e., the lifetime of the MSK). After receiving the PA-Server message, the PCP client then needs to generate a PA-Client message in response. If the PCP client also authenticates the PCP server, the result code of the PA-Client message is AUTHENTICATION_SUCCEEDED. In addition, the PCP client needs to update the PCP SA with the MSK and transport key, and it uses the derived transport key to secure the message. From then on, all the PCP messages within the session are secured with the transport key and the MAC algorithm specified in the PCP SA. The first secure PA-Client message from the client MUST include the set of PRF and MAC_ALGORITHM options received from the PCP server. The PCP server determines if the set of algorithms conveyed by the client matches the set it had initially sent, to detect an algorithm downgrade attack. If the server detects a downgrade attack, then it MUST send a PA-Server message with result code DOWNGRADE_ATTACK_DETECTED and terminate the session. If the PCP client sends a common PCP request within the PA session without an AUTHENTICATION_TAG option, then the PCP server rejects the request by returning an AUTHENTICATION_REQUIRED error code.

If a PCP client/server cannot authenticate its session partner, the device sends out a PA message with the result code AUTHENTICATION_FAILED. If the EAP authentication succeeds but authorization fails, the device making the decision sends out a PA message with the result code AUTHORIZATION_FAILED. In these two cases, after the PA message is sent out, the PA session MUST be terminated immediately. It is possible for independent PCP clients on the host to create multiple PA sessions with the PCP server.

3.2. Recovery from Lost PA Session

If a PCP server resets or loses the PCP SA due to reboot, power failure, or any other reason, then it sends an unsolicited ANNOUNCE response, as explained in Section 14.1.3 of [RFC6887], to the PCP client. Upon receiving the ANNOUNCE response with an anomalous Epoch Time, the PCP client deduces that the server may have lost state. The ANNOUNCE is either bogus (an attack), legitimate, or not seen by the client. These three cases are described below:

- The PCP client sends an integrity-protected unicast ANNOUNCE request to the PCP server to see whether the PCP server has indeed lost state or an attacker has sent the ANNOUNCE response.
  - If an integrity-protected success response is received from the PCP server, then the PCP client determines that the PCP server has not lost the PA session, and the unsolicited ANNOUNCE response was sent by an attacker.
  - If the PCP server responds to the ANNOUNCE request with an UNKNOWN_SESSION_ID error code, then the PCP client MUST initiate full EAP authentication with the PCP server, as explained in Section 3.1.1. After EAP authentication is successful, the PCP client updates the PCP SA and issues new common PCP requests to recreate any lost mapping state.
- In a scenario where the PCP server has lost the PCP SA but did not inform the PCP client, if the PCP client sends an integrity-protected PCP request, then the PCP server rejects the request with an UNKNOWN_SESSION_ID error code. The PCP client then initiates full EAP authentication with the PCP server, as explained in Section 3.1.1, and updates the PCP SA after successful authentication.

If the PCP client resets or loses the PCP SA due to reboot, power failure, or any other reason and sends a common PCP request, then the PCP server rejects the request with an AUTHENTICATION_REQUIRED error code. The PCP client MUST authenticate with the PCP server and, after EAP authentication is successful, retry the common PCP request with an AUTHENTICATION_TAG option. The PCP server MUST update the PCP SA after successful EAP authentication.

3.3. Session Termination

A PA session can be explicitly terminated by either session partner. A PCP server may explicitly request termination of the session by sending an unsolicited termination-indicating PA response (a PA response with a result code of SESSION_TERMINATED). Upon receiving a termination-indicating message, the PCP client MUST respond with a termination-indicating PA message and MUST then remove the associated PCP SA. To accommodate packet loss, the PCP server MAY transmit the termination-indicating PA response up to ten times (with an appropriate Epoch Time value in each to reflect the passage of time between transmissions), provided that (1) the interval between the first two notifications is at least 250 ms and (2) each interval between subsequent notifications at least doubles.

A PCP client may explicitly request termination of the session by sending a termination-indicating PA request (a PA request with a result code of SESSION_TERMINATED). After receiving a termination-indicating message from the PCP client, a PCP server MUST respond with a termination-indicating PA message and remove the PCP SA immediately. When the PCP client receives the termination-indicating PA response, it MUST remove the associated PCP SA immediately.

3.4. Session Re-authentication

A session partner may choose to perform EAP re-authentication if it would like to update the PCP SA without initiating a new PA session. For example, a re-authentication procedure could be triggered for the following reasons:

- The session lifetime needs to be extended.
- The sequence number is going to reach the maximum value. Specifically, when the sequence number reaches 2**32 - 2**16, the session partner MUST trigger re-authentication.

When the PCP server would like to initiate a re-authentication, it sends the PCP client a PA-Server message. The result code of the message is set to RE-AUTHENTICATION, which indicates that the message is for a re-authentication process. If the PCP client would like to start the re-authentication, it will send a PA-Client message to the PCP server, with the result code of the PA-Client message set to RE-AUTHENTICATION. Then, the session partners exchange PA messages to transfer EAP messages for the re-authentication. During the re-authentication procedure, the session partners protect the integrity of PA messages with the key and MAC algorithm specified in the current PCP SA; the sequence numbers associated with the message will continue to keep increasing as specified in Section 6.4. The result code for a PA-Server message carrying an EAP request will be set to AUTHENTICATION_REQUIRED, and a PA-Client message carrying an EAP response will be set to AUTHENTICATION_REPLY.

If the EAP re-authentication succeeds, the result code of the last PA-Server message is AUTHENTICATION_SUCCEEDED. In this case, before sending out the PA-Server message, the PCP server MUST update the SA and use the new key to generate a digest for the PA-Server message and subsequent PCP messages. In addition, the PA-Server message MUST be appended with a SESSION_LIFETIME option that indicates the new lifetime of the PA session. PA and PCP message sequence numbers must also be reset to zero.

If the EAP authentication fails, the result code of the last PA-Server message is AUTHENTICATION_FAILED. If the EAP authentication succeeds but authorization fails, the result code of the last PA-Server message is AUTHORIZATION_FAILED. In the latter two cases, the PA session MUST be terminated immediately after the last PA message exchange. If for some unknown reason re-authentication is not performed and the session lifetime has expired, then the PA session MUST be terminated immediately.

During re-authentication, the session partners can also exchange common PCP messages in parallel. The common PCP messages MUST be protected with the current SA until the new SA has been generated. The sequence of EAP messages exchanged for re-authentication will not change, regardless of the PCP device triggering re-authentication. If the PCP server receives a re-authentication request from the PCP client after the PCP server itself had sent a re-authentication request, then it should discard its request and respond to the re-authentication request from the PCP client.

4. PA Security Association

At the beginning of a new PA session, each PCP device must create and initialize state information for a new PA Security Association (PCP SA) to maintain its state information for the duration of the PA session. The parameters of a PCP SA are as follows:

- IP address and UDP port number of the PCP client.
- IP address and UDP port number of the PCP server.
- Session identifier.

- Sequence number for the next outgoing PA message.

- Sequence number for the next incoming PA message.

- Sequence number for the next outgoing common PCP message.

- Sequence number for the next incoming common PCP message.

- Last outgoing message payload.

- Retransmission interval.

- The Master Session Key (MSK) generated by the EAP method.

- The MAC algorithm that the transport key should use to generate digests for PCP messages.

- The pseudorandom function negotiated in the initial PA-Server and PA-Client message exchange for the transport key derivation.

- The transport key derived from the MSK to provide integrity protection and data origin authentication for the messages in the PA session. The lifetime of the transport key SHOULD be identical to the lifetime of the session.

- The nonce selected by the PCP client at the initiation of the session.

- The key ID associated with the transport key.

Specifically, the transport key is computed in the following way: transport key = prf(MSK, "IETF PCP" || Session ID || Nonce || key ID), where:

- prf is the pseudorandom function assigned in the PRF option (Section 5.7).

- MSK is the master session key generated by the EAP method.

- "IETF PCP" is the ASCII code representation of the non-null-terminated string (excluding the double quotes around it).

- '||' is the concatenation operator.

- Session ID is the ID of the session from which the MSK is derived.

- Nonce is the nonce selected by the client and transported in the initial PA-Client message.

- Key ID is the ID assigned for the transport key.

5. Packet Format

5.1. Packet Format of PCP Auth Messages

The format of the PA-Server message is identical to the response message format specified in Section 7.2 of [RFC6887]. The result code for a PA-Server message carrying an EAP request MUST be set to AUTHENTICATION_REQUEST.

This document updates the Reserved field (see Figure 1) in the Request header specified in Section 7.1 of [RFC6887] to carry Opcode-specific 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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  Version = 2  |R|   Opcode    |   Reserved    |Opcode-specific|
     |               | |             |               |   data        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                 Requested Lifetime (32 bits)                  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |            PCP Client’s IP Address (128 bits)                 |
     |                                                               |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     :                                                               :
     :                  Opcode-specific information                  :
     :                                                               :
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     :                                                               :
     :                   (optional) PCP Options                      :
     :                                                               :
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Request Packet Format

The PA-Client messages (as shown in Figure 2) use the Request header specified in Figure 1. The Opcode-specific data is used to transfer the result codes (e.g., INITIATION, AUTHENTICATION_FAILED). Other fields in Figure 2 are described in Section 7.1 of [RFC6887]. The result code for a PA-Client message carrying an EAP response MUST be set to AUTHENTICATION_REPLY.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 2  |R|   Opcode    |   Reserved    |  Result Code  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Requested Lifetime (32 bits)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            PCP Client’s IP Address (128 bits)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                 Opcode-specific information                   :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                  (optional) PCP Options                       :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: PA-Client Message Format

The Requested Lifetime field of a PA-Client message and the Lifetime field of a PA-Server message are both set to zero on transmission and ignored on reception.

5.2. Opcode-Specific Information of AUTHENTICATION Opcode

The following diagram shows the format of the Opcode-specific information for the AUTHENTICATION Opcode.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                       Session ID                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     Sequence Number                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Session ID: This field contains a 32-bit PA session identifier.

Sequence Number: This field contains a 32-bit sequence number. A sequence number needs to be incremented on every new (non-retransmission) outgoing PA message in order to provide an ordering guarantee for PA messages.

5.3. NONCE Option

Because the session identifier of a PA session is determined by the PCP server, a PCP client does not know the session identifier that will be used when it sends out a PA-Initiation message. In order to prevent an attacker from interrupting the authentication process by sending spoofed PA-Server messages, the PCP client needs to generate a random number as a nonce in the PA-Initiation message. The PCP server will append the nonce within the initial PA-Server message. If the PA-Server message does not carry the correct nonce, the message MUST be silently discarded.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |  Option Code  |  Reserved     |       Option-Length           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         Nonce                                 |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 4.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Nonce: A random 32-bit number that is transported within a PA-Initiation message and the corresponding reply message from the PCP server.

5.4. AUTHENTICATION_TAG Option

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  Option Code  |  Reserved     |       Option-Length           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                         Session ID                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                       Sequence Number                         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                           Key ID                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      |                Authentication Data (Variable)                 |
      ~                                                               ~
      |                                                               |
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Because there is no authentication Opcode in common PCP messages, the authentication tag for common PCP messages needs to carry the Session ID and Sequence Number.

Option Code: 5.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: The length of the AUTHENTICATION_TAG option for the common PCP message (in octets), including the 12-octet fixed-length header and the variable-length authentication data.

Session ID: A 32-bit field used to identify the session to which the message belongs and identify the secret key used to create the message digest appended to the PCP message.

Sequence Number: A 32-bit sequence number. In this option, a sequence number needs to be incremented on every new (non-retransmission) outgoing common PCP message in order to provide an ordering guarantee for common PCP messages.

Key ID: The ID associated with the transport key used to generate authentication data. This field is filled with zeros if the MSK is directly used to secure the message.

Authentication Data: A variable-length field that carries the Message Authentication Code for the common PCP message. The generation of the digest varies according to the algorithms specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with zeros when necessary.

5.5. PA_AUTHENTICATION_TAG Option

This option is used to provide message authentication for PA messages. In contrast to the AUTHENTICATION_TAG option for common PCP messages, the Session ID field and the Sequence Number field are removed because such information is provided in the Opcode-specific information of the AUTHENTICATION Opcode.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Code  |  Reserved     |       Option-Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Key ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Authentication Data (Variable)                |
~                                                               ~
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 6.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: The length of the PA_AUTHENTICATION option for the PCP Auth message (in octets), including the 4-octet fixed-length header and the variable-length authentication data.

Key ID: The ID associated with the transport key used to generate authentication data. This field is filled with zeros if the MSK is directly used to secure the message.

Authentication Data: A variable-length field that carries the Message Authentication Code for the PCP Auth message. The generation of the digest varies according to the algorithms specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with null characters when necessary.

5.6. EAP_PAYLOAD Option

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Code  |  Reserved     |       Option-Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           EAP Message                         |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 7.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: Variable.

EAP Message: The EAP message transferred. Note that this field MUST end on a 32-bit boundary, padded with zeros when necessary.

5.7. PRF Option

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Code  |  Reserved     |       Option-Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           PRF                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 8.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

PRF: The pseudorandom function that the sender supports to generate an MSK. This field contains a value indicating Internet Key Exchange Protocol version 2 (IKEv2) Transform Type 2 [RFC7296] [RFC4868]. A PCP implementation MUST support PRF_HMAC_SHA2_256 (transform ID = 5).

5.8. MAC_ALGORITHM Option

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Code  |  Reserved     |       Option-Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    MAC Algorithm ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 9.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

MAC Algorithm ID: Indicates the MAC algorithm that the sender supports to generate authentication data. The MAC Algorithm ID field contains a value indicating IKEv2 Transform Type 3 [RFC7296] [RFC4868]. A PCP implementation MUST support AUTH_HMAC_SHA2_256_128 (transform ID = 12).

5.9. SESSION_LIFETIME Option

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Code  |  Reserved     |       Option-Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Session Lifetime                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 10.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Session Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 2^32-1 seconds. The lifetime of the PA session, which is decided by the authorization result.

5.10. RECEIVED_PAK Option

This option is used in a PA-Acknowledgement message to indicate that a PA message with the contained sequence number has been received.

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  Option Code  |  Reserved     |       Option-Length           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Received Sequence Number                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 11.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Received Sequence Number: The sequence number of the last received PA message.

5.11. ID_INDICATOR Option

The ID_INDICATOR option is used by the PCP client to determine which credentials to provide to the PCP server.

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  Option Code  |  Reserved     |       Option-Length           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      |                          ID Indicator                         |
      ~                                                               ~
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: 12.

Reserved: 8 bits. MUST be set to zero on transmission and MUST be ignored on reception.

Option-Length: Variable.

ID Indicator: The identity of the authority that issued the EAP credentials to be used to authenticate the client. The field

MUST NOT be null terminated, and its length is indicated by the Option-Length field. In particular, when a client receives an ID_INDICATOR option, it MUST NOT rely on the presence of a null character in the wire format data to identify the end of the ID Indicator field.

The field MUST end on a 32-bit boundary, padded with zeros when necessary. The ID Indicator field is a UTF-8 encoded [RFC3629] Unicode string conforming to the UsernameCaseMapped profile of the PRECIS IdentifierClass [RFC7613]. The PCP client validates that the ID Indicator field conforms to the UsernameCaseMapped profile of the PRECIS IdentifierClass. The PCP client enforces the rules specified in Section 3.2.2 of [RFC7613] to map the ID Indicator field. The PCP client compares the resulting string with the ID indicators stored locally on the PCP client to pick the credentials for authentication. The two indicator strings are to be considered equivalent by the client if and only if they are an exact octet-for-octet match.

6. Processing Rules

6.1. Authentication Data Generation

After a successful EAP authentication process, every subsequent PCP message within the PA session MUST carry an authentication tag that contains the digest of the PCP message for data origin authentication and integrity protection.

- Before generating a digest for a PA message, a device needs to first locate the PCP SA according to the session identifier and then get the transport key. Then, the device appends a PA_AUTHENTICATION_TAG option for PCP Auth at the end of the PCP Auth message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then fills the Key ID field with the key ID of the transport key and sets the Authentication Data field to zero. After this, the device generates a digest for the entire PCP message (including the PCP header and PA_AUTHENTICATION_TAG option) using the transport key and the associated MAC algorithm, and inserts the generated digest into the Authentication Data field.

- Similar to generating a digest for a PA message, before generating a digest for a common PCP message, a device needs to first locate the PCP SA according to the session identifier and then get the transport key. Then, the device appends the AUTHENTICATION_TAG option at the end of the common PCP message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then uses the corresponding values derived from the SA to fill the Session ID field, the Sequence Number field, and the Key ID field, and sets the Authentication Data field to zero. After this, the device generates a digest for the entire PCP message (including the PCP header and AUTHENTICATION_TAG option) using the transport key and the associated MAC algorithm, and inserts the generated digest into the Authentication Data field.

6.2. Authentication Data Validation

When a device receives a common PCP message with an AUTHENTICATION_TAG option for common PCP messages, the device needs to use the Session ID transported in the option to locate the proper SA and then find the associated transport key (using the key ID in the option) and the MAC algorithm. If no proper SA or transport key is found or the sequence number is invalid (see Section 6.5), the PCP device stops processing the PCP message and silently discards the message. After storing the value of the Authentication field of the AUTHENTICATION_TAG option, the device fills the Authentication field with zeros. Then, the device generates a digest for the message (including the PCP header and AUTHENTICATION_TAG option) with the transport key and the MAC algorithm. If the value of the newly generated digest is identical to the stored one, the device can ensure that the message has not been tampered with, and the validation succeeds. Otherwise, the PCP device stops processing the PCP message and silently discards the message.

Similarly, when a device receives a PA message with a PA_AUTHENTICATION_TAG option for PCP authentication, the device needs to use the Session ID transported in the Opcode to locate the proper SA and then find the associated transport key (using the key ID in the option) and the MAC algorithm. If no proper SA or transport key is found or the sequence number is invalid (see Section 6.4), the PCP device stops processing the PCP message and silently discards the message. After storing the value of the Authentication field of the PA_AUTHENTICATION_TAG option, the device fills the Authentication field with zeros. Then, the device generates a digest for the message (including the PCP header and PA_AUTHENTICATION_TAG option) with the transport key and the MAC algorithm. If the value of the newly generated digest is identical to the stored one, the device can ensure that the message has not been tampered with, and the validation succeeds. Otherwise, the PCP device stops processing the PCP message and silently discards the message.

6.3. Retransmission Policies for PA Messages

Because EAP relies on the underlying protocols to provide reliable transmission, after sending a PA message, a PCP client/server MUST NOT send out any subsequent messages until it has received a PA message with a proper sequence number from the peer. If no such message is received, the PCP device will resend the last message according to retransmission policies. This specification uses the retransmission policies specified in Section 8.1.1 of the base PCP specification [RFC6887]. In base PCP, such retransmission policies are only applied by PCP clients. However, in this specification, such retransmission policies are also applied by the PCP servers. If the "maximum retransmission" duration (in seconds) has elapsed and no expected response is received, the device will terminate the session and discard the current SA.

As discussed in Section 3.1.3, in order to avoid unnecessary retransmission, the device receiving a PA message MUST send a PA-Acknowledgement message to the sender of the PA message when it cannot send a PA response immediately. The PA-Acknowledgement message is used to indicate the receipt of the PA message. When the sender receives the PA-Acknowledgement message, it will stop the retransmission.

Note that the last PA messages transported within the phases of session initiation, session re-authentication, and session termination do not have to follow the above policies, since the devices sending out those messages do not expect any further PA messages.

When a device receives a retransmitted last incoming PA message from its session partner, it MUST try to answer it by sending the last outgoing PA message again. However, if the duplicate message has the same sequence number but is not bitwise identical to the original message, then the device MUST discard it. In order to perform this function, the device may need to maintain the last incoming message and the associated outgoing messages. In this case, if no outgoing PA message has been generated for the received duplicate PA message yet, the device needs to send a PA-Acknowledgement message. The rate of replying to duplicate PA messages MUST be limited to provide robustness against denial-of-service (DoS) attacks. The details of rate limiting are outside the scope of this specification.

6.4. Sequence Numbers for PCP Auth Messages

PCP uses UDP to transport signaling messages. As an unreliable transport protocol, UDP does not guarantee ordered packet delivery and does not provide any protection from packet loss. In order to ensure that the EAP messages are exchanged in a reliable way, every PCP message exchanged during EAP authentication must carry a monotonically increasing sequence number. During a PA session, a PCP device needs to maintain two sequence numbers for PA messages: one for incoming PA messages and one for outgoing PA messages. When generating an outgoing PA message, the device adds the associated outgoing sequence number to the message and increments the sequence number maintained in the SA by 1. When receiving a PA message from its session partner, the device will not accept it if the sequence number carried in the message does not match the incoming sequence number maintained in the device. After confirming that the received message is valid, the device increments the incoming sequence number maintained in the SA by 1.

The above rules are not applicable to PA-Acknowledgement messages (i.e., PA messages containing a RECEIVED_PAK option). A PA-Acknowledgement message does not transport any EAP message and only indicates that a PA message is received. Therefore, reliable transmission of PA-Acknowledgement messages is not required. For instance, after sending out a PA-Acknowledgement message, a device generates an EAP response. In this case, the device does not have to confirm whether the PA-Acknowledgement message has been received by its session partner or not. Therefore, when receiving or sending out a PA-Acknowledgement message, the device MUST NOT increase the corresponding sequence number stored in the SA. Otherwise, loss of a PA-Acknowledgement message will cause a mismatch in sequence numbers.

Another exception is the message retransmission scenario. As discussed in Section 6.3, when a PCP device does not receive any response from its session partner, it needs to retransmit the last outgoing PA message, following the retransmission procedure specified in Section 8.1.1 of [RFC6887]. The original message and duplicate messages MUST be bitwise identical. When the device receives such a duplicate PA message from its session partner, it MUST send the last outgoing PA message again. In such cases, the maintained incoming and outgoing sequence numbers will not be affected by the message retransmission.

6.5. Sequence Numbers for Common PCP Messages

When transporting common PCP messages within a PA session, a PCP device needs to maintain a sequence number for outgoing common PCP messages and a sequence number for incoming common PCP messages. When generating a new outgoing PCP message, the PCP device updates the Sequence Number field in the AUTHENTICATION_TAG option with the outgoing sequence number maintained in the SA and increments the outgoing sequence number by 1.

When receiving a PCP message from its session partner, the PCP device will not accept it if the sequence number carried in the message is smaller than the incoming sequence number maintained in the device. This approach can protect the PCP device from replay attacks. After confirming that the received message is valid, the PCP device will update the incoming sequence number maintained in the PCP SA with the sequence number of the incoming message.

Note that the sequence number in the incoming message may not exactly match the incoming sequence number maintained locally. As discussed in the base PCP specification [RFC6887], if a PCP client is no longer interested in the PCP transaction and has not yet received a PCP response from the server, then it will stop retransmitting the PCP request. After that, the PCP client might generate new PCP requests for other purposes, using the current SA. In this case, the sequence number in the new request will be larger than the sequence number in the old request and so will be larger than the incoming sequence number maintained in the PCP server.

Note that, as discussed in the base PCP specification [RFC6887], a PCP client needs to select a nonce in each MAP or PEER request, and the nonce is sent back in the response. However, it is possible for a client to use the same nonce in multiple MAP or PEER requests, and this may cause a potential risk of replay attacks. This attack is addressed by using the sequence number in the PCP response.

6.6. MTU Considerations

EAP methods are responsible for MTU handling, so no special facilities are required in PCP to deal with MTU issues. Specifically, EAP lower layers indicate to EAP methods and Authentication, Authorization, and Accounting (AAA) servers the MTU of the lower layer. EAP methods such as EAP-TLS [RFC5216], TEAP [RFC7170], and others that are likely to exceed reasonable MTUs provide support for fragmentation and reassembly. Others, such as EAP - Generalized Pre-Shared Key (EAP-GPSK) [RFC5433], assume that they will never send packets larger than the MTU and use small EAP packets.

If an EAP message is too long to be transported within a single PA message, it will be divided into multiple sections and sent within different PA messages. Note that the receiver may not be able to know what to do in the next step until it has received all the sections and reconstructed the complete EAP message. In this case, in order to guarantee reliable message transmission, after receiving a PA message, the receiver replies with a PA-Acknowledgement message to notify the sender to send the next PA message.

7. IANA Considerations

The following PCP Opcode has been allocated from the Standards Action range of the "PCP Opcodes" registry (which is maintained in <http://www.iana.org/assignments/pcp-parameters>):

3 AUTHENTICATION.

The following PCP result codes have been allocated from the Standards Action range of the "PCP Result Codes" registry (which is maintained in <http://www.iana.org/assignments/pcp-parameters>):

14 INITIATION: The client includes this PCP result code in its request to the server for authentication.

15 AUTHENTICATION_REQUIRED: This error response is sent to the client if EAP authentication is required.

16 AUTHENTICATION_FAILED: This error response is sent to the client if EAP authentication failed.

17 AUTHENTICATION_SUCCEEDED: This success response is sent to the client if EAP authentication succeeded.

18 AUTHORIZATION_FAILED: This error response is sent to the client if EAP authentication succeeded but authorization failed.

19 SESSION_TERMINATED: This PCP result code indicates to the partner that the PA session must be terminated.

20 UNKNOWN_SESSION_ID: This error response is sent from the PCP server if there is no known PA session associated with the Session ID sent in the PA request or common PCP request from the PCP client.

21 DOWNGRADE_ATTACK_DETECTED: This PCP result code indicates to the client that the server detected a downgrade attack.

22 AUTHENTICATION_REQUEST: The server indicates to the client that the PA message contains an EAP request.

23 AUTHENTICATION_REPLY: The client indicates to the server that the PA message contains an EAP response.

The following PCP options have been allocated from the Standards Action range (the registry for PCP options is maintained in <http://www.iana.org/assignments/pcp-parameters>):

7.1. NONCE

Name: NONCE.

Value: 4.

Purpose: See Section 5.3.

Valid for Opcodes: AUTHENTICATION.

Length: 4 octets.

May appear in: Request and response.

Maximum occurrences: 1.

7.2. AUTHENTICATION_TAG

Name: AUTHENTICATION_TAG.

Value: 5.

Purpose: See Section 5.4.

Valid for Opcodes: MAP, PEER, ANNOUNCE.

Length: variable.

May appear in: Request and response.

Maximum occurrences: 1.

7.3. PA_AUTHENTICATION_TAG

Name: PA_AUTHENTICATION_TAG.

Value: 6.

Purpose: See Section 5.5.

Valid for Opcodes: AUTHENTICATION.

Length: variable.

May appear in: Request and response.

Maximum occurrences: 1.

7.4. EAP_PAYLOAD

Name: EAP_PAYLOAD.

Value: 7.

Purpose: See Section 5.6.

Valid for Opcodes: AUTHENTICATION.

Length: variable.

May appear in: Request and response.

Maximum occurrences: 1.

7.5. PRF

Name: PRF.

Value: 8.

Purpose: See Section 5.7.

Valid for Opcodes: AUTHENTICATION.

Length: 4 octets.

May appear in: Request and response.

Maximum occurrences: as many as fit within maximum PCP message size.

7.6. MAC_ALGORITHM

Name: MAC_ALGORITHM.

Value: 9.

Purpose: See Section 5.8.

Valid for Opcodes: AUTHENTICATION.

Length: 4 octets.

May appear in: Request and response.

Maximum occurrences: as many as fit within maximum PCP message size.

7.7. SESSION_LIFETIME

Name: SESSION_LIFETIME.

Value: 10.

Purpose: See Section 5.9.

Valid for Opcodes: AUTHENTICATION

Length: 4 octets.

May appear in: Response.

Maximum occurrences: 1.

7.8. RECEIVED_PAK

Name: RECEIVED_PAK.

Value: 11.

Purpose: See Section 5.10.

Valid for Opcodes: AUTHENTICATION.

Length: 4 octets.

May appear in: Request and response.

Maximum occurrences: 1.

7.9. ID_INDICATOR

Name: ID_INDICATOR.

Value: 12.

Purpose: See Section 5.11.

Valid for Opcodes: AUTHENTICATION.

Length: variable.

May appear in: Response.

Maximum occurrences: 1.

8. Security Considerations

As described in this specification, after a successful EAP authentication process is performed between two PCP devices, an MSK will be exported. The MSK will be used to derive the transport keys to generate MAC digests for subsequent PCP message exchanges. However, before a transport key has been generated, the PA messages exchanged within a PA session have little cryptographic protection, and if there is no already-established security channel between two session partners, these messages are subject to man-in-the-middle attacks and DoS attacks. For instance, the initial PA-Server and PA-Client message exchange is vulnerable to spoofing attacks, as these messages are not authenticated and integrity protected. In addition, because the PRF and MAC algorithms are transported at this stage, an attacker may try to remove the PRF and MAC options containing strong algorithms from the initial PA-Server message and force the client to choose the weakest algorithms. Therefore, the server needs to guarantee that all the PRF and MAC algorithms for which it provides support are strong enough.

In order to prevent very basic DoS attacks, a PCP device SHOULD generate state information as little as possible in the initial PA-Server and PA-Client message exchanges. The choice of EAP method is also very important. The selected EAP method must (1) be resilient to attacks that are possible in an insecure network environment, (2) provide user-identity confidentiality and protection against dictionary attacks, and (3) support session-key establishment.

When a PCP proxy [RFC7648] is located between a PCP server and PCP clients, the proxy may perform authentication with the PCP server before it processes requests from the clients. In addition, re-authentication between the PCP proxy and PCP server will not interrupt the service that the proxy provides to the clients, since the proxy is still allowed to send common PCP messages to the PCP server during that period.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <http://www.rfc-editor.org/info/rfc3748>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <http://www.rfc-editor.org/info/rfc4868>.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, DOI 10.17487/RFC5281, August 2008, <http://www.rfc-editor.org/info/rfc5281>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <http://www.rfc-editor.org/info/rfc6887>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <http://www.rfc-editor.org/info/rfc7170>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <http://www.rfc-editor.org/info/rfc7296>.

[RFC7613] Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", RFC 7613, DOI 10.17487/RFC7613, August 2015, <http://www.rfc-editor.org/info/rfc7613>.

[RFC7648] Perreault, S., Boucadair, M., Penno, R., Wing, D., and S. Cheshire, "Port Control Protocol (PCP) Proxy Function", RFC 7648, DOI 10.17487/RFC7648, September 2015, <http://www.rfc-editor.org/info/rfc7648>.

9.2. Informative References

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <http://www.rfc-editor.org/info/rfc5216>.

[RFC5433] Clancy, T. and H. Tschofenig, "Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method", RFC 5433, DOI 10.17487/RFC5433, February 2009, <http://www.rfc-editor.org/info/rfc5433>.

[RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA’)", RFC 5448, DOI 10.17487/RFC5448, May 2009, <http://www.rfc-editor.org/info/rfc5448>.

Acknowledgements

Thanks to Dan Wing, Prashanth Patil, Dave Thaler, Peter Saint-Andre, Carlos Pignataro, Brian Haberman, Paul Kyzivat, Jouni Korhonen, Stephen Farrell, and Terry Manderson for their valuable comments.

Authors' Addresses

Margaret Cullen
Painless Security
356 Abbott Street
North Andover, MA 01845
United States

Phone: +1 781 405 7464
Email: margaret@painless-security.com
URI: http://www.painless-security.com

Sam Hartman
Painless Security
356 Abbott Street
North Andover, MA 01845
United States

Email: hartmans@painless-security.com
URI: http://www.painless-security.com

Dacheng Zhang
Beijing, China
China

Email: zhang_dacheng@hotmail.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com