PANA Working Group
Internet-Draft
Expires: June 29, 2005

D. Forsberg
Nokia
Y. Ohba (Ed.)
Toshiba
B. Patil
Nokia
H. Tschofenig
Siemens
A. Yegin
Samsung

December 29, 2004

Protocol for Carrying Authentication for Network Access (PANA)
draft-ietf-pana-pana-07

Status of this Memo

Copyright Notice

Abstract

This document defines the Protocol for Carrying Authentication for Network Access (PANA), a link-layer agnostic transport for Extensible

Authentication Protocol (EAP) to enable network access authentication between clients and access networks. PANA can carry any authentication method that can be specified as an EAP method, and it can be used on any link that can carry IP. PANA protocol specification covers the client-to-network access authentication part of an overall secure network access framework, which additionally includes other protocols and mechanisms for service provisioning, access control as a result of initial authentication, and accounting.

Table of Contents

1. Introduction

Providing secure network access service requires access control based on the authentication and authorization of the clients and the access networks. Client-to-network authentication provides parameters that are needed to police the traffic flow through the enforcement points. A protocol is needed to carry authentication methods between the client and the access network.

Currently there is no standard network-layer solution for authenticating clients for network access. Appendix A of [I-D.ietf-pana-requirements] describes the problem statement that led to the development of PANA.

Scope of this work is identified as designing a link-layer agnostic transport for network access authentication methods. The Extensible Authentication Protocol (EAP) [RFC3748] provides such authentication methods. In other words, PANA will carry EAP which can carry various authentication methods. By the virtue of enabling transport of EAP above IP, any authentication method that can be carried as an EAP method is made available to PANA and hence to any link-layer technology. There is a clear division of labor between PANA (an EAP lower layer), EAP and EAP methods as described in [RFC3748].

Various environments and usage models for PANA are identified in Appendix A of [I-D.ietf-pana-requirements]. Potential security threats for network-layer access authentication protocol are discussed in [I-D.ietf-pana-threats-eval]. These have been essential in defining the requirements [I-D.ietf-pana-requirements] on the PANA protocol. Note that some of these requirements are imposed by the chosen payload, EAP [RFC3748].

There are components that are part of a complete secure network solution but are outside of the PANA protocol specification, including IP address configuration, authentication method choice, filter rule installation, data traffic protection and PAA-EP protocol. These components are described in separate documents (see [I-D.ietf-pana-framework] and [I-D.ietf-pana-snmp]). The readers are recommended to go through the PANA Framework document [I-D.ietf-pana-framework] prior to reading this protocol specification document.

1.1 Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. 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. Terminology

PANA Client (PaC):

The client side of the protocol that resides in the access device (e.g., laptop, PDA, etc.). It is responsible for providing the credentials in order to prove its identity (authentication) for network access authorization. The PaC and the EAP peer are co-located in the same access device.

PANA Authentication Agent (PAA):

The protocol entity in the access network whose responsibility is to verify the credentials provided by a PANA client (PaC) and authorize network access to the device associated with the client and identified by a Device Identifier (DI). The PAA and the EAP authenticator (and optionally the EAP server) are co-located in the same node. Note the authentication and authorization procedure can, according to the EAP model, be also offloaded to the backend AAA infrastructure.

PANA Session:

A PANA session begins with the handshake between the PANA Client (PaC) and the PANA Authentication Agent (PAA), and terminates as a result of an authentication or liveness test failure, a message delivery failure after retransmissions reach maximum values, session lifetime expiration, or an explicit termination message. A fixed session identifier is maintained throughout a session. A session cannot be shared across multiple network interfaces. Only one device identifier of the PaC is allowed to be bound to a PANA session for simplicity.

Session Identifier:

This identifier is used to uniquely identify a PANA session on the PAA and PaC. It includes an identifier of the PAA, therefore it cannot be shared across multiple PAAs. It is included in PANA messages to bind the message to a specific PANA session. This bidirectional identifier is allocated by the PAA following the handshake and freed when the session terminates.

PANA Security Association (PANA SA):

A PANA security association is formed between the PaC and the PAA by sharing cryptographic keying material and associated context. The formed duplex security association is used to protect the bidirectional PANA signaling traffic between the PaC and the PAA.

Device Identifier (DI):

The identifier used by the network as a handle to control and police the network access of a device. Depending on the access technology, this identifier may contain an address that is carried in protocol headers (e.g., IP or link-layer address), or a locally significant identifier that is made available by the local protocol stack (e.g., circuit id, PPP interface id) of a connected device.

Enforcement Point (EP):

A node on the access network where per-packet enforcement policies (i.e., filters) are applied on the inbound and outbound traffic of access devices. Information such as the DI and (optionally) cryptographic keys are provided by the PAA per client for generating filters on the EP. The EP and PAA may be co-located.

Network Access Provider (NAP):

A service provider that provides physical and link-layer connectivity to an access network it manages.

AAA-Key:

A key derived by the EAP peer and EAP server and transported to the authenticator [I-D.ietf-eap-keying].

For additional terminology definitions see the PANA framework document [I-D.ietf-pana-framework].

3. Protocol Overview

The PANA protocol is run between a client (PaC) and a server (PAA) in order to perform authentication and authorization for the network access service.

The protocol messaging consists of a series of request and responses, some of which may be initiated by either ends. Each message can carry zero or more AVPs as payload. The main payload of PANA is EAP which performs authentication. PANA helps the PaC and PAA establish an EAP session.

PANA is a UDP-based protocol. It has its own retransmission mechanism to reliably deliver messages.

PANA messages are sent between the PaC and PAA as part of a PANA session. A PANA session consists of distinct phases:

- Discovery and handshake phase: This is the phase that initiates a new PANA session. The PaC discovers the PAA(s) by either explicitly soliciting advertisements for them or receiving unsolicited advertisements. The PaC's answer sent in response to an advertisement starts a new session.

- Authentication and authorization phase: Immediately following the discovery and handshake phase is the EAP execution between the PAA and PaC. The EAP payload (which carry an EAP method inside) is what is used for authentication. The PAA conveys the result of authentication and authorization to the PaC at the end of this phase. This phase may involve execution of two EAP sessions back-to-back, one for the NAP and one for the ISP.

- Access phase: After a successful authentication and authorization the host gains access to the network and can send and receive IP data traffic through the EP(s). At any time during this phase, the PaC and PAA may optionally ping each other to test liveness of the PANA session on each end.

- Re-authentication phase: Following the access phase, the PAA must initiate re-authentication before the PANA session lifetime expires. Again EAP is carried by PANA to perform authentication. This phase may be optionally triggered by both the PaC and the PAA without any respect to the session lifetime. The session moves to this phase from the access phase, and returns back there upon successful re-authentication.

- Termination phase: The PaC or PAA may choose to discontinue the access service at any time. An explicit disconnect message can be

sent by either end. If either the PaC or the PAA disconnects without engaging in termination messaging, it is expected that either the expiration of a finite session lifetime or failed liveness tests would do the job.

```
  PaC  PAA    Message
-----------------------------------------------------
// Discovery and handshake phase
   ----->     PANA-PAA-Discover
   <-----     PANA-Start-Request
   ----->     PANA-Start-Answer

// Authentication and authorization phase
   <-----     PANA-Auth-Request /* EAP Request */
   ----->     PANA-Auth-Answer
   ----->     PANA-Auth-Request /* EAP Response */
   <-----     PANA-Auth-Answer
   <-----     PANA-Bind-Request /* EAP Success */
   ----->     PANA-Bind-Answer

// Access phase (IP data traffic allowed)
   <-----     PANA-Ping-Request
   ----->     PANA-Ping-Answer

// Termination phase
   ----->     PANA-Termination-Request
   <-----     PANA-Termination-Answer
```

Figure 1: Illustration of PANA messages in a session

Note that depending on the environment and deployment the protocol flow depicted in Figure 1 can be abbreviated.

Cryptographic protection of messages between the PaC and PAA is possible as soon as EAP in conjunction with the EAP method exports a shared key. That shared key is used to create a PANA SA. The PANA SA helps generating per-message authentication codes that provide integrity protection and authentication.

Throughout the lifetime of a session, various problems found with the incoming messages can generate a PANA error message sent in response.

4. Protocol Details

The following sections explain in detail the various phases of a PANA session.

4.1 Payload Encoding

The payload of any PANA message consists of zero or more AVPs (Attribute Value Pairs). The subsequent sections refer to these AVPs, therefore the list of AVPs are provided with a brief description before more extensive descriptions are included later in the document.

- Cookie AVP: contains a random value that is generated by the PAA and used for making PAA discovery robust against blind resource consumption DoS attacks.

- Protection-Capability AVP: contains the type of per-packet protection (link-layer vs. network-layer) when a cryptographic mechanism should be enabled after PANA authentication.

- Device-Id AVP: contains a device identifier (link-layer address or an IP address) of the PaC or an EP.

- EAP AVP: contains an EAP PDU.

- MAC AVP: contains a Message Authentication Code that integrity protects the PANA message.

- Termination-Cause AVP: contains the reason of session termination.

- Result-Code AVP: contains information about the protocol execution results.

- Session-Id AVP: contains the PANA session identifier value.

- Session-Lifetime AVP: contains the duration of authorized access.

- Failed-AVP: contains an offending AVP that caused a failure.

- Provider-Identifier AVP: contains the identifier of a NAP or an ISP.

- Provider-Name AVP: contains a name of a NAP or an ISP.

- NAP-Information AVP, ISP-Information AVP: contains the identifier of a NAP and an ISP, respectively.

- o Key-Id AVP: contains a AAA-Key identifier.

- o PPAC AVP: Post-PANA-Address-Configuration AVP. Used to indicate the available/chosen IP address configuration methods that can be used by the PaC after successful PANA authentication.

- o Nonce AVP: contains a randomly chosen value that is used in cryptographic key computations.

- o IP-Address AVP: contains an IP Address of the PaC.

- o Notification AVP: contains a displayable message.

4.2 Discovery and Handshake Phase

When a PaC attaches to a network, and knows that it has to discover a PAA, it SHOULD send a PANA-PAA-Discover message to a well-known link local multicast address (TBD) and UDP port (TBD). The PAA discovery assumes that the PaC and the PAA are one IP hop away from each other. If the PaC knows the IP address of the PAA (based on pre-configuration), it MAY unicast the PANA-PAA-Discover message to that address.

When the PAA receives a PANA-PAA-Discover message from a PaC, the PAA SHOULD unicast a PANA-Start-Request message to the PaC.

The PaC MAY also choose to start sending data packets before getting authenticated. The EP in an access network that implements PANA SHOULD drop unauthorized packets upon receipt. Additionally, the EP MAY also take this traffic as an indication of unauthorized PaC and notify the PAA. The EP-to-PAA notification SHOULD be sent via [I-D.ietf-pana-snmp]. In response, the PAA SHOULD send an unsolicited PANA-Start-Request message to the PaC. This is called "traffic-driven PAA discovery" (an alternative to the PaC explicitly soliciting for a PAA). Note that this optional feature MAY NOT be present in all deployments, therefore the PaC MUST NOT assume its availability. The EP-to-PAA notification MAY also be generated in response to receiving a link-up event notification on the EP [I-D.ietf-dna-link-information].

When the PaC receives a PANA-Start-Request message from a PAA, it responds with a PANA-Start-Answer message if it wishes to enter the authentication and authorization phase.

There can be multiple PAAs on the link and the PaC may receive multiple PANA-Start-Request messages from those PAAs. The authentication and authorization result does not depend on which PAA is chosen by the PaC. By default the PaC MAY choose the PAA that

sent the first response.

A PANA-Start-Request message MAY carry a Cookie AVP that contains a random value generated by the PAA. The random value is referred to as a cookie. The cookie is used for preventing the PAA from resource consumption DoS attacks by blind attackers which bombard the PAA with PANA-PAA-Discover messages. By relying on a cookie mechanism the PAA can avoid per-PaC state creation until after the PaC can produce the same cookie in its PANA-Start-Answer message. In order to do that, the cookie MUST be computed in such a way that it does not require any per-session state maintenance on the PAA in order to verify the cookie returned in the PANA-Start-Answer message. The PAA discovery that takes advantage of cookies is called "stateless PAA discovery". The exact algorithms and syntax used by the PAA to generate cookies does not affect interoperability and hence is not specified here. An example algorithm is described below.

```
Cookie =
  <secret-version> | HMAC_SHA1( <Device-Id of PaC> , <secret> )
```

where <secret> is a randomly generated secret known only to the PAA, <secret-version> is an index used for choosing the secret for generating the cookie and '|' indicates concatenation. The secret-version should be changed frequently enough to prevent replay attacks. The secret key is valid for a certain time frame. The device identifier of the PaC can be extracted from a link-layer or IP header of PANA messages.

When the PaC sends a PANA-Start-Answer message in response to a PANA-Start-Request containing a Cookie AVP, the answer MUST contain a Cookie AVP with the cookie value copied from the request.

When the PAA receives the PANA-Start-Answer message from the PaC, it verifies the cookie. The cookie is considered as valid if the received cookie has the expected value. If the computed cookie is valid, the protocol enters the authentication and authorization phase. Otherwise, it MUST silently discard the received message.

The initial EAP Request message MAY be optionally carried by the PANA-Start-Request (as opposed to by a later PANA-Auth-Request) message in order to reduce the number of round-trips. This optimization SHOULD NOT be used if the PAA discovery is desired to be stateless since transmission of an EAP Request message creates a state at EAP layer. See [I-D.ietf-eap-statemachine] for more information on the EAP state machine and the allocation of state information in the respective protocol steps.

A Protection-Capability AVP and a Post-PANA-Address-Configuration

(PPAC) AVP MAY be included in the PANA-Start-Request in order to indicate required and available capabilities for the network access. These AVPs MAY be used by the PaC for assessing the capability match even before the authentication takes place. Since these AVPs are provided during the insecure discovery and handshake phase, there are certain security risks involved in using the provided information. See Section 10 for further discussion on this.

If the initial EAP Request message is carried in the PANA-Start-Request message, an EAP Response message MUST be carried in the PANA-Start-Answer message returned to the PAA.

The PANA-Start-Request/Answer exchange is needed before entering the authentication and authorization phase even when the PaC is pre-configured with the IP address of the PAA and the PANA-PAA-Discover message is unicast.

A Nonce AVP MUST be included in the PANA-Start-Request and PANA-Start-Answer messages. The nonces are used to establish a fresh PANA_MAC_KEY (see Section 5.3) which is a transient session key in the EAP key hierarchy [I-D.ietf-eap-keying] and is used only in the PANA protocol. A Nonce AVP MUST be included in the PANA-Start-Request and PANA-Start-Answer messages. The nonces are used to establish a PANA SA.

A PANA-Start-Request message in stateless PAA discovery MUST NOT be retransmitted as this voids the statelessness on the PAA. Instead, the PaC MUST retransmit the PANA-PAA-Discover message until it receives a PANA-Start-Request message, and retransmit the PANA-Start-Answer message until it receives a PANA-Auth-Request message. The PaC can determine whether the PAA is using stateless PAA discovery by the presence of Cookie AVP. The PANA-Start-Request message MUST be retransmitted instead of the PANA-Start-Answer message when stateless PAA discovery is not used.

It is possible that both the PAA and the PaC initiate the discovery and handshake procedure at the same time, i.e., the PAA sends a PANA-Start-Request message while the PaC sends a PANA-PAA-Discover message. To resolve the race condition, the PAA SHOULD silently discard the PANA-PAA-Discover message received from the PaC after it has sent a PANA-Start-Request message with creating a state (i.e., no Cookie AVP is included in the message) for the PaC. In this case the PAA will retransmit the PANA-Start-Request message based on a timer, if the PaC doesn't respond in time (the message was lost for example). If the PAA had sent a PANA-Start-Request message without creating a state for the PaC (i.e., a Cookie AVP was included in the message), then it SHOULD answer to the PANA-PAA-Discover message.

Figure 2 shows an example sequence for the discovery and handshake phase when a PANA-PAA-Discover message is sent by the PaC. Figure 3 shows an example sequence for the discovery and handshake phase with traffic-driven PAA discovery.

```
   PaC      PAA         Message(sequence number)[AVPs]
   ------------------------------------------------------
      ----->            PANA-PAA-Discover(0)
      <-----            PANA-Start-Request(x)[Nonce, Cookie]
      ----->            PANA-Start-Answer(x)[Nonce, Cookie]
                        (continued to the authentication and
                         authorization phase)
```

Figure 2: Example sequence for the discovery and handshake phase when PANA-PAA-Discover is sent by the PaC

```
   PaC   EP      PAA    Message(sequence number)[AVPs]
   ------------------------------------------------------
    ---->o              (Data packet arrival or L2 trigger)
          ------>       PAA-to-EP protocol, or another mechanism
    <-------------      PANA-Start-Request(x)[Nonce, Cookie]
    ------------->      PANA-Start-Answer(x)[Nonce, Cookie]
                        (continued to the authentication and
                         authorization phase)
```

Figure 3: Example sequence for the discovery and handshake phase with traffic-driven PAA discovery

4.3 Authentication and Authorization Phase

The main task of the authentication and authorization phase is to carry EAP messages between the PaC and the PAA. EAP Request and Response messages are carried in PANA-Auth-Request messages. PANA-Auth-Answer messages are simply used to acknowledge receipt of the requests. As an optimization, a PANA-Auth-Answer message MAY include the EAP Response message. This optimization MAY not be used when it takes time to generate the EAP Response message (due to, e.g., intervention of human input), in which case returning an EAP-Auth-Answer message without piggybacking an EAP Response message can avoid unnecessary retransmission of the PANA-Auth-Request message. Another optimization allows optionally carrying the first EAP Request/Response message in PANA-Start-Request/Answer message as described in Section 4.2.

PANA allows execution of two separate authentication methods, one with NAP and one with ISP under the same PANA session. This optional

feature may be offered by the PAA and accepted by the PaC. When performed separately, the result of the first EAP authentication is signaled via PANA-FirstAuth-End-Request and PANA-FirstAuth-End-Answer message exchange which delineates the first method execution from the next. See Section 4.7 for a detailed discussion on separate NAP and ISP authentication.

The result of PANA authentication is carried in a PANA-Bind-Request message sent from the PAA to the PaC. This message carries the final EAP authentication result (whether it is the second EAP authentication result of NAP and ISP separate authentication, or the sole EAP authentication result) and the result of PANA authentication. The PANA-Bind-Request message MUST be acknowledged with a PANA-Bind-Answer (PBA) message. Figure 4 shows an example sequence in the authentication and authorization phase (no separate authentication).

```
   PaC      PAA  Message(sequence number)[AVPs]
   ---------------------------------------------------------------------
                 (continued from the discovery and handshake phase)
      <-----     PANA-Auth-Request(x+1)
                    [Session-Id, EAP{Request}]
      ----->     PANA-Auth-Answer(x+1)      // No piggybacking EAP Response
                    [Session-Id]
      ----->     PANA-Auth-Request(y)
                    [Session-Id, EAP{Response}]
      <-----     PANA-Auth-Answer(y)
                    [Session-Id]
      <-----     PANA-Auth-Request(x+2)
                    [Session-Id, EAP{Request}]
      ----->     PANA-Auth-Answer(x+2)      // Piggybacking EAP Response
                    [Session-Id, EAP{Response}]
      <-----     PANA-Bind-Request(x+3)
                    [Session-Id, Result-Code, EAP{Success}, Device-Id,
                     Key-Id, IP-Address, Lifetime, Protection-Cap., PPAC,
MAC]
      ----->     PANA-Bind-Answer(x+3)
                    [Session-Id, Device-Id, Key-Id, PPAC, MAC]
```

Figure 4: Example sequence for the authentication and authorization phase

When an EAP method that is capable of deriving keys is used during the authentication and authorization phase and the keys are successfully derived, the PANA message that carries the EAP Success message (i.e., a PANA-FirstAuth-End-Request or a PANA-Bind-Request message) and any subsequent message MUST contain a MAC AVP.

The PANA-Bind-Request and the PANA-Bind-Answer message exchange is also used for binding device identifiers of the PaC and EP(s), and the IP address of the PAA to the PANA SA. To achieve this, the PANA-Bind-Request message MUST contain the device identifier in a Device-Id AVP for each EP if a Protection-Capability AVP is included in the message. Otherwise, the message SHOULD contain the device identifier in a Device-Id AVP for each EP when a link-layer or IP address is used as the device identifier of the PaC. The PANA-Bind-Request message MUST also contain the IP address of the PAA in an IP-Address AVP. The PANA-Bind-Answer message MUST contain the PaC's device identifier in a Device-Id AVP when it is already presented with that of EP(s) in the request with using the same type of device identifier as contained in the request. If the PANA-Bind-Answer message sent from the PaC does not contain a Device-Id AVP with the same device identifier type contained in the request, the PAA sends a PANA-Error-Request message with a PANA_MISSING_AVP result code, and wait for a PANA-Error-Answer message to terminate the session. The PANA-Bind-Request message with a PANA_SUCCESS result code MUST also contain a Protection-Capability AVP if link-layer or network-layer ciphering is enabled after the authentication and authorization phase. The PANA-Bind-Request message MAY also contain a Protection-Capability AVP to indicate if link-layer or network-layer ciphering should be enabled after the authentication and authorization phase. No link-layer or network-layer specific information is included in the Protection-Capability AVP. It is assumed that the PAA is aware of the security capabilities of the access network. The PANA protocol does not specify how the PANA SA and the Protection-Capability AVP will be used to provide per-packet protection for data traffic.

Additionally, the PANA-Bind-Request message with a PANA_SUCCESS result code MUST include a Post-PANA-Address-Configuration (PPAC) AVP, which helps the PAA to inform the PaC about whether a new IP address MUST be configured and the available methods to do so. In this case, the PaC MUST include a PPAC AVP in the PANA-Bind-Answer message in order to indicate its choice of method when there is a match between the methods offered by the PAA and the methods available on the PaC. When there is no match, the PaC MUST send a PANA-Error-Request message with a PANA_PPAC_CAPABILITY_UNSUPPORTED result code and terminate the PANA session.

PANA-Bind-Request and PANA-Bind-Answer messages MUST be retransmitted based on the retransmission rule described in Section 5.2.

EAP authentication can fail at a pass-through authenticator without sending an EAP Failure message [I-D.ietf-eap-statemachine]. When this occurs, the PAA SHOULD send a PANA-Error-Request message to the PaC with using PANA_UNABLE_TO_COMPLY result code. The PaC SHOULD not change its state unless the error message is secured by PANA or

lower-layer. In any case, a more appropriate way is to rely on a timeout on the PaC.

There is a case where EAP authentication succeeds with producing an EAP Success message but network access authorization fails due to, e.g., authorization rejected by a AAA or authorization locally rejected by the PAA. When this occurs, the PAA MUST send a PANA-Bind-Request with a result code PANA_AUTHORIZATION_REJECTED. If a AAA-Key is established between the PaC and the PAA by the time when the EAP Success message is generated by the EAP server (this is the case when the EAP method provides protected success indication), the PANA-Bind-Request and PANA-Bind-Answer messages MUST be protected with a MAC AVP and carry a Key-Id AVP. The AAA-Key and the PANA session MUST be deleted immediately after the PANA-Bind message exchange.

4.4 Access Phase

Once the authentication and authorization phase or the re-authentication phase successfully completes, the PaC gains access to the network and can send and receive IP data traffic through the EP(s) and the PANA session enters the access phase. In this phase, PANA-Ping-Request and PANA-Ping-Answer messages can be used for testing the liveness of the PANA session on the PANA peer. Both the PaC and the PAA are allowed to send a PANA-Ping-Request message to the communicating peer whenever they need to make sure the availability of the session on the peer and expect the peer to return a PANA-Ping-Answer message. Both PANA-Ping-Request and PANA-Ping-Answer messages MUST be protected with a MAC AVP when a PANA SA is available.

Implementations MUST limit the rate of performing this test. The PaC and the PAA can handle rate limitation on their own, they do not have to perform any coordination with each other. There is no negotiation of timers for this purpose.

Figure 5 and Figure 6 show liveness tests as they are initiated by the PaC and the PAA respectively.

```
PaC      PAA     Message(sequence number)[AVPs]
------------------------------------------------------
   ----->        PANA-Ping-Request(q)[Session-Id, MAC]
   <-----        PANA-Ping-Answer(q)[Session-Id, MAC]
```

Figure 5: Example sequence for PaC-initiated liveness test

```
PaC      PAA     Message(sequence number)[AVPs]
------------------------------------------------------
   <-----        PANA-Ping-Request(p)[Session-Id, MAC]
   ----->        PANA-Ping-Answer(p)[Session-Id, MAC]
```

Figure 6: Example sequence for PAA-initiated liveness test

4.5 Re-authentication Phase

The PANA session in the access phase can enter the re-authentication phase to extend the current session lifetime by re-executing EAP. Once the re-authentication phase successfully completes, the session re-enters the access phase. Otherwise, the session is deleted.

When the PaC wants to initiate re-authentication, it sends a PANA-Reauth-Request message to the PAA. This message MUST contain a Session-Id AVP which is used for identifying the PANA session on the PAA. If the PAA already has an established PANA session for the PaC with the matching session identifier, it MUST first respond with a PANA-Reauth-Answer message, followed by a PANA-Auth-Request that starts a new EAP authentication. If the PAA cannot identify the session, it MAY respond with a PANA-Error-Request message with a result code PANA_UNKNOWN_SESSION_ID. Transmission of this error request is made optional in case this behavior is leveraged for a DoS attack on the PAA.

The PaC may receive a PANA-Auth-Request before receiving the answer to its outstanding PANA-Reauth-Request. This condition can arise due to packet re-ordering or a race condition between the PaC and PAA when they both attempt to engage in re-authentication. The PaC MUST keep discarding the received PANA-Auth-Requests until it receives the answer to its request.

When the PAA initiates re-authentication, it sends a PANA-Auth-Request message containing the session identifier for the PaC to enter the re-authentication phase. The PAA SHOULD initiate EAP re-authentication before the current session lifetime expires.

Re-authentication of an on-going PANA session MUST maintain the existing sequence numbers.

For any re-authentication, if there is an established PANA SA, PANA-Auth-Request and PANA-Auth-Answer messages MUST be protected by adding a MAC AVP to each message. Any subsequent EAP authentication MUST be performed with the same ISP and NAP that was selected during the discovery and handshake phase. An example sequence for re-authentication phase initiated by the PaC is shown in Figure 7.

```
   PaC      PAA  Message(sequence number)[AVPs]
   ------------------------------------------------------
      ----->     PANA-Reauth-Request(q)
                    [Session-Id, MAC]
      <-----     PANA-Reauth-Answer(q)
                    [Session-Id, MAC]
      <-----     PANA-Auth-Request(p)
                    [Session-Id, EAP{Request}, MAC]
      ----->     PANA-Auth-Answer(p)        // No piggybacking EAP Response
                    [Session-Id, MAC]
      ----->     PANA-Auth-Request(q+1)
                    [Session-Id, EAP{Response}, MAC]
      <-----     PANA-Auth-Answer(q+1)      // No piggybacking EAP Response
                    [Session-Id, MAC]
      <-----     PANA-Auth-Request(p+1)
                    [Session-Id, EAP{Request}, MAC]
      ----->     PANA-Auth-Answer(p+1)      // Piggybacking EAP Response
                    [Session-Id, EAP{Response}, MAC]
      <-----     PANA-Bind-Request(p+2)
                    [Session-Id, Result-Code, EAP{Success}, Device-Id, Key-
Id,
                     IP-Address, Lifetime, Protection-Cap., PPAC, MAC]
      ----->     PANA-Bind-Answer(p+2)
                    [Session-Id, Device-Id, Key-Id, PPAC, MAC]
```

Figure 7: Example sequence for the re-authentication phase initiated by PaC

4.6 Termination Phase

A procedure for explicitly terminating a PANA session can be initiated either from the PaC (i.e., disconnect indication) or from the PAA (i.e., session revocation). The PANA-Termination-Request and PANA-Termination-Answer message exchanges are used for disconnect indication and session revocation procedures.

The reason for termination is indicated in the Termination-Cause AVP.

When there is an established PANA SA between the PaC and the PAA, all messages exchanged during the termination phase MUST be protected with a MAC AVP. When the sender of the PANA-Termination-Request message receives a valid acknowledgment, all states maintained for the PANA session MUST be deleted immediately.

```
   PaC      PAA     Message(sequence number)[AVPs]
   ------------------------------------------------------
      ----->        PANA-Termination-Request(q)[Session-Id, MAC]
      <-----        PANA-Termination-Answer(q)[Session-Id, MAC]
```

Figure 8: Example sequence for the termination phase triggered by PaC

4.7 Separate NAP and ISP Authentication

PANA allows running at most two EAP sessions in sequence in the authentication and authorization phase to support separate NAP and ISP authentication as described in this section. A typical network access authentication includes execution of one EAP method with the ISP. This separation allows the PaC to perform an additional authentication method for receiving differentiated services from the NAP.

Currently, running multiple EAP sessions in sequence in the authentication and authorization phase is designed only for separate NAP and ISP authentication. It is not for running arbitrary number of EAP sessions in sequence, or giving the PaC another chance to try another EAP authentication method within an integrated NAP and ISP authentication when an EAP authentication method fails.

Within separate NAP and ISP authentication, the NAP authentication and the ISP authentication are considered completely independent. Presence or success of one should not effect the other. Making a network access authorization decision based on the success or failure of each authentication is a network policy issue.

4.7.1 Negotiating Separate NAP and ISP Authentication

When the PaC and PAA negotiates in the discovery and handshake phase to perform separate NAP and ISP authentication, the PaC and the PAA operate in the following way in addition to the behavior defined in Section 4.2

In the discovery and handshake phase, the PAA MAY advertise availability of separate NAP and ISP authentication ([I-D.ietf-pana-framework]) by setting the S-flag on the PANA header of the PANA-Start-Request message.

If the S-flag of the received PANA-Start-Request message is set, the PaC can indicate its desire to perform separate NAP and ISP authentication by setting the S-flag in the PANA-Start-Answer message. If the S-flag of the received PANA-Start-Request message is not set, the PaC MUST NOT set the S-flag in the PANA-Start-Answer message sent back to the PAA.

If the S-flag in the PANA-Start-Answer message is not set, only one authentication is performed (ISP-only) and the processing occurs as described in Section 4.2.

When the S-flag is set in a PANA-Start-Request message, the initial EAP Request message MUST NOT be carried in the PANA-Start-Request message. (If the initial EAP Request message were contained in the PANA-Start-Request message during the S-flag negotiation, the PaC cannot tell whether the EAP Request message is for NAP authentication or ISP authentication.)

4.7.2 Execution of Separate NAP and ISP Authentication

When the PaC and PAA have negotiated in the discovery and handshake phase to perform separate NAP and ISP authentication, the PaC and the PAA operate in the following way in addition to the behavior defined in Section 4.3

- The S-flag of PANA-Auth-Request and PANA-Auth-Answer messages MUST be set.

- An EAP Success/Failure message is carried in a PANA-FirstAuth-End-Request (PFER) message as well as a PANA-Bind-Request (PBR) message. The PANA-FirstAuth-End-Request message MUST be used at the end of the first EAP authentication and the PANA-Bind-Request MUST be used for the second EAP authentication. The PANA-FirstAuth-End-Request messages MUST be acknowledged with a PANA-FirstAuth-End-Answer (PFEA) message.

- If the first EAP authentication has failed, the PAA can choose not to perform the second EAP authentication by clearing the S-flag of the PANA-FirstAuth-End-Request message. In this case, the S-flag of the PANA-FirstAuth-End-Answer message sent by the PaC MUST be cleared. If the S-flag of the PANA-FirstAuth-End-Request message is set when the first EAP authentication has failed, the PaC can choose not to perform the second EAP authentication by clearing the S-flag of the PANA-FirstAuth-End-Answer message. If the first EAP authentication failed and the S-flag is not set in the PANA-FirstAuth-End-Answer message as a result of those operations, the PANA session MUST be immediately deleted. Otherwise, the

second EAP authentication MUST be performed.

- o The PAA determines the execution order of NAP authentication and ISP authentication. In this case, the PAA can indicate which authentication (NAP authentication or ISP authentication) is currently occurring by using N-flag in the PANA message header. When NAP authentication is being performed, the N-flag MUST be set. When ISP authentication is being performed, the N-flag MUST NOT be set. The N-flag MUST NOT be set when S-flag is not set.

When the PaC and PAA have negotiated in the discovery and handshake phase to perform separate NAP and ISP authentication, and the lower-layer is insecure, the two EAP authentication methods used in the separate authentication MUST be capable of deriving keys (AAA-Key).

4.7.3 AAA-Key Calculation

When the PaC and PAA have negotiated in the discovery and handshake phase to perform separate NAP and ISP authentication, if the lower-layer is insecure, the two EAP authentication methods used in the separate authentication MUST be capable of deriving keys. In this case, if the first EAP authentication is successful, the PANA-FirstAuth-End-Request and PANA-FirstAuth-End-Answer messages as well as PANA-Auth-Request and PANA-Auth-Answer messages in the second EAP authentication MUST be protected with the key derived from the AAA-Key for the first EAP authentication. The PANA-Bind-Request and PANA-Bind-Answer messages and all subsequent PANA messages exchanged in the access phase, re-authentication phase and termination phase MUST be protected either with the AAA-Key for the first EAP authentication if the first EAP authentication succeeds and the second EAP authentication fails, or with the AAA-Key for the second EAP authentication if the first EAP authentication fails and the second EAP authentication succeeds, or with the compound AAA-Key derived from the two AAA-Keys, one for the first EAP authentication and the other from the second EAP authentication, if both the first and second EAP authentication succeed. See Section 5.3 for how to derive the AAA-Key.

5. Protocol Design Details and Processing Rules

5.1 Transport Layer

PANA uses UDP as its transport layer protocol. The UDP port number is TBD. All messages except for PANA-PAA-Discover are always unicast. The PANA-PAA-Discover message MAY be unicast when the PaC knows the IP address of the PAA.

5.1.1 Fragmentation

PANA does not provide fragmentation of PANA messages. Instead, it relies on fragmentation provided by EAP methods and IP layer when needed.

5.2 Sequence Number and Retransmission

PANA uses sequence numbers to provide ordered and reliable delivery of messages.

The PaC and PAA maintain two sequence numbers: the next one to be used for a request it initiates and the next one it expects to see in a request from the other end. These sequence numbers are 32-bit unsigned numbers. They are monotonically incremented by 1 as new requests are generated and received, and wrapped to zero on the next message after 2^32-1. Answers always contain the same sequence number as the corresponding request. Retransmissions reuse the sequence number contained in the original packet.

The initial sequence numbers (ISN) are randomly picked by the PaC and PAA as they send their very first request messages. PANA-PAA-Discover message carries sequence number 0.

When a request message is received, it is considered valid in terms of sequence numbers if and only if its sequence number matches the expected value. This check does not apply to the PANA-PAA-Discover, PANA-Start-Request messages.

When an answer message is received, it is considered valid in terms of sequence numbers if and only if its sequence number matches that of the currently outstanding request. A peer can only have one outstanding request at a time.

PANA messages are retransmitted based on a timer until a response is received (in which case the retransmission timer is stopped) or the number of retransmission reaches the maximum value (in which case the PANA session MUST be deleted immediately).

The initial discovery and handshake phase requires special handling. The PaC MUST retransmit the PANA-PAA-Discover message if a subsequent PANA-Start-Request message is not received in time. Even though a PANA-Start-Request message is received, the PANA-PAA-Discover message may still have to be retransmitted. This is because stateless PAA discovery requires one time transmission of a solicited PANA-Start-Request message. The PAA MUST NOT start a timer and retransmit the request in order to avoid state creation. If the received PANA-Start-Request message included a Cookie AVP (an indication of stateless PAA discovery), the PaC MUST retransmit the PANA-PAA-Discover message until the first PANA-Auth-Request message is received. Otherwise, the PaC can rely on the PAA to retransmit the PANA-Start-Request message as soon as the PaC receives the first one (i.e., the PaC can stop sending the PANA-PAA-Discover message).

The retransmission timers SHOULD be calculated as described in [RFC2988] to provide congestion control. See Section 8 for default timer and maximum retransmission count parameters.

The PaC and PAA MUST respond to duplicate requests. The last transmitted answer MAY be cached in case it is not received by the peer and that generates a retransmission of the last request. When available, the cached answer can be used instead of fully processing the retransmitted request and forming a new answer from scratch.

PANA MUST NOT generate EAP message duplication. EAP payload of a retransmitted PANA message MUST NOT be passed to the EAP layer.

5.3 PANA Security Association

A PANA SA is created as an attribute of a PANA session when EAP authentication succeeds with a creation of a AAA-Key. A PANA SA is not created when the PANA authentication fails or no AAA-Key is produced by any EAP authentication method. In the case where two EAP sessions are performed in sequence in the PANA authentication and authorization phase, it is possible that two AAA-Keys are derived. If this happens, the PANA SA MUST be generated from both AAA-Keys. When a new AAA-Key is derived in the PANA re-authentication phase, any key derived from the old AAA-Key MUST be updated to a new one that is derived from the new AAA-Key. In order to distinguish the new AAA-Key from old ones, one Key-Id AVP MUST be carried in PANA-Bind-Request and PANA-Bind-Answer messages or PANA-FirstAuth-End-Request and PANA-FirstAuth-End-Answer messages at the end of the EAP authentication which resulted in deriving a new AAA-Key. The Key-Id AVP is of type Unsigned32 and MUST contain a value that uniquely identifies the AAA-Key within the PANA session. The PANA-Bind-Answer message (or the PANA-FirstAuth-End-Answer message) sent in response to a PANA-Bind-Request message (or a

PANA-FirstAuth-End-Request message) with a Key-Id AVP MUST contain a Key-Id AVP with the same AAA-Key identifier carried in the request. PANA-Bind-Request, PANA-Bind-Answer, PANA-FirstAuth-End-Request and PANA-FirstAuth-End-Answer messages with a Key-Id AVP MUST also carry a MAC AVP whose value is computed by using the new PANA_MAC_KEY derived from the new AAA-Key (or the new pair of AAA-Keys when the PANA_MAC_KEY is derived from two AAA-Keys). Although the specification does not mandate a particular method for calculation of the Key-Id AVP value, a simple method is to use monotonically increasing numbers.

The PANA session lifetime is bounded by the lifetime granted by the authentication server (same as the AAA-Key lifetime). The lifetime of the PANA SA (hence the PANA_MAC_KEY) is the same as the lifetime of the PANA session. The created PANA SA is deleted when the corresponding PANA session is deleted.

PANA SA attributes as well as PANA session attributes are listed below:

PANA Session attributes:

* Session-Id

* Device-Id of PaC

* IP address of PaC (may be the same as the Device-Id of PaC when IP address is used as the device identifier)

* IP address of PAA

* List of device identifiers of EPs

* Sequence number of the last transmitted request

* Sequence number of the last received request

* Last transmitted message payload

* Retransmission interval

* Session lifetime

* Protection-Capability

* PANA SA attributes:

+ Nonce generated by PaC (PaC_nonce)

+ Nonce generated by PAA (PAA_nonce)

+ AAA-Key

+ AAA-Key Identifier

+ PANA_MAC_KEY

The PANA_MAC_KEY is derived from the available AAA-Key(s) and it is used to integrity protect PANA messages. If there is only one AAA-Key available, e.g., due to ISP-only authentication, or with one failed and one successful separate NAP and ISP authentication (see Section 4.7), the PANA_MAC_KEY computation is based on that single key. Otherwise, two AAA-Keys available to PANA can be combined in following way ('|' indicates concatenation):

```
AAA-Key = AAA-Key1 | AAA-Key2
```

The PANA_MAC_KEY is computed in the following way:

```
PANA_MAC_KEY = The first N bits of
               HMAC_SHA1(AAA-Key, PaC_nonce | PAA_nonce | Session-ID)
```

where the value of N depends on the integrity protection algorithm in use, i.e., N=160 for HMAC-SHA1. The length of the AAA-Key MUST be N bits or longer. See Section Section 5.4 for the detailed usage of the PANA_MAC_KEY.

5.4 Message Authentication Code

A PANA message can contain a MAC (Message Authentication Code) AVP for cryptographically protecting the message.

When a MAC AVP is included in a PANA message, the value field of the MAC AVP is calculated by using the PANA_MAC_KEY in the following way:

```
MAC AVP value = PANA_MAC_PRF(PANA_MAC_KEY, PANA_PDU)
```

where PANA_PDU is the PANA message including the PANA header, with the MAC AVP value field first initialized to 0. PANA_MAC_PRF represents the pseudo random function corresponding to the MAC algorithm specified in the MAC AVP. In this version of draft, PANA_MAC_PRF is HMAC-SHA1. The PaC and PAA MUST use the same algorithm to calculate a MAC AVP they originate and receive. The algorithm is determined by the PAA when a PANA-Bind-Request with a MAC AVP is sent. When the PaC does not support the MAC algorithm

specified in the PANA-Bind-Request message, it MUST silently discard the message. The PAA MUST NOT change the MAC algorithm throughout the continuation of the PANA session.

5.5 Message Validity Check

When a PANA message is received, the message is considered to be invalid at least when one of the following conditions are not met:

o The IP Hop Limit (or TTL) field has a value of 255, i.e., the packet could not possibly have been forwarded by a router.

o Each field in the message header contains a valid value including sequence number, message length, message type, version number, flags, etc.

o The message type is one of the expected types in the current state. Specifically the following messages are unexpected and invalid:

  * In the discovery and handshake phase:

    + PANA-Termination-Request and PANA-Ping-Request.

    + PANA-Bind-Request.

    + PANA-Update-Request.

    + PANA-Reauth-Request.

    + PANA-Error-Request.

  * In the authentication and authorization phase and the re-authentication phase:

    + PANA-PAA-Discover.

    + PANA-Update-Request.

    + PANA-Start-Request after a PaC receives the first valid PANA-Auth-Request.

    + PANA-Termination-Request before the PaC receives the first successful PANA-Bind-Request.

  * In the access phase:

+ PANA-Start-Request as well as a non-duplicate PANA-Bind-Request.

+ PANA-PAA-Discover.

* In the termination phase:

  + PANA-PAA-Discover.

  + All requests but PANA-Termination-Request.

o The message payload contains a valid set of AVPs allowed for the message type and there is no missing AVP that needs to be included in the payload.

o Each AVP is decoded correctly.

o When a MAC AVP is included, the AVP value matches the MAC value computed against the received message.

o When a Device-Id AVP is included, the AVP is valid if the device identifier type contained in the AVP is supported (check performed by both the PaC and the PAA) and is the requested one (check performed by the PAA only) and the device identifier value contained in the AVP matches the value extracted from the lower-layer encapsulation header corresponding to the device identifier type contained in the AVP (check performed by the PAA only). Note that a Device-Id AVP carries the device identifier of the PaC in messages from the PaC to the PAA and the device identifier(s) of the EP(s) in messages from the PAA to the PaC.

o When an IP-Address AVP is received in a message, the AVP is valid if the IP address matches the source address in the IP header.

Invalid messages MUST be discarded in order to provide robustness against DoS attacks. In addition, an error notification message MAY be returned to the sender. See Section 5.10 for details.

5.6 Device ID Choice

The device identifier used in the context of PANA can be an IP address, a MAC address, or an identifier that is not carried in data packets but has local significance in identifying a connected device (e.g., circuit id, PPP interface id). The last type of identifiers are commonly used in point-to-point links where MAC addresses are not available and lower-layers are already physically or cryptographically secured.

It is assumed that the PAA knows the link type and the security mechanisms being provided or required on the access network (e.g., based on physical security, link-layer ciphers enabled before or after PANA, or IPsec). Based on that information, the PAA can decide what type of EP device id will be used when running PANA with the client. When IPsec-based security [I-D.ietf-pana-ipsec] is the choice of access control, the PAA SHOULD provide IP address(es) as EP(s)' device ID, and expect the PaC to provide its IP address in return. In case IPsec is not used, MAC addresses are used as device identifiers when available. If non-IPsec access control is enabled, and a MAC address is not available, device ID exchange does not occur within PANA. Instead, peers rely on lower-layers to provide locally-significant identifiers along with received PANA messages.

5.7 PaC Updating its IP Address

A PaC's IP address can change in certain situations. For example, the PANA framework [I-D.ietf-pana-framework] describes a case in which a PaC replaces a pre-PANA address (PRPA) with a post-PANA address (POPA), and the PaC and PAA create host routes to each other in order to maintain on-link communication based on the POPA. The PAA needs to be notified about the change of PaC address.

After the PaC has changed its address, it MUST send a PANA-Update-Request message to the PAA. The message MUST carry the new PaC address in an IP-Address AVP. If the address contained in the request is invalid, the PAA MUST send a PANA-Error message with a result code PANA_INVALID_IP_ADDRESS. Otherwise, the PAA MUST update the PANA session with the new PaC address and return a PANA-Update-Answer message. If there is an established PANA SA, both PANA-Update-Request and PANA-Update-Answer messages MUST be protected with a MAC AVP.

5.8 Session Lifetime

The authentication and authorization phase determines the PANA session lifetime when the network access authorization succeeds. The Session-Lifetime AVP MAY be optionally included in the PANA-Bind-Request message to inform the PaC about the valid lifetime of the PANA session. It MUST be ignored when included in other PANA messages.

The lifetime is a non-negotiable parameter that can be used by the PaC to manage PANA-related state. The PaC does not have to perform any actions when the lifetime expires, other than optionally purging local state. The PAA SHOULD initiate the PANA re-authentication phase before the current session lifetime expires.

The PaC and PAA MAY optionally rely on lower-layer indications to expedite the detection of a disconnected peer. Availability and reliability of such indications depend on the specific access technologies. A PANA peer can use the PANA-Ping exchange to verify the disconnection before taking an action.

The session lifetime parameter is not related to the transmission of PANA-Ping-Request messages. These messages can be used for asynchronously verifying the liveness of the peer. The decision to send a PANA-Ping-Request message is taken locally and does not require coordination between the peers.

When separate ISP and NAP authentication is performed, it is possible that different authorization lifetime values are associated with the two EAP authentication sessions. In this case, the smaller authorization lifetime value MUST be used for calculating the PANA Session-Lifetime value. As a result, both NAP and ISP authentication will be performed in the re-authentication phase.

5.9 Network Selection

The PANA discovery and handshake phase allows the PaC to learn identity of the NAP and a list of ISPs that are available through the NAP. The PaC can not only learn the ISPs but also convey the selected ISP explicitly during the handshake phase. The PAA is assumed to be pre-configured with the information of ISPs that are served by the NAP.

A PANA-Start-Request message sent from the PAA MAY contain zero or one NAP-Information AVP, and zero or more ISP-Information AVPs. The PaC MAY indicate its choice of ISP by including an ISP-Information AVP in the PANA-Start-Answer message. The PaC MAY convey its ISP even when there is no ISP-Information AVP contained in the PANA-Start-Request message. The PaC can do that when it is pre-configured with ISP information.

In the absence of an ISP explicitly selected and conveyed by the PaC, ISP selection is typically performed based on the client identifier (e.g., using the realm portion of an NAI carried in EAP method). A backend AAA protocol (e.g., RADIUS) will run between the AAA client on the PAA and a AAA server in the selected ISP domain.

The PANA-based ISP selection mechanism dictates the next-hop AAA proxy on the PAA. If the NAP requires all AAA traffic to go through its local AAA proxy, it may have to rely on a mechanism to relay the selected ISP information from PAA (AAA client) to the local AAA proxy. The local AAA proxy can forward the AAA traffic to the selected ISP domain upon processing. Further details, including how

the AAA client relays AAA routing information to the AAA proxy, are outside the scope of PANA.

An alternative ISP discovery mechanism is outlined in [I-D.adrangi-eap-network-discovery] which suggests advertising ISP information in-band with the ongoing EAP method execution. Deployments using the PANA's built-in ISP discovery mechanism need not use the other mechanism.

5.10 Error Handling

A PANA-Error-Request message MAY be sent by either the PaC or the PAA when a badly formed PANA message is received or in case of other errors. The receiver of this request MUST respond with a PANA-Error-Answer message. If the cause of this error message was a request message (e.g., PANA-PAA-Discover or *-Request), then the request MAY be retransmitted immediately without waiting for its retransmission timer to go off. If the cause of the error was a response message, the receiver of the PANA-Error-Request message SHOULD NOT resend the same response until it receives the next request.

Erroneous PANA messages may be exploited by adversaries to launch DoS attacks on the victims. Unless the PaC or PAA rate-limits the generated PANA-Error-Request messages it may be overburdened by having to respond to bogus messages. Limiting the number of error notifications sent to a given peer during a (configurable) period of time may be useful.

When an error message is sent unprotected (i.e., no MAC AVP) and the lower-layer is insecure, the error message is treated as an informational message. The receiver of such an error message MUST NOT change its state unless the error persists and the PANA session is not making any progress.

6. PANA Headers and Formats

This section defines message formats for PANA protocol.

6.1 IP and UDP Headers

The Hop Limit (or TTL) field of the IP header MUST be set to 255. When a PANA-PAA-Discover message is multicast, IP destination address of the message is set to a well-known link-local multicast address (TBD). A PANA-PAA-Discover message MAY be unicast in some cases as specified in Section 4.2. Any other PANA packet is unicast between the PaC and the PAA. The source and destination addresses SHOULD be set to the addresses on the interfaces from which the message will be sent and received, respectively.

When the PANA packet is sent in response to a request, the UDP source and destination ports of the response packet MUST be copied from the destination and source ports of the request packet, respectively. The destination port of an unsolicited PANA packet MUST be set to an assigned value (TBD), and the source port MUST be set to a value chosen by the sender.

The maximum PANA packet size is limited by the maximum UDP payload.

6.2 PANA Header

A summary of the PANA header format is shown below. The fields are transmitted in network byte order.

```
    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    |   Reserved    |        Message Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Flags              |         Message Type          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Sequence Number                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  AVPs ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Version

This Version field MUST be set to 1 to indicate PANA Version 1.

Reserved

This 8-bit field is reserved for future use, and MUST be set to zero, and ignored by the receiver.

Message Length

The Message Length field is three octets and indicates the length of the PANA message including the header fields.

Flags

The Flags field is two octets. The following bits are assigned:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R S N r r r r r r r r r r r r r|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

R(equest)

If set, the message is a request. If cleared, the message is an answer.

S(eparate)

When the S-flag is set in a PANA-Start-Request message it indicates that PAA is willing to offer separate NAP and ISP authentication. When the S-flag is set in a PANA-Start-Answer message it indicates that the PaC accepts on performing separate NAP and ISP authentication. The PaC may also respond with the S-flag not set which implies the PaC has chosen to authenticate with the ISP only. When the S-flag is set in a PANA-Auth-Request/Answer, PANA-FirstAuth-End-Request/Answer and PANA-Bind-Request/Answer messages it indicates that separate NAP and ISP authentication is being performed in the authentication and authorization phase. For other cases, S-flag MUST NOT be set.

N(AP authentication)

When the N-flag is set in a PANA-Auth-Request message, it indicates that the current EAP authentication is for NAP authentication. When the N-flag is unset in a PANA-Auth-Request message, it indicates that the current EAP authentication is for ISP authentication. The PaC MUST copy the value of the flag in its requests from the last received request of the PAA. The value of the flag on an answer MUST be copied from the request. The N-flag MUST NOT be set when S-flag is not set.

r(eserved)

These flag bits are reserved for future use, and MUST be set to zero, and ignored by the receiver.

Message Type

The Message Type field is two octets, and is used in order to communicate the message type with the message. The 16-bit address space is managed by IANA [ianaweb]. PANA uses its own address space for this field.

Sequence Number

The Sequence Number field contains a 32 bit value.

AVPs

AVPs are a method of encapsulating information relevant to the PANA message. See section Section 6.3 for more information on AVPs.

6.3 AVP Header

Each AVP of type OctetString MUST be padded to align on a 32-bit boundary, while other AVP types align naturally. A number of zero-valued bytes are added to the end of the AVP Data field till a word boundary is reached. The length of the padding is not reflected in the AVP Length field [RFC3588].

The fields in the AVP header are sent in network byte order. The format of the header is:

```
    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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           AVP Code            |           AVP Flags           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|          AVP Length           |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Vendor-Id (opt)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Data ...
+-+-+-+-+-+-+-+-+
```

AVP Code

The AVP Code, combined with the Vendor-Id field, identifies the attribute uniquely. AVP numbers are allocated by IANA [ianaweb]. PANA uses its own address space for this field although some of the AVP formats are borrowed from Diameter protocol [RFC3588].

AVP Flags

The AVP Flags field is two octets. The following bits are assigned:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V M r r r r r r r r r r r r r r|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

M(andatory)

The 'M' Bit, known as the Mandatory bit, indicates whether support of the AVP is required.
If an AVP with the 'M' bit set is received by the PaC or PAA and either the AVP or its value is unrecognized, the message MUST be rejected and the receiver MUST send a PANA-Error-Request message. If the AVP was unrecognized the PANA-Error-Request message result code MUST be PANA_AVP_UNSUPPORTED. If the AVP value was unrecognized the PANA-Error-Request message result code MUST be PANA_INVALID_AVP_DATA. In either case the PANA-Error-Request message MUST carry a Failed-AVP AVP containing the offending mandatory AVP.
AVPs with the 'M' bit cleared are informational only and a receiver that receives a message with such an AVP that is not supported, or whose value is not supported, MAY simply ignore the AVP.

V(endor)

The 'V' bit, known as the Vendor-Specific bit, indicates whether the optional Vendor-Id field is present in the AVP header. When set the AVP Code belongs to the specific vendor code address space.

r(eserved)

These flag bits are reserved for future use, and MUST be set to zero, and ignored by the receiver.

Unless otherwise noted, AVPs defined in this document will have the following default AVP Flags field settings: The 'M' bit MUST be set. The 'V' bit MUST NOT be set.

AVP Length

The AVP Length field is four octets, and indicates the number of octets in this AVP including the AVP Code, AVP Length, AVP Flags, and the AVP data.

Reserved

This two-octet field is reserved for future use, and MUST be set to zero, and ignored by the receiver.

Vendor-Id

The Vendor-Id field is present if the 'V' bit is set in the AVP Flags field. The optional four-octet Vendor-Id field contains the IANA assigned "SMI Network Management Private Enterprise Codes" [ianaweb] value, encoded in network byte order. Any vendor wishing to implement a vendor-specific PANA AVP MUST use their own Vendor-Id along with their privately managed AVP address space, guaranteeing that they will not collide with any other vendor's vendor-specific AVP(s), nor with future IETF applications.

Data

The Data field is zero or more octets and contains information specific to the Attribute. The format and length of the Data field is determined by the AVP Code and AVP Length fields.

7. PANA Messages, Message Specifications and AVPs

7.1 PANA Messages

Each Request/Answer message pair is assigned a message ID, and the sub-type (i.e., request or answer) is identified via the 'R' bit in the Message Flags field of the PANA header.

Every PANA message MUST contain a message ID in its header's Message-Id field, which is used to determine the action that is to be taken for a particular message. Figure 9 lists all PANA messages defined in this document:

Message-Name	Abbrev.	ID	PaC<->PAA	Ref.
PANA-PAA-Discover	PDI	1	-------->	7.2.1
PANA-Start-Request	PSR	2	<--------	7.2.2
PANA-Start-Answer	PSA	2	-------->	7.2.3
PANA-Auth-Request	PAR	3	<------->	7.2.4
PANA-Auth-Answer	PAN	3	<------->	7.2.5
PANA-Reauth-Request	PRAR	4	-------->	7.2.6
PANA-Reauth-Answer	PRAA	4	<--------	7.2.7
PANA-Bind-Request	PBR	5	<--------	7.2.8
PANA-Bind-Answer	PBA	5	-------->	7.2.9
PANA-Ping-Request	PPR	6	<------->	7.2.10
PANA-Ping-Answer	PPA	6	<------->	7.2.11
PANA-Termination-Request	PTR	7	<------->	7.2.12
PANA-Termination-Answer	PTA	7	<------->	7.2.13
PANA-Error-Request	PER	8	<------->	7.2.14
PANA-Error-Answer	PEA	8	<------->	7.2.15
PANA-FirstAuth-End-Request	PFER	9	<--------	7.2.16
PANA-FirstAuth-End-Answer	PFEA	9	-------->	7.2.17
PANA-Update-Request	PUR	10	<------->	7.2.18
PANA-Update-Answer	PUA	10	<------->	7.2.19

Figure 9: Table of PANA Messages

7.2 PANA Message ABNF Specification

Every PANA message defined MUST include a corresponding ABNF [RFC2234] specification, which is used to define the AVPs that MUST or MAY be present. The following format is used in the definition:

```
message-def      = Message-Name "::=" PANA-message

message-name     = PANA-name
```

```
PANA-name        = ALPHA *(ALPHA / DIGIT / "-")

PANA-message     = header  [ *fixed] [ *required] [ *optional]
                   [ *fixed]

header           = "< PANA-Header: " Message-Id
                   [r-bit] [s-bit] [n-bit] ">"

Message-Id       = 1*DIGIT
                   ; The message code assigned to the message

r-bit            = ", REQ"
                   ; If present, the 'R' bit in the Message
                   ; Flags is set, indicating that the message
                   ; is a request, as opposed to an answer.

s-bit            = ", SEP"
                   ; If present, the 'S' bit in the Message
                   ; Flags is set, indicating support for
                   ; separate NAP and ISP authentication.

n-bit            = ", NAP"
                   ; If present, the 'N' bit in the Message
                   ; Flags is set, indicating that current
                   ; EAP authentication is for NAP authentication.

fixed            = [qual] "<" avp-spec ">"
                   ; Defines the fixed position of an AVP

required         = [qual] "{" avp-spec "}"
                   ; The AVP MUST be present and can appear
                   ; anywhere in the message.

optional         = [qual] "[" avp-name "]"
                   ; The avp-name in the 'optional' rule cannot
                   ; evaluate to any AVP Name which is included
                   ; in a fixed or required rule.  The AVP can
                   ; appear anywhere in the message.

qual             = [min] "*" [max]
                   ; See ABNF conventions, RFC 2234 Section 6.6.
                   ; The absence of any qualifiers depends on whether
                   ; it precedes a fixed, required, or optional
                   ; rule.  If a fixed or required rule has no
                   ; qualifier, then exactly one such AVP MUST
                   ; be present.  If an optional rule has no
                   ; qualifier, then 0 or 1 such AVP may be
                   ; present.
```

```
                         ;
                         ; NOTE:  "[" and "]" have a different meaning
                         ; than in ABNF (see the optional rule, above).
                         ; These braces cannot be used to express
                         ; optional fixed rules (such as an optional
                         ; MAC at the end).  To do this, the convention
                         ; is '0*1fixed'.

   min                 = 1*DIGIT
                         ; The minimum number of times the element may
                         ; be present.  The default value is zero.

   max                 = 1*DIGIT
                         ; The maximum number of times the element may
                         ; be present.  The default value is infinity.  A
                         ; value of zero implies the AVP MUST NOT be
                         ; present.

   avp-spec            = PANA-name
                         ; The avp-spec has to be an AVP Name, defined
                         ; in the base or extended PANA protocol
                         ; specifications.

   avp-name            = avp-spec / "AVP"
                         ; The string "AVP" stands for *any* arbitrary
                         ; AVP Name, which does not conflict with the
                         ; required or fixed position AVPs defined in
                         ; the message definition.

   Example-Request ::= < "PANA-Header: 9999999, REQ >
                       < Session-Id >
                       { Result-Code }
                    *  [ AVP ]
                  0*1 < MAC >
```

7.2.1 PANA-PAA-Discover (PDI)

The PANA-PAA-Discover (PDI) message is used to discover the address of PAA(s). The sequence number in this message is always set to zero (0).

```
         PANA-PAA-Discover ::= < PANA-Header: 1 >
                       [ Notification ]
                    *  [ AVP ]
```

7.2.2 PANA-Start-Request (PSR)

The PANA-Start-Request (PSR) message is sent by the PAA to the PaC to advertise availability of the PAA and start PANA authentication. The PAA sets the sequence number to an initial random value.

```
PANA-Start-Request ::= < PANA-Header: 2, REQ [, SEP] >
                          { Nonce }
                          [ Cookie ]
                          [ EAP-Payload ]
                          [ NAP-Information ]
                       *  [ ISP-Information ]
                          [ Protection-Capability]
                          [ PPAC ]
                          [ Notification ]
                       *  [ AVP ]
```

7.2.3 PANA-Start-Answer (PSA)

The PANA-Start-Answer (PSA) message is sent by the PaC to the PAA in response to a PANA-Start-Request message. This message completes the handshake to start PANA authentication.

```
PANA-Start-Answer ::= < PANA-Header: 2 [, SEP] >
                          { Nonce }
                          [ Cookie ]
                          [ EAP-Payload ]
                          [ ISP-Information ]
                          [ Notification ]
                       *  [ AVP ]
```

7.2.4 PANA-Auth-Request (PAR)

The PANA-Auth-Request (PAR) message is either sent by the PAA or the PaC. Its main task is to carry an EAP-Payload AVP.

```
PANA-Auth-Request ::= < PANA-Header: 3, REQ [, SEP] [, NAP] >
                          < Session-Id >
                          < EAP-Payload >
                          [ Notification ]
                       *  [ AVP ]
                     0*1  < MAC >
```

7.2.5 PANA-Auth-Answer (PAN)

THe PANA-Auth-Answer (PAN) message is sent by either the PaC or the PAA in response to a PANA-Auth-Request message. It MAY carry an EAP-Payload AVP.

```
PANA-Auth-Answer ::= < PANA-Header: 3 [, SEP] [, NAP] >
                     < Session-Id >
                     [ EAP-Payload ]
                     [ Notification ]
                  *  [ AVP ]
                 0*1 < MAC >
```

7.2.6 PANA-Reauth-Request (PRAR)

The PANA-Reauth-Request (PRAR) message is sent by the PaC to the PAA to re-initiate EAP authentication.

```
PANA-Reauth-Request ::= < PANA-Header: 4, REQ >
                     < Session-Id >
                     [ Notification ]
                  *  [ AVP ]
                 0*1 < MAC >
```

7.2.7 PANA-Reauth-Answer (PRAA)

The PANA-Reauth-Answer (PRAA) message is sent by the PAA to the PaC in response to a PANA-Reauth-Request message.

```
PANA-Reauth-Answer ::= < PANA-Header: 4 >
                     < Session-Id >
                     [ Notification ]
                  *  [ AVP ]
                 0*1 < MAC >
```

7.2.8 PANA-Bind-Request (PBR)

The PANA-Bind-Request (PBR) message is sent by the PAA to the PaC to deliver the result of PANA authentication.

```
PANA-Bind-Request ::= < PANA-Header: 5, REQ [, SEP] [, NAP] >
                     < Session-Id >
                     { Result-Code }
                     { PPAC }
                     { IP-Address }
```

```
                      [ EAP-Payload ]
                      [ Session-Lifetime ]
                      [ Protection-Capability ]
                      [ Key-Id ]
                   *  [ Device-Id ]
                      [ Notification ]
                   *  [ AVP ]
                  0*1 < MAC >
```

7.2.9 PANA-Bind-Answer (PBA)

The PANA-Bind-Answer (PBA) message is sent by the PaC to the PAA in response to a PANA-Bind-Request message.

```
PANA-Bind-Answer ::= < PANA-Header: 5 [,SEP] [, NAP] >
              < Session-Id >
              [ PPAC ]
              [ Device-Id ]
              [ Key-Id ]
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.10 PANA-Ping-Request (PPR)

The PANA-Ping-Request (PPR) message is either sent by the PaC or the PAA for performing liveness test.

```
PANA-Ping-Request ::= < PANA-Header: 6, REQ >
              < Session-Id >
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.11 PANA-Ping-Answer (PPA)

The PANA-Ping-Answer (PPA) message is sent in response to a PANA-Ping-Request.

```
PANA-Ping-Answer ::= < PANA-Header: 6 >
              < Session-Id >
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.12 PANA-Termination-Request (PTR)

The PANA-Termination-Request (PTR) message is sent either by the PaC or the PAA to terminate a PANA session.

```
PANA-Termination-Request ::= < PANA-Header: 7, REQ >
              < Session-Id >
              < Termination-Cause >
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.13 PANA-Termination-Answer (PTA)

The PANA-Termination-Answer (PTA) message is sent either by the PaC or the PAA in response to PANA-Termination-Request.

```
PANA-Termination-Answer ::= < PANA-Header: 7 >
              < Session-Id >
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.14 PANA-Error-Request (PER)

The PANA-Error-Request (PER) message is sent either by the PaC or the PAA to report an error with the last received PANA message.

```
PANA-Error-Request ::= < PANA-Header: 8, REQ >
               < Session-Id >
               < Result-Code >
            *  [ Failed-AVP ]
               [ Notification ]
            *  [ AVP ]
           0*1 < MAC >
```

7.2.15 PANA-Error-Answer (PEA)

The PANA-Error-Answer (PEA) message is sent in response to a PANA-Error-Request.

```
PANA-Error-Answer ::= < PANA-Header: 8 >
               < Session-Id >
               [ Notification ]
            *  [ AVP ]
```

```
                    0*1 < MAC >
```

7.2.16 PANA-FirstAuth-End-Request (PFER)

The PANA-FirstAuth-End-Request (PFER) message is sent by the PAA to the PaC to signal the result of the first EAP authentication method when separate NAP and ISP authentication is performed.

```
PANA-FirstAuth-End-Request ::= < PANA-Header: 9, REQ [, SEP] [, NAP] >
              < Session-Id >
              { Result-Code }
              [ EAP-Payload ]
              [ Key-Id ]
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.17 PANA-FirstAuth-End-Answer (PFEA)

The PANA-FirstAuth-End-Answer (PFEA) message is sent by the PaC to the PAA in response to a PANA-FirstAuth-End-Request message.

```
PANA-FirstAuth-End-Answer ::= < PANA-Header: 9, REQ [, SEP] [, NAP] >
              < Session-Id >
              [ Key-Id ]
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.18 PANA-Update-Request (PUR)

The PANA-Update-Request (PUR) message is sent either by the PaC or the PAA to deliver attribute updates and notifications. In the scope of this specification only the PaC IP address attribute can be updated via this mechanism. An IP-Address AVP can only be included in the PUR messages sent by the PaC. The PUR message can be used to deliver just a notification as well.

```
PANA-Update-Request ::= < PANA-Header: 10, REQ >
              < Session-Id >
              [ IP-Address ]
              [ Notification ]
           *  [ AVP ]
          0*1 < MAC >
```

7.2.19 PANA-Update-Answer (PUA)

The PANA-Update-Answer (PUA) message is sent by the PAA to the PaC in response to a PANA-Update-Request.

```
PANA-Update-Answer ::= < PANA-Header: 10 >
                          < Session-Id >
                          [ Notification ]
                       *  [ AVP ]
                     0*1 < MAC >
```

7.3 AVPs in PANA

PANA defines several AVPs that are specific to the protocol. A number of others AVPs are reused. These are specified in other documents such as [RFC3588].

The following tables lists the AVPs used in this document, and specifies in which PANA messages they MAY, or MAY NOT be present.

The table uses the following symbols:

0 The AVP MUST NOT be present in the message.

0+ Zero or more instances of the AVP MAY be present in the message.

0-1 Zero or one instance of the AVP MAY be present in the message. It is considered an error if there are more than one instance of the AVP.

1 One instance of the AVP MUST be present in the message.

1+ At least one instance of the AVP MUST be present in the message.

Attribute Name	Message Type										
	PDI	PSR	PSA	PAR	PAN	PRAR	PRAA	PBR	PBA	PPR	PPA
Cookie	0	0-1	0-1	0	0	0	0	0	0	0	0
Device-Id	0	0	0	0	0	0	0	0+	0-1	0	0
EAP-Payload	0	0-1	0-1	1	0-1	0	0	0-1	0	0	0
Failed-AVP	0	0	0	0	0	0	0	0	0	0	0
IP-Address	0	0	0	0	0	0	0	1	0	0	0
ISP-Information	0	0+	0-1	0	0	0	0	0	0	0	0
Key-Id	0	0	0	0	0	0	0	0-1	0-1	0	0
MAC	0	0	0	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1
NAP-Information	0	0-1	0	0	0	0	0	0	0	0	0
Nonce	0	1	1	0	0	0	0	0	0	0	0
Notification	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1
PPAC	0	0-1	0	0	0	0	0	1	0-1	0	0
Protection-Cap.	0	0-1	0	0	0	0	0	0-1	0	0	0
Result-Code	0	0	0	0	0	0	0	1	0	0	0
Session-Id	0	0	0	1	1	1	1	1	1	1	1
Session-Lifetime	0	0	0	0	0	0	0	0-1	0	0	0
Termination-Cause	0	0	0	0	0	0	0	0	0	0	0

Figure 10: AVP Occurrence Table (1/2)

	Message Type							
Attribute Name	PTR	PTA	PER	PEA	PFER	PFEA	PUR	PUA
Cookie	0	0	0	0	0	0	0	0
Device-Id	0	0	0	0	0	0	0	0
EAP-Payload	0	0	0	0	0-1	0	0	0
Failed-AVP	0	0	0+	0	0	0	0	0
IP-Address	0	0	0	0	0	0	0-1	0
ISP-Information	0	0	0	0	0	0	0	0
Key-Id	0	0	0	0	0-1	0-1	0	0
MAC	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1
NAP-Information	0	0	0	0	0	0	0	0
Nonce	0	0	0	0	0	0	0	0
Notification	0-1	0-1	0-1	0-1	0-1	0-1	0-1	0-1
PPAC	0	0	0	0	0	0	0	0
Protection-Cap.	0	0	0	0	0	0	0	0
Result-Code	0	0	1	0	1	0	0	0
Session-Id	1	1	1	1	1	1	1	1
Session-Lifetime	0	0	0	0	0	0	0	0
Termination-Cause	1	0	0	0	0	0	0	0

Figure 11: AVP Occurrence Table (2/2)

7.3.1 Cookie AVP

The Cookie AVP (AVP Code 1) is used for carrying a random value generated by the PAA. The AVP data is of type OctetString. The random value is referred to as a cookie and used for making PAA discovery robust against blind resource consumption DoS attacks. The exact algorithms and syntax used by the PAA to generate a cookie does not affect interoperability and not specified in this document. An example cookie generation algorithm is shown in Section 4.2.

7.3.2 Device-Id AVP

The Device-Id AVP (AVP Code 2) is used for carrying device identifiers of PaC and EP(s). The AVP data is of Address type [RFC3588]. IPv4 and IPv6 addresses are encoded as specified in [RFC3588]. The content and format of data (including byte and bit ordering) for link-layer addresses is expected to be specified in specific documents that describe how IP operates over different link-layers. For instance, [RFC2464]. Address families other than that are defined for link-layer or IP addresses MUST NOT be used for

this AVP.

7.3.3 EAP-Payload AVP

The EAP-Payload AVP (AVP Code 3) is used for encapsulating the actual EAP message that is being exchanged between the EAP peer and the EAP authenticator. The AVP data is of type OctetString.

7.3.4 Failed-AVP AVP

The Failed-AVP AVP (AVP Code 4) provides debugging information in cases where a request is rejected or not fully processed due to erroneous information in a specific AVP. The AVP data is of type Grouped. The format of the Failed-AVP AVP is defined in [RFC3588].

7.3.5 IP-Address AVP

The IP-Address AVP (AVP Code 5) contains an IP address of the PaC or PAA. When it is sent by the PaC, it is used to convey the new IP address of the PaC to the PAA when the PaC reconfigures its IP address after the successful PANA authentication. This AVP is not used if the PaC's IP address used during the authentication and authorization phase is still valid. It is sent by the PAA in PANA-Bind-Request to bind the IP address of the PAA to the PANA session. The payload format of the IP-Address AVP is the same as that of the Device-Id AVP (see See Section 7.3.2). Address families for IPv4 or IPv6 MUST be used for this AVP.

7.3.6 ISP-Information AVP

The ISP-Information AVP (AVP Code 6) contains zero or one Provider-Identifier AVP which carries the identifier of the ISP and one Provider-Name AVP which carries the name of the ISP. The AVP data is of type Grouped, and it has the following ABNF grammar:

```
ISP-Information ::= < AVP Header: 6 >
           0*1 { Provider-Identifier }
               { Provider-Name }
             *  [ AVP ]
```

7.3.7 Key-Id AVP

The Key-Id AVP (AVP Code 7) is of type Integer32, and contains an AAA-Key identifier. The AAA-Key identifier is assigned by PAA and MUST be unique within the PANA session.

7.3.8 MAC AVP

The MAC (Message Authentication Code) AVP is used to integrity protect PANA messages. The first octet of the this AVP (AVP Code 8) data contains the MAC algorithm type. Rest of the AVP data payload contains the MAC encoded in network byte order. The 8-bit Algorithm name space is managed by IANA [ianaweb]. The AVP length varies depending on the used algorithm.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Algorithm   |           MAC...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Algorithm

      1   HMAC-SHA1 (20 bytes)

MAC

   The Message Authentication Code is encoded in network byte order.

7.3.9 NAP-Information AVP

The NAP-Information AVP (AVP Code 9) contains zero or one Provider-Identifier AVP which carries the identifier of the NAP and one Provider-Name AVP which carries the name of the NAP. The AVP data is of type Grouped, and it has the following ABNF grammar:

```
      NAP-Information ::= < AVP Header: 9 >
                 0*1 { Provider-Identifier }
                     { Provider-Name }
                  *  [ AVP ]
```

7.3.10 Nonce AVP

The Nonce AVP (AVP Code 10) carries a randomly chosen value that is used in cryptographic key computations. The AVP data is of type OctetString and it contains a randomly generated value in opaque format. The data length MUST be between 8 and 256 bytes inclusive.

7.3.11 Notification AVP

The Notification AVP (AVP Code 11) is optionally used to convey a displayable message sent by either the PaC or the PAA. It can be included in any message, whether it is a request or answer. In case

a notification needs to be sent but there is no outgoing PANA message to deliver this AVP, a PANA-Update-Request that only carries a Notification AVP SHOULD be generated.

Receipt this AVP does not change PANA state.

AVP data is of type OctetString and it contains UTF-8 encoded ISO 10646 characters [RFC2279]. The length of the displayable message is determined by the AVP Length field. The message MUST NOT be null terminated.

7.3.12 Post-PANA-Address-Configuration (PPAC) AVP

The PPAC AVP (AVP Code 12) is used for conveying the available types of post-PANA IP address configuration mechanisms when sent by the PAA, and the chosen one when sent by the PaC. Each possible mechanisms is represented by a flag. At least one or more of the flags MUST be set when sent by the PAA, and exactly one flag MUST be set when sent by the PaC. The AVP data is of type Unsigned32.

The format of the AVP data is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|N|D|A|T|I|                   Reserved                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PPAC Flags

N (No configuration)

The PaC does not have to (if sent by PAA) or will not (if sent by PaC) configure a new IP address after PANA.

D (DHCP)

The PaC can (if sent by PAA) or will (if sent by PaC) use DHCP [RFC2131][RFC3315] to configure a new IP address after PANA.

A (stateless autoconfiguration)

The PaC can/will use stateless IPv6 address autoconfiguration [RFC2462] to configure a new IP address after PANA.

T (DHCP with IPsec tunnel mode)

The PaC can/will use [RFC3456] to configure a new IP address after PANA.

I (IKEv2)

The PaC can/will use [I-D.ietf-ipsec-ikev2] to configure a new IP address after PANA.

Reserved

These flag bits are reserved for future use, and MUST be set to zero, and ignored by the receiver.

Unless the N-flag is set, the PaC MUST configure a new IP address using one of the methods indicated by the other flags. Refer to [I-D.ietf-pana-framework] for a detailed discussion on when these methods can be used.

7.3.13 Protection-Capability AVP

The Protection-Capability AVP (AVP Code 13) indicates the cryptographic data protection capability supported and required by the EPs. The AVP data is of type Unsigned32. Below is a list of valid data values and associated protection capabilities:

```
0          L2_PROTECTION
1          IPSEC_PROTECTION
```

7.3.14 Provider-Identifier AVP

The Provider-Identifier AVP (AVP Code 14) is of type Unsigned32, and contains an IANA assigned "SMI Network Management Private Enterprise Codes" [ianaweb] value, encoded in network byte order.

7.3.15 Provider-Name AVP

The Provider-Name AVP (AVP Code 15) is of type UTF8String, and contains the UTF8-encoded name of the provider.

7.3.16 Result-Code AVP

The Result-Code AVP (AVP Code 16) is of type Unsigned32 and indicates whether an EAP authentication was completed successfully or whether an error occurred. Here are Result-Code AVP values taken from [RFC3588] and adapted for PANA.

7.3.16.1 Authentication Results Codes

These result code values inform the PaC about the authentication and authorization result. The authentication result and authorization result can be different as described below, but only one result is returned to the PaC. These codes are used with PANA-Bind-Request and PANA-FirstAuth-End-Request messages.

PANA_SUCCESS 2001

Both authentication and authorization processes are successful.

PANA_AUTHENTICATION_REJECTED 4001

Authentication has failed. When this error is returned, it is assumed that authorization is automatically failed.

PANA_AUTHORIZATION_REJECTED 5003

The authorization process has failed. This error could occur when authorization is rejected by a AAA server or rejected locally by a PAA, even if the authentication procedure has succeeded.

7.3.16.2 Protocol Error Result Codes

These codes are used with PANA-Error-Request messages. Unless stated otherwise, they can be generated by both the PaC and the PAA.

PANA_MESSAGE_UNSUPPORTED 3001

Message type not recognized or supported.

PANA_UNABLE_TO_DELIVER 3002

The PAA was unable to deliver the EAP payload to the authentication server. Only the PAA can generate this code.

PANA_INVALID_HDR_BITS 3008

A message was received whose bits in the PANA header were either set to an invalid combination, or to a value that is inconsistent with the message type definition.

PANA_INVALID_AVP_FLAGS 3009

A message was received that included an AVP whose flag bits are set to an unrecognized value, or that is inconsistent with the AVP's definition.

PANA_AVP_UNSUPPORTED 5001

The received message contained an AVP that is not recognized or supported and was marked with the Mandatory bit. A PANA message with this error MUST contain one or more Failed-AVP AVP containing the AVPs that caused the failure.

PANA_UNKNOWN_SESSION_ID 5002

The message contained an unknown Session-Id. A PANA message indicating this error MUST include the unknown Session-Id AVP within a Failed-AVP AVP.

PANA_INVALID_AVP_DATA 5004

The message contained an AVP with an invalid value in its data portion. A PANA message indicating this error MUST include the offending AVPs within a Failed-AVP AVP.

PANA_MISSING_AVP 5005

The message did not contain an AVP that is required by the message type definition. If this value is sent in the Result-Code AVP, a Failed-AVP AVP SHOULD be included in the message. The Failed-AVP AVP MUST contain an example of the missing AVP complete with the Vendor-Id if applicable. The value field of the missing AVP should be of correct minimum length and contain zeroes.

PANA_RESOURCES_EXCEEDED 5006

A message was received that cannot be authorized because the client has already expended allowed resources. An example of this error condition is a client that is restricted to one PANA session and attempts to establish a second session. Only the PAA can generate this code.

PANA_CONTRADICTING_AVPS 5007

The PAA has detected AVPs in the message that contradicted each other, and is not willing to provide service to the client. One or more Failed-AVP AVPs MUST be present, containing the AVPs that contradicted each other. Only the PAA can generate this code.

PANA_AVP_NOT_ALLOWED 5008

A message was received with an AVP that MUST NOT be present. The Failed-AVP AVP MUST be included and contain a copy of the offending AVP.

PANA_AVP_OCCURS_TOO_MANY_TIMES 5009

A message was received that included an AVP that appeared more often than permitted in the message definition. The Failed-AVP AVP MUST be included and contain a copy of the first instance of the offending AVP that exceeded the maximum number of occurrences.

PANA_UNSUPPORTED_VERSION 5011

This error is returned when a message was received, whose version number is unsupported.

PANA_UNABLE_TO_COMPLY 5012

This error is returned when a request is rejected for unspecified reasons. For example, when an EAP authentication fails at an EAP pass-through authenticator without passing an EAP Failure message to the PAA, a Result-Code AVP with this error code is carried in the PANA-Error-Request message.

PANA_INVALID_AVP_LENGTH 5014

The message contained an AVP with an invalid length. The PANA-Error-Request message indicating this error MUST include the offending AVPs within a Failed-AVP AVP.

PANA_INVALID_MESSAGE_LENGTH 5015

This error is returned when a message is received with an invalid message length.

PANA_PROTECTION_CAPABILITY_UNSUPPORTED 5016

This error is returned when the PaC receives a PANA-Bind-Request message with a Protection-Capability AVP and a valid MAC AVP but does not support the protection capability specified in the Protection-Capability AVP. Only the PaC can generate this code.

PANA_PPAC_CAPABILITY_UNSUPPORTED 5017

This error is returned when there is no match between the list of PPAC methods offered by the PAA and the ones available on the PaC. Only the PaC can generate this code.

PANA_INVALID_IP_ADDRESS 5018

This error is returned in a PANA-Error-Request message when the IP-Address AVP in the received PANA-Update-Request message is invalid (e.g., a non-unicast address). Only the PAA can generate this code.

7.3.17 Session-Id AVP

All messages pertaining to a specific PANA session MUST include a Session-Id AVP (AVP Code 17) which carries a PAA-assigned fixed session identifier value throughout the lifetime of a session. When present, the Session-Id AVP SHOULD appear immediately following the PANA header.

The Session-Id MUST be globally and eternally unique, as it is meant to identify a PANA session without reference to any other information, and may be needed to correlate historical authentication information with accounting information. The PANA Session-Id AVP has the same format as the Diameter Session-Id AVP [RFC3588].

7.3.18 Session-Lifetime AVP

The Session-Lifetime AVP (AVP Code 18) contains the number of seconds remaining before the current session is considered expired. The AVP data is of type Unsigned32.

7.3.19 Termination-Cause AVP

The Termination-Cause AVP (AVP Code 19) is used for indicating the reason why a session is terminated by the requester. The AVP data is of type Enumerated. The following Termination-Cause data values are used with PANA.

LOGOUT 1 (PaC -> PAA)

The client initiated a disconnect

ADMINISTRATIVE 4 (PAA -> PaC)

The client was not granted access, or was disconnected, due to administrative reasons.

SESSION_TIMEOUT 8 (PAA -> PaC)

The session has timed out, and service has been terminated.

8. Retransmission Timers

The PANA protocol provides retransmissions for the PANA-PAA-Discover message and all request messages, with the exception that the PANA-Start-Answer message is retransmitted instead of the PANA-Start-Request message in stateless PAA discovery.

PANA retransmission timers are based on the model used in DHCPv6 [RFC3315]. Variables used here are also borrowed from this specification. PANA is a request response like protocol. The message exchange terminates when either the request sender successfully receives the appropriate answer, or when the message exchange is considered to have failed according to the retransmission mechanism described below.

The retransmission behavior is controlled and described by the following variables:

RT Retransmission timeout

IRT Initial retransmission time

MRC Maximum retransmission count

MRT Maximum retransmission time

MRD Maximum retransmission duration

RAND Randomization factor

With each message transmission or retransmission, the sender sets RT according to the rules given below. If RT expires before the message exchange terminates, the sender recomputes RT and retransmits the message.

Each of the computations of a new RT include a randomization factor (RAND), which is a random number chosen with a uniform distribution between -0.1 and +0.1. The randomization factor is included to minimize synchronization of messages.

The algorithm for choosing a random number does not need to be cryptographically sound. The algorithm SHOULD produce a different sequence of random numbers from each invocation.

RT for the first message transmission is based on IRT:

RT = IRT + RAND*IRT

RT for each subsequent message transmission is based on the previous value of RT:

```
RT = 2*RTprev + RAND*RTprev
```

MRT specifies an upper bound on the value of RT (disregarding the randomization added by the use of RAND). If MRT has a value of 0, there is no upper limit on the value of RT. Otherwise:

```
if (RT > MRT)
   RT = MRT + RAND*MRT
```

MRC specifies an upper bound on the number of times a sender may retransmit a message. Unless MRC is zero, the message exchange fails once the sender has transmitted the message MRC times.

MRD specifies an upper bound on the length of time a sender may retransmit a message. Unless MRD is zero, the message exchange fails once MRD seconds have elapsed since the client first transmitted the message.

If both MRC and MRD are non-zero, the message exchange fails whenever either of the conditions specified in the previous two paragraphs are met.

If both MRC and MRD are zero, the client continues to transmit the message until it receives a response.

8.1 Transmission and Retransmission Parameters

This section presents a table of values used to describe the message retransmission behavior of PANA requests and answers that are retransmitted (REQ_*) and PANA-PAA-Discover message (PDI_*). The table shows default values.

Parameter	Default	Description
PDI_IRT	1 sec	Initial PDI timeout.
PDI_MRT	120 secs	Max PDI timeout value.
PDI_MRC	0	Configurable.
PDI_MRD	0	Configurable.
REQ_IRT	1 sec	Initial Request timeout.
REQ_MRT	30 secs	Max Request timeout value.
REQ_MRC	10	Max Request retry attempts.
REQ_MRD	0	Configurable.

So for example the first RT for the PBR message is calculated using

REQ_IRT as the IRT:

RT = REQ_IRT + RAND*REQ_IRT

9. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the PANA protocol, in accordance with BCP 26 [IANA]. The following policies are used here with the meanings defined in BCP 26: "Private Use", "First Come First Served", "Expert Review", "Specification Required", "IETF Consensus", "Standards Action".

This section explains the criteria to be used by the IANA for assignment of numbers within namespaces defined within this document.

For registration requests where a Designated Expert should be consulted, the responsible IESG area director should appoint the Designated Expert. For Designated Expert with Specification Required, the request is posted to the PANA WG mailing list (or, if it has been disbanded, a successor designated by the Area Director) for comment and review, and MUST include a pointer to a public specification. Before a period of 30 days has passed, the Designated Expert will either approve or deny the registration request and publish a notice of the decision to the PANA WG mailing list or its successor. A denial notice must be justified by an explanation and, in the cases where it is possible, concrete suggestions on how the request can be modified so as to become acceptable.

9.1 PANA UDP Port Number

PANA uses one well-known UDP port number (Section 5.1, Section 4.2 and Section 6.1), which needs to be assigned by the IANA.

9.2 PANA Multicast Address

PANA uses one well-known IPv4 multicast address for which the scope is limited to be link-local by setting the TTL field to 255, and one well-known IPv6 link-local scoped multicast address (Section 4.2 and Section 6.1), which need to be assigned by the IANA.

9.3 PANA Header

As defined in Section 6.2, the PANA header contains two fields that requires IANA namespace management; the Message Type and Flags field.

9.3.1 Message Type

The Message Type namespace is used to identify PANA messages. Values 0-65,533 are for permanent, standard message types, allocated by IETF Consensus [IANA]. This document defines the Message Types 1-10. See Section 7.2.1 through Section 7.2.19 for the assignment of the

namespace in this specification.

The values 65,534 and 65,535 (hexadecimal values 0xfffe - 0xffff) are reserved for experimental messages. As these codes are only for experimental and testing purposes, no guarantee is made for interoperability between the communicating PaC and PAA using experimental commands, as outlined in [IANA-EXP].

9.3.2 Flags

There are 16 bits in the Flags field of the PANA header. This document assigns bit 0 ('R'equest), bit 1 ('S'eparate) and bit 2 ('N'AP Authentication). The remaining bits MUST only be assigned via a Standards Action [IANA].

9.4 AVP Header

As defined in Section 6.3, the AVP header contains three fields that requires IANA namespace management; the AVP Code, AVP Flags and Vendor-Id fields where only the AVP Code and AVP Flags create new namespaces.

9.4.1 AVP Code

The AVP Code namespace is used to identify attributes. There are multiple namespaces. Vendors can have their own AVP Codes namespace which will be identified by their Vendor-ID (also known as Enterprise-Number) and they control the assignments of their vendor-specific AVP codes within their own namespace. The absence of a Vendor-ID or a Vendor-ID value of zero (0) identifies the IETF IANA controlled AVP Codes namespace. The AVP Codes and sometimes also possible values in an AVP are controlled and maintained by IANA.

AVP Code 0 is not used. This document defines the AVP Codes 1-19. See Section 7.3.8 through Section 7.3.5 for the assignment of the namespace in this specification.

AVPs may be allocated following Designated Expert with Specification Required [IANA]. Release of blocks of AVPs (more than 3 at a time for a given purpose) should require IETF Consensus.

Note that PANA defines a mechanism for Vendor-Specific AVPs, where the Vendor-Id field in the AVP header is set to a non-zero value. Vendor-Specific AVPs codes are for Private Use and should be encouraged instead of allocation of global attribute types, for functions specific only to one vendor's implementation of PANA, where no interoperability is deemed useful. Where a Vendor-Specific AVP is implemented by more than one vendor, allocation of global AVPs should

be encouraged instead.

9.4.2 Flags

There are 16 bits in the AVP Flags field of the AVP header, defined in Section 6.3. This document assigns bit 0 ('V'endor Specific) and bit 1 ('M'andatory). The remaining bits should only be assigned via a Standards Action .

9.5 AVP Values

Certain AVPs in PANA define a list of values with various meanings. For attributes other than those specified in this section, adding additional values to the list can be done on a First Come, First Served basis by IANA [IANA].

9.5.1 Algorithm Values of MAC AVP

As defined in Section 7.3.8, the Algorithm field of MAC AVP (AVP Code 8) defines the value of 1 (one) for HMAC-SHA1.

All remaining values are available for assignment via IETF Consensus [IANA].

9.5.2 Post-PANA-Address-Configuration AVP Values

As defined in Section 7.3.12, the Post-PANA-Address-Configuration AVP (AVP Code 12) defines the bits 0 ('N': no configuration), 1 ('D': DHCP), 2 ('A' stateless autoconfiguration), 3 ('T': DHCP with IPsec tunnel mode) and 4 ('I': IKEv2).

All remaining values are available for assignment via a Standards Action [IANA].

9.5.3 Protection-Capability AVP Values

As defined in Section 7.3.13, the Protection-Capability AVP (AVP Code 13) defines the values 0 and 1.

All remaining values are available for assignment via a Standards Action [IANA].

9.5.4 Result-Code AVP Values

As defined in Section 7.3.16.1 and Section 7.3.16.2 the Result-Code AVP (AVP Code 16) defines the values 2001, 3001-3002, 3008-3009, 4001, 5001-5009 and 5011-5019.

All remaining values are available for assignment via IETF Consensus [IANA].

9.5.5 Termination-Cause AVP Values

As defined in Section 7.3.19, the Termination-Cause AVP (AVP Code 19) defines the values 1, 4 and 8.

All remaining values are available for assignment via IETF Consensus [IANA].

10. Security Considerations

The PANA protocol defines a UDP-based EAP encapsulation that runs between two IP-enabled nodes on the same IP link. Various security threats that are relevant to a protocol of this nature are outlined in [I-D.ietf-pana-threats-eval]. Security considerations stemming from the use of EAP and EAP methods are discussed in [RFC3748]. This section provides a discussion on the security-related issues that are related to PANA framework and protocol design.

An important element in assessing security of PANA design and deployment in a network is the presence of lower-layer (physical and link-layer) security. In the context of this document, lower-layers are said to be secure if they can prevent eavesdropping and spoofing of packets. Examples of such networks are physically-secured DSL networks and 3GPP2 networks with crytographically-secured cdma2000 link-layer. In these examples, the lower-layer security is enabled even before running the first PANA-based authentication. In the absence of such a pre-established secure channel, one needs to be created in conjunction with PANA using a link-layer or network-layer cryptographic mechanism (e.g., IPsec).

10.1 General Security Measures

PANA provides multiple mechanisms to secure a PANA session.

Since the PaC and PAA are on the same IP link, a simple TTL check on the received PANA messages prevents off-link attacks.

PANA messages carry sequence numbers, which are monotonically incremented by 1 with every new request message. These numbers are randomly initialized at the beginning of the session, and verified against expected numbers upon receipt. A message whose sequence number is different than the expected one is silently discarded. In addition to accomplishing orderly delivery of EAP messages and duplicate elimination, this scheme also helps prevent an adversary spoof messages to disturb ongoing PANA and EAP sessions unless it can also eavesdrop to synchronize on the expected sequence number. Furthermore, impact of replay attacks is reduced as any stale message (i.e., a request or answer with an unexpected sequence number) and any duplicate answer are immediately discarded, and a duplicate request can trigger transmission of the cached answer (i.e., no need to process the request and generate a new answer).

The PANA framework defines EP which is ideally located on a network device that can filter traffic from the PaCs before the traffic enters the Internet/intranet. A set of filters can be used to discard unauthorized packets, such as a PANA-Start-Request message

that is received from the segment of the access network where only the PaCs are supposed to be connected.

The protocol also provides authentication and integrity protection to PANA messages when the used EAP method can generate cryptographic session keys. A PANA SA is generated based on the AAA-Key exported by the EAP method. This SA is used for generating per-packet MAC to protect the PANA header and payload (including the complete EAP message).

The cryptographic protection prevents an adversary from acting as a man-in-the-middle, injecting messages, replaying messages and modifying the content of the exchanged messages. Any packet that fails to pass the MAC verification is silently discarded. The earliest this protection can be enabled is when the very first PANA-Bind-Request or PANA-FirstAuth-End-Request message that signals a successful authentication is generated. Starting with these messages, any subsequent PANA message until the session gets torn down can be cryptographically protected.

The PANA SA enables authenticated and integrity protected exchange of the device ID information between the PaC and PAA. This ensures there were no man-in-the-middle during the PANA authentication.

The lifetime of the PANA SA is set to PANA session lifetime which is bounded by the lifetime granted by the authentication server. An implementation MAY add a tolerance period to that value. Unless the PANA session is extended by executing another EAP authentication, the PANA SA is removed when the current session expires.

The ability to use cryptographic protection within PANA is determined by the used EAP method, which is generally dictated by the deployment environment. Insecure lower-layers necessitate use of key-generating EAP methods. In networks where lower-layers are already secured, cryptographic protection of PANA messages is not necessary.

10.2 Discovery

The discovery and handshake phase is vulnerable to spoofing attacks as these messages are not authenticated and integrity protected. In order to prevent very basic denial-of service attacks an adversary should not be able to cause state creation by sending discovery messages to the PAA. This protection is achieved by using a cookie-based scheme (similar to [RFC2522] which allows the responder (PAA) to be stateless in the first round of message exchange. A return-routability test does not provide additional protection as PANA traffic is not routed but simply forwarded on-link. It is difficult to prevent this threat entirely.

In networks where lower-layers are not secured prior to running PANA, the capability discovery enabled through inclusion of Protection-Capability and Post-PANA-Address-Configuration AVPs in a PANA-Start-Request message is susceptible to spoofing leading to denial-of service attacks. Therefore, usage of these AVPs during the discovery and handshake phase in such insecure networks is NOT RECOMMENDED. The same AVPs are delivered via an integrity-protected PANA-Bind-Request upon successful authentication.

10.3 EAP Methods

Eavesdropping EAP messages might cause problems when the EAP method is weak and enables dictionary or replay attacks or even allows an adversary to learn the long-term password directly. Furthermore, if the optional EAP Response/Identity payload is used then it allows the adversary to learn the identity of the PaC. In such a case a privacy problem is prevalent.

To prevent these threats, [I-D.ietf-pana-framework] suggests using proper EAP methods for particular environments. Depending on the deployment environment an EAP authentication method which supports user identity confidentiality, protection against dictionary attacks and session key establishment must be used. It is therefore the responsibility of the network operators and users to choose a proper EAP method.

10.4 Separate NAP and ISP Authentication

The PANA design allows running two separate EAP sessions for the same PaC in the authentication and authorization phase: one with the NAP, and one with the ISP. The process of arriving at the resultant authorization, which is a combination of the individual authorizations obtained from respective service providers, is outside the scope of this protocol. In the absence of lower-layer security, both authentications MUST be able to generate a AAA-Key, leading to generation of a PANA SA. The resultant PANA SA cryptographically binds the two AAA-Keys together, hence it prevents man-in-the-middle attacks.

10.5 Cryptographic Keys

When the EAP method exports a AAA-Key, this key is used to produce a PANA SA with PANA_MAC_KEY with a distinct key ID. The PANA_MAC_KEY is unique to the PANA session, and takes PANA-based nonce values into computation to cryptographically separate itself from the AAA-Key.

The PANA_MAC_KEY is solely used for authentication and integrity protection of the PANA messages within the designated session.

Two AAA-Keys may be generated as a result of separate NAP and ISP authentication. In that case, the AAA-Key used with the PANA SA is the combination of both keys.

The PANA SA lifetime is bounded by the AAA-Key lifetime. Another execution of EAP method yields in a new AAA-Key, and updates the PANA SA, PANA_MAC_KEY and key ID.

When link-layer or network-layer ciphering [I-D.ietf-pana-ipsec] is enabled as a result of successful PANA authentication, a separate PaC-EP master key is generated based on the AAA-Key, session identifier, key identifier, and EP device identifier.

The lifetime of PaC-EP master key is bounded by the lifetime of the PANA SA. This key may be used with a secure association protocol [I-D.ietf-ipsec-ikev2] to produce further cipher-specific and transient keys.

10.6 Per-packet Ciphering

Networks that are not secured at the lower-layers prior to running PANA can rely on enabling per-packet data traffic ciphering upon successful PANA session establishment. The PANA framework allows generation of a PaC-EP master key from AAA-Key for using with a per-packet protection mechanism, such as link-layer or IPsec-based ciphering [I-D.ietf-pana-ipsec]. In case the master key is not readily useful to the ciphering mechanism, an additional secure association protocol [I-D.ietf-ipsec-ikev2] may be needed to produce the required keying material. These mechanisms ultimately establish a cryptographic binding between the data traffic generated by and for a client and the authenticated identity of the client. Data traffic must be minimally data origin authenticated, replay and integrity protected, and optionally encrypted.

10.7 PAA-to-EP Communication

The PANA framework allows separation of PAA from EP(s). SNMPv3 [I-D.ietf-pana-snmp] is used between the PAA and EP for provisioning authorized PaC information on the EP. This exchange MUST be always physically or cryptographically protected for authentication, integrity and replay protection. It MUST also be privacy-protected when PaC-EP master key for per-packet ciphering is transmitted to the EP.

The PaC-EP master key MUST be unique to the PaC and EP pair. The session identifier and the device identifier of the EP are taken into computation for achieving this effect [I-D.ietf-pana-ipsec]. Compromise of an EP does not automatically lead to compromise of

another EP or the PAA.

10.8 Liveness Test

A PANA session is associated with a session lifetime. The session is terminated unless it is refreshed by a new round of EAP authentication before it expires. Therefore, at the latest a disconnected client can be detected when its session expires. A disconnect may also be detected earlier by using PANA ping messages. A request message can be generated by either PaC or PAA at any time and the peer must respond with an answer message. A successful round-trip of this exchange is a simple verification that the peer is alive.

This test can be engaged when there is a possibility that the peer might have disconnected (e.g., after the discontinuation of data traffic for an extended period of time). Periodic use of this exchange as a keep-alive requires additional care as it might result in congestion and hence false alarms.

This exchange is cryptographically protected when a PANA SA is available in order to prevent threats associated with the abuse of this functionality.

Any valid PANA answer message received in response to a recently sent request message can be taken as an indication of peer's liveness. The PaC or PAA MAY forgo sending an explicit PANA-Ping-Request if a recent exchange has already confirmed that the peer is alive.

10.9 Updating PaC's IP Address

Even though the IP-Address AVP in a PANA-Update-Request can be cryptographically protected by the MAC AVP, there is not way to prove the ownership of the IP address presented by the PaC. Hence an authorized PaC can launch a redirect attack by spoofing a victim's IP address.

10.10 Early Termination of a Session

The PANA protocol supports the ability for both the PaC and the PAA to transmit a tear-down message before the session lifetime expires. This message causes state removal, a stop of the accounting procedure and removes the installed per-PaC state on the EP(s). This message is cryptographically protected when PANA SA is present.

11. Acknowledgments

We would like to thank Jari Arkko, Mohan Parthasarathy, Julien Bournelle, Rafael Marin Lopez, Pasi Eronen, Randy Turner, Erik Nordmark, Lionel Morand, Avi Lior, Susan Thomson, Giaretta Gerardo and all members of the PANA working group for their valuable comments to this document.

12. References

12.1 Normative References

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

[RFC2234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

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

[RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[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.

[RFC3456] Patel, B., Aboba, B., Kelly, S. and V. Gupta, "Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode", RFC 3456, January 2003.

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

[I-D.ietf-eap-keying] Aboba, B., "Extensible Authentication Protocol (EAP) Key Management Framework", draft-ietf-eap-keying-04 (work in progress), November 2004.

[IANA] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

12.2 Informative References

[I-D.ietf-pana-requirements]
Yegin, A. and Y. Ohba, "Protocol for Carrying Authentication for Network Access (PANA)Requirements", draft-ietf-pana-requirements-09 (work in progress), August 2004.

[RFC2522] Karn, P. and W. Simpson, "Photuris: Session-Key Management Protocol", RFC 2522, March 1999.

[I-D.ietf-pana-threats-eval]
Parthasarathy, M., "Protocol for Carrying Authentication and Network Access Threat Analysis and Security Requirements", draft-ietf-pana-threats-eval-07 (work in progress), August 2004.

[I-D.ietf-pana-ipsec]
Parthasarathy, M., "PANA enabling IPsec based Access Control", draft-ietf-pana-ipsec-05 (work in progress), December 2004.

[I-D.ietf-pana-framework]
Jayaraman, P., "PANA Framework", draft-ietf-pana-framework-02 (work in progress), September 2004.

[I-D.ietf-pana-snmp]
Mghazli, Y., Ohba, Y. and J. Bournelle, "SNMP usage for PAA-2-EP interface", draft-ietf-pana-snmp-02 (work in progress), October 2004.

[I-D.ietf-eap-statemachine]
Vollbrecht, J., Eronen, P., Petroni, N. and Y. Ohba, "State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator", draft-ietf-eap-statemachine-05 (work in progress), September 2004.

[I-D.ietf-ipsec-ikev2]
Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-17 (work in progress), October 2004.

[I-D.ietf-dna-link-information]
Yegin, A., "Link-layer Event Notifications for Detecting Network Attachments", draft-ietf-dna-link-information-00 (work in progress), September 2004.

[I-D.adrangi-eap-network-discovery]
Adrangi, F., "Mediating Network Discovery in the Extensible Authentication Protocol (EAP)", draft-adrangi-eap-network-discovery-07 (work in progress), December 2004.

[ianaweb] IANA, "Number assignment", http://www.iana.org.

[IANA-EXP]
Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.

[RFC2279] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

Authors' Addresses

Dan Forsberg
Nokia Research Center
P.O. Box 407
FIN-00045 NOKIA GROUP
Finland

Phone: +358 50 4839470
EMail: dan.forsberg@nokia.com

Yoshihiro Ohba
Toshiba America Research, Inc.
1 Telcordia Drive
Piscataway, NJ 08854
USA

Phone: +1 732 699 5305
EMail: yohba@tari.toshiba.com

Basavaraj Patil
Nokia
6000 Connection Dr.
Irving, TX 75039
USA

Phone: +1 972-894-6709
EMail: Basavaraj.Patil@nokia.com

Hannes Tschofenig
Siemens Corporate Technology
Otto-Hahn-Ring 6
81739 Munich
Germany

EMail: Hannes.Tschofenig@siemens.com

Alper E. Yegin
Samsung Advanced Institute of Technology
75 West Plumeria Drive
San Jose, CA 95134
USA

Phone: +1 408 544 5656
EMail: alper.yegin@samsung.com

Appendix A. Example Sequence of Separate NAP and ISP Authentication

A PANA message sequence with separate NAP and ISP authentication is illustrated in Figure 12. The example assumes the following scenario:

- o The PaC initiates the discovery and handshake phase.
- o The PAA offers separate NAP and ISP authentication, as well as a choice of ISP from "ISP1" and "ISP2". The PaC accepts the offer from PAA, with choosing "ISP1" as the ISP.
- o NAP authentication and ISP authentication is performed in this order in the authentication and authorization phase.
- o An EAP authentication method with a single round trip is used in each EAP sequence.
- o After a PANA SA is established, all messages are integrity and replay protected with MAC AVPs.
- o The access, re-authentication and termination phases are not shown.

```
PaC      PAA  Message(sequence number)[AVPs]
-----------------------------------------------------
// Discovery and handshake phase
   ----->     PANA-PAA-Discover(0)
   <-----     PANA-Start-Request(x)              // S-flag set
                 [Nonce, Cookie,
                  ISP-Information("ISP1"),
                  ISP-Information("ISP2"),
                  NAP-Information("MyNAP")]
   ----->     PANA-Start-Answer(x)               // S-flag set
                 [Nonce, Cookie,                 // PaC chooses "ISP1"
                  ISP-Information("ISP1")]

// Authentication and authorization phase
   <-----     PANA-Auth-Request(x+1)             // NAP authentication
                 [Session-Id, EAP{Request}]      // S- and N-flags set
   ----->     PANA-Auth-Answer(x+1)              // S- and N-flags set
                 [Session-Id]                    // No piggybacking
   ----->     PANA-Auth-Request(y)               // S- and N-flags set
                 [Session-Id, EAP{Response}]
   <-----     PANA-Auth-Answer(y)[Session-Id]    // S- and N-flags set
   <-----     PANA-Auth-Request(x+2)             // S- and N-flags set
                 [Session-Id, EAP{Request}]
```

```
      ----->     PANA-Auth-Answer(x+2)              // S- and N-flags set
                    [Session-Id, EAP{Response}]      // Piggybacking
      <-----     PANA-FirstAuth-End-Request(x+3)     // S- and N-flags set
                    [Session-Id, EAP{Success}, Key-Id, MAC]
      ----->     PANA-FirstAuth-End-Answer(x+3)      // S- and N-flags set
                    [Session-Id, Key-Id, MAC]
      <-----     PANA-Auth-Request(x+4)              // ISP authentication
                    [Session-Id, EAP{Request}, MAC]  // S-flag set
      ----->     PANA-Auth-Answer(x+4)               // S-flag set
                    [Session-Id, MAC]                // No piggybacking
      ----->     PANA-Auth-Request(y+1)              // S-flag set
                    [Session-Id, EAP{Response}, MAC]
      <-----     PANA-Auth-Answer(y+1)               // S-flag set
                    [Session-Id, MAC]
      <-----     PANA-Auth-Request(x+5)              // S-flag set
                    [Session-Id, EAP{Request}, MAC]
      ----->     PANA-Auth-Answer(x+5)               // S-flag set
                    [Session-Id, EAP{Response}, MAC] // Piggybacking
      <-----     PANA-Bind-Request(x+6)              // S-flag set
                    [Session-Id, Result-Code, EAP{Success}, Device-Id,
                    IP-Address, Key-Id, Lifetime,
                    Protection-Cap., PPAC, MAC]
      ----->     PANA-Bind-Answer(x+6)               // S-flag set
                    [Session-Id, Device-Id, Key-Id,
                     PPAC, MAC]
```

Figure 12: A Complete Message Sequence for Separate NAP and ISP Authentication

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.