INTERNET DRAFT
Category: Standards Track
Title: draft-calhoun-diameter-02.txt
Date: March 1998

Pat R. Calhoun
Sun Microsystems, Inc.
Allan C. Rubens
Ascend Communications

DIAMETER
Base Protocol
<draft-calhoun-diameter-02.txt>

Status of this Memo

Abstract

The DIAMETER base protocol is intended to provide a framework for any services which require AAA/Policy support. The protocol is inteded to be flexible enough to allow services to add building blocks to DIAMETER in order to meet their requirements.

This draft MUST be supported by all DIAMETER implementations, regardless of the specific underlying service.

Table of Contents

1.0 Introduction

Since the RADIUS protocol is being used today for much more than simple authentication and accounting of dial-up users (i.e. authentication of WEB clients, etc), a more extensible protocol was

necessary which could support new services being deployed in the internet and corporate networks.

RADIUS in itself is not an extensible protocol largely due to its very limited command and attribute address space. In addition, the RADIUS protocol assumes that there cannot be any unsolicited messages from a server to a client. In order to support new services it is imperative that a server be able to send unsolicited messages to clients on a network, and this is a requirement for any DIAMETER implementation.

This document describes the base DIAMETER protocol. This document in itself is not complete and MUST be used with an accompanying applicability extension document.

An example of such a document would be [8] which defines extensions to the base protocol to support user authentication and [x] which defines extensions to support accounting.

1.1 Definitions

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

MUST — This word, or the adjective "required", means that the definition is an absolute requirement of the specification.

MUST NOT — This phrase means that the definition is an absolute prohibition of the specification.

SHOULD — This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighed before choosing a different course.

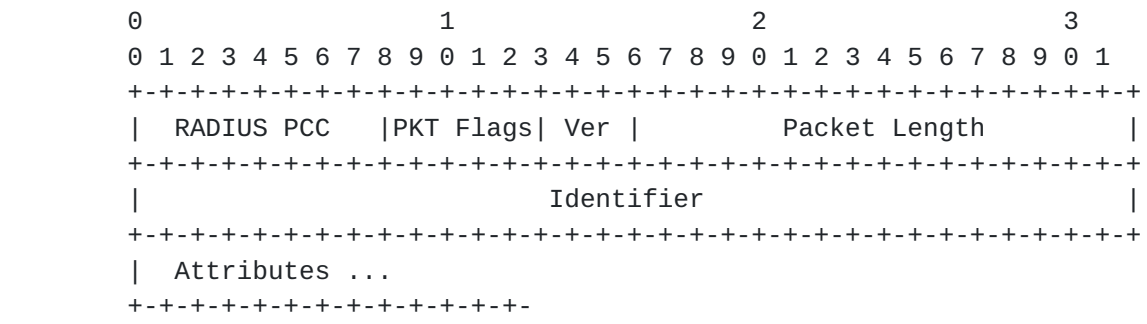
MAY — This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

2.0 DIAMETER Header Format

DIAMETER packets MAY be transmitted over UDP or TCP. Each Service Extensions draft SHOULD specify the transport layer. The destination port field for DIAMETER is 1645.

For UDP, when a reply is generated the source and destination ports are reversed.

A summary of the DIAMETER data format is shown below. The fields are transmitted from left to right.



RADIUS PCC (Packet Compatibility Code)

The RADIUS PCC field is a one octet field which is used for RADIUS backward compatibility. In order to easily distinguish DIAMETER packets from RADIUS a special value has been reserved and allows an implementation to support both protocols concurently using the first octet in the header. The RADIUS PCC field MUST be set as follows:

254 DIAMETER packet

PKT Flags

The Packet Flags field is five bits, and is used in order to identify any options. This field MUST be set initialized to zero. No flags are defined at this time.

Version

The Version field is three bits, and indicates the version number which is associated with the packet received. This field MUST be set to 1 to indicate DIAMETER Version 1.

Packet Length

The Packet Length field is two octets. It indicates the length of the packet including the header fields. For messages received via UDP, octets outside the range of the Length field should be treated as padding and should be ignored on receipt.

Identifier

The Identifier field is four octets, and aids in matching requests and replies. The identifier MUST be unique at any given time and one mechanism to ensure this is to use a monotonically increasing number. Given the size of the Identifier field it is unlikely that

2^32 requests could be outstanding at any given time.

Attributes

See section 3.0 for more information of attribute formats.

3.0 DIAMETER AVP Format

DIAMETER Attributes carry the specific authentication and authorization information as well as configuration details for the request and reply.

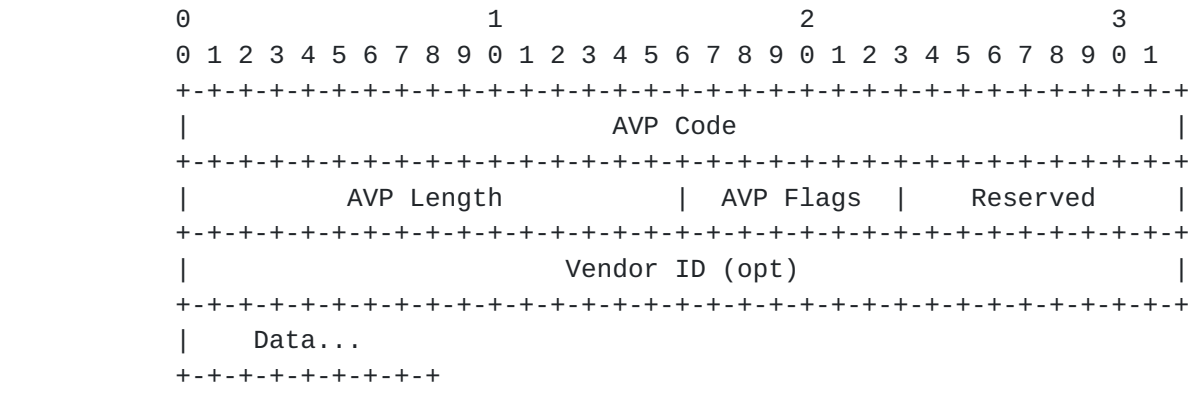
Some Attributes MAY be listed more than once. The effect of this is Attribute specific, and is specified by each such attribute description.

Each AVP MUST be padded to align on a 32 bit boundary. Although this is not problematic for most attribute types, it does require that AVP of string and data type be padded accordingly. The Padding size can be deduced using the following formula:

```
padding_size = Length % 4
```

The end of the list of attributes is defined by the length of the DIAMETER packet minus the length of the header.

A summary of the attribute format is shown below. The fields are transmitted from left to right.



AVP Code

The AVP Code field is four octets. The first 256 AVP numbers are reserved for backward RADIUS compatibility. Up-to-date values of the RADIUS Type field are specified in the most recent "Assigned Numbers" RFC [2].

AVP numbers 257 and above are used for DIAMETER. Each service MUST allocate AVP numbers through the IANA.

If the Vendor ID bit is set the AVP Code is allocated from the vendor's private address space.

AVP Length

The AVP Length field is two octets, and indicates the length of this Attribute including the Address Type, AVP Length, AVP Flags, Reserved, Vendor ID if present and the AVP data. If a packet is received with an Invalid attribute length, the packet SHOULD be rejected.

AVP Flags

The AVP Flags field informs the DIAMETER host how each attribute must be handled. The following values are currently defined:

Mandatory-Support 1
The receiver MUST support this attribute. If the attribute is NOT supported, the device MUST reject the Command. If this flag is not set, then the receiver MAY accept the command regardless of whether or not the particular attribute is recognized.

SS-Encrypted-Data 2
If this bit is set, the contents of the attributes are encrypted. Note that the attribute header is NOT encrypted in this case. See section 5.2 for more information.

PK-Encrypted-Data 4
If this bit is set, the contents of the attributes are encrypted. Note that the attribute header is NOT encrypted in this case. See section 5.2 for more information.

Vendor-Specific-AVP 8
If this bit is set, the optional Vendor ID field will be present in the AVP header and the AVP Code MUST be treated accordingly.

Reserved

The Reserved field MUST be set to zero (0).

Vendor ID

The optional four octet Vendor ID field contains the the IANA assigned "SMI Network Management Private Enterprise Codes" value, encoded in network byte order. Any vendor wishing to implement DIAMETER extensions can use their own Vendor ID along with private Attribute values, guaranteeing that they will not collide with any

other vendor's extensions, nor with future IETF extensions.

The value 0, reserved in this protocol, corresponds to IETF adopted Attribute values, defined within this document and MUST NOT be used in an AVP since it is implied with the absence of the Vendor-Specific-AVP bit.

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.

The format of the value field MAY be one of six data types. It is possible for an attribute to have a structure and this MUST be defined along with the attribute.

Data

0-65526 octets of arbitrary data.

String

0-65526 octets of string data in some agreed upon char set.

Address

32 bit or 48 bit value, most significant octet first. The length of the attribute is determined by the flag field.

Integer32

32 bit value, most significant octet first.

Integer64

64 bit value, most significant octet first.

Time

32 bit value, most significant octet first -- seconds since 00:00:00 GMT, January 1, 1970.

4.0 DIAMETER AVPs

This section will define the mandatory AVPs which MUST be supported by all DIAMETER implementations. Note the first 256 AVP numbers are reserved for RADIUS compatibility.

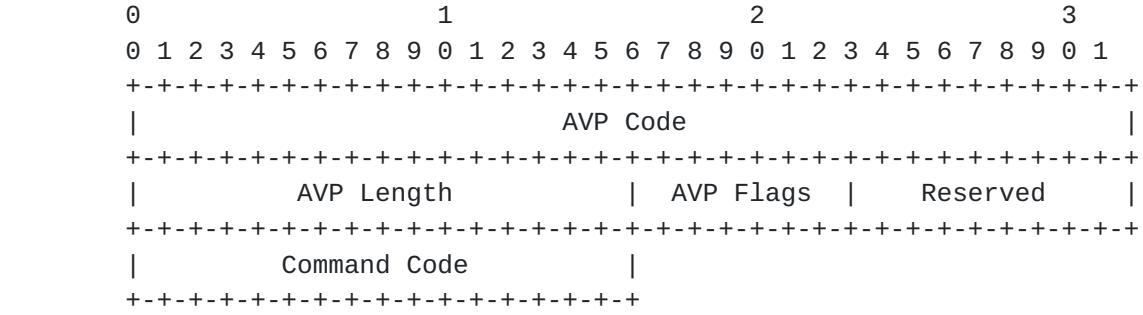
The following AVPs are defined in this document:

Attribute Name	Attribute Code
DIAMETER-Command	256
Host-IP-Address	4
Host-Name	32
Supported-Extension	258
Integrity-Check-Vector	259
Digital-Signature	260
Initialization-Vector	261
Timestamp	262
Session-Id	263
X509-Certificate	264
X509-Certificate-URL	265
Vendor-Name	266
Firmware-Revision	267

4.1 DIAMETER-Command AVP

Description

The Command AVP MUST be the first AVP following the DIAMETER header. This AVP is used in order to communicate the command associated with the message. There MUST only be a single Command AVP within a given message. The format of the AVP is as follows:



AVP Code

256 DIAMETER-Command

AVP Length

The length of this attribute MUST be at least 10. The exact length of the AVP is determined by the actual Command and is defined with each command.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set. Bit 2

(Encrypted AVP) SHOULD NOT be set. The optional Vendor ID bit MAY

be set if the command is vendor-specific.

Reserved

The Reserved field MUST be set to zero (0).

Command Code

The Command Code field contains the command number. The following commands are defined in this document:

The following commands MUST be supported by all DIAMETER implementations in order to conform to the base protocol specification:

Command Name	Command Code
Command-Unrecognized	256
Device-Reboot-Indication	257
Device-Reboot-Ack	258
Device-Feature-Query	259
Device-Feature-Response	260

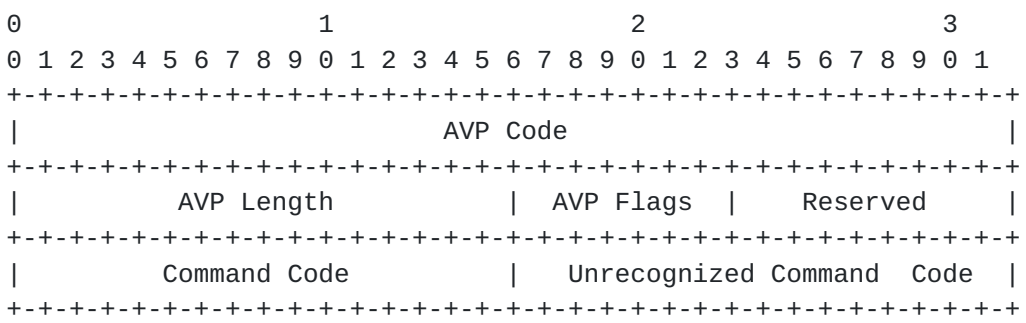
4.1.1 Command-Unrecognized

Description

Messages with the Command-Unrecognized AVP MUST be sent by a DIAMETER device to inform its peer that a message was received with an unsupported Command AVP value.

Since there certainly will exist a case where an existing device does not support a new extension to the DIAMETER protocol, a device which receives a packet with an unrecognized Command code MUST return a Command-Unrecognized packet.

A summary of the Command-Unrecognized packet format is shown below. The fields are transmitted from left to right.



AVP Code

256 DIAMETER Command

AVP Length

The length of this attribute MUST be 12.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Command Code

The Command Code field MUST be set to 256 (Command-Unrecognized).

Unrecognized Command Code

The Unrecognized Command Code field MUST contain the Command Code that resulted in this message.

4.1.2 Device-Reboot-Indication

Description

The Device-Reboot-Indication message is sent by a DIAMETER device to inform all of its peers that it has rebooted. The peer MUST respond to the message with a successful acknowledgement. Note that a device MUST only send this message once it is ready to receive packets.

This message is also used by a DIAMETER device in order to exchange the supported protocol version number as well as all supported extensions. The originator of this message MUST insert it's highest supported version number within the DIAMETER header. The response message MUST include the highest supported version up to and including the version number within the request.

Similarly the originator of this message MUST include all supported extensions within the message. The responder MUST include all supported extensions as long as they were present within the request message.

In the case where the receiver of this message is a proxy device, it is responsible for inserting the highest version number which it supports in the version field before sending the proxy request to the remote DIAMETER peer. The proxy device may then retain the version number of the remote peers as received in the message, and

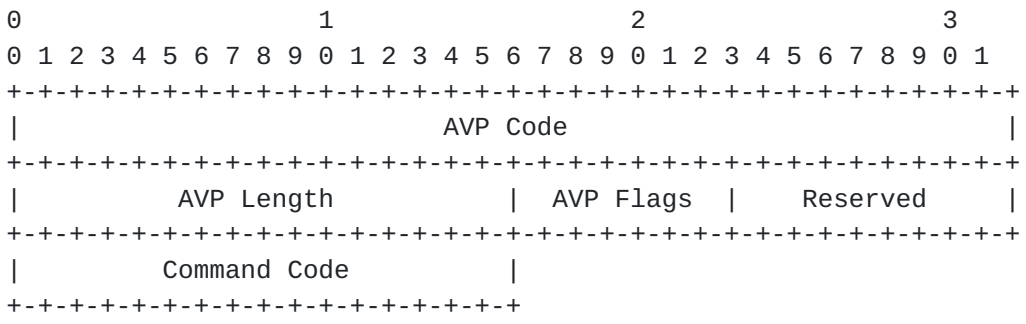
must insert its highest version number (with the limitations

described above) in the response to the initiator.

It is desireable for a DIAMETER device to retain the supported extensions as well as the version number in order to ensure that any requests issued to a peer will be processed.

This message MAY contain the Version-Number, Vendor-Name and Supported-Extension AVPs.

In the case where a DIAMETER device is configured to communicate with many peers, this message MUST be issued to each peer.



AVP Code

256 DIAMETER Command

AVP Length

The length of this attribute MUST be 10.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Command Code

The Command Code field MUST be set to 257 (Device-Reboot-Indication).

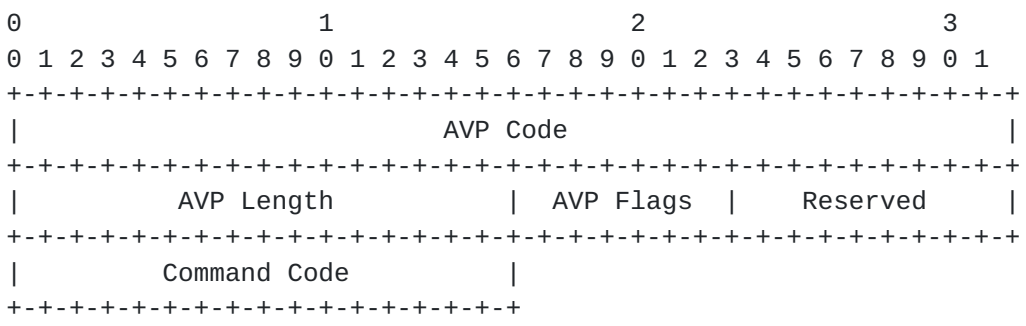
4.1.3 Device-Reboot-Ack

Description

The Device-Reboot-Ack message is sent by a DIAMETER device to acknowledge the receipt of the Device-Reboot-Indication message. The originator of this message MUST include the highest support version number (up to and including the value in the request) in the DIAMETER header as well as all supported extensions (as long as

they were present in the requesting message).

This message MAY contain the Version-Number, Vendor-Name and Supported-Extension AVPs.



AVP Code

256 DIAMETER Command

AVP Length

The length of this attribute MUST be 10.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

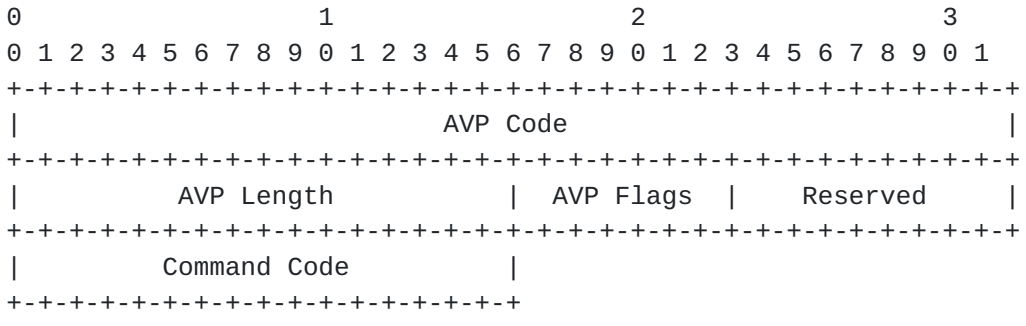
Command Code

The Command Code field MUST be set to 258 (Device-Reboot-Ack).

4.1.4 Device-Feature-Query

Description

The Device-Feature-Query message is used in order to query a peer about it's supported extensions. This message MAY contain the Version-Number, Vendor-Name and Supported-Extension AVPs.



AVP Code

256 DIAMETER Command

AVP Length

The length of this attribute MUST be 10.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

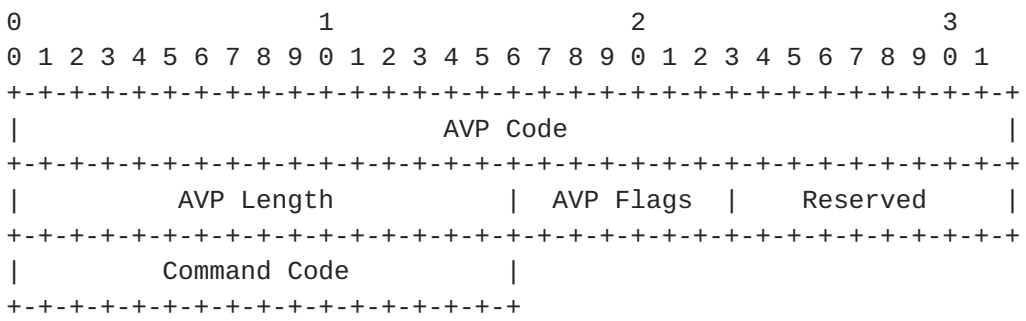
Command Code

The Command Code field MUST be set to 259 (Device-Feature-Request).

4.1.5 Device-Feature-Response

Description

The Device-Feature-Response message is sent in response to the Device-Feature-Query message. This message includes all supported extensions by the responder and MAY contain the Version-Number, Vendor-Name and Supported-Extension AVPs.



AVP Code

256 DIAMETER Command

AVP Length

The length of this attribute MUST be 10.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

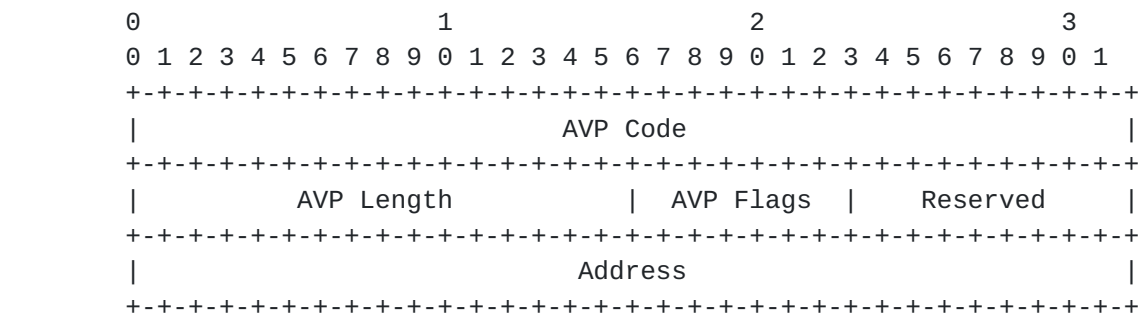
Command Code

The Command Code field MUST be set to 260 (Device-Feature-Response).

4.2 Host-IP-Address

Description

The Host-IP-Address attribute is used to inform a DIAMETER peer of the sender's identity.



AVP Code

4 Host-IP-Address

AVP Length

The length of this attribute MUST be at least 12.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

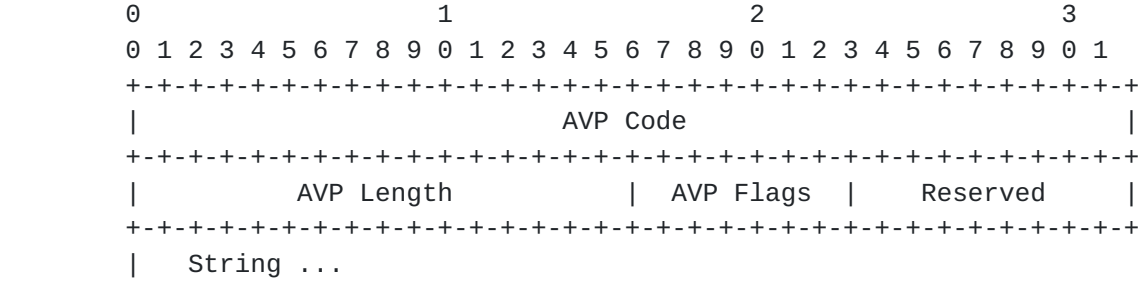
Address

The Address field contains the sender's IP address.

4.3 Host-Name

Description

The Host-Name attribute is used to inform a DIAMETER peer of the sender's identity.



AVP Code

32 Host-Name

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

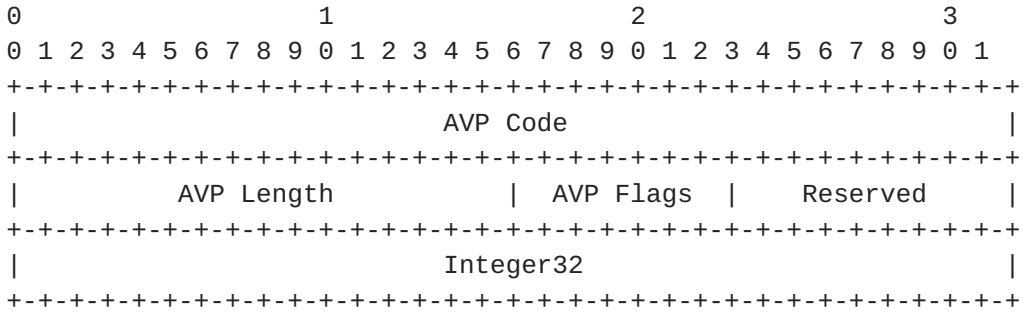
String

The String field is one or more octets, and should be unique to the DIAMETER host. It is strongly encouraged that the Host Name follow the NAI [8] conventions.

4.4 Version-Number

Description

The Version-Number AVP is used in order to indicate the current DIAMETER system version number to a peer.



AVP Code

257 Version-Number

AVP Length

The length of this attribute MUST be 12.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Integer32

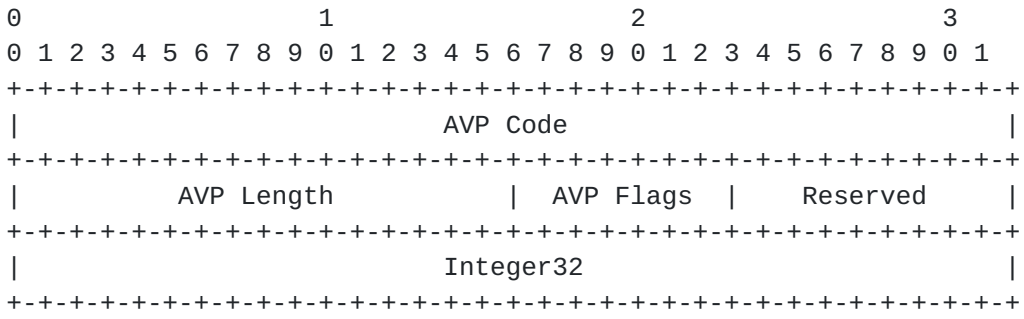
The Integer32 field contains the system version number.

4.5 Supported-Extension

Description

The Supported-Extension AVP is used to inform a peer of the supported extensions. Note that each supported extensions draft MUST have an identifier assigned. The base protocol is not considered an extension since its support is mandatory.

Each DIAMETER Extension draft will be assigned an extension number by the IANA. The value of the extension number is passed along in this AVP. There MAY be more than one Supported-Extension AVP within a DIAMETER message.



AVP Code

258 Supported-Extension

AVP Length

The length of this attribute MUST be 12.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Integer32

The Integer32 field contains the supported extension number as defined in the extension's document.

4.6 Integrity-Check-Vector

Description

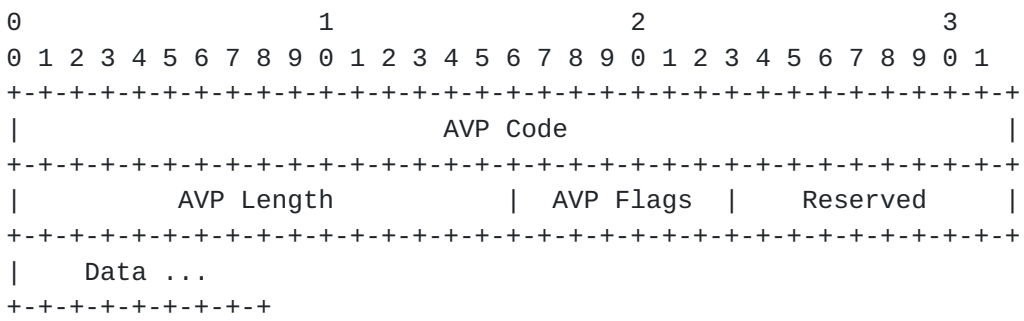
The Integrity-Check-Vector AVP is used for hop-by-hop

authentication and integrity, and is not recommended for use with untrusted proxy servers.

The DIAMETER header as well as all AVPs up to and including the AVP Code field of this AVP is protected by the ICV.

The ICV is generated in the method described in section 5.1.1.

All DIAMETER implementations MUST support this AVP.



AVP Code

259 Integrity-Check-Vector

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Data

The Data field contains an HMAC-MD5-96[6] of the message up to and including the attribute type field of this AVP.

4.7 Digital-Signature

Description

The Digital-Signature AVP is used for authentication, integrity as well as non-repudiation. The DIAMETER header as well as all AVPs up to and including the AVP Code field of this AVP is protected by the Digital-Signature.

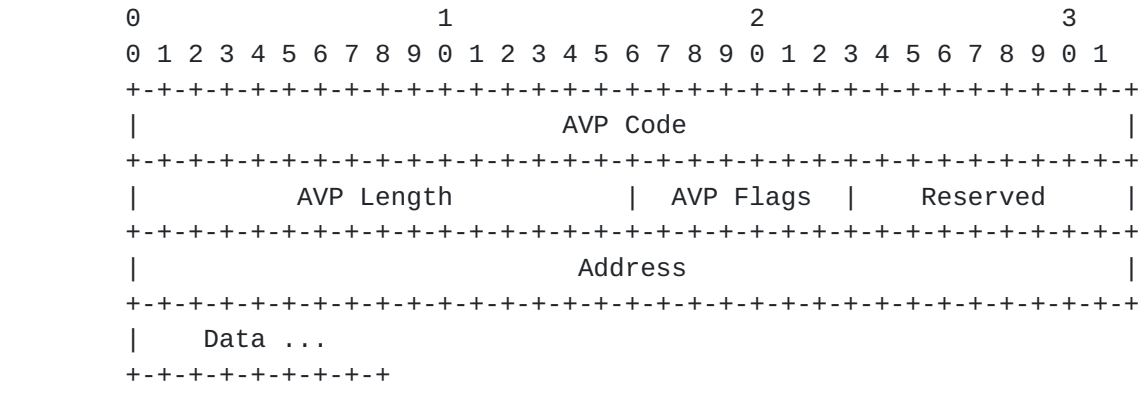
Note that for services which use the concept of a proxy server which forwards the request to a next hop server, the proxy server MUST NOT modify any attributes found prior to the Digital-Signature

AVP. This ensures that end-to-end security is maintained even

through proxy arrangements.

The Digital-Signature is generated in the method described in section 5.1.2.

All DIAMETER implementations SHOULD support this AVP.



AVP Code

260 Digital-Signature

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Address

The Address field contains the address of the DIAMETER host which generated the Digital-Signature.

Data

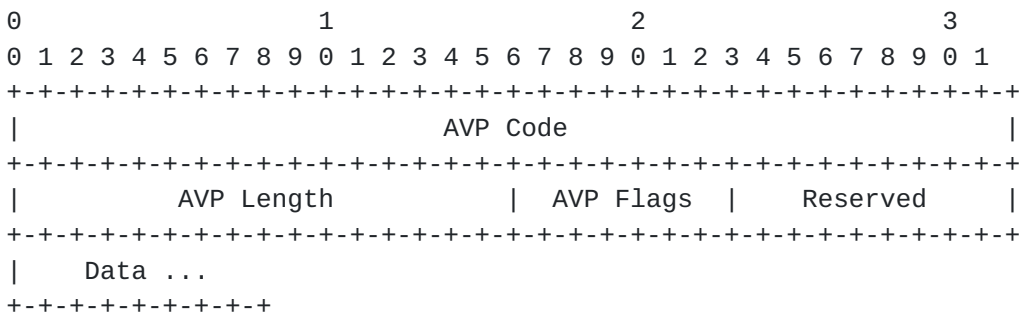
The Data field contains the digital signature of the packet up to and including the attribute type field within this AVP.

4.8 Initialization-Vector

Description

The Initialization-Vector AVP MUST be present prior to the Digital-Signature and Integrity-Check-Vector AVPs within a message and is used to ensure randomness within a message. The content of this AVP

MUST be a random 128 bit value.



AVP Code

261 Initialization-Vector

AVP Length

The length of this attribute MUST be 24.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Data

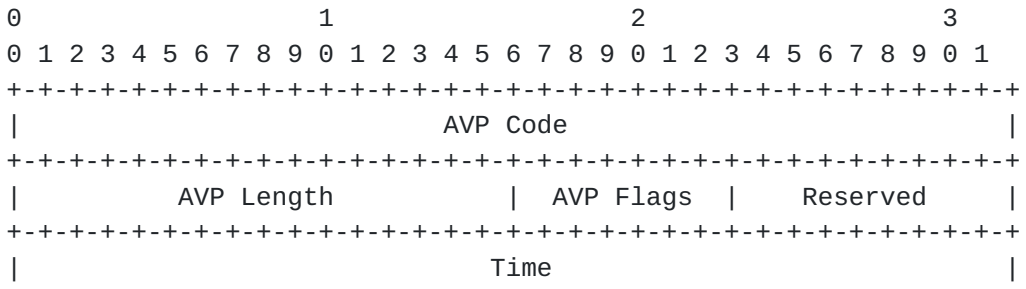
The Data field contains a random 128 bit value.

4.9 Timestamp

Description

The Timestamp field is used in order to enable replay protection of previous messages. The value of time is the most significant four octets returned from an NTP server which indicates the number of seconds expired since Jan. 1, 1970.

This document does not specify the window which an implementation will accept packets, however it is strongly encouraged to make this value user configurable with a reasonable default value (i.e. 4 seconds).



AVP Code

262 Timestamp

AVP Length

The length of this attribute MUST be 12.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Time

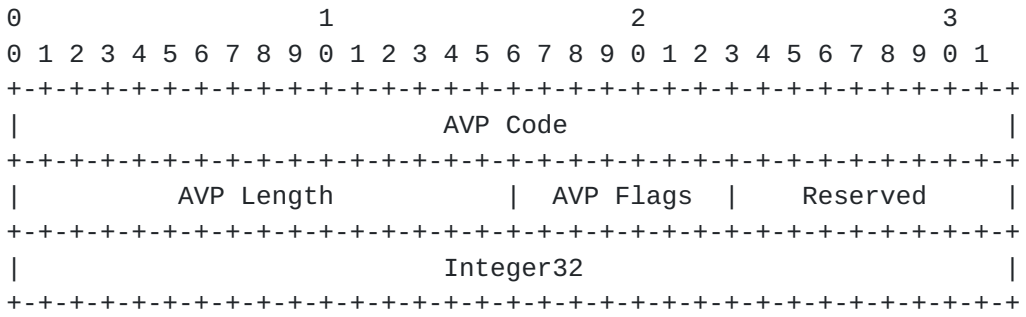
The Time field contains the number of seconds since Jan. 1, 1970 when the message was created.

4.10 Session-Id

Description

The Session-Id field is used in order to identify a specific session. All messages pertaining to a specific session MUST include this AVP and the same value MUST be used throughout the life of a session.

Note that in some applications there is no concept of a session (i.e. data flow) and this field MAY be used to identify objects other than a session.



AVP Code

263 Session-Id

AVP Length

The length of this attribute MUST be 12.

The flag field MUST have bit 1 (Mandatory Support) set.

Integer32

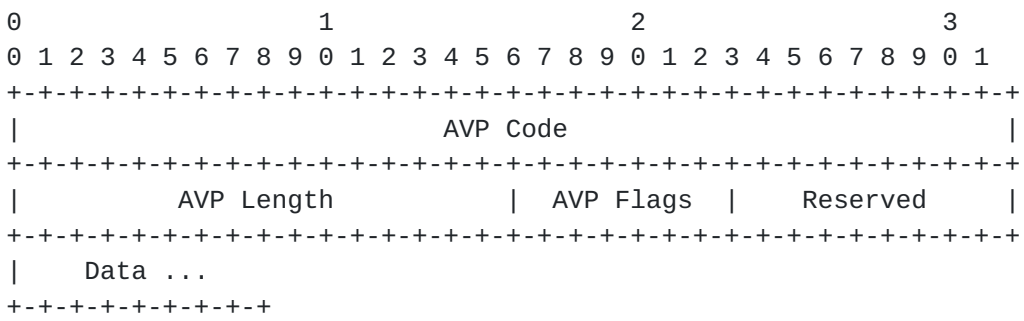
The Integer32 field contains the session identifier assigned to the session.

4.11 X509-Certificate

Description

The X509-Certificate is used in order to send a DIAMETER peer the local system's X.509 certificate chain, which is used in order to validate the Digital-Signature attribute.

Section 5.3 contains more information about the use of certificates.



AVP Code

264 X509-Certificate

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Data

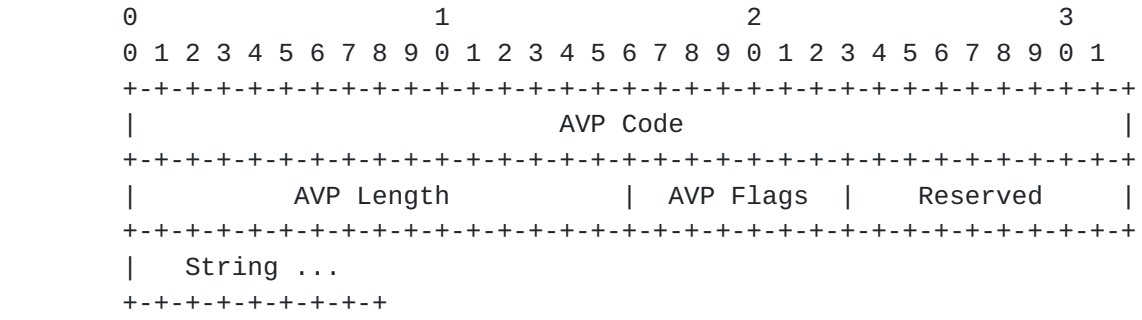
The Data field contains the X.509 Certificate Chain.

4.12 X509-Certificate-URL

Description

The X509-Certificate-URL is used in order to send a DIAMETER peer a URL to the local system's X.509 certificate chain, which is used in order to validate the Digital-Signature attribute.

Section 5.3 contains more information about the use of certificates.



AVP Code

265 X509-Certificate-URL

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

String

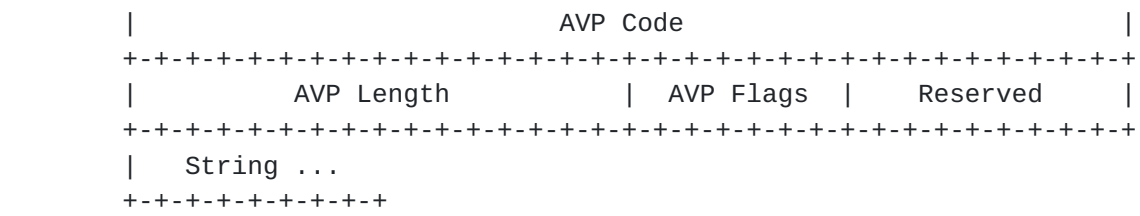
The String field contains the X.509 Certificate Chain URL.

4.13 Vendor-Name

Description

The Vendor-Name attribute is used in order to inform a DIAMETER peer of the Vendor of the DIAMETER protocol stack. This MAY be used in order to know which vendor specific attributes may be sent to the peer.

It is also envisioned that the combination of the Vendor-Name and the Firmware-Revision AVPs can provide very useful debugging information.



AVP Code

266 Vendor-Name

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

String

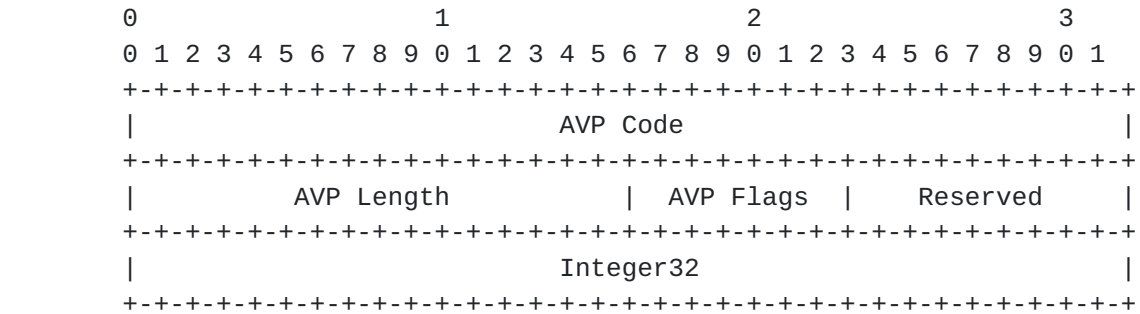
The String field contains the vendor name.

4.14 Firmware-Revision

Description

The Firmware-Revision AVP is used to inform a DIAMETER peer of the firmware revision of the issuing device.

For devices which do not have a firmware revision (general purpose computers running DIAMETER software modules, for instance), the revision of the DIAMETER software module may be reported instead.



AVP Code

267 Firmware-Revision

AVP Length

The length of this attribute MUST be at least 12.

AVP Flags

The flag field MUST have bit 1 (Mandatory Support) set.

Integer32

The Integer32 field contains the firmware revision number of the issuing device.

5.0 Protocol Definition

This section will describe how the base protocol works (or is at least an attempt to).

5.1 Data Integrity

This section will describe how data integrity is achieved using the Data Integrity AVPs.

Note that the Timestamp and Initialization-Vector AVPs MUST be present in the message PRIOR to any of the Data Integrity AVPs discussed in this section.

5.1.1 Using Integrity Check Vector

The use of the Integrity-Check-Vector AVP requires a pre-configured shared secret. Although this mechanism does not scale as well as the Digital Signature, it may be desireable to use this mechanism in the case where asymetric technology is not required or available.

Note that in the case where two DIAMETER nodes need to communicate through an intermediate node (i.e. Proxy) it does not offer any end-to-end data integrity or encryption as each node must re-compute the Integrity-Check-Vector AVP.

The Data field of the AVP contains an HMAC-MD5-96[6] of the message up to and including the attribute type field of the AVP. Using the example code provided in [6], the following call would be used to generate the ICV:

```
hmac_md5(DiameterMessage, MessageLength, Secret, Secretlength,
         Output)
```

The Timestamp attribute provides replay protection and this AVP MUST

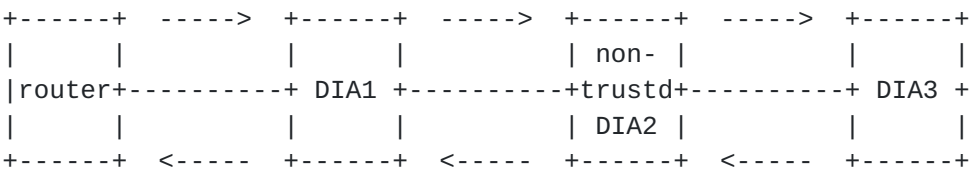
be present prior to the Integrity-Check-Vector AVP. In addition the

Initialization Vector AVP MUST also be present prior to the Integrity-Check-Vector AVP in order to provide some randomness.

5.1.2 Using Digital Signatures

In the case of a simple peer to peer relationship the use of IPSEC is sufficient for data integrity and non-repudiation. However there are instances where a peer must communicate with another peer through the use of a proxy server. IPSEC does not provide a mechanism to protect traffic when two peers must use an intermediary node to communicate at the application layer therefore the Digital-Signature AVP MUST be used.

The following diagram shows an example of a router initiating a DIAMETER message to DIA1. Once DIA1 has finished processing the message it adds its signature and forwards the message to the non-trusted DIA2 proxy server. If DIA2 needs to add or change any muteable AVPs it SHOULD add its digital signature before forwarding the message to DIA3.



Since some fields within the DIAMETER header will change "en route" towards the final DIAMETER destination, it is necessary to set the mutable fields to zero (0) prior to calculating the signature. The two mutable fields are the identifier and the length.

The Timestamp attribute provides replay protection and this AVP MUST be present prior to the Digital Signature AVP. In addition the Initialization Vector AVP MUST also be present prior to the Digital Signature AVP in order to provide some randomness.

Note that Digital Signatures only protect the DIAMETER header as well as all AVPs found prior to the digital signature. It is therefore possible to have AVPs following the digital signature and these are considered unprotected.

The Data field of the Digital-Signature AVP contains the RSA/MD5 signature algorithm as defined in [9].

5.1.3 Using Mixed Data Integrity AVPs

Both previous sections describe the differences between the Integrity-

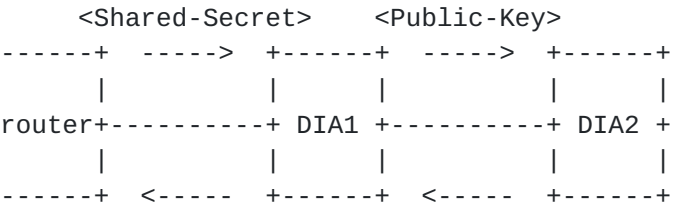
Check-Vector and the Digital-Signature AVP. Note that it is valid to

use both within a single DIAMETER message.

In the case where an intermediate DIAMETER server is used to reach the end DIAMETER Server, the Integrity-Check-Vector AVP provides hop-by-hop integrity and requires that each set of DIAMETER peers share a secret.

The Digital-Signature AVP provides end-to-end integrity and requires knowledge of all parties public keys.

If both AVPs co-exist within a single DIAMETER message, it is necessary to ensure that the Digital-Signature appears prior to the Integrity-Check-Vector since the ICV will be removed by the next hop.



There are cases, such as in remote access, where the device initiating the DIAMETER request does not have the processing power to generate Digital-Signatures as required by the protocol. In such an arrangement, there normally exists a first hop DIAMETER Server (DIA1) which acts as a proxy to relay the request to the final authenticating DIAMETER server (DIA2). It is valid for the first hop server to remove the Integrity-Check-Vector AVP inserted by the router and replace it with a Digital-Signature AVP.

5.2 AVP Data Encryption

DIAMETER supports two methods of encrypting AVP data. One is using a shared secret and the other is used with private keys.

This feature can be used to encrypt sensitive data such as user ID's and passwords. The Encryption bits MUST NOT be set in the Command Type or Initialization-Vector AVPs.

5.2.1 AVP Encryption with Shared Secrets

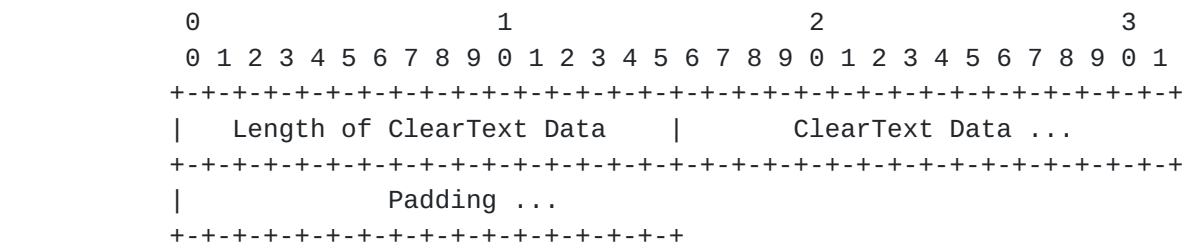
This method of encrypting AVP data is the simplest to use and MUST be supported by all DIAMETER implementations. However, local policy MAY determine that the use of this mechanism is not permitted.

The SS-Encrypted-Data bit MUST only be set if a shared secret exists between both DIAMETER peers. If the SS-Encrypted-Data bit is set in

any DIAMETER AVP, the Initialization-Vector AVP MUST be present prior

to the first encrypted AVP.

The length of the AVP value to be encrypted is first encoded in the following Subformat:



Length

The Length field contains the length of the decrypted data.

ClearText Data

Data of AVP that is to be obscured.

Padding

Random additional octets used to obscure length of the ClearText Data.

The resulting subformat MAY be padded to a multiple of 16 octets in length. For example, if the ClearText Data to be obscured is a string containing 6 characters (e.g. password 'foobar'), then 8 + n * 16 octects of padding would be applied. Padding does NOT alter the value placed in the Length of the ClearText Data, only the length of the AVP itself.

Next, An MD5 hash is performed on the concatenation of:

- the 2 octet Command Code of the AVP
- the shared authentication secret
- an arbitrary length random vector

The value of the random vector used in this hash is passed in the Data field of a Initialization-Vector AVP. This Initialization-Vector AVP must be placed in the message by the sender before any hidden AVPs. The same IV may be used for more than one hidden AVP in the same message. If a different IV is used for the hiding of subsequent AVPs then a new Initialization-Vector AVP must be placed before the first AVP to which it applies.

The MD5 hash value is then XORed with the first 16 octet or less segment of the AVP Subformat and placed in the Data field of the AVP.

If the AVP Subformat is less than 16 octets, the Subformat is

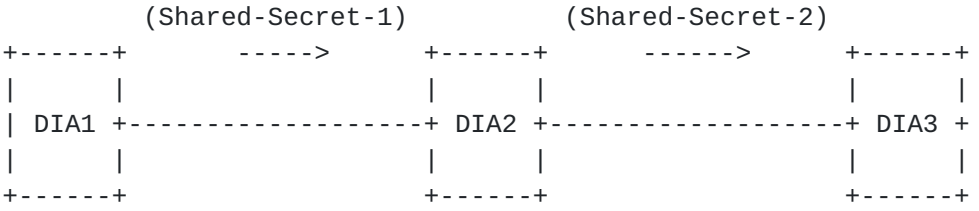
transformed as if the Value field had been padded to 16 octets before the XOR, but only the actual octets present in the Subformat are modified, and the length of the AVP is not altered.

If the Subformat is longer than 16 octets, a second one-way MD5 hash is calculated over a stream of octets consisting of the shared secret followed by the result of the first XOR. That hash is XORed with the second 16 octet or less segment of the Subformat and placed in the corresponding octets of the Data field of the AVP.

If necessary, this operation is repeated, with each XOR result being used along with the shared secret to generate the next hash to XOR the next segment of the value with. This technique results in the content of the AVP being obscured, although the length of the AVP is still known.

On receipt, the IV is taken from the last Initialization-Vector AVP encountered in the message prior to the AVP to be decrypted. The above process is then reversed to yield the original value. For more details on this hiding method, consult RFC2138 [1].

Please note that in the case where the DIAMETER message needs to be processed by an intermediate non-trusted DIAMETER server (also known as a proxy server, depicted as DIA2 in the figure below) the AVP needs to be decrypted using Shared-Secret-1 and re-encrypted by DIA2 using Shared-Secret-2.



Unfortunately in this case the non-trusted server DIA2 has access to sensitive information (such as a password).

5.2.2 AVP Encryption with Public Keys

AVP encryption using public keys is much more complex than the previously decribed method, yet it is desirable to use it in cases where the DIAMETER message will be processed by an untrusted intermediate node (proxy).

Public Key encryption SHOULD be supported, however it is permissible for a low powered device initiating the DIAMETER message to use shared secret encryption with the first hop (proxy) DIAMETER server, which would decrypt and encrypt using the Public Key method.

The PK-Encrypted-Data bit MUST only be set if the final DIAMETER host is aware of the sender's public key. This information can be relayed in three different methods as described in section 5.3.

The AVP is encrypted in the method described in [9].

5.3 Public Key Cryptography Support

A DIAMETER peer's public key is required in order to validate a message which includes the the Digital-Signature AVP. There are three possibilities on retrieving public keys:

5.3.1 X509-Certificate

A message which includes a Digital-Signature MAY include the X509-Certificate AVP. Given the size of a typical certificate, this is very wasteful and in most cases DIAMETER peers would cache such information in order to minimize per packet processing overhead.

It is however valid for a DIAMETER host to provide its X509-Certificate in certain cases, such as when issuing the Device-Reboot-Indication. It is envisioned that the peer would validate and cache the certificate at that time.

5.3.2 X509-Certificate-URL

The X509-Certificate-URL is a method for a DIAMETER host sending a message that includes the Digital-Signature to provide a pointer to its certificate.

Upon receiving such a message a DIAMETER host MAY choose to retrieve the certificate if it is not locally cached. Of course the process of retrieving and validating a certificate is lengthy and will require the initiator of the message to retransmit the request. However once cached the certificate can be used until it expires.

5.3.3 Static Public Key Configuration

Given that using certificates requires a PKI infrastructure which is very costly, it is also possible to use this technology by locally configuring DIAMETER peers public keys.

Note that in a network involving many DIAMETER proxies this may not scale well.

6.0 References

[1] Rigney, et alia, "RADIUS", RFC-2138, Livingston, April 1997
[2] Reynolds, J., and J. Postel, "Assigned Numbers", RFC 1700, USC/Information Sciences Institute, October 1994.
[3] Postel, J., "User Datagram Protocol", RFC 768, USC/Information Sciences Institute, August 1980.
[4] Rivest, R., and S. Dusse, "The MD5 Message-Digest Algorithm", MIT Laboratory for Computer Science and RSA Data Security, Inc., RFC 1321, April 1992.
[5] Kaufman, C., Perlman, R., and Speciner, M., "Network Security: Private Communications in a Public World", Prentice Hall, March 1995, ISBN 0-13-061466-1.
[6] Krawczyk, H., Bellare, M., Canetti, R., "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, January 1997.
[7] P. Calhoun, "DIAMETER User Authentication Extensions", draft-calhoun-diameter-authen-01.txt, March 1998.
[8] B. Aboba, M. Beadles, "Network Address Identifier", draft-ietf-roamops-nai-10.txt, February 1998.
[9] B. Kaliski, "PKCS #1: RSA Encryption Version 1.5", draft-hoffman-pkcs-rsa-encrypt-03.txt, October 1997.

7.0 Acknowledgements

The Authors would like to acknowledge the following people for their contribution in the development of the DIAMETER protocol:

Bernard Aboba, Jari Arkko, William Bulley, Daniel C. Fox, Lol Grant, Nancy Greene, Peter Heitman, Ryan Moats, Victor Muslin, Kenneth Peirce, Sumit Vakil, John R. Vollbrecht, Jeff Weisberg and Glen Zorn

8.0 Author's Address

Questions about this memo can be directed to:

Pat R. Calhoun
Technology Development
Sun Microsystems, Inc.
15 Network Circle
Menlo Park, California, 94025
USA

Phone: 1-847-548-9587
Fax: 1-650-786-6445
E-mail: pcalhoun@toast.net

Allan C. Rubens
Ascend Communications

Ann Arbor, MI 48105-1812
USA

Phone: 1-734-647-0417
E-Mail: acr@del.com