DIAMETER Proxy Server Extensions
<draft-calhoun-diameter-proxy-00.txt>

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working
documents of the Internet Engineering Task Force (IETF), its areas,
and its working groups. Note that other groups may also distribute
working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the
"1id-abstracts.txt" listing contained in the Internet-Drafts Shadow
Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe),
munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or
ftp.isi.edu (US West Coast).

Abstract

This DIAMETER Extension defines commands and AVPs that are used when
DIAMETER messages must be proxied by DIAMETER Servers. This extension
is intended to clearly define how proxying can be done with DIAMETER.
Table of Contents

1.0    Introduction
1.1    Definitions
1.2    Terminology
2.0    Command Codes
2.1    Domain-Discovery-Request (DDR)
2.2    Domain-Discovery-Answer (DDA)
3.0    DIAMETER AVPs
3.1    Proxy-State
3.2    Digital-Signature
3.3    X509-Certificate
3.4    X509-Certificate-URL
3.5    Next-Hop
4.0    Protocol Definition
4.1    Data Integrity
4.1.1  Using Digital Signatures
4.1.1  Using Mixed Data Integrity AVPs
4.2    AVP Data Encryption
4.2.1  AVP Encryption with Public Keys
4.3    Public Key Cryptography Support
4.3.1  X509-Certificate
4.3.2  X509-Certificate-URL
4.3.3  Static Public Key Configuration
5.0    References
6.0    Acknowledgements
7.0    Author’s Address

1.0    Introduction

Many services, including ROAMOPS and MobileIP, have a requirement for DIAMETER Server to proxy a request to another DIAMETER Server. The concept of proxying AAA requests was introduced by RADIUS and has been in use for many years.

Unfortunately due to the fact that RADIUS only supports hop-by-hop security, where each node has a security association with the next hop, this does introduce some security flaws. Specifically a fraudulent proxy server can modify some portions of an AAA request in order to make the next hop improperly believe that some services were rendered. For example, a DIAMETER Proxy Server could modify an accounting request, such as the number of bytes that a user transferred, and the end system would have no way of determining that this change occurred.

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 Command Codes

This document defines the following DIAMETER Commands. All DIAMETER implementations supporting this extension MUST support all of the following commands:

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Command Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain-Discovery-Request</td>
<td>261</td>
</tr>
<tr>
<td>Domain-Discovery-Answer</td>
<td>262</td>
</tr>
</tbody>
</table>

2.1 Domain-Discovery-Request (DDR)

Description

The Domain-Discovery-Request message is used by a DIAMETER device wishing to get contact information about a domain’s home authentication server as well as to receive password policy information. This message MUST contain the User-Name attribute in order to pass along the user’s domain information.

It is not necessary for an implementation to issue a DDR in order to make use of a proxy server.
The X509-Certificate or the X509-Certificate-URL [2] MUST be present in this message in order to inform the home authentication server of the issuing host’s certificate.

At least one Extension-Id AVP MUST be present in the DDR in order to inform the peer about the locally supported extensions.

Message Format

```
<Domain-Discovery-Req> ::= <DIAMETER Header>
    <Domain-Discovery-Req Command AVP>
    <Host-IP-Address AVP>
    [<Host-Name AVP>]
    <Extension-Id AVPs>
    <User-Name AVP>
    [<X509-Certificate AVP>]
    [<X509-Certificate-URL AVP>]
    <Timestamp AVP>
    <Initialization-Vector AVP>
    {<Integrity-Check-Vector AVP> ||
    <Digital-Signature AVP> }
```

AVP Format

A summary of the Domain-Discovery-Request packet format is shown below. The fields are transmitted from left to right.

```
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 Length           |     Reserved      |P|T|V|E|H|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Command Code                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AVP Code

```
256     DIAMETER Command
```

AVP Length

```
The length of this attribute MUST be 12.
```

AVP Flags

```
The 'M' bit MUST be set. The 'H' and 'E' MAY be set depending
```
upon the security model used. The 'V', 'T' and the 'P' bits MUST NOT be set.

Command Type

The Command Type field MUST be set to 261 (Domain-Discovery-Request).

2.2 Domain-Discovery-Answer (DDA)

Description

The Domain-Discovery-Answer message is sent in response to the Domain-Discovery-Request message by the domain’s Home authentication server. The message MUST contain either the Host-Name or Host-IP-Address and either the X509-Certificate or the X509-Certificate-URL attribute and SHOULD contain at least one Framed-Password-Policy AVP.

At least one Extension-Id AVP MUST be present in the DDA in order to inform the requestor about the locally supported extensions.

The Domain-Discovery-Answer message MUST include the Result-Code AVP to indicate whether the request was successful or not. The following Error Codes are defined for this command:

DIAMETER_ERROR_UNKNOWN_DOMAIN 1
This error code is used to indicate to the initiator of the request that the requested domain is unknown and cannot be resolved.

DIAMETER_ERROR_BAD_CERT 2
This error code is used to indicate that the X509-Certificate or the X509-Certificate-URL in the Domain-Discovery-Request was invalid.

DIAMETER_ERROR_CANNOT_REPLY 3
This error code is returned when either an intermediate DIAMETER node or the home authentication server cannot reply to DIAMETER messages directly. This could be that the policy of an intermediate DIAMETER server does not permit direct contact and therefore requires proxying. It could also signify that the home authentication server does not have public key support.

Message Format
<Domain-Discovery-Answer> ::= <DIAMETER Header>
<Domain-Disc-Answer Command AVP>
<Result-Code AVP>
[<Error-Code AVP>]
<Host-IP-Address AVP>
[<Host-Name AVP>]
<Extension-Id AVPs>
<Framed-Password-Policy AVP>
[<X509-Certificate AVP>]
[<X509-Certificate-URL AVP>]
<Timestamp AVP>
<Initialization-Vector AVP>
[<Integrity-Check-Vector AVP> ||
<Digital-Signature AVP>]

AVP Format

A summary of the Domain-Discovery-Answer packet format is shown below. The fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVP Code</td>
</tr>
<tr>
<td>AVP Length</td>
</tr>
<tr>
<td>Command Code</td>
</tr>
</tbody>
</table>

AVP Code

256 DIAMETER Command

AVP Length

The length of this attribute MUST be 12.

AVP Flags

The 'M' bit MUST be set. The 'H' and 'E' MAY be set depending upon the security model used. The 'V', 'T' and the 'P' bits MUST NOT be set.

Command Type

The Command Type field MUST be set to 262 (Domain-Discovery-Answer).
3.0 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:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy-State</td>
<td>33</td>
</tr>
<tr>
<td>Digital-Signature</td>
<td>260</td>
</tr>
<tr>
<td>X509-Certificate</td>
<td>264</td>
</tr>
<tr>
<td>X509-Certificate-URL</td>
<td>265</td>
</tr>
<tr>
<td>Next-Hop</td>
<td>278</td>
</tr>
</tbody>
</table>

3.1 Proxy-State

Description

The Proxy-State AVP is used by proxy servers when forwarding requests and contains opaque data that is used by the proxy to further process the response.

This attribute should be removed by the proxy server before the response is forwarded to the NAS, and SHOULD therefore not be protected by the Integrity-Check-Vector or the Digital-Signature.

AVP Format

A summary of the Proxy-State AVP format is shown below. The fields are transmitted from left to right.

```
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 Length |     Reserved       |P|T|V|E|H|M|  
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Data                       |     Data ...           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

AVP Code

33 for Proxy-State.
```
AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The 'M' bit MUST be set. The 'H' and 'E' MAY be set depending upon the security model used. The 'V', 'T' and the 'P' bits MUST NOT be set.

String

The String field is one or more octets. The actual format of the information is site or application specific, and a robust implementation SHOULD support the field as undistinguished octets.

3.2 Digital-Signature

Description

The Digital-Signature AVP is used for authentication, integrity as well as non-repudiation. A DIAMETER entity adding AVPs to a message MUST ensure that all AVPs appear prior to the Digital-Signature AVP (with the exception of the Integrity-Check-Vector AVP that MUST appear after the Digital-Signature AVP). The Timestamp AVP MUST be present to provide replay protection and the Initialization-Vector AVP must be present to add randomness to the packet.

The DIAMETER header as well as all AVPs with the 'P' bit disabled are protected by the Digital-Signature.

In order to support proxy DIAMETER servers, which forwards messages to next hop server, the proxy server MUST NOT modify any AVPs with the 'P' bit disabled. This ensures that end-to-end security is maintained even through proxy arrangements.

The Digital-Signature is generated in the method described in section 4.5.2.

All DIAMETER implementations supporting this extension MUST support this AVP.

AVP Format

A summary of the Digital-Signature AVP format is shown below. The
fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>AVP Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>260</td>
</tr>
</tbody>
</table>

AVP Length

The length of this attribute MUST be at least 17.

AVP Flags

The ‘M’ bit MUST be set. The ‘H’ MAY be set if the request is protected with an ICV AVP. The ‘E’, ‘V’, ‘T’ and the ‘P’ bits MUST NOT be set.

Address

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

Transform ID

The Transform ID field contains a value that identifies the transform that was used to compute the signature. The following values are defined in this document:

- **RSA** [9] 1

Data

The Data field contains the digital signature of the packet up to this AVP.
3.3 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 4.3.1 contains more information about the use of certificates.

AVP Format

A summary of the X509-Certificate AVP format is shown below. The fields are transmitted from left to right.

<p>| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 |
|-------------------------------------------------|--------------------------|</p>
<table>
<thead>
<tr>
<th>AVP Code</th>
<th>AVP Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>P</td>
</tr>
<tr>
<td>Data ...</td>
<td>--------------------------</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>X509-Certificate</td>
<td>--------------------------</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>264</td>
<td>9</td>
</tr>
</tbody>
</table>

AVP Code

264   X509-Certificate

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The 'M' bit MUST be set. The 'H' and 'E' MAY be set depending upon the security model used. The 'V', 'T' and the 'P' bits MUST NOT be set.

Data

The Data field contains the X.509 Certificate Chain.

3.4 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 4.3.1 contains more information about the use of certificates.

AVP Format

A summary of the X509-Certificate-URL AVP format is shown below. The fields are transmitted from left to right.

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+---------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>+---------------------------------------------</td>
</tr>
</tbody>
</table>

AVP Code

265 X509-Certificate-URL

AVP Length

The length of this attribute MUST be at least 9.

AVP Flags

The 'M' bit MUST be set. The 'H' and 'E' MAY be set depending upon the security model used. The 'V', 'T' and the 'P' bits MUST NOT be set.

String

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

3.5 Next-Hop

Description

The Next-Hop AVP MUST preceed a Digital-Signature AVP and is used to validate that a packet traversed the proxy chain that was intended. A DIAMETER message that includes a Host-IP-Address with
a different Next-Hop AVP that preceeds it is considered invalid.

AVP Format

A summary of the Next-Hop AVP format is shown below. The fields are transmitted from left to right.

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           AVP Code                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          AVP Length           |     Reserved      |P|T|V|E|H|M|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Address                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AVP Code

278    Next-Hop

AVP Length

The length of this attribute MUST be 12.

AVP Flags

The ‘M’ bit MUST be set. The ‘H’ and ‘E’ MAY be set depending upon the security model used. The ‘V’, ‘T’ and the ‘P’ bits MUST NOT be set.

Address

This field contains the IP Address of the next DIAMETER Server.

4.0 Protocol Definition

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

4.1 DIAMETER Proxying

The DIAMETER protocol also makes use of proxies in order to keep the existing arrangements while migrating from RADIUS to DIAMETER. However since DIAMETER proxying introduces asymmetric encryption and digital signatures it solves many of the problems found when using
In this example NASB generates a Request that is forwarded to DIA2. The Request contains a digital signature AVP which "protects" all mandatory (or non-editable) AVPs within the request. All AVPs which may be modified, or removed appear after the digital signature AVP. Once DIA2 receives the request, it MAY authenticate the request to ensure that it was originated by NASB (verifying the signature is not necessary if the link between NASB and DIA2 is secured using IPSEC).

The DIA2 Server SHOULD add the Proxy-State AVP, which contains opaque data that MUST be present in the response and is used to identify state information related to the request or response. The Server MAY also add other new AVPs to the request. All AVPs that are protected by the Digital-Signature MUST have the 'P' bit set, and all AVPs MUST precede the Digital-Signature AVP. The message is then forwarded towards the DIA1 server.

Since all packets between NASB and DIA1 must flow through DIA2, it is not possible to use IPSEC between both hosts. Therefore DIA1 MUST validate NASB’s digital signature AVP. However it is not necessary to validate DIA2’s digital signature if the link between DIA2 and DIA1 is secured using IPSEC.

This mechanism now provides a method for DIA1 to prove that NASB was the initiator of the request (note that DIAMETER also includes a timestamp to prevent replay attacks). It also provides a method of ensuring that DIA2 cannot modify any protected AVPs (such as length of call, etc.).

In addition, this same mechanism can be used for end-to-end encryption of AVPs. In the case where NASB needs to encrypt an AVP it is done using asymmetric encryption using DIA1’s public key. This ensures that only DIA1 can decrypt the AVP.

An attack has been identified in this proposal which allows a malicious man in the middle attack as shown in the following diagram.
In this example, DIA3 traps packets generated from DIA2 towards DIA1, removes the AVPs added by DIA2 and inserts its own AVPs (possibly by trying to convince DIA1 to pay DIA3 for the services). This attack can be prevented by supporting a new Next-Hop AVP. In this case when NASB prepares a request it inserts a non-editable Next-Hop AVP which contains DIA2’s identity. DIA2 also adds a Next-Hop AVP with DIA1 as the next hop.

This mechanism ensures that a man in the middle cannot alter the packet by overriding the previous hop’s additions and signature. DIA1 could easily validate the packet’s path with the use of the Next-Hop AVPs.

4.2 Domain Discovery

The Domain Discovery message set is very useful in determining the Home authentication server, the password policies for the domain, as a mechanism to retrieve a certificate (or a pointer to a certificate).

Note that it is not necessary for a host to issue a Domain Discovery in order to make use of a proxy. A DIAMETER Request MAY be proxied by an intermediate server without the knowledge of the client, however the client will be unable to validate any Digital-Signatures if the home authentication server’s certificate or public key is not known.

The following example shows a case where DIA1 needs to communicate with DIA3. In this example it is necessary to use DIA2 as a proxy in order for both ISPs to communicate. Although this MAY be desireable in some business models, there are cases where it is beneficial to remove the proxy altogether and allow both DIA3 and DIA1 to communicate in a secure fashion.
The way the Domain Discovery works is that prior to sending out an authentication request DIA1 would issue a Domain Discovery message towards DIA2. This message is protected with the digital signature as well as the Next-Hop AVP. DIA2 would then forward the request to DIA3 including the Next-Hop and the digital signature AVP.

When DIA3 receives the request, it MUST save the certificate (or the pointer to the certificate) and respond back including the local password policy, DIA3’s certificate, its contact information (i.e. IP address) and protect the response with the digital signature.

Note that in all cases the TimeStamp AVP is also present to ensure no replay packets are accepted.

When DIA2 receives the packet, it must add the Next-Hop AVP as well as the digital signature AVP. When DIA1 receives the packet it then knows a direct route to communicate with DIA3 since the contact information is present in the response. The fact that both DIA1 and DIA3 can now communicate directly allows both peers to use IPSEC to protect the message exchange (it may be desirable to use the Digital-Signature AVP in instances where records of digitally signed packets must be kept).

In addition, the password policy is also present which can indicate whether DIA3 is willing to accept CHAP, PAP or EAP authentication.

Note that the Domain-Discovery-Request/Answer MUST include at least one Extension-Id AVP [2].

4.3 Data Integrity

This section will describe how data integrity and non-repudiation is achieved using the Digital-Signature AVP.

Note that the Timestamp and Initialization-Vector AVPs MUST be present in the message PRIOR to the Digital-Signature AVPs discussed in this section. The Timestamp AVP provides replay protection and the Initialization-Vector AVP provides randomness.
Any AVPs in a message that is not followed by either the ICV or the Digital-Signature AVPs MUST be ignored.

### 4.3.1 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 protected 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 unprotected fields to zero (0) prior to calculating the signature. The two unprotected fields are the identifier and the length in the DIAMETER header.

The following is an example of a message that includes end-to-end security:

```
<DIAMETER Message> ::= <DIAMETER Header>
COMMAND AVP
[<Additional AVPs>]
NEXT-HOP AVP
<Timestamp AVP>
<Initialization-Vector AVP>
<Digital-Signature AVP>
```

The AVP Header’s ‘P’ bit is used to identify which AVPs are considered protected when applying a digital signature to a DIAMETER message. Protected AVPs cannot be changed "en route" since they are protected by the Digital Signature AVP. All AVPs added by a DIAMETER
entity MUST appear prior to the Digital Signature AVP that is added (with the exception of the Integrity-Check-Vector AVP). However, only AVPs with the 'P' bit set are used in the digital signature calculation.

The Next-Hop AVP indicates the intended recipient of the DIAMETER message. When a DIAMETER message is received with a Next-Hop AVP that does not correspond with the Host-IP-Address that follows, the message MUST be considered invalid and MUST be rejected.

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

4.3.2 Using Mixed Data Integrity AVPs

The previous sections described the Integrity-Check-Vector and the Digital-Signature AVP. Since the ICV offers hop-by-hop integrity and the digital signature offers end to end integrity, it is possible to use both AVPs within a single DIAMETER message.

The following diagram provides an example where DIAMETER Server 1 (DIA1) communicates with DIA3 using Digital-Signatures through DIA2. In this example ICVs are used between DIA1 and DIA2 as well as between DIA2 and DIA3.

Using the previous diagram, the following message would be sent between DIA1 and DIA2:

<DIAMETER Message> ::= <DIAMETER Header>
<Command AVP>
[<Additional AVPs>]
<Timestamp AVP>
<Initialization-Vector AVP>
<Digital-Signature AVP>
<Integrity-Check-Vector AVP (DIA1->DIA2)>

The following message would be sent between DIA2 and DIA3:
<DIAMETER Message> ::= <DIAMETER Header>  
<Command AVP>  
[<Additional AVPs>]  
<Timestamp AVP>  
<Initialization-Vector AVP>  
<Digital-Signature AVP>  
<Timestamp AVP>  
<Initialization-Vector AVP>  
<Integrity-Check-Vector AVP (DIA2->DIA3)>  

Note that in the above example messages the ICV AVP appear after the Digital-Signature AVP. This is necessary since DIA2 above removes the ICV AVP (DIA1->DIA2) and adds its own ICV AVP (DIA2->DIA3). The ICVs provide hop-by-hop security while the Digital-Signature provides integrity of the message between DIA1 and DIA3.

<Shared-Secret>    <Public-Key>  
|      |          |      |          |      |  
|router+----------+ DIA1 +----------+ DIA2 |  
|      |          |      |          |      |  
|------- <----- +------- <----- +-------|

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.

4.4 AVP Encryption with Public Keys

AVP encryption using public keys is much more complex than the previously described 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 4.3.
The AVP is encrypted in the method described in [9].

4.5 Public Key Cryptography Support

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

4.5.1 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.

4.5.2 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.

4.5.3 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.

5.0 References

6.0 Acknowledgements

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

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

7.0 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: 1-650-786-7733
Fax: 1-650-786-6445
E-mail: pcalhoun@eng.sun.com

William Bulley
Merit Network, Inc.