Secure Ad hoc On-Demand Distance Vector (SAODV) Routing
draft-guerrero-manet-saodv-06.txt

Intellectual Property Rights Statement

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Status of this Memo

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Copyright

Copyright (C) The Internet Society (2006). All Rights Reserved.

Abstract

The Secure Ad hoc On-Demand Distance Vector (SAODV) is an extension of the AODV routing protocol that can be used to protect the route discovery mechanism providing security features like integrity and authentication.
Table of Contents

1. Introduction ............................................. 3
2. Preliminary notes ........................................ 3
   2.1. Security Features .................................... 4
   2.2. Interaction with IPSec ................................ 4
3. Overview .................................................... 4
4. Terminology ................................................ 6
5. RREQ (Single) Signature Extension .......................... 6
6. RREP (Single) Signature Extension .......................... 8
7. RREQ Double Signature Extension ......................... 9
8. RREP Double Signature Extension ......................... 11
9. RERR Signature Extension .................................. 13
10. RREP-ACK Signature Extension ............................ 14
11. SAODV Operation ......................................... 14
   11.1. SAODV Signatures ..................................... 14
   11.2. SAODV Hash Chains ................................... 16
12. Adaptations to AODV that are needed .................... 17
13. Security Considerations .................................. 17
14. Modifications of the draft ............................... 18
15. Acknowledgments ......................................... 21
1. Introduction

In an ad hoc network, from the point of view of a routing protocol, there are two kinds of messages: the routing messages and the data messages. Both have a different nature and different security needs. Data messages are point-to-point and can be protected with any point-to-point security system (like IPSec). On the other hand, routing messages are sent to immediate neighbors, processed, possibly modified, and resent.

Another consequence of the nature of the transmission of routing messages is that, in many cases, there will be some parts of those messages that will change during their propagation. This is very common in Distance-Vector routing protocols, where the routing messages usually contain a hop count of the route they are requesting or providing. Therefore, in a routing message one could distinguish between two types of information: mutable and non-mutable. It is desired that the mutable information in a routing message is secured in such a way that no trust in intermediate nodes is needed. Otherwise, securing the mutable information will be much more expensive in computation, plus the overall security of the system will greatly decrease.

Moreover, as a result of the processing of the routing message, a node might modify its routing table. This creates the need for the intermediate nodes to be able to authenticate the information contained in the routing messages (a need that does not exist in point-to-point communications).

SAODV is an extension of the AODV[1] routing protocol that protects the route discovery mechanism providing security features like integrity and authentication. It uses digital signatures to authenticate the non-mutable fields of the messages, and hash chains to secure the hop count information (the only mutable information in the messages).

SAODV can use the Simple Ad hoc Key Management (SAKM)[2] as a key management system.

2. Preliminary notes

It is important to have in mind that this paper is describing how to protect the routing messages, not the data messages. This section contains some preliminary notes about which security features SAODV provides, and about IPSec interacting with SAODV.
2.1. Security Features

Before designing a protocol extension that provides security to AODV it is required to think what are the security needs and what issues just cannot be solved. The main thing that cannot be avoid is that there might be malicious nodes that do not respect protocols (they will forge AODV packets, listen to the others, reply packets in their own interests, report errors where there are none, etc).

It is needed to have integrity, authentication. But what about confidentiality? Well, maybe it is needed for scenarios with a very high security needs, but it does not make sense if the scenario is a public ad hoc network that everybody can joint at any moment. Therefore, it is not taken into account in the proposed protocol extension.

2.2. Interaction with IPSec

When trying to use IPSec to secure network transmissions in a MANET network, it is needed that the IPSec implementation can use as a selector the TCP or UDP port number. Sadly, there are quite many implementations that cannot do that. The importance of that is because it is needed that the IPSec policy will be able to apply certain security mechanisms to the data packets and just bypass the routing packets.

3. Overview

The solution presented in this paper is an extension of the AODV protocol mainly by using new extension messages. In these extension messages there is a signature of the AODV packet with the private key of the original sender of the Routing message (not of the intermediate nodes that just forward it).

Concerning to RREQ and RREP messages there are two alternatives: The first one in which only final destinations are allowed to reply a RREQ, and the second in which there is no such limitation.

In the first one, when a RREQ is sent, the sender signs the message. Intermediate nodes verify the signature before creating or updating a reverse route to that host. And only if the signature is fine they store the reverse route. The final destination node signs the RREP with its private key. Intermediate and final nodes, again verify the signature before creating or updating a route to that host, also storing the signature with the route entry.

In the second one, when a RREQ is sent, the sender signs the message. Intermediate nodes verify the signature before creating or updating a
reverse route to that host. And, again, only if the signature is fine
they store the reverse route. But the difference is that the RREQ
message has also a second signature that is always stored with the
reverse route. This second signature is needed to be added in the
gratuitous RREPs of that RREQ and in regular RREPs to future RREQs
that the node might reply as an intermediate nodes. An intermediate
node that wants to reply a RREQ needs not only the correct route, but
also the signature corresponding to that route to add it in the RREP
and the ‘Lifetime’ and the ‘Originator IP address’ fields that work
with that signature. If it has them, it generates the RREP, (adding
the stored signature, lifetime and the originator IP address) signs
the actual lifetime and the actual originator IP address and sends
it. All the nodes that receive the RREP and that update the route
store the signature the lifetime and the originator IP address with
that route.

If a node wants to be able to reply as an intermediate node for a
route to a node that has been added due to a RREQ or to a RREP, it
has to store the ‘RREQ Destination’ or ‘RREP Originator’ IP address,
the lifetime and the signature. And use them as the ‘Signature’, ‘Old
Lifetime’, and ‘Old Originator IP address’ fields in the RREP-DSE
message.

Hello messages are RREP messages, so they are signed in the same way.
Hello Interval extensions are not signed. There is no attack from
changing hello interval extension. Actually, if the hello interval
extension would be added in the signature, the nodes that received a
hello message from a node ‘D’ would not be able to reply as
intermediate node when a node ‘S’ would issue a RREQ for ‘D’, because
they wouldn’t have a valid signature for the RREP without the hello
interval extension.

Extension messages that include a second signature also include the
RREP fields (right now only the prefix size) that are not derivable
from the RREQ but not zeroed when computing the signature.

RREP-ACK messages may be authentified by using a digital signature,
that might be verified by any one that receives them.

Every node, generating or forwarding a RERR message, uses digital
signatures to sign the whole message and any neighbor that receives
verifies the signature.

The hop count of all these messages is authentified by using a hash
chain.
4. Terminology

This memo uses the conventional meanings [3] for the capitalized words MUST, SHOULD and MAY. It also uses terminology taken from the specifications of AODV and IPSec [4].

5. RREQ (Single) Signature Extension

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Hash Function</th>
<th>Max Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Top Hash</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Length</td>
<td>Hash Function</td>
<td>Max Hop Count</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Top Hash</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Length of Type-Specific Data, Not Including the Type and Length Fields of the Extension in Bytes

Hash Function
The hash function used to compute the Hash and Top Hash fields.

Max Hop Count
The Maximum Hop Count supported by the hop count authentication.

Top Hash
The top hash for the hop count authentication. This field has variable length, but it must be 32-bits aligned.
Signature Method
The signature method used to compute the signatures.

H            Half Identifier flag. If it is set to ’1’ indicates the use of HID and if it is set to ’0’ the use of FID.
Reserved     Sent as 0; ignored on reception.

Padding Length
Specifies the length of the padding field in 32-bit units. If the padding length field is set to zero, there will be no padding.

Public Key   The public key of the originator of the message. This field has variable length, but it must be 32-bits aligned.

Padding      Random padding. The size of this field is set in the Padding Length field.

Signature    The signature of all the fields in the AODV packet that are before this field but the Hop Count field. This field has variable length, but it must be 32-bits aligned.

Hash         The hash corresponding to the actual hop count. This field has variable length, but it must be 32-bits aligned.
6. RREP (Single) Signature Extension

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |     Type      |    Length     | Hash Function | Max Hop Count |
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |                           Top Hash                            |
| ...                                                         ... |
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |  Sign Method  |H|         Reserved            |  Padd Length  |
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |                          Public Key                           |
| ...                                                         ... |
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |                          Padding (optional)                       |
| ...                                                         ... |
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |                           Signature                           |
| ...                                                         ... |
| +---+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| |                             Hash                              |
| ...                                                         ... |

Type 65

Length The length of the type-specific data, not including the Type and Length fields of the extension in bytes.

Hash Function The hash function used to compute the Hash and Top Hash fields.

Max Hop Count The Maximum Hop Count supported by the hop count authentication.

Top Hash The top hash for the hop count authentication. This field has variable length, but it must be 32-bits aligned.

Signature Method ... Padding The same than in RREQ (Single) Signature Extension.

Signature The signature of the all the fields in the AODV packet that are before this field but the Hop Count field. This field has variable length, but it must be 32-bits
7. RREQ Double Signature Extension

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Hash Function</th>
<th>Max Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reserved</th>
<th>Prefix Size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Top Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sign Method</th>
<th>Padd Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Public Key</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Padding (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signature for RREP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Type 66

Length The length of the type-specific data, not including the Type and Length fields of the extension in bytes.

Hash Function The hash function used to compute the Hash and Top Hash fields.

Max Hop Count The Maximum Hop Count supported by the hop count authentication.
Reserved Sent as 0; ignored on reception.

Prefix Size The prefix size field for the RREP (it is 7 bit long to allow IPv6 prefixes).

Top Hash The top hash for the hop count authentication. This field has variable length, but it must be 32-bits aligned.

Signature Method ... Padding
The same than in RREQ (Single) Signature Extension.

Signature for RREP
The signature that should be put into the Signature field of the RREP Double Signature Extension when an intermediate node (that has previously received this RREQ and created a reverse route) wants to generate a RREP for a route to the source of this RREQ. This field has variable length, but it must be 32-bits aligned.

Signature
The signature of all the fields in the AODV packet that are before this field but the Hop Count field. This field has variable length, but it must be 32-bits aligned. Both signatures are generated by the requesting node.

Hash The hash corresponding to the actual hop count. This field has variable length, but it must be 32-bits aligned.
8. RREP Double Signature Extension

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Hash Function</th>
<th>Max Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top Hash</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Sign Method</td>
<td>H</td>
<td>Reserved</td>
<td>Padd Length</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Padding (optional)</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Sign Method 2</td>
<td>H</td>
<td>Reserved</td>
<td>Padd Length 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Padding 2 (optional)</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Signature of the new Lifetime and Originator IP address</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Old Lifetime</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Old Originator IP address</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Type 2</td>
<td>H</td>
<td>Reserved</td>
<td>Padd Length 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Padding 2 (optional)</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Signature of the new Lifetime and Originator IP address</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Type 67

Length The length of the type-specific data, not including the Type and Length fields of the extension in bytes.

Hash Function The hash function used to compute the Hash and Top Hash fields.
Max Hop Count
The Maximum Hop Count supported by the hop count authentication.

Top Hash
The top hash for the hop count authentication. This field has variable length, but it must be 32-bits aligned.

Signature Method ... Padding
The same than in RREQ (Single) Signature Extension.

Signature
The signature of all the fields of the AODV packet that are before this field but the Hop Count field, and with the Old Lifetime value instead of the Lifetime. This signature is the one that was generated by the originator of the RREQ-DSE). This field has variable length, but it must be 32-bits aligned.

Old Lifetime
The lifetime that was in the RREP generated by the originator of the RREQ-DSE).

Old Originator IP address
The Originator IP address that was in the RREP generated by the originator of the RREQ-DSE).

Signature Method 2 ... Padding 2
The whole block of fields is repeated. This time for the ‘Signature of the New Lifetime and Originator IP address’ signature.

Signature of the new Lifetime and Originator IP address
The signature of the RREP with the actual lifetime (the lifetime of the route in the intermediate node) and with the actual Originator IP address. This signature is generated by the intermediate node. This field has variable length, but it must be 32-bits aligned.

Hash
The hash corresponding to the actual hop count. This field has variable length, but it must be 32-bits aligned.
9. RERR Signature Extension

| Type | Length | Reserved |
+-----+--------+----------|
| 68   |        |          |

Length: The length of the type-specific data, not including the Type and Length fields of the extension in bytes.

Reserved: Sent as 0; ignored on reception.

Signature Method: The same as in RREQ (Single) Signature Extension.

Signature: The signature of all the fields in the AODV packet that are before this field. This field has variable length, but it must be 32-bits aligned.
10. RREP-ACK Signature Extension

```
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+-------------------+
|     Type      |    Length     |
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
|  Sign Method  |H|         Reserved            |  Padd Length  |
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
... |                          Public Key                           |
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
...                                                         ...
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
|                      Padding (optional)                       |
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
...                                                         ...
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
|                           Signature                           |
+----------------+-+-----------------+------------------+-+----------------+-+-------------------+
...                                                         ...
```

Type 69

Length The length of the type-specific data, not including the Type and Length fields of the extension in bytes.

Signature Method ... Padding
The same than in RREQ (Single) Signature Extension.

Signature The signature of all the fields in the AODV packet that are before this field. This field has variable length, but it must be 32-bits aligned.

11. SAODV Operation

This section describes how SAODV allows to authenticate the AODV routing data. Two mechanisms are used to achieve this: hash chains and signatures.

11.1. SAODV Signatures

When calculating signatures, Hop Count field is always zeroed, because it is a mutable field. In the case of the Signature for RREP field of the RREQ Double Signature Extension, what is signed is the future RREP message that nodes might send back in response to the RREQ. To construct this message it uses the values of the RREQ and the Prefix Size (the RREP field that is not derivable from the RREQ but not zeroed when computing the signature.

In the case of RREPs, R and A flags are also zeroed. SAODV is not
designed taking into account AODV multicast (‘R’ flag is used in multicast) and ‘A’ flag is mutable and, if an attacker alters it, it can only lead to some sort of denial of service.

Every time a node generates a RREQ it decides if it should be signed with a Single Signature Extension or with a Double Signature Extension. All implementations MUST support RREQ Single Signature Extension, and SHOULD support RREQ Double Signature Extension. A node that generates a RREQ with the gratuitous RREP flag set SHOULD sign the RREQ with a Double Signature Extension. A node SHOULD never generate a RREQ without adding a Signature Extension.

When a node receives a RREQ, first verify the signature before creating or updating a reverse route to that host. Only if the signature is verified, it will store the route. If the RREQ was received with a Double Signature Extension, then the node will also store the signature, the lifetime and the Destination IP address for the RREP in the route entry.

If a node receives a RREQ without a Signature Extension it SHOULD drop it.

An intermediate node will reply a RREQ with a RREP only if fulfills the AODV requirements to do so, and the node has the corresponding signature and the old lifetime and old originator IP address to put into the ‘Signature’, ‘Old Lifetime’ and ‘Old Originator IP address’ fields of the RREP Double Signature Extension. Otherwise, it will rebroadcast the RREQ.

When a RREQ is received by the destination itself, it will reply with a RREP only if fulfills the AODV requirements to do so. This RREP will be sent with a RREP Single Signature Extension.

All implementations MUST support RREP Single Signature Extension, and SHOULD support RREP Double Signature Extension. A node SHOULD never generate a RREP without adding a Signature Extension. This also applies to gratuitous RREPs.

When a node receives a RREP, first verifies the signature before creating or updating a route to that host. Only if the signature is verified, it will store the route with the signature and the lifetime and the originator IP address of the RREP. If a node receives a RREP without a Signature Extension it SHOULD drop it.

Every node, generating or forwarding a RERR message, uses digital signatures to sign the whole message and any neighbor that receives verifies the signature. In this way it can verify that the sender of the RERR message is really the one that claims to be. And, since
destination sequence numbers are not signed by the corresponding node, a node SHOULD never update any destination sequence number of its routing table based on a RERR message.

Although nodes will not trust destination sequence numbers in a RERR message, they will use them to decide whether they should invalidate a route or not.

RREP-ACK messages MAY be authenticated by using the RREP-ACK Signature Extension.

The block ’Signature Method ... Padding’ is included before the ‘Signature’ field in all the extension messages, and before the ‘Signature of the new Lifetime and Originator IP address’ field in the RREQ-DSE message.

SAKM specifies the list of possible values of the Signature Method field and how public keys and signatures are encoded in the extension messages.

11.2. SAODV Hash Chains

Hash chains are used in SAODV to authenticate the hop count of the AODV routing messages (not only by the end points, but by any node that receives one of those messages).

Every time a node wants to send a RREQ or a RREP it generates a random number (seed). Selects a Maximum Hop Count. Maximum Hop Count SHOULD be set to the TTL value in the IP header, and it SHOULD never exceed its configuration parameter NET_DIAMETER. The Hash field in the Signature Extension is set to the seed. The Top Hash field is set to the seed hashed Max Hop Count times.

Every time a node receives a RREQ or a RREP it verifies the hop count by hashing Max Hop Count - Hop Count times the Hash field, and checking that the resultant value is the same than the Top Hash. If the check fails, the node SHOULD drop the packet.

Before rebroadcasting a RREQ or forwarding a RREP, a node hashes one time the Hash field in the Signature Extension.

The function used to compute the hash is set in the Hash Function field. Since this field is signed, a forwarding node will only be able to use the same hash function that the originator of the routing message has selected. If an node cannot verify or forward a routing message because it does not support the hash function that has been used, then it drops the packet.
The list of possible values of the Hash Function field are the same as the one for the hash functions used for the signature ('Hash F Sign') that are specified in SAKM.

12. Adaptations to AODV that are needed

According to the AODV RFC, the originator of a RREQ can put (on purpose) a much more bigger destination sequence number than the real one. This allows a very easy attack that consists in setting the destination sequence number to 0xFFFFFFFF (the maximum value that fits in the 32-bits field). Then, the originator of the RREP and all the intermediate nodes will have that as sequence number for the route. The next time the node increments the sequence number, its sequence number counter will overflow. This might cause completely unexpected results, none of them good.

The fact that the originator of the RREQ can set the sequence number of the destination is because it is going to be needed if the destination node has rebooted (see section 6.13. ‘Actions After Reboot’ in the AODV RFC). After rebooting, a node does not remember its sequence number anymore and trusts anybody that sends to it a RREQ with the number. But this just cannot be allowed.

Therefore, all the AODV-enabled nodes SHOULD have a way to keep their destination sequence number even after rebooting. In addition, in the case that the destination sequence number in the RREQ is bigger than the destination sequence number of the destination node, the destination node SHOULD NOT take into account the value in the RREQ. Instead, it will realize that the originator of the RREQ is misbehaving and will send the RREP with the right sequence number. Also, intermediate nodes MUST not replace the destination sequence number in the RREQ.

Finally, and concerning to the AODV port (the UDP port used to send AODV messages), AODV nodes SHOULD never accept AODV messages sent from a different port than the standard one.

13. Security Considerations

The goal of the protocol extension described here, is to achieve that a node that plans to build an attack by not behaving according to the AODV routing protocol, will be only able to selectively don’t reply to certain routing messages and to lie about information about itself. Nevertheless, It does not do much to avoid denial-of-service attacks.

If a malicious node receives a packet and resends it after a while, it will not alter the network topology because of the sequence number
system.

It might seem that lifetime is not very strongly authenticated in the case that intermediate nodes are allowed to reply RREQs, because they could lie about the lifetime. Anyway, the goal of the protocol extension is achieved, because the node would be only lying about itself.

What about the originator IP address (also in the case that intermediate nodes are allowed to reply RREQs)? If an intermediate node lies about it, the RREP will travel to the fake originator IP address but the routes that will be generated by the nodes that will propagate the routing message will be correct. So the attack is practically equivalent to the one in which the intermediate node ignores the RREQ.

Using hash chains for authentifying hop counter has a problem: A malicious node forwarding a route might not increment the hop counter by using the same hash value. If it does so, the subsequent nodes will think that this route is one hop shorter (having more chances to be chosen as the route to use). This is not really a big threat, because to launch an attack, a group of malicious nodes should be close to the shortest path (each of the malicious nodes forwarding the routing messages would not increment the hop counter), and the less malicious nodes are, the more close they have to be to the shortest path. A path that is changing with the time.

14. Modifications of the draft

Version 6

- Qing Li, noticed that in the section about adaptations to AODV it was missing the fact that intermediate nodes cannot modify the destination sequence number of a RREQ.

Version 5

- The intro has been changed.
- RERR cannot use delayed verification.
- The key management part has been moved to draft-guerrero-manet-sakm-00.txt. And now is called ‘Simple Ad hoc Key Management (SAKM)’.

Version 4

- ‘A’ flag is not signed (as proposed by Francesco Dolcini). Neither is ‘R’ flag.
- **Section 14.4.** SAODV Key Management: IPv4 addresses can now be generated in a similar fashion than IPv6 ones.

- **Section 7.** RREQ Double Signature Extension: Prefix Size is now 7 bit long to be able to hold IPv6 Prefix Sizes.

Version 3

- Clarification: Now, in section ’3. Overview’, it explicitly says that Hello Interval extension is not signed.


- Clarification: Now all lengths specify if we are talking about bytes or 32-bit words.

- In section ’14.4. SAODV Key Management’, adds the list of what is used as PublicKey depending on which Signature Method is use.

- In section ’14.2. SAODV Hash Chains’, the list of hash functions has changed, and now includes more hash functions. Note that the hash functions that already existed in the previous version now have a different value.

Version 2

- Correction: In section ’14.1. SAODV Signatures’ instead of ”and the lifetime (that is REV_ROUTE_LIFE) and the Originator IP address for the RREP in the route entry” now it says “the lifetime and the Destination IP address for the RREP in the route entry.”. Thanks to Moritz Killat.

- Adds a bit more of explanation of what a node has to do if it wants to be able to reply as an intermediate node for a route that has been added due to a RREQ or to a RREP in the section ’3. Overview’.

- Correction: When an intermediate node generates a RREP, the ‘Originator IP Address’ of the AODV message with a RREP-DSE might be different than the one that was in the RREQ with a RREQ-DSE (so we have to add a field in the RREP-DSE for the old Originator IP Address just in the same way as we do with the lifetime). Thanks to Moritz Killat for noticing it.

- Correction: In RREQ-DSE ‘Signature’ should also sign the ‘Signature for RREP’ and, to make things clear the ‘Signature for RREP’ field goes before the ‘Signature’ field. I noticed this when discussing the
DSE mechanism with Moritz Killat.

- Correction: Hash functions must be MD5 and SHA1 (not HMACs). Thanks to Varaporn Pangboonyanon for noticing it.

- Correction: In the HMACs used to get the SAODV_HID and the SAODV_FID, the data to which the HMACs are going to be applied was missing (now it is PublicKey). So it is an HMAC of the public key with the public key as a key.

Version 1

- Adds this section. ;)

- Adds the following fields just before the ’Signature’ field in all the extension messages:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sign Method  |H|         Reserved            |  Padd Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Public Key                           |
...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Padding (optional)                       |
...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

- And adds these other fields just before the ’Signature of the new Lifetime’ field in the RREQ-DSE extension message:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sign Method 2 |H|         Reserved            | Padd Length 2 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Public Key 2                          |
...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Padding 2 (optional)                      |
...                                                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

- Adds the section "11. Duplicated Address (DADD) Detected Message".

- Adds the section "12. New Address (NADD) Notification Message".

- Adds the section "13. New Address Acknowledgment (NADD-ACK) Message".

- Adds some text at the end of the section "14.1. SAODV Signatures"
to explain the new fields of the extension messages.

- Adds the section "14.3. SAODV Delayed Verification of Signatures".

- Adds the section "14.4. SAODV Key Management".

- Removes the section "2.3. Key distribution".

- Other stuff I might be forgetting.

15. Acknowledgments

I want to thank all the people from the Nokia Research Center in Helsinki (where I worked for five years) that helped to make SAODV a reality. Special mention deserve my colleague N. Asokan and my bosses Jari Juopperi and Asko Vilavaara.

N. Asokan (from Nokia Research Center) has contributed to this draft with several improvements and corrections. He suggested the use of hash chains for authenticate the hop count and that intermediate nodes should sign the lifetime of the RREPs.

I also want to thank the following persons for their help and improvements to the draft: Sampo Sovio (from Nokia Research Center), Toni Barrera Arboix (while he was working for Nokia Research Center), Varaporn Pangboonyanon, Moritz Killat (from NEC Europe Ltd.) and Francesco Dolcini.

References


Author’s Address:

Questions about this memo can be directed to the author:

Manel Guerrero Zapata
Computer Architecture Department (DAC)
Technical University of Catalonia (UPC)
UPC-AC C6-123 Campus Nord
C. Jordi Girona 1-3
08034 Barcelona SPAIN
(+34) 93 4054044
guerrero@ac.upc.edu

Appendix A. Full Copyright Statement

Copyright (C) The Internet Society 2005. This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

(See RFC 3667 sections 5.4 and 5.5.)