Chapter 5

Simple Ad hoc Key Management

5.1 Introduction

One of the most important consequences of the nature of the MANET networks is that one cannot assume that a node that is part of a network will be always reachable by all the other nodes. This implies that there cannot be servers in the conventional meaning of the fixed networks. Therefore, the use of Certification Authorities (CAs) in MANET networks is not feasible.

The approach of distributing the Certification Authority functionality among ad hoc nodes (by dividing the private keys into shares) discussed in [52] implies a huge overhead, and it may be ineffective in a network were partitions occur or where there is high mobility. In addition, it will not work at all in trivial scenarios like when a network partition is composed of only two nodes.

Another characteristic of servers in fixed networks, besides its continuous availability, is the fact that clients have to know the server's IP address (or to know its human address and have the IP address of a DNS server). The same thing happens in MANET networks for any node you want to make a request or initiate an exchange of data.

However, current trends about addressing in ad hoc networks are driving

towards dynamic address allocation and autoconfiguration [47, 4]. In these schemes, typically a node picks a tentative address and checks if it is already in use by broadcasting a query. If no conflict is found, the node is allowed to use that address. If a conflict is found, the node is required to pick another tentative address and repeat the process.

But then, If IP addresses do not identify a node (because they are dynamically allocated), how does a node know the IP address of the node to which it wants to sent data. In fixed networks, if a node wants to send data to another one, it needs to know its address (it cannot send anything to a node that has a dynamic address, because it does not know its IP address).

The Binding between public keys and other attributes is typically achieved by using public key certificates. In some limited scenarios, a possible approach could be for a certification authority (that would live in a fixed network) to issue such certificates that the nodes could collect before going to the MANET "playground". However, this is not feasible for a big group of the targeted scenarios. An added problem is that the IP address should be one of the attributes binded to the public keys, because it is binded to your identity.

To sum up, what is required is a system that achieves that: IP addresses will be assigned dynamically, nodes will be identifiable by their IP addresses, there should be a binding between the public key and the IP address of a node, and all this without any kind of certification authorities. Which is quite a challenge.

A couple of papers [32, 33] have proposed a solution to solve the "address ownership" problem in the context of Mobile IP. It consists in to pick a key pair, and map the public key to a tentative address in some deterministic way.

The proposal of SAKM is to generate IP addresses in a similar way than in [32]. In that paper, they where using what they called SUCV (Statistically Unique and Cryptographically Verifiable) addresses. SUCV addresses where designed to protect Binding Updates in Mobile IPv6. SUCV addresses are generated by hashing an "imprint" and the public key. That imprint (that can be a random value) is used to limit certain attacks related to Mobile IP.

For MANET networks it is only needed to hash the public key. The hash digest (or a substring of it) maybe formated in some specific way (in order to be a valid IP address) will be a "Cryptographically Generated Address" (CGA) which will also be statistically unique. When a message that uses the CGA as source IP address and includes the public key of a node is signed by its private key, it can be verified by any other node that the node has a certain identity (represented by the knowledge of the secret key).

5.2 Generation of the IP address for SAODV

In SAODV, it is recommended to use IPv6 (instead of IPv4) due to its bigger address length (that would guaranty the statistically uniqueness of the IP addresses). The address can be, then, a network prefix of 64 bits with a 64 bit SAODV_HID (Half IDentifier) or a 128 bit SAODV_FID (Identifier). These two identifiers are generated almost in the same way than the sucvHID and the sucvID in SUCV (with the difference that they hash the public key instead of an imprint):

$$SAODV\_HID = SHA1HMAC\_64(PublicKey, PublicKey)$$

$$SAODV\_FID = SHA1HMAC\_128(PublicKey, PublicKey)$$

There will be a flag in the SAODV routing message extensions (the 'H' flag) that will be set to '1' if the IP address is a HID and to '0' if it is a FID.

Finally, if it has to be a real IPv6 address, there is a couple of things that should be done [19].

If HID is used, then the HID behaves as an interface identifier and, therefore, its sixth bit (the universal/local bit) should be set to zero (0) to indicate local scope (because the IP address is not guaranteed to be globally unique).

And, if FID is used, then a format prefix corresponding to the MANET network should be overwritten to the FID. Format prefixes '010' through '110' are unassigned and would take only three bits of the FID. Format prefixes '1110' through '1111 1110 0' are also unassigned and they would take between 4 and 9 bits of the FID. All of these format prefixes required to have to have

64-bit interface identifiers in EUI-64 format, so universal/local bit should be set to zero (0).

The length of an IPv4 address is probably too short to provide the statistically uniqueness that this scheme requires when the number of nodes is very big. Nevertheless, if the number of nodes is assumed to be low enough, (let's say, under 100 nodes) it is not very unrealistic to expect that the statistically uniqueness property will hold.

The SAODV IPv4 address will have a network preffix of 8 bits and a SAODV_4ID (IPv4 Identifier). The network preffix can be any number between 1 and 126 (both included) with the exception of 14, 24 and 39 [22]. The network preffix 10 can only be used if it is granted that it will not be connected to any other network [41].

The SAODV_4ID will be the first bits of the SAODV_HID and the 'H' flag will be set.

5.3 New fields in the SAODV messages

The public key should be included in the routing messages that are signed, so that the nodes can verify the signature. Since, obviously, that public key should be signed by the signature, it is placed before the signature field.

The identifier of the algorithm that is used to sign the message is specified in the Signature_Method field. The possible values are shown in Table 5.1 (being mandatory to support RSA). Since SAODV could allow more than one possible signature method, it might happen that a node has to verify a signature with a method it does not know. If this happens the node will consider that the verification of the signature has failed.

This implies that all the nodes that form part of a MANET network should know all the methods used by all the other nodes to sign their messages. This is not a problem since, typically, all nodes of a MANET network will use the same method (or two different methods the most). The fact that there is more than one possible signature methods is because different networks may have tighter security requirements than some others and, therefore, use different signature methods.

Value	Signature method
0	Reserved
1	RSA [42]
2	DSA [48]
3	Elliptic curve [26]
4-127	Reserved
128-255	Implementation dependent

Table 5.1: Possible values of the Signature Method field

5.4 Duplicated IP Address Detection

If a node 'A' receives a routing message that is signed by a node 'B' that has the same IP address than one of the nodes for which 'A' has a route entry (node 'C'), it will not process normally that routing message. Instead, it will inform 'B' that it is using a duplicated IP and it will prove it by adding the public key of 'C' (so 'B' can verify the truthfulness of the claim).

When the node 'B' receives a routing message that indicates that somebody else has the same IP address than itself (or it realizes about it by itself), it will have to generate a new pair of public/private keys. After that, it will derive its IP address from its public key and it might inform all the other nodes (through a broadcast) of which is its new IP address with an special message that contains: the two IP addresses (the old and the new ones) and the two public signatures (old and new) signed with the old private key and, all this, signed with the new private key. Nevertheless, it is much better if, that message, is unicast (instead of broadcast) to all the nodes it considers that should receive this information (in the case they are just a few). This unicast will be answered with an acknowledge message by the receiver if it verifies that everything is in order.

After this, the node will generate a route error message for his old IP address. Its propagation will delete the route entries for the old IP address and, therefore, eliminate the duplicated addresses. This route error message may have a message extension that tells which is the new address. In this way, the nodes that receive the routing message can already create the route to the new IP address.

This solution allows two nodes to coexist in the same network with the same IP address until one of them realizes about it. However, in the author's opinion, it gives a good trade-off between the impact of changing address (and having a coexisting period of two nodes with the same IP address) and the extremely low probability of having address collision.

Intermediate nodes could decide to store the IP addresses and public keys of all the nodes they would meet (or of the last 'N' nodes, depending on their capabilities). That would allow an earlier detection of duplicated IP addresses in the network.

An alternative to this solution could be that, when a node detects that another node is using the same IP address, it would keep its public/private key pair and change the used IP address by applying a salt to the algorithm that derives the IP address from the public key. Salt variations of hash algorithms have been used in order to avoid dictionary attacks of passwords [31]. The "salt" is a random string that is added to the password before being hashed. This idea can be adapted with a very different purpose. If the statistically unique IP address is the derived from the public key and a salt (instead of only from the public key), the node that detects or is informed that its IP address is also used by another node can change its IP address without change its public key by just changing the salt.

Nevertheless, that would imply that the salt used by a node should be included in all the routing messages and stored in all the entries of the routing tables. And, still, the node has to inform the others of its change of IP address. Therefore, it will not be used for the purpose of SAKM.

In conclusion, the approach described in here handles properly the very unlikely situation of two nodes with the same IP address, without adding any complexity to the typical situation.

5.4.1 Duplicated IP Address Detection for SAODV

SAODV can deal with the duplicated IP address problem as described before. Duplicate Address (DADD) Detected message is send to notify to a node that its address is already being used by another node. New Address (NADD)

Notification Message is used to inform that the node has change key pair and IP address. Finally, New Address Acknowledgment (NADD-ACK) Message is used to confirm the reception of the NADD. In SAODV, NADD is always unicast (never broadcast).

5.4.2 Network Leaders

The original SAODV design established that besides how key distribution is achieved, when distributing a public key, this should be binded to the identity of the node (of course) and also to its netmask (in the case the node is a network leader). This was to prevent the type attack in which a malicious node becomes a black hole for a whole subnet by claiming that it is their network leader.

In the new approach presented in here, ad hoc nodes will typically never be network leaders. Network leaders will be only fixed nodes that typically give access to the fixed network and the nodes in the MANET network should know their IP addresses, prefix size and public keys.

Network leaders will not change its IP address in case that there is a MANET node that happen to generate the same IP address. A node generating its IP address will check if the resulting IP address corresponds to the network leader or to the subnet corresponding to its prefix size. A node detecting another node using the network leader IP address or any of the ones corresponding to the leader subnet will inform to the MANET node, and not to the network leader.

5.5 Delayed Verification of Signatures

As stated in the introduction, there has been some concern (e.g., [34], [20], [21]) that SAODV's signatures might require a processing power that might be excessive for certain kinds of ad hoc scenarios. Delayed verification addresses this problem by revising one of SAODV's security requirements from the list that was stated in [15].

5.5.1 Revised Security Requirements

The security requirements that will be provided are source authentication and integrity (that combined provide data authentication) and delayed import authorization.

Import authorization was defined in [15] as:

- **Import authorization**: The ultimate authority about routing messages regarding a certain destination node is that node itself. Therefore, a node will only authorize route information in its routing table if that route information concerns the node that is sending the information. In this way, if a malicious node lies about it, the only thing it will cause is that others will not be able to route packets to the malicious node.

Delayed import authorization allows to have route entries and route entry deletions in the routing table that are pending of verification. They will be verified whenever the node has spared processor time or before these entries should be used to forward data packages.

The security requirements will not include confidentiality and non-repudiation because they are not necessarily critical services in the context of routing [18]. They will not include either availability (since an attacker can focus on the physical layer without bothering to study the routing protocol) and they will not address the problem of compromised nodes (since it is arguably not critical in non military scenarios).

5.5.2 Achieving Delayed Import Authorization

In reactive ad hoc routing protocols, most of the routing messages that circulate in the network are (by far) route requests. This is due to the fact that route requests are broadcast. Route replies are unicast back through the selected path. And, route error messages are unicast down through the tree of nodes that had a route to the now unreachable node that is advertised by the route error message.

When a node receives a routing message, it creates a new entry in its routing table (the so called “reverse route”). Therefore, after the broadcast

of the route request, all the nodes in the network (or in the broadcast ring) have created reverse routes to the originator of the route request. From all these reverse routes, most of them will expire soon (typically all but the ones that are in the selected path through which the route reply will travel).

Then, the question is: why should all this route requests be verified (with the consequent delay in the propagation of the broadcast), when most of them are going to be soon discarded. The answer is: there is no need to verify them until the corresponding route reply comes back and the node knows that it is in the selected path. The other reverse routes will expire without being verified.

Actually, the two signatures (the ones from the route request and route reply) will be verified after the node has forwarded the route reply. In this way transmissions of the route requests and replies occur without any kind of delay due to the verification of the signatures.

Following the same idea, the signature of route error messages (and in general, any routing message that has to be forwarded) can also be verified after forwarding them.

Routes pending of verification will not be used to forward any packet. If a packet arrives for a node for which there is a route pending of verification, the node will have to verify it before using that route. If the verification fails, it will delete the route and request a new one.

5.5.3 SAODV with Delayed Verification

When a node needs to send or to forward a packet to a destination for which it does not have an active route, first it will check if it has a route pending of validation. If it does, it will try to validate it and, if it was successfully validated, it will mark it as active and use it. If after all this there is not an active route the node will start a route discovery process.

As shown in figure 5.1, only once the validation is done successfully, the route is incorporated in the routing table of the node. That avoids doing dirty hacks into the routing table of the operating system of the node: The packets can be routed normally, and only when there is a route lookup that

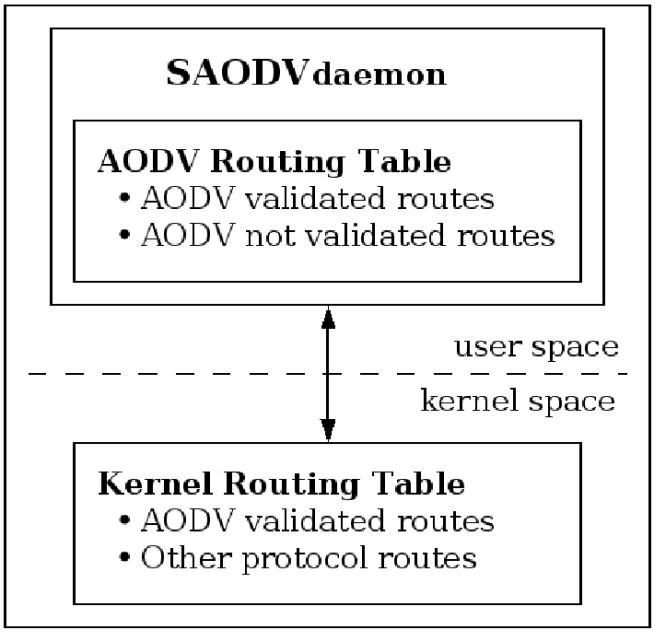

Figure 5.1: SAODV daemon

the routing table cannot resolve, the petition is captured by the SAODV routing daemon.

Figure 5.2 shows that in the case where there is a routing middleware (like zebra[1] or quagga[2]), the middleware routing table will contain the validated routes from the SAODV daemon combined with the ones from the other routing daemons and the routing table in the kernel the ones with lowest "administrative distance" (in case there is a route to the same destination provided by two different routing daemons).

Talking about administrative distances, none of the MANET routing protocols that are being designed or standardized have specified which would be the appropriate administrative distance for them. Let us look to the "standard de facto" (Cisco, Zebra, etc.) default administrative distance values. Probably a good default distance value would be between 160 (Cisco's On-Demand Routing) and 170 (external routes in EIGRP). Therefore, a default distance value of 165 for SAODV (and also for AODV in general) would be

[1] www.zebra.org
[2] www.quagga.net

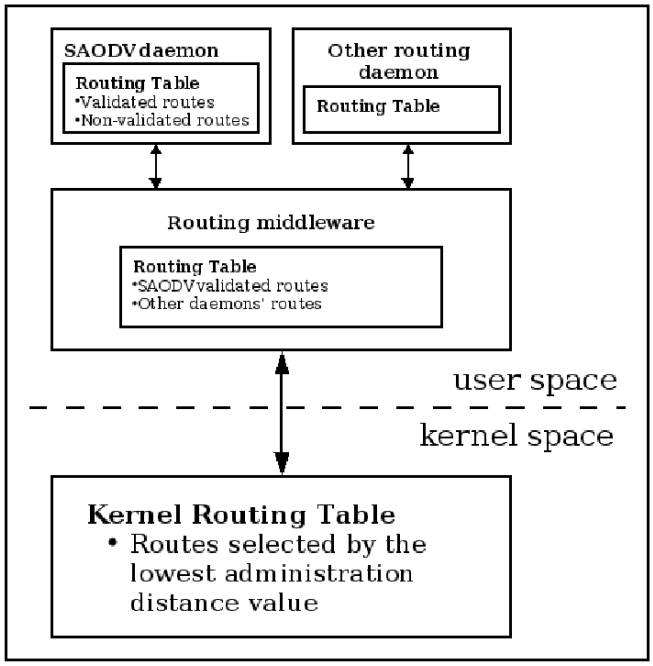

Figure 5.2: SAODV daemon with a routing middleware

appropriate.

5.6 Simulation Results

The purpose of using SAODV with delayed verification is to obtain the same level of security than with the original SAODV but without its main drawbacks. These drawbacks are a quite bigger average end to end delay and a higher power consumption by the nodes (when compared with AODV). These drawbacks are due to the computation of asymmetric cryptography primitives (message signature and verification). Through the use of simulations it was shown that delayed verification actually achieves this.

The simulations were done with 30 nodes moving at a maximum speed of 10 meters per second in a square of 1000x1000 meters. They simulated the establishment of 10 connections that started between second 0 and second 25 (according to an uniform distribution) and ended at the end of the simulation. The simulation time was of 100 seconds, and the connections where constant

bit rate (a packet of 512 each 0.25 seconds).

The nodes in the simulations have used as routing protocols: plain AODV, SAODV with RSA, SAODV with ECC (Elliptic Curve Cryptography), and SAODV with delayed verification (SAODV2 in the figure) with ECC. There is no point to use delayed verification with RSA since its verification time is completely negligible (delayed verification reduces the amount of verifications that have to be done). That means that SAODV with RSA with or without delay verification will give practically identical results. RSA, DSA and ECC have been used with key lengths that provide equivalent security (1368 bit for RSA and DSA, and 160 bit for ECC).

	RSA	DSA	ECC
Key length	1368	1368	160
Sign	210	90	42
Verify	6	110	160

Table 5.2: Times for a Compaq iPAQ 3670

Table 5.2 shows the times for signing/verifying in a Compaq iPAQ 3670 (206Mhz, 16M ROM, 64M RAM) according to [49]. DSA is not used in the simulations as it presents the worst of RSA and ECC (slow signature and verification, and fast increase of computational overhead as the key length needs to be bigger).

In the simulations, end to end delay of the packets, packet delivery fraction, and normalized routing load where measured. Figure 5.3 shows the averaged result of the end to end delay in data packet transmission. There were practically no differences among the routing protocols in packet delivery fraction (that was around 90 percent) and in normalized routing load (that was around 1).

One could expect quite different results with some other simulation scenarios, but almost always having SAODV with delayed verification and ECC as the best of the SAODV options and with a performance very close to plain AODV.

One could argue that, in scenarios in where the routes have more hops, the results of SAODV with delayed verification will be quite worst. But, actually,

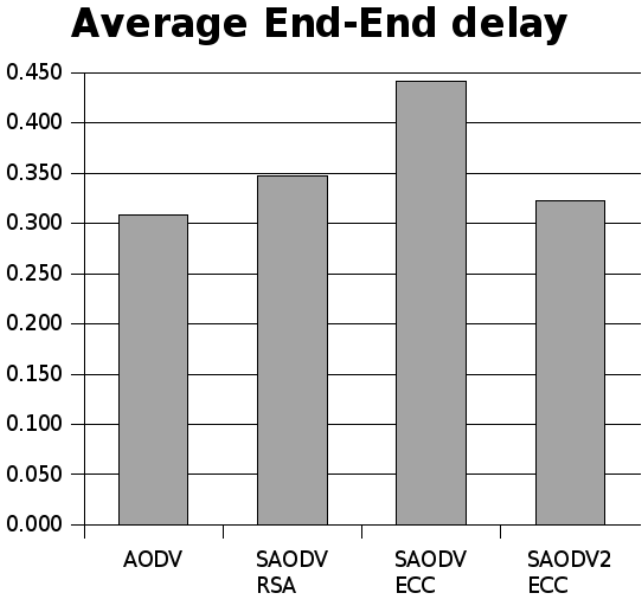

Figure 5.3: Simulation Results
The delay is measured in milliseconds

the results do not depend that much on the number of hops. This is due to the fact that intermediate nodes forward the Route Reply before verifying the signatures of the Route Request (RREQ) and Route Reply (RREP). Therefore, it is most probable that by the time the node that forwards the RREP to the final destination verifies the signatures of the RREQ and RREP, all the nodes of the route will also have verified them.

In the future, when longer keys are needed, ECC results will look even better than with the key lengths used in these simulations. This is due to the fact that, as they key size increases the computational overhead of ECC increases in a much more slowly manner than for RSA.

Therefore, these simulations have shown that SAODV used with delayed verification and ECC performs best that the other combinations with SAODV and that the performance penalty that introduces is almost negligible.

5.7 Analysis

Although it is true that there is no way to preclude a node of inventing many identities, that cannot be used to create an attack against the secure routing algorithm. An attacker cannot supplant another node, and a node can always prove that it is the same node.

Delayed verification makes possible that a malicious node creates invalid route requests that could flood the MANET network. But, the same malicious node can flood the network with perfectly valid route requests. And there would be no easy way to know if it is trying to flood the network or if it is just trying to see if any of its friend nodes are present in the network (for instance).

As explained before, an attacker cannot forge a public/private key pair from an IP address so the identity token becomes the IP address itself. Users of nodes might have a mechanism outside the MANET network in order to bind their public key to their physical identity.

With the current technology, SAODV with delayed verification and ECC provides security features to AODV with an almost negligible performance penalty.

In the future, when longer keys are required, the gain of using delayed verification in conjunction to ECC compared to other SAODV options will be even bigger that it is nowadays. This is due to the fact that as key length gets bigger the cost of signing/verifying in RSA and other cryptoalgorithms increases exponentially as in ECC (for the equivalent key length) it increases in a logarithmic way.

5.8 SAODV Duplicate Address Detection Messages

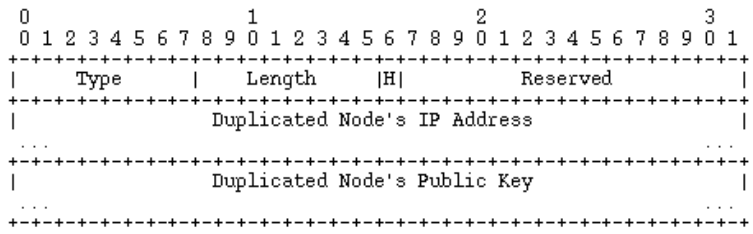

Figure 5.4: Duplicated Address (DADD) Detected Message

Field	Value
Type	64
Length	The length of the type-specific data, not including the Type and Length fields of the message.
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.
Duplicated Node's IP Address	The IP Address of the node that uses a Duplicated IP Address.
Duplicated Node's Public Key	The Public Key of the node that uses a Duplicated IP Address.

Table 5.3: Duplicated Address (DADD) Detected Message Fields

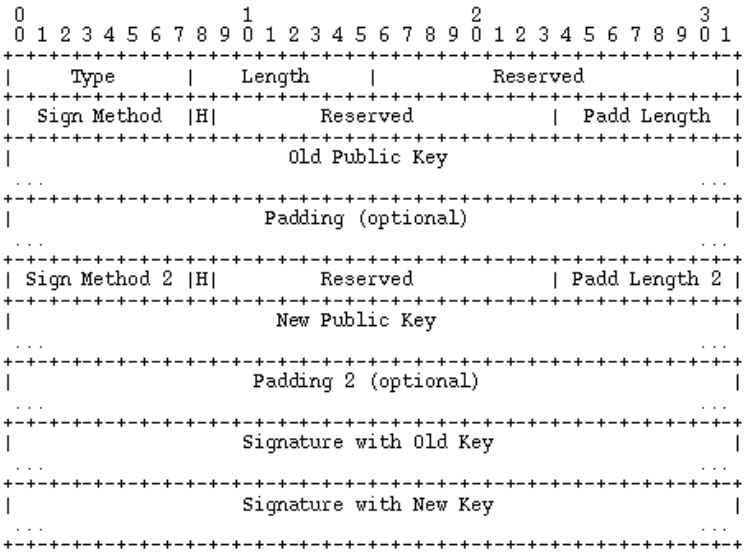

Figure 5.5: New Address (NADD) Notification Message

Field	Value
Type	65
Length	The length of the type-specific data, not including the Type and Length fields of the message.
Reserved	Sent as 0; ignored on reception.
Signature Method ... Padding	The same than in the Message Extensions. Corresponds to the 'Signature with Old Public Key' signature.
Signature Method 2 ... Padding 2	The whole block of fields is repeated. Corresponds to the 'Signature of the New Public Key' signature.
Signature with Old Key	The signature (with the old key) of the all the fields in the AODV packet that are before this field.
Signature with New Key	The signature (with the new key) of the all the fields in the AODV packet that are before this field.

Table 5.4: New Address (NADD) Notification Message Fields

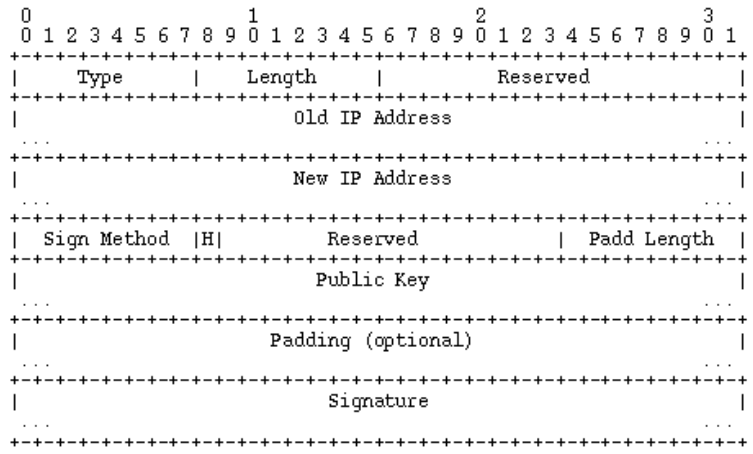

Figure 5.6: New Address Acknowledgment (NADD-ACK) Message

Field	Value
Type	66
Length	The length of the type-specific data, not including the Type and Length fields of the message.
Reserved	Sent as 0; ignored on reception.
Old IP Address	The old IP address.
New IP Address	The new IP address.
Signature Method ... Padding	The same than in the Message Extensions.
Signature	The signature of the all the fields in the AODV packet that are before this field.

Table 5.5: New Address Acknowledgment (NADD-ACK) Message Fields