Key Management and Delayed Verification for Ad hoc Networks

Manel Guerrero Zapata
Technology Department
Universitat Pompeu Fabra
Passeig de Circumval·lació 8, 08003 Barcelona
Email: manel.guerrero@upf.edu

***Abstract*— MANET (mobile and ad hoc networks) are networks in which nodes are mobile and link connectivity might change all the time. In this kind of networks key management is an important and complex problem.**

This paper studies how to design key management schemes for such networks that will allow to identify nodes without the need of any kind of certification authority. In addition, it presents a method to reduce the delays in route establishment in cases where routing messages are signed and need to be verified. Finally, it applies all these to SAODV (an extension of the AODV MANET routing protocol that protects the route discovery mechanism providing security features like integrity and authentication), and presents results from simulations that show how this method provides the same security with minimum impact in the network performance. Therefore, providing a more complete solution to the problem of security in MANET networks.

I. 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 an 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 [1] 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). The use of digital signatures (asymmetric cryptography) has generated some concern (e.g., [2], [3], [4]) that SAODV's signatures might require a processing power that might be excessive for certain kinds of ad hoc scenarios and that not providing a key management scheme that explains how nodes get the public keys they require it does not solve the whole problem.

This paper studies both problems and provides a general solution and a specific method for SAODV. Section II takes a look at related work. Section III considers different ways to achieve the key management in MANET networks. Section IV provides a method that reduces the required processing power due to the use of asymmetric cryptography. Section V gives an overview of AODV. Section VI describes the security mechanism to protect AODV's routing information: Secure AODV (SAODV) [1]. Section VII focuses on how the key management methods explained in this paper can be used in conjunction with SAODV. Finally, section VIII presents simulation results of using SAODV with delayed verification.

II. Related Work

There is very little published prior work on the security issues in ad hoc network routing protocols. Neither the survey by Ramanathan and Steenstrup [5] nor the survey by Royer and Toh [6] mention security. None of the proposals in the IETF *MANET* working group have a non-trivial "security considerations" section. Actually, most of them assume that all the nodes in the network are friendly, and a few declare the problem out-of-scope by assuming some canned solution like IPSec may be applicable.

In their paper on securing ad hoc networks [7], Zhou and Haas primarily discuss key management (key management is discussed in Section III). They devote a section to secure routing, but essentially conclude that "nodes can protect routing information in the same way they protect data traffic". They also observe that denial-of-service attacks against routing will be treated as damage and routed around.

Security issues with routing in general have been addressed by several researchers (e.g., [8], [9]). And, lately, some work has been done to secure ad hoc networks by using misbehavior detection schemes (e.g., [10]). This approach has two main problems: first, it is quite likely that it will be not feasible to detect several kinds of misbehaving (especially because it is very hard to distinguish misbehaving from transmission failures and other kind of failures); and second, it has no real means to guarantee the integrity and authentication of the routing messages.

Dahill et al. [11] proposed ARAN, a routing protocol for ad hoc networks that uses authentication and requires the use of a trusted certificate server. In ARAN, every node that forwards a route discovery or a route reply message must also sign it, (which is very computing power consuming and causes the size of the routing messages to increase at each hop), whereas the proposal presented in this paper only require originators to sign the message. In addition, it is prone to reply attacks using error messages unless the nodes have time synchronization.

Papadimitratos and Haas [2] proposed a protocol (SRP) that can be applied to several existing routing protocols (in particular DSR [12] and IERP [13]). SRP requires that, for every route discovery, source and destination must have a security association between them. Furthermore, the paper does not even mention route error messages. Therefore, they are not protected, and any malicious node can just forge error messages with other nodes as source.

Hash chains have being used as an efficient way to obtain authentication in several approaches that tried to secure routing protocols. In [9], [14] and [15] they use them in order to provide delayed key disclosure. While, in [16], hash chains are used to create one-time signatures that can be verified immediately. The main drawback of all the above approaches is that all of them require clock synchronization.

We suggested the use of hash chains to authenticate hop counts [17], [1]. This technique is used in SAODV. In SEAD [3] (by Hu, Johnson and Perrig) hash chains are also used in combination with DSDV-SQ [18] in a very similar way (this time to authenticate both hop counts and sequence numbers). At every given time each node has its own hash chain. The hash chain is divided into segments, elements in a segment are used to secure hop counts in a similar way as it is done in SAODV. The size of the hash chain is determined when it is generated. After using all the elements of the hash chain a new one must be computed.

SEAD can be, in theory, used with any suitable authentication and key distribution scheme. But finding such a scheme is not straightforward.

Ariadne [4], by the same authors, is based on DSR [12]. The authentication mechanism of Ariadne is based on TESLA [19]. It also requires clock synchronization. Clock synchronization introduces a big overhead in the network due to the messages needed to be exchanged to achieve it. Therefore, it is arguably not appropriate for MANET protocols.

It is quite likely that, for a small team of nodes that trust each other and that want to create an ad hoc network where the messages are only routed by members of the team, the simplest way to keep secret their communications is to encrypt all messages (routing and data) with a “team key”. Every member of the team would know the key and, therefore, it would be able to encrypt and decrypt every single packet. Nevertheless, this does not scale well and the members of the team have to trust each other. So it can be only used for a subset of the possible scenarios.

This is why SAODV uses asymmetric cryptography. But then, the challenge is to design a key management scheme that works in a mobile and ad hoc network where you cannot assume network connectivity with any kind of server.

Solving this challenge is the one of the aims of this paper.

III. Key Management in MANET Networks

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 [7] 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 [20], [21]. 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 [22], [23] 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. Our earlier paper [1] already proposed that this approach of "cryptographically generated addresses" could be used in the key management for SAODV. In this paper, we describe the details of CGA-based key management.

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 [24]. 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 this paper.

In conclusion, the approach described in this section handles properly the very unlikely situation of two nodes with the same IP address, without adding any complexity to the typical situation. Next section, explains how to reduce the number of verification of signatures which reduces importantly the computer power required by a node to run SAODV.

IV. Delayed Verification of Signatures

As stated in the introduction, there has been some concern (e.g., [2], [3], [4]) that SAODV's signatures might require a processing power that might be excessive for certain kinds of ad hoc scenarios. This section addresses this problem by revising one of SAODV's security requirements from the list that was stated in [1].

A. 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 [1] 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 [9]. 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).

B. How does it work?

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.

V. AODV

This section gives an introduction to AODV, necessary to understand how it is secured and how the key management technique is applied to it.

Ad Hoc On-Demand Vector Routing (AODV) protocol [25] is a reactive routing protocol for ad hoc and mobile networks that maintains routes only between nodes which need to communicate. The routing messages do not contain information about the whole route path, but only about the source and the destination. Therefore, routing messages do not have an increasing size. It uses destination sequence numbers to specify how fresh a route is (in relation to another), which is used to grant loop freedom.

Whenever a node needs to send a packet to a destination for which it has no 'fresh enough' route (i.e., a valid route entry for the destination whose associated sequence number is at least as great as the ones contained in any RREQ that the node has received for that destination) it broadcasts a route request (RREQ) message to its neighbors. Each node that receives the broadcast sets up a reverse route towards the originator of the RREQ (unless it has a 'fresher' one). When the intended destination (or an intermediate node that has a 'fresh enough' route to the destination) receives the RREQ, it replies by sending a Route Reply (RREP). It is important to note that the only mutable information in a RREQ and in a RREP is the hop count (which is being monotonically increased at each hop). The RREP travels back to the originator of the RREQ (this time as a unicast). At each intermediate node, a route to the destination is set (again, unless the node has a 'fresher' route than the one specified in the RREP). In the case that the RREQ is replied to by an intermediate node (and if the RREQ had set this option), the intermediate node also sends a RREP to the destination. In this way, it can be granted that the route path is being set up bidirectionally. In the case that a node receives a new route (by a RREQ or by a RREP) and the node already has a route 'as fresh' as the received one, the shortest one will be updated.

If there is a subnet (a collection of nodes that are identified by a common network prefix) that does not use AODV as its routing protocol and wants to be able to exchange information with an AODV network, one of the nodes of the subnet can be selected as their 'network leader'. The network leader is the only node of the subnet that sends, forwards and processes AODV routing messages. In every RREP that the leader issues, it sets the prefix size of the subnet.

In addition to these routing messages, Route Error (RERR) messages are used to notify the other nodes that certain nodes are not anymore reachable due to a link breakage.

VI. SAODV

SAODV assumes that there is a key management sub-system that makes it possible for each ad hoc node to obtain public keys from the other nodes of the network. Further, each ad hoc node is capable of securely verifying the association between the identity of a given ad hoc node and the public key of that node. This paper provides a possible solution of how this can be achieved. This section provides an overview to SAODV that will be need it to understand how this solution is applied to SAODV. Please, refer to [1] for a detailed analysis of SAODV.

Two mechanisms are used to secure the AODV messages: 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). For the non-mutable information, authentication is perform in an end-to-end manner, but the same kind of techniques cannot be applied to the mutable information.

The information relative to the hash chains and the signatures is transmitted with the AODV message as an extension message (let us refer to it as Signature Extension). To see the exact format of the SAODV Signature Extensions, please, refer to the version 0 of the SAODV draft [26].

A. SAODV hash chains

SAODV uses hash chains to authenticate the hop count of RREQ and RREP messages in such a way that allows every

TABLE I
POSSIBLE VALUES OF THE SIGNATURE METHOD FIELD

Value	Signature method
0	Reserved
1	RSA [27]
2	Elliptic curve [28]
3-127	Reserved
128-255	Implementation dependent

node that receives the message (either an intermediate node or the final destination) to verify that the hop count has not been decremented by an attacker.

The delayed verification could also be applied to the hash chains. But, since the time that it requires to verify a hash chain is practically negligible, there is no need for that.

B. SAODV digital signatures

Digital signatures are used to protect the integrity of the non-mutable data in RREQ and RREP messages. That means that they sign everything but the Hop_Count of the AODV message and the Hash from the SAODV extension.

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

When a node receives a RREP, it first verifies the signature before creating or updating a route to that host. Only if the signature is verified, will it store the route with the signature of the RREP and the lifetime.

VII. SAODV WITH DELAYED VERIFICATION AND KEY MANAGEMENT

This section shows how SAODV could be modify to implement the different techniques developed in this paper.

A. New fields

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 I (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.

B. 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 this paper, 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.

C. Generation of the IP address

SAODV can generate the IP addresses is very similar to the generation of SUCV (Statistically Unique and Cryptographically Verifiable) addresses [22]. SUCV addresses where designed to protect Binding Updates in Mobile IPv6. The main difference between SUCV and the method proposed in this paper is that SUCV addresses are generated by hashing an "imprint" in addition to the public key. That imprint (that can be a random value) is used to limit certain attacks related to Mobile IP.

In SAODV, the address can be 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 do not include an imprint):

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

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

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 [29].

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

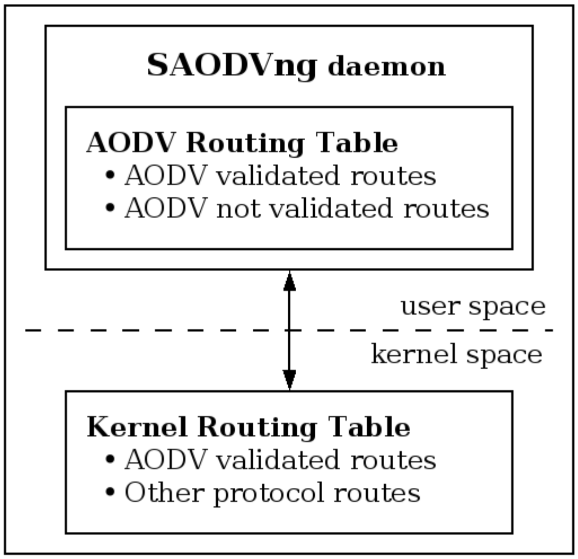

Fig. 1. SAODV daemon

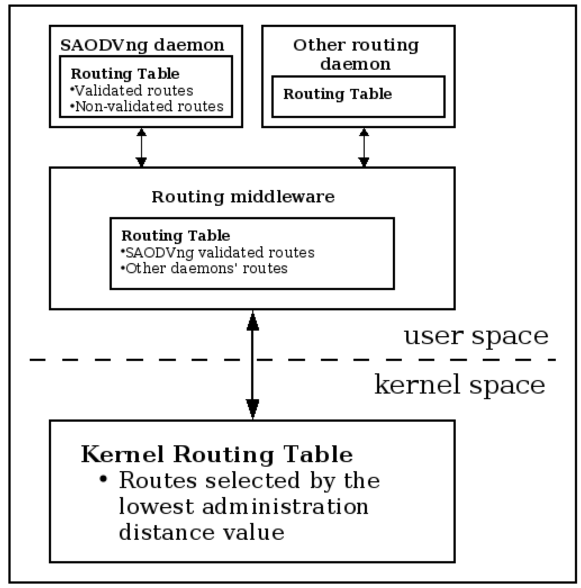

Fig. 2. SAODV daemon with a routing middleware

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).

This paper does not propose a scheme for IPv4 since the author considers the length of an IPv4 address to be too short to provide the statistical uniqueness that this scheme requires.

D. Duplicated IP Address Detection

SAODV can deal with the duplicated IP address problem as described in section III. 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).

E. Implementation Considerations

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

Figure 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, this paper recommends a default distance value of 165 for SAODV (and also for AODV in general).

VIII. Simulation Results

The simulations were done with 30 nodes moving at a maximum speed of 10 meters per second in a square of 1000x1000 meters. They established 10 connections that started between second 0 and second 25 (according to an uniform distribution). The simulation time was of 100 seconds, and the connections where constant bit rate (a packet of 512 each 0.25 seconds).

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. RSA and ECC have used key lengths with equivalent security (1368 bit RSA and 160 bit ECC).

Table II shows the times for signing/verifying in a Compaq iPAQ 3670 (206Mhz, 16M ROM, 64M RAM) according to

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

TABLE II
TIMES FOR A COMPAQ IPAQ 3670

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

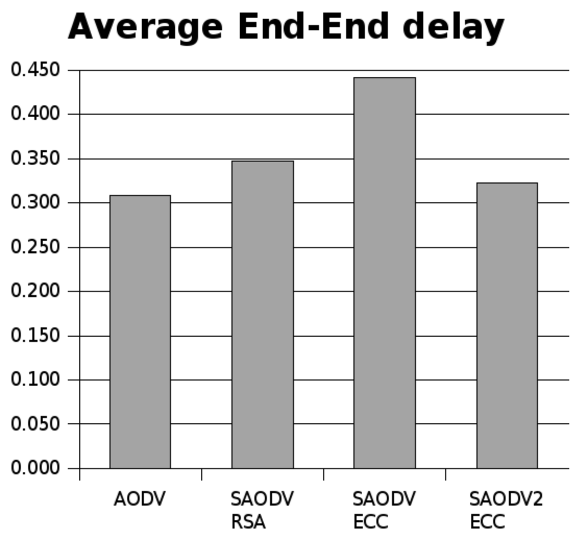

Fig. 3. Simulation Results
The delay is measured in milliseconds

[30]. 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).

Figure 3 shows the averaged result of the simulations. 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.

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 much more slowlier.

IX. CONCLUDING REMARKS

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.

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 in the paper an attacker cannot forge a public/private key pair from an IP address so the identity token becomes the IP address itself.

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.

ACKNOWLEDGMENT

This work was supported in part by CYCIT TIC2003-09279-C02-01 and by the I2CAT Foundation.

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

He also wants to thank Ana Escudero, from the Department of Technology at the Universitat Pompeu Fabra, for her help with the simulations.

REFERENCES

[1] M. Guerrero Zapata and N. Asokan, "Securing Ad hoc Routing Protocols," in *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*, September 2002, pp. 1–10.

[2] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), Jan 2002.

[3] Y. C. Hu, D. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," in *4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02), June 2002*, June 2002, pp. 3–13. [Online]. Available: citeseer.nj.nec.com/hu02sead.html

[4] Y. C. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," Rice University, Tech. Rep. TR01-383, Dec. 2001. [Online]. Available: citeseer.nj.nec.com/531013.html

[5] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communications networks," *Mobile Networks and Applications*, vol. 1, no. 2, pp. 89–104, 1996. [Online]. Available: citeseer.nj.nec.com/ramanathan96survey.html

[6] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications*, pp. 46–55, Apr. 1999.

[7] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, November/December 1999. [Online]. Available: citeseer.nj.nec.com/zhou99securing.html

[8] B. R. Smith, S. Murthy, and J. J. Garcia-Luna-Aceves, "Securing distance-vector routing protocols," in *Symposium on Network and Distributed Systems Security (NDSS '97)*. San Diego, California: Internet Society, Feb. 1997, pp. 85–92.

[9] R. Hauser, A. Przygienda, and G. Tsudik, "Reducing the cost of security in link state routing," in *Symposium on Network and Distributed Systems Security (NDSS '97)*. San Diego, California: Internet Society, Feb. 1997, pp. 93–99.

[10] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, 2000, pp. 255–265. [Online]. Available: citeseer.nj.nec.com/marti00mitigating.html

[11] B. Dahill, B. N. Levine, E. Royer, and C. Shields, "A secure routing protocol for ad hoc networks," University of Massachusetts, Departament of Computer Science, Tech. Rep. UM-CS-2001-037, Aug. 2001.

[12] D. B. Johnson *et al.*, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," INTERNET DRAFT, MANET working group, Feb. 2002, draft-ietf-manet-dsr-07.txt.

[13] Z. J. Haas, M. R. Pearlman, and P. Samar, "The interzone routing protocol (IERP) for ad hoc networks," INTERNET DRAFT, MANET working group, July 2002, draft-ietf-manet-zone-ierp-02.txt.

[14] S. Cheung, "An efficient message authentication scheme for link state routing," in *13th Annual Computer Security Applications Conference*, 1997, pp. 90–98.

[15] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "SPINS: security protocols for sensor netowrks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, 2001, pp. 189–199. [Online]. Available: citeseer.nj.nec.com/article/perrig01spins.html

[16] K. Zhang, "Efficient protocols for signing routing messages," in *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS'98)*, July 2001.

[17] N. Asokan, "Presentation at an informal workshop on mobile and ad hoc networking security, EPFL, Lausanne, December 2001," Dec. 2001.

[18] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking*, 1998, pp. 85–97. [Online]. Available: citeseer.nj.nec.com/broch98performance.html

[19] A. Perrig, R. Canetti, D. Song, and D. Tygar, "Efficient and secure source authentication for multicast," in *Network and Distributed System Security Symposium (NDSS'01)*, Feb. 2001. [Online]. Available: citeseer.nj.nec.com/perrig01efficient.html

[20] S. Thomson and T. Narten, "Ipv6 stateless address autoconfiguration," IETF Request for Comments, Dec. 1998, `RFC 2462`.

[21] S. Cheshire and B. Aboba, "Dynamic configuration of ipv4 link-local addresses," IETF INTERNET DRAFT, zeroconf working group, June 2001, draft-ietf-zeroconf-ipv4-linklocal-03.txt.

[22] G. Montenegro and C. Castelluccia, "Statistically unique and cryptographically verifiable (SUCV) identifiers and addresses," Network and Distributed System Security Symposium (NDSS '02), Feb. 2002. [Online]. Available: citeseer.nj.nec.com/502628.html

[23] G. O'Shea and M. Roe, "Child-proof authentication for mipv6 (CAM)," ACM Computer Communication Review, Apr. 2001. [Online]. Available: http://doi.acm.org/10.1145/505666.505668

[24] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *The Handbook of Applied Cryptography*. CRC Press, 1996, iSBN 0-8493-8523-7. [Online]. Available: http://perlmonks.thepen.com/113573.html

[25] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," Internet Request for Comments RFC 3561, Nov. 2003.

[26] M. Guerrero Zapata, "Secure ad hoc on-demand distance vector (saodv) routing," first published in the IETF MANET Mailing List (October 8th 2001), Aug. 2002, iNTERNET-DRAFT draft-guerrero-manet-saodv-00.txt.

[27] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, February 1978.

[28] RSA Laboratories, "Elliptic Curve Cryptography Standard," Public-Key Cryptography Standards (PKCS) 13, 1998.

[29] R. Hinden and S. Deering, "Ip version 6 addressing architecture," Internet Request for Comments RFC 2373, 1998.

[30] J. Walter, J. Oleksy, and J. Kong, "The role of ecdsa in wireless communications," Master Thesis. Computer Science Department. University of California, 2002.