Volume 2, No. 8, August 2011
Journal of Global Research in Computer Science
ISSN-2229-371X
TECHNICAL NOTE
Available Online at www.jgrcs.info

EFFICIENT TECHNIQUES FOR SAODV IN MOBILE ADHOC NETWORK

Anil Suryavanshi*[1] and Dr. Poonam Sinha[2]
Department of Information Technology
Barkatullah University Institute of Technology, Bhopal, M.P., India
ak_suryavanshi@yahoo.com[1]
poonamuit@yahoo.com[2]

Abstract: MANET (Mobile and Ad hoc NETworks) is a highly challenged network environment due to its special characteristics such as decentralization, dynamic topology and neighbor based routing. MANET is networks in which nodes are mobile and link connectivity might change all the time. In this kind of networks routing, security and key management are important and complex problems. The problem of routing has been properly addressed by the research community. Nevertheless, the research in security and key management has been postponed or relegated to a second term. This thesis work tries to give a solution to the needs in security for MANET networks using as a base a pre-existing routing protocol: Ad Hoc On-Demand Vector Routing (AODV). The selection of AODV is because the author of this research work is one of the contributors to AODV (so he knows perfectly how it works) and because it seemed that it would be the one that could more easily accommodate the needed modifications. The proposed solution in an extension to AODV called Secure AODV (SAODV). This thesis work includes an enhancement to AODV that allows using shorter routes, which will result in lower end-to-end delays, and longer battery life better than existence works.

Keywords- MANET, AODV, Detection Techniques.

INTRODUCTION

MANET (Mobile and Ad hoc NETworks) is networks formed by nodes that are mobile. They use wireless communication to speak among them and they do it in an ad hoc manner. In this kind of networks, routing protocols have to be different than from the ones used for fixed networks. In addition, nodes use the air to communicate, so a lot of nodes might hear what a node transmits and there are messages that are lost due to collisions. The concept of servers has to be modified: there is no guarantee that a node will be able to reach another node, so things like DNS servers, certification authorities (CAs) and other entities that are assumed to be found in fixed networks cannot exist.

Ad Hoc On-Demand Vector Routing (AODV) protocol is a reactive routing protocol for ad hoc and mobile networks. That means that AODV does nothing until a node needs to transmit a packet to a node for which it does not know a route. In addition, it only maintains routes between nodes which need to communicate. Its routing messages do not contain information about the whole route path, but only about the source and the destination. Therefore, routing messages have a constant size, independently of the number of hops of the route. It uses destination sequence numbers to specify how fresh a route is (in relation to another), which is used to grant loop freedom. In AODV, a node does route discovery by flooding the network with a 'Route Request' message (RREQ). Once it reaches a node that knows the requested route, it replies with a 'Route Reply' message (RREP) that travels back to the originator of the RREQ. After this, all the nodes of the discovered path have routes to both ends of the path [1] and [3] and [4].

In most domains, the primary security service is authorization. Routing is no exception. Typically, a router needs to make two types of authorization decisions. First, when a routing update is received from the outside, the router needs to decide whether to modify its local routing information base accordingly. This is import authorization. Second, a router may carry out export authorization whenever it receives a request for routing information. Import authorization is the critical service.

In traditional routing systems, authorization is a matter of policy. For example, gated, a commonly used routing program1, allows the administrator of a router to set policies about whether and how much to trust routing updates from other routers: e.g., statements like "trust router X about routes to networks A and B". In mobile ad hoc networks, such static policies are not sufficient (and unlikely to be relevant anyway). Authorization may require other security services such as authentication and integrity. Techniques like digital signatures and message authentication codes are used to provide these services.

In the context of routing, confidentiality and non-repudiation are not necessarily critical services. Non-repudiation is useful in an ad hoc network for isolating misbehaving routers: a router A which received an "erroneous message" from another router B may use this message to convince other routers that B is misbehaving. This would indeed be useful if there is a reliable way of detecting erroneous messages. This does not appear to be an easy task.

The problem of compromised nodes is not addressed here since it is, arguably, not critical in non military scenarios. Availability is considered to be outside of scope. Although of course it would be desirable, it does not seem to be feasible to prevent denial-of-service attacks in a network that uses wireless technology (where an attacker can focus on the physical layer without bothering to study the routing protocol). Therefore, in this thesis work the following requirements will considered:

A. Import authorization: It is important to note that in here it is not referring to the traditional meaning of authorization. What means is that the ultimate authority on routing messages regarding a certain destination node is that node itself. Therefore, route information will only be authorized in a 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.
B. Source authentication: Nodes need to be able to verify that the node is the one it claims to be.
C. Integrity: In addition, nodes need to be able to verify that the routing information that it is being sent to us has arrived unaltered.
D. The two last security services combined build data authentication, and they are requirements derived from our import authorization requirement.

The objectives of this thesis are to examine the additional cost of adding a security feature into non-secure routing protocols in various scenarios. The additional cost includes delay in packet transmission, the low rate of data packets over the total packets sent.

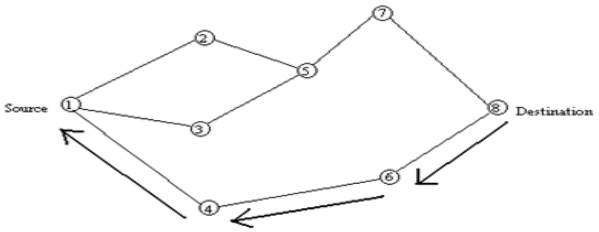
BACKGROUND

On-Demand Routing Protocols:-

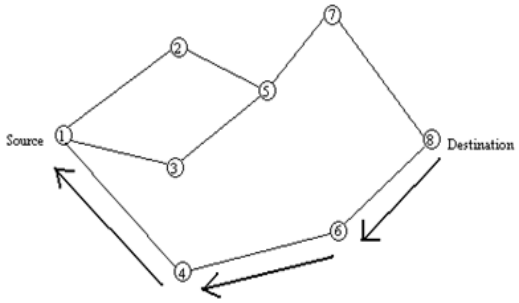
These protocols take a lazy approach to routing. In contrast to table-driven routing protocols all up-to-date routes are not maintained at every node, instead the routes are created as and when required. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. The route remains valid till the destination is reachable or until the route is no longer needed.

Ad hoc on-demand Distance Vector Routing (AODV):-

Ad hoc On-demand Distance Vector Routing (AODV) is an improvement on the DSDV algorithm. AODV minimizes the number of broadcasts by creating routes on-demand as opposed to DSDV that maintains the list of all the routes. To find a path to the destination, the source broadcasts a route request packet. The neighbors in turn broadcast the packet to their neighbors till it reaches an intermediate node that has recent route information about the destination or till it reaches the destination (Figure 2.6a). A node discards a route request packet that it has already seen. The route request packet uses sequence numbers to ensure that the routes are loop free and to make sure that if the intermediate nodes reply to route requests, they reply with the latest information only. When a node forwards a route request packet to its neighbors, it also records in its tables the node from which the first copy of the request came. This information is used to construct the reverse path for the route reply packet. AODV uses only symmetric links because the route reply packet follows the reverse path of route request packet. As the route reply packet traverses back to the source (Figure 1b), the nodes along the path enter the forward route into their tables. If the source moves then it can reinitiate route discovery to the destination. If one of the intermediate nodes move then they moved nodes neighbor realizes the link failure and sends a link failure notification to its upstream neighbors and so on till it reaches the source upon which the source can reinitiate route discovery if needed [2] and [5].



(a) Propagation of Route Request (RREQ) Packet



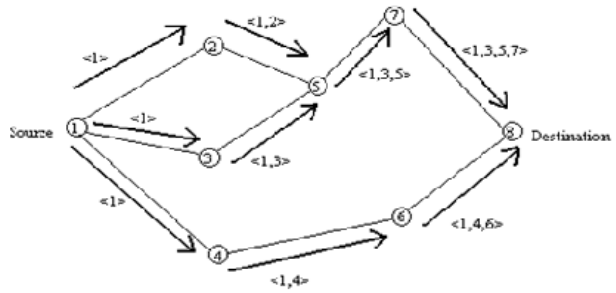
(b) Path taken by the Route Reply (RREP) Packet
Figure 1 Route Discovery in AODV

A. Dynamic Source Routing Protocol:-

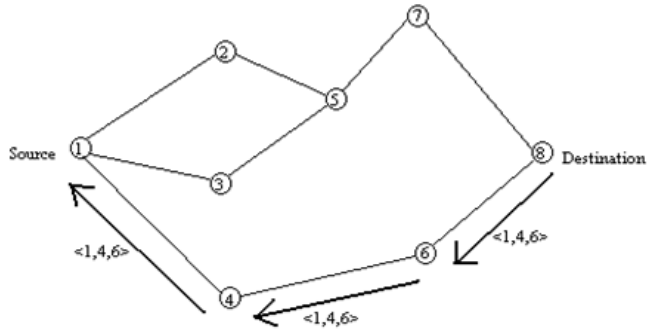
The Dynamic Source Routing Protocol is a source-routed on-demand routing protocol. A node maintains route caches containing the source routes that it is aware of. The node updates entries in the route cache as and when it learns about new routes. The two major phases of the protocol are: route discovery and route maintenance. When the source node wants to send a packet to a destination, it looks up its route cache to determine if it already contains a route to the destination. If it finds that an unexpired route to the destination exists, then it uses this route to send the packet. But if the node does not have such a route, then it initiates the route discovery process by broadcasting a route request packet. The route request packet contains the address of the source and the destination, and a unique identification number. Each intermediate node checks whether it knows of a route to the destination. If it does not, it appends its address to the route record of the packet and forwards the packet to its neighbors. To limit the number of route requests propagated, a node processes the route request packet only if it has not already seen the packet and it's address is not present in the route record of the packet. A route reply is generated when either the destination or an intermediate node with current information about the destination receives the route request packet. A route request packet reaching such a node already contains, in its route record, the sequence of hops taken from the source to this node

As the route request packet propagates through the network, the route record is formed as shown in figure 2a. If the route reply is generated by the destination then it places the route record from route request packet into the route reply packet. On the other hand, if the node generating the route reply is an intermediate node then it appends its cached route to destination to the route record of route request packet and puts that into the route reply packet. Figure 2.7b shows the

route reply packet being sent by the destination itself. To send the route reply packet, the responding node must have a route to the source. If it has a route to the source in its route cache, it can use that route. The reverse of route record can be used if symmetric links are supported.



(a) Building Record Route during Route discovery



(b) Propagation of Route Reply with the route record
Figure 2 Creation of record route in DSRP

In case symmetric links are not supported, the node can initiate route discovery to source and piggyback the route reply on this new route request. DSRP uses two types of packets for route maintenance: - Route Error packet and Acknowledgements. When a node encounters a fatal transmission problem at its data link layer, it generates a Route Error packet. When a node receives a route error packet, it removes the hop in error from it's route cache. All routes that contain the hop in error are truncated at that point. Acknowledgment packets are used to verify the correct operation of the route links. This also includes passive acknowledgments in which a node hears the next hop forwarding the packet along the route [1] and [3].

B. Temporally Ordered Routing Algorithm (TORA):-

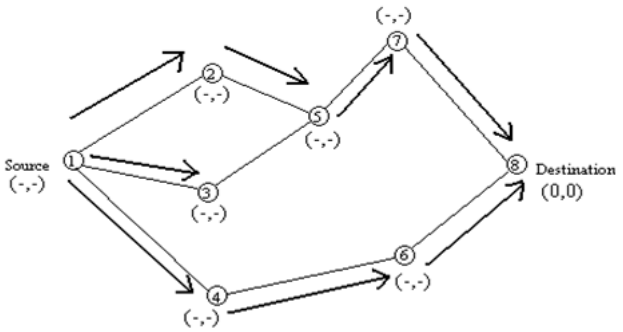
The Temporally Ordered Routing Algorithm (TORA) is a highly adaptive, efficient and scalable distributed routing algorithm based on the concept of link reversal. TORA is proposed for highly dynamic mobile, multihop wireless networks. It is a source-initiated on-demand routing protocol. It finds multiple routes from a source node to a destination node. The main feature of TORA is that the control messages are localized to a very small set of nodes near the occurrence of a topological change. To achieve this, the nodes maintain routing information about adjacent nodes. The protocol has three basic functions: Route creation, Route maintenance, and Route erasure.

Each node has a quintuple associated with it –

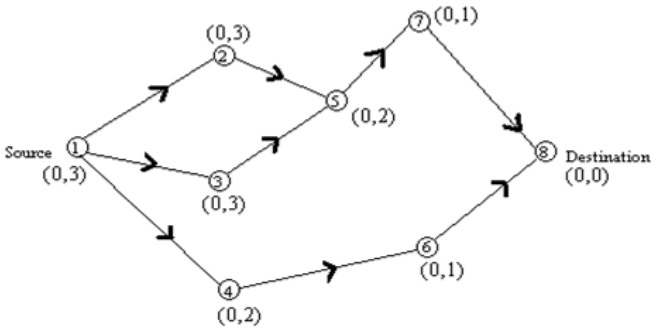
a. Logical time of a link failure
b. The unique ID of the node that defined the new reference level
c. A reflection indicator bit
d. A propagation ordering parameter
e. The unique ID of the node

The first three elements collectively represent the reference level. A new reference level is defined each time a node loses its last downstream link due to a link failure. The last two values define a delta with respect to the reference level. Route Creation is done using QRY and UPD packets.

The route creation algorithm starts with the height (propagation ordering parameter in the quintuple) of destination set to 0 and all other node's height set to NULL (i.e. undefined). The source broadcasts a QRY packet with the destination node's id in it. A node with a non-NULL height responds with a UPD packet that has its height in it. A node receiving a UPD packet sets its height to one more than that of the node that generated the UPD. A node with higher height is considered upstream and a node with lower height downstream. In this way a directed acyclic graph is constructed from source to the destination. Figure 6 illustrates a route creation process in TORA. As shown in figure 3a, node 5 does not propagate QRY from node 3 as it has already seen and propagated QRY message from node 2. In figure 3b, the source (i.e. node 1) may have received a UPD each from node 2 or node 3 but since node 4 gives it lesser height, it retains that height.



(a) Propagation of QRY message through the network



(b) Height of each node updated as a result of UDP message
Figure 3 Route creation in TORA. (Numbers in braces are reference level, height of each node)

When a node moves the DAG route is broken, and route maintenance is needed to reestablish a DAG for the same destination. When the last downstream link of a node fails, it generates a new reference level. This results in the propagation of that reference level by neighboring nodes as shown in figure 4. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links.

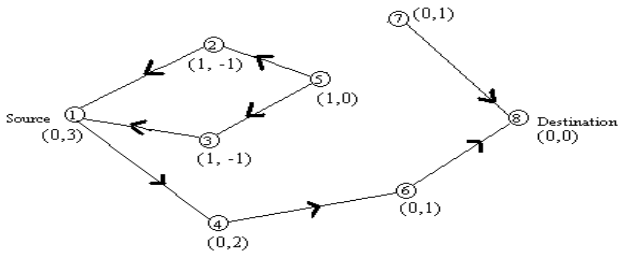


Figure 4 Re-establishing route on failure of link 5-7. The new reference level is node 5.

In the route erasure phase, TORA floods a broadcast clear packet (CLR) throughout the network to erase invalid routes. In TORA there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other. Because TORA uses internodal coordination, its instability problem is similar to the "count-to-infinity" problem in distance-vector routing protocols, except that such oscillations are temporary and route convergence will ultimately occur [5] and [7].

RELATED WORKS

An ad hoc network is often defined as an "infrastructure less" network, meaning a network without the usual routing infrastructure like fixed routers and routing backbones. Typically, the ad hoc nodes are mobile and the underlying communication medium is wireless. Each ad hoc node may be capable of acting as a router. Such ad hoc networks may arise in personal area networking, meeting rooms and conferences, disaster relief and rescue operations, battlefield operations, etc.

There is very little published prior work on the security issues in ad hoc network routing protocols. Neither the survey by Ramanathan and Steenstrup nor the survey by Royer and Toh mention security. None of the draft 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.

There are some works on securing routing protocols for fixed networks that also deserved to be mentioned here. Perlman, in her thesis, proposed a link state routing protocol that achieves Byzantine Robustness. Although her protocol is highly robust, it requires a very high overhead associated with public key encryption. Secure BGP attempts to secure the Border Gateway Protocol by using PKI (Public Key Infrastructure) and IPsec. In their paper on securing ad hoc networks, Zhou and Haas primarily discuss key management. 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. And, lately, some work has been done to secure ad hoc networks by using misbehavior detection schemes. 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 [6] and [7].

In recent 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 here only require originators to sign the message. In addition, it is prone to reply attacks using error messages unless the nodes have time synchronization [5] and [8].

In previous proposed a protocol (SRP) that can be applied to several existing routing protocols (in particular DSR and IERP. 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 order to provide delayed key disclosure. While, 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.

In SEAD (by Hu, Johnson and Perrig) hash chains are also used in combination with DSDV-SQ (this time to authenticate hop counts and sequence numbers). At every given time each node has its own has 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 used with any suitable authentication and key distribution scheme. But finding such a scheme is not straightforward.

Ariadne, by the same authors, is based on DSR and TESLA (on which it is based its authentication mechanism). It also requires clock synchronization, which is, arguably, an unrealistic requirement for ad hoc networks [8] and [9].

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 very small subset of the possible scenarios.

Looking at the work that had been done in this area previously, it could be felt that the security needs for ad hoc networks had not been yet satisfied (at least for those scenarios where everybody can freely participate in the network) [8] and [10].

PROPOSED TECHNIQUES

Securing Ad hoc Protocols:-

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. Moreover, as a result of the processing of the routing message, a node might modify its routing. 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) to be able to apply their import authorization policy. 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 two types of information could be distinguished: 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.

If the security system being used to secure the network transmissions in a MANET network is IPSec, it is necessary that the IPSec implementation can use as a selector the TCP and UDP port numbers. This is because it is necessary that the IPSec policy will be able to apply certain security mechanisms to the data packets and just bypass the routing packets (that typically can be identified because they use a reserved transport layer port number).

Propose Security flaws of AODV:-

Since AODV has no security mechanisms, malicious nodes can perform many attacks just by not behaving according to the AODV rules. A malicious node M can carry out the following attacks (among many others) against AODV:

a. Impersonate a node S by forging a RREQ with its address as the originator address.
b. When forwarding a RREQ generated by S to discover a route to D, reduce the hop count field to increase the chances of being in the route path between S and D so it can analyze the communication between them. A variant of this is to increment the destination sequence number to make the other nodes believe that this is a 'fresher' route.
c. Impersonate a node D by forging a RREP with its address as a destination address.
d. Impersonate a node by forging a RREP that claims that the node is the destination and, to increase the impact of the attack, claims to be a network leader of the subnet SN with a big sequence number and send it to its neighbors. In this way it will became (at least locally) a blackhole for the whole subnet SN.
e. Selectively, not forward certain RREQs and RREPs, not reply to certain RREPs and not forward certain data messages. This kind of attack is especially hard to even detect because transmission errors have the same effect.
f. Forge a RERR message pretending it is the node S and send it to its neighbor D. The RERR message has a very high destination sequence number dsn for one of the unreachable destinations (U). This might cause D to update the destination sequence number corresponding to U with the value dsn and, therefore, future route discoveries performed by D to obtain a route to U will fail (because U's destination sequence number will be much smaller than the one stored in D's routing table).
g. According to the current AODV draft, the originator of a RREQ can put a much bigger destination sequence number than the real one. In addition, sequence numbers wraparound when they reach the maximum value allowed by the field size. This allows a very easy attack in where an attacker is able to set the sequence number of a node to any desired value by just sending two RREQ messages to the node.

Propose Securing AODV:-

Let us assume 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. How this is achieved depends on the key management scheme. 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 performing 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 that will be refereed as Signature Extension.

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 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.. A hash chain is formed by applying a one-way hash function repeatedly to a seed. Every time a node originates a RREQ or a RREP message, it performs the following operations:

a. Generates a random number (seed)

b. Sets the Max Hop Count field to the TimeToLive value (from the IP header).

Max Hop Count = TimeToLive

c. Sets the Hash field to the seed value.

Hash = seed

d. Sets the Hash Function field to the identifier of the hash function that it is going to use.

Hash Function = h

e. Calculates Top Hash by hashing seed Max Hop Count times.

$$\text{Top Hash} = h^{\text{Max Hop Count}}(\text{seed})$$

Where:

– h is a hash function.

– $h^i(x)$ is the result of applying the function h to x i times.

In addition, every time a node receives a RREQ or a RREP message, it performs the following operations in order to verify the hop count:

• Applies the hash function h Maximum Hop Count minus Hop Count times to the value in the Hash field, and verifies that the resultant value is equal to the value contained in the Top Hash field.

$$\text{Top Hash} == h^{\text{Max Hop Count}-\text{Hop Count}}(\text{Hash})$$

Where:

– a == b reads: to verify that a and b are equal.

f. Before rebroadcasting a RREQ or forwarding a RREP, a node applies the hash function to the Hash value in the Signature Extension to account for the new hop.

Hash = h(Hash)

The Hash Function field indicates which hash function has to be used to compute the hash. Trying to use a different hash function will just create a wrong hash without giving any advantage to a malicious node. Hash Function, Max Hop Count, Top Hash, and Hash fields are transmitted with the AODV message, in the Signature Extension. And, as it will be explained later, all of them but the Hash fields are signed to protect its integrity.

Propose 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. The main problem in applying digital signatures is that AODV allows intermediate nodes to reply RREQ messages if they have a 'fresh enough' route to the destination. While this makes the protocol more efficient it also makes it more complicated to secure. The problem is that a RREP message generated by an intermediate node should be able to sign it on behalf of the final destination. And, in addition, it is possible that the route stored in the intermediate node would be created as a reverse route after receiving a RREQ message (which means that it does not have the signature for the RREP).

To solve this problem, SAODV offers two alternatives. The first one (and also the obvious one) is that, if an intermediate node cannot reply to a RREQ message because it cannot properly sign its RREP message, it just behaves as if it didn't have the route and forwards the RREQ message.

The second is that, every time a node generates a RREQ message, it also includes the RREP flags, the prefix size and the signature that can be used (by any intermediate node that creates a reverse route to the originator of the RREQ) to reply a RREQ that asks for the node that originated the first RREQ. Moreover, when an intermediate node generates a RREP message, the lifetime of the route has changed from the original one. Therefore, the intermediate node should include both lifetimes (the old one is needed to verify the signature of the route destination) and sign the new lifetime.

In this way, the original information of the route is signed by the final destination and the lifetime is signed by the intermediate node. To distinguish the different SAODV extension messages, the ones that have two signatures are called RREQ and RREP Double Signature Extension. When a node receives a RREQ, it first verifies the signature before creating or updating a reverse route to that host. Only if the signature is verified, will it store the route. If the RREQ was received with a Double Signature Extension, then the node will also store the signature for the RREP and the lifetime (which is the 'reverse route lifetime' value) in the route entry. An intermediate node will reply to a RREQ with a RREP only if it fulfills the AODV's requirements to do so and the node has the corresponding signature and old lifetime to put into the Signature and Old Lifetime 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 it fulfills the AODV's requirements to do so. This RREP will be sent with a RREP Single 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.

Propose Route Repair:-

Some routing protocols in MANET networks have a mechanism to try to repair a broken route (due to a link breakage) that does not imply a complete route discovery. An example would be the "local repair" in AODV in which when a link used to send data packets breaks, the node upstream of the link that got broken may (if it was close to the destination) do a route discovery of the destination broadcasting the route request with a TimeToLive that is assumed to be enough to reach the destination.

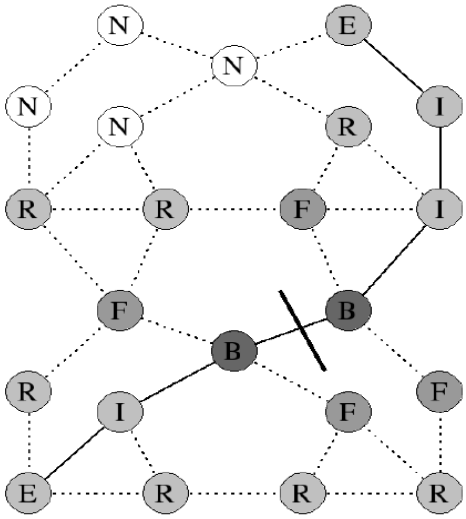


Figure 5: Propagation of SREQs

This method has the problem that it only repairs the route in one direction. Chances are that the route is used in both directions. Therefore, if it only repairs the route in one direction, another route discovery will be needed to repair the route in the other direction. A possible solution, would be to use the shortcut discovery method described in here to do the route repair. To do so, when a link breakage occurs, the two nodes that were connected through that link will initiate a "repaired route discovery". This repaired route discovery will consist of sending a SREQ to the end of the route to which they are still connected. The differences with a normal SREQ message will be:

• The message will be flagged as repair route SREQ.

• The hop count to the endpoint that is not available anymore will be set to infinity (typically indicated by the value 255).

• Optionally, the original SREQ (the one originated by one of the two nodes that were connected through that link) might be also forwarded by all their immediate neighbors that were not part of the original route. Of course, if they forward it, the forwarded SREQ should have increased the hop count that is not set to infinity in the SREQ (to account for the new hop that has been done).

Figure 5 shows how SREQs are propagated. End points of the previous route are marked as ′E′. The two nodes that where connected through the link that has just broken are mark as ′B′ and intermediate nodes that where part of the route as ′I′. The neighbors of the ′B′ nodes that are not part of the route but will forward the SREQ are marked as ′F′ nodes. The rest of the nodes that will receive a SREQ are marked as ′R′ nodes. Finally, the other nodes are marked as ′N′. Due to the fact that the neighbors of the ′B′ nodes (the ′F′ nodes) forward the SREQs, there will be a broader diffusion of the SREQs in the zone nearby the link breakage.

AODV-SDR:-

AODV-SDR (AODV with shortcut discovery and route repair) incorporates two new types of messages to the standard AODV: Shortcut REQuest (SREQ) and Shortcut REPly (SREP). SREQs have a "R flag" that is set if the SREQ is used to do a route repair. They also contain a "SREQ ID", that is a sequence number that identifies uniquely the SREQ with the end point that originated the SREQ. In case this SREQ was originated due to a route repair both nodes that where connected through the link that broke will generate SREQs that will probably have different sequence numbers. SREQs also contain the following information about both end points of the route: IP address, the next hop of the route that goes to the end point, the sequence number of that route, and the hop count to the end point. SREPs are basically AODV's "Route Reply" (RREP) messages with a flag set to indicate that they are SREPs. Once the shortcut is discovered they propagate back the shortcut route. Therefore they contain all the information about that route: hop count, IP address, lifetime, etc.

RESULTS

In our simulation results shows different parameters where changed to see how it affects the following metrics: Completed transmissions, and average end-to-end delay.

The figures 6 and 7 show the effects of changing nodes' speed. An important thing to note is that shortcut discovery does not improve the metrics in the case that there nodes do not move. Nevertheless, if they all move (even if it is only 1 meter/second) the numbers of completed transmissions drop, but with our detection much more transmissions are completed than without it. Another thing to note is that the average distance between to nodes that move with our mobility pattern is smaller than between two nodes that do not move. This happens with most mobility patterns that use a finite area, and it justifies why the average hop count and the average end-to-end delay are bigger in the case of non-moving nodes.

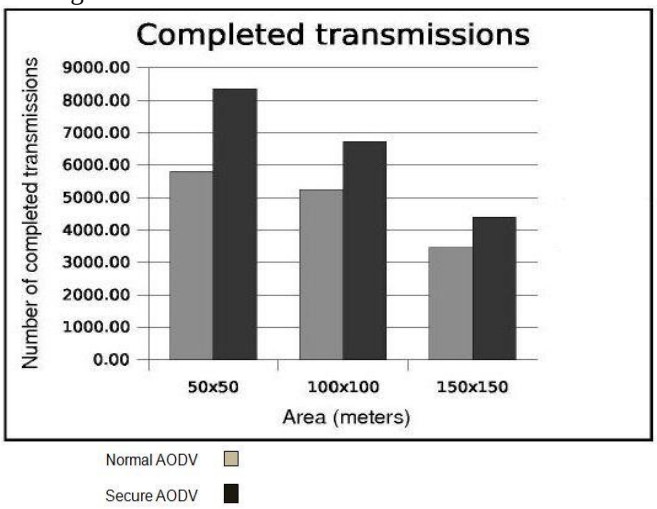


Figure 6: Completed Transmission on Normal and Secure AODV

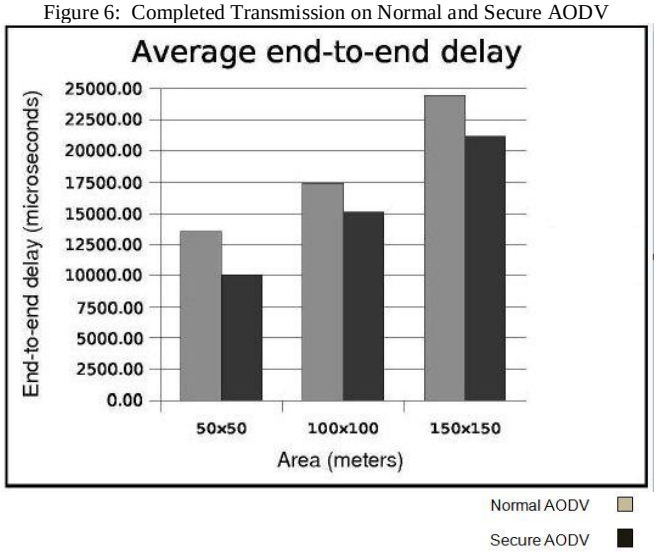


Figure 7: Average end-to-end on Normal and Secure AODV

In all the scenarios, with the exception of non-moving nodes or moving, simulation has shown that one hop shortcut detection clearly improves completed transmissions while reducing the average hop count (which makes the batteries of the nodes to drain slower) and reducing the average end-to-end delay.

CONCLUSION

One of the most important lessons learned while designing SAODV has been the need to keep things clear, so they can be properly analyzed. In a security system there should be a clear distinction of the following items:

A. The scenario (or scenarios) that are going to protect.
B. The security features that this scenario requires.

C. The security mechanisms that will fulfill those security features.

Once the design of the cryptosystem is done, it is time to analyze if it indeed works. And, since the three items listed above are clearly separated in the design, it is much more easier to perform such analysis because it can be splited into the following parts:

D. The analysis of requirements: Whether the security features are enough for the targeted scenario.
E. The analysis of mechanisms: Whether the security mechanisms are indeed fulfilling all the security requirements. When doing this, it will be found that there are still some attacks that can be performed against your system. Some of them, typically, are not avoided because a tradeoff between security and feasibility.

REFERENCES

[1]. Zhiyuan LIU, Shejie LU, Jun YAN, " Secure Routing Protocol based Trust for Ad Hoc Networks", Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE 2007, pp- 279-283.

[2]. Wenchao Huang, Yan Xiong, Depin Chen, "DAAODV: A Secure Ad-hoc Routing Protocol based on Direct Anonymous Attestation", IEEE 2009 International Conference on Computational Science and Engineering, pp- 809-816.

[3]. Anand Patwardhan and Michaela Iorga, "Secure Routing and Intrusion Detection in Ad Hoc Networks", Proceedings of the 3rd IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom 2005).

[4]. Jun Pan and Jianhua Li, "MASR: An Efficient Strong Anonymous Routing Protocol for Mobile Ad Hoc Networks", IEEE 2009.

[5]. Wenchao Huang, Yan Xiong, Depin Chen, "DAAODV: A Secure Ad-hoc Routing Protocol based on Direct Anonymous Attestation", IEEE 2009 International Conference on Computational Science and Engineering, pp 809-816.

[6]. A.H Azni, Azreen Azman, Madihah Mohd Saudi, AH Fauzi, DNF Awang Iskandar, "Analysis of Packets Abnormalities in Wireless Sensor Network", IEEE 2009 Fifth International Conference on MEMS NANO, and Smart Systems, pp 259-264.

[7]. Cuirong Wang, Shuxin Cai and Rui Li, "AODVsec: A Multipath Routing Protocol in Ad-Hoc Networks for Improving Security", IEEE 2009 International Conference on Multimedia Information Networking and Security, pp 401-404.

[8]. A Nagaraju and B.Eswar, "Performance of Dominating Sets in AODV Routing protocol for MANETs", IEEE 2009 First International Conference on Networks & Communications, pp 166-170.

[9]. Sheng Cao and Yong Chen, "AN Intelligent MANet Routing Method MEC", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp 831-834.

[10]. WANG Xiao-bo ,YANG Yu-liang, AN Jian-wei, "Multi-Metric Routing Decisions in VANET", 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp 551-556.

