Securing MAODV in Wireless Sensor Network

Gaurang Deshmukh, MTech-NIMS(COMP) VJTI Mumbai
Madhulika Bhandure, MTech-NIMS(COMP) VJTI Mumbai
Prof. M. M. Chandane, Prof Computer Science, VJTI Mumbai

ABSTRACT
Most of the multicast routing protocols proposed for ad hoc networks assume a trusted, non-adversarial environment and do not take security issues into account in their design. In this paper, we investigate the security of MAODV (Multicast Ad hoc On-Demand Distance Vector protocol), a well-known multicast routing protocol, and identify several attacks on it. We show, via simulation, that these attacks can have a significant impact on the performance of MAODV. We present an authentication framework for MAODV and propose countermeasures that can prevent or mitigate the impact of these attacks.

KeyWords: MAODV, GRPH, AODV, PREP, PREQ, MACT, GL, PART, INV, MTF

Introduction
Many important applications for ad hoc networks are group oriented in nature, and can therefore benefit from a multicast communication service. Example applications include mobile conferencing outside the office, battlefield communications, and disaster recovery operations. For the most part, these protocols assume a trusted, non adversarial environment and their design do not take security issues into account. Although, several articles have investigated security issues for unicast routing protocols for ad hoc networks, there has been relatively little attention paid so far to securing multicast routing protocols for ad hoc networks. Specifically, we examine the security vulnerabilities of MAODV [1], a well-known protocol that is representative of tree-oriented multicast routing protocols for ad hoc networks [2]. Security attacks on routing protocols can loosely be classified into two categories: insider attacks and outsider attacks. Outsider attacks are launched by unauthorized nodes that do not possess the credentials to join an ad hoc network, whereas insider attacks are launched by compromised nodes that are part of the ad hoc network. Many insider attacks can be prevented by requiring all communication to be authenticated, thereby preventing nodes that do not possess the requisite cryptographic credentials from being able to inject messages into the network with the goal of disrupting the functioning of the routing protocol.

This approach however is not sufficient to prevent insider attacks, since the attacker possesses all the cryptographic material of the compromised node(s). In mobile ad-hoc networks, nodes are more vulnerable to capture than in traditional networks with a fixed infrastructure. Previously, Ning et al. [3] have presented a systematic study of insider attacks on AODV [4]. In securing tree-oriented multicast protocols such as MAODV, there is an additional complication in that the data delivery tree set up by the protocol includes some nodes that are not members of the multicast group. However, these nodes execute the routing protocol and participate in the forwarding of data and routing control packets. This paper makes the following contributions: We assess the vulnerability of MAODV to attacks launched by both inside and outside nodes. In particular, we identify attacks on multicast tree formation and maintenance that have no counterpart in unicast routing protocols.

WEB SERVICE SECURITY STANDARDS:
MAODV is a multicast routing protocol for ad hoc networks that dynamically constructs a shared multicast tree which connects the group members possibly via some non-member nodes. MAODV also allows a non-member node, which may or may not be on the tree, to multicast data to all the group members. MAODV is the multicast extension of Ad hoc On-Demand Distance Vector (AODV) routing protocol, and it shares many similarities and packet formats with AODV. The Route Request (RREQ) and Route Reply (RREP) packet formats are similar to those used in AODV. In addition to these packets, MAODV uses two routing control packets – Multicast Activation (MACT) and Group Hello (GRPH) packets. MAODV maintains a sequence number for the multicast group that is initialized by the group leader and increased periodically.
The primary responsibility of the group leader is to periodically broadcast Group Hello (GRPH) messages across the network and to maintain the group sequence number. A GRPH message contains the group sequence number, multicast group address and corresponding group leader IP address. The sequence number is used to ensure that routes discovered to the multicast group are always the most current ones available. Given the choice between two routes to a multicast tree, a requesting node selects the one with greater sequence number. We now describe the operations of MAODV.

A. Route Discovery and Link Activation
When a node either wants to join the multicast group or find a route to the multicast group, the node broadcasts a RREQ message. For join requests, a reply is sent when the RREQ reaches a node that is already a member of the multicast tree, and the node’s record of the multicast group sequence number is at least as great as that contained in the RREQ. For non-join requests, any node with a current route to the multicast tree may respond to the RREQ. The route to the multicast tree is made available by unicasting a RREP back to the source of the RREQ. Since each node receiving the request caches a route back to the source of the request, the RREP can be unicast back to the source from any node able to satisfy the request. In case of join requests, after waiting for a specified period to receive RREPs, the requester node selects the best route to the multicast tree and unicasts a MACT message (multicast activation), with J (join) flag set, to the next hop which is on the selected route. This message officially grafts the selected route onto the existing multicast tree.

B. Multicast tree maintenance
Multicast tree maintenance mainly involves three operations
1. Tree Pruning, 
2. Link Repair, and 
3. Partition Merging.

a) Tree Pruning: If a tree-node, which is not a member of the multicast group, becomes a leaf node, it prunes itself from the tree. It sends a MACT message, with P (prune) flag set, to its next hop. The next hop, on receiving this MACT message, deletes the entry for the sender node from its multicast route table. If this node is itself not a member of the multicast group, and if the pruning of the other node has made it a leaf node, it can similarly prune itself from the tree.

b) Link Repair: Each node on the multicast tree always monitors the status of the links with its immediate neighbors. When a link breakage is detected, the node downstream of the break (i.e., the node that is further from the multicast group leader) is responsible for repairing the broken link. If it has more than one downstream neighbours it sends a MACT packet, with G (group leader) flag set, to one of these neighbours. This process continues till a group member is reached and it becomes the group leader.

c) Partition Merging: Nodes in one partition of the multicast tree will discover another partition when they receive Group Hello packets from the group leader of another partition. These two partitions should be merged to improve the group connectivity. The multicast group leader with lower IP address initiates the merging process (say P1). P1 sends a RREQ with J (join) and R (repair) flags set to the group leader of the other partition (say P2). Upon receiving the RREQ, P2 takes the larger of its own and the received multicast group sequence numbers, increments this value by one, and sends a RREP with R flag set to P1. As the RREP is propagated back to P1, new links are grafted to connect these two trees. The next time P2 broadcasts a GRPH, it sets the U (update) flag to indicate that there is a change in group leader information and the nodes of the other partition update their group leader address and group sequence number.

ATTACKS on MAODV
MAODV does not include any provisions for security; thus, it is susceptible to attacks by outsiders as well as malicious insiders. Attackers may drop, modify, replay or fabricate routing messages [19]. Nodes may also impersonate other nodes [20] while sending fabricated messages. Further, multiple attacker nodes may collude to launch attacks, e.g. wormhole attacks [21]. In general, attacks on MAODV can be divided into two categories:
(i) Attacks on route discovery and establishment, 
(ii) Attacks on multicast tree maintenance.

The attacks on MAODV are similar to the attacks on AODV. In contrast, the attacks on the multicast tree formation and maintenance in MAODV have no counterpart in unicast routing protocols. We describe below several attacks on the operation of MAODV. To launch an attack, if the message misused (e.g., MACT) includes a special flag (e.g., Join flag J), the message name includes the flag in parentheses (e.g., MACT(J)). In the following discussion, we frequently use some terms which we define here. The term tree-node refers to a node that is on the shared multicast tree, and non-member refers to a node...
which is not a member of the multicast group. We assume that there is only one multicast group.

A. Attacks against Route Discovery and Establishment

1) Attacks Against Route Discovery:
   a) RREP-INV: When a node either wants to join a multicast group or find a route to a multicast group, it broadcasts a RREQ message. In this attack, a malicious node replies to the RREQ message with the goal of deceiving the node into believing it has found the best route to the multicast group. Consider a group member A that wants to join the multicast group. A broadcast a RREQ packet with the multicast group address as the destination address and with the J (Join) flag set. Only nodes on the multicast tree qualify to send a reply (RREP) to this request. However, a malicious node M can respond to the RREQ packet with a RREP even if it is not on the multicast tree. A RREP packet includes the replying node’s view of the group sequence number. Since A is likely to receive RREPs from multiple nodes, in order to increase the chances of the route to M being selected as the best route to the multicast group, M can fabricate the sequence number field in its RREP. This attack can be launched by a non-tree-node or by a tree-node. If a non-tree-node succeeds in deceiving the joining node into believing it has a route to the multicast group, it can discard any future messages sent by the node to the group, effectively negating the work done by the route discovery protocol.

2) Attacks against Link Activation:
   a) MACT (J)-MTF: A node (say M) may receive replies in response to its route request from more than one node. In normal MAODV operation, M will select the best route and send a MACT packet, with J flag set, to graft the corresponding edge onto the tree. If M is malicious, it may select more than one route, which will result in many extra edges being grafted on to the multicast tree, i.e., creating a mesh instead of a tree. If M sends MACT packets to all the nodes (say A, B and C) from which it received RREPs, then many unnecessary branches will be grafted on the multicast tree, connecting M to the existing tree. To launch this attack, M does not need to be the initiator of RREQ; it may be simply an intermediate node which rebroadcasts the RREQ. This attack is an instance of a resource consumption attack, since it will result in unnecessary packet duplication and energy expenditure. Note that in this attack a single malicious node can increase the energy expenditure of many other nodes.

B. Attacks on Multicast Tree Maintenance

Each of the tree maintenance operations, i.e., tree pruning, link repair, and merging partitions, is vulnerable to attacks by malicious nodes.

1) Attacks on the tree pruning process:
   a) MACT (P)-PART: In this attack, a malicious node impersonates a tree node and sends a MACT (P) packet, i.e., a prune message, to the tree node’s children in the multicast tree. If a downstream node is a non-member and has only one downstream link, it also prunes itself and sends a similar prune message to its downstream node. This may lead to the multicast tree being partitioned as explained below. As an example, consider the multicast tree. Suppose nodes (say B, C) are non-members whereas A is a group member. The attacker M impersonates A and sends a MACT packet with P flag set to A’s immediate downstream node B. Since B is not a group member, it prunes itself from the tree by forwarding the MACT packet to its downstream node. In this example, each of B, C will prune itself.

2) Attacks on the link repair process:
   a) RREP-PART: When a node’s link to its upstream node in the multicast tree breaks, it attempts to repair the link by broadcasting a RREQ packet with the J (Join) flag set. Suppose there is multicast tree with node L acting as the group leader. Suppose M is a malicious node. Suppose that the link between A and B breaks. Following MAODV, B starts route discovery by broadcasting RREQ packets with the multicast group address as the destination address. B’s RREQ packets will include the group sequence number and its hop count from the group leader L, which is equal to three. Only tree-nodes with current group sequence number greater or equal to the sequence number indicated in the RREQ packet and whose hop count from L is at most the hop count indicated in the RREQ packet should respond with a RREP to this request. However, a malicious node M, even if it is not a tree-node, may respond with a RREP message with sequence number higher than the current group sequence number and with a false hop count that is smaller than three. Even if B receives an RREP from E, it will select M because M has larger sequence number and smaller hop count. Thus nodes B, C and D get partitioned from other group members by M.

b) MACT (U)-PART: After repairing a link, it is possible that the node that initiated the repairing process is now at a different distance from the group leader than it was before the link repair. In this case,
it informs its downstream neighbors of their new distance from the group leader via a MACT message with U (update) flag set, and the hop count field set to the node’s new distance from the group leader. The downstream nodes repeat this process by sending MACT messages with U flag set to their downstream nodes. In this attack, the attacker impersonates a tree node and sends false information to its downstream nodes, which makes them decrease their hop count from the group leader. This may lead to the creation of a loop in that branch of the multicast tree and may isolate that branch. Suppose there is a treess on which the attacker impersonates node C and sends a false MACT packet, with update flag U set, to node D with hop count from the group leader less than the correct one (say one instead of three). On receiving this message, D updates its hop count field. Now if the link between A and B breaks, node B will try to repair the link by broadcasting a RREQ packet. Node D qualifies to send node B an RREP and the RREP can propagate possibly through an intermediate node F. So there is a possibility that the loop DFBC will be created after the route repair and the nodes B, C, D, E and F will be isolated from other members of the multicast group.

3) Attacks on the partition merge process:
   a) GL-PART: In MAODV, a group leader is responsible for broadcasting Group Hello packets, and taking the necessary steps for reconnecting two partitioned trees when connectivity is restored. If group leader (say M) detects that two partitions are within communication range, it needs to take one of the following actions: (i) if M has a lower IP address than that of the other leader (say N), M unicasts a RREQ packet to N, with R (repair) and J (join) flag set,
     
     (ii) Otherwise, after receiving the RREQ packet from N, M unicasts a RREP packet (with R flag set) towards N. If M is malicious, it may not perform these actions and the two partitions will remain disconnected. We note that in the absence of authentication, any malicious node (say M) can become a group leader simply by broadcasting GRPH packets. If node M’s GRPH packets include a (potentially fabricated) IP address that is higher than the IP address of the current group leader (say N), M will replace N as the group leader of the nodes in its tree.

SECURING MAODV

A. Authentication Framework

1) Design Goals: We distinguish between three types of nodes that can launch attacks on the operation of MAODV. First, attacks can be launched by outsiders, who do not possess the credentials to join the ad hoc network. At the other extreme, attacks can be launched by compromised group members. Insider attacks are very difficult to prevent since the attacker possesses the credentials to participate in all the operations of the protocol. The third category of attackers includes non-member nodes that possess the credentials to join the ad hoc network but are not members of the multicast group under consideration. In MAODV, such nodes can participate in a subset of the operations of the protocol. For example, a non-member can become a tree node and participate in the routing of packets. However, a non-member tree node cannot become a group leader. Given these three categories of attackers, our authentication framework has three objectives. First, an unauthorized node should not be able to participate in the MAODV protocol. Second, a non-member node should not be able to impersonate a group member. The third goal is to design an efficient authentication scheme that prevents any node which is not on the multicast tree from impersonating a tree node.

2) Authentication Mechanisms: To achieve the goals outlined above, we propose the use of an authentication framework in which nodes need the appropriate credentials to participate in the MAODV protocol as a group member or tree node. The routing control messages exchanged between nodes are augmented to include additional fields that allow the receiving node to verify the authenticity of the message. The different elements of our authentication framework are described below.

1) Each authorized node (group members as well as nonmembers) in the network possesses a public/private key pair and a certificate signed by a Certification Authority (CA), which can be verified by all nodes. This certificate binds a node’s public key with its IP address. We refer to this certificate as a node certificate. Only nodes that possess a node certificate are eligible to participate in routing.

2) A group member has an additional group membership certificate that proves that the certificate holder belongs to a particular multicast group. This certificate binds the group member’s public key and IP address with the IP
address of the multicast group. When a node sends a routing control message that only group members are entitled to send it includes its group membership certificate. Thus non-members cannot impersonate a group member. A node can obtain its certificate(s) off line before it joins the network or by some out-of-band communication with the CA.

3) In MAODV, a non-member node joins the multicast tree if it is needed for the tree construction at that point of time, and it prunes itself from the tree when its presence on the tree no longer improves group connectivity. To distinguish tree nodes from other nodes, all current tree nodes are given a credential, which we refer to as the tree key. The tree key is periodically refreshed so that only nodes that are currently on the tree will possess a valid tree key.

4) A node on the multicast tree establishes pair wise shared keys with each of its immediate neighbors. This can be done using the public keys of the two nodes, as in the SSL Handshake Protocol [22]. All messages exchanged between neighboring tree nodes include a MAC computed using this pair wise key to provide strong source authentication, and prevent impersonation attacks.

5) The Group Hello packets broadcast by a group leader are digitally signed for authentication. Alternatively, a lightweight broadcast authentication scheme such as TESLA [23] can be used. Group Hello message will be known in advance to the group leader, the variant of TESLA that allows immediate authentication [23] can be used.

3) Secure Tree Key Dissemination & Verification: An important element of our authentication framework for MAODV is the use of the tree key credential for distinguishing between tree nodes and other nodes. Starting with the group leader, each tree node sends the tree key separately to each of its downstream neighbors after encrypting it using the pair wise key shared with that neighbor. This procedure is repeated recursively down the multicast tree until all the leaf nodes have received the tree key. In addition, the Group Hello packets broadcast by the group leader include the tree key authenticator f(tree key), where f is a one-way function. Nodes that receive the tree key can verify its authenticity by applying the one-way function f to it and comparing the result against the authenticator included in the latest GRPH packet. This approach is used both by tree nodes to verify the authenticity of a new tree key received from its upstream neighbor in the multicast tree, and by nodes that receive a RREP in response to a route request to verify that the replying node is a tree node.

Authenticated MAODV operations

Route Discovery and Establishment:

a) Route Discovery: When a group member (say S) wants to join the multicast group, it broadcasts a RREQ packet. When a tree node (say C) receives this RREQ it unicasts a RREP back to S possibly via some intermediate nodes. S broadcasts a signed RREQ packet that includes the IP address of the multicast group, the request ID, and its group membership certificate. The first node that receives the message (say A) signs the message using its private key before forwarding it. Each subsequent node (say B) that forwards the request checks the outer and the inner signatures, and if the verifications are successful, it replaces the outer signature with its own signature. This authentication approach is adapted from the approach used by ARAN [20]. When the request reaches the tree node C; it responds with an RREP to the node from which it received the request (B in this example). The requester S decrypts this field with its private key to extract the tree key. Using a recent Group Hello packet, S can verify whether C is a tree node using the tree key authenticator f(tree key) present in the Group Hello packet. The RREP message sent by C also includes a hop count authenticator (kNC ) and hop count anchor (k0C ). These fields correspond to the first and last elements of a one-way hash chain generated by C. They are used to authenticate the distance (in hops) from C of an intermediate node on the path from C to S. The scheme used for hop count authentication. This approach prevents an intermediate node on the path from C to S from inserting a false hop count in place of its actual distance from C. The RREP message is authenticated in the same manner as the RREQ message, i.e., each intermediate node verifies an outer and inner signature, and replaces the outer signature with its own signature before forwarding the message.

b) Link Activation: To establish a route to the multicast group, the node S sends a signed MACT packet with J (join) flag set to the neighbor (say A) with the shortest distance to the multicast tree (as indicated by the hop count field in the RREP message received from A). To prove that S was the node that initiated the route discovery, the MACT packet
includes a field which is computed by applying a one-way hash function \( H \) to the tree key and its IP address, i.e., \( H(S; \text{tree key}) \). Only the node that initiated the route discovery could have obtained the tree key from the RREP message since the tree key is encrypted with the public key of S. When the MACT (J) message is received by the tree node that sent the RREP message (say C), it checks this field to verify that S has the credentials to activate a route to the multicast group. In this each intermediate node verifies the outer and inner signatures in the message, and replaces the outer signature with its own signature before forwarding the message. As the MACT (J) message is propagated to the tree node C, each link on the path is grafted on to the multicast tree. In addition, neighboring nodes on the path establish pair wise shared keys with each other with the help of their public keys. These keys are also used to authenticate routing control messages exchanged between neighboring nodes on the multicast tree.

2) Tree Maintenance: We discuss below how the tree maintenance operations, namely tree pruning, link repair, and partition merging can be authenticated.

a) Tree Pruning: If a tree-node, which is not a member of the multicast group, notices that it has become a leaf node on the multicast tree, it prunes itself from the tree after sending a MACT packet, with P (prune) flag set, to its neighbor. To prevent impersonation attacks, this message needs to be authenticated. To this end, MACT(P) messages are authenticated using a MAC computed using the pair wise key of I and its downstream node.

b) Link Repair: For the most part, the RREQ, RREP, and MACT (J) messages exchanged in MAODV’s link repair protocol can be authenticated in the same way as these messages are authenticated in the route discovery and link activation protocols. However, one key difference is that an Extra hop count authenticator field in a RREP is used to verify the replying node’s distance (in hops) from the group leader, in addition to the hop count authenticator used for distance between the replying node and the node that sent the RREQ. Another difference between the link repair and route discovery protocols is that the initiator (say I) of the link repair is a tree node before the link breakage occurs. Hence, it will already possess the current tree key. Further, it is not necessary for a tree node to be a group member. Therefore, unlike the RREQ packet sent by a node that wants to join the multicast group, the RREQ request sent by I does not include a group membership certificate. Instead, the RREQ contains a field \( H(I; \text{tree key}) \) where H is a one-way function. After the link is repaired, I may find that its hop count from the group leader has been changed. In that case, I send a MACT message with U (update) flag set to its downstream nodes to pass this information. This MACT (U) packet is authenticated by a MAC computed using the pair wise key of I and its downstream node.

c) Partition Merging: The RREP, RREQ, and MACT (J) messages exchanged between two groups leaders for merging their partitions can be authenticated in the same fashion as the corresponding messages exchanged during route discovery and link activation. After a route is established between the two group leaders, the new group leader broadcasts a Group Hello packet with U flag set. All the tree nodes which were previously in the partition with a different group leader update their group leader address. The current tree key is also disseminated from group leader to all the tree nodes of the merged tree.

C. Security Analysis

The security of our authentication framework is derived from the following facts – (i) Any node which does not have a valid node certificate cannot participate in routing; thus the protocol is protected from outsider attacks (ii) Only nodes that possess the latest tree key can claim to be tree nodes, (iii) No tree node can claim to have a group sequence number higher than the correct one because the group sequence number is broadcast throughout the network with a signed Group Hello packet from the group leader, (iv) A tree node cannot claim to have a hop count from the group leader less than its actual distance by more than one hop; this property is achieved with the use of the hop count authenticator disseminated from the group leader to all the tree nodes, (v) No node can impersonate as a one-hop neighbor of a tree node because any one hop communication between two neighboring tree nodes is authenticated using their pair wise key. The tree key sent with the RREP is encrypted with the public key of the requesting node so that only that node can obtain the tree key.
Proposed system

Many of the attacks discussed above can be prevented by authentication framework. However, some insider attacks cannot be prevented by authentication mechanism. Some attacks against link activation like MACT (J)-MTF. In this type of attack in which malicious node can’t follow MAODV protocol. So this type of attacks can’t be prevented by authentication framework. For such type of attack additional countermeasures are required which we are proposing in the following part.

MACT (J)-MTF: The authentication framework prevents attacks from non-tree members to launch this attack by sending false RREQ (J) packets. This malicious tree-node attack prevention is done after receiving a MACT(J) packet to the every node in tree (e.g., R in Fig. 1) who is monitoring the outgoing packet traffic from the sender (i.e., T) for a particular time period. This is to show that if sender node transmits a second MACT (J) packet for activating a second route to the same multicast group. If this is seen then node R prunes the link from T and this change is also shown in its routing table. This countermeasure is useful only for few attackers in the network and they can also mitigate the effect of such attacks.

![Packet transferring in multicast tree](image)

Fig.1 Packet transferring in multicast tree

Conclusion

In this paper, we discuss the security of MAODV, which is a representative of tree-based multicast routing protocols for ad hoc networks. We identified several insider and outsider attacks on MAODV. The goal of these attacks is either to create a partition in the multicast tree or to build an energy inefficient multicast tree. Furthermore, the countermeasures have a negligible impact on MAODV performance during the normal operation of the protocol. As a part of our future work, we plan to further explore the impact of these attacks where multiple attackers collude with each other. We also plan to do security analysis of other types of multicast protocols e.g., mesh-based multicast protocols.

References


