BSMR: BYZANTINE-RESILIENT SECURE MULTICAST ROUTING in MULTIHOP WIRELESS NETWORKS

Mahamood ul Hasan MD[1], Syed Shaheen[2]

[1]Student, (M.Tech.) Department of Computer Science and Engineering, Chaitanya Engineering College, Visakhapatnam, A.P., India.
[2]Assistant Professor, Department of Computer Science and Engineering, Chaitanya Engineering College, Visakhapatnam, A.P., India.

Abstract—In this work we identify vulnerabilities of on-demand multicast routing protocols for multi-hop wireless networks and discuss the challenges encountered in designing mechanisms to defend against them. We propose BSMR, a novel secure multicast routing protocol that withstands insider attacks from colluding adversaries. Our protocol is a software-based solution and does not require additional or specialized hardware. We present simulation results which demonstrate that BSMR effectively mitigates the identified attacks.

I. INTRODUCTION

Multicast routing protocols deliver data from a source to multiple destinations organized in a multicast group. Several protocols were proposed to provide multicast services for multi-hop wireless networks. These protocols rely on node cooperation and use flooding [1], gossip [2], geographical position [3], or dissemination structures such as meshes [4], [5], or trees [6], [7].

A major challenge in designing protocols for wireless networks is ensuring robustness to failures and resilience to attacks. Wireless networks provide a less robust communication than wired networks due to frequent broken links and a higher error rate. Security is also more challenging in multi-hop wireless networks because the open medium is more susceptible to outside attacks and the multi-hop communication makes services more vulnerable to insider attacks coming from compromised nodes. Although an effective mechanism against outside attacks, authentication is not sufficient to protect against insider attacks because an adversary that compromised a node also gained access to the cryptographic keys stored on it. Insider attacks are also known as Byzantine [8] attacks and protocols able to provide service in their presence are referred to as Byzantine resilient protocols.

Previous work focused mainly on the security of unicast services. Several routing protocols [9]-[12] were proposed to cope with outsider attacks. Methods proposed to address insider threats in unicast routing include monitoring [13], multi-path routing [14] and acknowledgment-based feedback [15], [16]. The problem of secure multicast in wireless networks was less studied and only outside attacks were considered [17].

Security problems related to multicast routing can be classified in routing specific security, such as the management of the routing structure and data forwarding, and application specific security such as data confidentiality and authenticity. Solutions to the latter problem also referred to as secure group communication focus mainly on group key management [18], [19]. In this work we are concerned with multicast routing specific security.

Several aspects make the multicast communication model more challenging than its unicast counterpart. First, designing secure multicast protocols for wireless networks requires a more complex trust model, as nodes which are members of the multicast group cannot simply organize themselves in a dissemination structure without the help of other non-member nodes acting as routers.

Second, unlike unicast protocols which establish and maintain routes between two nodes, multicast protocols establish and maintain more complex structures, such as trees or meshes. For example, protocols relying on trees require additional operations such as route activation, tree pruning and tree merging. These actions do not have a counterpart in the unicast case and may expose the routing protocol to new vulnerabilities.

Third, multicast protocols deliver data from one sender to multiple receivers making scalability a major problem when designing attack-resilient protocols. In particular, solutions that offer resiliency against Byzantine attacks for unicast are not scalable in a multicast setting. For example, multi-path routing affects significantly the data dissemination efficiency, while strategies based on end-to-end acknowledgments have high overhead.

In this paper we study vulnerabilities of multicast routing protocols in multi-hop wireless networks and propose a new protocol that provides resilience against Byzantine attacks. Our main contributions are:

- We identify several aspects that make the design of secure multicast routing protocols more challenging than their unicast counterpart. They are a more complex trust model and underlying routing structure, and scalability. We also discuss potential attacks against such protocols.
- We propose BSMR, a Byzantine-resilient on-demand multicast protocol for multi-hop wireless networks. BSMR uses a selective data forwarding detection mechanism based on a reliability metric capturing adversarial behavior. Nodes determine the reliability of links by comparing the perceived data rate with the one advertised by the source. Adversarial links are avoided during the route discovery phase. BSMR also deters attacks that try to prevent or influence route establishment.
- We show through simulations that the impact of several Byzantine attacks on a previously proposed secure multicast routing protocol is considerable and cannot be ignored. We also demonstrate through simulations that our protocol BSMR mitigates the attacks, while incurring a small overhead.

The remainder of the paper is organized as follows. Section II overviews related work. Section III presents our network and system model. We discuss the attacks against tree-based multicast protocols in IV-B and present BSMR in Section V. We present experimental results in Section VI and conclude in Section VII.

II. RELATED WORK

Significant work addressed vulnerabilities of unicast routing protocols in wireless networks. Several secure routing protocols resilient to outside attacks were proposed in the last few years such as Ariadne [11], SEAD [10], ARAN [12], and the work in [9].

Wireless specific attacks such as flood rushing and wormhole were recently identified and studied. RAP [20] prevents the rushing attack by waiting for several flood requests and then randomly selecting one to forward, rather than always forwarding only the first one. Techniques to defend against wormhole attacks include *Packet Leashes* [21]which restricts the maximum transmission distance by using either tight time synchronization or location information, Truelink [22] which uses MAC level acknowledgments to infer if a link exists or not between two nodes, and the technique in [23], which relies on directional antennas.

The problem of insider threats in unicast routing was studied in [13]-[16]. Watchdog [13] relies on a node monitoring its neighbors if they forward packets to other destinations. If a node does not overhear a neighbor forwarding more than a threshold number of packets, it concludes that the neighbor is adversarial. SDT [14] uses multi-path routing to prevent a malicious node from selectively dropping data. ODSBR [15], [16] provides resilience to Byzantine attacks caused by individual or colluding nodes by detecting malicious links based on an acknowledgement-based feedback technique.

Most of the work addressing application security issues related to multicast in wireless networks focused on the problem of group key management in order to ensure data confidentiality and authenticity [24]-[28]. Work studying multicast routing specific security problems in wireless networks is scarce with the exception of the authentication framework in [17]. The framework allows MAODV to withstand several external attacks against the creation and maintenance of the multicast tree. However, it does not provide resilience against Byzantine attacks.

Multicast routing specific security was also studied in overlay networks [29]-[31]. Solutions proposed exploit overlay specific properties, which do not hold in multihop wireless networks, such as: existence of network connectivity between each pair of nodes, allowing direct probing of non-neighboring nodes, and highly redundant connectivity, ensuring that many disjoint paths exist.

III. NETWORK AND SYSTEM MODEL

A. Network Model

We consider a multi-hop wireless network where nodes participate in the data forwarding process for other nodes. We assume that the wireless channel is symmetric. All nodes have the same transmitting power and consequently the same transmission range. The receiving range of a node is identical to its transmission range.

Nodes are not required to be tamper resistant, nor to be equipped with additional hardware such as GPS receivers or tightly synchronized clocks.

B. Multicast Protocol

We assume a tree-based on-demand multicast protocol such as [6]. The protocol maintains bi-directional shared multicast trees connecting multicast sources and receivers. Each multicast group has a corresponding multicast tree. The multicast source is a special node, the group leader, whose role is to eliminate stale routes and coordinate group merges. Route freshness is indicated by a group sequence number updated by the group leader and broadcast periodically in the entire network. Higher group sequence numbers denote fresher routes.

The main operations of the protocol are route discovery, route activation and tree maintenance. During route discovery a node discovers a path to a node that is part of the multicast tree. A requester first broadcasts a route request message (RREQ) that includes the latest known group sequence number. The RREQ message is flooded in the network using a basic flood suppression mechanism and establishes reverse routes to the source of the request. Upon receiving the RREQ, a node that is part of the multicast tree and has a group sequence number at least as large as the one in the RREQ, generates a route reply

(RREP) message and unicasts it on the reverse route. The RREP message includes the last known group sequence number and the number of hops to the node that originated the RREP.

During route activation, the requester selects the freshest and shortest route (i.e., with the smallest number of hops to the multicast tree) from the routes returned by the route discovery operation. The requester activates that route by unicasting a multicast activation message.

Three main operations ensure the tree maintenance: tree pruning, broken link repair and tree merging. Tree pruning occurs when a group member that is a leaf in the multicast tree decides to leave the group. A node initiates the pruning from the tree by sending a message to its parent. The pruning message travels up the tree causing leaf nodes that are not members of the multicast group to prune themselves from the tree, until it reaches either a non-leaf node or a group member. A non-leaf group member must continue to act as a router and cannot prune itself from the multicast tree.

A node initiates a link repair procedure when the upstream link in the multicast tree breaks. If the node cannot reconnect to the tree, it means the tree is partitioned. In this case the node runs a special procedure to prune non-member leaf nodes and elect a group leader for the partition. When two partitions of the same tree reconnect, the leader of one of the partitions coordinates the merge of the partitions, suppressing the other leader.

IV. ATTACKS AGAINST MULTICAST ROUTING

A. Adversarial Model

We assume that nodes may exhibit Byzantine behavior, either alone or colluding with other nodes. Examples of such behavior include: not forwarding packets, injecting, modifying or replaying packets. We refer to any arbitrary action by authenticated nodes resulting in disruption of the routing service as Byzantine behavior, and to such an adversary as a Byzantine adversary.

We consider a three-level trust model that captures the interactions between nodes in a wireless multicast setting and defines a node's privileges: a first level includes the source which must be continually available and assumed not to be compromised; a second level consists of the multicast group member nodes, which are allowed to initiate requests for joining multicast groups; and a third level consists of non-member nodes which participate
in the routing but are not entitled to initiate group join requests. In order to cope with Byzantine attacks, even group members cannot be fully trusted.

We do not consider general attacks such as Sybil and node replication attacks. Techniques such as [32], [33] or [34], complementary to our routing protocol can be used to address these attacks. This work only considers attacks targeted against the network level. Also, preventing traffic analysis is not the goal of this work, which focuses instead on survivable routing.

B. Attacks in Multicast in Multi-Hop Wireless Networks

An adversary can attack control messages corresponding to the route discovery, route activation and tree management components of the routing protocol, or can attack data messages.

The route discovery can be disrupted by outside attackers by injecting, replaying, or modifying control packets. Malicious nodes that are not in the tree can mislead correct nodes into believing that they found and are connected to the tree. Nodes can flood the network with bogus requests for joining multicast groups. A Byzantine node can prevent a route from being established by dropping the request and/or response, or can influence the route selection by using wireless specific attacks such as wormhole and flood rushing. A Byzantine node can also modify the packets carrying the route selection metric such as hop count or node identifiers.

Outsider nodes can inject bogus route activation messages, while Byzantine nodes can prevent correct route activation messages to reach correct nodes.

Nodes can maliciously report that other links are broken or generate incorrect pruning messages resulting in correct nodes being disconnected from the network or tree partitioning. In the absence of authentication, any node can pretend to be the group leader. Although many routing protocols do not describe how to select a new group leader when needed, we note that the leader election protocol can also be influenced by attackers.

Attacks against data messages consist of eavesdropping, modifying, replaying, injecting data, or selectively forwarding data after being selected on a route. A special form of packet delivery disruption is a denial of service attack, in which the attacker overwhelms the computational, sending or receiving capabilities of a node. In general, data source authentication, integrity and encryption can solve the first attacks and are usually considered application specific security. Defending against selective data forwarding and denial of service cannot be done exclusively by using cryptographic mechanisms.

V. SECURE MULTICAST ROUTING PROTOCOL

A. BSMR Overview

Our protocol ensures that multicast data is delivered from the source to the members of the multicast group, even in the presence of Byzantine attackers, as long as the group members are reachable through non-adversarial paths and a non-adversarial path exists between a new member and a node in the multicast tree.

We use an authentication framework to eliminate outside adversaries and ensure that only authorized nodes perform certain operations (e.g., only tree nodes can perform tree operations and only group nodes can connect to the corresponding multicast tree).

BSMR mitigates inside attacks that try to prevent a node from establishing a route to the multicast tree by flooding both route request and route reply, unlike the basic multicast protocol presented in Sec. III-B which unicasts the route reply. This ensures that if an adversarial-free route exists, and then a route is established.

BSMR ensures resilience to selective data forwarding attacks by using a reliability metric that captures adversarial behavior. The metric consists of a list of link weights in which high weights correspond to low reliability. Each node in the network maintains its own weight list and includes it in each route request to ensure that a new route to the tree avoids adversarial links.

A link's reliability is determined based on the number of packets successfully delivered on that link over time. Tree nodes monitor the rate of receiving data packets and compare it with the transmission rate indicated by the source in the form of an MRATE message. If the perceived transmission rate falls below the rate indicated in the MRATE message by more than a threshold, an honest node that is a direct descendant of an adversarial node updates its weight list by penalizing the link to its parent and then tries to discover a new route to the tree.

We note that a strategy based on end-to-end acknowledgments, although shown effective in unicast [14], [16], is not scalable: As the size of the multicast group increases, ACK implosion occurs at the source, which may cause a drastic decrease in data delivery. Solutions that address the problem of feedback implosion in multicast protocols (e.g., feedback aggregation or a combination of ACK/NACK messages) were designed to operate under non-adversarial conditions; It is questionable if they will work in adversarial networks.

Without loss of generality, we limit our description to one multicast group. Below we describe the previously mentioned *authentication framework, the route discovery, the route activation, multicast tree maintenance and the selective data forwarding detection* mechanisms.

B. Authentication Framework

In order to protect from external attacks against the creation and maintenance of the multicast tree BSMR uses a framework similar with the one in [17]. The framework prevents unauthorized nodes to be part of the network, of a multicast group, or of a multicast tree. These forms of authentication correspond to the trust model described in Section IV-A. Each node authorized to join the network has a pair of public/private keys and a *node certificate* that binds its public key to its IP address. Each node authorized to join a multicast group has an additional *group certificate* that binds its public key and IP address to the IP address of the multicast group.

Nodes in the multicast tree are authenticated using a tree token, which is periodically refreshed and disseminated by the group leader in the multicast tree with the help of *pairwise shared keys* established between every direct tree neighbors. Thus, only nodes that are currently on the tree will have a valid tree token. To allow any node in the network to check that a tree node possesses a valid tree token, the group leader periodically broadcasts in the entire network a tree token authenticator *f(tree token)*, where *f* is a collision resistant one-way function. Nodes can check the validity of a given tree token by applying the function *f* to it and comparing the result with the latest received tree token authenticator.

To prevent tree nodes from claiming to be at a smaller hop distance from the group leader than they actually are, we use a technique based on a one-way hash chain. The last element of this hash chain, referred to as hop count anchor, is broadcast periodically by the group leader.

We assume that nodes have a method to determine the source authenticity of the received data (e.g., TESLA [37]). This allows a node to correctly determine the rate at which it receives multicast data.

C. Route Discovery

BSMR's route discovery allows a node that wants to join a multicast group to find a route to the multicast tree. The protocol follows the typical route request/route reply procedure used by on-demand routing protocols with several differences. To prevent outsiders from interfering, all route discovery messages are authenticated using the public key corresponding to the network certificate. Only group authenticated nodes can initiate route requests and the group certificate is required in each request. Tree nodes use the tree token to prove their current tree status.

Several mechanisms are used to address internal attackers: (a) both route request and route reply are flooded in order to ensure that, if an adversarial-free path exists, it will be found; (b) the path selection relies on the weights list carried in the response flood and allows the requester to select a non-adversarial path; (c) the propagation of weights and path accumulation is performed using an onion-like signing to prevent forwarding nodes from modifying the path carried in the response.

The requesting node broadcasts a route request (RREQ) message that includes the node identifier and its weight list, the multicast group identifier, the last known group sequence number, and a request sequence number. The RREQ message is flooded in the network until it reaches a tree node that has a group sequence number at least as great as that in the RREQ. Only new requests are processed by intermediate nodes.

When a tree node receives for the first time a RREQ from a requester and the node's group sequence number is at least as great as that contained in the RREQ, it initiates a response. The node broadcasts a route reply (RREP) message that includes that node identifier, its recorded group sequence number, the requester's identifier, a response sequence number, the group identifier and the weight list from the request message. To prove its current tree node status, the node also includes in the response the

current tree token, encrypted with the requester's public key. The RREP message is flooded in the network until it reaches the requester, using the following *weighted flood suppression* mechanism. Tree nodes with a group sequence number at least as great as that in the RREP ignore RREP messages. Otherwise, a node computes the total path weight by summing the weight of all the links on the specified path from the multicast tree to itself. If the total weight is less than any previously forwarded matching response (same requester, multicast group and response sequence number), and all the signatures accumulated on the reply are valid, the node appends its identifier to the end of the message, assigns the entire message and rebroadcasts it. As the RREP message propagates across the network, nodes establish the *forward route* by setting pointers to the node from which the RREP was received. Although several tree nodes may initiate the response flood, the weighted flood suppression mechanism insures the communication overhead is equivalent to only one flood.

When the requester receives a response, it performs the same computation as an intermediate node during the response propagation. The requester updates its information upon receipt of a valid response that contains a better path according to our reliability metric.

D. Multicast Route Activation

The requester signs and unicasts on the selected route a multicast activation (MACT) message that includes its identifier, the group identifier and the sequence number used in the route request phase. An intermediate node on the route checks if the signature on MACT is valid and if MACT contains the same sequence number as the one in the original RREQ message. The node then adds to its list of tree neighbors the previous node and the next node on the route as downstream and upstream neighbors, respectively, and sends the MACT message along the forward route.

The requester and the nodes that received the MACT message could be prevented from being grafted to the tree by an adversarial node, selected on the forward route, which drops the MACT message. To mitigate the attack, these nodes will start a *waiting to connect timer (WTC_Timer)* upon whose expiration nodes isolate a faulty link and initiate Route Discovery (Fig. 3). The timer expires after a value proportional to a node's hop distance to the tree, in the hope that the nodes closer to the tree will succeed in avoiding the adversarial node and will manage to connect to the tree. After a node becomes aware of its expected receiving data rate, it cancels its *WTC_Timer* and behaves as described in Sec. V-F.

E. Multicast Tree Maintenance

The tree maintenance phase ensures the correct operation of the protocol when confronted with events such as pruning, link breakage and node partitioning. Routing messages exchanged by tree neighbors, such as pruning messages (described in Sec. III-B) are authenticated using the pairwise keys shared between tree neighbors. If a malicious node prunes itself even if it has a subtree below it, the honest nodes in this subtree will reconnect to the tree following the procedure described in Sec. V-F.

The link repair procedure is initiated by nodes that detect a broken link and is similar with Route Discovery.

The group leader periodically broadcast in the entire network a signed Group Hello message that contains the current group sequence number, the tree token authenticator and the hop count anchor (described in Sec. V-B).

F. Selective Data Forwarding Detection

The source periodically signs and sends in the tree a multicast rate (MRATE) message that contains its data transmission rate p_0. As this message propagates in the multicast tree, nodes may add their perceived transmission rate to it. The information in the MRATE message allows nodes to detect if tree ancestors perform selective data forwarding attacks. Depending on whether their perceived rate is within acceptable limits of the rate in the MRATE message, nodes alternate between two states. The initial state of a node is *disconnected*; After it joins the multicast group and becomes aware of its expected receiving data rate, the node switches to the *Connected* state. Upon detecting a selective data forwarding attack, the node switches back to the *Disconnected* state.

A network operating normally exhibits some amount of natural "loss", which may cause the rate perceived by a node to be smaller than the rate perceived by its tree parent. This natural rate decrease is cumulative as data travels further away from the source. We define a threshold σ as the upper bound for the tolerable loss rate on a single link. If a node's perceived rate is smaller than the last recorded rate in MRATE by more than σ, the node concatenates its identifier and its rate to MRATE and signs the entire message before forwarding it. These added rates serve as proofs that nodes which previously forwarded the MRATE message did not perceive losses much larger than natural losses.

In order to prevent a malicious node from introducing a rate decrease significantly larger than σ, we use another threshold $\Delta > \sigma$. Upon receiving an MRATE message, each node first checks if the difference between the last rate in MRATE and the node's perceived rate is greater thanΔ. If so, this indicates that there exists at least an adversarial node in between this node and the node that added the last rate to MRATE. The first honest node that notices a difference larger than Δ incriminates the link to its tree

parent as faulty (by using an exponential weight increase scheme) and assumes responsibility for finding a new route to the tree. The nodes in the subtree below this node will notice there is a "gap" greater than Δ between the rates included in MRATE; They will defer taking any action to isolate the faulty link for an amount of time proportional to the distance from the node that already started the repair procedure, in the hope that the nodes closer to the faulty link will succeed in isolating it. Upon detecting that the expected data packet rate has been restored, nodes cancel the repair procedure.

Figures 1, 2 and 3 describe how a *Connected* node reacts to the following events, respectively: (1) receipt of an MRATE message, (2) timeout of the *MRATE_Timer*, and (3) timeout of the *WTC_Timer*. p_{node} denotes the rate at which the node receives packets from its tree parent.

Tree nodes expect to periodically receive MRATE messages, otherwise the *MRATE_Timer* will expire. Note that each tree node stores the latest received MRATE message and uses it to re-initiate the propagation of MRATE if *MRATE_Timer* expires. When *MRATE_Timer* expires, a node compares its perceived rate with the expected rate from the stored MRATE message.

VI. EXPERIMENTAL RESULTS

To the best of our knowledge, the only previous secure on-demand multicast protocol is the one to which we refer as A-MAODV. Although A-MAODV withstands several external attacks against the creation and maintenance of the multicast tree, it does not provide resilience against Byzantine attacks. In this section, we study the effect of several Byzantine attacks on the performance of A-MAODV and demonstrate the effectiveness of BSMR in mitigating the attacks.

We implemented BSMR using the ns2 [38] network simulator, starting from an MAODV implementation [39]. We assumed the protocol uses RSA [40] with 1024-bit keys for public key operations, AES [41] with 128-bit keys for symmetric encryptions and HMAC [42] with SHA1 as the message authentication code.

The values used for σ and Δ were 10% and 20% of the source's rate, respectively. We developed a protocol independent Byzantine attack simulation module for ns2.

A. Experimental Methodology

To capture a protocol's effectiveness in delivering data to the multicast group, we used as a performance metric the packet delivery ratio (PDR), defined as:

$$PDR = \frac{Pr}{Ps.N}$$

where Pr is the number of data packets received by multicast group members, Ps is the number of data packets sent by the source and N is the size of the group.

Because external attacks can be prevented using the authentication framework described in Sec. V-B, in this paper we focus on the following two Byzantine attacks:

- *Black hole attack*: This is a selective data forwarding attack, in which adversaries only forward routing control packets, while dropping all data packets.
- *Flood rushing attack*: The attack exploits the flood duplicate suppression technique used by many wireless routing protocols. By "rushing" an authenticated flood through the network before the flood traveling through a legitimate route, a Byzantine adversary ends up controlling many routes. The attack was implemented by simply ignoring the small randomized delays which are normally required to reduce the number of collisions.

In order to quantify the impact of adversarial positioning, we consider the following scenarios:

- *Random placement*: Adversaries are placed randomly in the simulation area;
- *Strategic placement*: Adversaries are placed strategically around the multicast source, equidistant on a circle with radius of 200 meters.

To study the influence of whether the adversaries explicitly join the multicast group and the order of joining, we consider two scenarios:

- **NJOIN**: Adversarial nodes do not join the group;
- **JOIN**: Adversarial nodes explicitly join the group before any of the honest members join. The adversaries are considered group members in the formula for PDR.

We chose these test case scenarios in order to study the impact of the attacks under a light set of conditions (adversaries are placed randomly, or they do not explicitly join the multicast group) and under a more extreme set of conditions (adversaries are placed strategically, or they join the group before honest nodes do).

B. Simulation Setup

We performed simulations using the ns2 network simulator [38]. Nodes were set to use 802.11 radios with 2 Mbps bandwidth and 250 meters nominal range. The simulated time was 600 seconds. We randomly placed 100 nodes within a 1500 by 1500 meter area and the multicast source in the center of the area at coordinates (750,750). We experimented with different values for group size, number of adversaries and speed. Due to lack of space, we only include results for medium-sized groups (30 and 50), for adversaries between 14% - 66% of the group size and for "max" speeds of 0 and 5 m/s.

Group members join the group sequentially in the beginning of the simulation, at 3-second intervals. Then the source transmits multicast data for 600 seconds at a rate of 5 packets per second, each packet of 256 bytes. The members stay in the group until the end of the simulation. Adversaries added to the network replace honest nodes, thus modeling the capture of honest nodes.

Node movement pattern is defined by the random waypoint model. Data points are averaged over 30 different random environments and over all group members.

We use the following additional notation:
– p_{node} is the rate at which this node receives packets from its tree parent.
– hop_distance($node_k$, node) is the hop distance between the node that added the k-th rate to MRATE and this node.

Receipt of MRATW = (p_0, (id_1,d_1,p_1),…,(id_k,d_k,p_k))

1. **If** (this is the first MRATE message received) **then**
2. switch to *Connected* state
3. cancel *WTC_Timer*
4. store MRATE message and cancel *MRATE_Timer*
5. **if** ((state = *Conneted)* AND (W*TC_Timer* ≠ PENDING)) **then**
6. **if** ((MRATE contains a "gap" larger than Δ) **OR** (MRATE Contains a "gap" larger than σ that corresponds to a single hop)) **then**
7. start *WTC_Timer* timer
8. forward MRATE
9. **return**
10. **else if** (*WTC_Timer* = PENDING) **then**
11. **if** ((MRETE contains a "gap" larger than Δ) OR (MRATE contains a "gap" larger tha σ that corresponds to a single hop)) **then**
12. MRATE – cat_and_sign(MRATE, id_{node}, d_{node}, p_{node}))
13. forward MRATE
14. **return**
15. **else**
16. cancel *WTC_Timer*
17. switch to *Connected* stae
18. **if** ((p_k – p_{node} > Δ) OR ((p_k – p_{node} > σ) AND (hop_distance(node+k+, node) =1)) **then**
19. MRATE = cat_and_sign(MRATE, (id_{node}, d_{node},p_{node}))
20. **if** (*WTC_Timer* = PENDING) **then**
21. cancel *WTC_Timer*
22. switch to *Disconnected* state
23. increase weight of the link to the parent
24. initiate Route Discovery
25. **else if** (p_k – p_{node} > σ) **then**
26. MRATE = cat_and_sign(MRATE, (id_{node}, d_{node} , p_{node}))
27. forward MRATE message
28. start *MRATE Timer*

Fig. 1: receipt of MRATE = (p_0, (id_1,p_1), . . . , (id_k, p_k))

1. **If** state = *Connected* and WTC_Timer ≠ PENDING **then**
2. retrieve stored MRATE = (p_0, (id_1,p_1),…,(id_k,p_k))
3. **if** p_k –p_{node} > Δ **then**
4. MRATE = cat_and_sign(MRATE, (id_{node}, p_{node}))
5. switch to *Disconnected* state
6. increase weight of the link to the parent
7. initiate Route Discovery
8. **else if** p_k – p_{node} > σ **then**
9. MRATE = cat_and_sign(MRATE, (id_{node},p_{node}))
10. forward MRATE message

Fig. 2: timeout of *MRATE_Timer*

1. switch to *Disconnected* state
2. increase weight of the link to the parent
3. initiate Route Discovery

Fig. 3: timeout of *WTC_Timer*

C. Attack Resilience

We evaluate the PDR as a function of the number of adversaries, for different group sizes and levels of mobility. Each graph illustrates the effect of the black hole attack with and without flood rushing.

1) Impact of Adversarial Placement: When adversaries are randomly placed (Fig. 4), for the same group size, the PDR of A-MAODV decreases as the number of adversaries increases. For the same number of adversaries, it also decreases as we increase the group size. However, random adversarial placement causes the number of group members in the subtree below an adversary to be low; thus a relatively large number of adversaries are needed to cause a significant disruption (e.g., 30 adversaries for a group of size 50 cause a PDR drop below 50%). In the presence of flood rushing, the PDR decreases further because adversaries actively try to get selected themselves as part of the tree.

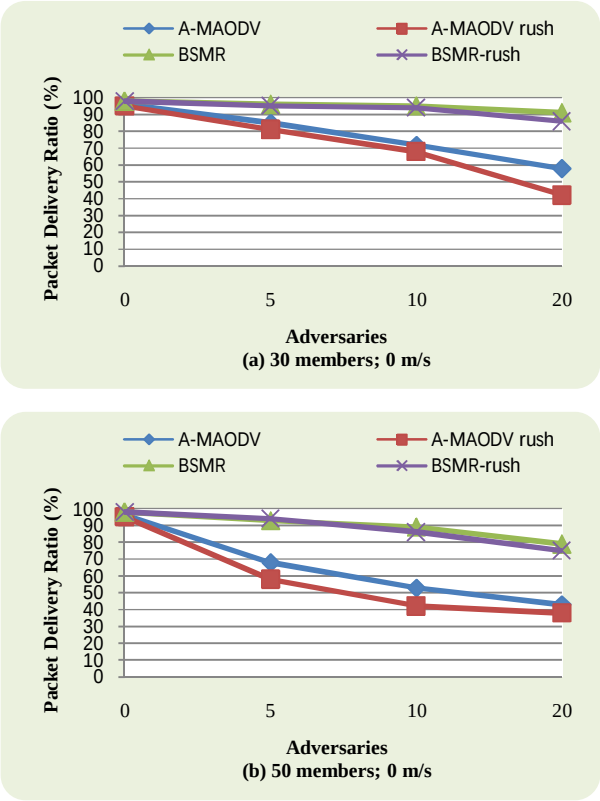
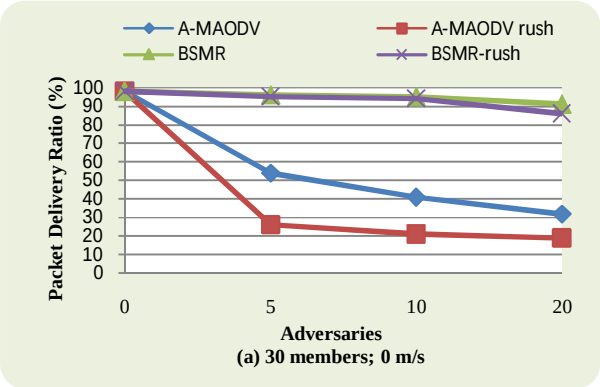


Fig. 4: Black hole attack and flood rushing combined with black hole: **Random placement (NJOIN)**. (a) 30 members; 0 m/s. (b) 50 members; 0m/s.



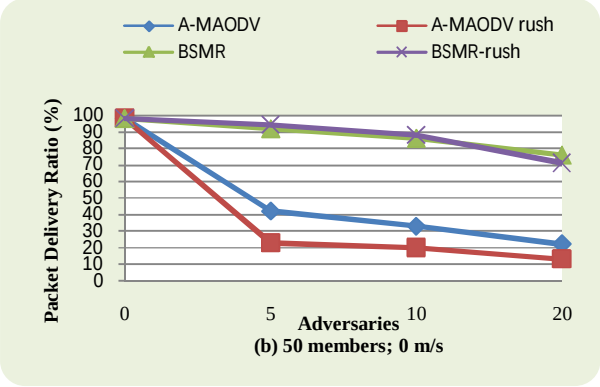


Fig. 5: Black hole attack and flood rushing combined with black hole: **Strategic placement (NJOIN)**. (a) 30 members; 0 m/s. (b) 50 members; 0m/s.

We notice that, for A-MAODV, increasing the nodal speed does not have a negative effect on the PDR; On the contrary, at higher speeds we even see a slight increase in PDR. The effect of link breaks due to mobility is compensated by the fact that group members get a chance to reconnect to the multicast tree in a different position, possibly connected to the source through an adversarial-free path. For the same reason, the effect of flood rushing is diminished as the nodal speed increases.

BSMR is almost unaffected by the black hole attack (Fig. 4). The PDR drops by less than 10% even in the presence of 20 adversaries. In addition, the influence of flood rushing is unnoticeable. This shows that BSMR's strategy that includes the processing of all response flood duplicates and the metric capturing past behavior of adversarial nodes, is effective against flood rushing. Mobility causes a slight PDR decrease, which is natural because higher speeds will cause more link breaks.

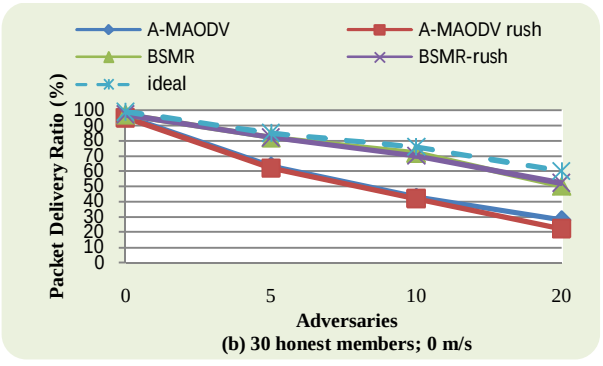
In the previous experiments, the adversaries were randomly placed. Fig. 5 shows the results when adversaries are strategically positioned as described in Section VI-A.

When adversaries are strategically positioned (Fig.5) as described in Sec. VI-A, we notice for A-MAODV a drastic drop in the PDR. For example, at 0 m/s, when the group size is 30, only 5 adversaries (representing 16% of the group size) are able to reduce the PDR to 25% by executing the black hole attack with flood rushing. This is a direct consequence of the fact that an adversary is now selected in the tree closer to the root and the subtree below it may potentially contain many group members. For the same reason, the negative effect of the flood rushing attack is now emphasized when compared to the random placement case. We conclude that strategic adversarial positioning has a crippling effect on the performance of A-MAODV.

On the contrary, the effect of strategic adversarial positioning on BSMR is minor (Fig. 5). Like for random placement, the PDR drops by less than 10% even in the presence of 20 adversaries, at low nodal speeds. When more adversaries are present, we see a slightly larger PDR decrease because there are less available honest nodes left in the network to serve as intermediaries for the group members. The resilience of BSMR to attacks that otherwise have a devastating effect on A-MAODV validates the effectiveness of BSMR's approach.

2) Impact of Explicit Join and Join Order: To analyze the impact of explicit join of adversaries to the multicast group (JOIN), as compared to the NJOIN case, we look again at the cases where adversaries are randomly and strategically placed (Fig. 6 and 7). Figures include the ideal PDR (labeled ideal), which would be obtained if every honest group member receives all packets sent by the source. Attack effectiveness should be read as the difference between the ideal line and a protocol's PDR line; Thus, attack resilience will appear as a protocol line that stays parallel to the ideal line.

For random adversarial placement in Fig. 6, just like in the NJOIN case, the PDR decreases as the number of adversaries increases. However, we see a major difference from the NJOIN case: When the adversaries explicitly join the group before the honest nodes join, the impact of flood rushing is minimal because the adversaries are already part of the group and rushing control packets does not provide any additional benefit. On the contrary, in this case flood rushing may actually improve the PDR because, by rushing control packets, adversaries may help legitimate nodes to find routes faster.



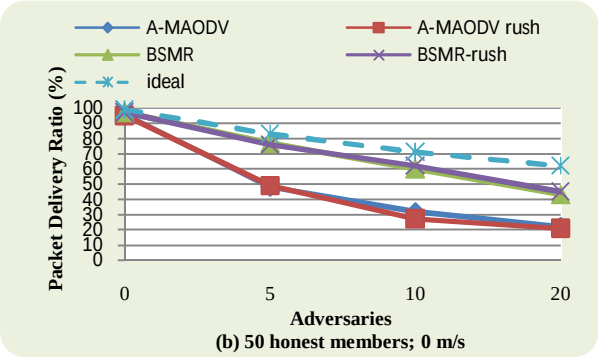
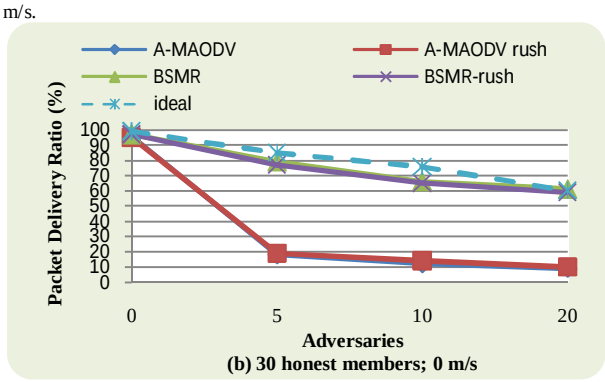


Fig. 6: Black hole attack and flood rushing combined with blackhole: **Random placement (JOIN)**. (a) 30 members; 0 m/s. (b) 50 members; 0 m/s.



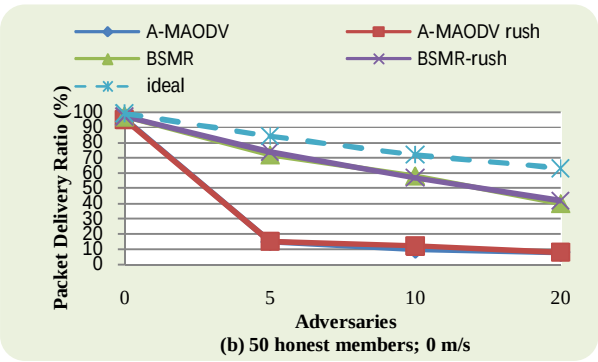


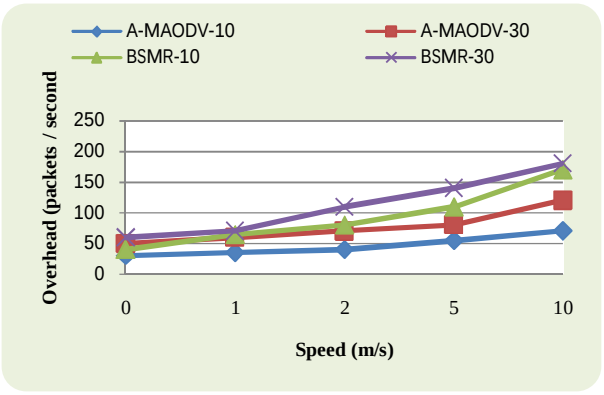
Fig. 7: Black hole attack and flood rushing combined with black hole: **Strategic placement (JOIN)**. (a) 30 members; 0 m/s. (b) 50 members; 0m/s.

A drastic drop in the PDR is observed for A-MAODV when adversaries are placed strategically (Fig. 7). We conclude that strategic positioning has a more crippling effect on the performance of A-MAODV even when adversaries explicitly join the multicast group. For both random and strategic adversarial placement, BSMR is barely affected by the attacks: In most cases the PDR line remains almost parallel to the ideal line, which shows little degradation occurs as the number of adversaries increases. The impact of the attacks on BSMR increases slightly when a large number of adversaries have joined the group, because there are less available honest nodes left in the network to serve as intermediaries for honest group members. We conclude that BSMR's strategy is effective in the JOIN case as well.

D. Protocol Overhead

In a non-adversarial scenario (Fig. 8(a)), BSMR has higher overhead than A-MAODV because the route reply is flooded, and because of the extra MRATE packets broadcast periodically. BSMR's overhead becomes more noticeable especially at higher levels of mobility.

For an adversarial setting (Fig. 8(b)), we focus on a strong attack configuration: Black hole with strategic adversarial placement. For the NJOIN case, BSMR's additional overhead compared to A-MAODV grows slowly as the number of adversaries increases (from 40 more packets/sec. for 0 adversaries to 55 more packets/sec. for 20 adversaries). For the JOIN case, the additional overhead does not grow as we increase the number of adversaries, indicating that BSMR incurs little extra overhead over the non-adversarial case.



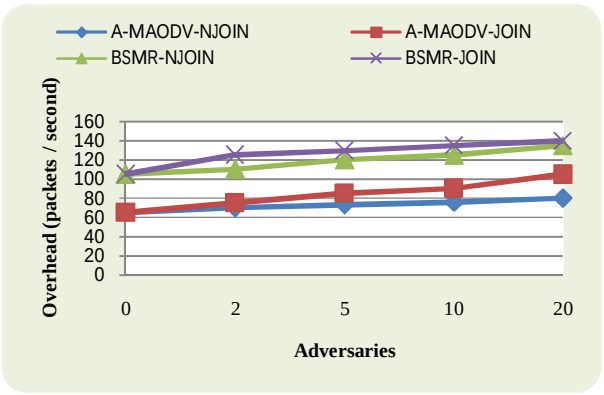


Fig. 8: BSMR overhead.
(a) Nonadversarial scenario (group size □ {10; 30}).
(b) Attack scenario (group size = 30;2 m/s, strategic placement).

VII. CONCLUSION

In this paper we have discussed several aspects that make designing attack-resilient multicast routing protocols for multi-hop wireless networks more challenging when compared to their unicast counterpart. A more complex

trust model and underlying structure for the routing protocol make solutions tailored for unicast settings not applicable for multicast protocols.

We have proposed BSMR, a routing protocol which relies on novel general mechanisms to mitigate Byzantine attacks. BSMR identifies and avoids adversarial links based on a reliability metric that captures adversarial behavior. Our experimental results show that BSMR's strategy is effective against strong insider attacks such as black holes and flood rushing.

ACKNOWLEDGEMENTS

The first author would like to thank R˘azvan Mus˘aloiu-E. for fruitful "copy-room" discussions in the early stages of this work. This work is supported by National Science Foundation CyberTrust Award No. 0545949. The views expressed in this research are not endorsed by the National Science Foundation.

REFERENCES

[1] Y. B. Ko and N. H. Vaidya, "*Flooding-based geocasting protocols for mobile ad hoc networks*," Mob. Netw. Appl., vol. 7, no. 6, pp. 471–480, 2002.

[2] R. Chandra, V. Ramasubramanian, and K. Birman, "*Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks*," in Proc. of ICDCS, 2001.

[3] Y.-B. Ko and N. H. Vaidya, "*GeoTORA: a protocol for geocasting in mobile ad hoc networks*," in Proc. of ICNP. IEEE Computer Society, 2000, p. 240.

[4] E. L. Madruga and J. J. Garcia-Luna-Aceves, "*Scalable multicasting: the core-assisted mesh protocol*," Mob. Netw. Appl., vol. 6, no. 2, pp. 151–165, 2001.

[5] S. J. Lee, W. Su, and M. Gerla, "*On-demand multicast routing protocol in multihop wireless mobile networks*," Mob. Netw. Appl., vol. 7, no. 6, pp. 441–453, 2002.

[6] E. Royer and C. Perkins, "*Multicast ad-hoc on-demand distance vector (MAODV) routing*," in Internet Draft, July 2000.

[7] J. G. Jetcheva and D. B. Johnson, "*Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks*." In Proc. Of MobiHoc, 2001, pp. 33–44.

[8] L. Lamport, R. Shostak, and M. Pease, "*The Byzantine generals problem*," in Advances in Ultra-Dependable Distributed Systems. IEEE Computer Society Press, 1995.

[9] P. Papadimitratos and Z. Haas, "*Secure routing for mobile ad hoc networks*," in Proc. of CNDS, January 2002, pp. 27–31.

[10] Y.-C. Hu, D. B. Johnson, and A. Perrig, "*SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks*," in Proc. of WMCSA, June 2002.

[11] Y.-C. Hu, A. Perrig, and D. B. Johnson, "*Ariadne: A secure on-demand routing protocol for ad hoc networks*," in Proc. Of MOBICOM, September 2002.

[12] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer, "*A secure routing protocol for ad hoc networks*," in Proc. Of ICNP, November 2002.

[13] S. Marti, T. Giuli, K. Lai, and M. Baker, "*Mitigating routing misbehavior in mobile ad hoc networks*," in Proc. of MOBICOM, August 2000.

[14] P. Papadimitratos and Z. Haas, "*Secure data transmission in mobile ad hoc networks*," in Proc. of WiSe, 2003, pp. 41–50.

[15] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "*An on-demand secure routing protocol resilient to byzantine failures*," in Proc. of WiSe'02. ACM Press, 2002.

[16] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "*On the survivability of routing protocols in ad hoc wireless networks*," in Proc. of SecureComm'05. IEEE, 2005.

[17] S. Roy, V. G. Addada, S. Setia, and S. Jajodia, "*Securing MAODV: Attacks and countermeasures*," in Proc. 2nd IEEE Int'l. Conf. SECON. IEEE, 2005.

[18] C. Zhang, B. DeCleene, J. Kurose, and D. Towsley, "*Comparison of inter-area rekeying algorithms for secure wireless group communications*," Perform. Eval., vol. 49, no. 1-4, 2002.

[19] K. H. Rhee, Y. H. Park, and G. Tsudik, "*An architecture for key management in hierarchical mobile ad hoc networks*," Journal of Communication and Networks, vol. 6, no. 2, June 2004.

[20] Y.-C. Hu, A. Perrig, and D. B. Johnson, "*Rushing attacks and defense in wireless ad hoc network routing protocols*," in Proc. of WiSe. ACM, 2003.

[21] ——, "*Packet leashes: A defense against wormhole attacks in wireless ad hoc networks*," in Proc. of INFOCOM, April 2003.

[22] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "*Truelink: A practical countermeasure to the wormhole attack in wireless networks*," in Proc. of ICNP'06. IEEE, 2006.

[23] L. Hu and D. Evans, "*Using directional antennas to prevent wormhole attacks*," in Proc. of NDSS, 2004.

[24] D. Bruschi and E. Rosti, "*Secure multicast in wireless networks of mobile hosts: protocols and issues*," Mobile Networks and Applications, vol. 7, no. 6, pp. 503–511, 2002.

[25] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz, "*Secure multicast groups on ad hoc networks*," in Proc. of SASN'03. ACM Press, 2003, pp. 94–102.

[26] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "*Gkmpan: An efficient group rekeying scheme for secure multicast in ad-hoc networks*," in Proc. of Mobiquitos'04. IEEE, 2004, pp. 42–51.

[27] L. Lazos and R. Poovendran, "*Power proximity based key management for secure multicast in ad hoc networks*," 2005, aCM Journal on Wireless Networks (WINET).

[28] R. Balachandran, B. Ramamurthy, X. Zou, and N. Vinodchandran, "*CRTDH: an efficient key agreement scheme for secure group communications in wireless ad hoc networks*," in Proc. of ICC 2005, vol. 2, 2005, pp. 1123– 1127.

[29] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan, "*Resilient multicast using overlays*," SIGMETRICS Perform. Eval. Rev., vol. 31, no. 1, pp. 102–113, 2003.

[30] V. Pappas, B. Zhang, A. Terzis, and L. Zhang, "*Fault-tolerant data delivery for multicast overlay networks*," in Proc. of ICDCS '04, 2004, pp. 670–679.

[31] L. Xie and S. Zhu, "*Message dropping attacks in overlay networks: Attack detection and attacker identification*," in Proc. SecureComm '06. IEEE and Create-NET, 2006.

[32] J. Newsome, E. Shi, D. Song, and A. Perrig, "*The Sybil attack in sensor networks: analysis & defenses*," in Proc. of IPSN '04. New York, NY, USA: ACM Press, 2004, pp. 259–268.

[33] C. Piro, C. Shields, and B. N. Levine, "*Detecting the Sybil attack in mobile ad hoc networks*," in Proc. SecureComm, 2006.

[34] B. Parno, A. Perrig, and V. D. Gligor, "*Distributed detection of node replication attacks in sensor networks*." in IEEE Symposium on Security and Privacy, 2005, pp. 49–63.

[35] R. Curtmola and C. Nita-Rotaru, "*Secure multicast routing in wireless networks*," to be published in ACM MC2R, 2006.

[36] K. Tang, K. Obraczka, S.-J. Lee, and M. Gerla, "*A reliable, congestion-controlled multicast transport protocol in multimedia multi-hop networks*," in Proc. of IEEE WPMC'02, 2002.

[37] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "*Efficient and secure source authentication for multicast*," in Proc. Of NDSS'01, 2001.

[38] "The network simulator - ns2," http://www.isi.edu/nsnam/ns/.

[39] Y. Zhu and T. Kunz, "*MAODV implementation for NS-2.26*," Carleton University, Technical Report SCE-04-01.

[40] R. L. Rivest, A. Shamir, and L. M. Adleman, "*A method for obtaining digital signatures and public-key cryptosystems*," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[41] Advanced Encryption Standard (AES). National Institute for Standards and Technology (NIST), 2001, no. FIPS 197.

[42] The Keyed-Hash Message Authentication Code (HMAC). NIST, 2002, no. FIPS 198.