Secure High-Throughput Multicast Routing in Wireless Mesh Networks

Jing Dong, Reza Curtmola, *Member, IEEE,* and Cristina Nita-Rotaru, *Senior Member, IEEE*

Abstract—Recent work in multicast routing for wireless mesh networks has focused on metrics that estimate link quality to maximize throughput. Nodes must collaborate in order to compute the path metric and forward data. The assumption that all nodes are honest and behave correctly during metric computation, propagation, and aggregation, as well as during data forwarding, leads to unexpected consequences in adversarial networks where compromised nodes act maliciously.

In this work we identify novel attacks against high-throughput multicast protocols in wireless mesh networks. The attacks exploit the local estimation and global aggregation of the metric to allow attackers to attract a large amount of traffic. We show that these attacks are very effective against multicast protocols based on high-throughput metrics. We conclude that aggressive path selection is a double-edged sword: While it maximizes throughput, it also increases attack effectiveness in the absence of defense mechanisms. Our approach to defend against the identified attacks combines measurement-based detection and accusation-based reaction techniques. The solution accommodates transient network variations and is resilient against attempts to exploit the defense mechanism itself. A detailed security analysis of our defense scheme establishes bounds on the impact of attacks. We demonstrate both the attacks and our defense using ODMRP, a representative multicast protocol for wireless mesh networks, and SPP, an adaptation of the well-known ETX unicast metric to the multicast setting.

Index Terms—Wireless mesh networks, high-throughput metrics, secure multicast routing, metric manipulation attacks, Byzantine attacks.

1 Introduction

Wireless mesh networks (WMNs) emerged as a promising technology that offers low-cost high-bandwidth community wireless services. A WMN consists of a set of stationary wireless routers that form a multi-hop backbone, and a set of mobile clients that communicate via the wireless backbone. Numerous applications envisioned to be deployed in WMNs, such as webcast, distance learning, online games, video conferencing, and multimedia broadcasting, follow a pattern where one or more sources disseminate data to a group of changing receivers. These applications can benefit from the service provided by multicast routing protocols.

Multicast routing protocols deliver data from a source to multiple destinations organized in a multicast group. In the last few years, several protocols [2]–[8] were proposed to provide multicast services for multi-hop wireless networks. These protocols were proposed for mobile ad hoc networks (MANETs), focusing primarily on network connectivity and using the number of hops (or *hop count*) as the route selection metric. However, it has been shown that using hop count as routing metric can result in selecting links with poor quality on the path, negatively impacting the path throughput [9], [10]. Instead, given the stationary nature of WMNs, recent protocols [11], [12] focus on maximizing path throughput by selecting paths based on metrics that capture the quality of the wireless links [10], [13]–[16]. We refer to such metrics as *link-quality* metrics or *high-throughput* metrics, and to protocols using such metrics as *high-throughput protocols*[1].

In a typical high-throughput multicast protocol, nodes periodically send probes to their neighbors to measure the quality of their adjacent links. During route discovery, a node estimates the cost of the path by combining its own measured metric of adjacent links with the path cost accumulated on the route discovery packet. The path with the best metric is then selected. High-throughput protocols require the nodes to collaborate in order to derive the path metric, thus relying on the assumption that nodes behave correctly during metric computation and propagation. However, this assumption is difficult to guarantee in wireless networks that are vulnerable to attacks coming from both insiders and outsiders, due to the open and shared nature of the medium and the multi-hop characteristic of the communication. An aggressive path selection introduces new vulnerabilities and provides the attacker with an increased arsenal of attacks leading to unexpected consequences. For example, adversaries may manipulate the metrics in order to be selected on more paths and to draw more traffic, creating opportunities for attacks such as data dropping, mesh partitioning, or traffic analysis.

Although there has been extensive work on using high-throughput metrics to improve performance in wireless networks, work studying the security implications of this choice is relatively scarce. Previous work primarily focused on vulnerabilities of unicast routing protocols that use hop count as a metric [17]–[24]. Secure wireless multicast was less studied, and the existing work [25], [26] focused primarily on using hop count metric in tree-based protocols.

In this work, we study the security implications of using high-throughput metrics for multicast in WMNs. In particular, we use ODMRP [6] as a representative multicast protocol for WMNs. We selected ODMRP, as it is a mesh-based protocol, which has

A preliminary version of this paper appeared in [1].
Manuscript received ???; revised ???; accepted ???; published online ???

- *J. Dong is with the Department of Computer Science, Purdue University, West Lafayette, IN 47907. E-mail: dongj@cs.purdue.edu.*
- *R. Curtmola is with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102. E-mail: crix@njit.edu.*
- *C. Nita-Rotaru is with the Department of Computer Science, Purdue University, West Lafayette, IN 47907. E-mail: crisn@cs.purdue.edu.*

[1] Note that the term high-throughput protocols only refers to protocols that select paths based on link quality. It is not necessary that a high-throughput protocol can deliver a high throughput (in terms of kbps), which is determined by the wireless link bandwidth and source-destination distance, etc.

the potential to be more attack resilient. We focus on the SPP [11] metric based on the well-known ETX [10] unicast metric, since it was shown to outperform all the other multicast metrics for ODMRP demonstrated in [11]. To the best of our knowledge, this is the first paper to examine vulnerabilities of high-throughput metrics in general, and in multicast protocols for WMNs in particular. We summarize our contributions:

- We identify a class of severe attacks against multicast protocols that exploit the use of high-throughput metrics, including *local metric manipulation* (LMM) and *global metric manipulation* (GMM). We show that aggressive path selection is a double-edged sword: It leads to increased throughput, but it also leads to devastating effects in the presence of attacks. For example, our simulations show that 5 GMM attackers can cause the same attack impact as 20 packet dropping attackers.
- We propose a secure high-throughput multicast protocol S-ODMRP that incorporates a novel defense scheme RateGuard. RateGuard combines measurement-based detection and accusation-based reaction techniques to address the metric manipulation and packet dropping attacks. To prevent attackers from exploiting the defense mechanism itself, RateGuard limits the number of accusations that can be generated by a node. RateGuard also adopts a temporary accusation mechanism that accommodates false positive accusations that may be caused by transient network variations.
- We perform a detailed security analysis and establish bounds on the impact of the attacks under our defense scheme. Extensive simulations with ODMRP and SPP confirm our analysis and show that our strategy is very effective in defending against the attacks, while incurring a low overhead.

2 High-Throughput Multicast Routing

We consider a multi-hop wireless network where nodes participate in the data forwarding process for other nodes. We assume a mesh-based multicast routing protocol, which maintains a mesh connecting multicast sources and receivers. Path selection is performed based on a metric designed to maximize throughput. Below, we provide an overview of high-throughput metrics for multicast, then describe in details how such metrics are integrated with mesh-based multicast protocols.

2.1 High-Throughput Metrics

Traditionally, routing protocols have used hop count as a path selection metric. In static networks however, this metric was shown to achieve sub-optimal throughput because paths tend to include lossy wireless links [10], [27]. As a result, in recent years the focus has shifted toward high-throughput metrics that seek to maximize throughput by selecting paths based on the quality of wireless links (*e.g.*, ETX [10], PP [15], [27], RTT [14]). In such metrics, the quality of the links to/from a node's neighbors is measured by periodic probing. The metric for an entire path is obtained by aggregating the metrics reported by the nodes on the path.

Several high-throughput metrics for multicast were proposed in [11]. All of these metrics are adaptations of unicast metrics to the multicast setting by taking into account the fundamental differences between unicast and multicast communication. Transmissions in multicast are less reliable than in unicast for several reasons. In unicast, a packet is sent reliably using link-layer unicast transmission, which involves link-layer acknowledgments and possibly packet retransmissions; in multicast, a packet is sent unreliably using link-layer broadcast, which does not involve link layer acknowledgments or data retransmissions. Moreover, unicast transmissions are preceded by a RTS/CTS exchange; in multicast there is no RTS/CTS exchange, which increases collision probability and decreases transmission reliability. Many metrics for unicast routing minimize the medium access time, while metrics for multicast capture in different ways the packet delivery ratio.

All the high-throughput multicast metrics proposed in [11] showed improvement over the original path selection strategy. The SPP metric [11], an adaptation of the well-known ETX [10] unicast metric, was shown to outperform the other multicast metrics [11], [28]. Thus, in the remainder of the paper and in our experimental evaluation, we consider SPP for demonstrative purposes. Below, we first give an overview of ETX, then show how it was extended to SPP.

ETX Metric. The ETX metric [10] was proposed for unicast and estimates the expected number of transmissions needed to successfully deliver a unicast packet over a link, including retransmissions. Each node periodically broadcasts probe packets which include the number of probe packets received from each of its neighbors over a time interval. A pair of neighboring nodes, A and B, estimate the quality of the link $A \leftrightarrow B$ by using the formula $\text{ETX} = \frac{1}{d_f \times d_r}$, where d_f and d_r are the probabilities that a packet is sent successfully from A to B (forward direction) and from B to A (reverse direction), respectively. The value of ETX for a path of k links between a source S and a receiver R is $\text{ETX}_{S\to R} = \sum_{i=1}^{k} \text{ETX}_i$, where ETX_i is the ETX value of the i-th link on the path; $\text{ETX}_{S\to R}$ estimates the total number of transmissions by all nodes on the path to deliver a packet from a source to a receiver.

SPP Metric. ETX was adapted to the multicast setting by Roy *et al.* in the form of the SPP metric [11]. The value of SPP for a path of k links between a source S and a receiver R is $\text{SPP}_{S\to R} = \Pi_{i=1}^{k}\text{SPP}_i$, where the metric for each link i on the path is $\text{SPP}_i = d_f$ and d_f is defined as in ETX. The rationale for defining SPP as above is twofold:

- Unlike in unicast, where a successful transmission over a link depends on the quality of both directions of that link, in multicast only the quality of the forward direction matters because there are no link layer acknowledgments. The quality of a link $A \to B$, as perceived by node B, is $\text{SPP}_i = d_f$ and represents the probability that B receives a packet successfully from A over the link $A \to B$. Node B obtains d_f by counting the probes received from A over a fixed time interval.
- Also unlike unicast, in which the individual link metrics are summed, in multicast they are multiplied. This reflects the fact that for SPP the probability of a packet being delivered over a path from a source to a receiver is the product of the probabilities that the packet is successfully delivered to each of the intermediate nodes on the path. If any of the intermediate nodes fails to receive the packet, this causes the transmission for the entire route to fail, since there are no retransmissions. $\text{SPP}_{S\to R}$ (in fact $1/\text{SPP}_{S\to R}$) estimates the expected number of transmissions needed at the source to successfully deliver a packet from a source to a receiver.

SPP takes values in the interval $[0, 1]$, with higher metric values being better. In particular, $\text{SPP} = 1$ denotes perfect reliability, while SPP $= 0$ denotes complete unreliability.

2.2 High-Throughput Mesh-Based Multicast Routing

Multicast protocols provide communication from sources to receivers organized in groups by establishing dissemination structures such as trees or meshes, dynamically updated as nodes join or leave the group. Tree-based multicast protocols (*e.g.*, MAODV [7]) build optimized data paths, but require more complex operations to create and maintain the multicast tree, and are less resilient to failures. Mesh-based multicast protocols (*e.g.*, ODMRP [6]) build more resilient data paths, but have higher overhead due to redundant retransmissions.

We focus on ODMRP as a representative mesh-based multicast protocol for wireless networks. Below we first give an overview of ODMRP, then describe how it can be enhanced with any link-quality metric. The protocol extension to use a high-throughput metric was first described by Roy *et al.* [11], [28]. We refer to the ODMRP protocol using a high-throughput metric as ODMRP-HT in order to distinguish it from the original ODMRP [6] protocol.

ODMRP overview. ODMRP is an on-demand multicast routing protocol for multi-hop wireless networks, which uses a mesh of nodes for each multicast group. Nodes are added to the mesh through a route selection and activation protocol. The source periodically recreates the mesh by flooding a JOIN QUERY message in the network in order to refresh the membership information and update the routes. We use the term *round* to denote the interval between two consecutive mesh creation events. JOIN QUERY messages are flooded using a *basic flood suppression* mechanism, in which nodes only process the first received copy of a flooded message.

When a receiver node gets a JOIN QUERY message, it activates the path from itself to the source by constructing and broadcasting a JOIN REPLY message that contains entries for each multicast group it wants to join; each entry has a *next hop* field filled with the corresponding upstream node. When an intermediate node receives a JOIN REPLY message, it knows whether it is on the path to the source or not, by checking if the next hop field of any of the entries in the message matches its own identifier. If so, it makes itself a node part of the mesh (the FORWARDING GROUP) and creates and broadcasts a new JOIN REPLY built upon the matched entries.

Once the JOIN REPLY messages reach the source, the multicast receivers become connected to the source through a mesh of nodes (the FORWARDING GROUP) which ensures the delivery of multicast data. While a node is in the FORWARDING GROUP, it rebroadcasts any non-duplicate multicast data packets that it receives.

ODMRP takes a “soft state” approach in that nodes put a minimal effort to maintain the mesh. To leave the multicast group, receiver nodes are not required to explicitly send any message, instead they do not reply to JOIN QUERY messages. Also, a node’s participation in the FORWARDING GROUP expires if its forwarding-node status is not updated.

ODMRP-HT. We now describe ODMRP-HT, a protocol that enhances ODMRP with high-throughput metrics. The main differences between ODMRP-HT and ODMRP are: (1) instead of selecting routes based on minimum delay (which results in choosing the fastest routes), ODMRP-HT selects routes based on a link-quality metric, and (2) ODMRP-HT uses a *weighted flood suppression* mechanism to flood JOIN QUERY messages instead of using a basic flood suppression.

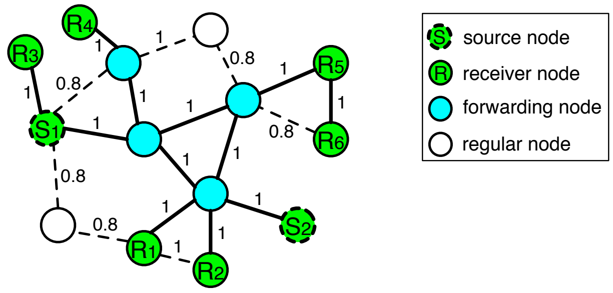


Fig. 1: An example of ODMRP-HT mesh creation for a multicast group with 2 sources (S_1, S_2) and 6 receivers ($R_1, \ldots, R_6$). The label on each link represents the value of the link’s SPP metric.

As required by the link-quality metric, each node measures the quality of the links from its neighbors to itself, based on the periodic probes sent by its neighbors. The JOIN QUERY message is flooded periodically by a source S and contains a *route cost* field which accumulates the metric for the route on which the message travelled. Upon receiving a JOIN QUERY, a node updates the route cost field by accumulating the metric of the last link travelled by the message. Because different paths may have different metrics, JOIN QUERY messages are flooded using a *weighted flood suppression* mechanism, in which a node processes flood duplicates for a fixed interval of time and rebroadcasts flood messages that advertise a better metric (indicated by the route cost field)[2]. Each node also records the node from which it received the JOIN QUERY with the best quality metric as its *upstream* node for the source S.

After waiting for a fixed interval of time, during which it may receive several JOIN QUERY packets that contain different route metrics, a multicast receiver records as its upstream for source S the neighbor that advertised the JOIN QUERY with the best metric. Just like in ODMRP, the receiver then constructs a JOIN REPLY packet, which will be forwarded towards the source on the optimal path as defined by the metric and will activate the nodes on this path as part of the FORWARDING GROUP. In Fig. 1 we give an example of how ODMRP-HT selects the mesh of nodes in the FORWARDING GROUP based on the SPP link-quality metric.

3 ATTACKS AGAINST HIGH-THROUGHPUT MULTICAST

In this section, we present attacks against high-throughput multicast protocols. In particular, we focus on attacks that exploit vulnerabilities introduced by the use of high-throughput metrics. These attacks require little resource from the attacker, but can cause severe damage to the performance of the multicast protocol. We first present the adversarial model, followed by the details of the attacks.

3.1 Adversarial Model and Goal

Malicious nodes may exhibit Byzantine behavior, either alone or in collusion with other malicious nodes. We refer to any arbitrary action by authenticated nodes deviating from protocol specification as Byzantine behavior, and to such an adversary as

[2] Several studies [26], [28] show that the overhead caused by rebroadcasting some of the flood packets is reasonable, validating the effectiveness of this weighted flood suppression strategy.

a Byzantine adversary. Examples of Byzantine behavior include: Dropping, injecting, modifying, replaying, or rushing packets, and creating wormholes. We consider attacks that aim to cause denial-of-service (DoS) on the multicast data delivery in which the attacker actively tries to attract and control data traffic in the path establishment process and then conduct packet dropping attacks to disrupt the packet delivery process. The specific types of attacks are discussed in the next section (Section III-B). We focus on attacks that seek to disrupt routing and do not consider other attacks such as traffic analysis and eavesdropping. We also do not consider selfish attacks such as nodes refusing to route packets for other nodes or falsely claiming to be in a poor quality location in order to repel traffic and preserve their resources; these could be mitigated with incentive mechanisms [29], [30]. We assume that adversaries do not have control on the physical or MAC layers, which can be protected with jamming-resilient techniques such as spread spectrum [31] and a more resilient MAC (*e.g.*, [32]). We also do not consider the Sybil attack, which can be addressed using techniques such as [33], [34], complementary to our protocol.

3.2 Attacks

In general, the attacker can achieve the goal of disrupting the multicast data delivery by either exhausting network resource (*resource consumption attacks*), by causing incorrect mesh establishment (*mesh structure attacks*), or by dropping packets (*data forwarding attacks*). The packet dropping attack is straightforward: The attacker node on the data delivery path simply drops data packets instead of forwarding them. We describe next resource consumption and mesh structure attacks.

3.2.1 Resource consumption attacks

ODMRP-HT floods JOIN QUERY messages in the entire network, allowing an attacker to inject either spoofed or its own legitimate JOIN QUERY messages at a high frequency to cause frequent network-wide flooding. The attacker can also activate many unnecessary data paths by sending many JOIN REPLY messages to cause unnecessary data packet forwarding. Finally, the attacker can inject invalid data packets to be forwarded in the network.

If the attackers are insider nodes, an effective attack is to establish a legitimate group session with high data rate in order to deprive the network resource from honest nodes. Addressing such an attack requires admission control mechanisms (*e.g.* [35]), which can limit the admission and duration of such groups. However, such mechanisms are out of the scope of this paper and are not considered further.

3.2.2 Mesh structure attacks

Mesh structure attacks disrupt the correct establishment of the mesh structure in order to disrupt the data delivery paths. These attacks can be mounted by malicious manipulation of the JOIN QUERY and JOIN REPLY messages.

For the JOIN QUERY messages, the attacker can spoof the source node and inject invalid JOIN QUERY messages, which can cause paths to be built toward the attacker node instead of the correct source node. The attackers may also act in a selfish manner by dropping JOIN QUERY messages, which allows them to avoid participation in the multicast protocol. Since JOIN QUERY messages are flooded in the network, unless the attacker nodes form a vertex cut in the network, they cannot prevent legitimate nodes from receiving JOIN QUERY messages. Finally, the attacker may also modify the accumulated path metrics in the JOIN QUERY messages incorrectly. Such *metric manipulation attacks* can pose a severe threat to the correctness of path establishment, and are discussed in more detail in the next section.

For the JOIN REPLY messages, the attacker can drop JOIN REPLY messages to cause its downstream nodes to be detached from the multicast mesh. The attacker can also forward JOIN REPLY to an incorrect next hop node to cause an incorrect path being built.

In many of the above attacks the power of the attacker relates directly to its ability to control the mesh structure and to be selected on paths. For example, if the attacker is on the path of many receivers, the attacker can affect many receivers by dropping the JOIN REPLY messages or the data packets. Traditionally, such ability is achieved via wireless-specific attacks such as rushing and wormholes. In a rushing attack, the attacker uses illegitimate means to forward packets faster than legitimate nodes (e.g., by ignoring the randomized small delays necessary in packet forwarding), whereas in a wormhole attack, attackers tunnel packets among themselves via in-band or out-of-band channels without being in physical proximity. The use of high-throughput metrics gives attackers additional opportunities to manipulate the mesh structure by manipulating the routing metric. Rushing and wormholes are general attacks against wireless routing protocols that have been studied extensively [36]–[39]. We focus below on metric manipulation attacks, which require only little effort to execute, yet are extremely detrimental to the protocol performance; such attacks have not been studied before.

3.3 Metric Manipulation Attacks

As discussed in Section 2, multicast protocols using high-throughput metrics prefer paths to the source that are perceived as having high-quality, while trying to avoid low-quality paths. Thus, a good strategy for an attacker to increase its chances of being selected in the FORWARDING GROUP is to advertise artificially good metrics for routes to the source.

The use of high-throughput metrics requires each node to collect *local* information about its adjacent links based on periodic probes from its neighbors. This local information is accumulated in JOIN QUERY packets and propagated in the network, allowing nodes to obtain *global* information about the quality of the routes from the source. Adversaries can execute two types of metric manipulation attacks: *local metric manipulation* (LMM) and *global metric manipulation* (GMM). These attacks are Byzantine in nature, as they are conducted by nodes that have the credentials to participate in the routing protocol, but are under adversarial control.

Local Metric Manipulation (LMM) Attacks. An adversarial node artificially increases the quality of its adjacent links, distorting the neighbors' perception about these links. The falsely advertised "high-quality" links will be preferred and malicious nodes have better chances to be included on routes.

A node can claim a false value for the quality of the links towards itself. In Fig. 2 a malicious node C_1 claims that $\text{SPP}_{B_1 \to C_1} = 0.9$ instead of the correct metric of 0.6. Thus, C_1 accumulates a false local metric for the link $B_1 \to C_1$ and advertises to R the metric $\text{SPP}_{S \to C_1} = 0.9$ instead of the correct

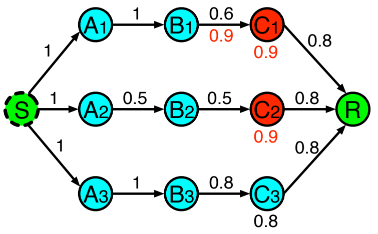


Fig. 2: Metric manipulation attack during the propagation of the flood packet from the source S to receiver R. A label above a link is the link's real SPP metric; a label below a link is the link's metric falsely claimed by a node executing a LMM attack; a label below a node is the accumulated route metric advertised by the node.

metric $\text{SPP}_{S\rightarrow C_1} = 0.6$. The route S-A_1-B_1-C_1-R will be chosen over the correct route S-A_3-B_3-C_3-R.

Global Metric Manipulation (GMM) Attacks. In a GMM attack, a malicious node arbitrarily changes the value of the route metric accumulated in the flood packet, before rebroadcasting this packet. A GMM attack allows a node to manipulate not only its own contribution to the path metric, but also the contributions of previous nodes that were accumulated in the path metric. For example, in Fig. 2 attacker C_2 should advertise a route metric of 0.25, but instead advertises a route metric of 0.9 to node R. This causes the route S-A_2-B_2-C_2-R to be selected over the correct route S-A_3-B_3-C_3-R.

3.4 Impact of Metric Manipulation Attacks on Routing

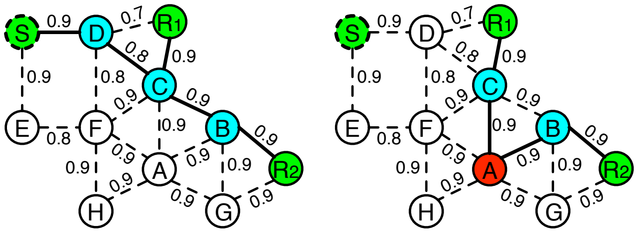
The danger of metric manipulation attacks comes from the epidemic attack propagation due to the epidemic nature of metric derivation. As a result, even a few number of attackers can "poison" the metrics of many nodes in the network and create powerful blackholes that attract and control the traffic to many receivers. We exemplify these effects with the scenario in Fig. 3.

When no attackers are present (Fig. 3(a)), nodes B, C and D are activated as part of the FORWARDING GROUP. Consider that node A executes a metric manipulation attack (Fig. 3(b)): Upon receiving the JOIN QUERY, node A changes the metric and advertises a perfect metric with value 1. Consequently, node C derives an incorrect metric of 0.9, and then propagates it to its neighbors, causing them to derive an incorrect metric as well. Both receivers R_1 and R_2 are also "attracted" to the attacker and only nodes B and C will be selected as part of the FORWARDING GROUP. The net effect is that both R_1 and R_2 are disconnected from the source.

Besides distorting path establishment for data delivery, a severe side effect of the attack is that it introduces a significant challenge for attack recovery. As the epidemic nature of metric poisoning causes the metric of many nodes to be incorrect, these metrics cannot be used for attack recovery even after the attacker is identified. Instead, one has to either resort to a fallback procedure not using the metric or refresh the metric of all the nodes in the network in a trustworthy manner before recovery.

4 SECURE HIGH-THROUGHPUT MULTICAST ROUTING

In this section, we present our secure multicast routing protocol, S-ODMRP, with a novel defense scheme RateGuard to accommodate high-throughput metrics.



(a) benign case

(b) attack case

Fig. 3: Metric manipulation attack in a network with one source (S), two receivers (R_1, R_2) and one attacker (A). The label on each link represents the value of the link's SPP metric.

4.1 Authentication Framework

We assume that each user authorized to be part of the mesh network has a pair of public and private keys and a *client certificate* that binds its public key to a unique user identifier. This defends against external attacks from nodes that are not part of the network. We assume source data is authenticated, so that receivers can distinguish authentic data from spurious data. Efficient source data authentication can be achieved with existing schemes such as TESLA [40]. Finally, we assume the existence of a secure neighbor discovery scheme [41].

4.2 S-ODMRP Overview

S-ODMRP ensures the delivery of data from the source to the multicast receivers even in the presence of Byzantine attackers, as long as the receivers are reachable through non-adversarial paths. To achieve this, S-ODMRP uses a combination of authentication and rate limiting techniques against resource consumption attacks and a novel technique, RateGuard, against the more challenging packet dropping and mesh structure attacks, including metric manipulations and JOIN REPLY dropping.

S-ODMRP uses source message authentication to avoid processing non-authenticated messages. This eliminates a large class of attacks, including outsider attacks, message spoofing and modification attacks targeting JOIN QUERY and JOIN REPLY messages, and the injection of corrupted data packets.

Even with message authentication, an insider attacker can still mount the resource consumption attack by flooding JOIN QUERY messages frequently with itself as the source. Such an attack can be countered by rate limiting, for example, a honest node only forwards JOIN QUERY messages for a source node up to a maximum frequency.

To address the resource consumption attack in which the attacker activates many unnecessary data delivery paths by injecting many JOIN REPLY messages, we can limit to at most one the number of JOIN REPLY messages a node may send in one round. Each node monitors the number of different signed JOIN REPLY messages that originate from its neighbors. If a node is observed to have broadcast two or more different signed JOIN REPLY messages, then punitive actions can be taken against the node (*e.g.*, isolation).

The attacks on the mesh structure and packet dropping attacks are much more challenging to defend against, particularly in the context of high-throughput metrics. In the following, we focus on defending against these attacks. We will first present the high level overview of our defense scheme, RateGuard, and then present the details of S-ODMRP with the RateGuard scheme.

4.3 RateGuard Overview

RateGuard relies on the observation that regardless of the attack strategy, either by dropping Join Reply, metric manipulations, or by dropping packets, attackers do not affect the multicast protocol unless they cause a drop in the packet delivery ratio (PDR). We adopt a reactive approach in which attacker nodes are detected through a *measurement-based detection* protocol component, and then isolated through an *accusation-based reaction* protocol component. Next we describe these two components.

Measurement-based attack detection. Whether by packet dropping alone or by combining it with metric manipulation to attract routes, the effect of an attack is that data is not delivered at a rate consistent with the advertised path quality. We propose a generic attack detection strategy that relies on the ability of honest nodes to detect the discrepancy between the *expected* PDR (ePDR) and the *perceived* PDR (pPDR). A node can estimate the ePDR of a route from the value of the metric for that route[3]; the node can determine the pPDR for a route by measuring the rate at which it receives data packets from its upstream on that route[4].

Both Forwarding Group nodes and receiver nodes monitor the pPDR of their upstream node. If ePDR − pPDR for a route becomes larger than a detection threshold δ, then nodes suspect that the route is under attack because the route failed to deliver data at a rate consistent with its claimed quality[5].

Accusation-based attack reaction. We use a *controlled-accusation* mechanism in which a node, on detecting malicious behavior, temporarily accuses the suspected node by flooding in the network an Accusation message containing its own identity (the accuser node) and the identity of the accused node, as well as the duration of the accusation. As long as the accusation is valid, metrics advertised by an accused node will be ignored and the node will not be selected as part of the Forwarding Group. This strategy also successfully handles attacks against path establishment. From the downstream node point of view, the dropping of a Join Reply message causes exactly the same effect as the attacker dropping all data packets, thus the downstream nodes will react and accuse the attacker.

To prevent the abuse of the accusation mechanism by attackers, a node is not allowed to issue a new accusation before its previously issued accusation expires. Accused nodes can still act as receivers even though they are excluded from the Forwarding Group. We use a temporary accusation strategy to cope with transient network variations: The accusation duration is calculated proportional to the observed discrepancy between ePDR and pPDR, so that accusations caused by metric inflation and malicious data dropping last longer, while accusations caused by transient network variations last shorter.

Finally, to address the metric poisoning effect caused by metric manipulation attacks, the metric in the entire network is refreshed shortly after attack detection. In S-ODMRP, the metric refreshment is achieved automatically through the periodic Join Query messages.

Sign(m): sign message m using this node's private key
Verify(n_id, sig): verify the signature sig using node n_id's public key and exit the procedure if the verification fails
Start_timer(timer, t): start timer timer with timeout t
Refresh_timer(timer, t): if timer is not active, then call Start_timer(timer, t); otherwise, set timeout of timer to t
Broadcast(m): broadcast message m one hop
Flood(m): flood message m in the entire network
Send_message(m, n_id): reliably send message m to neighbor n_id
Link_metric(n_id): return the measured link metric to neighbor n_id
Get_best_metric(query_set): return the best metric of all queries in the set query_set, regardless of accusation status
Get_neighbor_best_metric(query_set): return the neighbor that has the best metric in the set query_set, regardless of accusation status

Fig. 4: Basic procedures used in the S-ODMRP protocol description

4.4 S-ODMRP Detailed Description

To describe S-ODMRP in detail, we use the list of procedures described in Fig. 4. For simplicity, we limit the description to one multicast group and one multicast source. However, the scheme can easily be extended to multicast groups with multiple sources.

4.4.1 Mesh Creation

S-ODMRP mesh creation follows the same pattern of ODMRP-HT presented in Sec. 2.2. As described in Fig. 5, the source node S periodically broadcasts to the entire network a Join Query message in order to refresh the membership information and to update the routes (lines 1-5). The Join Query message is signed by S and is propagated using a weighted flood suppression mechanism. Nodes only process Join Query messages that have valid signatures (line 8) and that are received from nodes not currently accused (indicated by an Accusation List maintained by each node) (lines 18-19). Nodes record the upstream node and the metric corresponding to the route with the best metric as best_upstream and best_metric (line 23).

The Join Reply messages are then sent from receivers back to S along optimal paths as defined by the high-throughput metric, leading to the creation of the Forwarding Group (the multicast mesh) (lines 28-33). After sending a Join Reply to its best_upstream, a node starts to monitor the PDR from its best_upstream in order to measure its perceived PDR (pPDR) (line 38).

To address attackers that strategically accuse certain nodes in order to disconnect the network, we make one exception from the rule that only non-accused nodes are included in the Forwarding Group: If the best metric is advertised by an accused neighbor, a node also activates this neighbor (by sending a Join Reply) in addition to the best non-accused neighbor (line 39-41). This ensures that good paths are still utilized, even if honest nodes on these paths are falsely accused. The additional transmissions are kept to a minimum because the neighbors with the best and second best metric usually share the same upstream node. Thus, as shown in Sec. 6.5, this strategy only adds a very low overhead.

4.4.2 Attack Detection

As described in the protocol overview, we detect attacks using a measurement-based mechanism, where each Forwarding Group and receiver node continuously monitors the discrepancy between ePDR and pPDR and flags an attack if ePDR − pPDR $> \delta$.

The most straightforward method for estimating pPDR is to use a sliding window method, with pPDR calculated as pPDR $= r/w$,

[3] For the SPP metric, a route's ePDR is equal to the route's metric.

[4] Source data authentication allows nodes to distinguish authentic packets from spurious ones and only authentic packets are counted towards pPDR.

[5] Note that the rate inconsistency may also be caused by natural link quality variations. We do not differentiate between losses caused by adversarial behavior and natural link variations because lossy links must also be avoided in order to maintain a good performance level.

```
Executed at the source node to initiate a new JOIN QUERY message:
create a JOIN QUERY message q
q.source = source_id; q.from = source_id
q.path_metric = 1; q.seq = join_seq
join_seq + +
Sign(q); Broadcast(q)

Executed at a node upon receipt of a JOIN QUERY message q:
if (latest_received_join_seq > q.seq) then
    return // ignore old queries
Verify(q.from, q.sig)
get_new_query = FALSE
if (latest_received_join_seq < q.seq) then
    // get a new (non-duplicate) query
    latest_received_join_seq = q.seq
    best_metric = 0
    best_upstream = INVALID_NODE
    fastest_upstream = q.from // for fallback recovery
    get_new_query = TRUE
received_queries.insert(q) // store the query
if (accusation_list.contains_accused_node(q.from)) then
    q.path_metric = 0
else
    q.path_metric = q.path_metric × Link_metric(q.from)
if (get_new_query OR q.path_metric > best_metric) then
    best_upstream = q.from; best_metric = q.path_metric;
    q.from = node_id
    Sign(q); Broadcast(q)
if (get_new_query AND is_receiver) then
    Start_timer(reply_timer, REPLY_TIMEOUT)

Executed at a node upon timeout of reply_timer:
Send_reply()

Executed at a node upon receipt of a JOIN REPLY message r:
if (latest_received_reply_seq < r.seq) then
    latest_received_reply_seq = r.seq
    Refresh_timer(FG_timer, FG_TIMEOUT)
    if (not is_receiver) then
        Send_reply()

Send_reply()
create a JOIN REPLY message r
r.seq = latest_received_join_seq
Send_message(r, best_upstream)
if (best_metric > 0) then
    start monitoring the PDR of best_upstream
if (Get_best_metric(received_queries) > best_metric) then
    // Activate the accused neighbor with best metric
    Send_message(r, Get_neighbor_best_metric(received_queries))
received_queries.clear() // purge stored queries
```

Fig. 5: Mesh creation algorithm

where r is the number of packets received in the window and w is the number of packets sent by the source (derived from packet sequence numbers) in the window. Albeit being simple, this method is sensitive to bursty packet loss. In addition, this approach requires a node to wait until at least w packets are sent in a round before being able to make any decision. Therefore, setting w too large causes delay in making decisions, whereas setting w too small results in inaccurate pPDR estimation and hence more frequent false positives. In general, it is difficult to determine the optimal value for w, as it depends on the network conditions and the specific position of a node. To avoid these shortcomings, we propose an efficient statistical-based estimation method for pPDR that naturally adapts to the network environment of each node.

The main idea is to use the Wilson estimate [42] to determine a confidence interval for pPDR, instead of trying to obtain a single estimated value. Let m be the number of packets received by a node and n be the number of packets sent by the source in the same time period, which can be derived from packet sequence numbers. The Wilson estimate requires that $n \geq 5$ [42], so whenever $n \geq 5$, we can obtain a confidence interval for pPDR as $(\hat{p} - e, \hat{p} + e)$, where

$$\hat{p} = \frac{m+2}{n+4} \text{ and } e = z\sqrt{\frac{\hat{p}(1-\hat{p})}{n+4}}.$$

We assign $z = 1.96$ to obtain the commonly used confidence level of 95%. An attack is detected if the upper bound of the confidence interval for pPDR is less than the estimated PDR even after accounting for normal network variations, *i.e.*, if:

$$\hat{p} + e < \text{ePDR} - \delta,$$

where δ is the estimated PDR discrepancy under normal network conditions. In this method, the exact number of packets required for attack detection naturally adapts to the path quality and the severity of the attack. In addition, there is a precise level of confidence on the accuracy of our estimation. This method has the advantage that it applies for both constant rate and variable rate data sources.

Addressing silent periods. If a node fails to receive any data packets in a round, the above method will be not able to compute the confidence interval of pPDR for its upstream node, since the value of n is derived from sequence numbers contained in received packets. We address this issue by including the current data sequence number in JOIN QUERY packets, which are periodically flooded in the network. Thus, unless a node does not have any adversarial-free path to the source, it can always obtain the current data sequence number and compute the pPDR confidence interval to detect attacks.

4.4.3 Attack Reaction

To isolate attackers, our protocol uses a controlled-accusation mechanism which consists of three components, staggered reaction timeout, accusation message propagation and handling, and recovery message propagation and handling. As described in Fig. 6, when a node detects attack behavior, it starts a React_Timer with timeout value $\beta(1 - \text{ePDR})$, where β is a system parameter that determines the maximum timeout for reaction timer (line 1). Since ePDR decreases monotonically along a multicast data path, nodes farther away from the source will have a larger timeout value for the React_Timer. This staggered timeout technique ensures nodes immediately below the attacker will take action first, before any of their downstream nodes mistakenly accuse their upstream node. When the React_Timer of a node N expires, N accuses its best_upstream node and cancels the React_Timer at its downstream nodes with the following actions:

- Create, sign, and flood an ACCUSATION message in the network, which contains N's identity (the accuser node) and the identity of N's best_upstream node (the accused node). The message also contains a value accusation_time $= \alpha(\text{ePDR} - \text{pPDR})$, indicating the amount of time the accusation lasts (lines 8-13). α is a tunable system parameter that determines the severity of attack punishment.
- Create, sign, and send to its downstream nodes a RECOVERY message, which contains the ACCUSATION message (lines 15-19). This message serves the role of canceling the React_Timer of nodes in N's subtree and activating the fallback procedure at the receivers in N's subtree (see Sec. 4.4.4).

Upon receipt of an ACCUSATION message, a node checks if it does not have an unexpired accusation from the same accuser node and verifies the signature on the message. This enforces our

```
On detecting a discrepancy between ePDR and pPDR:
 1: Start_timer(React_Timer, $\beta(1 - \text{ePDR})$)
Executed at node on timeout of React_Timer:
 2: if (is_receiver) then
 3:     create salvage message ss // fallback
 4:     Send_message(ss, fastest_upstream)
 5: if (accusation_list.contains_accuser_node(node_id)) then
 6:     return // each node can only accuse once
 7: // create and flood accusation message
 8: create accusation message acc
 9: acc.accused = best_upstream
10: acc.accuser = node_id
11: acc.accusation_time = $\alpha(\text{ePDR} - \text{pPDR})$
12: accusation_list.add(acc)
13: Sign(acc); Broadcast(acc)
14: // send recovery message to the subtree
15: create recovery message rr
16: rr.accusation = acc
17: Sign(rr)
18: for each downstream node d do
19:     Send_message(rr, d)
Executed at a node on receipt of an accusation message acc:
20: if (accusation_list.contains_accuser_node(acc.accuser)) then
21:     return // only allow one accusation from a node at a time
22: Verify(acc.accuser, acc.sig)
23: accusation_list.add(acc)
24: Broadcast(acc)
Executed at a node on receipt of a recovery message rr:
25: if (handled_recovery_messages.contains(rr)) then
26:     return // ignore duplicate recovery
27: if (accusation_list.contains_accuser_node(rr.acc.accuser)
    OR rr.acc.accusation_time < $\alpha(\text{ePDR} - \text{pPDR})$) then
28:     return // ignore recovery mesage if the accuser has an unexpired
        accusation or if the accused time is inconsistent
29: Verify(rr)
30: handled_recovery_messages.insert(rr)
31: if (React_Timer is active) then
32:     cancel React_Timer
33: for each downstream node d do
34:     Send_message(rr, d)
35: if (is_receiver) then
36:     create salvage message ss // fallback
37:     Send_message(ss, fastest_upstream)
Executed at a node on receipt of a salvage message ss:
38: Refresh_timer(FG_timer, FG_TIMEOUT)
39: Send_message(ss, fastest_upstream)
```

Fig. 6: Attack reaction algorithm

limited accusation mechanism, which allows nodes to only have one active accusation at a time. If both checks pass, the node adds a corresponding entry to its ACCUSATION LIST (lines 20-23). Accusations are removed from the ACCUSATION LIST after the accusation_time has elapsed.

Upon receipt of a RECOVERY message rr from its best_upstream node, a FORWARDING GROUP node N checks if it does not have an unexpired accusation from the same accuser node and verifies the signature on the message. In addition, the node also checks that the accusation_time in the message is at least as much as its own observed discrepancy (the ePDR − pPDR value) (lines 27-28). This prevents attackers who cause a large PDR drop from bypassing our defense by accusing its upstream node only for a short amount of time. If all checks pass, it cancels its pending React_Timer, forwards rr to its downstream nodes (lines 31-34), and if it is a receiver, activates the recovery procedure (lines 35-37) (see below).

Avoiding redundant accusations. Typically, an attacker node will attract many honest neighbor nodes to connect to it. In order to avoid all the neighboring nodes accusing the same attacker node and losing their accusing ability, we require each of the neighboring nodes to add a random jitter before flooding its accusation message. If a neighboring node receives an accusation message accusing the same node it is about to accuse, it aborts its pending accusation. If the node is required to send a RECOVERY message to its downstream, it includes the received accusation message in RECOVERY instead of its own accusation message.

4.4.4 Fallback Recovery

The accusation mechanism ensures that when the metric is refreshed in the round after the attack detection, the accused nodes are isolated. However, during the round when an attack is detected, the receiver nodes in the subtree of the attacker need to find alternative routes to "salvage" data for the rest of the round. As shown in Sec. 3.3, a side effect of metric manipulation attacks is *metric poisoning*, which prevents recovery by relying on the metrics in the current round. We address this inability by falling back to the fastest route for routing during the remainder of the round[6]. Specifically, during the JOIN QUERY flooding, besides recording the best_upstream node, each node also records the upstream for the fastest route as fastest_upstream (Fig. 5, line 15). To recover from an attack, a receiver sends a special JOIN REPLY message (a salvage message) to its fastest_upstream node (Fig. 6, lines 2-4 and 35-37). Each node on the fastest route forwards the special JOIN REPLY message to their fastest_upstream node and becomes part of the FORWARDING GROUP (Fig. 6, lines 38-39).

4.5 Impact of False Positives

Even though our defense scheme takes into account normal network variations with the parameter δ, it is still possible that some honest nodes are mistakenly accused. We argue that such false positive accusations have little impact on the performance of the system for two main reasons. First, under most cases honest nodes cause only a small discrepancy on the PDR, thus even if mistakenly accused, their accusation duration is relatively short. Second, for most receivers there are redundant paths to the source. Thus, even if some honest nodes are wrongly accused, affected receiver nodes can obtain a similar performance by using nearby alternate routes.

4.6 Practical Implementation Issues

4.6.1 Parameter Selection

S-ODMRP has three tunable parameters, the attack detection threshold δ, the coefficient for accusation duration α, and the coefficient for React_Timer β.

The selection of δ trades off tolerance to normal network variations with sensitivity of the scheme to attacks. A larger value for δ reduces false positives of accusing honest nodes, however, it also allows attackers to inflict more impact without being detected. An optimal value for δ is the estimated normal network variation, *i.e.* the sum of the PDR discrepancy under normal network conditions and the error in estimating ePDR from the advertised metric.

The value for α trades off the effectiveness of the scheme in isolating attacker nodes and the severity of isolating honest nodes due to false positives. In a stable network where the number of

[6] The strategy is not attack-proof, as the fastest route may include malicious nodes. However, since the route is only used for the remainder of the round, we prefer to use an efficient procedure than to find attacker-free paths, which is itself a challenging task and requires expensive protocols [24]. We further discuss in Sec. 5.1 and 5.2 the impact of attacks against the recovery phase.

false positives is small, or in a dense network where the impact of false positives is small due to path redundancy, it is advisable to have a large value for α in order to reduce the impact of attacks. In Sec. 5.1, we give lower bounds for α in order for the scheme to bound the impact of attacks effectively.

The value for β trades off the attack reaction delay and the effectiveness of the staggered reaction timeout technique for preventing honest nodes from mistakenly accusing each other. A smaller β results in quicker attack reaction, however, it also results in a smaller difference in the reaction timeout value for consecutive nodes on a path, increasing the chance of honest nodes being mistakenly accused. In our experiments, we find $\beta = 20ms$ achieves a good balance between these two effects.

4.6.2 Ensuring Staggered Timeouts for Reaction Timers

To avoid honest nodes mistakenly accusing each other, it is critical that we ensure the React_Timer at a downstream node does not expire before the node receives a recovery message from its upstream node. Denote the link latency as t, and the estimated PDR of two consecutive nodes on a path as ePDR_1 and ePDR_2. The React_Timer timeout value for the two nodes is $TO_1 = \beta(1 - \text{ePDR}_1)$ and $TO_2 = \beta(1 - \text{ePDR}_2)$. We need to ensure $TO_2 - TO_1 > t$, that is,

$$\beta(1 - \text{ePDR}_2) - \beta(1 - \text{ePDR}_1) > t,$$

hence $\text{ePDR}_2 < \text{ePDR}_1 - t/\beta$. Therefore, to ensure staggered reaction timeout, we require a node artificially decreases its advertised metric if necessary so that its ePDR is at least t/β smaller than the ePDR of its upstream node.

5 S-ODMRP Security Analysis

In this section, we analyze the security of the S-ODMRP protocol and establish bounds on the attack impact and on protocol resilience to various types of attacks.

5.1 Attack Impact

We upper bound the attack impact on the throughput of S-ODMRP. We first give a precise definition of the attack impact. We then present two theorems that upper bound the attack impact and discuss their practical implications.

Let N denote the network of interest with k attacker nodes. We define N' as the exact same network as N, except that all the attacker nodes are removed. For a given non-attacker receiver node R, let $r_R(t)$ and $r'_R(t)$ be the perceived PDR of R at time t in network N and N', respectively. We define I_R, the *attack impact* on a node R, as

$$I_R = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (r'_R(t) - r_R(t))dt,$$

where t_0 and t_1 are the start and end times for the interval of interest. Intuitively, the attack impact is the average PDR degradation caused by the presence of attackers over time compared to a network with no attackers. Alternatively, the attack impact captures the discrepancy between a given defense scheme and a hypothetical perfect defense scheme where all the attackers nodes are perfectly isolated.

Recall that δ denotes the attack detection threshold and α denotes the accusation duration coefficient. Also, recall that a *round* is an interval between two consecutive mesh creation events. We use λ to denote the time duration of a round. In the following, we show two theorems that bound the attack impact of metric manipulation attackers (colluding or individual) on any non-attacker receiver node in the presence of our defense mechanisms.

Theorem 1. *In a network with k metric manipulation attackers, for any $\alpha \geq \frac{k\lambda}{\delta^2}$, the attack impact on any non-attacker receiver node in* S-ODMRP *is upper bounded by δ during any time interval of duration $T \gg \alpha$.*

Implications of Theorem 1. According to Theorem 1 for large enough α, the impact of metric manipulation attacks is bounded by the attack detection threshold δ. For example, with $\delta = 20\%$, round duration of $\lambda = 3$ seconds, and a total of 10 attackers, according to Theorem 1, we can set $\alpha \geq 750$ seconds to ensure the attack impact on any non-attacker receiver node is bounded by δ. Theorem 1 assumes the attacker nodes can coordinate perfectly and completely disrupt the fallback procedure, thus it gives an upper bound on the impact of the attack.

Theorem 2. *In a network with k metric manipulation attackers, if* S-ODMRP *uses a fallback procedure that restores the PDR to the same level as in a benign network, then for any $\alpha \geq \frac{k\lambda}{\delta}$, the attack impact on any non-attacker receiver node is upper bounded by δ during any time interval of duration $T \gg \alpha$.*

Implications of Theorem 2. In Theorem 2, we see that if we assume the ideal case where the fallback procedure is always able to restore the data rate to the normal level, then we can bound the attack impact under δ with a much smaller value for the accusation duration α. For example, with the same settings of δ and λ as above, we *only* need to set $\alpha \geq 150$ seconds.

The fallback recovery procedure uses fastest paths to recover data after attack detection. Assuming resilience against rushing attacks, the attackers cannot attract the recovery paths toward incorrect directions. Furthermore, the broadcast nature of wireless transmission and the mesh nature of ODMRP can also tolerate occasional attackers on the data forwarding paths. Hence, in a relatively dense network where there are many possible paths from receivers to the source, most receivers are able to restore the data rate to a level similar with the fallback recovery phase (this is also confirmed by our experiments in Sec. 6). Therefore, compared to Theorem 1, Theorem 2 is a closer approximation to reality: We only need to set α to be slightly larger (*e.g.*, 250 seconds in our experiments) than the value derived from Theorem 2 to bound the attack impact under the estimated normal network variations.

Proofs. To prove these two theorems, we first introduce some additional notations and two lemmas. We label the start of the time duration of interest of T as t_0. Without loss of generality, let time t_i and t_{i+1} be the start and end times of round i, for $i \geq 0$. Since a node only estimates its metric at the beginning of a round, let $m_B(i)$ and $m'_B(i)$ be the estimated PDR from metric for node B in round i in network N and in N', respectively. We use $\bar{r}_B(i)$ and $\Delta_B(i)$ to denote the average perceived PDR and average PDR discrepancy of node B in round i in network N, respectively, that is $\bar{r}_B(i) = \frac{1}{\lambda}\int_{t_i}^{t_{i+1}} r_B(t)dt$ and $\Delta_B(i) = m_B(i) - \bar{r}_B(i)$. Similarly, we define $\bar{r}'_B(i)$ and $\Delta'_B(i)$ for network N'. With a slight abuse of notation, we denote the attack impact on node B in round i as $I_B(i)$, that is, $I_B(i) = I_B(t_i, t_{i+1})$. It is easy to see that $I_B(i) = \bar{r}'_B(i) - \bar{r}_B(i)$.

For simplicity, we assume that the network is stable and the PDR estimation from the metric is accurate. Hence, for benign network N', we have $m'_B(i)$ and $\bar{r}'_B(i)$ are constant and $m'_B(i) = \bar{r}'_B(i)$ for all i. Thus, without ambiguity, we use $\bar{r}'_B$ to denote both the estimated and perceived PDR at node B in network N'. Thus, we have $I_B(i) = \bar{r}'_B - \bar{r}_B(i)$.

We also discount any physical layer effects (*e.g.*, interference), which means that $m_U(i) \geq r'_U$ for any node U, since additional attacker nodes cannot decrease the metric derived by honest nodes.

Lemma 1. *For any round i and any non-attacker node B, we have $I_B(i) \leq \Delta_B(i)$.*

Proof: Under the attack scenarios considered in Section 3.2, in order for an attack to have any impact on R, it must be the case that $m_B(i) \geq r'_B$, since otherwise attacker nodes will not be selected on the path. Therefore,

$$I_B(i) = r'_B - \bar{r}_B(i) \leq m_B(i) - \bar{r}_B(i) = \Delta_B(i).$$

□

Intuitively, Lemma 1 says that the attack impact of any node is always upper bounded by its observed PDR discrepancy.

Lemma 2. *For any consecutive sequence of time intervals $(t_0, t_1), (t_1, t_2), \cdots, (t_{k-1}, t_k)$ and a non-attacker node B, if $I_B(t_i, t_{i+1}) \leq d$ for all $0 \leq i \leq k-1$, then $I_B(t_0, t_k) \leq d$.*

Proof: This is immediate from the definition of attack impact. □

Proof of Theorem 1: For ease of exposition, we first analyze the single attacker case, followed by the multiple attackers case.

For the single attacker case, let A be the attacker node in network N and let R be a non-attacker receiver node that is downstream from A. Let node B be the immediate downstream of node A on the path from A to R. Let p_{BR} denote the path PDR between B and R. Since there is no attacker between B and R, we have $r_R = r_B \cdot p_{BR}$ and $r'_R = r'_B \cdot p_{BR}$. Thus,

$$\begin{aligned} I_R &= \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (r'_R(t) - r_R(t))dt \\ &= \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (r'_B(t) \cdot p_{BR} - r_B(t) \cdot p_{BR})dt \\ &= \frac{p_{BR}}{t_1 - t_0} \int_{t_0}^{t_1} (r'_B(t) - r_B(t))dt \\ &= I_B \cdot p_{BR} \leq I_B \end{aligned}$$

Therefore, we only need to show $I_B \leq \delta$.

We classify rounds into two categories, Category A for rounds in which the attacker is not detected, and Category B for rounds in which the attacker is detected or isolated. An attack in a Category A round i implies, by definition, that the attacker is not detected, *i.e.*, it drops data below the δ threshold: $\Delta_B(i) < \delta$. By Lemma 1, we have $I_B(i) \leq \Delta_B(i) \leq \delta$.

Let round a be the round in which the attack is detected and let w be the discrepancy observed at node B when the attack is detected. Then our protocol ensures that $w \geq \delta$ and that node B accuses and isolates node A for time $\alpha w \geq \alpha\delta$. If we denote the time when the attacker recovers from the accusation as t_r, then $t_r - t_a \geq \alpha w$. Therefore, the attack impact on node B from time t_a to time t_r is:

$$I_B(t_a, t_r) = \frac{\int_{t_a}^{t_{a+1}} (r'_B - r_B(t))dt}{t_r - t_a} \leq \frac{\lambda}{\alpha w} \leq \frac{\lambda}{\alpha\delta}$$

Hence, if $\alpha \geq \frac{\lambda}{\delta^2}$, we have $I_B(t_a, t_r) \leq \delta$. Therefore, by Lemma 2, we have $I_B(t_0, t_r) \leq \delta$. Since the maximum accusation time is α, for any time interval with duration $T \gg \alpha$, we have $I_R \leq I_B \leq \delta$.

For the case of multiple attackers, for rounds in Category A, where no attackers are detected, we also have $I_B(i) \leq \delta$. For rounds in Category B, node B may switch to another attacker node in the round after detecting an attacker node, hence for a total of k attacker nodes, we have

$$I_R(t_a, t_r) \leq I_B(t_a, t_r) \leq \frac{k\lambda}{t_r - t_a} \leq \frac{k\lambda}{\alpha\delta}$$

Hence, if $\alpha \geq \frac{k\lambda}{\delta^2}$, we have $I_R(t_a, t_r) \leq \delta$. □

Proof of Theorem 2: In the proof of Theorem 1, we assume a node has perfect path quality in the benign network, whereas the node has zero PDR in the round when the attacker is detected. If the fallback procedure can restore the PDR to the level of the benign network, then the attack impact during that round is bounded by δ (because, once the discrepancy on the average PDR exceeds δ, the attack is detected and the node invokes the fallback procedure). Therefore, we can derive an upper bound for I_R for rounds in Category B as follows:

$$I_R(t_a, t_r) \leq \frac{k\delta\lambda}{t_r - t_a} \leq \frac{k\delta\lambda}{\alpha\delta} = \frac{k\lambda}{\alpha}$$

Hence, if $\alpha \geq \frac{k\lambda}{\delta}$, then $I_R(t_a, t_r) \leq \delta$. Following a similar analysis as in Theorem 1, we obtain that $I_R \leq \delta$ for any time interval of duration $T \gg \alpha$. □

5.2 RateGuard Attack Resiliency

We discuss the resilience of the RateGuard protocol to various types of attacks. In particular, the attacker may inject, modify, or drop accusation and recovery messages. Since both accusation and recovery messages are signed by the sender, the modification attack is prevented. We consider the other attacks as follows.

Accusation message dropping. Since accusation messages are flooded in the network, unless the attacker nodes form a vertex-cut in the network, they cannot cause accusation messages to be missed by other nodes.

Accusation message injection. This is the false accusation attack. Our limited accusation mechanism restricts attackers to only have one active false accusation at any time. In addition, our technique of activating the neighbor advertising the best metric regardless of its accused status ensures that falsely accused nodes are also used in routing. This prevents attacker nodes from partitioning the network by strategically accusing certain honest nodes. Therefore, false accusation attacks only cause falsely accused nodes to be ignored in the metric propagation process. In a dense enough network, this only results in limited impact on the metric derived at each node, as each node typically has multiple disjoint paths with similar metrics to the source. Therefore, the overall impact of the false accusation attack is limited.

Recovery message injection. Since a node ignores recovery message unless there is a corresponding accusation message, the one active accusation only policy also prevents the attacker from causing its downstream nodes to constantly resort to the fall-back procedures.

Recovery message dropping. Dropping a recovery message will only cause the attacker node itself to be accused by its downstream honest node, because the downstream node does not cancel its reaction timer unless it receives a recovery message.

Attacks on the fall-back procedure. Since we do not protect the fallback recovery phase, attackers that are selected as forwarders during the recovery phase may drop packets without being punished. However, since the fallback recovery is only used to salvage data for the remaining of the current round, the impact of the attack is limited. As shown in Theorem 1, even if the attacker is able to completely block all packets to a node during the fallback recovery procedure, the average attack impact is still bounded by δ for sufficiently large values of α.

5.3 Limitations of S-ODMRP

In S-ODMRP, a node is detected as an attacker only if the PDR drop caused by the node exceeds the threshold δ. Therefore, this leaves the room for an attacker to drop some amount of data below the threshold δ without being detected. Since the threshold δ models the normal PDR variation exhibited by legitimate nodes, it is impossible to distinguish such attackers from normal nodes. One may address this shortcoming with more accurate modeling of the behavior of normal nodes. For example, we can incorporate both the mean and the variance of PDR. Since we expect PDR to vary around its mean under normal network variations, a node whose PDR is constantly below its advertised value, even only for a small amount, can be seen as abnormal. We defer such enhancements as future work.

S-ODMRP restricts a node to accuse at most one other node at a time. This implies that attacker nodes should be a minority in the network. Otherwise, some attacker nodes will be left unaccused and will be free to attract and deny service to many receivers through metric manipulation. It is extremely difficult to secure a network where the majority of nodes are insider attackers, and we do not address this scenario in this work.

6 Experimental Evaluation

In this section, we demonstrate through experiments the vulnerability of metric enhanced multicast protocol by examining the impact of different attacks, and investigate the effectiveness of our defense mechanisms and its associated overhead.

6.1 Experimental Methodology

Simulation Setup. We implemented ODMRP-HT and S-ODMRP using the ODMRP version available in the Glomosim [43] simulator. Nodes use 802.11 radios with 2 Mbps bandwidth and 250m nominal range. We simulate environments representative of mesh network deployments by using the two-ray radio propagation model with the Rayleigh loss model, which models environments with large reflectors, *e.g.*, trees and buildings, where the receiver is not in the line-of-sight of the sender.

The network consists of 100 nodes randomly placed in a 1500m×1500m area. We randomly select 20 nodes as multicast group members and one randomly selected node among them as the data source. Attackers are randomly selected among nodes that are not group members. Group members join the group in the beginning of the simulation. At second 100, the source starts multicasting 512-byte data packets for 400 seconds at a rate of 20 packets/second. For S-ODMRP, we use RSA signatures with 1024-bit keys, simulating delays to approximate the performance of a 1.3 GHz Intel Centrino processor. We empirically tune the threshold $\delta = 20\%$ to accommodate random network variations in the simulated scenarios. The timeout for React_Timer is set to $20(1 - \text{ePDR})$ millisecond (i.e. $\beta = 20$) and the accusation_time is set to $250(\text{ePDR} - \text{pPDR})$ second (i.e. $\alpha = 250$). Nodes use the statistical-based method described in Sec. 4.4.2 to determine their pPDR.

We used the SPP high-throughput metric, configured with optimal parameters as recommended in [11]. Data points are averaged over 10 different random environments and over all group members.

Attack Scenarios. We consider the following scenarios:

• *No-Attack*: The attackers do not perform any action in the network. This represents the ideal case where the attackers are identified and completely isolated in the network, and serves as the baseline for evaluating the impact of the attack and the performance of our defense.

• *Drop-Only*: The attackers drop data packets, but participate in the protocol correctly otherwise. The attack has effect only when attackers are selected in the Forwarding Group. We use this scenario to demonstrate that metric manipulation amplifies data dropping attacks.

• *LMM-Drop*: The attackers combine local metric manipulation (LMM) with the data dropping attack. The attackers conduct the LMM attack by re-advertising the same metric they received in Join Query, which is equivalent to making their link metric of the previous hop equal to 1 (best).

• *GMM-Drop*: The attackers combine global metric manipulation (GMM) with the data dropping attack. The attackers conduct the GMM attack by re-advertising a metric of 1 (best) after receiving a Join Query.

• *False-Accusation*: The attackers exploit our accusation mechanism by falsely accusing random a honest node at startup for the whole experiment period in order to reduce the PDR. Due to space constraint, we do not present results for attacks that aim to cause large bandwidth overhead through frequent flooding of accusation messages using false accusations. We can upper bound the frequency of the accusation message flooding from any attacker node to only once a few seconds by imposing a lower bound on the accusation timeout, thus the inflation of overhead is limited.

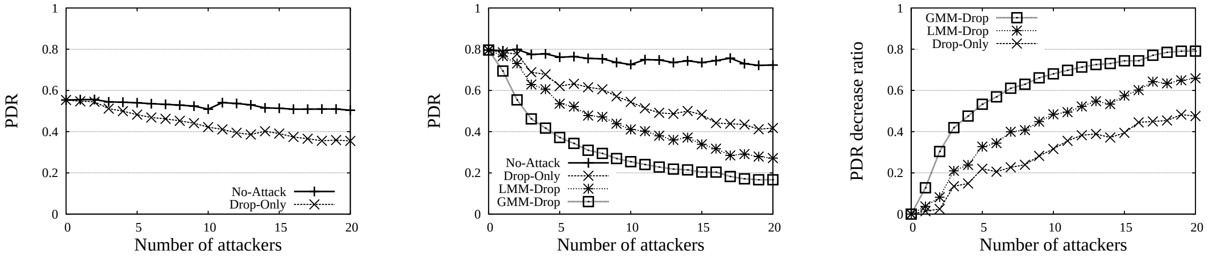
Metrics. We measure the performance of data delivery using the packet delivery ratio (PDR), defined as $\text{PDR} = n_r/n_s$, where n_r is the average number of packets received by all receivers and n_s is the number of packets sent by the source.

We also measure the strength of the attacks using as metric the PDR decrease ratio (PDR-DR), defined as

$$\text{PDR-DR} = \frac{\text{PDR}_{\text{noattack}} - \text{PDR}_{\text{attack}}}{\text{PDR}_{\text{noattack}}},$$

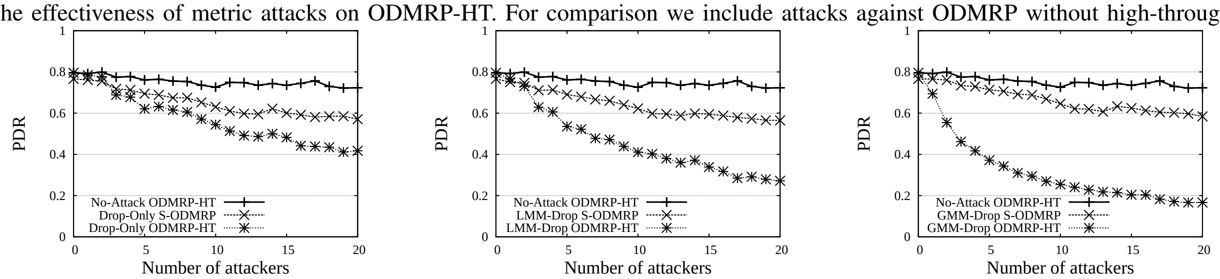
where $\text{PDR}_{\text{attack}}$ and $\text{PDR}_{\text{noattack}}$ represent the PDR when the network is under attack and not under attack, respectively.

The overhead of our defense consists of three components, the control bandwidth overhead due to additional messages and larger message size (*e.g.*, accusation messages, signatures on query messages), the computational overhead due to cryptographic operations, and the additional data packet transmissions caused



(a) Attacks on ODMRP (b) Attacks on ODMRP-HT (c) Attack strength against ODMRP-HT

Fig. 7: The effectiveness of metric attacks on ODMRP-HT. For comparison we include attacks against ODMRP without high-throughput metrics.



(a) *Drop-Only* attack (b) *LMM-Drop* attack (c) *GMM-Drop* attack

Fig. 8: The effectiveness of S-ODMRP for different attacks.

by our protocol. We measure the control bandwidth overhead per node, defined as the total control overhead divided by the number of nodes. The computational overhead is measured as the number of signatures performed by each node per second. To measure redundant data packet transmissions, we define *data packet transmission efficiency* as the total number of data packets transmitted by all nodes in the network divided by the total number of data packets received by all receivers. Thus, data packet transmission efficiency captures the cost (number of data packet transmissions) per data packet received.

6.2 Effectiveness of Metric Manipulation Attacks

Fig. 7(a) shows the impact of *Drop-Only* attack on the original ODMRP (not using high-throughput metric). The protocol is quite resilient to attacks, *i.e.*, PDR decreases by only 15% for 20 attackers. This reflects the inherent resiliency of mesh based multicast protocols against packet dropping, as typically a node has multiple paths to receive the same packet.

Fig. 7(b) shows the PDR of the protocol when using a high-throughput metric (ODMRP-HT) under different types of attacks. We observe that with the *Drop-Only* attack, the PDR drops quickly to a level below the case when no high throughput metric is used. Thus, simple packet dropping completely nullifies the benefits of high throughput metrics. By manipulating the metrics as in *LMM-Drop* and *GMM-Drop*, the attacker can inflict a much larger decrease in PDR. For example, the PDR decreases from 72% to only 25% for 10 attackers using *GMM-Drop*, in contrast to 55% for *Drop-Only*. Fig.7(c) compares the impact of the attack in terms of the PDR decrease ratio (PDR-DR). We see that metric manipulation significantly increases the attack strength. For example, with 10 attackers, the PDR-DR of *GMM-Drop* (68%) is more than double the PDR-DR of *Drop-Only* (32%). Thus, we conclude that metric manipulation attacks pose a severe threat to high-throughput protocols.

6.3 Effectiveness of the Defense

In Fig. 8 we show the effectiveness of our defense (S-ODMRP) against different types of attacks, compared to the insecure ODMRP-HT protocol. S-ODMRP suffers only a small PDR decrease relative to the baseline *No-Attack* case. For example, a total of 20 attackers causes a PDR drop of only 12%, considerably smaller than the case without defense, which shows a PDR decrease by as much as 55% in the *GMM-Drop* attack. To rule out random factors, we performed a paired t-test [42] on the results showing that S-ODMRP improves the PDR for all attack types, with P-value less than 2.2×10^{-16}. For 10 attackers, S-ODMRP improves the PDR of ODMRP-HT for *Drop-Only*, *LMM-Drop* and *GMM-Drop* by at least 4.5%, 16.7%, 33%, with 95% confidence level. Thus, our defense is very effective against all the attacks. The small PDR decrease for S-ODMRP can be attributed to two main factors. First, common to all reactive schemes, attackers can cause some initial damage, before action is taken against them. Second, as the number of attackers increases, some receivers become completely isolated and are not able to receive data.

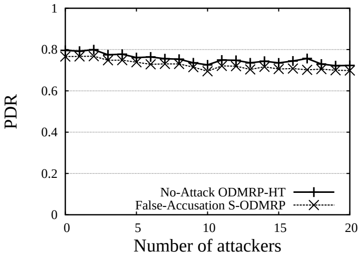
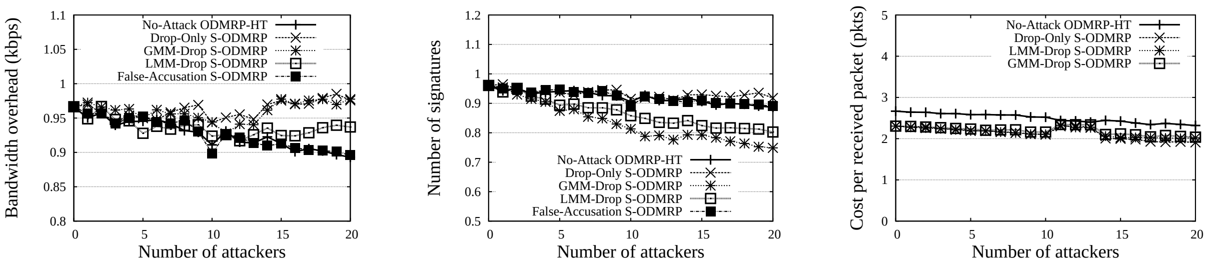


Fig. 9: Impact of the *False-Accusation* attack on S-ODMRP.

Fig. 8 also shows an interesting phenomenon: The PDR decrease for S-ODMRP is similar for all attacks, despite the varying strength of the attacks. This outcome reflects the design of our defense mechanism in which accusations last proportional to the discrepancy between ePDR and pPDR: Attacks that cause a small discrepancy (*e.g.*, *Drop-Only*) are forgiven sooner and can be executed again, while attacks that cause a large discrepancy (*e.g.*, *GMM-Drop*) result in a more severe punishment and can be executed less frequently. Finally, we note that the attack impact on S-ODMRP is less than $\delta = 20\%$, which is consistent with the bound in our analysis in Sec. 5.1.

6.4 Defense Resiliency to Attacks

Attackers may attempt to exploit the accusation mechanism in S-ODMRP. Fig. 9 shows that S-ODMRP is very resilient against



(a) Bandwidth overhead per node

(b) Computational overhead per node

(c) Data transmission efficiency

Fig. 10: The overhead of S-ODMRP.

the *False-Accusation* attack, in which attackers falsely accuse one of their neighbors. This comes from the controlled nature of accusations, which allows an attacker to accuse only one honest node at a time. Also, as described in Sec. 4.4, falsely accused nodes that advertise a good metric may continue to forward data.

6.5 Overhead of S-ODMRP

Fig. 10(a) and 10(b) show the control bandwidth and computational overhead for S-ODMRP. We observe that for all attack configurations, the bandwidth and computational overhead are maintained at a stable low level of around 0.95 kbps and 0.9 signatures per node per second. To understand the source of the overhead better, we analyzed different components of the overhead. The result shows that the overhead due to reacting to attackers (such as dissemination of ACCUSATION and RECOVERY messages) is negligible, since the attackers, once detected, are accused for a relatively long period of time The bulk of the overhead comes from the periodic network-wide flooding of authenticated JOIN QUERY packets. Since query flooding is common in all scenarios, we obtain a similar level of overhead across different scenarios. The reason for the slight overhead decrease for an increasing number of attackers for the *False-Accusation* attack is that JOIN QUERY from the falsely accused honest nodes are ignored by their neighbors, resulting a smaller number of transmissions of JOIN QUERY packets.

In Fig. 10(c), we notice that S-ODMRP under various attacks even improves slightly the *data transmission efficiency* of ODMRP-HT with no attacks. This apparent anomaly can be explained because in S-ODMRP nodes further away from the source are more likely to be affected by attacks and these are the nodes that require more transmissions to receive data packets.

7 RELATED WORK

There has been extensive work in the area of secure unicast routing in multi-hop wireless networks. Examples include [18]–[24], [44], [45]. In general, attacks on routing protocols can target either the route establishment process or the data delivery process, or both. Ariadne [22] and SRP [17] propose to secure on demand source routing protocols by using hop-by-hop authentication techniques to prevent malicious packet manipulations on the route discovery process. SAODV [44], SEAD [18], and ARAN [19] propose to secure on demand distance vector routing protocols by using one-way hash chains to secure the propagation of hop counts. [45] proposes a secure link state routing protocol that ensures the correctness of link state updates with digital signatures and one-way hash chains. To ensure correct data delivery, [20] proposes the watchdog and pathrater techniques to detect adversarial nodes by having each node monitor if its neighbors forward packets correctly. SMT [21] and Ariadne [22] use multi-path routing to prevent malicious nodes from selectively dropping data. ODSBR [23], [24] provides resilience to colluding Byzantine attacks by detecting malicious links based on an end-to-end acknowledgment-based feedback technique.

In contrast to secure unicast routing, the work studying security problems specific to multicast routing in wireless networks is particularly scarce, with the notable exception of the work by Roy *et al.* [25] and BSMR [26]. [25] proposes an authentication framework that prevents outsider attacks in a tree-based multicast protocol, MAODV [7], while BSMR [26] complements the work in [25] and presents a measurement-based technique that addresses insider attacks in tree-based multicast protocols.

A key point to note is that all of the above existing work in either secure unicast or multicast routing considers routing protocols that use only basic routing metrics, such as hop count and latency. None of them consider routing protocols that incorporate high-throughput metrics, which have been shown to be critical for achieving high performance in wireless networks. On the contrary, many of them even have to remove important performance optimizations in existing protocols in order to prevent security attacks.

There are also a few studies ([46], [47]) on secure QoS routing in wireless networks. However, they require strong assumptions, such as symmetric links, correct trust evaluation on nodes, ability to correctly determine link metrics despite of attacks. In addition, none of them consider attacks on the data delivery phase. To the best of our knowledge, our work is the first work that encompasses both high performance and security as goals in multicast routing and considers attacks on both path establishment and data delivery phases.

Besides attacks on the routing layer, wireless networks are also subject to wireless specific attacks, such as flood rushing and wormhole attacks. Defenses against these attacks have been extensively studied in previous work, *e.g.*, [36]–[39], and are complementary to our protocol. RAP [36] 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* [37] which restricts the maximum transmission distance by using time or location information, Truelink [38] which uses MAC level acknowledgments to infer if a link exists or not between two nodes, and the work in [39], which relies on directional antennas.

8 CONCLUSION

We considered the security implication of using high throughput metrics in multicast protocols in wireless mesh networks. In particular, we identified metric manipulation attacks that can inflict significant damage on the network. The attacks not only have a direct impact on the multicast service, but also raise additional challenges in defending against them due to their metric poisoning effect. We overcome the challenges with our novel defense scheme, RateGuard, that combines measurement-based attack detection and accusation-based reaction. Our defense also copes with transient network variations and malicious attempts to attack the network indirectly by exploiting the defense itself. We demonstrate through analysis and experiments that our defense is effective against the identified attacks, resilient to malicious exploitations, and imposes a small overhead.

References

[1] J. Dong, R. Curtmola, and C. Nita-Rotaru, "On the pitfalls of using high-throughput multicast metrics in adversarial wireless mesh networks," in *Proc. of IEEE SECON '08*, 2008.

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

[3] R. Chandra, V. Ramasubramanian, and K. Birman, "Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks," in *ICDCS '01*.

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

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

[6] S. J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mob. Netw. Appl.*, 2002.

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

[8] J. G. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks." in *MobiHoc*, 2001.

[9] H. Lundgren, E. Nordstrom, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *WOWMOM '02*.

[10] D. S. J. D. Couto, D. Aguayo, J. C. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *MOBICOM '03*.

[11] S. Roy, D. Koutsonikolas, S. Das, and C. Hu, "High-throughput multicast routing metrics in wireless mesh networks," in *Proc. of ICDCS*, 2006.

[12] A. Chen, D. Lee, G. Chandrasekaran, and P. Sinha, "HIMAC: High throughput MAC layer multicasting in wireless networks," in *MASS '06*.

[13] B. Awerbuch, D. Holmer, and H. Rubens, "The medium time metric: High throughput route selection in multirate ad hoc wireless networks," *MONET, Spec. Iss. on Internet Wireless Access: 802.11 and Beyond*, 2005.

[14] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," in *BroadNets '04*.

[15] S. Keshav, "A control-theoretic approach to flow control," *Proc. of the Conference on Communications Architecture and Protocols*, 1993.

[16] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. of MOBICOM '04*. ACM, 2004.

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

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

[19] 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*, 2002.

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

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

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

[23] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure byzantine resilient routing protocol for wireless ad hoc networks," *ACM Transactions on Information Systems Security (TISSEC)*, vol. 10, no. 4, 2007.

[24] 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.

[25] S. Roy, V. G. Addada, S. Setia, and S. Jajodia, "Securing MAODV: Attacks and countermeasures," in *Proc. of SECON '05*. IEEE, 2005.

[26] R. Curtmola and C. Nita-Rotaru, "BSMR: Byzantine-resilient secure multicast routing in multi-hop wireless networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 8, no. 4, pp. 445–459, 2009.

[27] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *Proc. of SIGCOMM '04*, 2004.

[28] S. Roy, D. Koutsonikolas, S. Das, and C. Hu, "High-throughput multicast routing metrics in wireless mesh networks," *Elsevier Ad Hoc Netw*, 2007.

[29] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretic and cryptographic techniques," *Wirel. Netw.*, vol. 13, no. 6, pp. 799–816, 2007.

[30] L. Buttyan and J.-P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," *Mob. Netw. Appl.*, vol. 8, no. 5, 2003.

[31] R. Pickholtz, D. Schilling, and L. Milstein, "Theory of spread-spectrum communications–a tutorial," *Communications, IEEE Transactions on*, vol. 30, no. 5, pp. 855–884, May 1982.

[32] N. Abramson, "The aloha system - another alternative for computer communications," in *Proc. of Fall Joint Comp. Conf., AFIPS Conf.*, 1970.

[33] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," in *Proc. of IPSN '04*. ACM Press, 2004.

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

[35] Y. Yang and R. Kravets, "Contention-aware admission control for ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 4, no. 4, 2005.

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

[37] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *Proc. of INFOCOM '03*.

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

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

[40] A. Perrig, R. Canetti, D. Song, and D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. of NDSS*, February 2001.

[41] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure neighbor discovery in wireless networks: formal investigation of possibility," in *ASIACCS '08*.

[42] D. S. Moore and G. P. McCabe, *Introduction to the Practice of Statistics*. New York: W.H.Freeman, 2003.

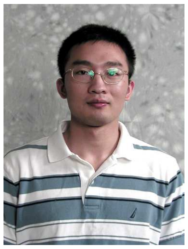
[43] "Global mobile information systems simulation library - glomosim." [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[44] M. Guerrero Zapata and N. Asokan, "Securing Ad hoc Routing Protocols," in *Proc. of ACM WiSe '02*, 2002.

[45] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in *Proc. of IEEE SAINT '03 Workshops*, 2003.

[46] P. P. Papadimitratos and Z. J. Haas, "Secure Route Discovery for QoS-Aware Routing in Ad Hoc Networks," in *IEEE Sarnoff Symp.*, 2006.

[47] T. Zhu and M. Yu, "A dynamic secure QoS routing protocol for wireless ad hoc networks," in *Proc. of IEEE Sarnoff Symposium '06*, 2006.



Jing Dong received his PhD degree in Computer Science from Purdue University in 2009. During his PhD study, he was a member of the Dependable Distributed Systems Laboratory. He received his BS and MS degree in Computer Science in 2003 and 2004, both from University of Massachusetts, Boston. His research interests are in wireless networks with a focus on resilience and security of such networks. He is a member of the ACM.

Reza Curtmola is an Assistant Professor in the Department of Computer Science at NJIT. He received the BS degree in Computer Science from the "Politehnica" University of Bucharest, Romania, in 2001, the MS degree in Security Informatics in 2003, and the PhD degree in Computer Science in 2007, both from The Johns Hopkins University. He spent one year as a post-doctoral research associate at Purdue University. His research focuses on applied cryptography and security aspects of wireless networks. He is a member of the ACM and the IEEE Computer Society.

Cristina Nita-Rotaru is an Associate Professor in the Computer Science Department of Purdue University. She leads the Dependable and Secure Distributed Systems Laboratory. She received the BS and MS degrees in Computer Science from "Politehnica" University of Bucharest, Romania, in 1995 and 1996, and the MSE and PhD degrees in Computer Science from The Johns Hopkins University in 2000 and 2003. She served on the technical program committee of numerous conferences in information security, distributed systems, fault-tolerance, and networking. She received the National Science Foundation CAREER award in 2006. Her research interests are in security aspects of distributed systems and network protocols. She is a member of the ACM and IEEE Computer Society.