

Defense against Routing Disruption Attacks in Mobile Ad Hoc Networks

Ruiliang Chen*, Michael Snow*, Jung-Min Park*, M. Tamer Refaei, and Mohamed Eltoweissy

*Laboratory for Advanced Research in Information Assurance and Security (ARIAS)

Bradley Department of Electrical and Computer Engineering

Virginia Polytechnic Institute and State University

{rlchen, sno, jungmin, mtamer, toweissy}@vt.edu

Abstract—We propose a secure routing architecture for Mobile Ad hoc NETWORKS (MANETs) called Throughput-Feedback (TUF) routing, which is resilient against a wide range of routing disruption Denial-of-Service (DoS) attacks. Unlike many existing solutions, TUF does not focus on a particular type of attack, but instead takes an approach that is fundamentally more general. TUF is a cross-layer technique that detects attacks at the transport layer but responds to attacks at the network layer. Because most routing disruption attacks cause a significant drop in end-to-end goodput, monitoring the goodput of a route at the transport layer can detect abnormalities in the network (e.g., node or link failures, DoS attacks, etc.). Once an abnormal event is detected, a route rebuilding process is initiated at the network layer to find a new route. Using analysis and simulation results, we show that the TUF architecture is effective in thwarting a wide range of attacks, including protocol-compliant (also known as “JellyFish”) attacks.

Index Terms—Denial-of-Service Attacks, Mobile Ad Hoc Networks, Routing Disruption Attacks.

I. INTRODUCTION

The dependability and security aspects of a Mobile Ad hoc NETWORK (MANET), including reliability and availability, are of great importance for mission-critical and other information-sensitive applications. As a major threat to MANET security, quite a few Denial of Service (DoS) attacks have been discovered and discussed in the literature. According to their goals, DoS attacks can be broadly classified into two classes: *routing disruption* attacks and *resource consumption* attacks [8]. A routing disruption attack attempts to cause legitimate data packets to be routed in a dysfunctional way, whereas a resource consumption attack injects packets into the network to consume valuable network resources. In this paper we focus on routing disruption attacks.

We divide routing disruption DoS attacks based on their different levels of sophistication into three categories: *outsider* attacks, *insider* attacks, and *protocol-compliant* attacks. In an outsider attack, the attackers are assumed to have no knowledge of the keys that are used to encrypt and authenticate the data and routing control packets. Preventing outside attackers from tampering with the data is accomplished simply by encryption and authentication schemes [8, 16].

In an insider attack, the attacker has compromised or captured a node, thus gaining access to encryption and authentication keys. The primary method of detecting and mitigating insider attacks is to monitor the packet forwarding behavior among the nodes [2, 4, 14, 20]. Also, there are approaches that focus on thwarting specific forms of insider attacks [9, 10].

A new class of DoS attacks, called protocol-compliant DoS attacks, was introduced recently [1]. This class of attacks is the most difficult to defend against. In [1], Aad et al. refer to such attacks as “JellyFish” (JF) attacks. While the two types of attacks discussed above disobey protocol rules, JF attacks conform to all routing and

forwarding rules. They are also passive, and therefore difficult to detect. A typical target of JF attacks is closed-loop flows that respond to packet delay and loss, such as TCP. Protecting MANETs against JF attacks is a formidable task that has yet to be addressed.

We propose a routing architecture for MANETs, called *Throughput-Feedback routing (TUF)*, which is resilient to a wide range of attacks, including protocol-compliant attacks. TUF is a *cross-layer* approach that monitors the end-to-end “good” throughput (or “goodput”) at the transport layer to detect abnormalities and reacts to those abnormalities at the network layer. The justification for this is based on the observation that most known forms of route disruption attacks, including protocol-complaint attacks, degrade transport layer goodput. The TUF architecture is compatible with on-demand *source routing* protocols such as Dynamic Source Routing (DSR) [11]. TUF is composed of two modules: *Throughput Monitoring (TM)* and *Route Rebuilding (RR)*. TM is responsible for detecting any abnormalities that might occur on a route. If any abnormalities are detected, TM invokes RR, which employs the *least-alike re-routing (LARR)* algorithm to build a new route. Our simulation results show that TUF can effectively mitigate protocol-compliant attacks. Moreover, we show that TUF is capable of circumventing a variety of insider attacks such as blackhole, grayhole, rushing, and wormhole attacks.

The remainder of this paper is organized as follows. Section II discusses research related to MANET routing security and provides technical background of TUF. Section III introduces TUF and its modules. The security aspects of TUF are explored in Section IV. The simulation results are shown in Section V. Finally, we discuss conclusions and future work in Section VI.

II. RELATED RESEARCH

A. Routing Disruption Attacks on MANET and Countermeasures

Spoofing and *replay* are typical outsider attacks. They can be countered by encryption and packet authentication [8, 16]. The Secure Routing Protocol (SRP) [16] designs a security extension header that is attached to control packets for route discovery. SRP can be used for the DSR. Ariadne [8] provides efficient authentication for the DSR using a variant of TESLA source authentication technique.

Various insider attacks have been discussed in the literature, including *blackhole*, *grayhole* [8], *rushing* [10], *wormhole*¹ [9], *blackmail*, and *selfish* attacks. The research community has made a great effort to combat insider attacks [2, 4, 9, 10, 14, 20]. Awerbuch et al. [2] introduce a technique for detecting faulty links on a path from the source to the destination using a binary search. Hu et al.

¹ It can be argued that a malicious node in a rushing or wormhole attack could act like a repeater without requiring knowledge of any keys. However, to really do harm, the attacker needs to distinguish control packets from data packets in either of the attacks. This is possible only when the attacker has knowledge of the encryption key. Therefore, we classify both attacks as insider attacks.

propose the Rushing Attack Prevention (RAP) scheme [10] as a generic defense against the rushing attack for protecting on-demand routing protocols. The same authors propose “packet leashes” [9] to thwart wormhole attacks during a route search process. A reputation-based system is another approach that thwarts attacks by monitoring the traffic in a network [4, 14, 20]. Marti et al. [14] adopt two modules—“watchdog” and “pathrater”—for this purpose. Watchdog is a module that detects neighbor node misbehavior in promiscuous mode of the wireless interface; and pathrater defines the route quality as the average reputation of the nodes on a route and chooses the route with the best quality. Several schemes use similar ideas [4, 20].

Protocol-compliant DoS attacks, a.k.a. JF attacks [1], is by far the most difficult to defend against. In a JF attack, the malicious node can reorder packets, periodically drop packets, or increase packet jitter. Although such behavior can be considered a network-layer attack it will affect the transport-layer goodput by exploiting the vulnerabilities of the congestion control mechanism. It was shown in [1] that the JF attack can result in near zero goodput in the transport layer while keeping network-layer throughput fairly stable. Currently, there is no known countermeasure for the JF attack.²

B. Overview of the Dynamic Source Routing Protocol (DSR)

TUF is an architecture for secure routing in MANETs that can be most readily integrated into on-demand source routing protocols. To demonstrate TUF’s functionalities, we describe it in the context of the DSR [11], which is the first on-demand routing protocol. DSR uses the source routing option in data packets to carry the routing information. Each node, using a route cache, stores one or more complete lists of node addresses that form a path towards a destination. DSR is composed of two phases: route discovery and route maintenance.

Route discovery: When a node has packets to send, it first checks its route cache. If a route entry corresponding to the destination is not present in its route cache, a ROUTE REQUEST packet is broadcast over the network. The ROUTE REQUEST packet is uniquely identified by the source address, the destination address, and a sequence number that is incremented by the source node for each route discover request. An intermediate node appends its own address to the node list in the ROUTE REQUEST packet and forwards it. When the ROUTE REQUEST packet reaches the destination node, it has accumulated the path from the source to the destination. Assume that the underlying MAC layer supports bidirectional links, the destination node can get a valid route back to source node simply by reversing the source route recorded in the ROUTE REQUEST packet. Then the destination node sends back to the source node a ROUTE REPLY packet along the same route in the opposite direction. The ROUTE REPLY packet contains the information needed for the source node to route its packets to the destination node. It is possible for a node to receive the same ROUTE REQUEST packet multiple times because it is broadcast over the network. DSR requires an intermediate node to respond only to the first ROUTE REQUEST packet received and ignore the other duplicates, which is known as “duplicate suppression”. Note that duplicate suppression makes DSR vulnerable to rushing and wormhole attacks—if a malicious node manages to disseminate ROUTE REQUEST packets quickly so that they reach the other nodes before the legitimate packets, then the malicious node on a route with the least delay will always be included in the selected route.

Route maintenance: Every node along a route is responsible for the validity of the downstream link connecting itself and its next hop node. If link breakage is found, the source node will be notified with a ROUTE ERROR packet. The source node then initiates another route discovery procedure. Note that this procedure has a vulnerability: since sending a ROUTE ERROR packet is voluntary, malicious nodes can break links without being detected by the source node.

III. THE THROUGHPUT-FEEDBACK ROUTING ARCHITECTURE

A. Assumptions and Overview

We assume all links in the network to be bidirectional. We only consider network-layer route disruption attacks and disregard attacks to the physical or link layer of a wireless network. In a communication session, we assume that both the source node and the destination node are trustworthy but intermediate nodes are not. It is assumed that all control packets used in TUF are authenticated using some security mechanism (e.g., [8, 16]). This means that TUF is inherently resistant against outsider attacks. We focus our discussions on insider attacks and protocol-compliant attacks. A route with one or more malicious nodes is considered an *infected* route.

In this paper, we focus on TCP as it is the most widely-used transport layer protocol and it is the major target of JF attacks. TUF uses a TM module to estimate the TCP goodput. If the actual observed goodput deviates from the estimation beyond a certain threshold, an alarm is raised to activate the RR module. The RR module uses a routing algorithm called LARR to find a new route. The justification for such an approach is based on the following observations: (1) most known forms of DoS attacks lower transport layer goodput and (2) it is possible to estimate a typical TCP goodput value for a given ad hoc network, as was shown in [15]. Therefore, a significant gap between the estimated goodput and the observed goodput can act as a telltale sign indicating an abnormal event. It is true, however, that not only attacks but non-attack events (e.g., routing failures due to wireless link contention or node mobility) may also cause a temporary drop in goodput, thus incurring a “false positive”. Fortunately, TUF will build a new route to maintain an adequate level of goodput for the current flow, irrelevant of whether the deterioration of the goodput was caused by an attack or a non-attack event. In essence, TUF provides a general blanket of protection against a wide range of route failures by responding to attack events and non-attack events in the same manner; this approach significantly simplifies the overall design of a secure routing protocol and avoids the use of “customized” security solutions whose effectiveness is limited to only certain attack types.

B. Throughput Monitoring (TM) and TUF Flow

TM monitors a route’s status by making periodic observations on its goodput, which can be done by observing the 32-bit *acknowledgement number* field in the TCP header. The period is denoted as t (sec). We also denote the estimated threshold of acceptable throughput as G_{th} (bytes/sec). If the observed goodput averaged over t seconds is less than G_{th} , then the TM module determines that the current route is problematic and raises an alarm.

We use the round trip time (RTT) between the source and destination node to determine the value of t . It is known that the size of the TCP congestion window changes over time, which results in multiple crests and troughs in the curve of the TCP throughput over multiple RTT time spans [13]. We take a simplistic macroscopic view of TCP throughput and compute the average throughput value as the average of a given crest and trough value. This means that t needs to cover at least a full time period from a

² *L’Hospital* [12] uses a community-based secure routing to thwart JF attack. However, this scheme assumes an intrusion detection system that is capable of detecting JF attacks.

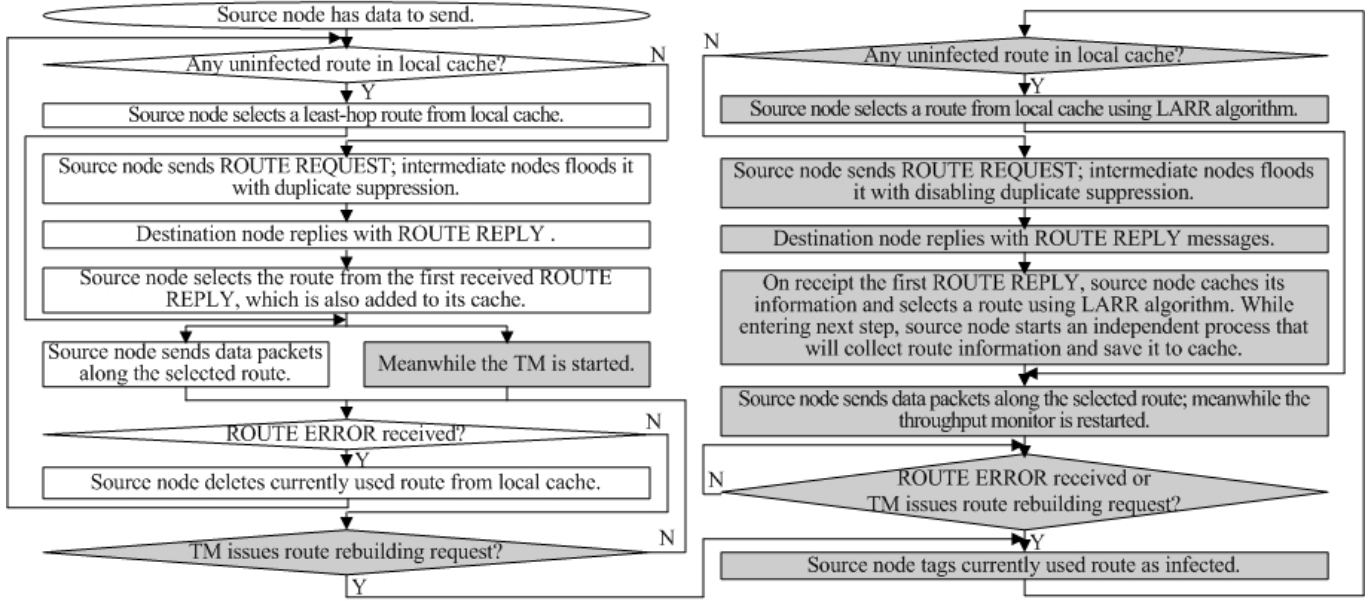


Figure 1. Flowchart of TUF routing.

crest to the next crest. In [7], the authors showed that the average measured congestion window is approximately 20 MSS (maximum segment size) in a typical size MANET (such as a 169-node grid); we use this value in our calculations. Using the TCP Reno [6] congestion control algorithm, we can calculate that the time it takes the congestion window to change from a crest value c (MSS) to a trough value t (MSS) is approximately 14 RTT. This was calculated from $c = 2t$, $(c + t) / 2 = 20$, and using the fact that the congestion window is increased by 1 MSS for every RTT in the congestion avoidance phase. Here, the slow start phase is ignored since it is relatively short when the congestion window increases exponentially. We propose to set t as the time interval between a crest to the next crest (i.e., one period), which is approximately 15 RTT. Note that one RTT is added to 14 RTT (the time interval from a trough to a crest) because we are measuring the time interval between two consecutive crests.

For G_{th} , we use the TCP performance model proposed in [17], which is valid for ad hoc networks [15]. In this model, the average TCP throughput T (bytes/sec) can be expressed as:

$$T = \frac{1}{RTT \sqrt{\frac{2B}{3}} p + T_0 \min\left\{1, 3\sqrt{\frac{3B}{8}} p\right\} p(1 + 32p^2)}, \quad (1)$$

where $T_0 = 2RTT$ is the timeout, B denotes the TCP maximum segment size (bytes), and p denotes the packet loss ratio.

Since RTT and B are known to the source node and p can be measured or collected with ROUTE REQUEST messages [19], a source node can estimate the average throughput of a route using (1). Then, one can set the value of G_{th} as

$$G_{th} = T \cdot r_a, \quad (2)$$

where $r_a \in [0, 1]$ is a coefficient that is introduced to reduce the number of false positives. The relationship between r_a and the number of false positives is investigated in Section V using simulations.

We now need to integrate TM to the routing process. Fig. 1 shows a flowchart that represents the flow of operations performed by TUF when it is applied to DSR. The functionalities in common with DSR are shown as non-shaded blocks and TUF-specific functionalities are shown as shaded blocks. In the flowchart, TUF enhances DSR in two aspects. First, when there are abnormalities in a route, TM triggers a proactive route rebuilding rather than only responding to

ROUTE ERROR passively. Second, TUF allows the source node to collect more routing information by disabling duplicate suppression. This information is used by the source node to build a new route via the LARR routing algorithm.

C. Route Rebuilding (RR)

The primary function of the RR module, the other major TUF component, is to build a new route after the TM module has invoked a route rebuild request. The *least-alike re-routing (LARR)* algorithm is proposed for this purpose.

Assume that the source node's set of cached (uninfected) routes are represented by $S_N = \{N_j : 1 \leq j \leq U\}$, where $U = |S_N|$ ($|X|$ denotes the number of elements in set X), and the routes that are tagged as infected are represented by $S_R = \{R_i : 1 \leq i \leq M\}$, where $M = |S_R|$. Here, N_j or R_i denotes a set of nodes contained in a given route. Let us define the "aliqueness degree" of N_j with respect to S_R to be $E(j) = \max_i (|N_j \cap R_i|)$. Then, LARR can be expressed as follows: the sender selects a new route by selecting a route in the cache with the smallest alikeness degree. That is, it selects a new route $N_{j'}$ such that $j' = \arg \min_j (E(j))$. If there are multiple

routes that satisfy the aforementioned condition, then the one with the least number of nodes is selected among them. If there are still more than one routes to choose from, then the one with the smallest index is chosen. It can be shown that for a k -hop route, if n nodes are inserted by an attacker and can be deployed anywhere, then the complexity of route rebuilding is $O(n^{k-1})$ using LARR. (Please refer to our technical report [5] for the proof.)

IV. SECURITY ANALYSIS OF TUF

TUF is designed to counter a wide range of attacks, including blackhole, grayhole, rushing, wormhole, and JF attacks. In this section, we analyze TUF in terms of its ability to defend against these attacks. We assume there is at least one uninfected route between a source node and a destination node. We will refer to the example network shown in Fig. 2 in our discussions, where S and D are source and destination nodes respectively, and A, B, C, E, F , and G are intermediate nodes.

Blackhole, Grayhole and JF attacks: In all three attacks, a

malicious node attracts routes passing through it by forwarding control packets normally, but manipulate data packets once it is included in a route. A backhole drops all data packets while a grayhole drops data packets probabilistically. A JF attacker tampers with packets more artistically as described in Section II. If A is a blackhole/grayhole/JF node while other nodes are all benign, the throughput of any route containing A will be affected. Suppose that route $S-F-A-B-G-D$ has been selected and is currently in use. Since the route was chosen because of its minimum delay, it has a small RTT and according to Eq. (2) the route's G_{th} will be relatively high. Suppose A , the malicious node, tampers with data packets, thus dropping the goodput of the $S-D$ connection. Once the goodput drops below G_{th} , TM will readily detect this and the route will be tagged as being infected. Then, TUF will trigger a new route discovery with duplicate suppression disabled to find more routes, including $S-F-A-E-G-D$, $S-F-C-B-G-D$, and $S-F-C-E-G-D$. The LARR algorithm will output either $S-F-C-B-G-D$ or $S-F-C-E-G-D$, whichever one has the smaller route index.

Rushing attack: In a rushing attack, the malicious node suppresses ROUTE REQUEST packets forwarded by other nodes by disseminating ROUTE REQUESTs very quickly. If the malicious node forwards packets without any modification and makes itself transparent to the network layer, this behavior is called a *repeater attack*. In Fig. 2, assume that A is malicious while other nodes are all benign. Assuming that A has successfully launched a rushing attack, it can take two types of actions to disrupt routing. The first action A can take is not forwarding the ROUTE REPLY packet. However, this abnormality can be easily detected; S will send a new ROUTE REQUEST packet disabling duplicate suppression in response to no receipt of ROUTE REPLY. Then S will receive at least one ROUTE REPLY packet via routes that do not include A (e.g., $S-F-C-B-G-D$). The other action A can take is to forward ROUTE REPLY packets and place itself in the established route. Then A can launch a blackhole/grayhole/JF attack to disrupt communications between S and D . Again, this is ineffective since as discussed above, TUF counter the latter attacks.

Wormhole attack: In a wormhole attack, a pair of malicious nodes tunnel packets from one part of the network to another, thus disrupting routing by short circuiting the normal flow of routing packets. This attack can be regarded as a colluding rushing attack. Similar to rushing attacks, a wormhole attack is harmful only when launched together with blackhole/grayhole/JF attacks. Since TUF can detect and thwart these concomitant attacks, it can effectively counter harmful wormhole attacks. In Fig. 2, suppose that A and E compose a wormhole, and other nodes are all benign. Also, suppose that both A and E attempt to disrupt routing by launching a blackhole attack after a route has been established. Two scenarios are possible. In the first case, A and E add themselves to the list of route nodes and are seen by other nodes. Because A and E are blackhole nodes, route $S-F-A-E-G-D$ will be tagged as being infected by TM, and the LARR algorithm will return $S-F-C-B-G-D$ as the next route to use, which is an uninfected route. In the second scenario, both A and E make themselves transparent to other nodes. Initially, route $S-F-(A)-(E)-G-D$ will be selected and used. Because A and E are blackhole nodes, TM will detect that the route is infected, and a route rebuilding process will be invoked. As a result, three new routes will be found: $S-F-(A)-B-G-D$, $S-F-C-(E)-G-D$, and $S-F-C-B-G-D$. All three routes have the same alikeness degree with respect to the routes recognized as being infected. Hence, either the first or second route will be selected, since they have shorter hop counts compared to the third route. Because both $S-F-(A)-B-G-D$ and $S-F-C-(E)-G-D$ contain a blackhole node, they will both be tagged as infected routes in the next two trials. In the final trial, the

uninfected route $S-F-C-B-G-D$ will be selected.

V. SIMULATION

A. Simulation Environment

In this section, we perform simulations using the simulation tool *ns-2* to characterize the parameters used in TUF and demonstrate its effectiveness in mitigating JF attacks.

We consider networks of 200 nodes in a 2000m \times 2000m square area. Nodes use the 802.11 MAC with a 250m communication range. Nodes move based on the Manhattan model [3], which is designed to model the movement of independent nodes in an urban environment. In the model, node movement is restricted to a well-defined map of intersecting, bi-directional vertical and horizontal streets. For our simulations, horizontal streets are spaced 75m apart and vertical streets are spaced 150m apart. At intersections, nodes will continue on the same street with a probability of 0.5 or turn left or right with a probability of 0.25 for either direction. A node's maximum velocity is 3m/s (a fast walking pace) for these simulations. When a node reaches the end of the street, it will switch lanes and go the opposite direction. Using this model, we generate six random movement patterns for our simulation. All the results to be presented are averaged over the six movement patterns.

We simulate 5-flow and 50-flow networks with 0, 16, 25, and 49 malicious nodes. The flows use TCP Reno [6] senders with standard TCP receivers. Each flow generates traffic at a constant rate of 1K Bytes/second. Malicious nodes launch a Periodic Drop JF attack in which JF nodes forward all control packets and drop data packets for 300ms every second. The period of one second was shown to be the worst case in [1]. The parameters for the number of nodes, flows and malicious nodes are based on those used in the simulation experiments of [1]. Each simulation run lasts 300 seconds.

B. Simulation Results

1) Characterization of TUF parameters

We need to define two parameters used in TUF, p and r_a . We first simulate DSR without malicious nodes, and observed an average packet loss ratio $p = 0.05$. Then fixing p to the observed value, we vary r_a to calculate the number of false positives, which is the number of re-routing requests raised by TM when there is no attack. Fig. 3 shows the results. It can be seen that the false positive is not zero even when r_a approaches zero. This is because it is possible for the TM to invoke route rebuilding requests when link quality degrades, which otherwise would trigger re-routing with ROUTE ERROR messages. Therefore, this scenario cannot be really regarded as false positives. In fact, as Figs. 4 and 5 show, TUF has greater throughput when there is no malicious node. This means that our so-called "false positives" are actually beneficial because it helps the nodes build up new routes more quickly. Since r_a should be maximized to make TM more sensitive to attacks, we should choose the maximum value of r_a that does not cause too many false positives. Finally $r_a = 0.8$ was used for the rest of our simulations because its false positives are no greater than $r_a = 0.4$. A value smaller than 0.4 was not favored since in that case the minimal loss of sensitivity to attacks (50%, from 0.8 to 0.4) would exceed the maximal decrease of false positives (19%, from 91.3 to 74).

2) Data Throughput and Control Traffic

Figs. 4 and 5 show the data throughput and control traffic for both 5-flow and 50-flow simulations. All results are normalized to the value obtained with the network running DSR with no malicious nodes. TUF shows an increase in the achieved data throughput for both 5 and 50 flows for all cases of malicious nodes. The observed benefit from TUF is smaller with the larger flow set because the

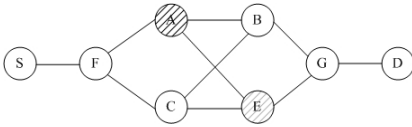


Fig. 2. Example network.

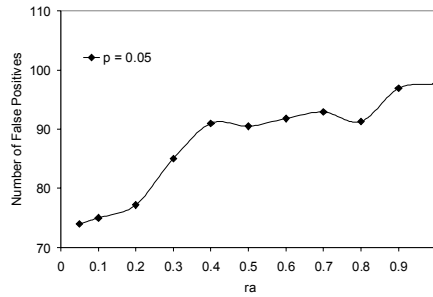


Fig. 3: Number of false positives.

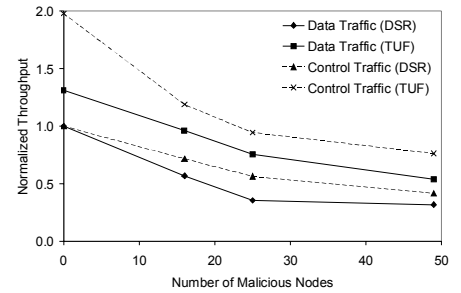


Fig. 4: Data and control throughput for 5 flows.

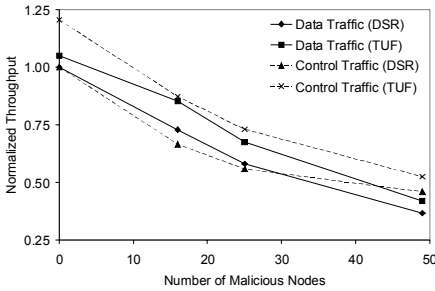


Fig. 5: Data and control throughput for 50 flows.

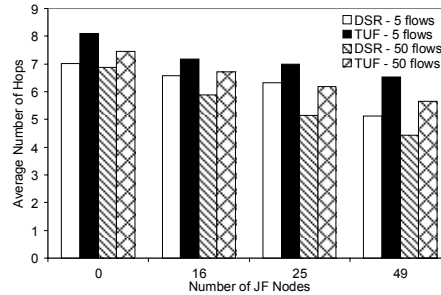


Fig. 6: Average route length.

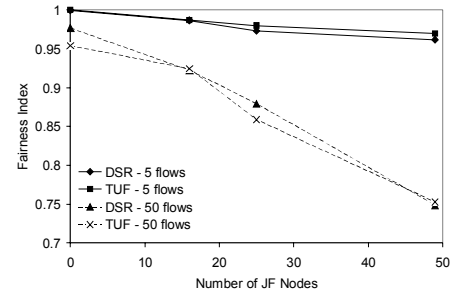


Fig. 7: System fairness.

flows are limited not only by malicious nodes but also by congestion caused by contention for network resources. We also noticed increased control traffic as the expense of the increased data throughput, which results from increased route discovery attempts.

3) Average Successful Route Length

The route length averaged over all successfully transmitted packets is an indicator of a protocol's ability to maintain multi-hop routes which tend to be affected the worst by malicious nodes. Fig. 6 shows that the average route length, measured in number of hops, is longer using TUF than using DSR under all scenarios, indicating that TUF is more resilient against attacks.

4) System Fairness

Jain's fairness index, used in [1], is computed using long-term average throughput and is a way of evaluating how well network bandwidth is shared. An index closer to one is desirable since it indicates that all flows receive an equal share of the network bandwidth. The averaged simulation results, shown in Fig. 7, indicate that TUF does not affect system fairness, which implies that the increased throughput in TUF is relatively equal for every flow, no matter how many hops a flow has.

VI. CONCLUSION

This paper presents a novel secure routing architecture for MANETs called TUF routing. TUF takes a cross-layer approach that uses TM to detect abnormalities at the transport layer while responding to them by rebuilding a route at the network layer. Our analysis shows that TUF is able to thwart a variety of insider attacks as well as protocol-compliant attacks. In addition, simulation results show that TUF is effective in thwarting JF attacks in certain network environments.

As part of our future work, we plan to explore the possibility of adapting TUF to on-demand distance-vector routing protocols. We are also interested in the optimization of TUF parameters under differing network conditions.

REFERENCES

[1] I. Aad, J. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc

networks," In *Proc. MobiCom*, Sep. 2004, pp. 202–215.
 [2] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to Byzantine failures," In *Proc. WiSe*, Sep. 2002, pp. 21–30.
 [3] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of Routing protocols for Adhoc Networks," In *Proc. INFOCOM*, Mar. 2003, pp. 825–835.
 [4] S. Buchegger and J.-Y. Le Boudec, "Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks," In *Proc. 10th EuroMicro Workshop on Parallel, Distributed and Network-based Processing*, Jan. 2002, pp. 403–410.
 [5] R. Chen, M. Snow, J.-M. Park, M. T. Refaei, M. Eltoweissy, *Defending against Routing Disruption Denial-of-Service Attacks in Mobile Ad Hoc Networks*, Technical Report TR-ECE-05-11, Dept. of ECE, Virginia Tech, Nov. 2005, available at: <http://www.arias.ece.vt.edu/publications/TechReports/Chen-2005-11.pdf>.
 [6] S. Floyd and T. Henderson, *The New Reno Modification to TCP's Fast Recovery Algorithm*, RFC 2582, Apr. 1999.
 [7] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," In *Proc. INFOCOM*, Mar. 2003, pp. 1744–1753.
 [8] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure on-demand routing protocol for ad hoc networks," In *Proc. MobiCom*, Sep. 2002, pp. 12–23.
 [9] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet leases: a defense against wormhole attacks in wireless networks," In *Proc. INFOCOM*, Mar. 2003, pp.1976–1986.
 [10] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," In *Proc. WiSe*, Sep. 2003, pp. 30–40.
 [11] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) (Internet-Draft)*, Mobile Ad-hoc Network (MANET) Working Group, IETF, July 2004.
 [12] J. Kong, X. Hong, J. Park, Y. Yi, M. Gerla, *L'Hospital: Self-healing Secure Routing for Mobile Ad-hoc Networks*, Technical Report TR-040055, Computer Science Department, University of California, Los Angeles, Jan. 2005.
 [13] J. Kurose and K. Ross, *Computer Networking: A Top-down Approach Featuring the Internet, Third Edition*, Addison Wesley, 2004.
 [14] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," In *Proc. MobiCom*, Aug. 2000, pp. 255–265.
 [15] K. Nahm, A. Helmy, and C. Kuo, "TCP over Multihop 802.11 Networks: Issues and Performance Enhancement," In *Proc. MobiHoc '05*, May 2005, pp. 277–287.
 [16] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," In *Proc. CNDS*, Jan. 2002.
 [17] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," In *Proc. SIGCOMM*, 1998, pp. 303–314.
 [18] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," In *Proc. SIGCOMM*, Oct 1994, pp.234–244.
 [19] H. Takahashi, M. Saito, H. Aida, Y. Tobe, and H. Tokuda, "Estimated-TCP-throughput maximization based routing," In *Proc. LCN*, Oct. 2003, pp.120–129.
 [20] W. Yu, Y. Sun, and K. J. R. Liu, "HADOF: Defense against routing disruptions in mobile ad hoc networks", In *Proc. INFOCOM*, Mar. 2005, pp. 1252–1261.