This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

On Mitigating In-band Wormhole Attacks in Mobile Ad Hoc Networks

Xu Su **Rajendra V. Boppana**
Computer Science Department, UT San Antonio
xsu@cs.utsa.edu boppana@cs.utsa.edu

Abstract— **Colluding malicious insider nodes with no special hardware capability can use packet encapsulation and tunnelling to create bogus short-cuts (in-band wormholes) in routing paths and influence data traffic to flow through them. This is a particularly hard attack using which even a handful of malicious nodes can conduct traffic analysis of packets or disrupt connections by dropping packets when needed. Using simulations we show that a disproportionately large amount of traffic goes through routes with wormholes even when a secure routing protocol such as Ariadne is used. To mitigate this, we propose distributed techniques based on the propagation speeds of requests and statistical profiling; they do not require network-wide synchronized clocks, do not impose any additional control packet overhead, and need only simple computations by the sources or destinations of connections. We implemented our techniques in Ariadne and evaluated their effectiveness using the Glomosim simulator. Our results indicate that in-band wormhole creation and usage can be reduced by a factor of 2-10. Also, the false alarm rates of the proposed techniques are very low and have little impact on normal network operation, making them practical for MANETs.**

I. Introduction

Mobile ad hoc networks (MANETs) have a wide range of applications, especially in military operations and emergency and disaster relief efforts. However, MANETs are more vulnerable to security attacks than conventional wired and wireless networks due to the open wireless medium used, dynamic topology, distributed and cooperative sharing of channels and other resources, and power and computation constraints. Attacker nodes may be insiders – nodes that have the necessary cryptographic keys, participate in normal network operations but attack surreptitiously as needed – or outsiders – nodes that do not have the keys and can only attack by jamming radio channels or by replaying transmissions of legitimate nodes. Attacks launched by colluding malicious nodes are very hard to detect and mitigate. A widely studied example of colluding attacks is the wormhole attack [1] in which colluding nodes with high speed channels and other resources replay packet transmissions to create wormholes and cause route falsification. We call these attacks *out-of-band wormhole* attacks.

We are interested in route falsification attacks caused by insider nodes without special resources such as out-of-band high-speed channels. We show that if an adversary compromises the software of a few insider nodes, then powerful wormhole type attacks can be launched using only the network channels and without requiring physical access to the compromised nodes. In such attacks, colluding insider nodes create bogus short-cuts (wormholes) to routes via existing wireless data paths (in-band channels) and induce other nodes to use these falsified routes. We call these attacks *in-band wormhole* attacks. The current secure on-demand routing protocols (SRPs) for ad hoc networks [2], [3], [4], [5], [6] mitigate other forms of route falsification, but are susceptible to these in-band wormhole attacks.

Intrusion detection techniques (IDTs) [7], [8], [9], [10] may not be able to detect and mitigate those attacks since (a) the frequency of falsification and the additional transmissions by malicious nodes are low and (b) the current IDTs consider packet dropping as the only (or major) attack. Instead of dropping packets, in-band wormhole attackers can conduct traffic analysis [11], which includes not only cryptanalysis but also obtaining communication patterns such as sender-recipient matchings, traffic volume, traffic shape, and traffic duration, and launch other attacks later accordingly. Traffic analysis is problematic, especially in military operations.

We propose mechanisms to complement the existing secure routing protocols to resist the creation of these in-band wormholes, and thus reduce the incidence of in-band wormhole attacks. Our techniques are based on reducing request packet delays and statistical profiling. These techniques do not require network-wide synchronized clocks and do not impose any additional control packet overhead. We implemented our techniques in a widely studied secure routing protocol called Ariadne [5] and evaluated their effectiveness using the Glomosim simulator [12]. The results show that our techniques achieve high detection rate and have negligible impact on network throughput.

II. Background and In-band Wormhole Attacks

A. Secure Route Discovery

Most of the existing secure on-demand routing protocols are based on either Ad hoc On-demand Distance Vector routing protocol (AODV) [13] or Dynamic Source Routing (DSR) [14]. They use route discovery to learn new routes and route error propagation to remove stale routes. The route discovery consists of two stages. (1) *Route request stage* – the source node floods the network with a route request control packet (RREQ), and each intermediate node rebroadcasts the RREQ the first time it hears. (2) *Route reply stage* – upon receiving a RREQ, the destination sends a route reply packet

This research was partially supported by NSF grants EIA-0117255, CRI-0551501 and AIA grant F30602-02-1-0001.

1-4244-0353-7/07/$25.00 ©2007 IEEE

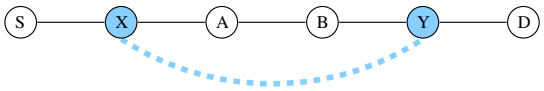

Fig. 1. Route discovery example. Solid line represents physical wireless link. The dotted line represents in-band wormhole or packet tunnel between X and Y via A and B.

(RREP), which is propagated to the source in the reverse path of the RREQ.

We use Ariadne [5], designed to prevent both RREQ and RREP falsification, as the example secure routing protocol. Ariadne requires each node to attach an authentication code to each RREQ packet it forwards. The authentication code is based on the RREQ packet contents including the previous node's authentication code. Either destination or source (depending on the version of Ariadne used) verifies these authentication codes. So, for successful route falsification, RREQs must be modified carefully.

B. In-band Wormhole Attacks

We describe how malicious insider nodes can collude without *a priori* knowledge of the network and using only in-band channels and induce legitimate nodes to use routes through them. Such attacks ensure that there are two or more malicious nodes in a route, one close to the source and another close to the destination. This is desirable for traffic analysis requiring message timing and volume [11]. We use a 5-hop path $S-X-A-B-Y-D$ taken by a RREQ packet from S to D, Fig. 1, to illustrate these attacks. Nodes X and Y are colluding malicious nodes and create a packet tunnel between them via normal nodes A and B.

If Y obtains the authentication code generated by X for RREQ from S, then it can fabricate a RREQ which indicates $S-X-Y$ as the path instead of $S-X-A-B-Y$ and send it to D. If necessary, the corresponding RREP is tunnelled from Y to X via B and A. This results in a false route $S-X-Y-D$ with fewer hops; it cannot be detected even after verification by source/destination. If S chooses this bogus path, X and Y have the option of delivering the data packets or dropping them. We show below two ways in which a malicious node can obtain the authentication code generated by its colluder.

Reactive Attack (Attack 1): If RREQs carry the path traversed in clear text, a malicious node (Y, in our example), upon receiving a RREQ, can check if the path already contains another malicious node more than one hop away from it, and query that node (X) for the authentication information it generated. This attack is effective only when RREQs carry path list in clear text. However, the malicious nodes do not generate traffic unnecessarily, which reduces the risk of detection by IDTs. This attack succeeds in SRP [3], Ariadne [5] and endairA [6].

Proactive Attack (Attack 2): Another approach is to have the node close to source (X, in our example) send the authentication information to all other malicious nodes proactively. To faciliate this, malicious nodes may occasionally initiate RREQs to discover the routes among themselves. This attack succeeds in all route discovery based SRPs including SAODV [2] and ARAN [4].

Number of Nodes	50
Node Speed	[1-19]m/s
Node Mobility	Modified Random Waypoint
Pause Time	0-900 seconds
Field Size	1500 m $\times$ 300 m ($\rho = 22$) 1300 m $\times$ 800 m ($\rho = 10$)
Radio Range	250 m
MAC	802.11
Number of Traffic Pairs	10
Traffic Load	100-300 Kbps (CBR/UDP)
Data Packet Payload	500 bytes
Link BW	2 Mbps
Initial RREQ Timeout	0.5 seconds
Maximum RREQ Timeout	10 seconds
Route Cache Size	32 routes with FIFO replacement
# of Attackers	0, 4, 8, or 12
Hash length	128 bits
Warmup time	50 seconds
Filter parameters: δ, μ, and ϕ	$\frac{1}{8}$, $\frac{1}{8}$, and 2, respectively

Fig. 2. Simulation Parameters. Traffic load, pause times, or number of attackers are varied (default values: traffic load = 100kbps, pause time = 0 second). The modifications to random waypoint model for node mobility are as given in [15] to avoid clustering of nodes in the middle and gradual decay of average node speed. Node density, ρ, is the average number of nodes in a radio transmission area.

III. Impact of In-band Wormhole Attacks

We use the low overhead, MAC version of Ariadne given in [5] as the representative SRP. We used the Glomosim simulator, v2.03 [12] to evaluate the impact of the attacks. We implemented Ariadne and the two types of in-band wormhole attacks described earlier in Glomosim.

The simulation parameters used are listed in Fig. 2. We used two rectangular shapes: corridor with length 5 times the width ($1500 \times 300m^2$) and golden rectangle with length approximately 1.6 times the width ($1300 \times 800m^2$). With 50 nodes, the node densities (ρ), the average number of nodes in a radio transmission area, vary from 10 to 22. The following metrics are used to evaluate the performance of Ariadne and the impact of in-band wormhole attacks on it.

- *Tunnelled Paths Created.* The fraction of the routes that are compromised by malicious nodes.
- *Fraction of Packets Sent over Malicious Paths.* The fraction of packets sent through paths which contains two or more malicious nodes out of the total number of packets sent by sources.
- *Route Request Latency.* The average time elapsed from the time a route request packet is first sent to the time it is received at its destination;
- *Route Discovery Latency.* The average time elapsed from the time a route request packet is sent to the time a reply packet is received. If a source receives multiple replies to its request, then route discovery latency is calculated for each reply.

All experiments were run for 900 seconds with first 50 seconds used for warm up. No attacks are launched by malicious

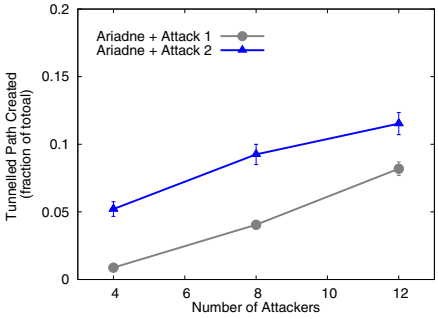

Fig. 3. Fraction of routes that are falsified (node density ρ=22).

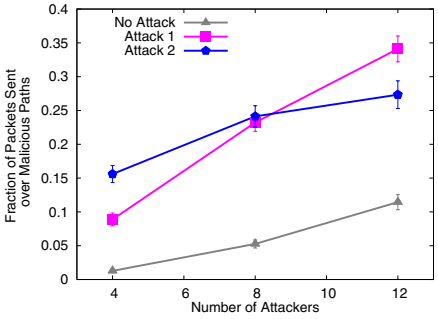

Fig. 4. Fraction of packets sent over malicious paths (node density ρ=22).

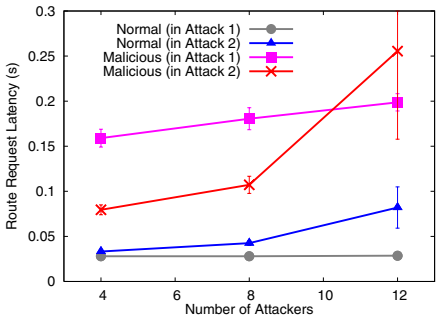

Fig. 5. Route request latency (node density ρ=22).

nodes during the warm up period. Each configuration was repeated 20 times and the results were averaged; the 95%-level confidence intervals are indicated for all data points. We show the impact of the attacks for the high-density corridor network only; the results for the cases not reported are similar to those reported.

The number of wormholes created by the attacks is low as shown in Fig. 3. However, paths with wormholes are used more frequently than regular ones due to "shorter" path length. Therefore, the percentage of data packets sent via tunnelled paths is much higher. (We kept track of but did not drop data packets sent over tunnelled paths.) Fig. 4 gives the fraction of data packets sent over malicious paths (which contains two or more malicious nodes). In Ariadne with 8 malicious nodes, about 5% of total data packets went through malicious paths when no in-band wormhole attacks were launched, while about 20% of total data packets went through malicious paths in Attack 1 and Attack 2. If malicious nodes keep track of the paths used and change their movement to get closer to sources and destinations (not implemented), the damage will increase dramatically.

In order to improve the resistance of Ariadne to wormhole attacks, we analyzed the delays of malicious and normal requests and replies. The request delays are given in Fig 5. The average route request latency for malicious RREQs is about 4 times that of normal RREQs in Attack 1, and 2 times that of the same in Attack 2, since malicious nodes need to exchange information through existing data paths. Fig 6 shows a similar difference in delays of route discoveries for both malicious and normal paths. Even though malicious RREQs arrive late at the destination and the corresponding RREPs sent from destination will reach the source later than legitimate RREPs, some of them will be accepted by the source since they contain shorter paths. (DSR, on which Ariadne is based, selects shorter paths in both ns-2 [16] and Glomosim [12] implementations.)

IV. Techniques to Mitigate Wormhole Attacks

In this section, we present packet filtering techniques to reject bogus requests and replies that contain in-band wormhole paths. Our techniques are applicable to existing secure routing protocols that require authentication by each hop during RREQ propagation and end-to-end authentication for RREQs and RREPs. They are based on reducing RREQ delays and statistical profiling of RREQ or RREP delays to prevent creation of in-band wormholes. These techniques may be used by the destination or the source of route discovery.

A. Reduce request packet delays

Routing protocols such as AODV [13], DSR [14] and those based on them specify that routing packets should be propagated at a higher priority than normal data packets. However, that is not enough since malicious nodes can use bogus route reply or route error packets among themselves to exchange attack information speedily. We suggest that, for on demand route discovery schemes that use flooding, requests should be transmitted at a higher priority than all other packets. As we mentioned in Section II-B, in order to create an in-band wormhole, two malicious nodes collude and exchange information between each other using data packets (the use of any other packets increases the risk of detection by IDTs). By ensuring that requests travel faster than all other types of packet, we implicitly increase the time to exchange information among malicious nodes. In other words, the differences between the delays of falsified requests/replies and legitimate requests/replies are likely to be even larger than those shown in shown in Figs. 5 and 6.

B. Use statistical profiling

We propose a distributed and adaptive statistical profiling technique to filter RREQs (by destination) or RREPs (by source) that have excessively large delays. Since different RREQs take varying number of hops, we calculate the upper bound on the per hop time of RREQ/RREP packets so that most normal packets are retained and most falsified packets are filtered. We adapted the retransmit timeout (RTO) calculations used by TCP [17], which captures both the average and deviation of round trip times of a connection, for our purpose. Compared to the intrusion detection techniques [8], [18], only the end nodes in a route discovery monitor and analyze the control packets in our approach. Compared to prior approaches to mitigate wormhole attacks using timing information [1], we do not require network-wide synchronized clocks; each node uses only the RREQs/RREPs it received and its local clock for stastical profiling.

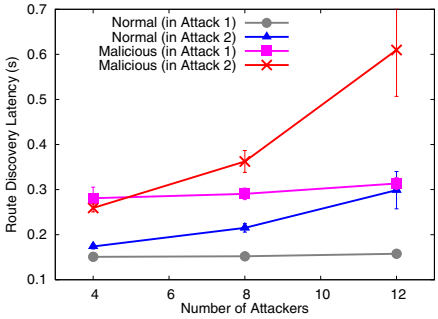

Fig. 6. Route discovery latency (node density ρ=22).

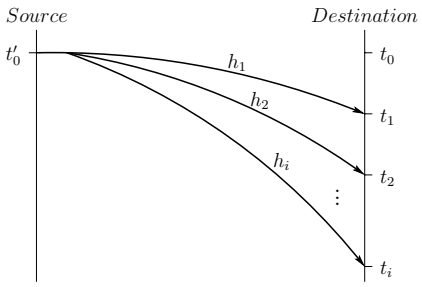

Fig. 7. Multiple RREQs received by destination in a route discovery.

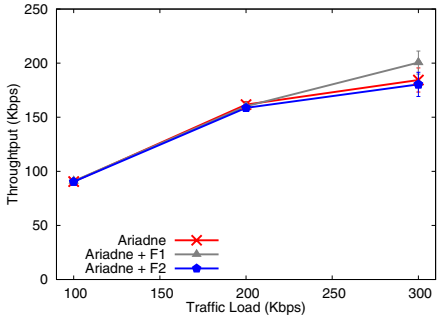

Fig. 8. Impact of filters in a normal network (node density $\rho = 22$).

1) Destination performs statistical profiling (Filter 1): In this design, each destination node filters (discards) RREQs that are targeted to it and have excessively large delays. Consider a route discovery from source S to destination D. Then D receives the first copy of this RREQ with hop count h_1 at its local time t_1, and the second copy of the RREQ with hop count h_2 at time t_2, and so on. Let t_0 denote the destination local time at which the request originated at source. Since the actual value of t_0 is not known, we show below how D can estimate it. The first RREQ with a new sequence number is considered to be a legitimate one, and the destination sends a RREP back to the source. For each duplicate RREQ received, the destination calculates the route request hop time (RHT) which is the time taken by the request packet to reach destination divided by its hop count as given in (1). The destination also computes smoothed average, denoted $avgRHT$, and deviation, $devRHT$, of RHT for all accepted RREQs, as given in (2) and (3). To distinguish between malicious route requests and normal ones, we calculate a cut-off request hop time, $cutoffRHT$, as given in (4). For each duplicate RREQ received, a corresponding reply is generated and $avgRHT$ and $cutoffRHT$ are updated only if this RREQ's RHT is below the $cutoffRHT$. A pseudo code for statistical profiling by destination is given in Fig. 9. Each destination maintains separate $avgRHT$ and $devRHT$ values for each source.

$$RHT_i = \frac{(t_i - t_0)}{h_i} \quad (1)$$

$$diff_i = RHT_i - avgRHT$$

$$avgRHT = avgRHT + \delta \times diff_i \quad (2)$$

$$devRHT = devRHT + \mu \times (|diff_i| - devRHT) \quad (3)$$

$$cutoffRHT = avgRHT + \phi \times devRHT \quad (4)$$

We have experimented with various values, $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$, for δ and μ and found that $\frac{1}{8}$ is the best for both parameters. Assuming that devRHT approximates the standard deviation of sample RHTs, by the law of large numbers in statistics [19], fewer than 5% of normal requests will have RHTs above the cutoffRHT calculated with $\phi = 2$.

Now, we address the issue that the destination does not know the actual value of t_0, its local time when the route discovery is launched. It is noteworthy that knowing source's local time is not useful since the clocks at source and destination are not necessarily synchronized. Therefore, the destination estimates t_0 using (5).

$$t_0 = t_1 - avgRHT \times h_1 \quad (5)$$

In order to compute t_0 for the first time, the initial value of $avgRHT$ is set to 10 ms, which is typically upper end of one hop time (including processing, queueing and transmitting) in the MANETs we simulated. This value is revised to match the network conditions during the warmup time. The initial value of $devRHT$ is set to 0 ms.

2) Source performs statistical profiling (Filter 2): In this design, each source node monitors the RREPs it receives and filters those that have excessively large delays. When the source receives a RREP, it can compute route discovery hop time (RDHT) which is the route discovery time divided by its hop count. RDHT includes the delays of a successful request and the corresponding reply. The source also computes smoothed average, $avgRDHT$, and deviation, $devRDHT$, of route discovery hop time for all accepted RREPs, and $cutoffRDHT$ in the same manner as those for RHT. The initial values for $avgRDHT$ and $devRDHT$ are set to 10 ms and 0 ms, respectively. This technique can be implemented incrementally without requiring that other nodes implement it, but it is likely to be less accurate due to the inclusion of reply packet delays. The pseudo code for source filtering parallels that for destination filtering.

C. Simulation Analysis of Packet Filters

Since random node movements and the contention for shared wireless channels in an ad hoc network can result in unpredictable packet propagation times, it seems unlikely that the time-based profiling techniques described above can actually be effective. Therefore, we reran the simulations of the example MANETs with the packet filters implemented in Ariadne. We use the additional performance metrics given below in this analysis.

- *Throughput.* The total amount of data packets received at all destination nodes in a specified amount of time.
- *Detection Rate.* The percentage of malicious paths rejected by our proposed techniques (related to false negative).

Algorithm for Filter 1: *HandleRequest(RREQ)*

```
1.  if RREQ has a new sequence number then
2.      estimate t_0 using (5);
3.      send a RREP back to the source;
4.      return;
5.  endif
6.  compute RHT using (1);
7.  if RHT > cutoffRHT then
8.      return; //filter this RREQ
9.  endif
10. update avgRHT using (2);
11. update devRHT using (3);
12. update cutoffRHT using (4);
13. send a RREP back to the source;
14. return;
```

Fig. 9. Pseudo code for statical profiling by destination.

- *Shortest Paths Rejected.* The fraction of legitimate paths rejected that are shortest paths (related to false positive).

We experimented with different node pause times (0, 300, 600, and 900 seconds). Due to limited space, we only present results with 0 second pause time; the results with different pause times are similar.

In the first set of experiments, we evaluated the impact of the packet filters on Ariadne in a normal network for different traffic loads. Fig. 8 gives the throughput of Ariadne and proposed filters applied to Ariadne (Fn, n =1, 2, indicates Filter n) for the network with high node density. (The impact of the filters in the network with low node density is similar and the resulta are not given.) At low to moderate traffic loads, the performance impact of the filters is negligible. At high traffic loads, filtering RREQs improves the performance of Ariadne slightly since it reduces the RREP traffic, which is beneficial in a congested network.

In the second set of experiments, we evaluated the impact of the previously described in-band wormhole attacks on Ariadne fortified with the proposed packet filters. We give the results for 100 Kbps traffic load; the results for other traffic loads are similar. Figs. 10 and 11 give the percentage of bogus control packets detected by packet filters. Detection rates of destination based filters are higher than those of source-based filter since most of the delay in creating a wormhole path occurs during request propagation and the cutoff time calculations are more accurate even without synchronized clocks in Filter 1 design. Filters are more effective for attack 1 than attack 2 since the necessary information is exchanged proactively in attack 2 and thus it is likely to be available to the second malicious node sooner than it will be in attack 1. Complementing the high detection rates, the proposed filters have low false alarm rates, as shown in Figs. 12 and 13.

Next, we analyzed the incidence and usage of in-band wormholes on Ariadne with and without packet filters. The case without filters is analyzed in Section III and is included here for comparison purpose. We only give results for the high density network, the results for low density network are similar. Figs. 14 and 15 give the fraction of data packets sent over malicious paths in Ariadne under Attack 1 and under Attack 2, respectively. With packet filters, the fraction of data packets sent over malicious paths is reduced dramatically. Under Attack 1, for Filter 1, the fraction of packets going through malicious paths is even less than that when no attacks are launched. Under Attack 2, the proposed packet filters reduce the use of wormhole by a factor or two or better.

The simulation results indicate that the proposed packet filters help reduce the creation of in-band wormholes without affecting network throughput on Ariadne. Particularly noteworthy is Filter 1, in which a destination performs statistical profiling without requiring clock synchronization between source and destination, and improves the security performance of Ariadne significantly. It is possible that malicious nodes may attack the packet filters by artificially delaying all RREQs/RREPs through them. However, this is unlikely to work when there are other paths (not necessarily shortest paths) among sources and destinations.

V. Conclusions

Secure routing protocols for ad hoc networks are designed to minimize route falsification attacks by non-colluding nodes or avoid problematic routes only when packets are dropped. But they do not handle in-band wormhole attacks launched by colluding, compromised insider nodes without any special hardware capabilities or knowledge of network topology. These attacks are less powerful than the commonly studied wormhole attacks in which the adversary uses special high-speed channels to make wormhole routes faster and the presence of replay nodes is hard to detect. But, in-band wormhole attacks require only software modifications of nodes already inside the network, and thus can be launched more easily.

We have proposed mechanisms to complement the existing secure routing protocols to resist the creation of in-band tunnels, and thus reduce the incidence of in-band wormhole attacks. Our techniques are based on reducing request packet delays, making attacker nodes exchange extra messages in order to fabricate route request packets, and filtering packets with abnormally high per-hop time using statistical profiling. These techniques do not require a network-wide synchronized clocks and do not impose any additional control overhead, and can be incorporated in the current secure routing protocols that do not address wormhole attacks explicitly. Also, our techniques complement the techniques that allow wormholes paths but attempt to mitigate packet dropping by malicious nodes.

We have investigated the effectiveness of our techniques using simulations. Our results show that Ariadne's resistance to in-band wormholes improves by a factor of 2-10. Also, the false alarm rates of the proposed techniques are very low and have little impact on normal network operation, making them highly suitable for MANET protocols.

References

[1] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *Proceedings of IEEE INFOCOM*, 2003.

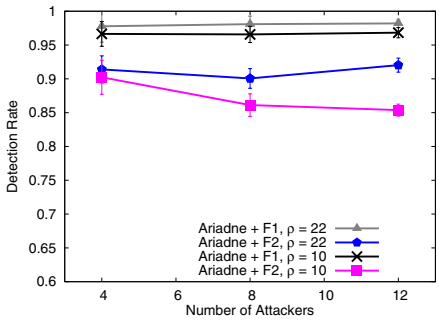

Fig. 10. Fraction of malicious paths rejected by filters in Attack 1.

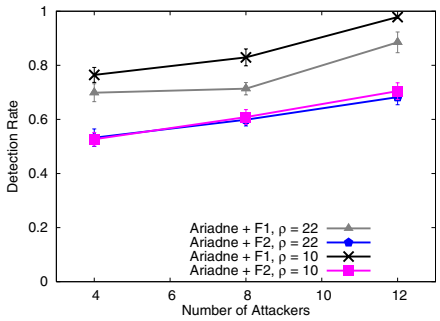

Fig. 11. Fraction of malicious paths rejected by filters in Attack 2.

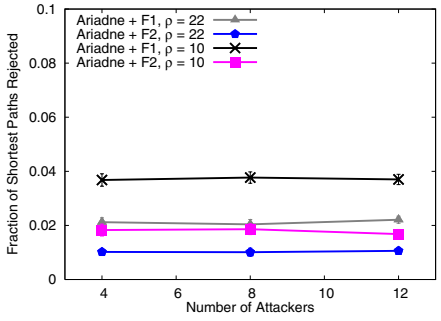

Fig. 12. Fraction of legitimate shortest paths rejected by filters in Attack 1.

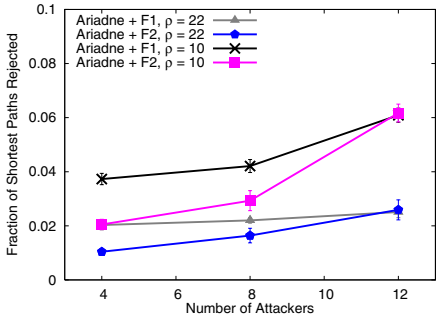

Fig. 13. Fraction of legitimate shortest paths rejected by filters in Attack 2.

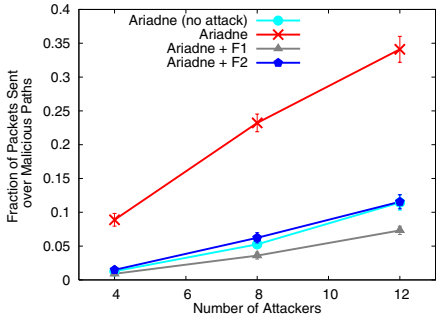

Fig. 14. Fraction of packets sent over malicious paths in Attack 1 (node density $\rho = 22$).

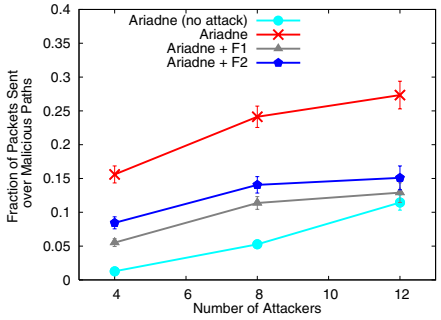

Fig. 15. Fraction of packets sent over malicious paths in Attack 2 (node density $\rho = 22$).

[2] M. G. Zapata, "Secure ad hoc on-demand distance vector (SAODV) routing," in *IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-guerrero-manet-saodv-00.txt*, August 2001.

[3] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, Jan. 2002.

[4] K. Sanzgiri, B. Dahill, B. Levine, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," *Proceedings of IEEE ICNP*, 2002.

[5] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 21–38, 2005.

[6] G. Ács, L. Buttyán, and I. Vajda, "Provably secure on-demand source routing in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1533–1546, 2006.

[7] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to byzantine failures," in *Proceedings of ACM WiSe*, Sept. 2002, pp. 21–30.

[8] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *ACM/Kluwer Wireless Networks Journal (ACM WINET)*, vol. 9, no. 5, September 2003.

[9] S. Buchegger and J.-Y. L. Boudec, "Performance analysis of the confidant protocol: Cooperation of nodes fairness in dynamic ad-hoc networks," in *Proceedings of IEEE/ACM MobiHOC*, 2002.

[10] W. Yu, Y. Sun, and K. J. R. Liu, "HADOF: Defense against routing disruption in mobile ad hoc networks," in *Proceedings of IEEE INFOCOM*, 2005.

[11] J.-F. Raymond, "Traffic analysis: Protocols, attacks, design issues, and open problems," in *Anonymity 2000, LNCS 2009*, 2001, pp. 10–29.

[12] R. Bagrodia et al., "Glomosim: A scalable network simulation environment, v2.03," Parallel Computing Lab, UC Los Angeles, CA, Dec. 2000. [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim

[13] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, *Ad Hoc On Demand Distance Vector (AODV) Routing*, IETF, July 2003, RFC 3561.

[14] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (dsr)," in *Internet Draft, draft-ietf-manet-dsr-09.txt*, April 2003.

[15] J.-Y. L. Boudec and M. Vojnovic, "Perfect simulation and stationarity of a class of mobility models," in *Proceedings of IEEE INFOCOM*, 2005, pp. 2743–2754.

[16] D. Johnson, et al., "Wireless and mobility extensions to ns-2," 1999, Rice Univerity Monarch Project.

[17] W. R. Stevens, Ed., TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley, 1994.

[18] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 48–60, February 2004.

[19] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. John Wiley & Sons, 1991.