Secure Routing Techniques to Mitigate Insider Attacks in Wireless Ad Hoc Networks

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

***Abstract*— Route falsification attacks are easy to launch in wireless ad hoc networks with on demand routing protocols that employ network-wide flooding of control packets for route discoveries. Colluding insider malicious nodes with no special hardware capability can use packet encapsulation and tunnelling to create bogus short-cuts in routing paths and influence data traffic to flow through them. The current secure on demand routing protocols for ad hoc networks are susceptible to these attacks. This paper presented several design guidelines to mitigate the impact of such attacks and a secure on-demand routing (SOR) protocol that incorporates these techniques. SOR is based on pairwise symmetric keys among all nodes in the network. It is flexible and can be tweaked to accommodate desired security and performance criteria. We implemented SOR and a previously proposed secure routing protocol called Ariadne in the Glomosim simulator and evaluated their performances with and without route falsification attacks by colluding insider nodes. Our analysis indicates that SOR is resistent to these attacks and performs well with low overhead in normal networks.**

I. Introduction

An ad hoc network consists of several wireless nodes that are capable of communicating with each other without the use of a network infrastructure or any centralized administration. To facilitate multi-hop communication between non-neighbor nodes, all nodes act as routers. Since ad hoc networks can be formed easily and can handle node mobility and frequent topology changes, they have a wide range of applications, especially in military operations and emergency and disaster relief efforts [6] and to support communication among data sinks in large wireless sensor networks [2].

However, ad hoc networks 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 energy and computation constraints. Malicious nodes can easily launch physical attacks that jam the wireless channel, passive eavesdropping of wireless transmissions and active route falsification attacks due to the open wireless medium and cooperative nature of ordinary nodes acting as routers. The limited energy availability at most, if not all, nodes can be exploited by malicious nodes using resource depletion attacks in which they inject bogus packets and cause normal nodes waste their energy in forwarding them. Physical attacks are addressed by using frequency hopping protocols at physical layer, and eavesdropping can be addressed by link layer and application layer encryption techniques. Of particular interest and challenging are the active route falsification and resource depletion attacks.

We are interested in preventive solutions to route falsification attacks in on demand routing protocols for MANETs. In a route falsification attack, malicious nodes falsify route requests and/or route reply packets to indicate a better path to the source of a data connection, make disproportionately large portion of traffic go through them. When the source selects the falsified path, the malicious nodes can drop data packets they receive silently (denoted, blackhole attack), or forward the packets but keep the information to conduct analysis of communication patterns such as sender-recipient matchings, traffic timing, volume, and shape [12]. The current secure on-demand routing protocols (SRPs) for ad hoc networks [1], [7], [10], [14], [13], [8] mitigate some forms of route falsification by non-colluding malicious nodes, but are susceptible to attacks by colluding insider nodes.

In this paper, we describe route falsification attacks on existing secure on demand routing protocols by colluding insider nodes without special hardware capability. We propose a secure on demand routing (SOR) mechanism which can mitigate those attacks. SOR can be tuned to satisfy the security and performance constraints. We have implemented a routing protocol based on SOR and compared it with Ariadne [7], a secure routing protocol for ad hoc networks. Our simulation results indicate that the SOR protocol has low control overhead and fast route discoveries in a normal network and mitigates route falsification attacks by insider attackers.

The rest of the paper is organized as follows. Section II describes route falsification attacks by colluding insider nodes. Section III presents a new secure on demand routing (SOR) mechanism. Section IV presents the performance of the proposed routing mechanism. Section V concludes the paper.

II. Background and Route Falsification Attacks

A. Basic Route Discovery and Maintenance

Most of the on-demand routing protocols 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 (REQ), and each intermediate node rebroadcasts the REQ the first time it hears. (2) *Route reply stage* – upon receiving a REQ, the destination sends a route reply packet (REP), which is propagated to the source in the reverse path of the REQ.

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

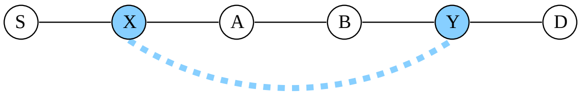

Fig. 1. Route discovery example. Solid lines represent actual wireless links. The dotted line represents packet tunnel between X and Y via A and B.

B. Route Falsification Attacks

In route falsification attacks, malicious nodes falsify REQ and/or REP packets to indicate a better (shorter, faster, or fresher) path to the source of a data connection, make disproportionately large portion of traffic go through them. To prevent arbitrary modification of REQ and REP packets by malicious nodes, a secure routing protocol such as Ariadne [7] requires each node to attach an authentication code (based on the contents of the REQ it received) to each REQ packet it forwards. Either the destination or the source of the route being discovered verifies these authentication codes. A non-colluding malicious node cannot arbitrarily remove some nodes in the path list of REQ because the verification of authentication codes will fail. To be able to successfully shorten the path list in a REQ, the malicious node needs to know the authentication code in the REQ packet that first of the removed nodes received.

Although existing secure on-demand routing protocols can prevent route falsification attacks (e.g., hop count or path modification) by non-colluding malicious nodes, they are still vulnerable to those attacks in which malicious nodes collude. Recently, some researchers have started investigating colluding attacks. For example, endairA [1] can prevent a particular route falsification attack on Ariadne launched by two colluding attackers that are exactly two hops apart.

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 [12]. We use a 5-hop path S-X-A-B-Y-D taken by a REQ packet from source S to destination 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 the REQ from S, then it can fabricate a REQ to indicate S-X-Y as the path instead of S-X-A-B-Y and send it to D. If necessary, the corresponding REP is tunnelled from Y to X via path Y-B-A-X. 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): This attack is effective only when REQs carry path list in clear text. However, the malicious nodes do not generate traffic unnecessarily, which reduces the risk of detection by intrusion detection techniques [9]. An example of this attack is given in Fig. 2. A malicious node (Y, in the figure), upon receiving a REQ (at step 4 in the figure), 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 (step 5 in the figure). This attack succeeds in SRP [10], Ariadne [7], endairA [1], and SDSR [8]. Since SAODV [14] and ARAN [13] do not indicate the path information in REQ packets, they are immune to this attack.

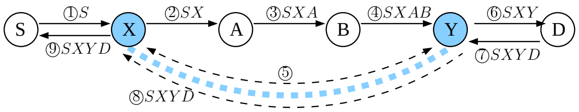

Fig. 2. Reactive attack example. Solid arrows represent REQ (right arrow) or REP packets (left arrow). The dotted line represents the packet tunnel between X and Y. Dotted arrows represent messages tunnelled between X and Y. The numbers indicate the sequence of steps in the attack.

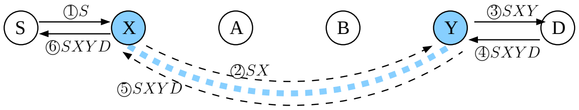

Fig. 3. Proactive attack example.

Proactive Attack (Attack 2): This attack is effective even on protocols that do not indicate path information in REQs. To facilitate this attack, malicious nodes may occasionally initiate REQs to discover the routes among themselves. However, the control traffic generated by them is low enough that they cannot be easily distinguished from normal nodes by an intrusion detection system. An example of this attack is given in Fig. 3. In this attack, malicious nodes close to source (X, in the figure) send the authentication information to all other malicious nodes proactively (step 2 in the figure). This attack succeeds in all route discovery based SRPs including SAODV [14], ARAN [13], SRP [10], Ariadne [7], endairA [1] and SDSR [8].

III. Secure On Demand Routing (SOR)

Based on our analysis of previous protocols, we have identified several design guidelines to mitigate insider attacks. We first describe these features, then present a secure on-demand routing (SOR) protocol that incorporates them.

- *Authenticate intermediate nodes.* Each intermediate node creates an authentication code (marking) based on the contents of the received request packet and places the same in the packet prior to forwarding. These cumulative authentication codes will be verified by the destination (or source) node of the path being discovered. It makes falsification of request packets difficult since the latter malicious node needs to know the authentication code of request packets at the time first malicious node received to falsify request packets. In the reply packet, the destination puts an authentication code based on the contents of REQ packet it accepted and generated a reply; this code can be verified by the source. Therefore, if the REQ packet is not falsified by the time it reaches the destination, then the route discovered by the source is not compromised. This design feature is used in Ariadne. SOR also incorporates this feature.

- *Hide paths taken by route request packets.* If the path taken by a REQ packet is revealed to intermediate nodes in its path, then malicious nodes have the necessary topology information to create short-cuts (Attack 1 in Section II). The path can be hidden by not indicating the path list. Once the request packet reaches destination and a reply is generated, the path may be revealed for performance reasons if necessary. SOR prevents Attack 1 by incorporating this feature.
- *Propagate requests faster.* Since malicious nodes need to exchange information in real-time to fabricate requests, propagating request packets faster than all other types of packets ensures that falsified request packets are usually slower than normal request packets. To achieve this, we (i) give request packets higher priority than all other types of packets (including route reply and error packets), (ii) reduce the size of request packets, and (iii) use light-weight security mechanisms to reduce processing delays at each hop, and (iv) reduce the control traffic in general.
- *Avoid route selection based on hop count.* By using the routes taken by the fastest requests, the chances of having falsified routes can be reduced. For this selection technique to be effective, however, REQs should be forwarded fast. Though ARAN [13] avoids hop-based route selection, it has excessive processing delays at each hop due to the use of digital signatures and thus gives ample time for colluding nodes to fabricate REQs. SOR uses *time-based route selection* that consists of two parts: (i) after the first request is received and replied, the destination should respond to additional requests from the same route discovery only if they arrive within some short duration; (ii) each reply contains a monotonically increasing reply number, and the source chooses the route in the reply with smallest reply number, which corresponds to the fastest request. The reply duration, denoted δ_t, can be used as a design parameter to trade off false positives and resistance to route falsification by colluding insider nodes. SOR combines this feature along with faster propagation of REQs to mitigate Attack 2.

One or more of above features have been used in previous secure on-demand routing protocols, but SOR is the first one to integrate all of them and has the following properties: (i) It is designed to be used as a source routing as well as a table-driven routing protocol; (ii) SOR has several design alternatives such that a route may be disseminated to all nodes in the route (like DSR and Ariadne), only to the source, only to the destination, or any combination of these possibilities even with source routing; (iii) SOR mitigates both non-colluding and colluding route falsification attacks.

We now present a basic version of the SOR protocol. We assume that each node in the path being discovered shares a secret key with the source or destination of the path being discovered. These pair-wise shared keys are used to generate message authentication codes (MACs) and encrypt route information.

A. SOR Protocol

Each node maintains a routing table with the ability to store entire path lists for destinations (same as in AODV [11]). A routing table entry contains source S, source sequence number n_S, destination D, destination reply number n_D, backward hop (BH), forward hop (FH), path list (if available), and maintenance information such as route expiration time.

S → *	: {REQ, S, D, n_S, M_S, CM_S}
X → *	: {REQ, S, D, n_S, M_S, CM_X}
A → *	: {REQ, S, D, n_S, M_S, CM_A}
B → *	: {REQ, S, D, n_S M_S, CM_B}
Y → *	: {REQ, S, D, n_S M_S, CM_Y}
D → Y	: {REP, S, D, n_S, n_D, CM_D}
Y → B	: {REP, S, D, n_S, n_D, (Y), CM_D}
B → A	: {REP, S, D, n_S, n_D, (B, Y), CM_D}
A → X	: {REP, S, D, n_S, n_D, (A, B, Y), CM_D}
X → S	: {REP, S, D, n_S, n_D, (X, A, B, Y), CM_D}

Fig. 4. Route discovery example in SOR. REP and REQ indicate the packet type. n_S is source sequence number and n_D is destination reply number. M_S is a message authentication code (MAC) computed with shared key between S and D over the REQ that S initiates. CM_S is a cumulative MAC computed by S using shared key between S and D over M_S. CM_i ($i = X, A, B, Y$) is a cumulative MAC computed by node i with shared key between itself and S over the cumulative MAC in the received REQ. CM_D is computed by D using shared key between S and D over n_D and CM_Y in the received REQ. The first five lines indicate 1-hop broadcasts, and remaining lines indicate unicasts.

1) Route Discovery: We use the path from S to D through intermediate nodes X, A, B, C, and Y to illustrate the route discovery in the basic SOR protocol. The sequence of messages used for the route discovery is given in Figure 4. The source S of the route discovery process generates a REQ, which contains source S, source sequence number n_S, destination D, a message authentication code (MAC) generated by source with shared key between S and D (M_S), and cumulative MAC computed by S using shared key between S and D over M_S. Each intermediate node, upon receiving the first copy of the REQ from S, computes a new cumulative MAC using shared key between itself and S over the cumulative MAC in the received REQ, replaces the old cumulative MAC with the new one, records transmitter of the REQ as its backward hop (BH), and re-broadcasts the REQ.

When the REQ arrives at the destination D, D first checks the authenticity and freshness of the REQ by verifying the MAC generated by S (M_S). After D verifies M_S, it sends a REP to its previous hop with a monotonically increasing and unique reply number n_D and a MAC which is based on n_D and the cumulative MAC in the received REQ using shared key between D and S. When an intermediate node receives a REP, appends its ID into the path list, records related route information (S, n_S, D, n_D, FH – the node that sent the REP, etc.) in its routing table, and forwards the REP to its BH – the node that sent the corresponding REQ.

After S receives a REP, it can verify the cumulative MAC marked by intermediate nodes and destination. The format of request and reply packets generated or forwarded by node i are given by (1) and (2). M_S enables the destination to screen bogus requests quickly and not reply to them.

$$\begin{aligned} REQ_i &= \{\text{REQ}, S, D, n_S, M_S, CM_i\} \qquad (1)\\ \text{where } M_S &= MAC_{K_{SD}}(\text{REQ}, S, D, n_S),\\ \text{and } CM_i &= MAC_{K_{iS}}(CM_{BH_i})\\ \text{with } CM_S &= MAC_{K_{SD}}(M_S) \end{aligned}$$

$$\begin{aligned} REP_i &= \{\text{REP}, S, D, n_S, n_D, PathList, CM_D\} \qquad (2)\\ \text{where } CM_D &= MAC_{K_{DS}}(n_D, CM_{BH_D}) \end{aligned}$$

BH_i denotes the backward hop of node i and $MAC_{K_{ij}}(M)$ denotes message authentication code (MAC) computed over message M using shared key between node i and node j.

2) SOR Route Maintenance: When a node is unable to transmit a packet to its forward hop in the route, it creates a route error packet (RER) which contains the following information: ⟨RER, RER's originator, unreachable node, route source, error MAC⟩. The RER's originator is set to the address of the intermediate node encountering the error, and the unreachable node is set to the intended next hop to which the packet was attempting to forward.

In order to prevent unauthorized nodes from sending RERs, MAC is used to authenticate RERs using shared key between RER's originator and the source. When the source receives a RER, it can verify the validity of the RER. This can not prevent insider attackers from flooding the network with RERs. Monitoring and detection techniques should be used to determine the nodes that generate unusually large number of RER packets.

SOR is designed to be used as a source routing as well as a table-driven routing protocol. SOR has several design alternatives such that a route may be disseminated to all nodes in the route (like DSR and Ariadne), only to the source, only to the destination, or to both even with source routing. SOR variants and simulation evaluations of them are given in [4]. To make SOR scalable for large networks, probabilistic on-demand key generation mechanisms may be used.

IV. Performance Evaluation

To evaluate the performances of SOR, we used the Glomosim simulator, version 2.03 [3]. We implemented the basic SOR protocol (Section III-A) and the low overhead MAC version of Ariadne (denoted as Ariadne). For comparison purposes, we also simulated the DSR protocol without various optimizations— intermediate node replies, gratuitous route replies, data salvage, and promiscuous listening for routes turned off— as done in [7]. We compared the basic SOR protocol with DSR and Ariadne. The simulation parameters used are listed in Figure 11. The modifications to random way-point model for node mobility [5] are used to avoid clustering of nodes in the middle and gradual decay of average node speed. We use two rectangular shapes: corridors with length 5 times the width ($1500 \times 300m^2$ and $2200 \times 440m^2$) and golden rectangles (GRs) with length approximately 1.6 times the width ($1300 \times 800m^2$ and $900 \times 560m^2$). With 50 nodes, the node densities (ρ, the average number of nodes in a radio transmission area) are about 10 for the larger fields and 22 for the smaller fields. Owing to limited space, we only present results for low-density GR and high-density corridor networks. We chose high-density corridor network to facilitate easy comparisons with prior results on DSR and Ariadne [7]. The results for the cases not reported are similar to those reported for the corresponding node densities.

Number of Nodes	50
Node Speed	[1-19]m/s
Node Mobility	Modified Random Waypoint
Pause Time	0-900 seconds
Field Size	1500 m × 300 m ($\rho = 22$) 1300 m × 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
Reply duration δ_t	10 milliseconds
Initial REQ Timeout	0.5 seconds
Maximum REQ Timeout	10 seconds
Route Cache Size	32 routes with FIFO replacement
# of Attackers	0, 4, 8, or 12
Hash length	128 bits

Fig. 11. Simulation Parameters. Traffic load, pause times, or number of attackers are varied. Reply duration is the duration in which a destination may reply to additional REQs after the first one is received.

The following metrics are used to evaluate the performance of Ariadne and basic SOR and the impact of colluding route falsification attacks (Section II) on them.

- *Throughput.* The total amount of data received in bits/second at all destination nodes in a specified amount of time.
- *Control Overhead.* The amount of control information transmitted in bits/second.
- *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.
- *Fraction of Packets Sent over Malicious Paths.* The fraction of packets sent through malicious paths, which contains 2 or more malicious nodes, out of the total number of packets sent by sources.

Each configuration was simulated 20 times with different random number streams, and the results were averaged; the 95%-level confidence intervals are indicated for all data points.

In the first set of experiments, we compared the performances of DSR, Ariadne, basic SOR in both low-density GR and high-density corridor networks without attacks. The node pause time was 0s in simulations with traffic load varied.

Figures 5 and 6 show the throughput for each protocol. At low traffic loads of 100 kbps and 200 kbps, all protocols

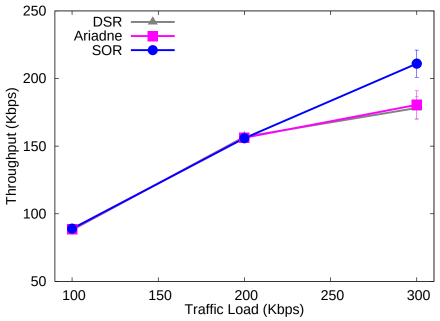

Fig. 5. Throughput (ρ = 10).

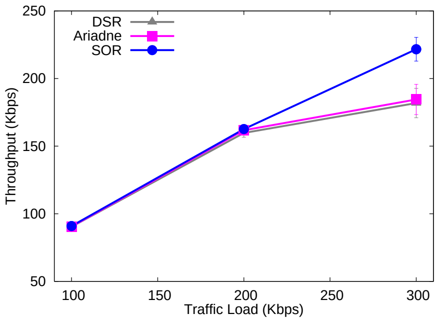

Fig. 6. Throughput (ρ = 22).

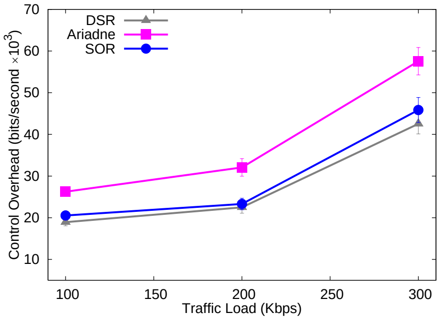

Fig. 7. Control traffic overhead (ρ = 10).

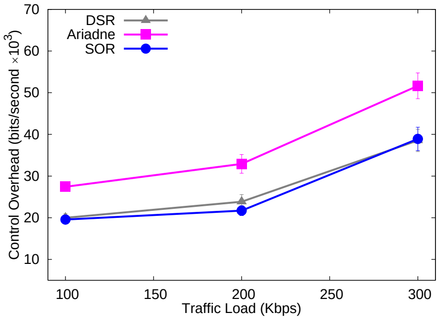

Fig. 8. Control traffic overhead (ρ = 22).

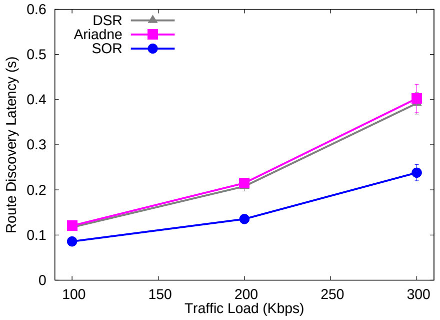

Fig. 9. Route discovery latency (ρ=10).

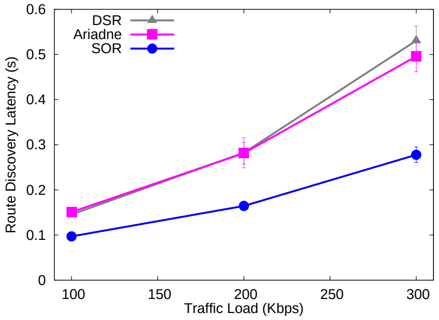

Fig. 10. Route discovery latency (ρ=22).

have nearly same throughput. When the network is congested, at 300 kbps load, SOR gives about 20% higher throughput. Since REQs propagate faster and reach the destination earlier in SOR, broken routes are repaired faster. Also, after the first request is received, a destination accepts request from that route discovery for only a short period of time; this reduces the number of replies sent and, hence the overhead. The reduction in overhead is illustrated in Figures 7 and 8. SOR and DSR have same lower control overhead than Ariadne since both SOR and DSR have smaller REQ size. Figures 9 and 10 shows the average route discovery latencies in a normal network. SOR has low route latency than both DSR and Ariadne due to limited replies by destination, faster REQ propagation, and smaller REQ size.

In the second set of experiments, we evaluated the vulnerabilities of Ariadne and SOR to colluding route falsification attacks in low-density GR and high node density corridor networks. The attacker nodes are neither the sources or destinations of the traffic. Attack 1, in which malicious nodes attack only when clear text path is indicated in REQs, is applicable to Ariadne only. Attack 2 can be launched in both protocols. In these simulations, the traffic load is kept constant at 100 Kbps; the number of attackers is varied from 4 to 12. Nodes are continuously mobile (that is, the pause time between movements is 0 seconds).

We used the fraction of data packets sent over routes via two or more malicious nodes as the metric to measure a routing protocol's resistance to route falsification. (We kept track of but did not drop data packets sent over paths with multiple malicious nodes.) If malicious nodes do not launch route falsification attacks, they are on active routes only when the shortest or fastest paths go through them. The fraction of packets sent in such a case gives the baseline value (denoted as 'No Attack') that the attackers try to increase using the two attacks described in Section II. Figures 12 and 13 give the fraction of paths sent over malicious paths with Ariadne and

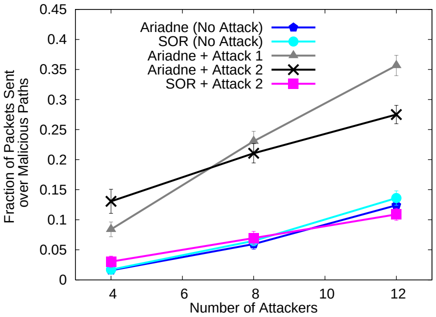

Fig. 12. Data packets sent over routes with multiple malicious nodes ($\rho = 10$).

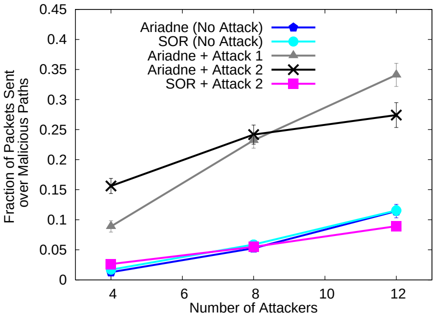

Fig. 13. Data packets sent over routes with multiple malicious nodes ($\rho = 22$).

SOR for low-density GR and high-density corridor networks, respectively. (As indicated before, Attack 1 is applicable to Ariadne only.) For both networks, SOR is very effective: the fraction of packets sent over malicious paths is about the same as it will be if the attackers behaved normally. However, with Ariadne, two (with 12 attackers) to ten (with 4 attackers) times more data packets are sent over malicious paths when malicious nodes launch Attack 1 or 2.

V. Conclusions

Secure routing protocols (SRPs) for ad hoc networks are designed to minimize route falsification attacks. Though there have been several SRPs proposed in literature, they do not handle falsification attacks launched by colluding insider nodes. We described an attack in which insider nodes can easily falsify routes, even when current secure routing protocols are used, without any special hardware capabilities or *a priori* knowledge of network topology. Furthermore, since they can attack without generating a large amount of traffic, their detection using intrusion detection techniques is not likely to be accurate. Using simulations, we showed that the impact of colluding route falsification attacks on existing secure on demand routing protocols (e.g., Ariadne) can be significant.

In this paper, we proposed a secure on demand routing (SOR) mechanism that is flexible and can be tuned to meet desired security and performance constraints. Using simulations, we showed that SOR performed well with low overhead and resists colluding insider attacks.

In future, we wish to evaluate the performances of SOR protocol combined with probabilistic or on-demand key exchange mechanisms.

References

[1] G. Ács, L. Buttyán, and I. Vajda. Provably secure on-demand source routing in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(11):1533–1546, 2006.

[2] IF Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, 2002.

[3] R Bagrodia et al. Glomosim: A scalable network simulation environment, v2.03. Parallel Computing Lab, UC Los Angeles, CA, December 2000.

[4] R. V. Boppana and X. Su. Mitigating colluding insider attacks in mobile ad hoc networks. Manuscript, CS Department, UT San Antonio, 2007.

[5] Jean-Yves Le Boudec and Milan Vojnovic. Perfect simulation and stationarity of a class of mobility models. In *Proceedings of IEEE INFOCOM*, pages 2743–2754, 2005.

[6] I. Chlamtac, Marco Conti, and J. J.-N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, 2003.

[7] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, 11(1-2):21–38, 2005.

[8] F. Kargl, A. Geiß, S. Schlott, and M. Weber. Secure dynamic source routing. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS)*, January 2005.

[9] A. Mishra, K. Nadkarni, and A. Patcha. Intrusion detection in wireless ad hoc networks. *IEEE Wireless Communications*, 11(1):48–60, February 2004.

[10] 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.

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

[12] Jean-Francois Raymond. Traffic analysis: Protocols, attacks, design issues, and open problems. In *Anonymity 2000, LNCS 2009*, pages 10–29, 2001.

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

[14] 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*, 2001.