A Ferry-based Intrusion Detection Scheme for Sparsely Connected Ad Hoc Networks

M. Chuah, P.Yang, J. Han
{chuah, pey204, jih206}@cse.lehigh.edu
Department of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015

Abstract— **Several intrusion detection approaches have been proposed for mobile adhoc networks. Many of the approaches assume that there are sufficient neighbors to help monitor the transmissions and receptions of data packets by other nodes to detect abnormality. However, in a sparsely connected adhoc network, nodes usually have very small number of neighbors. In addition, new history based routing schemes e.g. Prophet have been proposed because traditional adhoc routing schemes do not work well in sparse ad hoc networks. In this paper, we propose a ferry-based intrusion detection and mitigation (FBIDM) scheme for sparsely connected ad hoc networks that use Prophet as their routing scheme. Via simulations, we study the effectiveness of the FBIDM scheme when malicious nodes launch selective data dropping attacks. Our results with different mobility models, ferry speed, traffic load scenarios indicate that the FBIDM scheme is promising in reducing the impact of such malicious attacks.**

Keywords-intrusion detection, routing, prophet, DoS resilience, sparsely connected adhoc networks, disruption tolerant networks.

I. INTRODUCTION

An ad hoc wireless network is a self-organizing network consisting of mobile nodes that are connected via wireless links where nodes not in direct range can communicate through intermediate nodes. On demand routing protocols e.g. [1],[2],[3] are commonly used in ad hoc wireless networks to establish the routing paths between a source-destination pair. However, there are scenarios where the ad hoc networks can be sparsely connected e.g. in battlefield scenarios, in vehicular ad hoc networks. Traditional adhoc routing protocols do not work well in such environments. Thus, recently new stored-and-forward architecture has been proposed to deal with such challenging network environments and new routing schemes [12], [15],[18],[19] have been proposed for such sparsely connected adhoc networks.

Security is critical in military ad-hoc networks since a disruption could cause loss of lives. Thus, both control (e.g. route discovery) and topology update messages need to be authenticated and data packets need to be encrypted. Many proposals have been made in securing adhoc routing protocols e.g. [4],[5],[6]. For example, Ariadne [6] uses a variant of Tesla[8] to provide source authentication for DSR while SEAD [5] uses one-way hash chains to provide efficient secure solutions for DSDV [7].

All the above approaches attempt to provide secured communications in mobile ad hoc networks. However, in a chaotic battlefield environment, authenticated devices are very likely to be captured by the enemy. Additional attacks can be launched by the adversary once he/she compromises an authenticated device. For example, blackhole attacks can be launched where the compromised nodes participate in a routing protocol correctly and then drop all received data packets. Wormhole attacks [9] where two adversaries collude by tunneling packets between each other in order to create a shortcut in the network can also be launched. In such attacks, the adversaries try to increase their chances of being selected as part of the route, and then attempt to disrupt the network by dropping all of the data packets.

Few researches were done to address attacks launched by compromised nodes. Marti et al [10] attempted to address how the routing service can survive selective data dropping attacks. They assume that trusted nodes monitor their neighbors. The solution in [10] may not work well if nodes cannot hear their neighbors forwarding communications due to hidden terminal problem or the use of different modulation schemes etc. In addition, in sparsely connected networks, there may not be enough neighbors that can act as trusted monitoring nodes. In [13], the authors apply intrusion detection techniques typically used in wired networks to ad hoc networks. They proposed that each node overhears all traffic its 1-hop neighbors sent so that it can compare currently observed values of some metrics, e.g. unconditional packet dropping ratio, selective random packet dropping ratio etc, with typical values observed in the past to detect abnormal behaviors. The intrusion detection approach [13] requires nodes to be in promiscuous mode and process all overheard packets, thus it is rather energy consuming. Furthermore, not enough neighbors can be used as monitoring nodes in sparsely connected networks.

In [16], we have proposed a ferry-based intrusion detection scheme (FBIDM) for sparsely connected adhoc networks when a multihop routing scheme [15] with custody transfer feature [17] is used. However, we found that this scheme does not perform well when the network becomes very sparse. Table 1(a) shows the delivery ratio for a network scenario with 40 nodes distributed over 3000x3000 m where 4,8 and 12 malicious nodes are used to launch selective dropping attacks. The results indicate that the delivery ratio drops by 7.9% to 13% as the number of malicious nodes increases from 4 to 12 without any mitigation scheme. With our mitigation scheme in [16], the performance degradation can be improved to 4.6 to 9.5% for similar attack scenarios. Thus in this paper, we describe a new FBIDM scheme that can be used for sparsely connected adhoc networks that run history-based routing schemes e.g. Prophet [18], MaxProp[19]. We also present some simulation results that demonstrate the effectiveness of the new FBIDM scheme in mitigating the delivery degradation caused by the data dropping attacks. A quick overview of the effectiveness of the new FBIDM scheme over the old FBIDM scheme [16] can be seen in Table 1(b). The new scheme can improve the delivery ratio by 73% (from 0.26 to 0.45) to 86% (0.14 to 0.26). In addition, we study how mobility models and some design parameters affect the detection and mitigation capability of the FBIDM scheme. Our results indicate that the FBIDM scheme is quite promising.

# of malicious nodes	4	8	12
Without attack	0.546	0.546	0.546
With attack but without mitigation	0.503	0.482	0.473
With attack but with mitigation	0.521	0.503	0.494

Table 1(a): Delivery Ratios with MH routing and the old FBIDM scheme.

# of malicious nodes	4	8	12
Without attack	0.64	0.64	0.64
Without mitigation	0.26	0.20	0.14
With Milcom06 mitigation scheme [16]	0.28	0.24	0.18
With new scheme	0.45	0.34	0.26

Table 1(b): Delivery Ratios with Prophet and the old and new FBIDM schemes.

The rest of the paper is organized as follows: in Section II, we give an overview of how Prophet routing scheme works and describe a threat model where attacks can degrade the delivery performance by launching selective data dropping attacks. In Section III, we present our ferry-based intrusion detection and mitigation (FBIDM) scheme. In Section IV, we present some simulation results demonstrating the usefulness of the FBIDM scheme. In Section V, we conclude with some future work that we intend to explore.

II. OVERVIEW OF PROPHET AND THREAT MODEL

In [18], the authors proposed a routing protocol called Prophet which uses the history of encounters and transitivity for intermittently connected networks. This probabilistic routing scheme establishes a probabilistic metric called delivery predictability at every node A for each known destination B. This metric indicates how likely it is that node A will be able to deliver a message to that destination. The delivery predictability ages with time and also has a transitive property, i.e., a node A that encounters node B which encounters node C allows node A to update its delivery predictability to node C based on its (A’s) delivery predictability to node B and node B’s delivery predictability to node C. In Prophet, a node will forward a message to another node it encounters if that node has higher delivery predictability to the destination than itself. Such a scheme was shown to produce superior performance than epidemic routing [20]. The three equations used for updating the delivery predictability are as follow:

$$P(a,b) = P(a,b)_{old} + (1 - P(a,b)_{old}) * \alpha$$

$$P(a,b) = P(a,b)_{old} \times \gamma^{k}$$

$$P(a,c) = P(a,c)_{old} + (1 - P(a,c)_{old}) * P(a,b) * P(b,c) * \beta$$

In [18], α is set to 0.75, β is set to 0.25 and γ is set to 0.98. Each node broadcasts a beacon periodically. The beacon contains the delivery predictability values from this node to all other nodes. Such delivery predictability values are updated upon receiving beacons from other nodes.

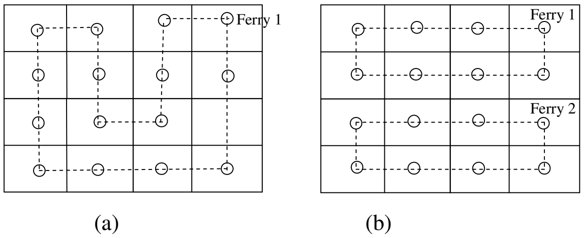
For the threat model, we consider the case where a device or a set of devices could be compromised and be under the control of an adversary or set of adversaries that can collude. Once an adversary has control of an authenticated device, protocols which rely on authentication to provide security services become of little use. Attacks where the adversary has full control of an authenticated device and can perform arbitrary behavior to disrupt the system are referred to as Byzantine attacks [11]. Authentication and data integrity mechanisms cannot protect against such attacks.

In this paper, we assume that compromised nodes launch the following attacks: Each malicious node, say node *a*, attacks six nodes by increasing the delivery predictabilities from itself to these 6 nodes by a constant value i.e. increases *P(a,b)* by 0.5 where b is one of the 6 chosen nodes. Such actions increase the chances of node *a* being selected as the next hop node for relaying messages to the nodes that are being attacked. Once node *a* is successful in being selected as the next hop node, node *a* will drop 50% of the messages that it receives from each flow. Other attack models are possible but we leave those for future work.

Destination	< DP on Encounter, Timestamp>		
	First	Second	Third
Node 1	(0.984, 210)	(0.964, 1860)	(0.950, 9800)
Node 2	(0.983, 910)	(0.984, 4890)	(-- , --)
Node 3	(0.938, 710)	(0.999, 5960)	(-- , --)
Node 4	(0.996, 7720)	(-- , --)	(-- , --)
Node 5	(0.980, 150)	(0.982, 1130)	(0.982, 4670)

III. OVERVIEW OF THE FBIDM SCHEME

In our FBIDM scheme, we use special nodes that are referred to as ferries to provide intrusion detection services to regular nodes. These ferries travel along fixed routes and stop at various locations within their routes. For example, one can divide the geographical area into multiple cells and have ferries visit the center of each cell using some fixed routes as shown in Figures 1(a), and (b) for the single ferry, and two ferries scenarios.



(a) (b)

Figure 1: Ferry Routes for Intrusion Detection

At each location that a ferry stops, a ferry broadcasts a secret service message that each legitimate node knows how to decipher. This can be done by having the ferry encrypt the message using a private key and assuming all legitimate nodes know the public key of the ferry. After receiving the secret message, each legitimate node shares some encounter and delivery predictability information it has with the ferry. The ferry correlate such information from all nodes to identify any potential malicious nodes.

Apart from keeping the delivery predictability to other nodes (as shown in Table 2(a)), each node *i* also maintains a table of the last M values of its delivery predictability to other nodes (say node *j*) in the network just before its connectivity with these nodes disappear, and the times when the node v loses such connectivity with the nodes (as shown in Table 2(b)). This table is referred to as the delivery encounter table (DET). As an example, in Table 1(b), we show that node 0's past three encounters with node 1 ends at time (210,1860, 9800) seconds respectively and the delivery predictability values at these times are (0.984, 0.964, 0.95) respectively.

Table 2(a): Node 0's Delivery Predictability Table (DPT)

Destination	DP
Node 1	0.687
Node 2	0.233
Node 3	0.083
Node 4	0.003
Node 5	0.822

Table 2(b): Node 0's Delivery Encounter Table (DET)

When a node *i* hears the ferry's announcement, it shares its DET with the ferry. The ferry then checks for consistencies between the values of *P(i,j)* and *P(j,i)* obtained from nodes *i* and *j* respectively. The pseudo code for such checking is shown in Figure 2. The ferry compares the most recent encounter times of these two nodes from their respective DETs. If these two times are close (less than Th1), then, the ferry computes an estimate of what *P(j,i)* will be assuming that the most recent $P(i,j)^{MR}_{encournter}$ is valid . If the estimated *P(j,i)* differs from *P(j,i)* reported, then, node j is listed as suspicious.

```
// evaluate whether $P_{real}(j,i)$ is false declaration; if yes, node $j$ is suspicious.
Procedure IntrusionDetection( $P_{real}(j,i)$ )
    Retrieve $DET_{i,j}$ and $DET_{j,i}$ for nodes $i$ and $j$ from $DETDB$;
    // if $T[P(i,j)_{encoutner}] \cong T[P(j,i)_{encounter}]$
    if $(|T[P(i,j)^{MR}_{encounter}] - T[P(j,i)^{MR}_{encounter}]| < th1)$ then
        // if $P(i,j)_{encoutner} \cong P(j,i)_{encounter}$
        if $(|P(i,j)^{MR}_{encounter} - P(j,i)^{MR}_{encounter}| < th2)$ then
            // elapsed time after encountering
            $t_{diff} = Now - T[P(i,j)^{MR}_{encounter}]$;
            $P(j,i)_{estimate} = P(i,j)^{MR}_{encounter} * \gamma^{t_{diff}/bp}$;
            // compare estimated and real value
            if $(P(j,i)_{real} - P(j,i)_{estimate} > th3)$ then
                Report node $j$ as a suspicious node;
        else
            //minimum disconnection time between $i$ and $j$
            $t^{\min}_{diff} = \frac{dist(j,i) - R}{\bar{v}}$;
            // $P_{encounter\_history}$: the history valid value of $P(j,i)_{encounter}$
            $P(j,i)_{\max} = P_{encounter\_history} * \gamma^{t^{\min}_{diff}/bp}$;
            if $(P_{real}(j,i) > \alpha P(j,i)_{\max})$ then
                Report node $j$ as a suspicious node;
    else
        The ferry cache $P_{real}(j,i)$ and wait for next round of visiting node
```

Figure 2. Pseudo code for Intrusion Detection

If the ferry finds that a node *j* is declared suspicious *K* times (i.e. the ferry finds *P(j,i)* higher than *P(i,j)* for at least K different nodes), then, the ferry will include node *j* in the blacklist that it broadcasts periodically. The regular nodes update their own blacklists after hearing the warning message from the ferry. They will not choose any nodes in the blacklist as the next hop node.

For the two ferries scenario, we assume that each ferry transfer the delivery encounter table information it hears from other nodes to the nodes that can hear both ferries. That way, either ferry can have a full view of the past encounter tables for all nodes. For a dense network, one may want to set a maximum of how many nodes will store such information but since we are dealing with sparse ad-hoc networks, we let the ferry store such information on all nodes that can hear both ferries. Our simulations show that at most we are talking about two to three such storage nodes in a sparse network environment. We refer to such nodes as the transfer nodes. Strong authentication and encryption schemes between the two ferries will ensure that the worst that can happen is that the encrypted delivery predictabilities information stored in the transfer nodes are dropped when these nodes are compromised but the encrypted information cannot be revealed to the attacker who compromised these nodes.

IV. SIMULATION STUDY

A. Simulation Set Up

We have implemented both the Prophet routing scheme and our FBIDM scheme in NS-2. We simulated two network scenarios: namely (a) 40 nodes over 2000x2000 m^2 (Network Scenario 1), and (b) 40 nodes over 3000x3000 m^2 (Network Scenario 2). The nodes are distributed randomly over the specified geographical area. The transmission range is set to 250m. 10 CBR connections are used in our simulation experiments. The source and destination of the connections are chosen randomly. Unless otherwise stated, each source generates 1 packet every second with a packet size of 512 bytes. The buffer size at each node is set to 600 messages. The nodes move either according to the random waypoint model with a maximum speed of 5 m/s and a pause time of 10 seconds or according to Zebranet model [12]. Unless otherwise stated, the ferry speed is 20 m/sec. Data traffic is generated for the first 3000 seconds but simulation continues until 10,000 seconds.

The metrics used in our experiments are (a) data delivery ratio, (b) percentage of the malicious nodes that are detected, (c) the average detection time of all detected malicious nodes, and (d) the false positive rate which is the percentage of good nodes that are declared malicious.

B. Results and Discussion

1) Effectiveness of the FBIDM scheme

In our first set of experiments, we let the nodes move according to the RWP mobility model. We fix the message generation rate of each flow to 0.25 msg/sec and vary the number of malicious nodes. We measure the average time taken to detect the malicious nodes, the percentage of good nodes that are falsely identified as “bad” nodes (false positive rate), the percentage of malicious nodes that are identified at the end of the simulation period, the delivery ratio achieved without the attack, with the attacks but without the mitigation scheme, and with the mitigation scheme. We use both network scenarios.

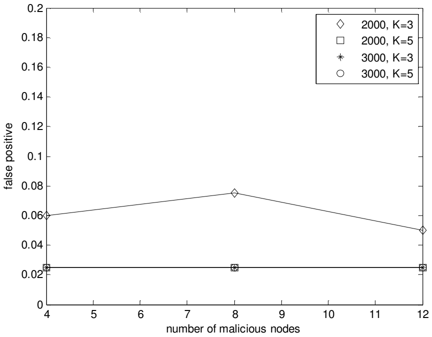


Figure 3: False positive rate versus Number of Malicious Nodes (RWP)

Figure 3 shows how the false positive rate varies with the threshold K for the two network scenarios. Figure 4 shows how the percentage of detected malicious nodes, and the detection time varies with different K values for the two network scenarios. From Figures 3 & 4, we see that for Network Scenario 1, setting K=5 allows us to achieve a false positive rate of less than 2.5%, and a detection rate of at least 80% for the malicious nodes. With K=5, the average detection time of the malicious nodes increases from 300 to 450 seconds as the number of malicious nodes increases from 4 to 12 malicious nodes for Network Scenario 1. With sparser network (Network Scenario 2), choosing K=3 allows us to achieve a false positive rate that is below 2.5% and a detection rate of more than 80% for the malicious nodes. Choosing K=5 for Network Scenario 2 (which is sparser) results in a lower percentage (65%) of malicious nodes being detected. Thus, we use K=3 for Network Scenario 2. The average detection time using K=3 for Network Scenario 2 ranges from 400 seconds (4 malicious nodes) to 700 seconds (12 malicious nodes).

Next, we plot the delivery ratio without attack, with attacks but without the mitigation scheme, and with attacks and with the mitigation schemes for the two network scenarios in Figures 5 & 6.

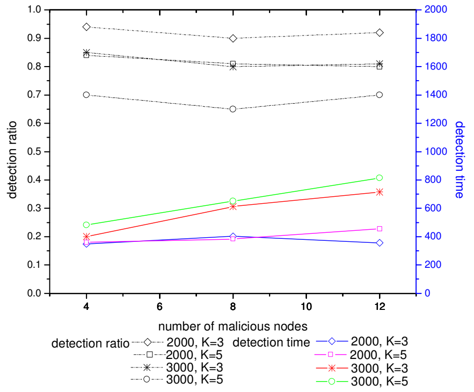


Figure 4: Detection ratio, and Detection time versus Number of Malicious Nodes (RWP).

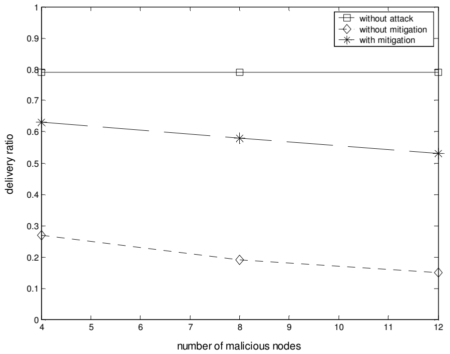


Figure 5: Delivery Ratio vs Number of Malicious Nodes (2000x2000 m^2 , K=5)

From Figures 5 & 6, we see that the mitigation scheme allows the system to recover some of the performance degradation caused by the attacks. For example, given 12 malicious nodes, the delivery ratio drops from 80% to 15% in Network Scenario 1 without the mitigation scheme but with the mitigation scheme, the delivery ratio improves to 52%. With Network Scenario 2 and 12 malicious nodes, the delivery ratio only improves to 32% with mitigation. Most of the lost packets occurred before the malicious nodes are detected

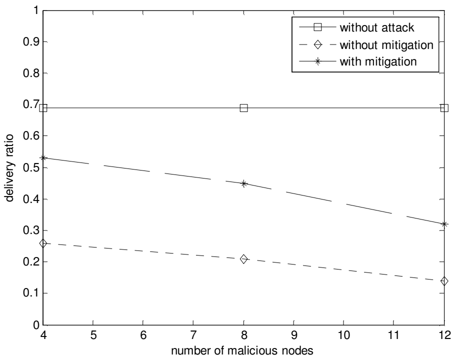


Figure 6: Delivery Ratio vs Number of Malicious Nodes (3000x3000 m^2, K=3)

2) *Impact of Mobility Models*

We repeat the experiment in Section IV.B.1 using the Zebranet mobility model. Our results using the Zebranet model are plotted in Figures 7 to 10.

Figure 7 indicates that with the same K value, the false positive rate is higher when nodes move according to the Zebranet model than with the RWP model. This is expected since the Zebranet model is more chaotic. However, the false positive rate can still be maintained below 5% with K=5 for Network Scenario 1 and below 2% with K=3 for Network Scenario 2. With these K values, the percentage of detected malicious nodes can be maintained above 85%. The average detection time (shown in Figure 8) can be kept around 375 to 750 seconds for Network Scenario 1 (K=5) and 600 to 750 seconds for Network Scenario 2 (K=2). The average detection time using Zebranet mobility model is higher than that using RWP mobility model because Zebranet movements are more chaotic.

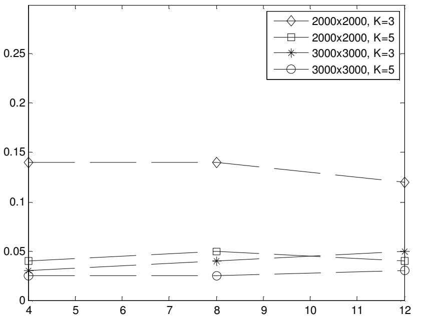


Figure 7: False Positive Rate vs Number of Malicious Nodes

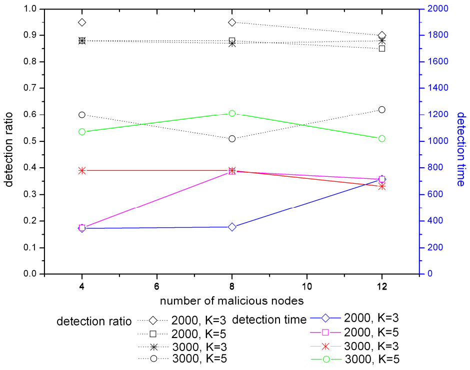


Figure 8: Percentage of Malicious Nodes Detected and Detection Time vs Number of Malicious Nodes.

From Figures 9 and 10, we see that the delivery ratio also improves with the mitigation scheme when the nodes move according to ZebraNet model. The improvement is not as good as in the RWP case. For Network Scenario 1, with 12 malicious nodes, the delivery ratio improves only to 40% in the Zebranet case as compared to 52% in the RWP case. For Network Scenario 2, it improves only to 25% for the Zebranet case as compared to 32% for the RWP case.

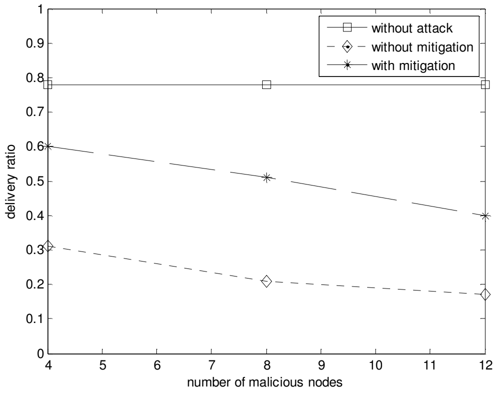


Figure 9: Delivery Ratio vs Number of Malicious Nodes (2000x2000 m^2 , K=5, ZebraNet)

3) *Impact of Traffic Load*

Next, we use the 40 nodes over 2000x2000 m^2 scenario, fix the number of malicious nodes to 8, and vary the traffic load to see if the increasing traffic load has any impact on the detection time, the percentage of detected malicious nodes, the false positive rate, and the delivery ratio under attack. Our simulation results show that the average detection time, the false positive rate, the percentage of detected malicious nodes are not sensitive to the traffic load.

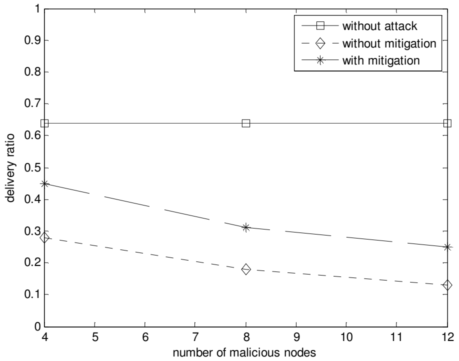


Figure 10: Delivery Ratio vs Number of Malicious Nodes (3000x3000 m^2 , K=3, ZebraNet)

Figures 11 & 12 show the delivery ratio without attack, with attacks but without mitigation, and with the mitigation scheme for the two mobility models. Since the delivery ratio drops with increasing traffic load, we see that the delivery ratio achieved with the mitigation scheme is closer to that without attacks with the RWP model. With the Zebranet model, the improvement in delivery ratio is not as good may be because there are fewer alternative next-hop nodes that can be considered with more chaotic Zebranet movements.

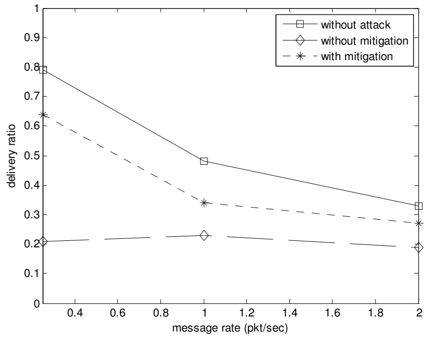


Figure 11: Delivery Ratio vs Message Rate (RWP)

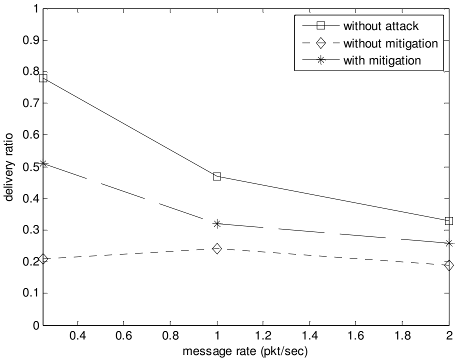
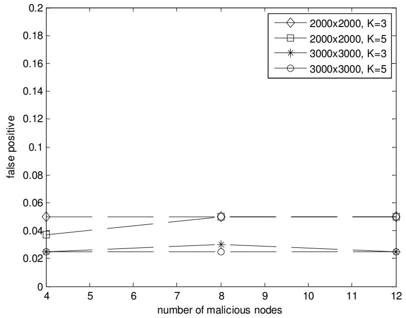


Figure 12: Delivery Ratio vs Message Rate (ZebraNet)

4) *Impact of Number of Ferries*

In our third set of experiments, we investigate the impact of the number of ferries on the intrusion detection and mitigation performance. We still use the same two network scenarios except that there are two ferries. The two ferries travel according to the routes shown in Figure 1(b). 10 CBR traffic flows with randomly selected source/destinations are used. Each flow generates 0.25 msg/sec. The nodes move according to the RWP model. Figures 13 to 14 show the false positive rate, the percentage of detected malicious nodes, and the average detection time. The false positive rate for Network Scenario 2 improves with two ferries. In addition, the percentage of detected malicious nodes improves and the average detection time decreases with two ferries. Figures 15 & 16 show the delivery ratios without attacks, with attacks but without mitigation and with mitigation for both network scenarios. Compared Figures 15 & 16 with Figures 5 &6, we see that the improvement in delivery ratios gets better with more ferries. For example with 8 malicious nodes, the delivery ratio for Network Scenario 2 only improves to 0.45 with a single ferry deployment but improves to 0.55 with two ferries deployment. Having two ferries allow the malicious nodes to be detected earlier and hence improves on the delivery ratio.



.

Figure 13: False Positive Rate vs No of Malicious Nodes (Two Ferries)

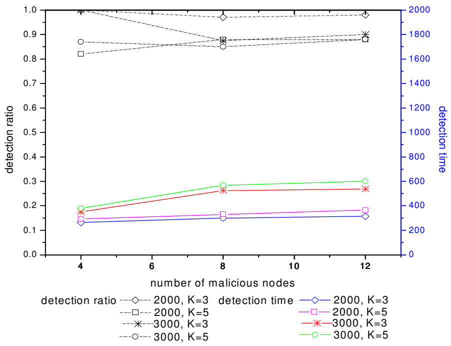


Figure 14: Average Detection Time and Percentage of Detected Malicious Nodes vs Number of Malicious Nodes (Two Ferries)

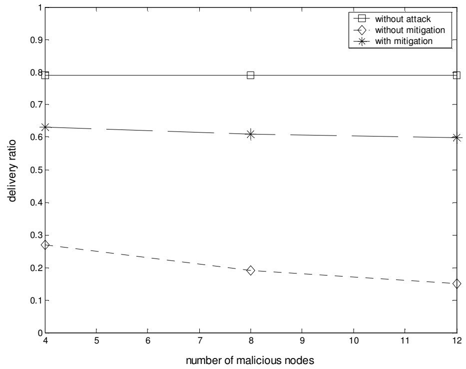


Figure 15: Delivery Ratio vs Number of Malicious Nodes for Network Scenario 1 (two ferries).

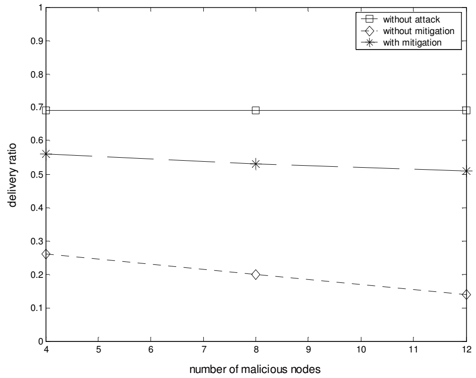


Figure 16: Delivery Ratio vs Number of Malicious Nodes (Network Scenario 2, K=3)

V. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we first describe how a DTN routing protocol Prophet works in sparse adhoc networks. Then, we describe how this routing protocol can be attacked and present a new ferry-based intrusion detection and mitigation (FBIDM) scheme for dealing with such attacks. Next, we present results of our simulation experiments that evaluate the usefulness of our FBIDM scheme. Our results show that our FBIDM scheme can mitigate effectively against the data dropping attacks in a sparsely connected adhoc network using Prophet as the routing scheme. The false positive rate can be kept within 2-5%, the percentage of detected malicious nodes can be higher than 80%, and the average detection time is within 300 to 450 seconds. We also evaluate the sensitivity of the FBIDM scheme with respect to different mobility models, traffic loads. The average detection time improves with more ferries. This is just a preliminary work. We intend to study the performance of the FBIDM scheme in new attack scenarios e.g. having malicious nodes move faster to increase its chances of being selected as relay nodes, wormhole attacks where attacking nodes collude.

ACKNOWLEDGMENT

. This work is sponsored by Defense Advanced Research Projects Agency (DARPA) under contract W15P7T-06-C-P430. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. This document is approved for public release, unlimited distribution.

REFERENCES

[1] C Perkins, etc, "Ad Hoc On-Demand Distance Vector (AODV) Routing", IETF internet draft, draft-ietif-manet-aodv-11.txt, 2002.

[2] D. B. Johnson, D. Maltz, "Dynamic Source Routing in Ad Hoc Networks", Mobile Computing, edited by T. Imielinski and H. Korth, Chapter 5, pp 153-181, Kluwer Academic Publishers, 1996.

[3] J. Hsu, et al, "Performance of Mobile Adhoc Networking Routing Protocols in Realistic Scenarios", Scalable Network Technologies White Paper, 2004.

[4] P. Papadimitratos, Z. Haas, "Secure routing protocol for mobile adhoc networks", SCS Communication Networks and Distributed System Modeling and Simulation Conference, Jan, 2002.

[5] Y.C. Hu etc, "Ariadne: A secure on-demand routing protocol for adhoc networks", 8th ACM International Conference on Mobicom, Sept, 2002.

[6] Y.C. Hu etc, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks", Proceedings of IEEE 4th Workshop on Mobile Computing Systems and Applications, June 2002.

[7] C. Perkins, P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers", Proceedings of ACM Sigcomm, 1994.

[8] A. Perrig etc, "Efficient and secure source authentication for multicast", Network and Distributed System Security (NDSS) Syposium, Feb 2001.

[9] Y.C. Hu etc, "Packet leashes: a defense against wormhole attacks in wireless adhoc networks", Proceedings of the IEEE Infocom, April 2003.

[10] S. Marti etc, "Mitigating routing misbehavior in mobile adhoc networks", 6th ACM International Conference in Mobile Computing and Networking, Aug 2000.

[11] B.Awerbuch etc, "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures", Proceedings of ACM WiSe, 2002.

[12] Y. Wang, S. Jain, M. Martonosi, K. Fall, "Erasure-Coding Based Routing for Opportunistic Networks", Proceedings of Sigcomm WDTN Workshop, Aug, 2005.

[13] Y. Huang, W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks", Proceedings of 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, 2003.

[14] Y. Hu, A. Perrig, D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols", Proceedings of ACM WiSe, 2003.

[15] M. Chuah, P. Yang, B. Davison, L. Cheng, "Store-and-Forward Performance in a DTN", VTC poster, Proceedings of IEEE VTC, 2006.

[16] M. Chuah, P. Yang, "Comparison of Two Intrusion Detection Schemes for Sparsely Connected Ad Hoc Networks", Proceedings of IEEE Milcom, Oct 2006.

[17] K. Fall, W. Hong, S. Madden, "Custody Transfer for Reliable Delivery in Delay Tolerant Networks", IRB-TR-33-030, July 2003.

[18] A. Lindgren, A. Doria, O. Schelen, "Probabilistic Routing in Intermittently Connected Networks", Mobile Computing and Communications Review,Vol 7(3),pp 19-20, 2003.

[19] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine, "MaxProp: Routing for vehicle-based disruption-tolerant networking", Proceedings of IEEE Infocom, April 2006

[20] A. Vahdat, D. Becker, "Epidemic Routing for partially connected adhoc networks", Technical Report CS-200006, Duke University, April, 2000.