A Novel Collaborative Approach for Sinkhole Detection in MANETs

Leovigildo Sánchez-Casado[1](✉), Gabriel Maciá-Fernández[1],
Pedro García-Teodoro[1], and Nils Aschenbruck[2]

[1] Department of Signal Theory, Telematics and Communications,
School of Computer Science and Telecommunications, CITIC-UGR,
University of Granada, C/Periodista Daniel Saucedo Aranda S/n,
18071 Granada, Spain
{sancale,gmacia,pgteodor}@ugr.es
[2] Distributed Systems Group, Institute of Computer Science,
University of Osnabrück, Albrechtstr. 28, 49076 Osnabrück, Germany
aschenbruck@uos.de

Abstract. This paper presents a novel approach intended to detect *sinkholes* in MANETs running AODV. The study focuses on the detection of the well-known sinkhole attack, devoted to attract most of the surrounding network traffic by providing fake routes, and thus, invalidating alternative legitimate routes and disrupting the normal network operation. Our detection approach relies on the existence of "contamination borders", formed by legitimate nodes under the influence of the sinkhole attack and, at the same time, neighbors of non-contaminated legitimate nodes. Thus, by collecting the routing information of the neighbors, these nodes are likely to be able to properly detect sinkholes. We evaluate our approach in a simulation framework and the experimental results show the promising nature of this approach in terms of detection capabilities.

Keywords: AODV · Intrusion detection systems · MANETs · Poisoning attacks · Sinkhole

1 Introduction

MANETs are a particular type of infrastructure-less networks composed of mobile devices communicating via a multi-hop strategy, *i.e.*, a given node can directly communicate with those within its communication range, but it makes use of other nodes to relay its messages to out-of-range destinations. These inherent characteristics make this kind of networks a particularly useful candidate in certain areas, such as military applications, disaster management, etc. [1]. As MANETs proliferate, specific security issues become more relevant and need to be appropriately addressed for these environments. Different factors, usually referred to the constrained nature of nodes (reduced bandwidth, battery lifetime, etc.), must be taken into account in the mentioned security related aspects.

Among others on the networking layer, *route poisoning* attacks [2] are among the most potentially disruptive threats in MANETs. The present work focuses on

M. Garcia Pineda et al. (Eds.): ADHOC-NOW Workshops 2014, LNCS 8629, pp. 123–136, 2015.
DOI: 10.1007/978-3-662-46338-3_11

the study of the *sinkhole* attack, possibly the most representative route poisoning attack. Nodes exhibiting this malicious behavior attempt to forge the source-destination routes in order to attract through them the surrounding network traffic. For this purpose, sinkhole nodes modify the control packets of the routing protocol and publish fake routing information that makes them appear as the best path to some destinations. In this manner, they achieve to be selected by other legitimate nodes as next hop on the forged route.

Focused on detecting sinkhole attacks, this work proposes an intrusion detection system (IDS) that relies on the existence of "contamination borders". These borders are formed by legitimate nodes under the influence of the sinkhole attack but with other neighbors which are not. We hyphotesize that by collecting and analyzing part of their own routing information and those belonging to their neighbors, these frontier nodes can precisely determine the existence of sinkhole behaviors. Based on this hypothesis, we suggest an IDS for the detection of sinkhole attacks which performs a collaborative process that collects from the neighbors the features for estimating the malicious behavior of a given node. The detection capabilities of our approach are enhanced regarding previous approaches due to the employment of information gathered from the contamination borders. These capabilities are proven in AODV (*Ad hoc On-Demand Distance Vector*) [3], one of the most representative and studied routing protocols in MANETs, obtaining promising results.

The rest of the paper is organized as follows. Section 2 describes the implementation of a sinkhole attack in AODV. Section 3 provides some related work regarding fighting against sinkhole attacks in MANETs. The existence of "contamination borders" and their utility as the basis for our detection approach is proven in Sect. 4. Our IDS is explained in Sect. 5, while Sect. 6 describes the experimental environment to evaluate the approach and the results obtained. Finally, main conclusions and future work are presented in Sect. 7.

2 Sinkhole Attacks in AODV

Among various routing protocols for MANETS, AODV is perhaps the most well-known and one of the most widely used ones. This is mainly due to its many useful characteristics. AODV is a reactive routing protocol for MANETs, *i.e.*, routes to a given destination are established on demand. If a source node N_s needs a connection with a destination node N_d and it does not have a valid route towards it, N_s initiates a route discovery process by broadcasting a *route request* message (RREQ). Upon receiving this RREQ, intermediate nodes forward it to their own neighbors, repeating the process until the RREQ reaches the intended destination. Once N_d receives the first RREQ, it sends a *route reply* message (RREP) backwards via the inverse route. Besides, AODV permits that intermediate nodes having a valid route to the destination generate RREP messages as a response to the received RREQ messages. Therefore, source and intermediate nodes are responsible for managing the routing information related to the next hop for every communication flow.

To avoid routing loops, AODV employs *destination sequence numbers*. These monotonically increasing numbers allow the nodes to determine the freshness of their information. Sequence numbers are updated whenever a node receives new (*i.e.*, not stale) information from control messages. This way, a node updates its routing information if the sequence number received in the RREP message is greater than the last stored sequence number. Given the choice between two routes, a node selects the one with the greatest sequence number. This fact can be exploited by malicious nodes to introduce themselves in the path.

Routing tables of the nodes in AODV are composed of the following fields: destination, next hop, distance to the destination measured in number of hops ($HopCount$), status (VAL -valid- or INV -invalid-) and sequence number ($SeqNum$), as well as other fields, like the lifetime of the route, several flags, the output interface, etc.

Once the very basics of AODV are known, it is easy to understand how a malicious node can carry out a sinkhole attack. It could modify or create a RREP message which announces an optimal metric, *i.e.*, a sequence number greater than the one received in the RREQ. If the sequence number is large enough, all other alternative routes will be invalidated. As a consequence, the malicious node guarantees that the requesting node will learn the route through the former, which will be selected as the next hop on the path. If the sinkhole node replies with fake RREP messages to every received RREQ packet, it will eventually become a sink of all data packets. Having achieved that, the malicious node will be able to apply different actions over the collected traffic, such as extracting sensitive information, modifying or discarding packets or carrying out more sophisticated attacks.

Figure 1 shows an example of a sinkhole attack. Here, the source node N_s broadcasts a RREQ message (1) asking for a route towards the destination N_d, this message being forwarded by the intermediate nodes. When the RREQ packet reaches the malicious node N_m, it replies with a fake RREP message (2a) claiming to have a shorter ($HopCount = 1$) and fresher ($SeqNum = 37$) route. At the same time, N_d is replying with a RREP message (2b) that includes the legitimate values for $HopCount$ and $SeqNum$ (3 and 8 respectively). Therefore, despite receiving other legitimate replies, N_s will choose the route through N_c, considered the most recent. Thus, the traffic from N_a towards N_d will eventually go through the malicious node N_m.

3 Related Work

Intrusion detection techniques have been recurrently used to determine the potential existence of non-legitimate events in a communication environment [4]. Consequently, in the literature a wide variety of IDS schemes was already proposed to detect sinkhole attacks in MANETs. Typically, they are classified as network-based IDS (NIDS) or host-based IDS (HIDS) dependending on the source of the features that support the detection process [4]. In what follows, we show that most of the IDS solutions adopted at present to detect sinkhole attacks are NIDS-like, that is, network parameters are monitored to determine the potential occurrence of malicious events.

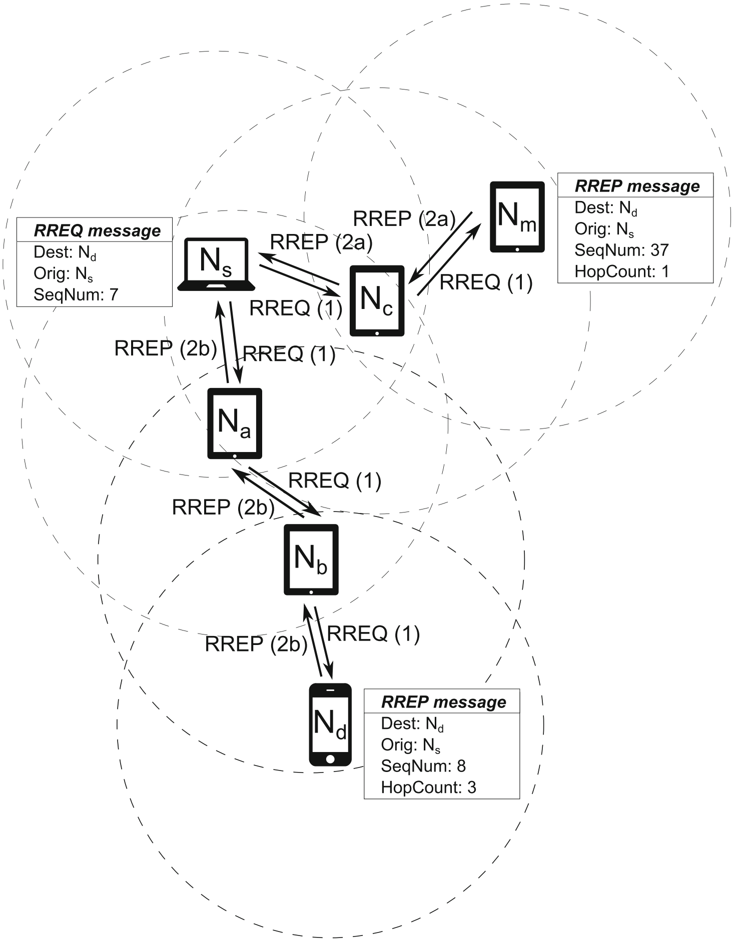


Fig. 1. Example of sinkhole node, N_m, replying with a fake RREP for a destination N_d.

Machine learning approaches are used in many approaches. Zhang *et al.*, in [5], introduce a local and cooperative scheme in which each mobile node runs a SVM-based IDS agent that monitors local traces, being responsible for locally detecting signs of intrusions. However, if an anomaly is detected among the local data, or if an evidence is inconclusive, neighboring IDS agents will collaboratively investigate, participating in a global detection procedure. A cross-feature method is described in [6], where a total of 141 traffic and topology related features are defined. This method also analyses correlations between features, in order to reduce the number of them. Then, a classifier like C4.5, RIPPER or Naïve-Bayes is used to carry out the anomaly detection procedure.

Other approaches perform some sort of matching techniques. For instance, IDAD [7] is an IDS solution to detect both single and multiple sinkholes.

This scheme compares every network activity of a host with a pre-collected set of anomaly and attack activities. The parameters used are obtained from each anomaly RREP packet: destination sequence number, hop count, route lifetime, destination IP address and timestamp. This way, IDAD is able to differentiate normal from abnormal RREP packets just by checking resemblances among them.

Finally, most of the techniques simply monitor the target environment, comparing the value of the collected features with a given threshold, which could be adaptive or not. Kurosawa *et al.* [8] introduce an anomaly detection scheme which uses a dynamic training method. They consider the number of RREQ packets sent and RREP packets received, as well as the average of the differences between the destination sequence numbers sent in RREQ packets and the ones received in RREP packets. Thus, this training set of features is employed to calculate the detection threshold based on the normal state of the network, which is dynamically adapted at regular time intervals to improve the detection accuracy. For the detection process, every sample in the data set is compared with the threshold to detect deviations from the normal network state. Similarly, in [9], the authors propose DPRAODV, in which the node receiving a RREP message checks whether the sequence number value exceeds a given threshold. To reduce inaccuracies which can lead to false alarms, this threshold value is dynamically updated at every time interval. If the sequence number is higher than the threshold, the intermediate node is suspected to be malicious.

Furthermore, a number of slight variations that also follow the approach of comparing the sequence number received in the RREP packet with the sequence number sent in the RREQ can be found in [10–13].

These schemes only consider the behavior of the sequence numbers in a local way, *i.e.*, without taking into account information of the network vicinity. By considering this behavior in a more global way, we will demonstrate that it is possible to improve the detection capabilities.

4 "Contamination Borders" in the Sinkhole Attack

Let us consider the existence of a MANET composed of L legitimate nodes $\{N_1, ..., N_L\}$. For every node N_i in the network, we extract some features following a time-based procedure, by using non-overlapping windows of δ seconds. As we assume mobility of the nodes, every node N_i has different sets of neighbors $NB_i(\omega)$ at the time of study $\omega \in \mathbb{N}$. Nodes can generate different traffic flows and they communicate by using AODV. We use the notation $R_{i,j}$ to refer to the route learned by node N_i towards a given destination N_j. Routes are composed, among other fields, by the following information: $R_{i,j}(\omega) = \{SN_{i,j}(\omega), NH_{i,j}(\omega)\}$, where $SN_{i,j}(\omega)$ is the sequence number learned for the route $N_i \rightarrow N_j$ and $NH_{i,j}(\omega)$ represents the next hop towards the destination at time $\omega{\cdot}\delta$.

In this general scenario, we additionally consider the existence of M malicious nodes behaving as sinkhole nodes, *i.e.*, nodes that reply to every RREQ packet with a forged RREP message, trying to include themselves as the next hop in the path to the destination.

4.1 Existence of "Contamination Borders"

In the above scenario, our approach relies on the existence of contamination zones, formed by legitimate nodes which are under the influence of the attack. Some of these nodes conform the "contamination border". The peculiarity of these last nodes is that they are simultaneously neighbors of contaminated nodes and nodes which are not under the influence of the sinkhole (*i.e.*, those that have the knowledge about the legitimate routes).

The nodes at the "contamination zone" forward traffic through the sinkhole node. At the same time, when a non-contaminated node requests to one of the contamination border nodes a route that has been compromised, it will reply with fake information, *i.e.*, the border node will unintentionally publish fake learned routes when asked for them. In such a situation, the border nodes behave in a similar way to how a malicious node would, being indistinguishable from actual sinkholes.

Let us illustrate this idea with the example shown in Fig. 2. Let us assume first that, at a given time t_0, node N_c has a legitimate route to a destination N_d with sequence number 35. At t_1, N_b needs a route towards N_d and generates a RREQ which is forwarded by N_a. As a consequence, at t_2, N_m replies with a fake RREP including an increased sequence number (for instance, 100) and N_c replies with its legitimate RREP. Since the sequence number in N_c is smaller, N_a learns the route through N_m and forwards that RREP to N_b. The routes are updated at t_3.

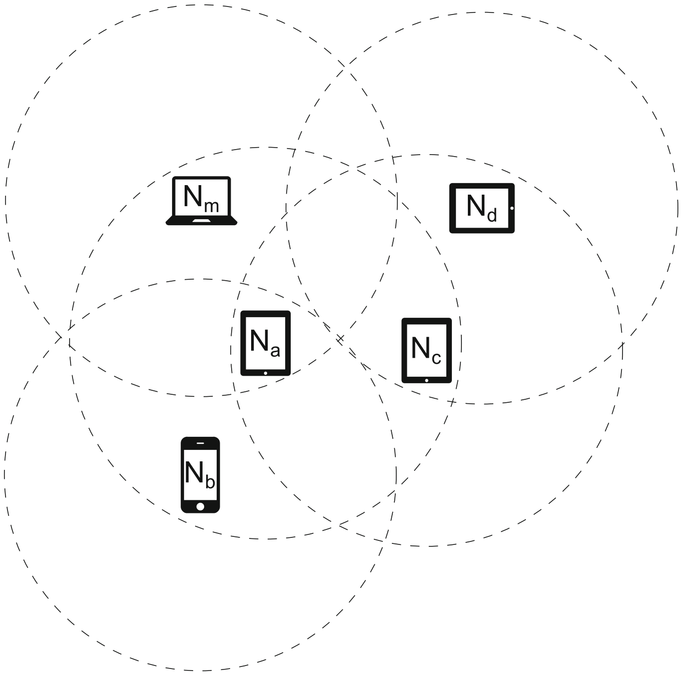
In such a situation, N_a will become a contamination border node, since it sends a fake RREP to N_b without a malicious intention. Thus, the contaminated area will be formed by N_a and N_b, N_m being the malicious sinkhole. Nodes N_c and N_d will remain without being contaminated.

Under these circumstances, the only difference between a sinkhole node and a contaminated node is that sinkhole nodes deliberately try to attract most of the surrounding traffic, whereas contaminated ones only act like the sinkhole for those requests related to fake routes learned from it, and not for every request they receive.

4.2 Use of "Contamination Borders" to Detect Sinkhole Behaviors

As shown in Sect. 3, most of the IDS schemes consider information directly extracted from the node carrying out the detection process, *i.e.*, they basically employ some metric related to the difference between sent and received sequence numbers. However, this approach suffers from some flaws which can lead to errors in the detection process.

The first weakness is related to the fact that these approaches provide good results as long as the increased sequence numbers published by the sinkholes are high. That is, the difference between the sent and received sequence numbers is noticeable. However, if the sinkhole node is somehow smart, it will publish fake sequence numbers moderately high, thus assuring that it is selected as the next hop whereas hindering the detection process. On the other hand, legitimate nodes learning fake routes are able to publish them and, as seen, they are



	N_a				N_b				N_c			
T	***Dst***	***NH***	***SN***	***St***	***Dst***	***NH***	***SN***	***St***	***Dst***	***NH***	***SN***	***St***
t_0	-	-	-	-	-	-	-	-	N_d	N_d	35	VAL
t_1	Forwards RREQ				Generates RREQ				N_d	N_d	35	VAL
t_2	Receives RREP				Receives RREP				N_d	N_d	35	VAL
t_3	N_d	N_m	100	VAL	N_d	N_a	100	VAL	N_d	N_d	35	VAL

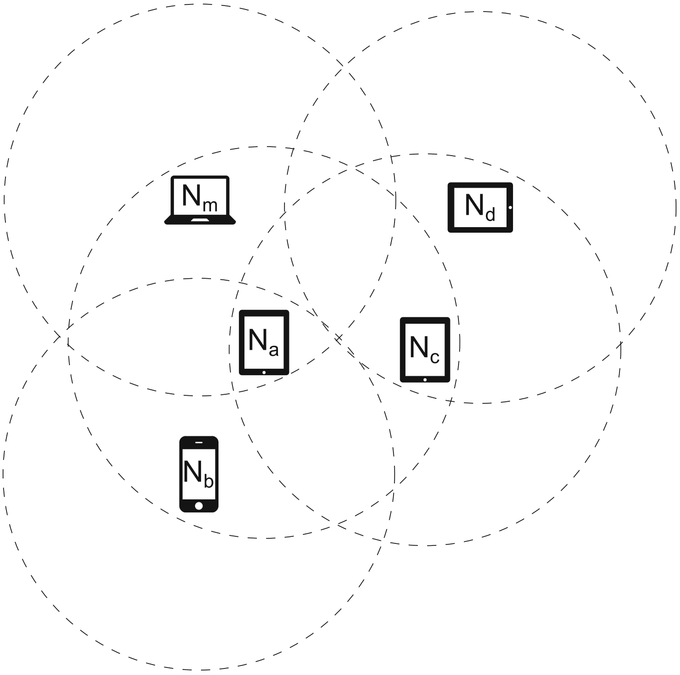
Fig. 2. Existence of contamination zones and border nodes.

prone to be erroneously detected as sinkhole as well. Therefore, both facts can degrade the detection capabilities of these schemes.

Our approach is based on the fact that, due to the existence of "contamination borders", if a border node compares the received sequence number for a given route not only with the sequence number sent, but also with the sequence numbers stored by their neighbors, the dynamic range of the detection will be increased, thus leading to a better performance of the IDS scheme. Therefore, by collaborating with their neighbors and sharing the features of interest, these border nodes are able to perform a better detection, properly distinguishing between sinkhole nodes and legitimate nodes.

Besides, in a network with sinkhole nodes, it is expected that contaminated nodes being neighbors of a sinkhole node have in their routing tables many entries whose next hop is the given sinkhole, and not that many whose next hop is other contaminated node. For this reason, we hyphotesize that those nodes which appear most often in the routing tables of the other nodes are more likely to be considered malicious. Thus, in our detection system, we will incorporate this information and combine it with the monitoring of suspicious evolutions in the value of the sequence number.

The simple example depicted in Fig. 3 shows the differences between the two approaches. At time t_0, nodes N_a and N_b have a fake route to a destination N_d with sequence number 100, since this route have been falsified before (example in Fig. 2). Besides, node N_c knows the legitimate route to N_d, with sequence number equals to 20. At time t_1, the route towards N_d in N_b becomes stale and it marks the route as invalid (INV) and increases the sequence number in one unit (101). At t_2, N_b needs again a route towards N_d and generates a RREQ which is forwarded by N_a. At t_3, N_m replies with a fake RREP that includes an increased sequence number (for instance, 121). However, as the sequence number for the required route in node N_c is smaller than the one included in the RREQ,



	N_a				N_b				N_c			
T	*Dst*	*NH*	*SN*	*St*	*Dst*	*NH*	*SN*	*St*	*Dst*	*NH*	*SN*	*St*
t_0	N_d	N_m	100	VAL	N_d	N_a	100	VAL	N_d	N_d	35	VAL
t_1	N_d	N_m	100	VAL	N_d	N_a	101	INV	N_d	N_d	35	VAL
t_2	Forwards RREQ				Generates RREQ				N_d	N_d	35	VAL
t_3	Receives RREP				Receives RREP				N_d	N_d	35	VAL
t_4	N_d	N_m	121	VAL	N_d	N_a	121	VAL	N_d	N_d	35	VAL

Fig. 3. Utility of a "contamination border" node, N_a, in the detection process.

N_c does not reply. The detection schemes compute the values at t_4, when routes have been updated.

In previous approaches, like [11] or [13], N_a would obtain the difference between the sent and received sequence numbers, resulting in $121 - 101 = 20$ units, which can be enough to attract the route but not to be detected by N_a. By using our approach, N_a computes the difference by comparing the sequence number received in the RREP with the minimum sequence number for the required route in its neighbors, giving as a result a difference of $121 - 35 = 86$ units.

As it has been explained by the static and straightforward scenario depicted in the example, by gathering very little information from the neighbors (basically the sequence number for some required routes), border nodes are able to increase the dynamic range of the metric usually employed to detect sinkhole attacks. This allows our approach to raise the detection threshold and therefore, to improve the detection rate whereas the misclassification rate remains low.

5 Deploying the Sinkhole Detection Scheme

This section presents the specific implementation of the proposed network-based intrusion detection system, which employs a simple heuristic to obtain an indicator value for the detection of sinkhole attacks. The IDS computes the heuristic by collecting information related to the routing tables of the node running the IDS and its neighbors. Even though the detection process is locally performed by each node running the IDS, the features involved in such a process are collaboratively gathered from the node itself and its neighbors.

This heuristic relies on the hypothesis that there are border nodes being under the influence of the sinkholes that have neighbors which are not under the influence and know the legitimate routes. The sequence number information provided by the neighbors allows to improve the detection capabilities in these border nodes. Besides, those nodes appearing most often in the routing tables as next hop are more likely to be considered malicious, since sinkhole nodes attract most of the surrounding traffic, and this fact must be taken into account in the heuristic.

It must also be noted that only those nodes that are neighbors of the actual sinkhole will be able to detect it, since non-neighbor nodes will detect as malicious those frontier nodes unintentionally sharing fake routes.

5.1 Overview of the Detection Approach

Our approach follows a window-based procedure to detect malicious nodes discretely over time. Every node N_i will run the IDS, and will check during each window if any of its neighbors is malicious or not. Thus, for every next hop node (NH), the following features are collected:

- $\boldsymbol{D_{i,NH}(\omega)}$**:** the set of all destinations for the routes in the routing table of N_i which use NH as next hop, at time $\omega \cdot \delta$. Only valid routes with $HopCount$ greater than 1 are taken into account, since routes with $HopCount = 1$ indicate neighbors and do not have to be published, so they will not indicate whether or not a node is publishing false routes.

- $\boldsymbol{SN_{i,j}(\omega)}$**:** sequence number at node N_i for every destination N_j, at time $\omega{\cdot}\delta$.
- $\boldsymbol{NB_i(\omega)}$**:** set of neighbors of node N_i, at time $\omega{\cdot}\delta$.

Taking the above into account, we apply a heuristic to obtain an indicator value about the node's behavior as sinkhole. For that, the following procedure is executed:

(1) The IDS at node N_i obtains, for each NH in its routing table, a set of destinations N_j in $D_{i,NH}(\omega)$.

(2) Then, it requests to its neighbors theirs sequence numbers for those destinations N_j.

(3) After gathering the information from all the neighbors, N_i obtains the minimum sequence number of their neighbors for each destination N_j, and computes the difference between their own sequence numbers and these minimums.

(4) Finally, the malicious value for the NH is obtained as the summatory of these differences, thus considering that nodes NH with more poissoned routes are more likely to be a sinkhole node than a contaminated node:

$$MV_{i,NH}(\omega) \ = \sum_{j \in D_{i,NH}(\omega)} \left(SN_{i,j}(\omega) - \min_{v \in NB_i(\omega)} SN_{v,j}(\omega) + 1 \right) \qquad (1)$$

Since, for a given destination, the computed difference between sequence numbers can be zero, we add 1 unit, thus taking every possible compromised destination into account in the summatory.

(5) After the calculation of the $MV_{i,NH}$, if it exceeds a given threshold, θ, the node NH is classified as a malicious sinkhole node:

$$class(NH) = \begin{cases} malicious, & \text{if } MV_{i,NH}(\omega) \geq \theta \\ legitimate, & \text{otherwise} \end{cases} \qquad (2)$$

It can be observed that the calculation of the malicious value is a simple process with low computational cost once all the neighbors' information have been gathered.

(6) After the classification of NH as sinkhole, N_i could apply some response mechanism, like that of including NH in a blacklist or notifying all the nodes in the network about the malicious behavior of NH. These and other possible reaction schemes are out of the scope of this detection-oriented contribution.

6 Experimental Results

This section presents the description of the experimental environment used to evaluate the proposed scheme. Besides, some tests have been performed to prove the proper performance of the IDS, the experimental results being discussed.

6.1 Experimental Environment

In this work we have simulated some MANET deployments by using the network simulator OMNeT++ [14]. To simulate the sinkhole nodes, we have used NETA [15], a framework built on top of OMNeT++ that allows to simulate different network attacks in a simple manner and permit to apply several configuration parameters over them.

The simulation area is restricted to a 1000 m × 1000 m square, with each node having a communication range of 250 m. As MAC and network layer protocols we have chosen 802.11-g and AODV. The simulation time is set to 300 s and the duration of the temporal window ω used for collecting the features is 1 s.

The total number of nodes is 25, with 24 legitimate nodes and only 1 sinkhole node. The attack is performed during the whole simulation by replying with false RREP every received RREQ, even if the sinkhole does not know a valid route. A value following a uniform distribution between 20 and 30 units is added to the one observed from the RREQ, giving the false increased sequence number. It must be noted that, in the literature, most of the works simply set the false sequence number to the maximum possible ($2^{31} - 1 = 4294967295$), meanwhile other works adds relatively high values, for instance, uniform values between 15 and 200 units. We consider a more realistic sinkhole which tries to hinder the detection process but assures being selected as the next hop.

To model the movement of the nodes the popular Random Waypoint Model (RWP) [16] has been chosen. In this model the node selects random destination and speed. When the node reaches the destination, it waits for a pause time before choosing a new random destination and speed and repeats the process. The minimum speed is fixed to 0.5 m/s and the maximum speed varies between 3 to 10 m/s, being the pause time set to 15 s. These maximum speeds (3–10 m/s $\equiv$ 10.8 –36 km/h) cover the range from pedestrian walk to a moderate vehicle speed.

The number of traffic flows is fixed to 24, each one simulating point to point voice traffic. Several calls per flow are obtained by modelling the pause time between calls (*inter arrival time* or IAT) with a exponential distribution with $\lambda = 7.5$ s and the duration of the call (*call holding time* or CHT), modelled as a lognormal with mean, μ, set to 2.5 and standard deviation, σ, set to 0.5 [17]. For each call, one of the legitimate nodes is randomly chosen as destination, being the traffic a Constant Bit Rate (CBR) connection, with 4 packets/second and payload size equal to 512 bytes.

6.2 Detection Results

We now evaluate the global effectiveness of the proposed IDS by means of several test based on simulations. The effectiveness is evaluated by computing two metrics, namely the true positives rate (TPR) and the false positives rate (FPR).

We study the detection efficiency for different mobility conditions, obtaining various operation points to conform the ROC (Relative Operation Characteristic) space by varying the decision threshold θ in (2). It is important to note that the ROC curve is derived by repeating 20 times (with different seeds) every simulation.

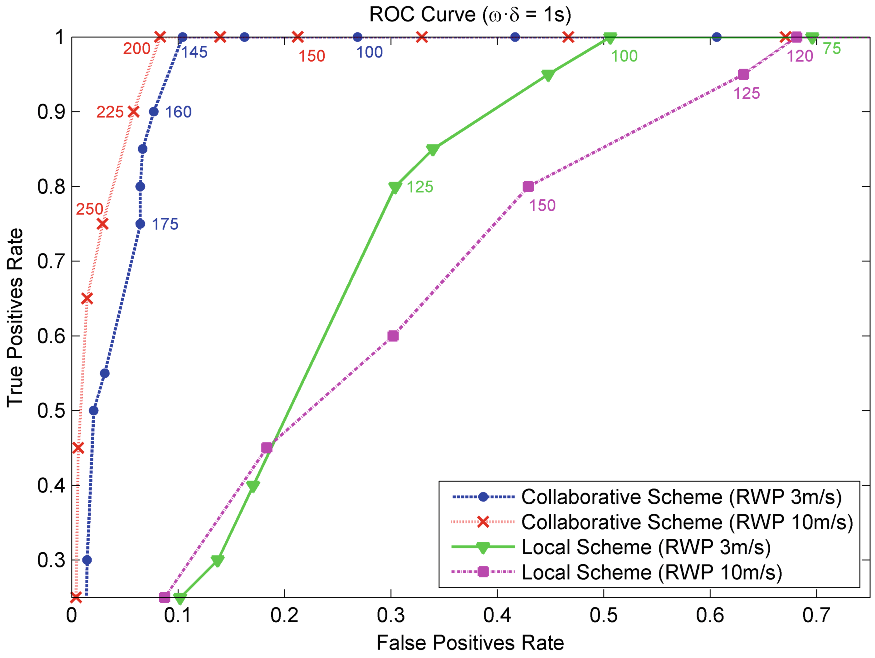


Fig. 4. ROC curve for sinkhole detection, for different values of the decision threshold θ.

Figure 4 depicts the ROC curves obtained by using our collaborative approach and those obtained by using an approach that compute a local heuristic only considering the sent and received sequence numbers in the node, as those introduced in [11] or [13]. The curves are obtained under two mobility conditions, by varying the maximum speed of the nodes between 3 m/s and 10 m/s. As it can be seen, by including information from the neighbors, our scheme overcomes the results achieved by the local approach used by some previous schemes.

Besides, it is shown that if the detection threshold is set to a high value, the system is expected to improve FPR, but to achieve worse TPR. On the other hand, the lower the threshold, the better the TPR value, at the expense of increasing the FPR. Thus, the optimal operation point of our system can be achieved empirically, and it depends on the mobility conditions.

As shown, the proposed IDS can achieve excellent results regarding the two metrics considered, TPR and FPR. By selecting the optimal operation point, TPR can achieve 100 % keeping FPR always below 10 %. These results confirm the capabilities of our model.

7 Conclusions and Future Work

In this paper we introduce a new methodology for the detection of sinkhole attacks in MANETs which relies on the existence of contamination zones and border nodes. The scheme is based on a simple heuristic that computes the differences between the sequence numbers on these frontier nodes and those belonging to their neighbors. This heuristic allows to estimate the malicious behavior of the nodes acting as sinkholes.

The use of a simple heuristic overcomes the computational overhead present in more sophisticated approaches based on data mining algorithms. We have confirmed by means of simulation the good performance of our system, where different scenarios have been analyzed. The results obtained clearly highlight the goodness of our IDS approach, which can experience 100 % overall TPR with less than 10 % potential FPR.

As shown, the experimental results obtained are very encouraging. This way, we are going for such direction through the improvement of some aspects of our approach in the near future:

- In distributed IDS for MANETs is highly recommended to reduce the information exchanged and shared. We are working on the development of a communication protocol that takes into account the limited bandwidth resulting from the MANET context, thus involving the lowest possible overhead.
- This way, we are also developing a pre-filtering phase in order to also reduce the overhead introduced by our approach.
- We are planning to extend our approach to include trust-based schemes as response mechanism to face collusions situations carried out to evade the detection process or to accuse legitimate nodes.
- Finally, the inclusion of more realistic mobility models in the experimentation is also of interest.

Acknowledgment. This work has been partially supported by Spanish MICINN through project TEC2011-22579 and by Spanish MECD through the grant “University Professor Training Program” (FPU, Ref.: AP2009-2926).

References

1. Lakhtaria, K.I. (ed.): Technological Advancements and Applications in Mobile Ad-Hoc Networks: Research Trends. IGI Global, Hershey (2012)
2. García-Teodoro, P., Sánchez-Casado, L., Maciá-Fernández, G.: Taxonomy and Holistic Detection of Security Attacks in MANETs, pp. 1–12. CRC Press, April 2014. http://www.crcpress.com/product/isbn/9781466578036
3. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. IETF, RFC 3561, July 2003
4. García-Teodoro, P., Díaz-Verdejo, J.E., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**(1–2), 18–28 (2009)
5. Zhang, Y., Lee, W., Huang, Y.A.: Intrusion detection techniques for mobile wireless networks. Wirel. Netw. **9**(5), 545–556 (2003)
6. Huang, Y., Fan, W., Lee, W., Yu, P.S.: Cross-feature analysis for detecting Ad-Hoc routing anomalies. In: Proceedings of 23rd IEEE International Conference on Distributed Computing Systems (ICDCS), pp. 478–487, May 2003

7. Alem, Y.F., Xuan, Z.C.: Preventing black hole attack in mobile Ad-Hoc networks using anomaly detection. In: Proceedings of 2nd International Conference on Future Computer and Communication (ICFCC), vol. 3, pp. 672–676, May 2010
8. Kurosawa, S., Nakayama, H., Kato, N., Jamalipour, A., Nemoto, Y.: Detecting blackhole attack on AODV-based mobile Ad Hoc networks by dynamic learning method. Int. J. Netw. Secur. **5**(3), 338–346 (2007)
9. Raj, P.N., Swadas, P.B.: DPRAODV: a dynamic learning system against blackhole attack in AODV based MANET. Int. J. Comput. Sci. Issues **2**, 54–59 (2009)
10. Al-Shurman, M., Yoo, S.M., Park, S.: Black hole attack in mobile Ad Hoc networks. In: Proceedings of 42nd Annual Southeast Regional Conference (ACM-SE), pp. 96–97, April 2004
11. Mistry, N., Jinwala, D.C., Zaveri, M.: Improving AODV protocol against blackhole attacks. In: Proceedings of International MultiConference of Engineers and Computer Scientists (IMECS), pp. 96–97, March 2010
12. Mandhata, S.C., Patro, S.N.: A counter measure to black hole attack on AODV-based mobile Ad-Hoc networks. Int. J. Comput. Commun. Technol. (IJCCT) **2**(VI), 37–42 (2011)
13. Himral, L., Vig, V., Chand, N.: Preventing AODV routing protocol from black hole attack. Int. J. Eng. Sci. Technol. (IJEST) **3**(5), 3927–3932 (2011)
14. Varga, A.: OMNeT++ Discrete Event Simulation System. http://www.omnetpp.org/doc/omnetpp/manual/usman.html. Accessed 14 March 2014
15. Sánchez-Casado, L., Rodríguez-Gómez, R.A., Magán-Carrión, R., Maciá-Fernández, G.: NETA: evaluating the effects of NETwork attacks. MANETs as a case study. In: Awad, A.I., Hassanien, A.E., Baba, K. (eds.) SecNet 2013. CCIS, vol. 381, pp. 1–10. Springer, Heidelberg (2013)
16. Johnson, D., Maltz, D.: Dynamic source routing in Ad Hoc wireless networks. In: Imielinski, T., Korth, H. (eds.) Mobile Computing. The Kluwer International Series in Engineering and Computer Science, vol. 353, pp. 153–181. Springer US, New York (1996)
17. Barceló, F., Jordán, J.: Channel holding time distribution in cellular telephony. Electron. Lett. **34**, 146–147 (1998)