A Generalized Intrusion Detection & Prevention Mechanism for Securing MANETs

Adnan Nadeem
Centre for Communication Systems Research
University of Surrey, United Kingdom
a.nadeem@surrey.ac.uk

Michael Howarth
Centre for Communication Systems Research
University of Surrey, United Kingdom
m.howarth@surrey.ac.uk

***Abstract* —Most of the research in securing Mobile ad hoc networks (MANETs) has focused on proposals which detect and prevent a specific kind of attack such as sleep deprivation, black hole, grey hole and rushing attacks. In this paper we broaden our previously develop algorithm AIDP and propose a generalized intrusion detection and prevention mechanism. We use a combination of anomaly-based and knowledge-based intrusion detection. This approach not only secures the MANET from a wide variety of routing attacks but also has the capability to detect new unforeseen attacks. Simulation results of a case study shows that our proposed mechanism can successfully detect attacks, including multiple simultaneous different attacks, and identify and isolate the intruders causing a variety of attacks, with an affordable network overhead.**

Keywords— ad hoc network security; intrusion detection & prevention; secure routing

I. INTRODUCTION

Mobile Ad hoc networks (MANETs) routing protocols, such as AODV and DSR, operate on the assumption that there is no malicious intruder node in the network. Malicious nodes can cause severe disruption without violating the routing protocol through a wide variety of attacks.

Intrusion detection and prevention (IDP) provides a way to protect nodes against routing attacks. There are two ID techniques: knowledge-based intrusion detection (KBID) and anomaly-based intrusion detection (ABID). KBID has a potentially low false detection rate but it can only detect attacks whose signatures are in the database. On the other hand ABID not only provides early warnings of potential intrusions but also can detect attempts to exploit new and unforeseen vulnerabilities; however it is more prone to generate false positives than KBID.

In our previous work [1] we proposed Adaptive Intrusion Detection and Prevention (AIDP), which used ABID to detect denial of service (DoS) attacks. In this paper we extend AIDP to a Generalised Intrusion Detection & Prevention (GIDP) mechanism. We propose a combination of anomaly-based and knowledge-based ID that takes advantage of both techniques. GIDP not only guards MANETs against a wide variety of attacks but also has the capability to detect new attacks or intrusive activities that degrade network performance; to the best of our knowledge this is novel.

The reminder of this paper is organized as follows. Section II describes the related research in securing MANETs. Section III reviews typical MANETs routing attacks. Section IV presents our proposed mechanism, GIDP. Section V illustrates the implementation of our proposed mechanism through a case study, including simulation. Finally, we summarize our results and future work in Section VI.

II. RELATED RESEARCH

Research in securing MANETs has to date mostly focused on detecting and preventing specific attacks. For example TOGBAD was proposed in [2] to identify nodes that attempt to create black hole attacks in MANETs that use the OLSR routing protocol. Kurosawa and Jamalipour [3] also propose a black hole detection mechanism, this time for AODV. Xiaopeng and Wei [4] proposed a grey hole attack detection scheme for the DSR routing protocol. Ping and Zhang [5] considered a route request (RREQ) flooding attack in MANETs. They proposed a RREQ flooding prevention mechanism based on neighbour's supervision. In [6] Perrig and Johnson analyzed how an attacker can launch a rushing attack (RU) in DSR and proposed a rushing attack prevention mechanism for MANETs.

Though most researchers have concentrated on protecting MANETs against specific types of attack, some have suggested a more general approach. For example ARAN [7] is a hop-to-hop authenticated routing mechanism that can protect MANETs against a number of attacks from external malicious nodes. A similar approach, Ariadne [8] has been proposed for end-to-end authentication based on shared key pairs. We believe more effort is needed on mechanisms which can guard MANETs against a wide variety of attacks.

Methods proposed in [7] & [8] protect MANETs mainly against external attacker through authenticated routing. However an insider trusted node can change

its behaviour and initiate activities that results in severe attacks as we describe below in section III.

III. AODV ROUTING ATTACKS

The on-demand routing protocols in MANETs, such as AODV and DSR, allow intruders to launch a wider variety of attacks. In order to illustrate these routing attacks we consider AODV as an example in this paper. Using AODV we now give examples of how different intrusive activities can cause various attacks in MANETs.

a) Sleep Deprivation through malicious RREQ flooding:

Sleep deprivation (SD) [9] is a denial of service attack in which an attacker interacts with the node in a manner that appears to be legitimate; but where the purpose of interaction is to keep the victim node out of its power conserving sleep mode. An intruder can cause SD of a node by exploiting the vulnerability of the route discovery process of protocol through malicious route request (RREQ) flooding in the following ways:

Malicious RREQ Flooding 1: an intruder broadcasts a RREQ with a destination IP address that is within the network address range but which does not exist. This will compel all nodes to forward this RREQ because no-one will have the route for this destination IP address.

Malicious RREQ Flooding 2: after broadcasting a RREQ an intruder does not wait for the *ring traversal time* and continues resending the RREQ for same destination with higher TTL values.

b) Black & Grey Hole by false RREP& packet dropping:

In AODV, the destination sequence number (*dest_seq*) is used to describe the freshness of the route. A higher value of *dest_seq* means a fresher route. On receiving a RREQ an intruder can advertise itself as having the fresh route by sending a Route Reply (RREP) packet with a new *dest_seq* number larger than the current *dest_seq* number. In this way the intruder becomes part of the route to that destination. The intruder can then choose to drop all packets, causing a black hole (BH) [3] in the network. The severity of the attack depends on the number of routes in the network the intruder successfully becomes part of; we analyze this further in section V.

Grey Hole (GH) is a special case of BH attack, in which intruder only drops packets selectively, e.g. from specific nodes.

c) Rushing attack through a forged RREQ:

In order to limit the control packet overhead an on-demand protocol only requires nodes to forward the first RREQ that arrives for each route discovery. An attacker can exploit this property by spreading RREQ packets quickly throughout the network so as to suppress any later legitimate RREQ packets. An intruder can forward the forged rushed RREQ, giving them a higher source sequence (*src_seq*) number and minimum delay. This will suppress the later legitimate RREQ and increase the probability that routes that include the intruder will be discovered rather than other valid routes, causing a rushing attack.

IV. GENERALIZED INTRUSION DETECTION & PREVENTION

A. Assumptions

We use ABID to detect intrusion in the network; this requires traffic traces that contain only normal activities to build a training profile. However, in contrast with fixed networks, data resources such as [10] that reflect normal activities or events are not currently available for MANETs. Therefore we assume that the initial behaviour of the network formed on-the-fly is free from anomalies. We also disregard attacks aimed at physical and link layer. Further, we have not considered attacks from colluding intruders in this paper. To illustrate the implementation of GIDP we assume a clustered MANET organization. We select the most capable nodes in terms of their processing abilities as cluster heads (CHs) and the others nodes becomes cluster nodes (CNs). At present we assume secure communication between CH and CNs. Most of these assumptions will be relaxed in our future work.

B. GIDP Architecture & Terminology

We now describe our proposed mechanism GIDP. GIDP is a hybrid IDP approach that uses a combination of anomaly-based and knowledge-based ID. The architecture of GIDP is shown in Fig.1. A cluster head first gathers data in the form of two matrices: network characteristic matrix (NCM) and a derived matrix (DM). The NCM contains data related to the network routing protocol; for example in the case study in this paper, NCM consists of seven parameters:

NCM={RREQ (route request), RREP (route reply), RERR (routeerror), TTL (time to live) values, RREQ src_seq, RREQ dest_seq, RREP dest_seq}

The DM consists of parameters which reflects the network performance and can be derived from NCM parameters. For example in the case study in this paper DM consists of three parameters:

DM= {CPO (control packet overhead), PDR (data packet delivery ratio), CPD (number of control packet dropped)}

Then the cluster head employs two phases: training and testing. Fig.2 shows the time-based operation of GIDP. When the network is established, the CH continuously gathers NCM and DM information and applies the GIDP training module for *N* time intervals (TI), resulting in initial training profiles (ITPs) of NCM

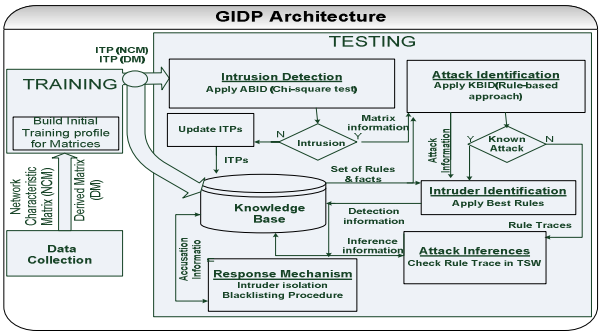

Figure 1. Architecture of GIDP.

and DM. The ITPs reflects the normal behaviour of the nodes in the network and the expected network performance. In the testing phase the CH applies the testing module after each TI. The testing phase consists of several tasks as shown in fig.1. Firstly it detects intrusion in the network. If there is no intrusion then it updates the ITPs in order to adapt the variation in the network behaviour as time progresses. If there is intrusion, in the second task the CH identifies the attack or attacks using existing information in the knowledge base. In the case of known attacks the CH identifies intruding nodes using intruder identification rules specific to the known attack. To optimise the probability of identifying intruders correctly with a low level of false positives, it maintains a test sliding window (TSW) as shown in fig.2, in which *d* detections of a node are required in *p* time intervals (TI). If this detection threshold is passed then the CH will blacklist the node and isolate the node by informing all CNs.

If attack identification detects an attack that does not match the rules for known attack then CH applies the attack inferences. Attack inference stores the rule trace of current TI as Detected Rule Trace and looks for its match in a TSW. If the match is found in a TSW then CH confirms the new attack by constructing & adding a rule for the new attack in the set of rules stored in knowledge base.

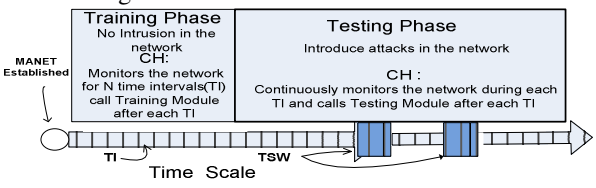

Figure 2. Time-based operation of GIDP.

C. Algorithm & Technical Details

We now explain the training & testing module of GIDP.

Training:

NCM consists of X_i parameters mentioned above, where i=1 to 7 and each $X_i = \{X_1, X2, X3, \ldots, X_M\}$is a set of random variables from 1 to M, where M is the maximum number of random variables of parameter X_i. For example *NCM [X_i]* represent the number of RREQ received by all CNs in *jth* time interval (TI), where M is the maximum number of RREQ received in a TI. The probability distribution of *NCM[X_i]* is calculated for the TI. CH then calculates the DM parameters CPO (i.e. Number of control packet / data packet delivered), PDR (i.e. Number of data packet received / data packet originated) & CPD (i.e. Number of control packet dropped in establishing & maintaining routes in the network) for the *jth* TI, and this whole process is then repeated for the *N* time intervals in the training phase. We then calculate mean $\overline{X}_i$ of $P(NCM[X_i])$ and means of CPO, PDR and CPD for *N* intervals, which is then stored as an ITP (NCM) and ITP (DM) respectively containing the expected values for that particular network observed for the total time of $N*TI$ seconds.

Testing:

In the testing phase GIDP operates in three stages: a) intrusion detection, b) attack identification and inferences and c) identification and isolation of intruding nodes (Fig.1). Now we explain the algorithms of stage a, b & c. For stage a) it employs ABID using chi-square goodness of fit test on NCM and then KBID using a rule-based approach on both matrices NCM & DM is applied in stage b) and c).

Testing Modules

This module only takes NCM parameters into account and applies chi-square test to identify any intrusion in the network.

a) Intrusion Detection

. Do after each TI
. collects NCM(X_i) from all other CNs in TI, for $\forall i$
. Calculate the probability distribution P (NCM(X_i))
.Calculate averages of P(NCM(X_i)) & stores as observed values
. End do
.For $\forall i$ Performs Hypothesis Testing by first calculating
Chi- computed ($\chi^2[i]$ using eq.1) for X_i
$H_o[i]$: Observed distribution of NCM (X_i) fits the expected
$H_a[i]$: Observed distribution of NCM (X_i) does not fit expected
.If (chi-computed[i] (α.d.f[i]) > P-value[i] (α.d.f[i]))
Reject $H_o[i]$. endif.
.End for
.Combined Null Hypothesis Testing
Combine H_o: Observed distribution of NCM fits the expected
Combine H_a: Observed distribution of NCM does not fit expected
.If (combined H_o is rejected)
*Perform **Attack identification & inferences Fig.3 (b)***
else: Update Expected values NCM($\overline{X}_i$) (i.e. ITP(NCM))
.Exit

Figure 3a. Pseudocode of intrusion detection module.

$$\chi^2[i] = \forall_i \left(\sum_{k=1}^{M} \frac{(NCM(X_{ik}) - NCM(\overline{X_{ik}}))^2}{NCM(\overline{X_{ik}})}\right) \ldots\ldots\ldots (1)$$

This module continuously monitors the network. In each TI the CH first performs hypothesis testing for each parameter X_i of NCM at calculated chi-computed values obtain from eq.1, where X_i is the parameter of NCM and k(1 to M) is the number of random variable in each parameter X_i. The CH then performs combine hypothesis testing of NCM as shown in fig 3a. If the

combined H_o is rejected then it assumes intrusion in the TI. Else we update the ITP (NCM) using an exponentially weighted moving average (EWMA) :

$$\forall_i(\overline{NCM(X_i^{(q,k_i^M)})} = \alpha * NCM(X_i^{(q,k_i^M)}) + (1-\alpha) * \overline{NCM(X_i^{(q,k_i^M)})}) \quad ..(2)$$

where $\overline{NCM(X_i^{(q,k_i^M)})}$ and $NCM(X_i^{(q,k_i^M)})$ represents the expected and observed value for update period number(q) respectively. The value of q is incremented in the TI when no intrusion in the MANET is detected. k represents the random variable from 1 to M in each X_i and α=2/(q-1) is the weighting factor. As q increases the weighting for older data points decreases exponentially giving more importance to the current observation.

b) Attack identification and inferences

```
.Reads set of rules Fig.3b (1)
.Set up the Interpreter for Rule-based approach
.Interpreter applies Forward-Chaining on set of rules Fig.3b (1)
   .If (Any Goal Condition of known attacks are fulfilled)
      Apply rules for IntruderIdentification & Isolation Fig.3c
   .endif.
   .If (Goal Condition==" POTENTIALUNKNOWNATTACK")
      Interpreter applies Attack Inferences Fig.3b (2),
   .endif.
. Exit.
```

Figure 3b. Pseudocode of Attack Identification & Inferences module.

Set of Rules example

```
Rule.1 ∃x (chi-squaretest(NCM[x]))-> (CheckDerivedMatrix=TRUE)
Rule.2 CheckDerivedMatrix Λ ∃y (Test(DM[y]))->
        (PotentialAttack=TRUE)
Rule.3  PotentialAttack  ->(BestRule=TRUE)
        Best Rules for some known attacks:
Rule.4  BestRules Λ (chi-squaretest(NCM[RREQ])) Λ
        Test(DM[CPO]) ->  "SLEEP DEPRIVATION"
Rule.5  BestRules Λ (chi-squaretest(NCM[RREPdest_seq])) Λ
        (Test(DM[PDR]) V Lowest(PDR) )  -> "BLACKHOLE"
Rule.6 BestRules Λ (chi-squaretest(NCM[RREPdest_seq])) Λ
        (Test(DM[PDR])  -> "GREYHOLE"
Rule.7 BestRules Λ (chi-squaretest(NCM[RREQsrc_seq])) Λ
        (Test(DM[CPD])  -> "RUSHING"
Rule.8  ¬( ∀x (chi-square-test(NCM[x]))) Λ ¬( ∀y( Test(DM[y]))) -->
        "POTENTIALFALSEALARM"
Rule.9  (Rule.1 Λ  Rule.2  Λ ¬BestRule)  ->
        " POTENTIALUNKOWNATTACK"
```

Figure 3b (1). Set of Rules example in knowledge base.

Attack Inferences

```
. If (Detected Rule Trace is Empty)
   Store Detected Rule Trace = Rule Trace
  Else If (Rule Trace == Detected Rule Trace)
        New attack Rule Trace= Rule Trace
        Construct a rule for New attack Rule Trace
       Append New attack Rule Trace in set of rule trace
       Set Detected Rule Trace =Empty . endif
.endif
```

Figure 3b (2). Pseudocode of Attack Inferences

In case of intrusion the CH calls the Attack Identification and Inferences module (Fig.3b). This module obtains a set of rules from knowledge base, an example set being presented in fig.3b(1). We have constructed these rules from our previous work [1] (i.e. AIDP simulation results), analyzing various attacks & their impact on network performance through

c) Intruder Identification &Isolation

```
a)  Identifying intruding nodes
   . Obtain known attack Rules for intruder Identification
   . for all Goal condition fulfilled
  Apply intruder identification rule for each detected known attack
      add each detected node Vi to List of Nodes Detected (LND)
   . endfor
b)  Response Mechanism
  For all nodes Vi in LND
   . If ( Vi detections in Potential Intruder List( PIL) >
                        Detections_required_To_ Accuse (d) )
      CH: Blacklist Vi & Broadcast Accusation Packet (AP)
    else :   enter Vi in PIL   .endif
  .End for
c)  Accusation Packet (AP) Handling
  . Each CN Vi maintain its local BlacklistTable (BLT)
   .if CN Vi receives an AP for CN Vj
     .If CN Vi has node Vj  in its BLT  then  Ignore AP
      else:   CN adds node Vj  to its  BLT & rebroadcast AP
     .endif
   .endif
d)  Isolating Intruding Nodes
   .if node Vi receives packet from node Vj
     .If node Vj is in node Vi  BLT
          Ignore packet & drop all packets queued from Vj
    Else:   handle & process packet   .endif
   .endif
```

Figure 3c Pseudocode of Intruder Identification & Isolation modules.

simulations and analysis of existing literature of known attacks for example [3, 4 & 6]. In fig.3b(1) *chi-square test(NCM[x])* predicate returns true if the parameter x is anomalous in NCM. Similarly predicate or propositional function *Test (DM[y])* returns true if the test on parameter y of DM fails. This test uses a tool of Statistical Process Control known as variable control chart based on standard deviation σ. In the Attack Identification & Inference module a ruled base approach is used in which an interpreter can either employ forward or backward chaining system. A forward chaining system process rules one by one by checking premise (condition in the rule) to reach conclusions, it can also draw new conclusions. On the other hand backward chaining is goal driven, that is it reaches the conclusion first and keeps looking for rules that would allow the conclusion. In GIDP an interpreter applies forward chaining on Set of rules fig.3b (1), at the end look for the Goal Condition fulfilled as described in fig.3b.

In case of any known attack detected in the TI, the interpreter applies the Intruder Identification & Isolation module (fig.3c) to identify and isolate the intruding nodes. This module first identifies the intruding nodes by applying known attack rules for intruder identification. For example in case of SD attack it employs control chart (explained above) based on σ of RREQ generated by all nodes and adds detected node V_i in the LND. Response mechanism (fig.3c(b)) then checks if detection threshold d is reached for any node V_i in the LND in p TI then it Blacklist the node V_i and inform all other CNs through sending an AP. When

a CN receives an AP it first checks the broadcast id & source address to avoid processing a duplicate AP. If the accused node is already blacklisted the CN will ignore & drop the AP to prevent unnecessary network traffic. Otherwise, the CN will blacklist the accused node and rebroadcast the AP. Finally, to isolate the intruder form the network all nodes will not only drop the packets from a blacklisted node but also immediately ignore all packets in their queue from the blacklisted nodes as shown in Fig.3c(d).

If Goal Condition with *POTENTIALUNKNOWNATTACK* is fulfilled during attack identification process then interpreter save this *Rule Trace* and looks for the match of this *Rule Trace* in current TSW. If match is found then it confirm new attack detection by constructing a new rule and appending the new rule in a Set of Rule stored in knowledge base (fig.3b(2)).

V. CASE STUDY AND EVALUATION

To assess the applicability and performance of GIDP, we considered a case study with different attack scenarios. We present the simulation results of these scenarios and some key findings from the analysis of attacks. We used GloMoSim to build the simulation environment and then evaluate GIDP using simulation & GIDP parameters shown in Table 1.

TABLE I. Simulation & GIDP Parameters

Simulation Parameters		
Number of nodes	25	50
Terrain dimensions	500*500 metres	707.10*707.10 metres
Node placement	Uniform distribution	
Simulation Traffic	CBR (Constant Bit Rate)	
Simulation time	2500 seconds	
Routing protocol	AODV	
MAC protocol	IEEE 802.11	
Mobility	Random Way Point Model (RWP)	
GIDP Parameters		
Time interval TI	100 seconds	
Training, Testing TI	Training=5 TIs, Testing= 20 TIs	
Number of parameters	NCM =7 & DM=3 parameters	
Chi-square test (α)	5% (i.e. 95% confidence interval)	
Test Sliding Window(TSW)	5 TIs	
Detections-Required To_Accuse (*d*)	2 in a TSW	
Number of intruders	1or 2 or 3 or 4	

A. *Scenario 1*

In the first scenario we test GIDP with a denial of service attack (sleep deprivation) through malicious RREQ flooding (MRF), as described in section III-a. The intruders launch MRF1 or MRF2 attacks. At each tested mean speed and for each network size (25 or 50 nodes) we performed 40 runs with no intrusion and 40 runs with intruders using a mix of both MRF1 and MRF 2.

The graph on the left in Fig. 4 depicts the success rate (SR) and false alarm (FA) rate of GIDP as a function of the nodes' mean speed in 25 & 50 node

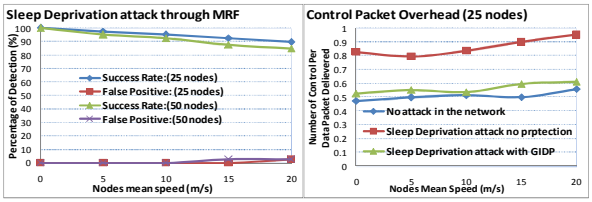

Fig. 4 Success rate, false alarm rate and control packet overhead as a function of nodes mean speed (m/s).

networks with SD attack. By SR here we mean the rate of correctly detecting intrusion in the network, identifying the attack type and then identifying & isolating the node which is causing the attack. A false alarm (FA) means that a correctly behaving node has been incorrectly identified and isolated. The graph shows good performance of GIDP in terms high SR and low FA rates against SD attack. The graph on the right in Fig.4 shows the control packet overhead in a 25 node network when there is a) no attack in the network, b) a sleep deprivation attack with no protection and c) a sleep deprivation attack with GIDP in place. The graph shows that GIDP reduces the control packet overhead and increases network performance when it is used in a network under sleep deprivation attack.

B. *Scenario 2*

In the second scenario we test GIDP with a mix of black and grey hole attacks caused by initiating a false RREP and then dropping packets as described in section III-b. In order to launch these attacks, on receiving a RREQ an intruder generates a false RREP packet with *dest_seq=current dest_seq+f*. Through simulations we observed that the value of *f* should be at least *5* in a 25 node network, and higher for larger networks, because some properly behaving nodes have routes fresher than the intruding node for the destination node. We also note that the severity of the attacks depends on the number of paths in the network that the intruder manages to capture. One false RREP packet only allows an intruder to capture the route of one node in the network, because RREP packets are unicast.

A single simulation consists of 20 test TIs. We monitor the number of false RREP packets (*e*) generated by an intruding node in a simulation and its impact on packet delivery ratio. Fig.5 shows that increasing the value of *e* reduces the packet delivery ratio during the BH attack and therefore increases the severity of the attack.

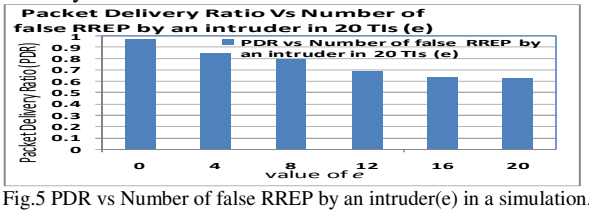

Fig.5 PDR vs Number of false RREP by an intruder(e) in a simulation.

The graph on the left in Fig.6 depicts the SR and FA of GIDP with black & grey hole attack with $8 \leq e < 20$ and $5 \leq f \leq 30$. The graph on the right in Fig.6 shows the packet delivery ratio with no attack, black & grey hole attack with no protection and black & grey hole attacks with GIDP in place. It shows that GIDP can successfully detect these attacks, and identify & isolate the intruding node and by doing so GIDP also improves the network performance in terms of packet delivery ratio.

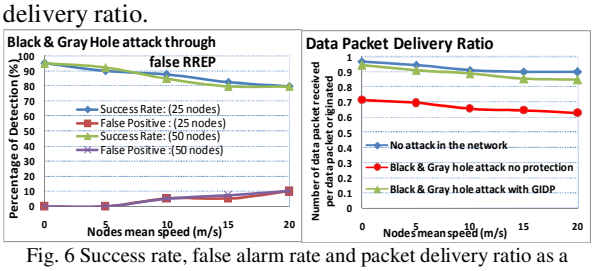

Fig. 6 Success rate, false alarm rate and packet delivery ratio as a function of nodes mean speed (m/s).

C. *Scenario 3*

In this scenario we tested GIDP with the rushing attack through forged RREQ as explained in section III-c. We note that intruders trying to cause rushing attack by sending a forged RREQ with a higher *src_seq* and minimum delay increase the number of control packets (i.e. RREQ+RREP+RERR) dropped in the network. Fig.7 shows that GIDP can detect rushing attacks and after isolating the intruder reduces the number of control packet dropped.

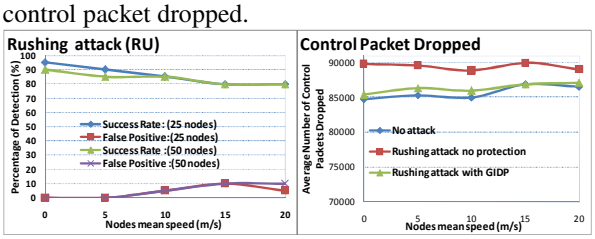

Fig. 7 Success rate, false alarm rate and control packet dropped as a function of nodes mean speed (m/s).

D. *Scenario 4*

In the last scenario we assess GIDP with a combination of simultaneous attacks (section III) launched by separate intruders in a simulation. We perform 20 runs with each combination of attacks. SR here means that GIDP has detected, identified and isolated *all* the intruders causing attacks. FA means GIDP has detected and isolated a properly behaving node as an intruder. Fig.8 depicts the success rate and false alarm rate of GIDP with the various combinations of attacks simulated. The graph shows generality of our proposed mechanism. During the experiments GIDP has flagged a *POTENTIALUNKNOWNATTACK* on a few occasions but they did not meet the criteria of GIDP attack inferences (i.e. *d* detections of same rule trace in a *TSW*) (fig3b.(2)) to mark them as a new attack.

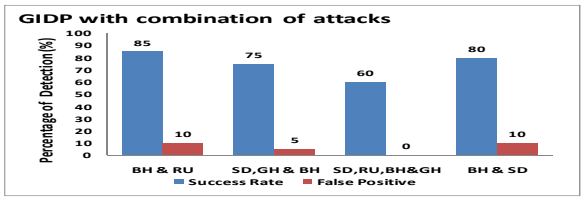

Fig. 8 GIDP success & false alarm rates with combinations of attacks.

VI. CONCLUSIONS AND FUTURE WORK

Many proposals have been made in the literature to detect various attacks, but most are attack-specific. Unlike some mechanisms that provide protection through authenticated routing, in this paper we have proposed a Generalized Intrusion Detection & Prevention mechanism which monitors both network layer characteristics (NCM) and performance statistics (DM). GIDP uses a combination of anomaly-based and knowledge-based ID that can protect MANETs against a variety of attacks from both external and internal intruders and also has the capability of detecting new unforeseen vulnerabilities. Simulation results show our proposed mechanism can secure MANETs from a wide variety of attacks with an affordable processing overhead. In our ongoing work we are focusing on implementation issues of GIDP and so that it can operate & adapt to networks with different security requirements.

REFERENCES

[1] A.Nadeem and M.Howarth, "Adaptive intrusion detection & prevention of Denial of Service attacks in MANETs", Proceeding of ACM 5th International Wireless Communication and Mobile Computing Conference, Germany, June 2009.

[2] E.Padilla, N.Aschenbruck, P.Martini, M.Jahnke and J.Tolle, "Detecting Black Hole Attack in Tactical MANETs using Topology Graph", Proceeding of 32nd IEEE Conference on Local Computer Networks, 2007.

[3] S.kurosawa and A.Jamalipour, "Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method", International Journal of Network Security, Vol.5, November 2007.

[4] G.Xiaopeng and C.Wei, "A Novel Grey Hole Attack Detection Scheme for Mobile Ad-Hoc Networks", Proceeding of IFIP International Conference on Network & Parallel Computing, 2007.

[5] P.Yi, Z.Dai and S.Zhang, "Resisting Flooding Attack in Ad Hoc Networks", Proceeding of IEEE Conference on Information Technology: Coding and Computing", Vol.2, pp 657-662, 2005.

[6] Y.Hu, A.Perrig and B.Johnson, "Rushing Attack and Defense in Wireless Ad Hoc Networks Routing Protocols", Proceeding of 2nd ACM workshop on Wireless Security, New York, 2003.

[7] K.Sanzgiri and M.Belding-Royer, "A Secure Routing Protocol for Ad Hoc networks", Proceedings of 10th IEEE International Conference on Network Protocol 2002, (ICNP' 02).

[8] Y.Hu, A.Perrig and B.Johnson, "A Secure On Demand Routing Protocol for Ad Hoc networks", Proceeding of MobiCom, Atlanta, Georgia, USA, pp 23-28, September 2002.

[9] M.Pirrete and R.Brooks, "The Sleep Deprivation Attack in Sensor Networks: Analysis and Methods of Defence", International Journal of Distributed Sensor networks, Vol.2, No.3, pp 267-287, 2006.

[10] KDD data set, 1999. URL:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.