Efficacy of Misuse Detection in Adhoc Networks

Dhanant Subhadrabandhu, Saswati Sarkar, and Farooq Anjum

Abstract— **We consider adhoc networks with multiple, mobile colluding intruders. We investigate the placement of the intrusion detection modules for misuse intrusion detection. Our goal is to maximize the detection performance subject to limitation in the computational resources. We mathematically formulate different detection objectives, and show that computing the optimal solution is NP-hard in each case. Thereafter, we propose a family of algorithms that approximate the optimal solution, and prove that some of these algorithms have guaranteeable approximation ratios. The algorithms that have analytically guaranteeable performance require re-computation every time the topology changes due to mobility. We next modify the computation strategy so as to seamlessly adapt to topological changes due to mobility. Using simulation we evaluate these algorithms, and identify the appropriate algorithms for different detection performance and resource consumption tradeoffs.**

I. INTRODUCTION

Adhoc networks are likely to find large scale applications in scenarios where infrastructure can not be used for communication, e.g., disaster recovery, battlefields, homeland security e.g., monitoring superbowl matches, ensuring security for dignitaries etc. For large scale deployment, adhoc networks must be designed so that the applications can be used by a large number of users; this introduces significant security risks. One such risk can be a user who subverts the functioning of the network by causing undesirable events. Such users are considered as intruders and the events as intrusions. Examples of intrusions are denial of service attacks e.g., TCP SYN flood *, smurf †, attacks that use abnormal packets, e.g., packets with the same source and destination address, packet fragmentation attacks, e.g., ping of death, attacks on the authentication mechanisms, bubonic attacks ‡ etc. [7] [6]. Some intrusions can be prevented by designing resistant protocols e.g., SCTP [23] resists TCP SYN flood attacks, and techniques like challenge response authentication mechanisms [2], etc. But, there have been many instances where intrusions could not be prevented even when these techniques have been used, since they leveraged vulnerabilities present in the system. For example, WEP authentication, which is based on a challenge-response mechanism, fails to prevent intruders from authenticating themselves even when the system key is not compromised [18]. Similarly, an intruder can use an authentication flaw in windows debugger [1] to log interactively into another machine and obtain root privileges. Also, the prevention mechanisms fail when system secrets like encryption keys are compromised. It is therefore important to devise mechanisms to recover from intrusion. The first step towards recovering is to detect intrusion. For example, when a node detects a malicious packet, it can drop the packet, and thereby recover from intrusion. We focus on efficiently detecting intrusions. Both intrusion prevention and recovery, though important, are beyond the scope of this paper.

Intrusion detection has been extensively investigated for wireline networks [8], [11]. But techniques geared towards wireline networks would not suffice in an adhoc network due to mobility, the ease of listening to wireless transmissions, lack of fixed infrastructure, etc. [13]. For example, several detection strategies in wireline networks are based on the presence of a small number of gateways that route and therefore monitor all traffic. But, adhoc networks typically do not have such choke points. Also, intrusion may be detected in wireline networks by detecting anomaly, i.e., by comparing the current system behavior with that in absence of intrusion. In adhoc networks, however, normal behavior can not be accurately characterized, e.g., a node may transmit false updates since the routing protocol is slow to converge and not because it is malicious. Further, unlike in wireline networks, nodes in an adhoc network have limited energy. Hence, only computationally simple, energy-efficient detection strategies can be used. The detection algorithms must also be distributed as communication with a central computing unit will consume significant energy. Finally, the detection algorithms must seamlessly adapt to topological changes due to mobility. These motivate the design of detection strategies specifically geared towards adhoc networks.

A strategy specifically suitable for adhoc networks is that of misuse detection that relies on the use of known patterns of unauthorized behavior. More specifically, this technique detects intrusion when the transmitted traffic contains abnormal packets which serve as "signatures" of attacks. For example, a UDP packet destined to port 0 can crash some machines [7]. The signature of ping-of-death attack is a very large sized ping packet, that of RPC locator attack is a packet intended for port 135 that contains a command that the system is not expecting, that of Bubonic attack are various values such as a TTL of 255, a TOS field value of $0xC9$, exactly 20 byte payload in the IP datagram and the fragment ID value with consistent increments of 256 [7]. The appearance of a large number of SYN packets without the corresponding ACK packets indicate a SYN flood attack [7]. Due to low false alarm rates, misuse detection is the mainstay of current commercial intrusion detection systems in wireline networks and wireless LAN. This technique can not however detect new attacks, i.e., the attacks whose signatures are unknown. Nevertheless, it is the most suitable technique in adhoc networks given that it does not require characterization

D. Subhadrabandhu and S. Sarkar are at the department of Electrical and Systems Engineering at the University of Pennsylvania. F. Anjum is at the Telcordia Technologies. Their emails are dhanant@seas.upenn.edu, swati@seas.upenn.edu, and fanjum@telcordia.com.

*Here, an attacker violates the 3 way handshake of TCP and opens a large number of half-open TCP connections.

†Here, an attacker sends forged ICMP packets to a broadcast address.

‡Here an attacker randomly sends TCP packets (e.g., without waiting for acknowledgements) with settings which crash the machine.

of normal behavior.

But a prerequisite for deploying misuse detection in adhoc networks is to determine which nodes should execute the sniffing and analysis software modules which we refer to as the intrusion detection system (IDS) modules in the sequel. We show that different selection strategies can have significantly different detection performance and execution costs (Section II). It is therefore crucial to deploy appropriate selection strategies that attain the desired tradeoff. We mathematically formulate the problem of selecting the nodes so as to minimize the execution cost subject to maximizing the detection performance (Section III-B). We prove that computing the optimal selection strategy is an NP-hard problem. Then, we present distributed approximation algorithms that attain guaranteeable approximation bounds. We also consider the dual problem of selecting the nodes so as to maximize the detection performance subject to not consuming more than a predetermined amount of resource (Section III-C). We show that this problem is also NP-hard, and outline the design of distributed approximation algorithms. The algorithms that have analytically guaranteeable performance require re-computation every time the topology changes due to mobility. We next modify the computation strategy so as to seamlessly adapt to topological changes due to mobility (Section III-D). We evaluate the proposed algorithms using extensive simulations (Section IV). In Section V, we describe the relevant literature. Due to lack of space, we present proofs in a technical report [24].

The characterization of the optimal selection strategy allows us to identify the appropriate selection strategy for realizing desired tradeoffs between the detection efficiency and resource consumption. Our investigations reveal that the optimal strategies consume significantly lower resource in detection as compared to heuristics when high detection rate is necessary, and thus must be deployed in this case. But, when the system can tolerate certain amount of intrusion and therefore the detection rate can be small, the heuristics and the optimal selection strategies consume similar resources. Thus heuristics may be deployed in these scenarios. We also observe that the optimal selection algorithms detect all malicious packets by executing the IDS in only a small fraction of the nodes (typically less than 15%). This is an encouraging outcome as in most adhoc networks at least a small number of nodes will have significant energy. Thus, it would be sufficient to execute the IDS in only these nodes.

II. System Architecture

The network consists of two types of nodes: insider nodes and outsider nodes. Insider nodes, e.g., PDAs, cellphones, laptops, media servers, location servers etc. perform various system tasks, e.g., serve as endpoints of the flows, relay the traffic, auto-configure the system, detect intrusion, etc. Outsider nodes are the sources of traffic, and may also be destinations, but do not perform any other system task. A session originates from an outsider node, traverses insider nodes and terminates at an insider or another outsider node. The sessions continuously join and leave the network. A node may serve as both a source and a destination (of different sessions) - such a node does not perform any other system task and is referred to as an outsider as well. An outsider node may wish to deliver malicious packets to the destination, which may be an insider or another outsider, resulting in harm to or failure of the destination node. Hence, an outsider is referred to as an intruder. There may be multiple intruders who may collude. Insider nodes execute the IDS so as to detect the malicious packets while in transit between the intruder and the destination. We assume that insider nodes are not compromised, i.e., there are no insider attacks [4]. Authentication mechanisms prevent an outsider from pretending to be an insider node.

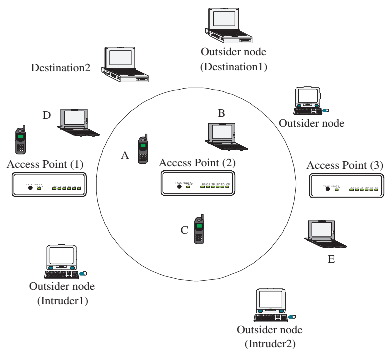


Fig. 1. This figure illustrates the system model. The outsider nodes (Intruder1, Intruder2) attack the destinations (1,2). The insider nodes are the access points, mobile terminals and laptops (B, D, E).

We now describe some systems that can be described as above. Consider a UN peacekeeping force whose mandate is to maintain peace in troubled areas and to provide basic networking services such as email, news, entertainment (games, music) etc. Adhoc networks consisting of small access points on buildings and mobile terminals carried by the personnel can be used for the communication (Figure 1). Civilians can communicate with each other by accessing the network using devices such as laptops, cell-phones and PDAs which are the outsider nodes. Some of these outsider nodes will be malicious. The access points and mobile terminals in the network relay the traffic and perform tasks related to the mission. The intruders may attack the machines that they deem important for the mission objectives. The access points, mobile terminals and the targeted machines are the insider nodes. Since the insider nodes are controlled by the peacekeeping force, authentication mechanisms can be implemented to protect communications amongst these.

A similar example can also be given in the setting of a university where static and mobile access points are deployed at various places on campus to allow outsiders (students) access to various destinations. Here, mobile access points are those on the vehicles. The destinations can be PDAs in labs that allow students to contact teaching assistants, lab personnel, or specialized servers that allow the students to check their grades, register for courses, participate in class discussions, etc. The insider nodes consist of the access points and the destinations (which may also relay traffic for other flows). Some students may send malicious packets towards the destinations.

We consider a misuse-based detection strategy. The nodes that execute the IDS detect malicious packets. Some insider nodes may not have the capability to execute the IDS. Thus, insider nodes are of two types: (a) *IDS capable* and (b) *IDS incapable*. Also, different IDS capable insider nodes consume different amount of resources to execute the IDS. This is because nodes, e.g., PDAs, laptops, access points etc. have different residual energy and computational capability. Depending on the system policy, some but not all the IDS capable insider nodes will execute the IDS - these are denoted as *IDS active*. The selection of the IDS active nodes will depend on the available bandwidth, computational resources, energy and the topology that change with time. We now examine the tradeoffs associated with different selection strategies.

An obvious IDS placement strategy (host intrusion detection or HID) [16] for adhoc networks is to execute the IDS at only the destinations of the sessions, e.g. destinations $1, 2$ in figure 1. Here, a node executes the IDS at its application layer, and can therefore analyze only the packets it receives as destination, and not those that it relays. The advantage of HID is that it is not affected by the use of end-to-end cryptography or by changes in topology and routing that may be triggered by node mobility. But HID has several serious drawbacks. First, an intruder can avoid detection and produce maximum damage by exploiting the knowledge that only the destination analyzes the packets. For example if the intruder knows that the destination is using windows 98 operating system, then it can transmit a packet that crashes the machine as soon as the destination's network layer assembles the packet and before the IDS at the application layer analyzes the packet. Second, the detection mechanism will use the computation resources and network interfaces at the end-host. But, an attack on the target may simultaneously exhaust the resources available for detecting and reporting such attacks. Third, many of the destination nodes may not be able to execute the IDS due to limited computational resource and low residual energy. Finally, if only the end-hosts execute the IDS, then the malicious packets would not be dropped until they reach the destination. Thus several nodes expend their limited energy and available bandwidth in relaying malicious packets.

The network intrusion detection (NID) technique [16] executes the IDS on some selected insider nodes, e.g., the access point(2) in figure 1, which may be relays or end-hosts. Here, a node executes the IDS at its network layer, and can therefore analyze both the packets it relays and receives as destination. In adhoc networks, NID has several advantages over HID. First, an intruder can no longer be certain that only the destination is executing the IDS. Moreover, the IDS active nodes can be selected so that they have different characteristics. Thus, it will be more difficult for the intruders to devise attacks that are not detected. Second, the IDS active nodes can be selected only among those that have the required capability. Third, NID captures malicious packets in transit and thus limits the wastage of bandwidth and energy in relaying them. We consider NID in this paper.

The challenge in deploying NID is to appropriately select the IDS active nodes. A straightforward strategy is to execute the IDS on every insider node. Thus every malicious packet will be detected. But this significantly increases the energy consumption and the computation. On the other hand, if the IDS are executed in very few nodes, then the resource consumption decreases but some malicious packets may escape inspection leading to undetected intrusion. The challenge is to select the IDS active nodes so that the resource consumption is minimized subject to ensuring that every packet is inspected at least once by an IDS active node. In wireline networks, the IDS is executed in choke-points or gateways that relay all traffic and can therefore capture and analyze all packets. But, as discussed before, adhoc networks typically do not have such choke-points. Also, even if such choke-points exist, their locations continuously change due to mobility. Thus, designing the optimal selection strategy is more complex in adhoc networks.

The IDS modules can clearly analyze un-encrypted traffic, but it is not limited to this case. It can also analyze encrypted traffic when encryption is not at a layer at which it operates. For example, when traffic is encrypted at the application layer, IDS modules can detect attacks at transport and lower layers, e.g., ping-of-death, TCP SYN flood, smurf, bubonic, etc. If encryption is used at all layers, e.g., in battlefield networks, then schemes can be designed to distribute the keys securely to the IDS active nodes. Investigation of key distribution schemes is beyond the scope of this paper.

The IDS active nodes may either operate individually or "cooperate," i.e., aggregate each others analysis. When acting individually each such node detects intrusion based only on the packets that it analyzes. Thus, an intrusion can be detected only when all the packets that constitute the attack are analyzed by one IDS active node. Now, individual operation is sufficient against single packet attacks such as Code Red and Slammer [7] and multiple packet attacks in which all malicious packets traverse the same route. But, unless nodes cooperate, multiple packet attacks e.g., ping of death or teardrop, in which malicious packets traverse different routes, will not be detected; different packets may traverse different routes due to topology changes or the intruder's route selection [19]. In our technical report, we demonstrate that the detection rate increases significantly due to this cooperation [24]. We therefore assume that nodes cooperate. Other authors have also assumed cooperation, e.g., [22], [25]. The disadvantage of cooperation is that it increases the complexity and the resource consumed in the detection - this is also the case in [22], [25]. For example, each node has only partial information and different nodes have different information. Therefore, nodes must exchange messages and agree whether a packet or a sequence of packets is malicious. Nevertheless, it is important to investigate the maximum possible detection assuming full cooperation among nodes. This is what we focus on. The design of schemes for achieving complete or partial cooperation is a topic of future research.

III. Intrusion detection in presence of resource limitation

We first mathematically formulate the system assumptions and capabilities (subsection III-A). Thereafter, we consider different detection objectives. Specifically, we consider the problems of selecting the IDS active nodes so as to: (a) minimize the resource consumption subject to maximizing the detection efficiency (subsection III-B), and (b) maximize the detection efficiency subject to upper bounding the resource consumption (subsection III-C). We prove that both problems are NP-hard. In each subsection, we present distributed computation algorithms for approximating the respective optimal solutions within provable approximation bounds. These algorithms are oblivious to the movement of outsider nodes, but require re-computation whenever insider nodes move. In subsection III-D, we present algorithms that do not have provable approximation bounds, but nevertheless avoid such re-computations and are therefore more suitable when insider nodes move rapidly.

A. Mathematical formulation for system assumptions and capabilities

We represent a wireless network by an undirected graph $G(V, E)$. Here, $V = \{1, \ldots, N\}$ consists of the insider nodes and E is the set of edges between the insider nodes. A node u can receive transmissions from any node v which is within a distance r from u. There exists an undirected edge between any two nodes whose distance is r or less.

Definition 1: A *neighborhood* N_i of an insider node i is the set of insider nodes that are within distance r from i. An insider node i *covers* every node in its neighborhood.
By this definition, an insider node is always its own neighbor and covers itself.

If an outsider node is within distance r of some insider node, it can transmit a packet through the network to an insider or another outsider. It may use any set of paths for transferring its packets. The number and the locations of the outsider nodes and their destinations are not known to the network, and vary with time. An IDS capable insider node i has weight w_i that represents its resource consumption when it executes the IDS.

An IDS active insider node operates in promiscuous mode, i.e., receives any packet that is transmitted by any of its neighbors. For example in figure 1, if access point(2) is IDS active and operates in promiscuous mode, it can analyze the packets transmitted by nodes A, B, C. Clearly, operation in promiscuous mode increases the power consumption of these nodes. But, many other authors also assume similar operation, e.g. [15],[20]. More importantly, the following observation shows that such operation actually reduces the overall energy consumption in the network.

Lemma 1: All malicious packets will be detected if and only if every insider node satisfies one of the following properties. (a) It executes the IDS. (b) It is covered by an IDS active insider node that operates in promiscuous mode.
We explain the consequence of this lemma. Suppose the system requires 100% detection, i.e., all malicious packets must be detected. But, only few IDS active nodes operate in promiscuous mode. Then, a large number of insider nodes will need to execute the IDS. For example, if none of the IDS active nodes operate in promiscuous mode, then all insider nodes must execute the IDS. Our simulations demonstrate that the

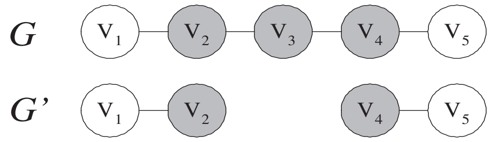


Fig. 2. We show example graphs G and G'. The shaded nodes are IDS incapable. Thus, only v_1, v_5 may be IDS active. Here, v_3 is a removal node, and hence can not be covered by any IDS active node, and hence can not be in G'. Now, v_1, v_5 constitute an IDS capable dominating set in G'.

algorithms we propose execute the IDS in only a small fraction of the insider nodes, and thus their operation in promiscuous mode consume much less energy than executing the IDS in all insider nodes.

We assume that a path between an outsider node and its destination always has an intermediate insider node, i.e., we do not allow direct communication between the source and the destination. A destination can enforce this assumption by refusing to accept any packet with MAC layer source address different from that of an insider node. This assumption has been motivated by the need for implementing recovery mechanisms. For example, if the destination is IDS incapable, which may happen if it has limited resources, then the destination can not recover from the attack if it receives a malicious packet directly from the intruder. Note that the intrusion may still be detected if the destination is covered by an IDS active insider node.

B. Selecting the IDS active nodes so as to minimize the resource consumption subject to maximizing the detection

We assume that the IDS active nodes cooperate. Thus a malicious packet is detected if it is transmitted by any node that is covered by an IDS active insider node. *Our goal now is to select the IDS active nodes among the IDS capable nodes such that they cover the maximum possible number of insider nodes while minimizing the sum of the weights of the IDS active insider nodes which is the resource consumed in the detection.*

Definition 2: A *dominating set* D in G is a set of nodes such that every node in G is either in D or covered by a node in D.

If the set of IDS active insider nodes is a dominating set in G, then any packet transmitted by an insider node is received by at least one IDS active insider node. Since irrespective of the position and the number of the sources, destinations and the paths between them, a packet must be relayed by at least one insider node, every packet is received by at least one IDS active insider node. Thus, every malicious packet is analyzed and hence detected since nodes cooperate. Since some nodes are IDS incapable, it may not be possible to have the IDS active nodes constitute a dominating set and therefore cover all insider nodes. Specifically, IDS active nodes will not cover the IDS incapable nodes that are not in the neighborhood of any IDS capable node (Figure 2). In this case, we have to opt for maximum possible coverage.

Definition 3: An IDS incapable node that is not in the neighborhood of any IDS capable node is denoted *removal node.*

Consider a graph G' which is obtained by removing the removal nodes and their edges from G (Figure 2). Thus, the vertex set in G', V', consists of insider nodes that are covered by at least one IDS capable insider node.

Lemma 2: The IDS active insider nodes cover the maximum possible number of nodes in G, if and only if they form a dominating set in G'.

The lemma follows from the observation that an insider node can be covered by an IDS active node if and only if it is in V'.

Definition 4: An *IDS capable dominating set* is a dominating set such that its members are IDS capable.

Definition 5: A *minimum weighted dominating set* is a dominating set such that the total weight of all its members is the minimum among all dominating sets.

Definition 6: An *IDS capable minimum weighted dominating set* is an IDS capable dominating set with minimum total weight among all IDS capable dominating sets.

Observation 1: There exists at least one IDS capable dominating set in G'.

Thus there exists an IDS capable minimum weighted dominating set in G'. *By Lemma 2, the optimal set of IDS active nodes will form an IDS capable minimum weighted dominating set in* G'. So far, we have not considered the weights of the IDS incapable insider nodes, as an IDS capable minimum weighted dominating set in G' will not include them. But, we now show that by appropriate selection of weights of these nodes, any IDS capable minimum weighted dominating set in G' becomes a minimum weighted dominating set in G' and vice versa.

Lemma 3: Let the weight of each IDS incapable insider node be a real number greater than the sum of the weights of the IDS capable insider nodes. Then, any IDS capable minimum weighted dominating set in G' becomes a minimum weighted dominating set in G' and vice versa.

Now, the optimal set of IDS active nodes forms a minimum weighted dominating set in G'. It is well-known that computing a minimum weighted dominating set is an NP-hard problem [10]. This motivates the following lemma.

Lemma 4: Optimally selecting the IDS active nodes is an NP-hard problem.

We present a distributed approximation algorithm for selecting the IDS active nodes based on the LP relaxation technique proposed by Kuhn *et al.* [14], and improve upon their approximation ratio.

Let N_i' be the neighborhood set in G' of an insider node $i \in V'$. For each $i \in V'$, we consider a variable x_i that is either 1 or 0. Consider a set V_d that consists of insider nodes i for which $x_i = 1$. Now, V_d is a dominating set in G' if and only if $\sum_{j \in N_i'} x_j \geq 1 \ \forall \ i \in V'$. Thus the computation of a minimum weighted dominating set in G' can be formulated as the following integer linear program, which we refer to as "minimizing resource dominating set (MRDOM_{IP})."

(MRDOM_{IP}) **Minimize :** $\sum_{i \in V'} w_i x_i$
subject to $\forall \ i \in V'$

1) $\sum_{j \in N_i'} x_j \geq 1$,
2) $x_i \in \{0, 1\}$.

By relaxing the integer constraints, we obtain the following linear program.

(MRDOM_{LP}) **Minimize:** $\sum_{i \in V'} w_i x_i$
subject to $\forall \ i \in V'$

1) $\sum_{j \in N_i'} x_j \geq 1$,
2) $0 \leq x_i \leq 1$.

Note that MRDOM_{IP} and MRDOM_{LP} are centralized optimization problems. But, we solve MRDOM_{LP} optimally using the following distributed iterative approach. Each insider node i in V' maintains the following variables: (a) node indicator ψ_i, (b) weight w_i and (c) x_i. Variables ψ_i and x_i are updated in every iteration; their values in the nth iteration are ψ_i^n and x_i^n. Removal nodes (i.e., nodes in $V \setminus V'$) do not participate in the computation or message exchange. We assume that each insider node knows whether any of its neighbors is IDS capable, and can therefore determine whether it is in V'.

Let γ be a constant greater than 1. An insider node $i \in V'$ updates ψ_i and x_i as follows.

$$\psi_i^n = \begin{cases} 0 & \text{if} \quad \sum_{j \in N_i'} x_j^n \geq 1 \\ 1 & \text{if} \quad \text{otherwise.} \end{cases}$$

$$x_i^{n+1} = [x_i^n - \tfrac{1}{n}(w_i - \gamma \textstyle\sum_{j \in N_i'} \psi_j)]_+ \qquad \forall i \in [1, n].$$

Here, $x_+ = \max(x, 0)$. We now explain the intuition behind this iterative procedure. Each $i \in V'$ updates x_i using the node indicators ψ_j from all its neighbors $j \in G'$. The goal of the update in each iteration n is to obtain the smallest possible value of x_i for each node i such that for each of i's neighbors j, $\sum_{k \in N_j'} x_k^n \geq 1$. If the above constraints are satisfied in the nth iteration, then each such neighbor j's node indicator ψ_j^n equals 0. Thus, $\sum_{j \in N_i'} \psi_j^n = 0$. Hence, the algorithm reduces x_i^n by w_i/n. When $\sum_{j \in N_i'} \psi_j > 0$, the constraint $\sum_{k \in N_j'} x_k^n \geq 1$ is violated at some j in N_i'. Hence, i increases x_i so that the constraint will be satisfied at j.

In each iteration n, each insider node $i \in V'$ broadcasts the values x_i^n and ψ_i^n in its neighborhood. The information can be piggy-backed in the data or acknowledgement packets.

Theorem 1: For all $\gamma > 1$, irrespective of the values of $\vec{x}^0$ and $\vec{\psi}^0$, as $n \to \infty$, $\vec{x}^n$ converges to an optimal solution of the linear program $MRDOM_{LP}$.

Now, we describe how to obtain a dominating set using the optimal solution x_i of the linear program $MRDOM_{LP}$. Each $i \in V'$ computes the maximum degree $\delta_i^{(2)}$ among all nodes in V' at distance at most 2 from itself. Consider a set D which is initially empty. Then i joins D with probability $\min\{1, x_i \ln(\delta_i^{(2)}+1)\}$. Each i informs its neighbors whether it is joining D. If none of i's neighbors have joined D, it invites a neighbor j that has the smallest weight w_j in N_i' to join D. Recall that by definition a node is its neighbor. When a node j receives an invitation message from one of its neighbors, it joins D. The resulting set D is a dominating set.

Theorem 2: The set D is an IDS capable dominating set. Its expected weight (resource consumption) is at most $O(\log \Delta)$ times that of the IDS capable minimum weight dominating set in G', where $\Delta = \max_{i \in V'} |N_i'|$.

C. Selecting the IDS active nodes for maximizing the detection efficiency subject to bounded resource consumption

Now we assume that the network wishes to limit the resource consumed for detecting intrusion, and subject to this aspires to maximize the detection efficiency. This happens when the network reserves a part of its resources for other functions. Specifically, we assume that the IDS active nodes must be selected such that they cover the maximum possible number of insider nodes subject to constraining the total weight of the IDS active set to be less than a constant, L, which is a system parameter. This goal is the dual of minimizing the resource consumed for maximizing the detection which we considered in subsection III-B.

Theorem 3: The selection of IDS active nodes so as to maximize the number of insider nodes they cover subject to constraining the total weight of the IDS active nodes to be less than a given constant is an NP-hard problem.

We now outline the design of a distributed algorithm that approximately computes the optimal set of IDS active nodes. For this, we first formulate the problem as an integer linear program, which we refer to as "maximizing detection dominating set (MDDOM$_{\text{IP}}$)." Now, for each insider node $i \in V'$ there exists two integer variables: (a) x_i and (b) y_i. Now, x_i indicates whether an IDS active node covers i, i.e., $x_i = 1$ if an insider node in N'_i is IDS active, and 0 otherwise. Also, $y_i = 1$ if i is IDS active, and 0 otherwise. Thus, $x_i = \min(1, \sum_{j \in N'_i} y_j)$. Thus, since each y_j is a non-negative integer, x_i is either 0 or 1. The upper bound on resource consumption introduces another constraint: $\sum_{j \in V'} w_j y_j \leq L$. The goal of MDDOM$_{\text{IP}}$ is to maximize the number of nodes covered by the IDS active nodes, i.e., $\sum_{j \in V'} x_j$ subject to these constraints. We would also like to minimize the resource consumption subject to attaining the above goal. This is captured when the goal of MDDOM$_{\text{IP}}$ is to maximize $\sum_{j \in V'} \left(x_j - \frac{w_j}{\sum_{j \in V'} w_j} y_j \right)$. Note that the coefficient of each negative term in the objective is small enough such that MDDOM$_{\text{IP}}$ tries to minimize the sum of the negative terms only subject to maximizing the positive terms.

(MDDOM$_{\text{IP}}$) Maximize: $\sum_{j \in V'} \left(x_j - \frac{w_j}{\sum_{j \in V'} w_j} y_j \right)$

subject to

1) $x_i \leq \sum_{j \in N'_i} y_j, \ \forall \ i \in V'$,
2) $x_i \leq 1, \ \forall \ i \in V'$,
3) $y_i \in \{0, 1\}, \ \forall \ i \in V'$,
4) $\sum_{j \in V'} w_j y_j \leq L$.

The next step will be to relax the integer constraints for y_i, i.e, replace constraint (3) by $0 \leq y_i \leq 1 \ \forall i \in V'$ in MDDOM$_{\text{IP}}$. As in the previous subsection, the resulting linear program (LP) can be solved in a distributed manner, and the IDS active set can be constructed from the optimal solution of the LP. We omit the details due to page constraints.

Finally, the optimal solution of MRDOM$_{\text{IP}}$ can be obtained by using $L = \sum_{i \in V'} w_i$ in MDDOM$_{\text{IP}}$. Therefore, MRDOM$_{\text{IP}}$ is a special case of MDDOM$_{\text{IP}}$.

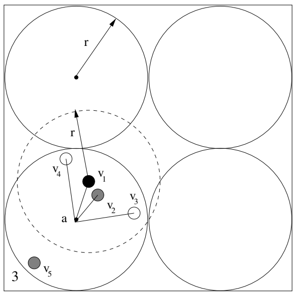


Fig. 3. This figure illustrates the operation of GO-DOM. The circles in solid lines are some of those that cover the geographic area of the network. For the current positions of the nodes, 2 nodes in circle 3, v_2, v_5, execute the IDS. Both v_2 and v_5 are the nearest in their neighborhoods to the center a of the circle they reside in, i.e., circle 3. Now, v_1 does not execute IDS as v_2 in v_1's neighborhood (the dashed circle) is nearer to a than v_1.

D. Robust heuristic algorithms for selecting the IDS active node when insider nodes move

The algorithms presented so far are oblivious to the position of outsider nodes, and are therefore not affected by their movements. But, the IDS active set must be recomputed each time an insider node's neighborhood changes due to its or its neighbors' movements. The computations and the related message exchanges consume significant resources particularly when they are executed frequently, i.e., when the insider nodes move rapidly. We now present computationally simple algorithms that do not require any re-computation with movement of either insider or outsider nodes, and require only limited message exchange when insider nodes move. The disadvantage is that we have not been able to prove any approximation bound for any of these algorithms. We evaluate them using simulation.

First, we consider a naive algorithm (RP) in which every IDS capable insider node executes the IDS with a probability which can be selected so as to regulate the resource consumed and detection rate. For example, if this probability is high, then a large number of nodes are IDS active. Thus, the detection consumes a lot of resource but most malicious packets are detected.

We now propose another heuristic, which we refer to as GO-DOM (geometric dominating set algorithm), that uses geometric information to select the IDS active nodes (Figure 3). The network is covered by the minimum possible number of circles each with radius r, where r is the transmission range of a node. The center of the outer most circle has a distance r from the closest network boundary. Each IDS capable insider node knows or computes the coordinate of the centers of the circles. Note that this is a one time computation or message exchange for each IDS capable insider node. We assume that each insider node knows its neighbors' coordinates by using GPS or other techniques e.g., [5]. An insider node selects an IDS capable neighbor which is the nearest to the center of a circle it currently resides in to execute the IDS (an insider node may select itself as well since by definition it is its own neighbor). For this, each IDS capable insider node broadcasts its distance from the center of each circle it resides in to its neighbors. It sends this broadcast packet when it joins the system, and thereafter each time it moves.

GO-DOM detects all malicious packets as it selects the IDS active nodes so as to cover the entire network. We now generalize GO-DOM so as to select fewer IDS active nodes at the expense of obtaining lower detection rates. Now, each node selected by GO-DOM decides whether to execute the IDS with a probability p which can be selected so as to regulate the resource consumed and detection rate. We refer to this version as generalized geometric dominating set algorithm (GGO-DOM).

In the next section, we compare the performances of the heuristics with the approximation algorithms, and determine when each may be deployed.

IV. Performance Evaluation

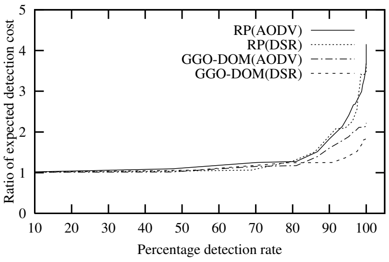
Using ns2-simulations, we compare the performance of the proposed approximation algorithms with the computationally simple heuristics. This comparison allows us to evaluate the benefits of (approximately) optimally selecting the IDS active nodes, and accordingly decide the appropriate algorithm for any desired tradeoff between detection efficiency and resource consumption. We do not present performance comparisons of the approximation and the optimal algorithms, since the optimal algorithms are computationally intensive and the optimal selections are NP-hard problems. But sample computations suggest that the approximation algorithms closely approximate the optimal algorithm, and in most cases the performance difference is much less than the upper bound guaranteed in Theorem 2.

We present averages of measurements for 200 different random topologies with nodes uniformly distributed in a square of side 670m. We simulate each topology for 200 seconds, and consider two different scenarios: (a) a single intruder launching an attack consisting of 5 packets and (b) 5 intruders colluding to launch an attack consisting of 10 packets. In (b) all intruders together send 10 malicious packets and in this sense they collude. In all cases, the intruders are selected randomly, and attack a single randomly selected outsider node. The route between the intruders and outsiders consist of multiple hops. For both (a) and (b) we again consider networks with (i) different number of insider nodes, e.g., 50 and 100, (ii) different transmission radii 250m and 150m, (iii) different routing protocols, e.g., AODV, DSR and (iv) different node mobility, e.g., mobile intruders and static insider nodes, mobile intruders and mobile insider nodes etc. Each mobile node moves as per the random way point mobility model with speed of 20 m/s and pause time 10 sec.

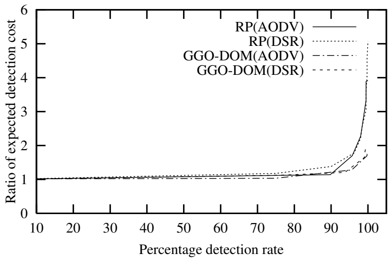
A malicious packet is analyzed and hence detected by an IDS active node if it is relayed by at least one neighbor of the node. The percentage detection rate is the percentage of the trials in which all malicious packets are detected. The total number of packets analyzed by the IDS active nodes in the system (*detection cost*) is a measure of the resource consumed in the detection process. This measure is related to the number of IDS active nodes, which we have considered in the mathematical formulation. We evaluate the detection cost using simulation as it depends not just on the selection of the IDS active nodes but also on the routing. For different algorithms, we plot (a) the ratio of their expected detection costs as a function of the percentage detection rate, and (b) the percentage detection rate as a function of the number of IDS active insider nodes. These plots will help us understand the tradeoff between detection efficiency and resource consumed in the detection process.

We now digress to explain why a packet may be analyzed by multiple IDS active nodes. First, a design objective is that each packet is analyzed by one IDS active node. But, this is generally not possible in practice. Since packets need not traverse the IDS active nodes that analyze them (the IDS active nodes can analyze the packets they observe due to promiscuous operation), they can not alter packets so as to indicate whether the packets have been analyzed. But, whenever an IDS active node l analyzes a packet p it can communicate a short message to a node m further along in p's path. If l finds p to be malicious, this message would instruct m to drop p; otherwise the message would instruct m to add an HMAC (message authentication code) [2] in p's header. Now, any other IDS active node n that receive p, (either through promiscuous reception or because p traverses n) checks if the HMAC value in p's header exists and matches the calculated value. If so, n need not analyze p. Otherwise, n can analyze p. But, if p is a good packet, in either case, n needs to process p (e.g., examine the HMAC etc.). Thus, good packets are likely to be processed several times by IDS active nodes. Note that most packets are good except when the attacks are worm-based [16], e.g., in our simulations outsiders transmit at most 10 malicious packets in 200 seconds.

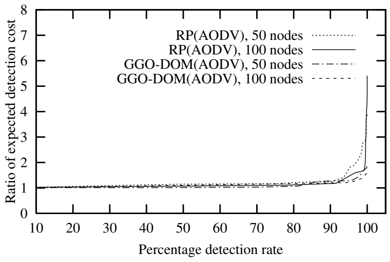
We first assume that insider nodes are static, and evaluate the performance of the approximation algorithms (subsections III-B and III-C) and the heuristics RP, GGO-DOM (subsection III-D). We assume that each insider node is IDS capable and has unit weight. We consider the maximum number of insider nodes that can be IDS active as a system parameter, and obtain approximate solutions of MDDOM_{IP} with L equaling different values of this parameter (subsection III-C). (Since MRDOM_{IP} is a special case of MDDOM_{IP}, we only consider MDDOM_{IP} here.) We refer to this solution as "DOM" in the figures. In RP and GGO-DOM, we select the probability of executing the IDS at each node such that the expected number of IDS active nodes equals the above system parameter. Using ns2-simulations, for each value of this parameter, we measure the expected detection costs and the percentage detection rates when the system selects the IDS active nodes as per DOM, GGO-DOM and RP in 200 topologies with static insiders and (a) a single mobile intruder launching an attack consisting of 5 packets on a single destination (figure 4) and (b) 5 mobile intruders colluding to launch an attack consisting of 10 packets on a single destination (figure 5). We plot the ratio between the expected detection costs of RP and DOM, and GGO-DOM and DOM as a function of the percentage detection rate in these figures. The simulations show that the same algorithm performs similarly for AODV and DSR. But, the performances of different algorithms can be quite different. For high detection rates, RP has significantly higher detection cost than DOM, and GGO-DOM has somewhat higher detection



(a) 100 nodes, $r = 250$ m

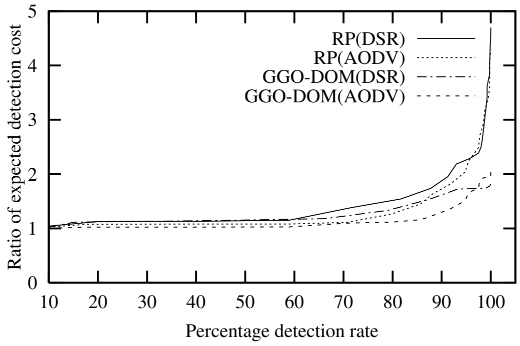


(b) 50 nodes, $r = 250$ m

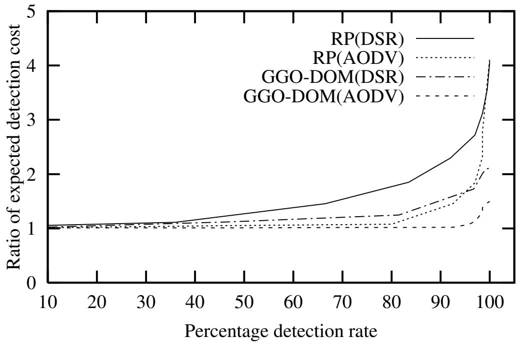


(c) $r = 150$ m

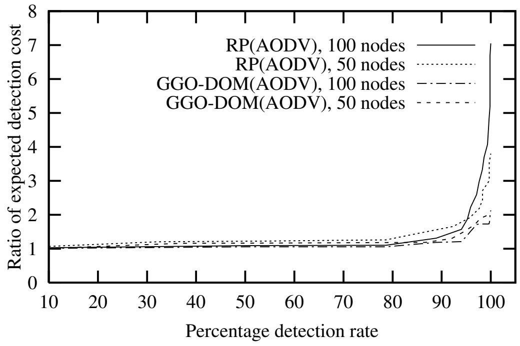
Fig. 4. We plot the ratio of the expected detection costs of RP and DOM, and GGO-DOM and DOM as a function of the percentage detection rate. We consider topologies where a mobile intruder attacks a single destination. The insider nodes are static. In figures (a) and (b), we consider both AODV and DSR routing protocols, transmission range (r) as 250m, and topologies with 100 and 50 insider nodes respectively. In figure (c), we consider AODV routing protocol, $r = 150$, and topologies with 100 and 50 insider nodes.



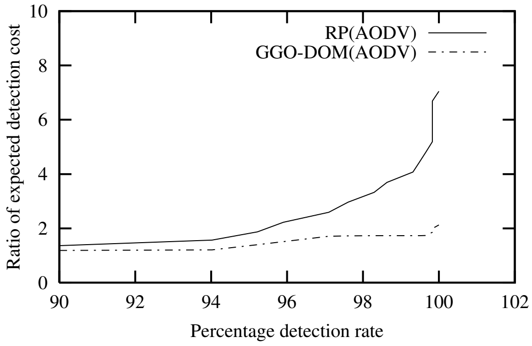
(a) 100 nodes, $r = 250$ m



(b) 50 nodes, $r = 250$ m

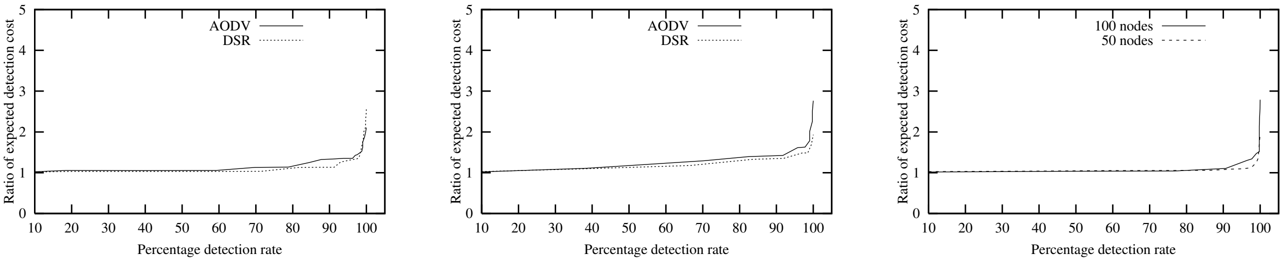


(c) $r = 150$ m



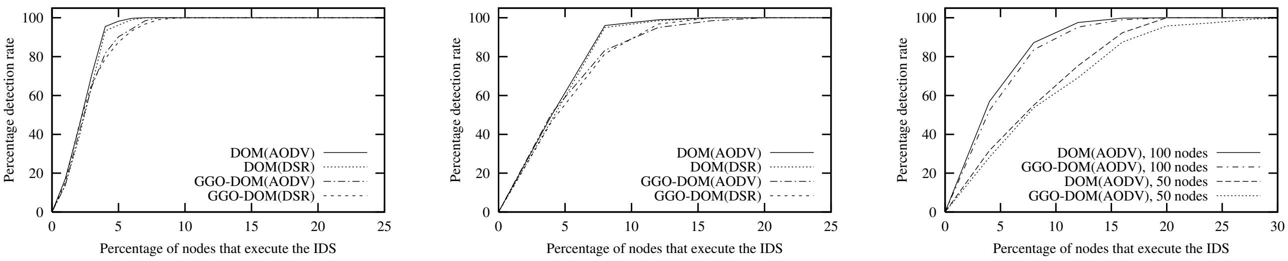
(d) 100 nodes, $r = 150$ m

Fig. 5. We plot the ratio of the expected detection costs of RP and DOM and GGO-DOM and DOM as a function of the percentage detection rate. We consider topologies with 5 mobile intruders colluding to attack a single destination. The insider nodes are static. In figures (a) and (b), we consider both AODV and DSR routing protocols, transmission range (r) as 250m, and topologies with 100 and 50 insider nodes respectively. In figure (c), we consider AODV routing protocol, $r = 150$m, and topologies with 100 and 50 insider nodes. In figure (d), we only consider detection rates greater than 90%.



(a) 100 nodes, $r = 250$ m (b) 50 nodes, $r = 250$ m (c) AODV, $r = 150$ m

Fig. 6. We plot the ratio of the expected detection costs of RP and GGO-DOM as a function of the percentage detection rate. We consider topologies with 5 mobile intruders colluding to attack a single destination. The insider nodes are mobile. In figures (a) and (b), we consider both AODV and DSR routing protocols, transmission range (r) as 250m, and topologies with 100 and 50 insider nodes respectively. In figure (c), we consider AODV routing protocol, transmission range (r) as 150m, and topologies with 100 and 50 insider nodes.



(a) 100 nodes, $r = 250$ m (b) 50 nodes, $r = 250$ m (c) $r = 150$ m

Fig. 7. We plot the percentage detection rates of DOM and GGO-DOM as function of the percentage of nodes executing the IDS. We consider topologies with 5 mobile intruders colluding to attack a single destination. In figures (a) and (b), we consider both AODV and DSR routing protocols, transmission range (r) as 250m, and topologies with 100 and 50 insider nodes respectively. In figure (c), we consider AODV routing protocol, $r = 150$m, and topologies with 100 and 50 insider nodes.

cost than DOM. But for low detection rates, RP, GGO-DOM and DOM have similar detection costs. The difference between the expected detection costs of GGO-DOM and DOM become noticeable only for very high detection rates (e.g., $> 95\%$). Thus, given its low detection cost, DOM is a clear choice when systems require very high detection rates (e.g., $> 95\%$). Given its simplicity, RP is a clear choice for systems that can accept low detection rates. GGO-DOM has intermediate computation and message exchange requirements, and should be used for medium to high detection rates. The thresholds for these "low, medium, high" detection rates can be decided on a case by case basis.

We now explain the observations for DOM and RP. For both DOM and RP, the detection cost and detection rate increase with increase in the number of IDS active nodes, but by different amounts for different detection rates. For both DOM and RP, the detection costs increase by equal amounts for equal increase in the number of IDS active nodes irrespective of the current value of the detection rate. Now, for DOM the total area covered by the IDS active nodes, and hence the detection rate increase by equal amounts for equal increase in the number of IDS active nodes irrespective of the current value of the detection rate. This is because DOM selects the IDS active nodes such that their coverage areas minimally overlap. This however does not hold for RP. Recall that RP selects the IDS active nodes uniformly in the entire set of insider nodes. Since in the simulations the insider nodes are uniformly distributed in a square, this is equivalent to selecting the IDS active nodes such that they are uniformly distributed in the square as well. When RP has a low detection rate, its IDS active nodes cover a small area. Now, since the IDS active nodes are uniformly distributed under RP and since the uncovered area is more than the covered area, a new IDS active node is more likely to be selected in the uncovered area. Thus, for low detection rates, similar to DOM, RP selects the IDS active nodes such that their coverage areas minimally overlap. Therefore, RP and DOM have similar increase in the coverage area and hence detection rates for the same increase in the number of IDS active nodes at this stage. Now, as argued before, both RP and DOM have similar increase in the detection costs for the same increase in the number of IDS active nodes. Thus, at low detection rates, RP and DOM have similar increase in the detection cost for the same increase in the detection rate. When RP has a high detection rate, its IDS active nodes together cover a large area. Now, since the IDS active nodes are uniformly distributed under RP and since the covered area is more than the uncovered area, a new IDS active node is more likely to be selected in the covered area. Thus, for high detection rates, unlike DOM, RP selects the IDS active nodes such that their coverage areas significantly overlap. Therefore, compared to DOM, RP has much smaller increase in coverage area and hence detection rates for the same increase in the number of IDS active nodes at this stage. Conversely, at high detection rates, for equal increase in detection rates, compared to DOM, RP needs to execute the IDS in many more nodes, which leads to higher increase in the detection cost. Thus, at high detection rates, RP has much larger increase in the detection cost for the same increase in the detection rate.

GGO-DOM and DOM select the IDS active nodes from those selected by GO-DOM and the approximate solution of $\mathrm{MDDOM_{IP}}$ respectively. The latter ensures less overlap among the coverage areas of the selected nodes than GO-DOM which does not have a provable approximation bound. Thus, GGO-DOM consumes some more resource than DOM.

Now, we consider mobile insider nodes, and compare the performance of GGO-DOM and RP (subsection III-D). Recall that the approximate solutions of $\mathrm{MRDOM_{IP}}$ (subsection III-B) and $\mathrm{MDDOM_{IP}}$ (subsection III-C) can not be used in this case as the solutions must be recomputed every time an insider node moves. We select the probabilities in GGO-DOM and RP just as for the static insider case. We consider topologies where 5 mobile intruders collude to launch an attack consisting of 10 packets on a single destination. We plot the ratio between the expected detection costs of RP and GGO-DOM as a function of the percentage detection rate in figure 6. The plots are similar to those for networks with static insiders, except that the difference between the resource consumption of RP and GGO-DOM is little lower than that between RP and DOM. We draw similar conclusions.

We now investigate the number of IDS active nodes selected by DOM and GGO-DOM for high detection rates. In figure 7 we plot the percentage detection rates of these algorithms as function of the expected number of IDS active nodes. We consider topologies with static insiders, and 5 mobile intruders launching an attack consisting of 10 malicious packets on a destination node. The figures show that both algorithms attain 90% and 100% detection rates when only 8% and 15% of the insider nodes execute the IDS respectively. Thus, high detection rates are obtained even when only a small fraction of the insider nodes operate in promiscuous mode. In most adhoc networks, at least a small number of nodes have significant energy. Thus, even for high detection rates, it would be sufficient to execute the IDS only in these nodes.

Finally, we comment on an approximation in the analytical model. Strictly speaking, all malicious packets need not be detected when the IDS active nodes constitute a dominating set in G'. Consider the following example adhoc network with 3 insider nodes A, B, C. All nodes are IDS capable. Hence, G and G' are the same. Let both A and C be B's neighbors. But, A and C are not each others neighbors. Let B, which constitutes the dominating set, execute the IDS. If A and C simultaneously relay malicious packets to outsider nodes, the packets collide at B, and therefore escape detection. Since the intruders transmit malicious packets only rarely, such simultaneous transmissions are rare, and have never occurred in our extensive simulations. Therefore, we have not considered them in the analysis.

V. Related Work

Ko *et al.* describe the challenges faced by conventional intrusion detection mechanisms when used in adhoc networks [13]. Misuse intrusion detection has been extensively investigated [3], [9]. We focus on the design tradeoffs involved in selecting the nodes that detect intrusion, rather than proposing or analyzing any misuse detection algorithm.

We now describe some related work in placing the detection modules. Ramanujam *et al.* [20] advocate the use of firewalls on every node with the firewalls being configured to contain the list of allowable packet flows. Like us, they require intermediate nodes to eavesdrop passively. Zhang *et al.* [25] also present a distributed intrusion detection and response framework for MANETs, where every node executes the IDS and responds to intrusion. Like us, they assume that the nodes cooperate. The disadvantage of both these schemes is that they consume significant energy and computational resource due to involvement of every node in the detection scheme. We present algorithms that maximize the detection performance while minimizing the resource consumption.

Kachirski *et al.* [12] describe a mobile agent based detection system for ad hoc networks based on anomaly detection technique. Here, only few nodes monitor the network traffic. These nodes are selected based on a voting scheme that considers the connectivity of the nodes. The authors only describe the architecture and propose to do more work to discover better detection algorithms and improve the robustness of the algorithm for selecting the monitoring nodes. Thus, the work seems to be in its infancy, though the basic idea is novel.

We now briefly describe a few other detection schemes that do not consider placement of IDS. Marti *et al.* [15] propose a cooperative routing scheme for avoiding transmitting packets through misbehaving nodes. Nodes promiscuously monitor traffic and cooperate so as to detect and report misbehavior to other nodes. Michiardi *et al.* [17] present the CORE mechanism in which reputation is used to enforce cooperation among nodes and prevent denial of service attacks. Buchegger *et al.* [22] propose the CONFIDANT scheme in which a node monitors its neighborhood to detect intrusion. When a node detects intrusion, it transmits alarm messages to other nodes in its friends list. Rao *et al.* [21] propose to detect intruders by observing node behavior. The idea is to estimate the congestion at intermediate nodes and decide if the intermediate node is not forwarding packets at the desired rate because of congestion or because of malicious behavior.

VI. Conclusion

We investigate the placement of the intrusion detection software for misuse detection in adhoc networks with multiple, mobile colluding intruders. Our goal is to maximize the detection performance while using limited computational resources. We mathematically formulate the problems of minimizing the resource consumption subject to attaining the maximum possible detection, and maximizing the detection subject to consuming no more than a certain amount of resource. We show that computing the optimal solution is NP-hard in each case. Thereafter, we propose algorithms that approximate the optimal solutions, and prove that these algorithms have guaranteeable approximation ratio. We demonstrate using simulation that these algorithms consume much less resource for attaining the same detection rate as compared to naive algorithms that randomly place the IDS. Our simulations also reveal that even for high detection rates ($90 - 100\%$), the optimal selection algorithms require only a small fraction of nodes to execute the IDS, which can therefore be those that are not limited in energy. Our investigation will be useful in designing intrusion detection systems, and evaluating the efficacy of misuse detection, which is widely used in wireline networks, in adhoc networks. Promising areas of future research are the design of efficient intrusion recovery mechanisms, protocols for cooperation among insider nodes that execute the IDS and schemes to detect attacks when the insider nodes are compromised.

REFERENCES

[1] http://www.microsoft.com/technet/security/bulletin/ms02-024.mspx. *Website*.

[2] Scott A. Vanstone Alfred J. Menezes, Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC press, 1999.

[3] F. Anjum, D. Subhadrabandhu, and S. Sarkar. Signature based intrusion detection for wireless ad-hoc networks: A comparative study of various routing protocols. In *IEEE VTC2003-Fall*, Oct. 2003.

[4] L. W. Bassham and T. Polk. Threat assessment of malicious code and human threats. *http://csrc.nist.gov/publications/nistir/threats/*.

[5] S. Capkun, M. Hamdi, and J. P. Hubaux. Gps-free positioning in mobile ad hoc networks. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, volume 9, 2001.

[6] W. Cheswic and W. Bellovin. *Firewalls and Internet Security*. Addison Wesley, 1999.

[7] Eric Cole. *Hackers Beware*. New Riding Publishing, 2001.

[8] D. Denning. An intrusion detection model. In *IEEE Transactions on Software Engineering*, volume SE-13, pages 222–232, Feb. 2001.

[9] T. Escamilla. *Intrusion Detection: Network Security Beyond the Firewall*. John Wiley and Sons, 1998.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 2000.

[11] S. Garfinkel and G. Spafford. *Practical UNIX and Internet Security*. O'Reilly and Associates, 2nd edition, 1996.

[12] O. Kachirski and R. Guha. Effective intrusion detection using multiple sensors in wireless ad hoc networks. In *36th Annual Hawaii International Conference on System Sciences (HICSS)*, Jan. 2003.

[13] C. Ko, P. Brutch, J. Rowe, G. Tsafnat, and K. Levitt. System health and intrusion monitoring using a hierarchy of constraints. In *4th International Symposium, Recent Advances in Intrusion Detection*, pages 190–204, Oct. 2001.

[14] F. Kuhn and R. Wattenhofer. Constant-time distributed dominating set approximation. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 25–32, 2003.

[15] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, 2000.

[16] John McHugh. Intrusion and intrusion detection. *International Journal of Information Security*, 2001.

[17] P. Michiardi and R. Molva. A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Comminication and Multimedia Security Conference)*, 2002.

[18] David Wagner Nikita Borisov, Ian Goldberg. ntercepting mobile communications: The insecurity of 802.11. In *Proceedings of MOBICOM 2001*, 2001.

[19] T. Ptacek and T. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, an SNI Technical Report, Jan. 1998.

[20] R. Ramanujan, S. Kudige, T. Nguyen, S. Takkella, and F. Adelstein. Intrusion-resistant ad hoc wireless networks. In *Proceedings of MILCOM*, Oct. 2002.

[21] R. Rao and G. Kesidis. Detection of malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited. In *Brazilian Journal of Telecomunications*, 2003.

[22] J.Y.L. Boudec S. Buchegger. Performance analysis of the confidant protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne*, June 2002.

[23] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream control transmission protocol. *RFC 2960*, Oct. 2000.

[24] D. Subhadrabandhu, S. Sarkar, and F. Anjum. Efficacy of misuse detection in adhoc networks. Technical report, University of Pennsylvania Technical Report, June 2004.

[25] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the Sixth International Conference on Mobile Computing and Networking (MobiCom)*, Aug. 2000.