Wireless/Mobile Network Security
Y. Xiao, X. Shen, and D.-Z. Du (Eds.) pp. 170 - 196

Chapter 7

A Survey on Intrusion Detection in Mobile Ad Hoc Networks

Tiranuch Anantvalee
Department of Computer Science and Engineering
Florida Atlantic University, Boca Raton, FL 33428
E-mail: tanantva@fau.edu

Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University, Boca Raton, FL 33428
E-mail: jie@cse.fau.edu

Abstract

In recent years, the use of mobile ad hoc networks (MANETs) has been widespread in many applications, including some mission critical applications, and as such security has become one of the major concerns in MANETs. Due to some unique characteristics of MANETs, prevention methods alone are not sufficient to make them secure; therefore, detection should be added as another defense before an attacker can breach the system. In general, the intrusion detection techniques for traditional wireless networks are not well suited for MANETs. In this paper, we classify the architectures for intrusion detection systems (IDS) that have been introduced for MANETs. Current IDS's corresponding to those architectures are also reviewed and compared. We then provide some directions for future research.

1 Introduction

A mobile ad hoc network (MANET) is a self-configuring network that is formed automatically by a collection of mobile nodes without the help of a fixed infrastructure or centralized management. Each node is equipped with a wireless transmitter and receiver, which allow it to communicate with other nodes in its radio communication range. In order for a node to forward a packet to a node that is out of its radio range, the cooperation of other nodes in the network is needed; this is known as multi-hop communication. Therefore, each node must act as both a host and a router at the same time. The network topology frequently changes due to the mobility of mobile nodes as they move within, move into, or move out of the network.

A MANET with the characteristics described above was originally developed for military purposes, as nodes are scattered across a battlefield and there is no infrastructure to help them form a network. In recent years, MANETs have been developing rapidly and are increasingly being used in many applications, ranging from military to civilian and commercial uses, since setting up such networks can be done without the help of any infrastructure or interaction with a human. Some examples are: search-and-rescue missions, data collection, and virtual classrooms and conferences where laptops, PDA or other mobile devices share wireless medium and communicate to each other. As MANETs become widely used, the security issue has become one of the primary concerns. For example, most of the routing protocols proposed for MANETs assume that every node in the network is cooperative and not malicious [1]. Therefore, only one compromised node can cause the failure of the entire network.

There are both passive and active attacks in MANETs. For passive attacks, packets containing secret information might be eavesdropped, which violates confidentiality. Active attacks, including injecting packets to invalid destinations into the network, deleting packets, modifying the contents of packets, and impersonating other nodes violate availability, integrity, authentication, and non-repudiation. Proactive approaches such as cryptography and authentication [10, 11, 12, 13] were first brought into consideration, and many techniques have been proposed and implemented. However, these applications are not sufficient. If we have the ability to detect the attack once it comes into the network, we can stop it from doing any damage to the system or any data. Here is where the intrusion detection system comes in.

Intrusion detection can be defined as a process of monitoring activities in a system, which can be a computer or network system. The mechanism by

which this is achieved is called an intrusion detection system (IDS). An IDS collects activity information and then analyzes it to determine whether there are any activities that violate the security rules. Once an IDS determines that an unusual activity or an activity that is known to be an attack occurs, it then generates an alarm to alert the security administrator. In addition, IDS can also initiate a proper response to the malicious activity.

Although there are several intrusion detection techniques developed for wired networks today, they are not suitable for wireless networks due to the differences in their characteristics. Therefore, those techniques must be modified or new techniques must be developed to make intrusion detection work effectively in MANETs.

In this paper, we classify the architectures for IDS in MANETs, each of which is suitable for different network infrastructures. Current intrusion detection systems corresponding to those architectures are reviewed and compared.

The rest of the paper is structured as follows. Section 2 describes the background on intrusion detection systems. Intrusion detection in MANETs - how it differs from intrusion detection in wired networks - is also presented in this section. In Section 3, architectures that have been introduced for IDS in MANETs are presented. Some of current intrusion detection systems for MANETs are given in Section 4. Then, some of the intrusion detection techniques for node cooperation are reviewed and compared in Section 5. Finally, the conclusion and future directions are given in Section 6.

2 Background

2.1 Intrusion Detection System (IDS)

Many historical events have shown that intrusion prevention techniques alone, such as encryption and authentication, which are usually a first line of defense, are not sufficient. As the system become more complex, there are also more weaknesses, which lead to more security problems. Intrusion detection can be used as a second wall of defense to protect the network from such problems. If the intrusion is detected, a response can be initiated to prevent or minimize damage to the system.

Some assumptions are made in order for intrusion detection systems to work [1]. The first assumption is that user and program activities are observable. The second assumption, which is more important, is that normal and intrusive activities must have distinct behaviors, as intrusion detection

must capture and analyze system activity to determine if the system is under attack.

Intrusion detection can be classified based on audit data as either host-based or network-based. A network-based IDS captures and analyzes packets from network traffic while a host-based IDS uses operating system or application logs in its analysis. Based on detection techniques, IDS can also be classified into three categories as follows [2].

- *Anomaly detection systems*: The normal profiles (or normal behaviors) of users are kept in the system. The system compares the captured data with these profiles, and then treats any activity that deviates from the baseline as a possible intrusion by informing system administrators or initializing a proper response.

- *Misuse detection systems*: The system keeps patterns (or signatures) of known attacks and uses them to compare with the captured data. Any matched pattern is treated as an intrusion. Like a virus detection system, it cannot detect new kinds of attacks.

- *Specification-based detection*: The system defines a set of constraints that describe the correct operation of a program or protocol. Then, it monitors the execution of the program with respect to the defined constraints.

2.2 Intrusion Detection in MANETs

Many intrusion detection systems have been proposed in traditional wired networks, where all traffic must go through switches, routers, or gateways. Hence, IDS can be added to and implemented in these devices easily [17, 18]. On the other hand, MANETs do not have such devices. Moreover, the medium is wide open, so both legitimate and malicious users can access it. Furthermore, there is no clear separation between normal and unusual activities in a mobile environment. Since nodes can move arbitrarily, false routing information could be from a compromised node or a node that has outdated information. Thus, the current IDS techniques on wired networks cannot be applied directly to MANETs. Many intrusion detection systems have been proposed to suit the characteristics of MANETs, some of which will be discussed in the next sections.

3 Architectures for IDS in MANETs

The network infrastructures that MANETs can be configured to are either flat or multi-layer, depending on the applications. Therefore, the optimal IDS architecture for a MANET may depend on the network infrastructure itself [9]. In a flat network infrastructure, all nodes are considered equal, thus it may be suitable for applications such as virtual classrooms or conferences. On the contrary, some nodes are considered different in the multi-layered network infrastructure. Nodes may be partitioned into clusters with one clusterhead for each cluster. To communicate within the cluster, nodes can communicate directly. However, communication across the clusters must be done through the clusterhead. This infrastructure might be well suited for military applications.

3.1 Stand-alone Intrusion Detection Systems

In this architecture, an intrusion detection system is run on each node independently to determine intrusions. Every decision made is based only on information collected at its own node, since there is no cooperation among nodes in the network. Therefore, no data is exchanged. Besides, nodes in the same network do not know anything about the situation on other nodes in the network as no alert information is passed. Although this architecture is not effective due to its limitations, it may be suitable in a network where not all nodes are capable of running an IDS or have an IDS installed. This architecture is also more suitable for flat network infrastructure than for multi-layered network infrastructure. Since information on each individual node might not be enough to detect intrusions, this architecture has not been chosen in most of the IDS for MANETs.

3.2 Distributed and Cooperative Intrusion Detection Systems

Since the nature of MANETs is distributed and requires cooperation of other nodes, Zhang and Lee [1] have proposed that the intrusion detection and response system in MANETs should also be both distributed and cooperative as shown in Figure 1. Every node participates in intrusion detection and response by having an IDS agent running on them. An IDS agent is responsible for detecting and collecting local events and data to identify possible intrusions, as well as initiating a response independently. However, neighboring IDS agents cooperatively participate in global intrusion detection

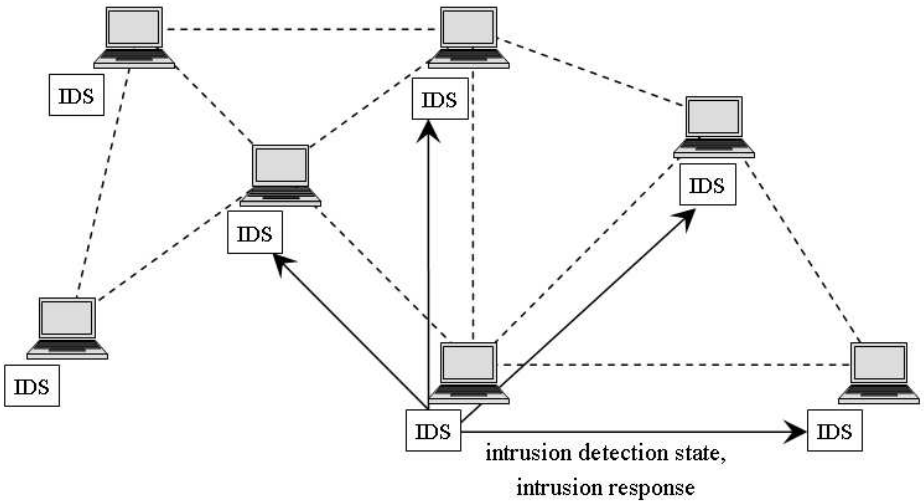

Figure 1: Distributed and Cooperative IDS in MANETs proposed by Zhang and Lee [1]

actions when the evidence is inconclusive. Similarly to stand-alone IDS architecture, this architecture is more suitable for flat network infrastructure, not multi-layered one.

3.3 Hierarchical Intrusion Detection Systems

Hierarchical IDS architectures extend the distributed and cooperative IDS architectures and have been proposed for multi-layered network infrastructures where the network is divided into clusters. Clusterheads of each cluster usually have more functionality than other members in the clusters, for example routing packets across clusters. Thus, these clusterheads, in some sense, act as control points which are similar to switches, routers, or gateways in wired networks. The same concept of multi-layering is applied to intrusion detection systems where hierarchical IDS architecture is proposed. Each IDS agent is run on every member node and is responsible locally for its node, i.e., monitoring and deciding on locally detected intrusions. A clusterhead is responsible locally for its node as well as globally for its cluster, e.g. monitoring network packets and initiating a global response when network intrusion is detected.

3.4 Mobile Agent for Intrusion Detection Systems

A concept of mobile agents has been used in several techniques for intrusion detection systems in MANETs. Due to its ability to move through the large network, each mobile agent is assigned to perform only one specific task, and then one or more mobile agents are distributed into each node in the network. This allows the distribution of the intrusion detection tasks.

There are several advantages for using mobile agents [2]. Some functions are not assigned to every node; thus, it helps to reduce the consumption of power, which is scarce in mobile ad hoc networks. It also provides fault tolerance such that if the network is partitioned or some agents are destroyed, they are still able to work. Moreover, they are scalable in large and varied system environments, as mobile agents tend to be independent of platform architectures. However, these systems would require a secure module where mobile agents can be stationed to. Additionally, mobile agents must be able to protect themselves from the secure modules on remote hosts as well.

Mobile-agent-based IDS can be considered as a distributed and cooperative intrusion detection technique as described in Section 3.2. Moreover, some techniques also use mobile agents combined with hierarchical IDS, for example, what will be described in Section 4.3.

4 Sample Intrusion Detection Systems for MANETs

Since the IDS for traditional wired systems are not well-suited to MANETs, many researchers have proposed several IDS especially for MANETs, which some of them will be reviewed in this section.

4.1 Distributed and Cooperative IDS

As described in Section 3.2, Zhang and Lee also proposed the model for a distributed and cooperative IDS as shown in Figure 2 [1].

The model for an IDS agent is structured into six modules. The *local data collection* module collects real-time audit data, which includes system and user activities within its radio range. This collected data will be analyzed by the *local detection engine* module for evidence of anomalies. If an anomaly is detected with strong evidence, the IDS agent can determine independently that the system is under attack and initiate a response through the *local response* module (i.e., alerting the local user) or the *global response* module (i.e., deciding on an action), depending on the type of intrusion, the type of network protocols and applications, and the certainty of the evidence. If an

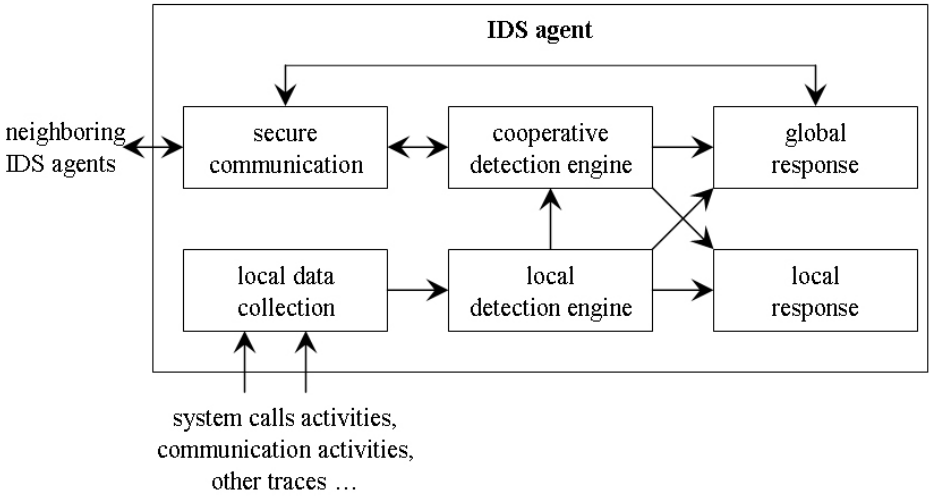

Figure 2: A Model for an IDS Agent [1]

anomaly is detected with weak or inconclusive evidence, the IDS agent can request the cooperation of neighboring IDS agents through a *cooperative detection engine* module, which communicates to other agents through a *secure communication* module.

4.2 Local Intrusion Detection System (LIDS)

Albers *et al.* [3] proposed a distributed and collaborative architecture of IDS by using mobile agents. A Local Intrusion Detection System (LIDS) is implemented on every node for local concern, which can be extended for global concern by cooperating with other LIDS. Two types of data are exchanged among LIDS: security data (to obtain complementary information from collaborating nodes) and intrusion alerts (to inform others of locally detected intrusion). In order to analyze the possible intrusion, data must be obtained from what the LIDS detects, along with additional information from other nodes. Other LIDS might be run on different operating systems or use data from different activities such as system, application, or network activities; therefore, the format of this raw data might be different, which makes it hard for LIDS to analyze. However, such difficulties can be solved by using SNMP (Simple Network Management Protocol) data located in MIBs (Management Information Base) as an audit data source. Such a data source not only eliminates those difficulties, but also reduces the in-

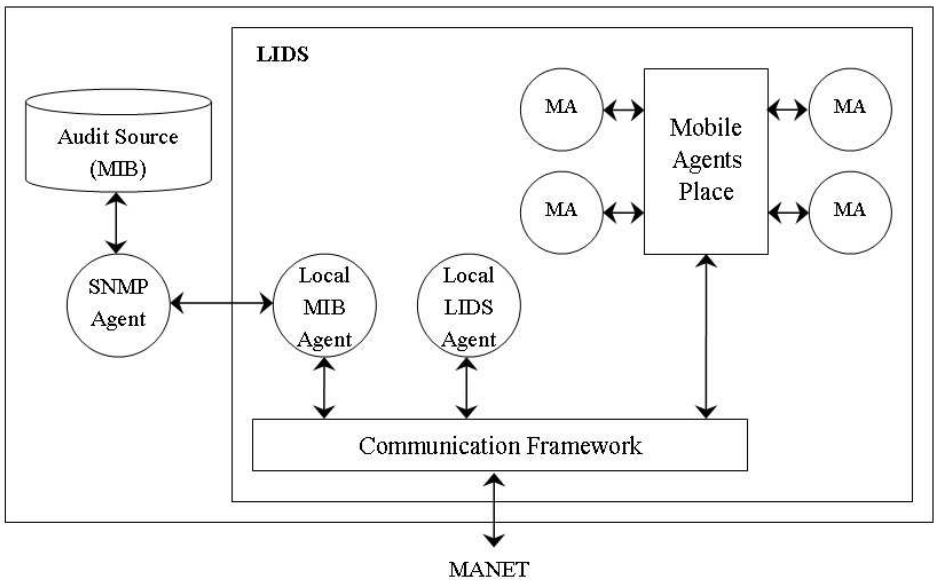

Figure 3: LIDS Architecture in A Mobile Node [3]

crease in using additional resources to collect audit data if an SNMP agent is already run on each node.

To obtain additional information from other nodes, the authors proposed mobile agents to be used to transport SNMP requests to other nodes. In another words, to distribute the intrusion detection tasks. The idea differs from traditional SNMP in that the traditional approach transfers data to the requesting node for computation while this approach brings the code to the data on the requested node. This is motivated by the unreliability of UDP messages used in SNMP and the dynamic topology of MANETs. As a result, the amount of exchanged data is tremendously reduced. Each mobile agent can be assigned a specific task which will be achieved in an autonomous and asynchronous fashion without any help from its LIDS.

The LIDS architecture is shown in Figure 3, which consists of

- **Communication Framework**: To facilitate for both internal and external communication with a LIDS.
- **Local LIDS Agent**: To be responsible for local intrusion detection and local response. Also, it reacts to intrusion alerts sent from other nodes to protect itself against this intrusion.
- **Local MIB Agent**: To provide a means of collecting MIB variables

for either mobile agents or the Local LIDS Agent. Local MIB Agent acts as an interface with SNMP agent, if SNMP exists and runs on the node, or with a tailor-made agent developed specifically to allow updates and retrievals of the MIB variables used by intrusion detection, if none exists.

- **Mobile Agents (MA)**: They are distributed from its LID to collect and process data on other nodes. The results from their evaluation are then either sent back to their LIDS or sent to another node for further investigation.
- **Mobile Agents Place**: To provide a security control to mobile agents.

For the methodology of detection, Local IDS Agent can use either anomaly or misuse detection. However, the combination of two mechanisms will offer the better model. Once the local intrusion is detected, the LIDS initiates a response and informs the other nodes in the network. Upon receiving an alert, the LIDS can protect itself against the intrusion.

4.3 Distributed Intrusion Detection System Using Multiple Sensors

Kachirski and Guha [4] proposed a multi-sensor intrusion detection system based on mobile agent technology. The system can be divided into three main modules, each of which represents a mobile agent with certain functionality: monitoring, decision-making or initiating a response. By separating functional tasks into categories and assigning each task to a different agent, the workload is distributed which is suitable for the characteristics of MANETs. In addition, the hierarchical structure of agents is also developed in this intrusion detection system as shown in Figure 4.

- **Monitoring agent**: Two functions are carried out at this class of agent: network monitoring and host monitoring. A host-based monitor agent hosting system-level sensors and user-activity sensors is run on every node to monitor within the node, while a monitor agent with a network monitoring sensor is run only on some selected nodes to monitor at packet-level to capture packets going through the network within its radio ranges.
- **Action agent**: Every node also hosts this action agent. Since every node hosts a host-based monitoring agent, it can determine if there is

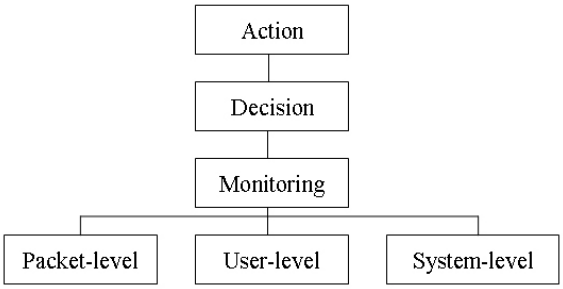

Figure 4: Layered Mobile Agent Architecture proposed by Kachirski and Guha [4]

any suspicious or unusual activities on the host node based on anomaly detection. When there is strong evidence supporting the anomaly detected, this action agent can initiate a response, such as terminating the process or blocking a user from the network.

- **Decision agent**: The decision agent is run only on certain nodes, mostly those nodes that run network monitoring agents. These nodes collect all packets within its radio range and analyze them to determine whether the network is under attack. Moreover, from the previous paragraph, if the local detection agent cannot make a decision on its own due to insufficient evidence, its local detection agent reports to this decision agent in order to investigate further. This is done by using packet-monitoring results that comes from the network-monitoring sensor that is running locally. If the decision agent concludes that the node is malicious, the action module of the agent running on that node as described above will carry out the response.

The network is logically divided into clusters with a single clusterhead for each cluster. This clusterhead will monitor the packets within the cluster and only packets whose originators are in the same cluster are captured and investigated. This means that the network monitoring agent (with network monitoring sensor) and the decision agent are run on the clusterhead.

In this mechanism, the decision agent performs the decision-making based on its own collected information from its network-monitoring sensor; thus, other nodes have no influence on its decision. This way, spoofing attacks and false accusations can be prevented.

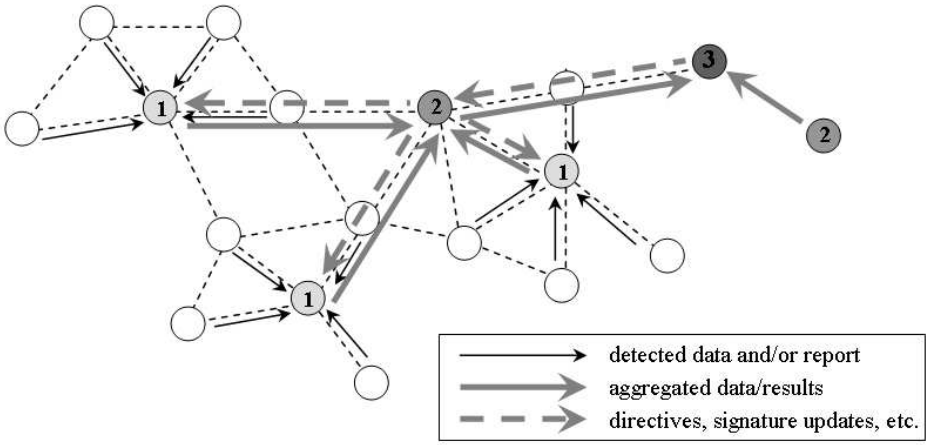

Figure 5: Dynamic Intrusion Detection Hierarchy [16]

4.4 Dynamic Hierarchical Intrusion Detection Architecture

Since nodes move arbitrarily across the network, a static hierarchy is not suitable for such dynamic network topology. Sterne *et al.* [16] proposed a dynamic intrusion detection hierarchy that is potentially scalable to large networks by using clustering like those in Section 4.3 and 5.5. However, it can be structured in more than two levels as shown in Figure 5. Nodes labeled "1" are the first level clusterheads while nodes labeled "2" are the second level clusterheads and so on. Members of the first level of the cluster are called leaf nodes.

Every node has the responsibilities of monitoring (by accumulating counts and statistics), logging, analyzing (i.e., attack signature matching or checking on packet headers and payloads), responding to intrusions detected if there is enough evidence, and alerting or reporting to clusterheads. Clusterheads, in addition, must also perform:

- **Data fusion/integration and data reduction**: Clusterheads aggregate and correlate reports from members of the cluster and data of their own. Data reduction may be involved to avoid conflicting data, bogus data and overlapping reports. Besides, clusterheads may send the requests to their children for additional information in order to correlate reports correctly.

- **Intrusion detection computations**: Since different attacks require different sets of detected data, data on a single node might not be able

to detect the attack, e.g., DDoS attack, and thus clusterheads also analyze the consolidated data before passing to upper levels.

- **Security Management**: The uppermost levels of the hierarchy have the authority and responsibility for managing the detection and response capabilities of the clusters and clusterheads below them. They may send the signatures update, or directives and policies to alter the configurations for intrusion detection and response. These update and directives will flow from the top of the hierarchy to the bottom.

To form the hierarchical structure, every node uses clustering, which is typically used in MANETs to construct routes, to self-organize into local neighborhoods (first level clusters) and then select neighborhood representatives (clusterheads). These representatives then use clustering to organize themselves into the second level and select the representatives. This process continues until all nodes in the network are part of the hierarchy. The authors also suggested criteria on selecting clusterheads. Some of these criteria are:

- *Connectivity*: the number of nodes within one hop
- *Proximity*: members should be within one hop of its clusterhead
- *Resistance to compromise (hardening)*: the probability that the node will not be compromised. This is very important for the upper level clusterheads.
- *Processing power, storage capacity, energy remaining, bandwidth capabilities*

Additionally, this proposed architecture does not rely solely on promiscuous node monitoring like many proposed architectures, due to its unreliability as described in [5]. Therefore, this architecture also supports direct periodic reporting where packet counts and statistics are sent to monitoring nodes periodically.

4.5 Zone-Based Intrusion Detection System (ZBIDS)

Sun et al. [24] has proposed an anomaly-based two-level nonoverlapping Zone-Based Intrusion Detection System (ZBIDS). By dividing the network in Figure 6 into nonoverlapping zones (zone A to zone I), nodes can be categorized into two types: the intrazone node and the interzone node (or a gateway node). Considering only zone E, node 5, 9, 10 and 11 are intrazone

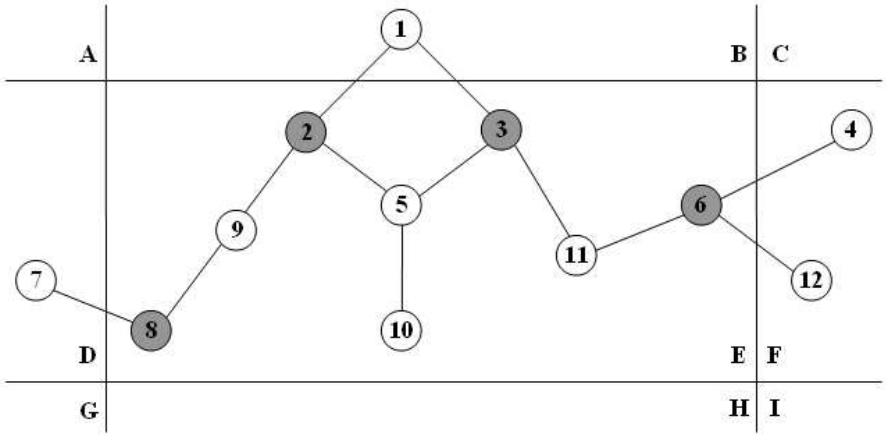

Figure 6: ZBIDS for MANETs [24]

nodes, while node 2, 3, 6, and 8 are interzone nodes which have physical connections to nodes in other zones. The formation and maintenance of zones requires each node to know its own physical location and to map its location to a zone map, which requires prior design setup.

Each node has an IDS agent run on it which the model of the agent is shown in Figure 7. Similar to an IDS agent proposed by Zhang and Lee (Figure 2), the *data collection module* and the *detection engine* are responsible for collecting local audit data (for instance, system call activities, and system log files) and analyzing collected data for any sign of intrusion respectively. In addition, there may be more than one for each of these modules which allows collecting data from various sources and using different detection techniques to improve the detection performance. The *local aggregation and correlation (LACE)* module is responsible for combining the results of these local detection engines and generating alerts if any abnormal behavior is detected. These alerts are broadcasted to other nodes within the same zone. However, for the *global aggregation and correlation (GACE)*, its functionality depends on the type of the node. As described in Figure 7, if the node is an intrazone node, it only sends the generated alerts to the interzone nodes. Whereas, if the node is an interzone node, it receives alerts from other intrazone nodes, aggregates and correlates those alerts with its own alerts, and then generates alarms. Moreover, the GACE also cooperates with the GACEs of the neighboring interzone nodes to have more accurate information to detect the intrusion. Lastly, the *intrusion response* module

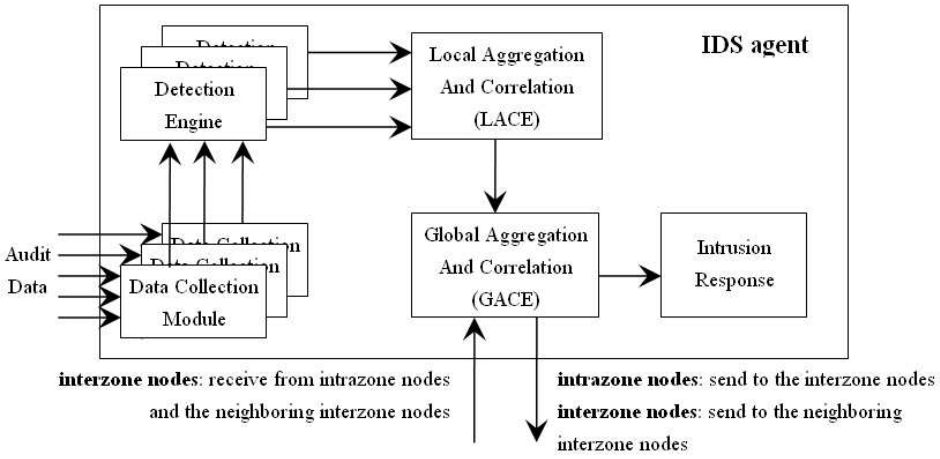

Figure 7: An IDS agent in ZBIDS [24]

is responsible for handling the alarms generated from the GACE.

The local aggregation and correlation algorithm used in ZBIDS is based on a local Markov chain anomaly detection. An IDS agent first creates a normal profile by constructing a Markov chain from the routing cache. A valid change in the routing cache can be characterized by the Markov chain detection model with probabilities, otherwise, it's considered abnormal, and the alert will be generated. For the global aggregation and correlation algorithm, it's based on information provided in the received alerts containing the type, the time, and the source of the attacks.

5 Intrusion Detection Techniques for Node Cooperation in MANETs

Since there is no infrastructure in mobile ad hoc networks, each node must rely on other nodes for cooperation in routing and forwarding packets to the destination. Intermediate nodes might agree to forward the packets but actually drop or modify them because they are misbehaving. The simulations in [5] show that only a few misbehaving nodes can degrade the performance of the entire system. There are several proposed techniques and protocols to detect such misbehavior in order to avoid those nodes, and some schemes also propose punishment as well [6, 7].

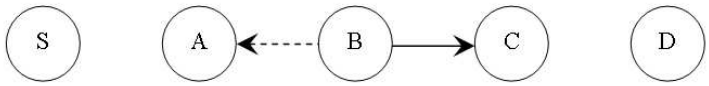

Figure 8: How watchdog works: Although node B intends to transmit a packet to node C, node A could overhear this transmission

5.1 Watchdog and Pathrater

Two techniques were proposed by Marti, Giuli, and Baker [5], watchdog and pathrater, to be added on top of the standard routing protocol in ad hoc networks. The standard is Dynamic Source Routing protocol (DSR) [8]. A watchdog identifies the misbehaving nodes by eavesdropping on the transmission of the next hop. A pathrater then helps to find the routes that do not contain those nodes.

In DSR, the routing information is defined at the source node. This routing information is passed together with the message through intermediate nodes until it reaches the destination. Therefore, each intermediate node in the path should know who the next hop node is. In addition, listening to the next hop's transmission is possible because of the characteristic of wireless networks - if node A is within range of node B, A can overhear communication to and from B.

Figure 8 shows how the watchdog works. Assume that node S wants to send a packet to node D, which there exists a path from S to D through nodes A, B, and C. Consider now that A has already received a packet from S destined to D. The packet contains a message and routing information. When A forwards this packet to B, A also keeps a copy of the packet in its buffer. Then, it promiscuously listens to the transmission of B to make sure that B forwards to C. If the packet overheard from B (represented by a dashed line) matches that stored in the buffer, it means that B really forwards to the next hop (represented as a solid line). It then removes the packet from the buffer. However, if there's no matched packet after a certain time, the watchdog increments the failures counter for node B. If this counter exceeds the threshold, A concludes that B is misbehaving and reports to the source node S.

Pathrater performs the calculation of the"path metric" for each path. By keeping the rating of every node in the network that it knows, the path metric can be calculated by combining the node rating together with link reliability, which is collected from past experience. Obtaining the path metric for all available paths, the pathrater can choose the path with the highest

metric. In addition, if there is no such link reliability information, the path metric enables the pathrater to select the shortest path too. As a result, paths containing misbehaving nodes will be avoided.

From the result of the simulation, the system with these two techniques is quite effective for choosing paths to avoid misbehaving nodes. However, those misbehaving nodes are not punished. In contrast, they even benefit from the network. In another word, they can use resources of the network - other nodes forward packets for them, while they forward packets for no one, which save their own resources. Therefore, misbehaving nodes are encouraged to continue their behaviors.

5.2 CONFIDANT

Buchegger and LeBoudec [6] proposed an extension to DSR protocol called CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks), which is similar to Watchdog and Pathrater. Each node observes the behaviors of neighbor nodes within its radio range and learns from them. This system also solves the problem of Watchdog and Pathrater such that misbehavior nodes are punished by not including them in routing and not helping them on forwarding packets. Moreover, when a node experiences a misbehaving node, it will send a warning message to other nodes in the network, defined as friends, which is based on trusted relationship.

Figure 9 shows the components of the CONFIDANT protocol, which are the Monitor, the Trust Manager, the Reputation System, and the Path Manager. The process of how they work can be divided into two parts: the process to handle its own observations and the process to handle reports from trusted nodes.

- From observations: The monitor uses a "neighborhood watch" to detect any malicious behaviors with in its radio range, i.e., no forwarding, unusually frequent route update, etc. (This is similar to the watchdog in the previous scheme) If a suspicious event is detected, the monitor then reports to the reputation system. At this point, the reputation system performs several checks and updates the rating of the reported node in the reputation table. If the rating result is unacceptable, it passes the information to the path manager, which then removes all paths containing the misbehavior node. An ALARM message is also sent by the trust manager to warn other nodes that it considers as friends.

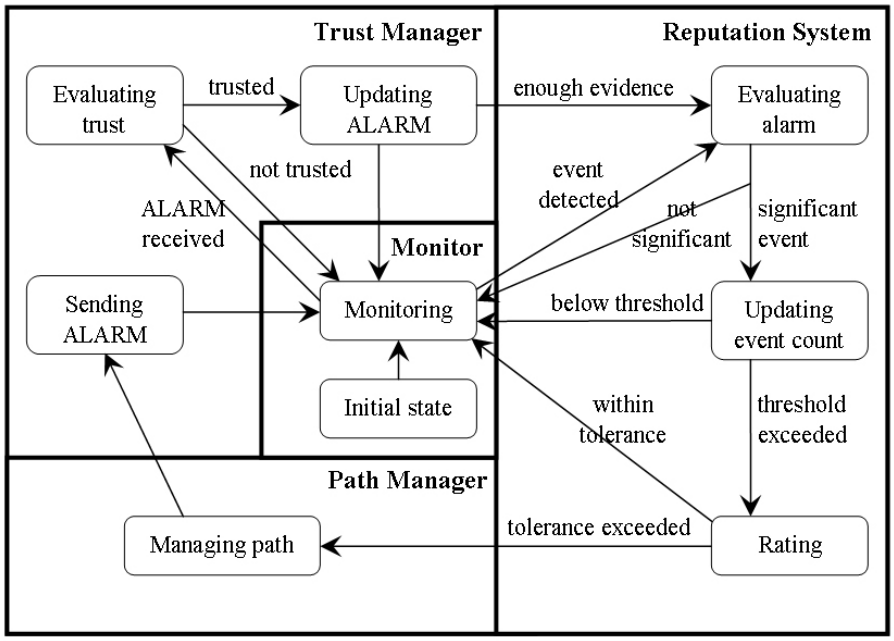

Figure 9: Components and State Diagram of CONFIDANT Protocol [6]

- From trusted nodes: When the monitor receives an ALARM message from its friends, the message will first be evaluated by the trust manager for the trustworthiness of the source node. If the message is trustworthy, this ALARM message, together with the level of trust, will be stored in the alarm table. All ALARM messages of the reported node will then be combined to see if there is enough evidence to identify that it is malicious. If so, the information will be sent to the reputation system, which then performs the same functions as described in the previous paragraph.

Since this protocol allows nodes in the network to send alarm messages to each other, it could give more opportunities for attackers to send false alarm messages that a node is misbehaving while it's actually not. This is one form of denial of service attacks.

5.3 CORE

Michiardi and Molva [7] presented a technique to detect a specific type of misbehaving nodes, which are selfish nodes, and also force them to cooper-

ate. Similar to those in Section 5.1 and 5.2, this technique is based on a monitoring system and a reputation system, which includes both direct and indirect reputation from the system as will be described shortly.

As nodes sometimes do not intentionally misbehave, i.e., battery condition is low, these nodes should not be considered as misbehaving nodes and excluded from the network. To do this, the reputation should be rated based on past reputation, which is zero (neutral) at the beginning. In addition, participation in the network can be categorized into several functions such as routing discovery (in DSR) or forwarding packets. Each of these activities has different level of effects to the network; for example, forwarding packets has more effect on the performance of the system than that of routing discovery. Therefore, significance weight of functions should be used in the calculation of the reputation.

Like CONFIDANT, each node can receive a report from other nodes. However, the difference is CORE allows only positive reports to be passed while negative reports are passed in CONFIDANT. In another word, CORE prevents false accusation, thus, it also prevents a denial of service attack, which cannot be done in CONFIDANT. The negative rating is given to a node only from the direct observation when the node does not cooperate, which results in the decreased reputation for that node. The positive rating, in contrast, is given from both direct observation and positive reports from other nodes, which results in the increased reputation.

CORE can then be said to have two components, the watchdog system and the reputation system. The watchdog modules, one for each function, work the same way as in the previous two schemes above. For the reputation system, it maintains several reputation tables, one for each function and one for accumulated values for each node. Therefore, if there is a request from a bad reputation node (the overall reputation is negative), the node will be rejected and not be able to use the network.

5.4 OCEAN

Bansal and Baker [19] also proposed an extension on top of the DSR protocol called OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks). OCEAN also uses a monitoring system and a reputation system. However, in contrast to the previous approaches above, OCEAN relies only on its own observation to avoid the new vulnerability of false accusation from second-hand reputation exchanges. Therefore, OCEAN can be considered as a stand-alone architecture.

OCEAN categorizes routing misbehavior into two types: misleading and

selfish. If a node has participated in the route discovery but not packet forwarding, this is considered to be misleading as it misleads other nodes to route packets through it. But if a node does not even participate in the route discovery, it is considered to be selfish.

In order to detect and mitigate the misleading routing behaviors, after a node forwards a packet to a neighbor, it buffers the packet checksum and monitors if the neighbor attempts to forward the packet within a given time. Then, a negative or positive event is given as the result of the monitoring to update the neighbor rating. If the rating falls below the faulty threshold, that neighbor node is added to a faulty list which will be added in the RREQ as an avoid-list. In addition, all traffic from the faulty neighbor node will be rejected. Nonetheless, the faulty timeout is used to allow the faulty node to join back to the network in case that it might be false accused or it behaves better.

Each node also has a mechanism of maintaining chipcounts for each neighbor to mitigate the selfish behavior. A neighbor node earns chips when forwarding a packet on behalf of the node and loses ships when asking the node to forward a packet. If the chipcount of the neighbor is below the threshold, packets coming from that neighbor will be denied.

5.5 Cooperative Intrusion Detection System

A cluster-based cooperative intrusion detection system, similar to Kachirski and Guha's system [4], has been presented by Huang and Lee [14]. In this approach, an IDS is not only able to detect an intrusion, but also to identify the attack type and the attacker, whenever possible, through statistical anomaly detection. Various types of statistics (or features), which are proposed in their previous work [15], are evaluated from a sampling period by capturing the basic view of network topology and routing operations, as well as traffic patterns and statistics, in the normal traffic. Hence, attacks could be identified if the statistics deviate from the pre-computed ones (anomaly detection).

Statistics can be categorized into two categories, non traffic-related and traffic-related. Non traffic-related statistics are calculated based on the mobility and the trace log files, which can be done separately on each node. Some of these statistics are route add count, route removal count, total route change, average route length, etc. Traffic-related statistics are involved in routing and packet forwarding and can be calculated by counting packets going in and out, e.g. the number of packet received, the number of packet forwarded, the number of route reply messages, etc. These statistics can

be captured by the node itself or the neighboring nodes who overhear the transmission.

Several identification rules are pre-defined for known attacks by using relationships of the mentioned statistics. Once an anomaly is detected, the IDS will perform further investigation to determine the detailed information of the attack from a set of these identification rules. These rules enhance the system to identify the type of the attack and, in some cases, the attacking node. Some notations of statistics are presented as follows. Let M represent the monitoring node and m represent the monitored node.

- $\#(*, m)$: the number of incoming packets on the monitored node m.
- $\#(*, [m])$: the number of incoming packets of which the monitored node m is the destination.
- $\#(m, *)$: the number of outgoing packets from the monitored node m.
- $\#([m], *)$: the number of outgoing packets of which the monitored node m is the source.
- $\#(m, n)$: the number of outgoing packets from m of which n is the next hop.
- $\#([s], M, m)$: the number of packets that are originated from s and transmitted from M to m.
- $\#([s], [d])$: the number of packets received on m which is originated from s and destined to d.
- $\#(*, m)(TYPE = RREQ)$: the number of incoming RREQ packets on m.

These statistics are computed over a long period L. Let $FEATURE^L$ represents the aggregated $FEATURE$ over time L. Some identification rules are defined for well known attacks as follows.

- ***Unconditional Packet Dropping***:This rule uses Forward Percentage (FP) over a period L to define the attack.

$$FP_m = \frac{\text{packets actually forwarded}}{\text{packets to be forwarded}} = \frac{\#^L(m, M) \ - \ \#^L([m], M)}{\#^L(M, m) \ - \ \#^L(M, [m])}$$

  If there are packets to be forwarded (denominator is not zero) and $FP_m = 0$, the unconditional packet dropping attack is detected and the attacker is m.

- ***Random Packet Dropping***: This rule also uses the same FP as unconditional packet dropping. However, the threshold ϵ_{FP} is defined ($\epsilon_{FP} < 1$). If $0 < FP_m < \epsilon_{FP}$, m is defined as an attacker using random packet dropping.

- ***Selective Packet Dropping***: This rule uses Local Forward Percentage (LFP) for each source s.

$$LFP_m^s = \frac{\text{packets from source } s \text{ actually forwarded}}{\text{packets from source } s \text{ to be forwarded}} = \frac{\#^L([s], m, M)}{\#^L([s], M, m) \ - \ \#^L([s], M, [m])}$$

  If the denominator is not zero and $LFP_m^s = 0$, the attack is the unconditional packet dropping targeted at s. However, if LFP_m^s is less than the threshold ($\epsilon_{LFP} < 1$), the attack is detected as random packet dropping targeted at s.

- ***Blackhole***: This rule uses Global Forward Percentage (GFP) and it must be computed on M locally because the rule relies on information available only on the node. Let $N(M)$ denote M's 1-hop neighbors.

$$GFP_m^s = \frac{\text{packets to be forwarded}}{\text{packets from} N(M) \text{destined to other nodes than itself or another} N(M)} = \frac{\#^L(*, M) \ - \ \#^L(*, [M])}{\sum\limits_{i \epsilon N(M)} \#^L(i, M) \ - \ \sum\limits_{i,j \epsilon N(M)} \#^L(i, [j]) \ - \ \#^L(*, [M])}$$

  If the denominator is not zero and $GFP = 1$, it means that the blackhole attack is detected and M is the attacker.

- ***Malicious Flooding on specific target***: This rule uses $\#^L([m], [d])$ for every destination d. If it is larger than the threshold the attack is Malicious Flooding. However, the attacker cannot be determined.

The authors also presented cluster formation algorithms and ensured that they are fair and secure. Each and every node has an equal chance of becoming a clusterhead and serves as a clusterhead for an equal service time. In addition, no node can manipulate the clusterhead selection process. Initially, each node forms a clique - a group of nodes where every pair of members can communicate via a direct wireless link. Then, members in the clique perform the selection of a clusterhead. The process of re-election,

Techniques		Watchdog/ Pathrater	CONFIDANT	CORE	OCEAN	Cooperative IDS
Architecture		Distributed and cooperative			Stand-alone	Hierarchical
Type of data collection		Reputation				Statistics
Data distribution		negative to source node	negative to friends	positive from RREP	no	to clusterhead
Observation	self to neighbor	yes	yes	yes	yes	yes
	neighbor to neighbor	no	yes	no	yes	yes
Misbehavior detection	Selfish – routing	no	yes	yes	yes	yes
	Selfish – packet forwarding	yes	yes	yes	yes	yes
	Malicious – routing	no	yes	no	no	yes
	Malicious – packet forwarding	yes	yes	no	no	yes
Punishment		no	yes	yes	yes	n/a
Avoid misbehaving node in route discovery		no	no	no	yes	n/a

Table 1: Comparison among IDS for Node Cooperation

to enforce fairness, and the process of recovery from lost clusterheads are defined as well.

Monitoring is how data is obtained in order to analyze for possible intrusions, however it consumes power. Therefore, instead of every node capturing all features themselves, the clusterhead is solely responsible for computing traffic-related statistics. This can be done because the clusterhead overhears incoming and outgoing traffic on all members of the cluster as it is one hop away (a clique). As a result, the energy consumption of member nodes is lessened, whereas the detection accuracy is just a little worse than that of not implementing clusters. Besides, the performance of the overall network is noticeably better - decreases in CPU usage and network overhead.

5.6 Summary of IDS for Detecting Misbehaving Nodes

Although the watchdog is used in all of the above IDS, the authors in [5] have pointed out that there are several limitations. The watchdog cannot work properly in the presence of collisions, which could lead to false accusations. Moreover, when each node has different transmission ranges or implements directional antennas, the watchdog could not monitor the neighborhood accurately.

All of the above IDS's presented are common in detecting selfish nodes. However, CORE doesn't detect malicious misbehaviors while the others detect some of them, i.e., unusually frequent route update, modifying header or payload of packets, no report of failed attempts, etc. Table 1 shows the comparison among these IDS.

6 Conclusions and Future Directions

As the use of mobile ad hoc networks (MANETs) has increased, the security in MANETs has also become more important accordingly. Historical events show that prevention alone, i.e., cryptography and authentication are not enough; therefore, the intrusion detection systems are brought into consideration. Since most of the current techniques were originally designed for wired networks, many researchers are engaged in improving old techniques or finding and developing new techniques that are suitable for MANETs.

With the nature of mobile ad hoc networks, almost all of the intrusion detection systems (IDSs) are structured to be distributed and have a cooperative architecture. The number of new attacks is likely to increase quickly and those attacks should be detected before they can do any harm to the systems or data. Hence, IDS's in MANETs prefer using anomaly detection to misuse detection [1, 3, 4, 14, 24]. Some techniques are proposed to implement on top of the existing protocols [5, 6, 7], others are proposed as independent modules to be added on mobile nodes [1, 3, 4, 14, 16, 24].

An intrusion detection system aims to detect attacks on mobile nodes or intrusions into the networks. However, attackers may try to attack the IDS system itself [5]. Accordingly, the study of the defense to such attacks should be explored as well.

Many researchers are currently occupied in applying game theory for cooperation of nodes in MANETs [20, 21, 22, 23] as nodes in the network represent some characteristics similar to social behavior of human in a community. That is, a node tries to maximize its benefit by choosing whether to cooperate in the network. There is not much work done in this area, therefore, it is an interesting topic for future research.

Acknowledgements

This work was supported in part by NSF grants CCR 0329741, CNS 0422762, CNS 0434533, ANI 0073736, EIA 0130806, and a grant from Motorola Inc.

References

[1] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques for Mobile Wireless Networks," *ACM/Kluwer Wireless Networks Journal (ACM WINET)*, Vol. 9, No. 5, September 2003.

[2] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks," *IEEE Wireless Communications*, Vol. 11, Issue 1, pp. 48-60, February 2004.

[3] P. Albers, O. Camp, J. Percher, B. Jouga, L. M, and R. Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches," *Proceedings of the 1st International Workshop on Wireless Information Systems (WIS-2002)*, pp. 1-12, April 2002.

[4] O. Kachirski and R. Guha, "Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks," *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, p. 57.1, January 2003.

[5] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00)*, pp. 255-265, August 2000.

[6] S. Buchegger and J. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes - Fairness In Dynamic Ad-hoc NeTworks)," *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, pp. 226-336, June 2002.

[7] P. Michiardi and R. Molva, "Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," *Communication and Multimedia Security Conference (CMS'02)*, September 2002.

[8] D. B. Johnson, and D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, October 1999.

[9] P. Brutch and C. Ko, "Challenges in Intrusion Detection for Wireless Ad-hoc Networks," *Proceedings of 2003 Symposium on Applications and the Internet Workshop*, pp. 368-373, January 2003.

[10] M. G. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing," *ACM Mobile Computing and Communication Review (MC2R)*, Vol. 6, No. 3, pp. 106-107, July 2002.

[11] Y. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*, pp. 3-13, June 2002.

[12] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure On-Demand Routing Protocol for Ad hoc Networks," *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, pp. 12-23, September 2002.

[13] A. Perrig, R. Canetti, D. Tygar and D. Song, "The TESLA Broadcast Authentication Protocol," *RSA CryptoBytes*, 5 (Summer), 2002.

[14] Y. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks," *Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN'03)*, pp. 135-147, October 2003.

[15] Y. Huang, W. Fan, W. Lee, and P. Yu, "Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies," *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS'03)*, May 2003.

[16] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C.-Y. Tseng, T. Bowen, K. Levitt, and J. Rowe, "A General Cooperative Intrusion Detection Architecture for MANETs," *Proceedings of the 3rd IEEE International Workshop on Information Assurance (IWIA'05)*, pp. 57-70, March 2005.

[17] Y. F. Jou, F. Gong, C. Sargor, X. Wu, S. Wu, H. Chang, and F. Wang, "Design and Implementation of a Scalable Intrusion Detection System for the Protection of Networks Infrastructure," *Proceedings of DARPA Information Survivability Conference and Exposition*, Vol. 2, pp. 69-83, January 2000.

[18] E. Y. K. Chan *et al.*, "IDR: An Intrusion Detection Router for Defending against Distributed Denial-of-Service (DDoS) Attacks," *Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'04)*, pp. 581-586, May 2004.

[19] S. Bansal and M. Baker, "Observation-Based Cooperation Enforcement in Ad hoc Networks," *Research Report cs.NI/0307012*, Stanford University, 2003.

[20] P. Michiardi and R. Molva, "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad Hoc Networks," *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, March 2003.

[21] A.Agah, S. K. Das, K. Basu, and M. Asadi, "Intrusion Detection in Sensor Networks: A Non-Cooperative Game Approach," *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (NCA'04)*, pp. 343-346, 2004.

[22] R. Mahajan, M. Rodrig, D. Wetherall and J. Zahorjan, "Experiences Applying Game Theory to System Design," *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PIN'04)*, pp. 183-190, September 2004.

[23] S. Zhong, L. Li, Y. G. Liu and Y. Yang, "On Designing Incentive-Compatible Routing and Forwarding Protocols in Wireless Ad-hoc Networks: An Integrated Approach Using Game Theoretical and Cryptographic Techniques," *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom'05)*, pp. 117-131, 2005.

[24] B. Sun, K. Wu, and U. W. Pooch, "Alert Aggregation in Mobile Ad Hoc Networks," *Proceedings of the 2003 ACM Workshop on Wireless Security (WiSe'03) in conjuction with the 9th Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, pp. 69-78, 2003.