A Comparative Evaluation of Intrusion Detection Architectures for Mobile Ad Hoc Networks

Christos Xenakis[1], *Christoforos Panos*[2], *Ioannis Stavrakakis*[2]

[1]*Department of Digital Systems, University of Piraeus, Greece*

[2]*Department of Informatics & Telecommunications, University of Athens, Greece*

e-mail: xenakis@unipi.gr, cpanos@di.uoa.gr, ioannis@di.uoa.gr

Abstract

Mobile Ad Hoc Networks (MANETs) are susceptible to a variety of attacks that threaten their operation and the provided services. Intrusion Detection Systems (IDSs) may act as defensive mechanisms, since they monitor network activities in order to detect malicious actions performed by intruders, and then initiate the appropriate countermeasures. IDS for MANETs have attracted much attention recently and thus, there are many publications that propose new IDS solutions or improvements to the existing. This paper evaluates and compares the most prominent IDS architectures for MANETs. IDS architectures are defined as the operational structures of IDSs. For each IDS, the architecture and the related functionality are briefly presented and analyzed focusing on both the operational strengths and weaknesses. Moreover, methods/techniques that have been proposed to improve the performance and the provided security services of those are evaluated and their shortcomings or weaknesses are presented. A comparison of the studied IDS architectures is carried out using a set of critical evaluation metrics, which derive from: (i) the deployment, architectural, and operational characteristics of MANETs; (ii) the special requirements of intrusion detection in MANETs; and (iii) the carried analysis that reveals the most important strengths and weaknesses of the existing IDS architectures. The evaluation metrics of the IDSs are divided into two groups: the first one is related to performance and the second to security. Finally, based on the carried evaluation and comparison a set of design features and principles are presented, which have to be addressed and satisfied in future research of designing and implementing IDSs for MANETs.

Keywords: Intrusion detection system, IDS architectures, mobile ad hoc networks, MANETs security, security attacks, security vulnerabilities.

1 Introduction

A mobile ad hoc network (MANET) is a collection of autonomous nodes that form a dynamic, purpose-specific, multi-hop radio network in a decentralized fashion. In a MANET, the nodes themselves implement the network management in a cooperative fashion and thus, all the network members share the responsibility for this. The wireless - mobile nature of MANETs in conjunction with the absence of access points, providing access to a centralized authority, make them susceptible to a variety of attacks [1]. An effective way to identify when an attack occurs in a MANET is the deployment of an Intrusion Detection System (IDS). The IDS is a sensoring mechanism that monitors network activity in order to detect malicious actions and, ultimately, an intruder. Upon detecting an intruder, the IDS takes an appropriate action ranging from a mere user notification to a more comprehensive defensive action against the intruder. An IDS can be divided in two main parts: (i) the architecture, which exemplifies the operational structure of the IDS; and (ii) the detection engine, which is the mechanism used to detect malicious behavior(s).

The existing IDS architectures for MANETs fall under three basic categories [3]: (a) stand-alone, (b) cooperative, and (c) hierarchical. The *stand-alone architectures* use an intrusion detection engine installed at each node utilizing only the node's local audit data [10][12][15]. However, the fact that these solutions are relying only on local audit data to resolve malicious behaviors limits them in terms of detection accuracy and the type of attacks that they detect [9] (due to the distributed nature of MANETs). On the other hand, the cooperative and hierarchical architectures process each host's audit data locally (i.e., similarly to stand-alone), but they also use *collaborative techniques* to detect more accurately a wider set of attacks. Thus, the majority of the most recent IDSs for MANETs is based on them [9]. More specifically, the *cooperative architectures* include an intrusion detection engine installed in every node, which monitors local audit data and exchanges audit data and/or detection outcomes with neighboring nodes in order to resolve inconclusive (based on single node's audit data) detections. The *hierarchical architectures* amount to a multilayer approach, by dividing the network into clusters. Specific nodes are selected (based on specific criteria) to act as cluster-heads and undertake various responsibilities and roles in intrusion detection that are usually different from those of the simple cluster members. The latter typically run a lightweight local intrusion detection engine that performs detection only on local audit data, while the cluster-heads run a more comprehensive engine that acts as a second layer of detection based on audit data from all the cluster members.

The employed intrusion detection engines are also classified into three main categories: (i) signature-based engines, which rely on a predefined set of patterns to identify attacks; (ii) anomaly-based engines, which rely on particular models of nodes' behavior and mark nodes that deviate from these models as malicious; and (iii) specification-based engines, which rely on a set of constrains (i.e., description of the correct operation of programs/protocols) and monitor the execution of programs/protocols with respect to these constraints.

IDS for MANETs have attracted much attention recently and thus, there are many publications that propose new IDS solutions or improvements to the existing focusing on both IDS architectures and detection engines. On the other hand, little work has been done in evaluating and comparing them revealing their advantages as well as their limitations and weakness, which constitute open issues that will drive the next research steps in the area of MANET security. Towards this direction, Sun et al. [5] have presented a survey of IDSs for MANETs and wireless sensor networks considering on the detection engines employed. Similarly, Azer et al. [6] briefly discuss the anomaly-based detection engines used in IDSs for MANETs. However, both works mainly focus on solutions published before 2004 (except for one [21] in the former).

Brutch and Ko [4] provide a brief analysis of several proposed IDSs for MANETs focusing mainly on their architectures. However, the analyzed solutions have been designed to protect the routing mechanism of the dynamic source routing protocol (DSR), operating as extensions to it, and thus, they do not address the wide area of intrusion detection in MANETs. Mishra et al. [2] present a more detailed analysis of IDSs for MANETs following: (i) an outline of the security vulnerabilities of MANETs; (ii) some design characteristics of IDSs for MANETs; and (iii) some fundamental requirements that an IDS for MANETs should meet. The architectures of the analyzed IDSs are elaborated and briefly compared with the set of fundamental requirements introduced by the authors. Li and Wei [7] briefly overview some IDS architectures for MANETs and compare them in terms of implementation-specific issues. Anantvalee and Wu [3] perform a more comprehensive analysis of some IDS architectures for MANETs. Finally, Sen et al. [9] present the latest survey of IDSs for MANETs, revealing the weaknesses of each one. However, the studied IDS solutions in the aforementioned works have been published before 2006. Moreover, the considered architectures are hardly evaluated and compared with respect to performance and security factors, such as the consumption of processing and communication resources, the fair distribution of the workload among the network nodes, the impact of nodes' mobility on the detection accuracy and the rate of false positives, the vulnerabilities of the architectures to attacks, etc.

This paper evaluates and compares the most prominent IDS architectures for MANETs, which represent the most recent developments in this area. For each IDS, the architecture and the related functionality are briefly presented and analyzed focusing on both the operational strengths and weaknesses. Moreover, methods/techniques that have been proposed to improve the performance and the provided security services of those are evaluated and their shortcomings or weaknesses are presented. A comparison of the studied IDS architectures is carried out using a set of critical evaluation metrics, which derive from: (i) the deployment, architectural, and operational characteristics of MANETs; (ii) the special requirements of intrusion detection in MANETs; and (iii) the carried analysis that reveals the most important strengths and weaknesses of the existing IDS architectures. The evaluation metrics of the IDSs are divided into two groups: the first one is related to performance and the second to security. Finally, based on the carried evaluation and comparison a set of design features and principles are presented, which have to be addressed and satisfied in future research when designing and implementing IDSs for MANETs.

The rest of this article is organized as follows. Sections 2-4 briefly analyze and evaluate the stand-alone, cooperative and hierarchical IDS architectures for MANETs, respectively, focusing on their advantages and limitations. Section 5 compares the studied IDS architectures, using a set of performance and security metrics. Section 6 highlights some design features and principles that are derived from the carried analysis, evaluation, and comparison. Finally, section 7 contains the conclusions.

2 Stand-alone IDS architectures

The stand-alone IDS architectures are based on a self-contained approach for detecting malicious actions at each network node. In this section, we briefly present and evaluate the most recent stand-alone IDS architectures for MANET (i.e., battery-based, threshold-based, and two-stage IDS architecture) focusing on the strengths and weaknesses of each one, which are summarized in Table 1, allowing their comparison.

Jacoby and Davis have proposed a stand-alone architecture for detecting malicious actions in MANETs, by monitoring power consumption in every node's battery [10]. Detection is achieved by comparing a node's power consumption with a set of power consumption patterns induced by known attacks, using smart battery technology. In an experimental implementation, the proposed IDS detected 99% of the attacks in cases that only one type of them occurred. It also detected multiple attacks, but only in cases that the nodes were idle and no other activity was present. The main advantage of this architecture is that it is more reliable (i.e., since it is based on hardware operation), compared to other IDSs that rely on audit data and anomaly-based detection, as these can be more easily manipulated by intruders. On the other hand, it detects only attacks that cause power consumption irregularities and only in cases that the nodes are idle, something that rarely occurs in real systems.

Nadkarni and Mishra [12] have proposed a stand-alone IDS architecture that uses compound detection aiming at reducing the amount of false positive alerts, which typically appear in anomaly detection. It employs adjusting thresholds to determine malicious behaviors. During initialization, the intrusion detection engine installed in every node creates the normalcy profile of the network traffic. Based on this, it estimates threshold values, beyond of which there is an indication of possible attacks. Every time a symptom of a known attack is detected, a counter called mis-incident is incremented and the node responsible for the symptom is marked as suspicious. If the incident repeats and the mis-incident counter exceeds the threshold value for the specific attack, the node from where the incident originates is labeled as malicious. After a preset period of time in which there are no malicious behaviors detected, the threshold is raised; otherwise is lowered.

The most important strength of this architecture is that it is adaptable to network changes, because of the use of variable thresholds. For example, periodic symptoms of suspicious behaviors, caused by network topology changes, will remain under the detection thresholds; while malicious behaviors that are constant will exceed the thresholds indicating the occurrence of attacks. On the other hand, the use of adjusting thresholds introduces new security weaknesses, since malicious nodes may exploit this mechanism. More specifically, a malicious node may increase the threshold values by performing legitimately for a certain period of time. Then, if the threshold values are high enough, it may perform an attack considering not exceeding the threshold values and raising alarms. Nodes that might not cooperate in the routing process or generate invalid routing updates due to outdated routing information (i.e., caused by high mobility) might be falsely characterized as malicious. Moreover, coordinated attacks (i.e., such as byzantine attacks) cannot be detected, since nodes do not cooperate.

Finally, Adrian Lauf et al. [15] have proposed a two-stage, stand-alone IDS architecture that aims at operating in resource-constrained environments, such as MANETs. It installs two different detection engines in every node, where the first one (referred to as the maxima detection system (MDS)) is used to rapidly identify a potential threat and calibrate the second engine (referred to as the cross-correlative detection system (CCDS)). MDS is an anomaly detection engine that identifies statistical oddities in the observed interactions of the application layer. This is achieved by maintaining the history of the application layer interactions and comparing them with a normalcy profile created offline. If a possible attack is identified, MDS activates CCDS that calibrates a threshold value considering the attack. Then, calculates average values of the application behavior of every node and compares them with the threshold. Behaviors that exceed the threshold are marked as malicious. By employing two detection engines at each node, the proposed IDS increases detection accuracy, compared to other single engine IDSs because the one engines supplements the other. However, CCDS is prone to false positives and negatives, since it calibrates the threshold value only once during startup. Thus, dynamic changes of the network, induced by nodes mobility, are note accommodated by CCDS.

Table 1. Strengths and weaknesses of the stand-alone IDS architectures

IDS architecture	Strengths	Weaknesses
Battery-based IDS	Reliability, since it is based on hardware operations	It detects only attacks that cause power irregularities
Threshold-based IDS	Adaptability to network changes using adjusting thresholds.	Introduces new security weaknesses
		It is prone to false positives
		Cannot detect coordinated attacks
Two-stage IDS	Increased detection accuracy by employing two detection engines at each node.	It is prone to false positives and negatives

3 Cooperative IDS architectures

In the cooperative IDS architectures an intrusion detection engine is installed in every node monitoring local audit data and providing intrusion detection. To resolve inconclusive intrusion detections and detect more accurately advanced types of attacks, detection engines may cooperate with engines of neighboring nodes through the exchange of audit data or detection outcomes.

3.1 A cooperative IDS architecture based on social network analysis

Wang et al. [14] have proposed a cooperative IDS architecture, which relies on a detection engine that utilizes social network analysis methods. In this architecture, each node deploys an intrusion detection engine that performs detections using audit data received from its “ego” network. An “ego” network consists of a hosting node (“ego”) and the nodes (“alters”) that are directly connected to it. The deployed engines operate similarly to anomaly detection, but they utilize social relations as metrics of interest, which require less computational overhead compared to standard anomaly detection engines [14]. Moreover, a training phase is also required to create normal profiles (i.e., as in anomaly detection), and according to the authors, the detection engines monitor the Medium Access Control (MAC) and network layers.

The proposed IDS is composed of three modules: (a) the data pre-processing module that collects and pre-processes audit data; (b) the social analysis module that performs intrusion detection; and (c) the response module that integrates local and global (i.e., gathered from neighboring nodes) intrusion alerts. During the IDS operation, the data pre-processing module collects audit data from its neighboring nodes in intervals of five seconds. The social analysis module, then, processes the collected data in order to realize social relations between the “ego” network nodes, which represent the behavior of these nodes at a certain time.

Subsequently, the realized relations are compared to the normal profile of expected behaviors, and any variation from these constitutes an intrusion. If an intrusion is detected, the response module notifies the neighboring nodes.

The main strength of this architecture is that the employed detection engines incur less computation complexity, compared to conventional anomaly detection engines [14]. On the other hand, it presents some weaknesses outlined bellow:

- The rate of false positives may increase and the detection accuracy may drop in cases of high nodes' mobility. In a high mobility scenario, a node would only have a limited period of time to create social relations with neighboring nodes, before it changes its location. As a result, there would not be enough information for the social analysis module to distinguish between normal and malicious behaviors.
- Audit data exchange may increase the communication load among nodes, causing degradation to the network performance. The authors have arbitrarily selected a five-second interval for audit data exchange within each "ego" network, without any evaluation of the impact of this parameter to the network performance.
- New security risks may arise from the exchange of audit data, since a malicious node may either transmit false audit data or avoid transmitting any of them, in order to hinder or mislead the detection process.

3.2 A multi-layer cooperative detection architecture

Bose et al. [16] have proposed a cooperative IDS architecture that uses three parallel anomaly detection engines, reffered as MAC layer detection engine, routing detection engine, and application layer detection engine, installed in every node. The use of multi-layer detection aims at increasing detection accuracy, since attacks that target upper-layer protocols can be seen as legitimate events at lower-layers, and vice versa. The MAC layer detection engine monitors both access control and addressing at the data link layer. The routing detection engine monitors the network layer and keeps track of the packet delivery and routing information. Finally, the application layer engine monitors the application layer. Each engine collects the appropriate audit data, processes them and looks for malicious behaviors within them. In every node, a local integration module combines the results from the three different detection engines, while a global integration module combines the results received from the neighboring nodes. A set of simulations has been performed (using the GloMoSim [17]) to evaluate the effectiveness of the proposed architecture.

The multi-layer IDS presents the following strengths:

- It increases the detection accuracy, compared to other single engine detection solutions, as the multiple detection engines supplement each other. In the simulation results, the detection accuracy increased up to 20% through integrating the results of all three engines, compared to the results that each detection engine yielded by itself (see Fig. 1) [16].

- Although it uses cooperation between the neighboring nodes, it induces relatively low communication overhead, since only the detection results and not the voluminous audit data are exchanged.

The considered IDS architecture also presents some weaknesses:

- Its operation increases the processing overhead in each node, compared to other single engine solutions, since the IDS deploys three detection engines instead of one. So far, the authors have not studied or evaluated the processing overhead of the proposed architecture.
- The ratio of false positives and the detection accuracy of the IDS are negatively affected by high packet loss and/or high nodes' mobility. This is because the routing detection engine relies on packet delivery and routing information to detect attacks. Except for the local integration module, the inaccurate detection results also influence the global integration modules of the neighboring nodes.
- The functionality of cooperation creates new security risks, since a malicious node may either transmit false detection results (i.e., "blackmail" attack) or modify detection results originating from another cooperating node (i.e., "man in the middle attack") in order to hinder or mislead the detection process in a node or set of nodes.

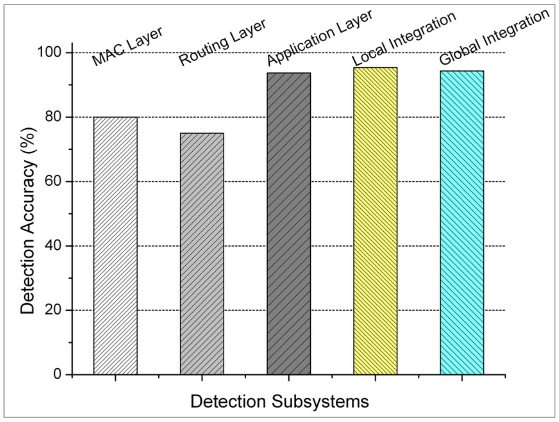

Fig. 1: Detection accuracy of the multi-layer cooperative detection architecture.

3.3 A Friend-assisted intrusion detection architecture for MANETs

Razak et al. [18] have proposed a cooperative two-tier (i.e., one for local and one for global detection) IDS architecture for MANETs, where each tier includes two detection engines, respectively. The first-tier uses a local-level detection mechanism that collects local audit data and processes them using a signature-based detection engine. If it detects a suspicious activity but it cannot determine accurately a specific attack, a second engine is activated (also located in the first-tier) that performs anomaly detection. If both engines at the first-tier cannot conclude whether the suspicious activity is malicious, the second-tier of the architecture is triggered. The second-tier uses a global detection mechanism that collects audit data from the neighboring nodes and first performs a signature-based detection and then an anomaly-based detection, similarly to the first-tier. The second-tier also maintains a list of friends (each node builds and maintains a list of trustful nodes), which is used to ensure that the nodes sharing their audit data with it are trustful.

The strengths of the friend-assisted IDS architecture are:

- It provides high detection accuracy since each node contains a two-tier detection module and each tier includes two different detection engines (i.e., one that uses signature-based detection and another that uses anomaly-based detection) that act complementary.
- It is not susceptible to blackmail attacks since only trustful nodes can send audit data to the second-tier of a node (i.e., global detection). Therefore, a malicious node cannot provide false audit data in order to mislead the IDS or falsely characterize legitimate nodes as malicious.

The weaknesses of this architecture are:

- The use of multiple detections (i.e., two tiers each of which contains two different detection engines) and the employment of trust management add a considerable complexity and processing load.
- The rate of false positives and the detection accuracy of the IDS are negatively affected by the lack of trust relationships among nodes and by nodes' mobility. In a network with limited trust relationships, the IDS might not find enough trustful nodes to collect a sufficient amount of audit data to determine whether an event occurring is legitimate or not. This also can be the result of trusted nodes that move continually.
- It imposes extra communication overhead, mainly for three reasons: (i) the second-tier detection requires the exchange of audit data; (ii) nodes have to exchange trust information in order to build lists of friends; and (iii) the use of signature-based detection requires the existence of a signature distribution authority that periodically transmits new signatures to each node.

3.4 Fork: A two pronged intrusion detection scheme for MANETs

Ramachandran et al. have proposed a cooperative IDS architecture [19], which uses lightweight modules (agents) able to perform different detection tasks and aim at reducing battery consumption. Each network node contains all the modules required to perform the detection tasks and is assigned a reputation value, which increases when the node successfully assists with intrusion detection tasks, and decreases if the node's performance during intrusion detection is unsatisfactory. Nevertheless, the authors do not clarify under what conditions the node’s performance is deemed unsatisfactory. The employed intrusion detection engine relies on anomaly detection and it is installed in every node. When the engine of a node detects a suspicious behavior, it initiates an auction scheme to select a set of nodes that are most suitable to assist in performing intrusion detection. Nodes with the highest amount of battery resources and reputation value are selected and specific tasks are assigned to them. These tasks include: (i) the execution of host or network monitoring, (ii) the decision making given a set of audit data, and (iii) the activation of defensive actions in case that malicious behaviors have been detected. The authors neither elaborate on how nodes’ cooperation is achieved nor evaluate the communication overhead imposed by the employed cooperation mechanism. Moreover, they did not consider node’s mobility in the performed simulations, thus the impact of mobility on the detection accuracy, the rate of false positives and the communication overhead cannot be determined.

The main advantage of the Fork architecture is the distribution of detection tasks among a set of nodes, which reduces the processing load for the initiating node and conserves its battery power. The selection of assisting nodes also considers - among other criteria - the available battery resources thus, nodes with lower battery power are not burdened with intrusion detection responsibilities.

On the other hand, the weaknesses of the architecture are:

- High nodes’ mobility may increase the communication overhead imposed by the IDS architecture. A node assigned with a detection task may move away from the initiating node thus, it has to route the results regarding its task through other nodes. However, this extra communication overhead has not been quantified through a simulation or analytic study.
- It is vulnerable to man in the middle attacks, since a malicious node, exploiting the task allocation mechanism, may capture and modify intrusion detection task messages. A malicious node might also cause blackmail attacks, by transmitting false detection results to the node that has initiated detection tasks. Finally, a malicious node may cause sleep deprivation attacks, by initiating fake tasks to other nodes in order to consume their resources.

3.5 Routing anomaly detection architecture

Sun et al. [20] have proposed a cooperative IDS architecture that focuses on routing disruption attacks. Since all the nodes of a MANET participate in routing, each one maintains a table that contains routing information, such as routing paths to reach other nodes and the required number of hops. Extensive changes in this table may be a symptom of malicious behaviors that attempt to disrupt the routing process. The proposed IDS uses the following two routing features to discover malicious behaviors: (i) the percentage of changes in the route entries (PCR), and (ii) the percentage of changes in the number of hops (PCH). PCR represents the added/deleted route entries during a certain period of time, while PCH indicates the change in the sum of hops of all route entries over the period of time.

In this IDS, one or several intrusion detection engines that rely on anomaly detection are installed in every node. These engines collect and process routing information to detect possible intrusions, using a modified Markov Chain anomaly detection method [32]. In case that more than one detection engines are deployed in a node, alerts and reports from each local engine are combined. Moreover, data reports and alerts from neighboring nodes are also correlated in order to reach more accurate decisions. Based on the performed simulations, the authors state that this IDS detects more than 90% of the routing disruption attacks, in scenarios with relative low nodes' mobility (i.e., nodes speed ranges from 3m/s to 5m/s).

The main advantage of this architecture is related to the increased detection accuracy that it presents, because of the deployment of multiple detection engines at each node (i.e., compared to other single engine solutions). This fact also makes this IDS fault tolerant in cases that a detection engine fails or becomes a target of an attack.

On the other hand, it presents some drawbacks:

- It cannot be used to detect all the types of possible attacks, since it monitors only for routing attacks.
- It imposes extra communication overhead, since detection engines hosted at neighboring nodes have to constantly exchange detection reports and alerts in order to reach more accurate decisions.
- The detection accuracy and the ratio of false positives are negatively affected by nodes' mobility, as illustrated (Routing Anomaly Detection curve) in Fig. 2 and Fig. 3, respectively. This occurs for two reasons: (i) in a high mobility scenario, a node would only notice a few falsified routing changes before changing its location; and (ii) in such scenarios, the changes in routing tables are rapid and inconsistent. Thus, there is not enough information for the detector to distinguish between normal behaviors provoked by nodes' mobility and abnormal behaviors provoked by malicious nodes.
- It is vulnerable to blackmail attacks, since a malicious node might transmit false detection reports or alerts in order to hinder the intrusion detection process and falsely accuse a legitimate node(s) as malicious.

Later on, Sun et al. [21] improved the aforementioned routing anomaly IDS architecture, by proposing the incorporation of a new intrusion detection engine with adjustable thresholds. This addresses some of the most important drawbacks of this architecture, such as the negative impacts of nodes' mobility on the detection accuracy and the ratio of false positives. The technique of adjustable thresholds ensures that the periodical changes in routing information, caused by nodes' mobility, will remain under the detection threshold; while malicious behaviors that are persistent will exceed the thresholds indicating the occurrence of attacks. The authors have performed a performance analysis (i.e., based on simulations) comparing the enhanced with the initial routing anomaly detection architecture. The enhanced architecture preserves the advantages of the initial, and as observed in Fig. 2 and Fig. 3 (Adaptive Routing Anomaly Detection curves), it reduces the negative impact of high nodes' mobility on the detection ratio and the rate of false positives. On the other hand, the technique of adjustable thresholds creates new security risks. More specifically, in case that a malicious node notices high mobility, it might act maliciously without being detected.

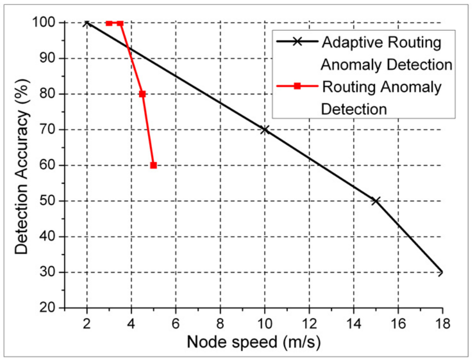

Fig. 2: The impact of nodes' mobility on detection accuracy

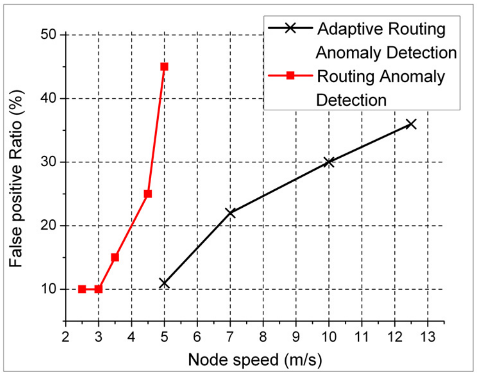

Fig. 3: The impact of nodes' mobility on the ratio of false positives

3.6 LIDF: Layered intrusion detection framework for ad-hoc networks

Komninos and Douligeris have proposed a cooperative IDS architecture [22], which relies on multilayered detection to capture malicious behaviors. In this architecture, every host maintains an intrusion detection unit, which is divided into three modules: (i) the collection, (ii) the detection, and (iii) the alert module. The collection module is responsible for collecting audit data from both the data link and the network layer. By monitoring these two layers the IDS has a close view of the networking activities (i.e., nodes' connectivity and routing). The detection module performs anomaly-based detection on the collected audit data in two steps, in order to conserve the host's resources and battery. First, it processes only the most recent local audit data. In case that these data are not sufficient to reach an accurate decision regarding a suspicious behavior, more audit data are requested from neighboring nodes via secure communication channels. However, the authors have not specified when do nodes decide to request neighbors' cooperation, and how this cooperation is achieved (i.e., exchange of audit data or detection results). As a result of these, the communication overhead imposed by nodes' cooperation cannot be determined. Finally, in case that a malicious behavior is detected, the alert module has the responsibility to notify the neighboring nodes.

The strengths of this IDS architecture are:

- Using multiple layers of detection, it is able to detect attacks at both the network and data link layer.
- The use of secure communication channels for nodes' cooperation defeats man in the middle attacks.

On the other hand, the weaknesses of this architecture are:

- It focuses only on attacks that target the network and data link layer. Attacks at the transport layer - such as a SYN flooding, where a malicious node sends a large number of SYN packets, or a session hijacking attack, where a malicious node takes control over a session between two nodes - will go undetected.
- Nodes' mobility reduces the detection accuracy of the IDS and increases the ratio of false positives, since it hinders cooperation as the nodes move away from each other.
- It is vulnerable to blackmail attacks, since a malicious node that cooperates might transmit modified audit data in order to hinder the intrusion detection process, hide malicious activities or falsely accuse legitimate nodes as malicious.

3.7 Strengths and weaknesses of the cooperative IDS architectures

This section summarizes the basic strengths and weaknesses of the studied cooperative IDS architectures (see Table 2) that derive from the curried analysis and evaluation, allowing their comparison. Regarding the strengths of the analyzed architectures, we can infer that: (i) the majority of them employ multiple detection engines in order to provide increased detection accuracy and detect a wide set of possible attacks; (ii) some of them attempt to minimize the imposed processing and communication overheads through task distribution or the exchange of detection results, instead of voluminous audit data among neighboring nodes; and (iii) a few of them try to defeat certain attacks by employing trust or secure communication channels. On the other hand, in regard to their weaknesses, we can deduce that: (i) in the entire set of the studied architectures the ratio of false positives and detection accuracy are negatively affected by high nodes' mobility; (ii) almost all of them impose extra processing and communication overhead (especially in cases that the underlying network presents high nodes' mobility); and (iii) the majority of them are vulnerable to attacks (i.e., man in the middle, blackmail, etc.).

Table 2. Strengths and weaknesses of the cooperative IDS architectures

IDS architecture	Strengths	Weaknesses
Cooperative IDS architecture based on social network analysis	The employed social-based detection engine incurs less computational complexity than the conventional anomaly-based engines.	The ratio of false positives and detection accuracy are negatively affected by high nodes' mobility.
		Audit data exchange increases the communication load among nodes

		Audit data exchange creates new security risks
Multi-layer cooperative IDS architecture	The multiple detection engines employed provide increased detection accuracy.	The employment of multiple engines at each node increases the processing overhead.
		The ratio of false positives and detection accuracy are negatively affected by high packet loss and/or high nodes' mobility.
	The exchange of detection results among the neighboring nodes achieves nodes' cooperation with the minimum communication overhead.	It is vulnerable to blackmail and man in the middle attacks
Friend-assisted IDS architecture	The multiple detection engines employed provide increased detection accuracy.	The employment of multiple engines and trust management at each node increase both the processing and communication overhead
	It defeats blackmail attacks by employing trust.	The ratio of false positives and detection accuracy are negatively affected by limited trust relationships between nodes and/or high nodes' mobility
FORK	It reduces the processing load and conserves the battery power of nodes through task distribution.	The communication overhead is increased under high nodes' mobility
		It is vulnerable to blackmail, man in the middle, and sleep deprivation attacks
Routing anomaly detection architecture	The multiple detection engines employed provide increased detection accuracy and a fault tolerant solution	In the initially proposed architecture, the ratio of false positives and detection accuracy are negatively affected by high nodes' mobility
		It detects only routing attacks
		It imposes extra communication overhead
		It is vulnerable to blackmail attacks
LIDF	It is able to detect attacks at multiple layers (i.e. network and data link layers)	It does not detect attacks at the transport layer (i.e. SYN flooding, session hijacking etc.).
		The ratio of false positives and detection accuracy are negatively affected by high nodes' mobility
	It defeats man-in-the-middle attacks using secure communication channels	It is vulnerable to blackmail attacks

4 Hierarchical IDS architectures

In the hierarchical IDS architectures the network nodes are divided into cluster-heads and cluster members. The latter typically run a lightweight local intrusion detection engine, while the former run a comprehensive engine that processes raw or pre-processed audit data from all the cluster members.

4.1 A cluster-based intrusion detection architecture with adaptive selection event triggering

The hierarchical IDS architecture, proposed by Ma and Fang [31], follows a modular approach based on clusters. The goal is to provide a clustered structure where cluster-heads are always hosted by nodes with the highest battery power. During network initialization, each node reports its battery power to its neighbors. Then, the node with the highest available battery power is elected as cluster-head. A cluster-head re-election process is triggered as soon as one the following event occurs: (i) a new node joins the network, (ii) the elected cluster-head leaves the network, or (iii) the battery power of the cluster-head is lower than a predefined threshold. When a new node joins the network, it should first notify all of its neighboring nodes. Likewise, if a cluster-head leaves the network, it broadcasts a packet to notify its cluster-member nodes in order to initiate the cluster-head re-election procedure.

In this IDS architecture, each network node contains four different modules, described bellow:

a. *The network detection module* that provides network packet monitoring within a cluster. It is activated only when the hosting node is elected as cluster-head.

b. *The local detection module* that monitors the hosting node and generates local alerts if malicious activities are detected. This module is always active at every node.

c. *The resource management module* that monitors the energy resources of a node acting as cluster-head. When the battery power is lower than a predefined threshold, the module first notifies the monitoring state manage module, and then initiates the cluster-head re-election procedure.

d. *The monitoring state manage module* that manages whether the network detection module is active (i.e., the hosting node is elected as cluster-head).

The proposed architecture presents a number of strengths including:

- The nodes with the highest battery power are elected to serve as cluster-heads.
- It supports two layers of detection (i.e., local and network) providing increased detection accuracy.
- The cluster-head monitors the network packets exchanged thus, there is no extra communication overhead between the cluster-head and the cluster members.

On the other hand, it also presents some weaknesses:

- Nodes elected as cluster-heads are unfairly overloaded, since they are responsible for running both local and network detection modules.
- High nodes' mobility may reduce the detection accuracy of the architecture and increase the ratio of false positives, since a number of nodes may move out of the range of a cluster-head. This limits the information that the network detection module may use to perform detection.

- The creation and maintenance of clusters and the election of cluster-heads add extra processing and communication overhead.
- Having a few nodes responsible for intrusion detection may create points of failure, at least locally in a cluster. If a cluster-head is attacked, crashes, or leaves the cluster or the network without initiating the re-election procedure, only the local detection modules will protect the nodes.
- It is vulnerable to man in the middle and blackmail attacks, since the communication channels between the network nodes are not protected. Thus, a malicious node may modify the transmitted messages in order to mislead the cluster-head.
- A malicious node may exploit the election procedure in order to be elected as cluster-head (i.e., by reporting false values of battery power). Similarly, a selfish node may avoid becoming a cluster-head.

4.2 A hierarchical IDS architecture that uses a game theoretic detection mechanism

Otrok et al. have proposed a hierarchical approach [29] that attempts to balance the consumption of resources (which results from intrusion detection tasks) among the nodes of a cluster. It encourages network nodes to participate in the election of cluster-heads and tries to prevent elected cluster-heads from misbehaving. In the proposed architecture, nodes can operate as: (i) cluster-members, which have no intrusion detection responsibilities; (ii) cluster-heads, which are responsible for intrusion detection within a cluster; or (iii) checkers, which are cluster-members selected randomly to monitor the cluster-head for selfish or malicious behavior.

During initialization, the network nodes report the power of their batteries to their neighboring nodes. Thus, every node creates a list composed of its neighbors' energy power. Based on this list, each node votes the node with the highest energy power to be elected as cluster-head. Then, the elected cluster-head deploys a detection engine that is based on a zero-sum, non-cooperative game, where the cluster-head and a possible intruder are players. The cluster-head monitors only the nodes that participated in the election process. Depending on the battery power of the elected cluster-head, the election process is repeated (after a time-period elapses) and a new cluster-head is elected. The randomly selected checkers partially monitor the cluster-head for selfish or malicious behavior. If a checker has some indications of a cluster-head misbehavior, it cooperates with other checkers to conclude to a decision.

Fig. 4 shows the energy levels of twenty (20) nodes that participate in a carried simulation of the proposed architecture, at three distinct time moments (i.e., 0 sec, after 1500 sec, and after 3000 sec) [29]. At the beginning of the simulation (0 sec), eight (8) nodes have energy power between 100% – 80% and twelve (12) nodes between 80% and 60%. After

1500 seconds, three (3) of them maintain energy power between 100% – 80%, four (4) between 80% and 60%, seven (7) between 60% and 40%, three (3) between 40% and 20%, and three (3) bellow 20% (but none runs out of battery). After 3000 seconds, two (2) nodes preserve energy power between 100% and 80%, two (2) of them have energy power between 80% and 60%, none (0) between 60% and 40%, two (2) between 40% and 20%, seven (7) between 20% and 0% and seven (7) have run out of battery. Therefore, it can be deduced that this architecture imposes unfair power consumption among the network nodes. Moreover, since the authors have not taken into account the nodes' mobility in the carried simulations, we cannot determine its impact on the detection accuracy and the rate of false positives of the architecture.

The operational strengths of this architecture are:

- The nodes with the highest battery power are elected to serve as cluster-heads.
- Misbehaving cluster-heads may be detected from the randomly selected checkers that monitor them.

The main weaknesses of the architecture are:

- Cluster-heads/checker nodes are unfairly overloaded with intrusion detection responsibilities.
- It creates extra processing and communication overhead due to: (i) the formation and maintenance of clusters; and (ii) the operation of checker nodes.
- It is vulnerable to man in the middle and blackmail attacks, since the communication channels between the network nodes are not protected. A malicious node may capture and re-transmit modified messages in order to mislead the cluster-head.
- A cluster-head poses a single point of failure in each cluster. An attack or malfunction of the cluster-head hinders intrusion detection at the respective cluster.
- Selfish nodes may exploit the employed election process by reporting false battery power values in order to participate in the process, but avoid being cluster-heads.
- If a malicious node is selected as a checker, it may falsely accuse a cluster-head for misbehaving.

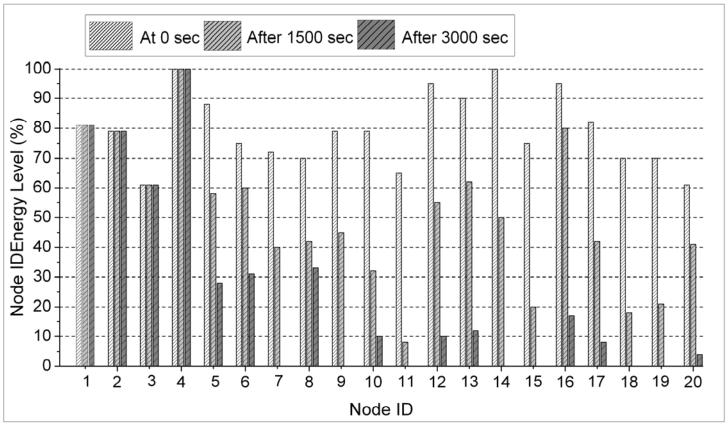

Fig. 4: Energy levels of a set of nodes that participate in a carried simulation of the hierarchical IDS architecture that uses a game theoretic detection mechanism.

4.3 A clustered architecture that uses collective decision for intrusion detection

Marchang and Datta [27] have proposed two intrusion detection architectures that rely on a voting scheme to perform intrusion detection, instead of employing an anomaly or signature-based intrusion detection engine. The difference between the two proposed architectures is that the first, called algorithm for detection in a clique (ADCLI), divides the network into cliques, while the second, called algorithm for detection in a cluster (ADCLU), divides the network into clusters. The concept of a clique is similar to that of a cluster with the difference that each member of a clique is a neighbor with all the others members. In each cluster or clique, where intrusion takes place independently, a monitoring node is elected using various schemes and it is rotated periodically. Upon receiving any suspicious or modified message from a member of its clique/cluster, the monitoring node asks the other clique/cluster members to initiate the intrusion detection process. During this process, (see Fig. 5) the monitoring node (i.e., node 1) sends a message to all the other clique/cluster members (node 0, 2, 3), which forward this message to their neighboring clique/cluster members. If any of the clique/cluster member receives a modified message (or no message at all), it marks the corresponding node that transmitted the modified message (or did not transmit anything) as suspicious. In Fig. 5, “R” denotes the correct message created by the monitoring node, while “W” denotes a modified or tampered message that is transmitted by a malicious node (i.e., node 0). Finally, there is a voting stage where every clique/cluster member notifies the monitoring node which nodes it believes that are suspicious. The monitoring node then decides which nodes are malicious, based on the votes received from the clique/cluster members and a threshold value. It is worth noting that the authors have assumed that a monitoring node can never be malicious and it is changed periodically in order to prevent unfair use of its resources and battery depletion.

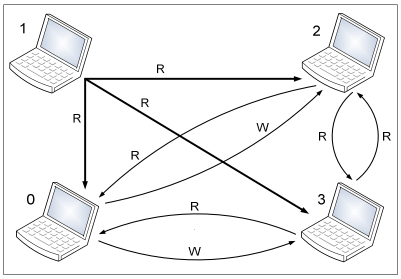

Fig. 5: Clustered IDS architecture using nodes' voting

The main advantage of these architectures is their low processing and communication overhead. This is because both of them avoid using bandwidth or computation intensive operations, such as sharing audit data or deploying anomaly detection algorithms. The only traffic exchanged between the clique/cluster members are the monitoring and voting messages of the detection process. On the other hand, both architectures present some weaknesses, which are analyzed bellow:

- In the performed simulations, the ratio of false positives increased substantially when packet loss reached or exceeded 9% for the ADCLI and 12% for the ADCLU, respectively. Therefore, in an environment that is characterized by high packet loss (e.g., due to high nodes' mobility or the presence of selfish nodes that drop packets), both architectures are ineffective.
- The monitoring node poses a single point of failure in the respective clique/cluster. In case of an attack against the monitoring node or node's failure, the intrusion detection process is disabled.
- Any type of attack that does not modify or drop packets (such as man in the middle, replay, flooding, session hijacking, etc) cannot be detected by these architectures.
- Malicious nodes may exploit the detection scheme by voting legitimate nodes as malicious.

4.4 An optimal hierarchical intrusion detection architecture

Manousakis et al. [33] have proposed a hierarchical IDS architecture that uses a dynamic tree-based structure in which detection data are aggregated upwards, from leaf nodes to authoritative nodes at the root of the hierarchy (i.e., upper layer nodes), and the latter dispatch directives down to the former (i.e., lower-level nodes). The objectives of this architecture are: (i) to form a tree-based structure that is robust to network changes and enables the rapid aggregation of detection data; and (ii) to detect attacks at a level of the hierarchy where enough aggregated detection data are supplied to reach an accurate decision. The tree-based structure is established and maintained using two algorithms: the initial solution generation and the state transition mechanism. The first creates the initial tree-based structure following two steps. In the first step, a network node is randomly selected to serve as a cluster-head and its neighbors are assigned as cluster members to the created cluster. The selected cluster-head represents the highest level of the tree-based hierarchy. In the second step, a cluster-member of the previously formed cluster(s) is selected as cluster-head and its neighbors that have not been previously assigned to another cluster are assigned to it as cluster-members. The second step is repeated until all the network nodes are members of the hierarchical structure.

The state transition mechanism reforms the created tree-based hierarchy, by doing some permutations, in order to be robust to network changes and enable the rapid aggregation of detection data. More specifically, it reassigns some of the branches of the tree structure (i.e., relationships between a cluster-head and cluster members) aiming at two goals: (i) the reformed tree should have the shorter possible height; and (ii) it is estimated that the reformed structure will last longer than any other, considering the nodes' location in the network topology, the nodes' speed, and the range of nodes' transmission. Intrusion detection occurs at the lowest possible level of the hierarchy, at which there are enough aggregated data that allow for an accurate decision. If the responsible cluster-head in a cluster is not capable of detecting an attack accurately, it forwards all the relative detection data to a higher-level cluster-head, which in turn attempts to accurately detect the attack.

The proposed IDS architecture presents the following strengths:

- It is more robust under high nodes' mobility, since clusters are selected with the objective of "lasting longer".
- It provides increased detection accuracy since it supports multiple levels of detection (i.e., compared to other single level detection architectures). The collected data are forwarded upwards until they reach a certain level where intrusion decision can be achieved.

This architecture also presents some weaknesses:

- Lower level cluster-heads are unfairly overloaded, since they constantly perform detections, while higher-level cluster-heads perform detections only in cases that a malicious behavior cannot be resolved at a lower level.
- It adds extra processing and communication overhead in order to create and maintain the hierarchical structure. Moreover, during permutations (i.e., state transition mechanism) the several calculations required are performed at each iteration.
- It is vulnerable to man in the middle and blackmail attacks since the communication channels between the network nodes are not protected. A malicious node may capture and re-transmit modified messages in order to mislead the cluster-head.
- If a cluster-head is compromised, it may provide false administrative directives to the lower-level nodes (i.e., false alarms) and falsely characterize legitimate nodes as

malicious, imposing damage to the network. If the randomly elected node at the highest level of the hierarchy is malicious, it can hinder intrusion detection throughout the entire network.

- A malicious node or set of nodes may exploit the tree optimization procedure in order to elect a malicious node as cluster-head (i.e., by reporting false parameters to the state transmission mechanism). Similarly, a selfish node may avoid becoming a cluster-head.

4.5 Clustered anomaly detection architecture

H. Deng et al. propose a clustered IDS architecture [24] in which only the cluster-heads carry out intrusion detection. It focuses on detecting attacks that target the routing infrastructure of a network and forms clusters using the "Distributed Efficient Clustering Approach" (DECA) protocol. In this protocol, each node votes as cluster-head its neighboring node that has the highest number of connections and residual energy. The nodes with the most votes become cluster-heads. Cluster-heads are re-elected after a predefined period of time. Each cluster-head employs an anomaly detection engine that monitors: (i) the propagation of protocol specific routing packets (i.e., hello, error, request, reply, etc.), (ii) the changes in routing tables, and (iii) the transmission of data packets. These features are monitored either randomly by selecting a cluster member that transmits its own set of features to the cluster head, or actively by configuring the cluster head to listen to the traffic generated in the cluster.

The operational strengths of the clustered anomaly detection architecture can be summarized bellow:

- Processing workload is fairly distributed among the nodes as the cluster-heads rotate after a certain period of time.
- Considering nodes' connectivity in cluster-heads election ensures that the elected cluster-heads monitor large portions of network activities, facilitating IDS to reach more accurate decisions.

The main weaknesses of this IDS architecture are:

- The employed detection engine is only capable of detecting routing attacks.
- The basic weaknesses that appear in previously analyzed hierarchical architectures are also present: (i) cluster-heads may become points of failure; (ii) malicious or selfish nodes that do not cooperate may hinder or mislead intrusion detection; (iii) malicious nodes may falsely accuse other legitimate nodes as malicious; (iv) malicious nodes may exploit the scheme of electing cluster-heads; and (v) the employed election schemes do not take into account the processing capabilities of nodes.

4.6 Strengths and weaknesses of the hierarchical IDS architectures

This section summarizes the basic strengths and weaknesses of the studied hierarchical IDS architectures (see Table 3) that derive from the carried analysis and evaluation, allowing their comparison. In regard to their strengths, we can deduce that: (i) the majority of them attempts to increase the detection accuracy (either by employing multiple layers of detection, or by employing one cluster-head to monitor large portions of a network, or by monitoring the elected cluster-heads); (ii) some of them focus on the fair distribution of the processing workload among nodes (either by considering nodes battery power, or by rotating cluster-heads); and (iii) a few of them try to eliminate the imposed processing and communication overhead (either by employing a detection mechanism based on voting or by selecting cluster-heads with the objective of "last longer"). On the other hand, regarding their weaknesses, it can be realized that: (i) the entire set of the studied hierarchical IDSs is vulnerable to a variety of attacks (i.e., man in the middle, blackmail, exploitation of the employed election scheme, malicious nodes may hinder or mislead detection, etc.); (ii) in the majority of them cluster-heads may become points of failure; (iii) many of them create extra processing and communication overhead because of the creation and maintenance of clustered structures; (iv) in some of them the elected cluster-heads are unfairly overloaded; and (v) a few of them detect only specific types of attacks and are negatively affected by high nodes' mobility.

Table 3. Strengths and weaknesses of the hierarchical IDS architectures

IDS architecture	Strengths	Weaknesses
Cluster-based IDS architecture with adaptive selection event triggering	Nodes with the highest battery power are elected as to serve as cluster-heads.	Nodes elected as cluster-heads are unfairly overloaded
		The ratio of false positives and detection accuracy are negatively affected by high nodes' mobility
		The creation and maintenance of clusters mainly creates extra processing and communication overhead
	Multiple layers of detection provide increased detection accuracy	Cluster-heads may become points of failure.
		It is vulnerable to man in the middle and blackmail attacks
		A malicious node may exploit the election scheme to be elected as cluster-head
Hierarchical IDS architecture that uses a game theoretic detection mechanism	Nodes with the highest battery power are elected as to serve as cluster-heads.	Nodes elected as cluster-heads or checkers are unfairly overloaded
		The creation and maintenance of clusters mainly creates extra processing and communication overhead
		It is vulnerable to man in the middle and blackmail attacks
	Cluster-heads are also monitored for malicious behavior	Cluster-heads may become points of failure.

		A selfish node may avoid being a cluster head.
		A malicious nodes operating as a checker may falsely accuse legitimate cluster-heads for misbehaving
Cluster-based architecture that a uses collective decision detection mechanism	It induces relatively low processing and communication overhead, as it relies on a voting scheme to perform detection.	The ratio of false positives and the detection accuracy are negatively affected by high packet loss and/or high nodes' mobility
		The monitoring node may become point of failure
		It detects only specific types of attacks
		Malicious nodes may exploit the detection scheme by voting legitimate nodes as malicious
Optimal hierarchical IDS architecture	It is more robust under high nodes' mobility as cluster-head are selected with the objective of "last longer"	Lower level cluster-heads are unfairly overloaded
		The creation and maintenance of clusters mainly creates extra processing and communication overhead
		It is vulnerable to man in the middle and blackmail attacks
	Multiple levels of detection provide increased detection accuracy	Compromised cluster-heads may falsely characterize legitimate nodes as malicious
		A malicious node or set of nodes may elect a malicious node as cluster-head hindering or misleading intrusion detection
Clustered anomaly detection architecture	Fair distribution of the processing workload among nodes, as cluster-heads rotate.	It detects only routing attacks
		Cluster-heads may become points of failure
		Malicious or selfish nodes that do not cooperate may hinder or mislead intrusion detection
	The elected cluster-heads monitor large portions of the network activities reaching more accurate decisions	Malicious nodes may falsely accuse other legitimate nodes as malicious
		A malicious node or set of nodes may elect a malicious node as cluster-head hindering or misleading intrusion detection
		The employed election schemes do not take into account the processing capabilities of nodes

5 A comparative evaluation of the IDS architectures

This section provides a comparative evaluation of the studied IDS architectures using a set of critical evaluation metrics, which are elaborated bellow. These metrics derive from: (i) the deployment, architectural, and operational characteristics of MANETs; (ii) the special requirements of intrusion detection in MANETs; and (iii) the carried analysis that reveals the most important strengths and weaknesses of the existing IDS architectures.

5.1 Evaluation metrics

MANETs retain a number of differences from traditional wireless networks. First of all, MANET nodes can be a variety of mobile devices (such as laptops, handheld devices, or mobile phones), which typically rely on the use of battery power and present various computational and bandwidth capabilities. The mobile nature of these nodes creates a dynamic network topology, in which nodes may independently join, leave, or change their position within the network. Moreover, there is no fixed infrastructure that manages the network nodes, routing or any other network operation, and thus, network management is done by the nodes themselves in a cooperative fashion. The nodes that are within radio range may communicate with each other directly (i.e., one-hop communication); or use intermediate nodes (i.e., multi-hop communication). Ad-hoc routing protocols, such as DSR and the Ad-Hoc On-demand Distance Vector (AODV), rely on nodes' cooperation and trust, and thus, do not take into account any security precautions [1][9]. In addition, the absence of access points that connect the nodes to any centralized authority does not leave much room for a clear line of defense or for a high level of trust between nodes. As a result, MANET nodes are susceptible to a variety of attacks, which mainly target the transport, network, and data-link layers of the protocol stack, since these layers are responsible for the most critical functionality of MANETs (i.e., one-hop/multi-hop communication, routing, etc.) [1].

Since MANETs are typically formed by devices with limited processing and communication capabilities, IDSs for MANETs should eliminate the *processing* and *communication overheads* that they impose on the network nodes. Moreover, an IDS should not equally overwhelm network nodes with intrusion detection responsibilities and tasks, since the later may have a variety of available resources. Therefore, IDS architectures have to fairly distribute the processing workload among the network nodes. Finally, as the majority of MANET nodes are mobile, nodes' mobility should not negatively affect the detection accuracy and the ratio of false positives of the IDS.

Regarding security, IDSs for MANETs have to satisfy two main objectives that derive from the definition of IDSs and the operational characteristics of MANETs: (i) detect all possible attacks, and (ii) do not introduce new security vulnerabilities. The first objective is more related to the employed detection engine(s) and less to the IDS architectures that we primarily focus on this paper. The second objective stems from fact that the deployment of new applications/protocols in a MANET should not augment the existing vulnerabilities of the network. However, from the carried analysis and evaluation (see sections 2, 3, 4), we can deduce that the application of the majority of existing IDS architectures for MANETs introduce new security vulnerabilities. These are mainly associated with the employed clustering, data exchange, task assignment, and detection mechanisms, which may be exploited by adversaries and lead to a variety of attacks (i.e., blackmail, man in the middle, byzantine, etc.) that either mislead or hinder intrusion detection.

Based on the above, we infer the following evaluation metrics for MANET IDSs, which we divided into two groups: the first group of metrics relates to *performance* and the second to *security*. The *performance* metrics include: (i) the *processing overhead* imposed by an architecture on each network node, (ii) the imposition of *communication overhead* on the links that connect the network nodes, (iii) the *fair distribution* of the processing workload among the network nodes and (iv) the impact of *nodes' mobility* on the *detection accuracy* and the *ratio of false positives*. The *security* metrics are: (i) the detection of *a limited set of possible attacks*, (ii) the occurrence of *points of failure*, and the vulnerability of an architecture to (iii) *byzantine,* (iv) *man in the middle* and (v) *blackmail* attacks. Sections 5.2 (also see Table 4) and 5.3 (also see Table 5) present the evaluation of the studied IDS architectures with respect to the performance and security metrics, respectively. The performance evaluation takes into account the experimental/simulation results published by the authors of the studied IDS architectures. However, these results are used to justify the advantages and drawbacks of the studied IDSs and not to evaluate their performance on a common basis. Since many details of the proposed algorithms are missing, we could not perform an experimental analysis of our own. Moreover, methods/techniques that have been proposed to improve the performance and the provided security services of the considered architectures are also commented.

5.2 Performance evaluation

It is evident that the *processing overhead,* imposed by the IDS architectures to the underlying network nodes, should be kept to a minimum. However, in almost all of the evaluated architectures, one or more comprehensive detection engines (which are based on signature or anomaly detection) are employed in every node, without considering the limited processing capabilities. Exceptions are: (i) the architecture that is based on social network analysis [14]; and (ii) FORK [19] that distributes the required detection tasks in order to conserve *processing* and *battery* resources. However, both of them impose extra *communication overhead* (another limitation of MANETs), since nodes have to frequently communicate and exchange audit data with each other. Moreover, the employed cooperation process and the mechanism of tasks' distribution create new security vulnerabilities. Finally, in the first architecture the rate of *false positives* and the *detection accuracy* are negatively affected by *nodes' mobility*.

The hierarchical IDS architectures attempt to minimize the *processing overhead* by employing comprehensive or multi-layer detection engines only at some key nodes (i.e., cluster-heads), while the remaining nodes use lightweight engines. However, the creation and maintenance of clustered/hierarchical structures adds extra *processing* load to the network nodes, which increases under conditions of relatively high nodes' mobility. This overhead is produced by the continuous execution of the clustering functionality, due to the constant change of cluster members within a cluster. An exception is the clustered architecture that uses collective decisions for intrusion detection [27], which relies on a voting scheme to perform detections, instead of an anomaly or signature-based engine. Nevertheless, the ratio of false positives in this architecture is negatively affected by packet loss and it cannot detect attacks that do not modify or drop packets.

Stand-alone IDS architectures do not incur any *communication overhead*, since no cooperation between IDSs takes place. However, this characteristic limits them in terms of detection accuracy and the type of attacks that they detect [9]. On the other hand, in both the cooperative and the hierarchical IDS architectures nodes have to exchange alerts, audit data, and detection results that impose extra *communication overhead* to the underlying network. In the cooperative architectures, cooperation and the related overhead takes place only when a suspicious behavior cannot be resolved as malicious using only local audit data. The employment of multiple detection engines per node (either multi-layer detection [16] [22] or multiple detections [18]) attempts to reduce the *communication* overhead, since more attacks are identified locally. However, this approach increases the *processing* workload at each node. Moreover, the exchange of detection results, instead of voluminous audit data, also reduces the *communication* load among nodes [16]. On the other hand, in the hierarchical IDS architectures the *communication* overhead cannot be reduced and takes place when clustered/hierarchical structures are formed, a cluster-head is elected (or re-elected), the cluster members move and change clusters, or a cluster-head and the cluster-members exchange audit data.

The hierarchical architectures impose *unfair workload distribution among the network nodes,* since the nodes elected as cluster-heads are overloaded with detection responsibilities. Election schemes that consider the processing capabilities and battery power of the nodes attempt to establish a fair distribution of detection responsibilities between nodes. Towards this direction, the rotation of cluster-heads also tries to minimize the disparity of the *workload* distribution among the nodes. On the other hand, this increases both the *processing* and the *communication overhead,* since it entails re-elections of the cluster-heads and the conveyance of the related detection information (i.e., audit data and detection results) from the old cluster-heads to the newly elected.

In all the types of IDS architectures (i.e., stand-alone, cooperative, and hierarchical) *nodes' mobility* decreases the *detection accuracy* and increases the *rate of false positives*. Mobility changes the network topology, the clusters' structure, the routing information maintained at each node, the created social and trusted relationships among the nodes, etc., influencing in that way the intrusion detection process. Moreover, a mobile node may move away from its neighboring nodes or from a detection engine that resides in a cluster-head, making cooperation for detection purposes or thorough inspection of the node unavailable. In order to limit the negative impacts of nodes' mobility on intrusion detection, adjustable thresholds have been proposed for stand-alone and cooperative IDS architectures. Moreover, hierarchical structures robust to network changes have been proposed for hierarchical IDS architectures. However, the later increases the *processing* and *communication* overhead to the underlying network. Finally, both of the above mentioned solutions create new security risks, making the respective IDS architectures vulnerable to attacks that are analyzed in the following section.

Table 4. Performance evaluation

Issue		Processing overhead	Communication overhead	Unfair workload distribution	Impacts of nodes' mobility
Stand-Alone	***Problem***	Every node maintains one or more comprehensive engines	N/A	N/A	Decreases the detection accuracy Increases the rate of false positives
	Solution / optimization	-	N/A	N/A	Use of adjustable thresholds
	Open issues	-	N/A	N/A	Adjustable thresholds create new security weaknesses
Cooperative	***Problem***	Every node maintains one or more comprehensive engines	Cooperation and exchange of audit data among neighboring nodes	N/A	Decreases the detection accuracy Increases the rate of false positives
	Solution / optimization	Detection based on social network analysis Distribution of detection tasks among nodes	Use more than one or multi-layer detection engines Exchange of detection results instead of audit data	N/A	Use of adjustable thresholds
	Open issues	Extra communication overhead New security vulnerabilities Social network analysis is negatively affected by nodes' mobility	Multiple or multi-layer engines increase processing overhead The exchange of audit data also imposes communication overhead	N/A	Adjustable thresholds create new security weaknesses
Hierarchical	***Problem***	The creation and maintenance of clustered / hierarchical structures	The formation of clustered / hierarchical structures; The election of cluster-heads The movement of cluster members The exchange of audit data between a cluster-head and the cluster-members	Cluster-heads are unfairly overloaded	Decreases the detection accuracy Increases the rate of false positives Increases the processing and communication overhead
	Solution / optimization	Use of collective decisions for intrusion detection	-	Election schemes that consider the processing and battery power of nodes The rotation of cluster-heads	Use of hierarchical structures that are robust to network changes
	Open issues	Negatively affected by	Communication overhead	The rotation of	The hierarchical

		packet loss It cannot detect attacks that do not modify or drop packets	remains	cluster-heads increases the processing and communication overhead	structures that are robust to network changes create new security risks and increase further the processing and communication overhead

5.3 Security evaluation

The stand-alone IDS architectures detect *a limited set of attacks,* since they rely only on local audit data to resolve malicious behaviors. More specifically, the battery-based IDS [10] detects only the attacks that cause power irregularities; while the threshold-based IDS [12] cannot detect any coordinated attack. On the other hand, the majority of both cooperative and hierarchical architectures are capable of detecting wider sets of possible attacks. This is achieved by employing multiple (or multi-layer) detection engines and by enabling cooperation between neighboring nodes. Exceptions are the routing anomaly detection architecture [20] (cooperative) and the clustered anomaly detection architecture [24] (hierarchical) that only detect routing attacks. Moreover, LIDF [22] (cooperative) only detects attacks that target the network and data link layers; while the cluster architecture that uses collective decision for intrusion detection [27] (hierarchical) only detects attacks that modify or drop packets.

The hierarchical IDS architectures present *points of failure,* since they place the responsibility of intrusion detection in a subset of elected nodes (i.e., cluster-heads). This fact makes these nodes potential targets of attacks, and if an attack succeeds then points of failure occur. Moreover, the hierarchical architectures are *vulnerable to byzantine attacks*. Such an attack can take place during the election phase of a cluster-head, where a number of malicious nodes attempt to elect a malicious node as cluster-head. A malicious cluster-head may hinder intrusion detection or falsely accuse legitimate nodes as malicious. To address such events, the game-theoretic IDS [29] uses randomly selected checker nodes to monitor the cluster-heads for selfish/malicious behaviors. However, such an approach increases the *processing* and *communication* load, since one or more checkers are activated in every cluster. Similarly, the stand-alone architectures are vulnerable to *byzantine attacks*, since attacks against a node by a coordinated group of attackers cannot be determined, due to the lack of cooperation.

Another security weakness that is common for both collaborative architectures (i.e., cooperative and hierarchical) is that they are exposed to *man in the middle* and *blackmail attacks.* Both architectures rely on the exchange of intrusion detection information, either between cooperating nodes or between a cluster-head and the cluster-members, in order to perform detections. This information might be captured, modified, and retransmitted by a malicious node resulting in a *man in the middle attack*. This vulnerability can be avoided by using encryption on the communication links among nodes. Nevertheless, a malicious node may transmit false information when requested upon by a cooperating neighbor or by a cluster-head, resulting in *a blackmail attack*. The friend-assisted IDS architecture [18] counters this vulnerability by deploying a trust mechanism. This mechanism denies cooperation between nodes that have not previously established trusted relationships. Although this solves the problem of blackmail attacks, it is likely to have an impact on the IDS's detection accuracy, especially in case of a network with limited trusted relationships among nodes. In such a scenario, an IDS might not find enough trustful nodes to collect a sufficient amount of audit data to detect an attack.

Table 5. Security evaluation

Issue		**Detection of a limited set of attacks**	**Points of failure**	**Byzantine attack**	**Man in the middle attack**	**Blackmail attack**
Stand-alone	***Problem***	Detects a limited set of attacks due to the lack of cooperation	N/A	Coordinated attacks by multiple attackers are not detectable	N/A	N/A
	Solution / optimization	-	N/A	-	N/A	N/A
	Open issues	-	N/A	-	N/A	N/A
Cooperative	***Problem***	Some solutions detect a limited set of attacks	N/A	N/A	Nodes' communication is susceptible to attacks	A malicious node may transmit false information upon request
	Solution / optimization	Use more than one or multi-layer detection engines	N/A	N/A	Encrypt communication links among nodes	Deploy a trust mechanism
	Open issues	Multiple or multi-layer engines increase processing overhead	N/A	N/A	None	It may decrease the detection accuracy
Hierarchical	***Problem***	Some solutions detect a limited set of attacks	Cluster-heads become targets of attacks	A malicious node may be elected as cluster-head	Nodes' communication is susceptible to attacks	A malicious node may transmit false information upon request
	Solution / optimization	Use more than one or multi-layer detection engines	-	Use randomly selected checker nodes that monitor cluster-heads for malicious behavior	Encrypt communication links among nodes	-
	Open issues	Multiple or multi-layer engines increase processing overhead	Cluster-heads become targets of attacks	The use of checker nodes increases the processing and communication overhead	None	Realization of blackmail attacks

6 Design principles for MANET IDSs

Based on the carried evaluation and comparison, this section presents a set of features and principles, which have to be addressed and satisfied in future research, when designing and implementing IDSs for MANETs. It may not be feasible for an IDS to deal with all of them, but their objective is to stimulate and drive research activities in this area.

IDSs for MANETs should consider the *limited resources* available in them and aim at *limiting the related processing* and *communication overheads*. Although these limitations are common to all the types of application/services deployed in MANETs, they become more critical for IDSs, which require: (a) uninterrupted monitoring of nodes' and network activities; and (b) endless processing of audit data in order to detect malicious behaviors. A possible solution is to assign detection responsibilities to *a subset of network nodes* (i.e., similarly to the hierarchical architectures), instead of operating an individual detection engine at each network node. However, the creation and maintenance of the employed structure (e.g., clustered, tree-based, etc.) *should minimize the imposed extra processing* and *communication overheads.* Moreover, the assigned nodes that perform detection tasks should rotate periodically, avoiding the *unfair workload distribution* among nodes.

The detection engines employed in IDSs for MANETs should use *sophisticated algorithms* that can *detect a variety of possible attacks*, avoiding the introduction of high computational load. A single detection engine should be able to detect attacks *at the three most important layers* (e.g., transport, network and data-link) of the protocol stack, since the majority of attacks in MANETs occur at these layers [1]. If a detection engine focuses only on one layer, then several attacks (i.e., that occur at the other layers) can go undetected. Otherwise, the employment of multiple engines (e.g., one for each layer) is required; but this increases the consumption of the available resources (i.e., battery, processing power, etc). The *more attacks detected locally*, the less communication overhead imposed to the underlying network. Nevertheless, if cooperation is required, then it is better to exchange *detection results*, instead of voluminous audit data.

Nodes' mobility should not negatively affect *the detection accuracy* and *the rate of false positives* of an IDS. However, these negative effects occur in the majority of the existing IDSs, since their architectures primarily inherited from static or mobile networks, which differ radically from MANETs with respect to the network topology. The architecture of an IDS for MANETs should be *independent of the underlying network topology.* Moreover, frequent changes in the topology should not cause repeated and extensive changes to the employed IDS architecture/structure, eliminating thus the execution of the required grouping/formation functionality, as well as the exchange of the related messages. The latter interrupt the intrusion detection process, and increase the processing and communication overhead imposed to the underlying nodes and network.

The deployment of IDSs on MANETs should not *introduce new security vulnerabilities* and *weaknesses*. However, the assignment of detection tasks to a subset of nodes (i.e., cluster-heads) presents *points of failure*, in cases that these nodes become targets of attacks, crash, leave the network, or run out of battery. Moreover, the hierarchical/clustering and election algorithms used may be exploited by adversaries, either hindering or misleading intrusion detection. Similarly, the cooperation/communication among detection engines may be captured or modified. Therefore, the IDSs for MANETs should be *fault/attack tolerant*, and the nodes assigned with detection responsibilities should be *robust, expendable, and replaceable*. In addition, the employed algorithms and cooperation between detection engines should *be resilient to attacks*.

Finally, a proposed IDS should be evaluated and tested *under realistic conditions,* which include *a variety of nodes' mobility scenarios* and *type of attacks*, helping us to deduce certain results about its effectiveness and accuracy. Table 6 summarizes proposed design principles of IDS for MANETs grouped by the related MANETs characteristics.

Table 6. Proposed design principles of IDS for MANETs grouped by the related MANETs characteristics

Characteristics of MANETs	***Proposed design principles***
Limited available resources (processing power, bandwidth, battery power)	Assign intrusion detection responsibilities to a subset of nodes, instead of operating an individual detection engine at each node
	The employed clustering or hierarchical algorithms should minimize the imposed processing and communication overheads
	Detection engines should not introduce high computational overhead
	Avoid multiple detection engines at each node
	Detect most attacks locally at a node
	If required, exchange detection results instead of audit data
Diverse range of devices	Fair workload distribution among nodes
Dynamic topology/node's mobility	The architecture of an IDS should be independent of the underlying network topology
Susceptible to a variety of attacks	A single sophisticated engine should detect a wide range of possible attacks at the tree most important protocol layers (i.e., transport, network and data link)
	IDSs should be fault/attack tolerant
	The nodes assigned with detection responsibilities should be robust, expendable, and replaceable

	A proposed IDS should be evaluated and tested under realistic conditions

7 Conclusions

IDSs for MANETs have attracted much attention recently and thus, there are many publications that propose new IDS solutions or improvements to the existing, focusing on both IDS architectures and detection engines. This paper has evaluated and compared the latest and most prominent IDS architectures for MANETs, classified as: (i) stand-alone, (ii) cooperative, and (ii) hierarchical. Based on the carried analysis, it can be deduced that the existing IDS architectures for MANETs present significant limitations and weaknesses. This mainly occurs since the majority of the IDS architectures is inherited from static or mobile networks, which differ radically from MANETs with respect to the network topology, available resources and variety of nodes, nodes' mobility, security vulnerabilities and possible attacks. The studied IDS architectures were comparatively evaluated using a set of performance and security metrics. It was concluded that all types (i.e., stand-alone, cooperative, and hierarchical) strain the limited processing and energy power of the nodes. Moreover, both the cooperative and the hierarchical architectures deplete the scarce bandwidth resources of the network. The detection accuracy and the ratio of false positives of the IDSs are negatively affected by nodes' mobility, encountered in MANETs. In addition, many of them are vulnerable to security attacks, which might: (i) hinder the network operation and the intrusion detection process, (ii) mislead detection, or (iii) falsely characterize legitimate nodes as malicious. Finally, some of the evaluated IDS architectures cannot detect all types of attacks, since they focus only on specific types of intrusions.

References

[1] D. Djenouri, L. Khelladi, N. Badache, "A Survey of Security Issues in Mobile Ad Hoc Networks," IEEE Communications Surveys, Vol. 7, No. 4, Fourth Quarter 2005.

[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] T. Anantvalee, J. Wu, "A Survey on Intrusion Detection in Mobile Ad Hoc Networks," Wireless/Mobile Network Security, Springer, Chapter 7, pp. 170 - 196, 2006.

[4] P. Brutch, C. Ko, "Challenges in Intrusion Detection for Wireless Ad-hoc Networks," Symposium on Applications and the Internet Workshops (SAINT'03 Workshops), pp. 368, 2003.

[5] B. Sun, L. Osborne, X. Yang, S. Guizani, "Intrusion Detection Techniques in Mobile Ad Hoc and Wireless Sensor Networks," IEEE Wireless Communications, vol. 14, issue 5, pp. 56-63, Oct. 2007.

[6] M. A. Azer, S. M. El-Kassas, and M. S. El-Soudani, "A Survey on Anomaly Detection Methods for Ad hoc Networks," Ubiquitous Computing and Communication Journal, vol. 2, issue 3, pp. 67–76, 2005.

[7] Y. Li, and J. Wei, “Guidelines on Selecting Intrusion Detection Methods in MANET”, The 21st Annual Conference for Information Systems Educators (ISECON), Rhode Island, USA, 4-7 November, 2004.

[8] S. Mandala, A. Ngadi, A.H. Abdullah “A Survey on MANETs Intrusion Detection,” International Journal of Computer Science and Security, vol. 2 issue: 1, pp. 1-11, August, 2007.

[9] S. Sen and J. A. Clark, "Intrusion Detection in Mobile Ad Hoc Networks". In: Guide to Wireless Ad Hoc Networks, S. Misra, I. Woungang, S.C. Misra (Eds.), Springer, 2009.

[10] G.A. Jacoby, N.J. Davis, "Mobile Host-Based intrusion Detection and Attack Identification," IEEE Wireless Communications, vol. 14, issue 4, pp. 53-60, August 2007.

[11] SANS Inst., “The Twenty Most Critical Internet Security Vulnerabilities,” http://www.sans.org/top20/, last accessed Jan. 04, 2010.

[12] K. Nadkarni, A. Mishra, "A Novel Intrusion Detection Approach for Wireless Ad Hoc Networks," IEEE Wireless Communications and Networking Conference (WCNC. 2004), vol. 2, pp. 831 – 836, March 2004.

[13] The Network Simulator (ns-2), http://www.isi.edu/nsnam/ns/, last accessed Jan. 04, 2010.

[14] W. Wang, H. Man, Y. Liu, “A Framework for Intrusion Detection Systems by Social Network Analysis Methods in Ad Hoc Networks.” Wiley Security and Communication Networks, vol. 2, issue 6, pp. 669 – 685, April, 2009.

[15] A. Lauf, R. A. Peters, W. H. Robinson, "A Distributed Intrusion Detection System for Resource-Constrained Devices in Ad Hoc Networks". Elsevier Journal of Ad Hoc Networks, vol. 8, issue 3, pp. 253-266, May 2010.

[16] S. Bose, S. Bharathimurugan, A. Kannan, “Multi-Layer Integrated Anomaly Intrusion Detection System for Mobile Ad Hoc Networks,” IEEE ICSCN 2007, MIT Campus, Anna University, Chennai, India, pp.360-365, February 2007.

[17] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a Library for Parallel Simulation of Large-Scale Wireless Networks,” Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS ’98), Banff, Canada, pp. 154-161, May, 1998.

[18] S.A. Razak, S.M. Furnell, N.L. Clarke, P.J. Brooke, “Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks,“ Elsevier Ad Hoc Networks, vol. 6, issue 7, pp. 1151 – 1167, September 2008.

[19] C. Ramachandran, S. Misra, M. Obaidat, “FORK: A novel two-pronged strategy for an agent-based intrusion detection scheme in ad-hoc networks,” Elsevier Computer Communications, vol. 31, issue 16, Performance Evaluation of Communication Networks (SPECTS 2007), pp. 3855-3869, October 2008.

[20] B. Sun, K. Wu, U. W. Pooch, "Routing anomaly detection in mobile ad hoc networks," IEEE International Conference on Computer Communications and Networks (ICCCN’03), pp. 25-31, 2003.

[21] B. Sun, K. Wu, Y. Xiao, R. Wang, “Integration of mobility and intrusion detection for wireless ad hoc networks,” Wiley International Journal of Communication Systems, vol. 20, issue 6, pp. 695 – 721, June 2007.

[22] N. Komninos, C. Douligeris, “LIDF: Layered intrusion detection framework for ad-hoc networks,” Elsevier Ad Hoc Networks, vol. 7, issue 1, pp. 171 – 182, January 2009.

[23] 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), January 2003.

[24] H. Deng, R. Xu, J. Li, F. Zhang, R. Levy, W. Lee, “Agent-based cooperative anomaly detection for wireless ad hoc networks,” Proceedings of the 12th Conference on Parallel and Distributed Systems, pp. 613-620, 2006.

[25] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, “A cluster-based approach for routing in dynamic networks,” ACM SIGCOMM Computer Communication Review, vol. 27, no. 2, pp. 49-65, April 1997.

[26] J. Li, M. Yu and R. Levy, “A distributed efficient clustering approach for ad hoc and sensor networks,” Proceedings of the International Conference on Mobile Ad-Hoc and Sensor Networks, 2005.

[27] N. Marchang, R. Datta, "Collaborative techniques for intrusion detection in mobile ad hoc networks," Elsevier Ad Hoc Networks, vol. 6, issue 4, pp. 508 – 523, June 2008.

[28] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C-Y. Tseng, T. Bowen, K. Levitt, J. Rowe, “A general cooperative intrusion detection architecture for MANETs,” Proceedings of the third IEEE International Workshop on Information Assurance, pp. 57 – 70, 2005.

[29] H. Otrok, N. Mohammed, L. Wang, M. Debbabi, P. Bhattacharya, “A game-theoretic intrusion detection model for mobile ad hoc networks,” Elsevier Computer Communications vol. 31, issue 4, pp. 708-721, March 2008.

[30] D. B. Roy, R. Chaki, N. Chaki, “BHIDS: a new, cluster based algorithm for black-hole IDS,” Wiley Security and Communication Networks, available online: September 21, 2009.

[31] Chuan-xiang Ma, Ze-ming Fang, "A novel intrusion detection architecture based on adaptive selection event triggering for mobile ad-hoc networks," IEEE Second International Symposium on Intelligent Information Technology and Security Informatics, pp.198-201, January 2009.

[32] S. Jha, K. Tan, and R. Maxion, “Markov chains, classifiers, and intrusion detection,” Proceedings of 14th IEEE Computer Security Foundations Workshop, Cape Breton, Nova Scotia, Canada, pp. 206-219, 2001.

[33] K. Manousakis, D. Sterne, N. Ivanic, G. Lawler, A. McAuley, “A stochastic approximation approach for improving intrusion detection data fusion structures,” IEEE Military Communications Conference (MILCOM 2008), San Diego, CA, pp. 1-7, November 2008.