An Evaluation of Anomaly-based Intrusion Detection Engines for Mobile Ad Hoc Networks

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

[1]Department of Informatics & Telecommunications, University of Athens, Greece,
[2]Department of Digital Systems, University of Piraeus, Greece,
cpanos@di.uoa.gr, xenakis@unipi.gr, ioannis@di.uoa.gr

Abstract. Mobile Ad Hoc Networks are susceptible to a variety of attacks that threaten their operation and the provided services. Intrusion Detection Systems may act as defensive mechanisms, since they monitor network activities in order to detect malicious actions performed by intruders. Anomaly-based detection engines are a topic of ongoing interest in the research community, due to their advantage in detecting unknown attacks. However, this advantage is offset by a number of limitations such as high rates of false alarms, imposition of processing overhead, lack of adaptability under dynamic network conditions etc. This paper presents a comprehensive evaluation and comparison of the most recent literature in the area of anomaly detection for MANETs. The provided weaknesses and limitations, which are thoroughly examined in this paper, constitute open issues in the area of MANET security and will drive future research steps.

Keywords: Intrusion detection system, IDS engines, mobile ad hoc networks, MANETs, security, security attacks, anomaly-based detection, 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 and cooperative fashion. Their wireless and mobile nature in conjunction with the absence of access to a centralized authority makes them susceptible to a variety of attacks [1]. An effective way to identify whether an attack occurs in a MANET is the deployment of an Intrusion Detection System (IDS). An IDS monitors network activities and utilizes one or more detection engines, which determine if the monitored activity corresponds to a malicious or legitimate behavior. The detection engines can be classified into three main categories [2]: (i) signature-based engines, which rely on a predefined set of patterns to identify attacks; (ii) 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; and (iii) anomaly-based engines, which rely on particular models

(i.e., normal profiles) of nodes' behavior and mark nodes that deviate from these models as malicious.

In general, anomaly-based detection consists of two phases: the training phase and the monitoring phase. During the training phase, which can be performed either offline (i.e., the network operation is simulated in a controlled environment, without actually deploying a MANET) or online (i.e., during the actual deployment of the MANET), the normal profile is created. Subsequently, during the monitoring phase, the engine monitors a set of carried activities (i.e., features) and compares them against the normal profile. The variation between them (i.e., monitored features and normal profile) is usually determined by utilizing statistical analysis, machine learning, or data mining techniques.

The majority of IDS literature in MANETs focuses on anomaly-based detection, due to its advantage in detecting unknown attacks. However, this advantage is offset by a number of limitations such as high rates of false alarms, imposition of processing overhead, lack of adaptability under dynamic network conditions, etc. These limitations stem from the fact that these engines were primarily inherited from static or mobile networks, which differ radically from MANETs. A number of recent publications attempt to address these limitations, through the introduction of several new mechanisms. On the other hand, little work has been done in evaluating and comparing these new approaches in anomaly detection. Existing surveying papers such as [16][17][18][19][20][21], either focus on outdated solutions, or mainly examine the architectural part of the studied IDSs and do not provide an analysis or evaluation of the deployed detection engines.

This paper presents a comprehensive analysis and evaluation of the most recent literature in the area of anomaly-based detection for MANETs. The works selected for evaluation introduce new mechanisms in anomaly-based detection, aiming to resolve existing limitations. For each evaluated detection engine, its functionality is considered and outlined as well as its advantages and weaknesses are elaborated. Furthermore, a comparison of the evaluated engines is performed using some critical evaluation metrics. These metrics derive from: (i) the deployment, architectural, and operational characteristics of MANETs; (ii) the functionality of anomaly-based detection; and (iii) the carried analysis that reveals the most important strengths as well as the limitations and weaknesses of the considered engines. The provided weaknesses and limitations, which are thoroughly examined in this paper, constitute open issues in the area of MANET security and will drive next research steps.

The rest of this article is organized as follows. In section 2, the selected anomaly-based detection engines for MANETs are analyzed and commented. Section 3, presents a comparative evaluation of the considered engines and finally, section 4 contains the conclusions.

2 Anomaly-based Detection Engines for MANETs

This section presents and analyses the most recent anomaly-based detection engines that have been proposed for MANETs. For each engine, the basic functionality is outlined as well as the provided advantages and weaknesses are elaborated.

2.1 A Dynamic Anomaly Detection Scheme for AODV-Based MANETs

Nakayama et al. [3] have proposed an anomaly-based engine for detecting malicious actions that target the Ad-hoc On-demand Distance Vector (AODV) [14] routing protocol. The proposed engine utilizes machine learning in order to generate and maintain a normal profile and relies on principal component analysis (PCA) [4] for resolving malicious behaviors. PCA has been widely used in image compression and pattern recognition. It transforms n correlated random variables into $d \leq n$ uncorrelated variables. The uncorrelated variables (i.e., principal components) are linear combinations of the original variables and can be used to express the data in a reduced form.

In the proposed engine, an offline training phase is required to generate the initial normal profile. During this phase, N simulated nodes are monitored and a set of training data is collected, which subsequently forms the normal profile. Then, during the monitoring phase, the engine records a set of features (i.e., monitored data) from the network layer (e.g., route control packets, sender and destination information, etc.) in fixed-time intervals of five seconds. The recorded data are transformed into a *p – dimension vector*, where p is the number of monitored features. In the sequel, using PCA on the normal profile, the first principal component is calculated, which reflects an approximate distribution of the normal profile. The first principal component is the linear combination of the original variables with the largest variance. On the other hand, by applying PCA on the collected data of the first monitored time slot, the deviation from the first principal component can be estimated. If this deviation exceeds a threshold M, the engine assumes that an attack takes place. Otherwise, the recorded data from the monitored time slot becomes the new normal profile. According to the authors, the computational complexity of this engine is $O(m_n \times p^2)$, where m_n represents the training data set for n monitored nodes and p is the number of monitored features.

The most important strength of this engine is the low rate of false positive alarms caused by dynamic network changes. This is achieved by dynamically updating the normal profile at runtime. However, this strength also causes the most important limitation of the engine. If, for example, in a monitored time slot the engine fails to detect a malicious behavior, while an attack(s) takes place (i.e., false negative) then, the attack(s) will become part of the normal profile. As a result, the attack(s) will remain undetected until the normal profile is updated again. In addition, updating the normal profile induces extra processing overhead, since the PCA has to be re-applied to the new normal profile. Another limitation results from the use of fixed-time monitoring slots, since the engine does not take advantage of correlations between features at nearby time slots. Finally, the proposed engine cannot be used to detect all the types of possible attacks, as it monitors features only at the network layer.

2.2 Cross-layer Detection of Sinkhole Attacks in MANETs

J. Felix et al. [5] have proposed an anomaly-based engine for detecting sinking attacks (i.e., nodes that do not cooperate in the routing and forwarding operations of a network) in MANETs. The proposed engine utilizes a support vector machine (SVM)

[6] classifier in order to distinguish malicious behaviors. SVM is a non-probabilistic binary linear classifier, which, given a training sample, builds a model that decides whether a new example falls within the same category as the training sample or not. According to the authors, the training process of the SVM has a computational complexity of $O(N^3)$, where N represents the number of training samples [5].

During the training phase, which takes place offline at a system with abundant resources [5], data are collected from the physical, medium access control (MAC) and network layers. Then, the collected training data are pre-processed using a data reduction process, which aims at reducing their size in order to be processed by SVM. The employed data reduction process includes three steps:

1. *Association*: collected data from different layers are correlated for associations, so that the number of features can be reduced.
2. *Feedback-Based Filtering*: uninformative and redundant features are removed.
3. *Feedback-Based Sampling*: data are further reduced by randomly selecting a subset of the original training data.

The training phase concludes with the application of SVM classifier on the reduced training data set. This produces a linear decision function, which is then used during the monitoring phase to resolve if a monitored event is legitimate or the result of a sinking attack.

The most important strength of this engine is the use of features from multiple layers, which may lead to increased detection accuracy. However, the application of data reduction process outweighs this advantage, since only 5 – 9 % of the original data features are used for training [5]. Usually, data reduction is used in engines that include online training, in order to conserve resources. On the contrary, the proposed engine uses offline training, which means that there are no limitations of resources. During training, the considered engine employs data reduction in order to make computationally feasible the use of the SVM classifier in MANETs, limiting in that way the information gain from the collected multi-layer data and thus, the associated advantages.

2.3 A Two-stage Anomaly Detection Engine for MANETs

Adrian Lauf et al. [7] have proposed a two-stage, anomaly-based detection engine that aims at operating in resource-constrained environments such as MANETs. The proposed engine can be divided into two stages: in the first stage detection is performed by the maxima detection system (MDS), while in the second by the cross-correlative detection system (CCDS). MDS is used to rapidly identify a potential threat as well as to calibrate a threshold for CCDS, while CCDS is used to accurately detect the source(s) of threat, as well as to detect multiple attacks simultaneously.

During the training phase, a normal profile is created offline. The monitored set of features consists of a set of application-level interactions, each of which corresponds to a specific function or behavior of the normal network operation. In the monitoring phase, MDS is deployed initially, which monitors and logs all application interactions in a history table. Then, MDS performs an analysis of global and local maxima in the probability density functions (PDF) of the monitored behaviors to isolate deviations

from the normal profile. If a deviation is detected, MDS traverses the history table to locate the node that statistically has the greatest contribution to the local maximum in the PDF and then calls CCDS.

CCDS performs detection by calculating individual PDFs for each node (based on data from the history log) and comparing them to a threshold. However, the threshold must be first calibrated through a transition period, before accurate detection can be performed. Initially, (transition period) the threshold of CCDS is set up to represent 100% deviation and then, both MDS and CCDS run simultaneously. If a suspected node is detected by MDS, CCDS check whether this node is included in the set of deviant nodes it detects too. If it is not, the corresponding threshold for the CCDS is reduced. This calibration procedure (i.e., indication of malicious behaviors, check MDS and CCDS results, and threshold adjustments) is repeated until there is a match between MDS and CCDS. If this is the case, the transition period (threshold calibration) ends, and the CCDS, properly calibrated, starts operating as described by the engine's specifications.

The proposed engine minimizes the consumption of resource, as it mainly employs the lightweight MDS detection mechanism, while the more computationally demanding CCDS is only executed when needed. It may also provide increased detection accuracy, compared to other single detection engines, because the two employed detection mechanisms supplement each other. However, the MDS feedback in the calibration of the CCDS threshold may result in improper threshold's tuning and thus, reduced overall detection accuracy. This is due to the fact that different attacks may present different application-level behaviors and thus, a single attack detected by MDS cannot be used to set up a generic/cumulative threshold in CCDS. Finally, the proposed engine is prone to high rates of false positives in cases that dynamic changes on the network occur, since the normal profile of MDS is static.

2.4 Anomaly Detection Engine with Optimal Features

P. Kabiri et al. [8] have proposed an anomaly-based engine that focuses on detecting denial of service attacks (DoS). The proposed engine shares a number of similarities with [3] analyzed in sect. 2.1. More specifically, it utilizes machine learning to generate and maintain a normal profile, and relies on PCA for resolving malicious behaviors. However, in the considered engine, the training phase (which takes place online) builds one normal profile for each neighboring node. Furthermore, the monitored features used by the engine are selected after evaluation, which reveals the features with the highest information gain in detecting DoS attacks.

The most important strength of this engine is that it limits the overhead of gathering and processing data, by using a set of optimal features, since it performs the training process online. However, the authors do not clarify how those data, which are collected during the training process, represent a node's normal operation. If dynamic changes in the network occur, the engine is prone to high rates of false positives (or even no detection at all) as well as presents increased processing and memory overhead. This is because the list of neighbors constantly changes, forcing the detection engine to build a new normal profile for each new neighboring node, without enough time to complete the training phase. Another limitation results from

the use of fixed-time monitoring slots, and thus, the engine does not take advantage of correlations between features at nearby time slots.

2.5 Adaptive Anomaly Detection of Denial of Service Attacks

A. Nadeem and M. Howarth [9] have proposed an anomaly-based engine for detecting DoS attacks in MANETs. The proposed engine utilizes a dynamic normal profile and relies on statistical analysis for resolving malicious behaviors. In the training phase, which takes place after network initialization, the engine counts incoming route request packets and calculates the probability distribution of the collected data. The authors assume that during training the behavior of a newly created network is free of anomalies. Subsequently, during the monitoring phase, the engine: (a) logs incoming route request packets, in five-second intervals; (b) calculates the probability distribution of the collected data; and (c) compares it with that of the normal profile, using the chi-square test [10]. If the distribution of the collected data does not fit the normal profile then, the observed behavior is considered suspicious. Whenever a suspicious behavior is detected, a counter is incremented and the node responsible for the symptom is marked as suspicious. If the incident repeats and the counter exceeds a threshold value within a fixed time window, the node from where the incident originates is labeled as malicious. Finally, if no suspicious behavior is detected within the monitored time interval, the collected data become the new normal profile.

The most important strength of this engine is the low rate of false positive alarms caused by dynamic network changes. This is achieved by dynamically updating the normal profile at runtime and employing a threshold mechanism in which only recurring malicious behaviors are considered as attacks. However, similarly to [3] (see sect. 2.1), if in a monitored time slot the detection engine fails to detect a malicious behavior, while an attack(s) takes place (i.e., false negative) then, the attack(s) will become part of the normal profile. As a result, the attack(s) will remain undetected until the normal profile is updated again. The authors assume that during initialization the network is free of attacks, but this assumption can be considered misleading. Furthermore, the online execution of the training phase induces extra processing overhead. Malicious nodes may attempt to exploit the threshold mechanism by performing sporadic attacks considering not exceeding the threshold values and raising alarms. Another limitation arises from the use of a fixed-time monitoring slot, since the engine does not take advantage of correlations between features at nearby time slots. Finally, the proposed engine is only capable of detecting DoS attacks.

3 Comparative Evaluation

This section provides a comparative evaluation of the studied anomaly-based detection engines for MANETS using some critical evaluation metrics. These metrics derive from: (i) the deployment, architectural, and operational characteristics of MANETs; (ii) the functionality of anomaly-based detection; and (iii) the carried

analysis that reveals the most important strengths, weaknesses and limitations of the latest anomaly-based detection engines for MANETs (see table 1).

MANETs retain a number of differences from traditional wireless networks. First, 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, memory, and bandwidth capabilities. The mobile nature of those nodes creates dynamic network topologies, in which nodes may independently join, leave or change their position. Moreover, 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.) [2].

On the other hand, anomaly-based detection requires the execution of: (i) a training phase in which the normal profile is created; and (ii) a monitoring phase in which malicious behaviors are resolved. The training phase can take place either online or offline and the resulting normal profile can be static or dynamic. During the monitoring phase, the features, collected in fixed-time intervals, indicate the type and range of malicious behaviors and actions, detected by the engine.

It is evident that anomaly-based detection engines for MANETs have to be *adaptable to dynamic network changes*, which means that their normal profile should always represent the normal network operation. However, in [7] and [9] the normal profile is static including only the initial conditions of the network when the normal profile is created. This may lead to high rates of false positives when dynamic changes on the network occur, since these changes are not incorporated into the normal profile and therefore, are falsely considered as results of malicious behaviors. In order to address this limitation, several detection engines [3][5][8] utilize dynamically updated normal profiles, which attempt to reduce the rate of false positive alarms, caused by dynamic network changes. However, this approach also creates an important limitation: if in a monitored time slot the detection engine fails to detect a malicious behavior, while an attack(s) takes place (i.e., false negative) then, the attack(s) will become part of the normal profile. Thus, these attack(s) will be undetected.

Table 1. A summary of the studied anomaly-based detection engines

IDS engine	Methodology	Strengths	Weaknesses
Dynamic detection for AODV	Dynamic profile using PCA analysis	Adaptability to network changes	False negatives become part of the normal profile
			Induces extra processing overhead
			Monitors a fixed time slot
			Cannot detect all possible attacks
Cross-layer detection of	Cross-layer data reduction and	Cross-layer monitoring	No benefits from the data reduction process

sinkhole attacks	use of SVM classifier		Can only detect sinking attacks
Two-stage IDS	Scalable use of two detection engines (MDS and CCDS)	Increased detection accuracy by employing two detection engines	The ratio of false positives and detection accuracy are negatively affected by high nodes' mobility
		Scalability	There is no way to adjust an improperly tuned threshold
		Incurs less processing overhead	
Optimal feature based IDS	Dynamic profile using PCA analysis	Uses an optimal set of features	The ratio of false positives and detection accuracy are negatively affected by high nodes' mobility
		Incurs less processing overhead	Monitors a fixed time slot
			Can only detect DoS attacks
Adaptive IDS	Dynamic profile using statistical analysis	Adaptability to network changes	False negatives become part of the normal profile
			Monitors a fixed time slot
			Malicious nodes may attempt to exploit the threshold mechanism
			Incurs extra processing overhead
			Can only detect DoS attacks

Another approach to address the limitation of the static normal profiles is through the use of thresholds [7][9]. In this approach, periodic symptoms of suspicious behaviors, mainly caused by network topology changes, remain under the detection thresholds; while malicious behaviors that are constant exceed the thresholds indicating the occurrence of attacks. Nevertheless, the use of thresholds introduces new security weaknesses, since malicious nodes may exploit this mechanism by performing an attack(s) considering not exceeding the threshold values and raising alarms.

Since MANETs are typically formed by devices with limited processing and communication capabilities, the *processing overhead* imposed by the detection engines to the underlying network nodes should be kept to a minimum. However, the majority of the evaluated detection engines induce computational overhead of approximately polynomial-time complexity [3][5][8]. Therefore, their deployment is computationally feasible only if the set of monitored features is extensively reduced. As a result, the detection engines are only *capable of detecting a limited set of possible attacks*, and thus, do not constitute comprehensive security solutions. An

exception is the two-stage detection engine [7], which attempts to minimize the processing overhead through a scalable detection mechanism. Finally, an approach to further reduce the computational overhead of the detection engine is proposed in [8]. In this approach, the set of monitored features is reduced through an evaluation process, which reveals the features with the highest information gain in detecting DoS attacks.

Summarizing, we can deduce that that the considered anomaly-based detection engines for MANETs share a number of limitations (see table 1). In particular, the majority of them have not resolved the issue of high rates of false positives under conditions of high nodes' mobility. Furthermore, in all of them (except for [7]) it is computationally infeasible to process a broad set of features, and thus, they are only capable of detecting a limited type and range of attacks. Finally, the engines that employ online training impose computational overhead to the network nodes.

In future work, more research effort should be given in the optimization of existing anomaly detection algorithms, as well as the introduction of new ones, which will enable the monitoring of larger sets of features. Furthermore, a number of limitations presented in detection engines might be addressed by utilizing the characteristics of the employed IDS architectures. For example, in a signature-based detection engine, the distribution and maintenance of a signature database under a MANET environment is a difficult task, due to the network's unique characteristics. However, Sterne et al. [22] have proposed an IDS, based on a hierarchical architecture, that addresses this limitation. In the proposed scheme, the detection engine takes advantage of the hierarchical IDS architecture in order to efficiently distribute and update signatures.

Specification-based detection engines constitute another promising alternative to anomaly-based detection. They are capable of detecting both known and unknown attacks, and they avoid high rates of false alarms, since they do not rely on normal profiles, as happens in anomaly detection. However, the development of specifications for an engine might be a lengthy and convoluted process, since the developer has to determine what is the expected behavior of each individual application and protocol, and then, develop constrains that characterize this behavior. Therefore, specification-based engines for MANETs have seen limited use, as they are employed to monitor only the network layer for routing attacks [11][12][13]. Nevertheless, the required overhead of developing specification can be reduced, since the un-hindered operation of MANETs relies on a specific set of protocols at the transport, network and data-link layer, where the majority of security attacks occur [1]. Moreover, aggregated specifications may be developed exploiting cross-layer features among the transport, network and link layer that provide the main functionality of MANETs. Finally, another possibility that should be explored is the development of hybrid detection engines that combine the advantages of more than one type of engines, aiming to eliminate the related drawbacks. This will be facilitated if we consider the special deployment and operational characteristics of MANETs, as well as the attacks that target them.

4 Conclusions

Intrusion detection algorithms 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 presented a comprehensive evaluation of the most recent literature in the area of anomaly detection for MANETs. For each of the considered engines, its functionality was outlined as well as its advantages and weaknesses were elaborated. Furthermore, the studied detection engines were comparatively evaluated based on the following evaluation metrics: (i) the adaptability to dynamic network changes, (ii) the imposition of processing overhead, and (iii) the type and range of possible attacks that they detect. These metrics were derived from the deployment, architectural, and operational characteristics of MANETs; the functionality of anomaly detection; and the carried analysis. The evaluation revealed that the most recent anomaly-based detection engines for MANETs still present significant limitations and weaknesses. In particular, the majority of them rely on a limited set of features in order to make their deployment computationally feasible on MANETs. Therefore, they detect a limited type and range of attacks. The detection accuracy of several proposed engines is negatively affected by nodes' mobility, encountered in MANETs. This can be addressed through the use of dynamic normal profiles and thresholds. However, these solutions may be exploited by malicious nodes allowing for attacks to remain undetected. Future research endeavors might address these limitations if they achieve a reduction in the computational complexity of anomaly detection algorithms. Other directions that can be followed include the utilization of the employed IDS architectures, the shift of development to other promising detection approaches (such as specification-based detection), and the use of hybrid detection schemes that attempt to combine the advantages of different detection engines.

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] C. Xenakis, C. Panos, I. Stavrakakis, "A comparative evaluation of intrusion detection architectures for mobile ad hoc networks," Computers & Security, Volume 30, Issue 1, January 2011.

[3] H., Nakayama, S., Kurosawa, A., Jamalipour, Y., Nemoto, N., Kato, "A Dynamic Anomaly Detection Scheme for AODV-Based Mobile Ad Hoc Networks," Vehicular Technology, IEEE Transactions on , vol.58, no.5, pp.2471-2481, Jun 2009.

[4] R. Duda, P. Hart, and D. Stork, "Pattern Classification and Scene Analysis," New York: Wiley, 1973.

[5] J.F.C., Joseph, Bu-Sung Lee, A., Das, Boon-Chong Seet, "Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA," Dependable

and Secure Computing, IEEE Transactions on , vol.8, no.2, pp.233-245, March-April 2011.

[6] C. Nello and S.-T. John, “An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods,” Cambridge Univ. Press, 2000.

[7] 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.

[8] P. Kabiri and M. Aghaei, “Feature Analysis for Intrusion Detection in Mobile Ad-hoc Networks,” International Journal of Network Security, Vol.12, No.2, PP.80–87, Mar. 2011.

[9] A., Nadeem, M., Howarth, “Adaptive intrusion detection and prevention of denial of service attacks in MANETs,” in International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly (pp. 926–930). Leipzig, Germany, 2009.

[10] H.O.Lancaster, “The Chi-Squared Distribution”, Wiley Publications in Statistics 1969.

[11] C.-Y. Tseng. et al., “A specification-based intrusion detection system for AODV,” in In Proc. Of ACM Workshop on Security of ad hoc and sensor networks, 2003.

[12] C. H. Tseng, T. Song, P. Balasubramanyam, C. Ko, K. Levitt, “A Specification-based Intrusion Detection Model for OLSR”, Proceeding of the 8th International Symposium, RAID 2005, Recent Advances in Intrusion Detection, Seattle, WA, September 7-9, 2005.

[13] H. Hassan, M. Mahmoud, S. El-Kassas, “Securing the AODV protocol using specification-based intrusion detection,” Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks, Terromolinos, Spain, 2006.

[14] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” Jul. 2003. IETF RFC 3561.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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, 2007.