AN EFFICIENT ATTACK DETECTION SYSTEM FOR MOBILE AD-HOC NETWORK

First A. Pushpraj Patel, Department of C.S.E., Shri Vaishnav Institute of Technology & Science, India,
E-mail:- patel.pushpraj@gmail.com[1];
Second B. Tejpal singh, Department of Information Technology, Technocrats Institute of Technology Bhopal, India,
E-mail:- tejpal1985@gmail.com [2];

Abstract

A mobile ad hoc network (MANET) is a wireless network that does not rely on any fixed infrastructure (i.e., routing facilities, such as wired networks and access points), and whose nodes must coordinate among themselves to determine connectivity and routing. The traditional way of protecting networks is not directly applicable to MANETs. Many conventional security solutions are ineffective and inefficient for the highly dynamic and resource-constrained environments where MANET use might be expected. Since prevention techniques are never enough, intrusion detection systems (IDSs), which monitor system activities and detect intrusions, are generally used to complement other security mechanisms. How to detect intrusions effectively and efficiently on this highly dynamic, distributed and resource-constrained environment is a challenging research problem. In this paper, we investigate the use of evolutionary computation techniques for synthesizing intrusion detection programs on MANETs. We evolve programs to detect the following attacks against MANETs: dropping attacks and power consumption attack. The proposed system is a novel architecture that uses knowledge-based intrusion detection techniques to detect the attacks that an adversary can perform against the routing fabric of mobile ad hoc networks. Moreover, the system is designed to take countermeasures to minimize the effectiveness of an attack and keep the performance of the network within acceptable limits. The novelty of the system lies in the usage of timed finite state machines that enable the real-time detection of attacks. Our system does not introduce any changes to the underlying routing protocol and operates as an intermediate component between the network traffic and the routing protocol. The system was developed and tested to operate in AODV-enabled networks. Our experimental results compare with normal AODV, under attack AODV and the results is more efficient than existing works.

Introduction

MANETs have different properties than conventional networks and present new vulnerabilities. They also share the vulnerabilities of wired networks, such as eavesdropping, denial of service, spoofing and the like; these are simply accentuated by the ad hoc context. First of all, the use of wireless links makes them susceptible to many attacks such as active interference and eavesdropping. Unlike wired networks, attackers do not need physical access to the network to carry out these attacks. Secondly, the dynamic topology of MANETs makes it harder to differentiate normal behaviour of the network from anomalous behaviour. Vulnerability is the use of cooperative algorithms to meet the basic network functions. Routing algorithms for MANETs usually assume that nodes are cooperative and non-malicious. Hence a malicious node can easily become an important routing agent and disrupt network operations by disobeying the protocol specifications. Resource-constraints are a further vulnerability. Devices on MANETs can vary from laptops to handheld devices (e.g. PDAs, mobile phones) and may exhibit a wide range of computing and storage capabilities. They are also generally dependent on battery power to provide mobility. This has led to the emergence of new attacks targeting this aspect [1].

The exibility provided by the open broadcast medium and the cooperativeness of the mobile devices (which have generally different resource and computational capacities, and run usually on battery power) introduce new security risks for MANETs. As part of rational risk management we must be able to identify these risks and take appropriate action. In some cases, we may prevent these risks cost-effectively. In other cases we may have to accept that vulnerabilities exist and seek to take appropriate action when we believe someone is attacking us. That's why intrusion detection systems (IDSs) which monitor system activities and detect anomalies are usually used to complement other security mechanisms. Intrusion detection on MANETs is the main focus of this thesis. The main aim of this thesis is to design and implement an intrusion detection component that will operate in ad hoc networks and more specifically in networks that utilize the Ad hoc On-demand Distance Vector (AODV) routing protocol. The intrusion detection component will be able to keep the network resources operating within normal parameters while malicious nodes attempt to paralyze the network by performing dropping attack and power consumption attack. An important element of this thesis is the design and the implementation of the attacks that an adversary can perform against the routing protocol.

Background

Attacks on MANETs

At the highest level, the security goals of MANETs are not that different from other networks: most typically authentication, confidentiality, integrity, availability, and non-repudiation. Authentication is the verification of claims about the identity of a source of information. Confidentiality means that only authorized people or systems can read or execute protected data or programs. It should be noted that the sensitivity of information in MANETs may decay much more rapidly than in other information systems.

Availability refers to the ability of the network to provide services as required. Denial of Service (DoS) attacks has become one of the most worrying problems for network managers. In a military environment, a successful DoS attack is extremely dangerous, and the engineering of such attacks is a valid modern war-goal. Lastly, non-repudiation ensures that committed actions cannot be denied. In MANETs security goals of a system can change in different modes (e.g. peace time, transition to war, and war time of a military network). The characteristics of MANETs make them susceptible to many new attacks. At the top level attacks can be classified according to network protocol stacks. Some type of attacks could occur in any layer of the network protocol stack, e.g. jamming at physical layer, hello flood at network layer, and SYN flood at transport layer are all DoS attacks. Because new routing protocols introduce new forms of attacks on MANETs. Attackers against a network can be classified into two groups: insider and outsider attackers. Whereas an outsider attacker is not a legitimate user of the network, an insider attacker is an authorized node and a part of the routing mechanism on MANETs. Routing algorithms are typically distributed and cooperative in nature and affect the whole system. While an insider MANET node can disrupt the network communications intentionally, there might be other reasons for its apparent misbehaviors. A node can be failed, unable to perform its function for some reason, such as running out of battery, or collisions in the network. The threat of failed nodes is particularly serious if they are needed as part of an emergency/secure route. Their failure can even result in partitioning of the network, preventing some nodes from communicating with other nodes in the network. A selfish node can also misbehave to preserve its resources. Selfish nodes avail themselves of the services of the other nodes, but do not reciprocate. This research focuses on the attacks carried out by malicious nodes who intentionally aim to disrupt the network communication [1] and [3].

Active Attacks: These attacks cause unauthorized state changes in the network such as denial of service, modification of packets, and the like. These attacks are generally launched by users or nodes with authorization to operate within the network. We classify active attacks into four groups: dropping, modification, fabrication, and timing attacks. It should be noted that an attack can be classified into more than one group.

Dropping Attacks

Malicious or selfish nodes deliberately drop all packets that are not destined for them. While malicious nodes aim to disrupt the network connection, selfish nodes aim to preserve their resources. Dropping attacks can prevent end-to-end communications between nodes if the dropping node is at a critical point. It might also reduce the network performance by causing data packets to be retransmitted, new routes to the destination to be discovered, and the like. Unfortunately most routing protocols (DSR is an exception) have no mechanism to detect whether data packets have been forwarded or not by intermediate nodes. However, attacks against a node can be detected by his neighboring nodes through passive acknowledgement or hop-by-hop acknowledgement at the data link layer. An attacker can choose to drop only some packets to avoid being detected; this is called a selective dropping attack. Besides data packets or route discovery packets, an attacker can also drop route error packets, causing the source node to be unaware of failed links (thus interfering with the discovery of alternative routes to the destination).

Resource Consumption Attack

In this attack the malicious node attempts to consume both the network and node resources by generating and sending frequent unnecessary routing traffic. This routing traffic can only be RREQ and RERR packets since all false RREP are automatically discarded by the specification of the AODV protocol. The goal of this attack is to flood the network with false routing packets to consume all the available network bandwidth with irrelevant traffic and to consume energy and processing power from the nodes [1] and [2] and [4] and [5].

Intrusion Detection Systems (IDS)

Intrusion is any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource and an intrusion detection system (IDS) is a system for the detection of such intrusions. The development of IDS is motivated by the following factors:

- Most existing systems have security was that render them susceptible to intrusions, and finding and fixing all these deficiencies are not feasible.

- Prevention techniques cannot be sufficient. It is almost impossible to have an absolutely secure system.
- Even the most secure systems are vulnerable to insider attacks.
- New intrusions continually emerge and new techniques are needed to defend against them.

Since there are always new intrusions that cannot be prevented, IDS is introduced to detect possible violations of a security policy by monitoring system activities and response. IDSs are aptly called the second line of defense, since IDS comes into the picture after an intrusion has occurred. If we detect the attack once it comes into the network, a response can be initiated to prevent or minimize the damage to the system. It also helps prevention techniques improve by providing information about intrusion techniques.

Taxonomy of Intrusion Detection Systems

There are three main components of IDS: data collection, detection, and response. The data collection component is responsible for collection and pre-processing data tasks: transferring data to a common format, data storage, and sending data to the detection module.

IDS can use different data sources which are the inputs to the system: system logs, network packets, etc. If an IDS monitors activities on a host and detects violations on the host, it is called host-based IDS (HIDS). An IDS that monitors network packets and detects network attacks is called network-based IDS (NIDS). NIDSs generally listen in promiscuous mode to the packets in a segment of the network, allowing them to detect distributed attacks. There are also intrusion detection systems that use both host-based IDS and network-based IDS. For example, a system can use NIDS and also HIDS for important hosts in the networks such as servers, databases, and the like. Since NIDS cannot monitor encrypted packets, a hybrid approach, network node IDS (NNIDS) is introduced where each host in the network has NNIDS to monitor network packets directed to the host [2] and [6].

Intrusion Detection Issues in MANETs

Even though there are many proposed IDSs for wired networks, MANET’s specific features make conventional IDSs ineffective and inefficient for this new environment. Consequently, researchers have been working recently on developing new IDSs for MANETs or changing the current IDSs to be applicable to MANETs. There are new issues which should be taken into account when a new ID is being designed for MANETs.

Lack of central points

MANETs do not have any entry points such as routers, gateways, etc. These are typically present in wired networks and can be used to monitor all network traffic that passes through them. A node of a MANET can see only a portion of a network: the packets it sends or receives together with other packets within its radio range. Since wireless ad hoc networks are distributed and cooperative, the intrusion detection and response systems in MANETs may also need to be distributed and cooperative. This introduces some difficulties. For example, distribution and cooperativeness of IDS agents are difficult in an environment where resources such as bandwidth, processor speed and power are limited. Furthermore, storing attack signatures in a central database and distributing them to IDS agents for misuse-based intrusion detection systems is not suited to this environment [6] and [8].

Mobility

MANET nodes can leave and join the network and move independently, so the network topology can change frequently. The highly dynamic operation of a MANET can cause traditional techniques of IDS to be unreliable. For example, it is hard for anomaly-based approaches to distinguish whether a node emitting out-of-date information has been compromised or whether that node has yet to receive update information. Another mobility effect on IDS is that IDS architecture may change with changes to the network topology.

Wireless Links

Wireless networks have more constrained bandwidth than wired networks and link breakages are common. IDS agents need to communicate with other IDS agents to obtain data or alerts and need to be aware of wireless links. Because heavy IDS traffic could cause congestion and so limit normal traffic, IDS agents need to minimize their data transfers. Bandwidth limitations may cause ineffective IDS operation. For example, an IDS may not be able to respond to an attack in real-time due to communication delay. Furthermore, IDS agents may become disconnected due to link breakages. An IDS must be capable of tolerating lost messages whilst maintaining reasonable detection accuracy.

Limited Resources

Mobile nodes generally use battery power and have different capacities. MANET devices are varied, e.g. laptops, hand held devices like PDAs (personal digital assistants), and mobile phones. The computational and storage capacities vary too. The variety of nodes, generally with scarce resources, affects effectiveness and efficiency of the IDS agents they support. For example, nodes may drop packets to conserve resources (causing difficulties in distinguishing failed or selfish nodes from attacker or compromised nodes) and memory constraints may prevent one IDS agent processing a significant number of alerts coming from others. The detection algorithm must take into account limited resources. For example, misuse-based detection algorithm must take into account memory constraints for signatures and anomaly-based detection algorithm needs to be optimized to reduce resource usage [3] and [7].

Lack of a Clear Line of Defense and Secure Communication

MANETs do not have a clear line of defense. In this environment IDS traffic should be encrypted to avoid attackers learning how the IDS works. However, cryptography and authentication are difficult tasks in a mobile wireless environment since they consume significant resources. In many cases IDS agents risk being captured or compromised with drastic consequences in a distributed environment. They can send false alerts and make the IDS ineffective. IDS communication can also be impeded by blocking and jamming communications on the network.

Cooperativeness

MANET routing protocols are usually highly cooperative. This can make them the target of new attacks. For example, a node can pose as a neighbor to the other nodes and participate in decision mechanisms, possibly affecting significant parts of the network [2] and [5].

Specification-based Anomaly Detection

This research study presents a new approach for detecting network intrusions. The new approach is called specification-based anomaly detection and it is a hybrid combination of anomaly-detection and knowledge-based intrusion detection techniques. The authors suggest that the new approach mitigates the weaknesses of the two approaches while magnifying their strengths. To realize their approach they have developed state machine specifications of network protocols, and then they augment these state machines with information about the statistics that need to be maintained to detect anomalies. Furthermore, a specification language was specifically developed in which all of the required information can be captured in a concise manner. The protocol specifications that it are utilized simplify the feature selection process that is required from the anomaly-detection component. Thus, the machine learning component is claimed to be robust enough to operate without human supervision. The experiments that were performed in this study indicate that the developed system has low rate of false alarms and that it is able to identify unseen stealthy email viruses in intranet environments [9] and [11].

Statistical Process Control for Computer Intrusion Detection

In this study an interesting architecture of distributed, host-based IDS is proposed. The system is developed based on statistical process control and employees both of the intrusion detections techniques mentioned earlier. By utilizing each technique it determines an intrusion warning level based on the audit data events. The intrusion warning levels are then fused to produce a combined intrusion level. The composite intrusion warning level can have values of 0 for normal to 1 for intrusive, any value that is in between signifies a level of intrusiveness [10] and [12].

Intrusion detection on MANETs is the main focus of this thesis using implement of dropping attack and power consumption attack detection method.

Proposed Technique

The proposed system utilizes the anomaly intrusion detection technique in order to identify newly and unseen attacks. The authors suggest that this system requires updated data describing the users’ behavior and the statistics in normal use.

The proposed system can be characterized as an architecture model for intrusion detection in wireless ad hoc networks, while its implementation targets specifically the AODV routing protocol. The reason why it can be classified as an architecture model is that it does not perform any changes in the underlying routing protocol but it merely intercepts routing and application traffic. Thus, the security component operates in a different layer without interfering with the normal operation of the routing protocol. Since the system does not utilize any cryptographic mechanism to ensure protection from malicious activities, it does not introduce any additional computation overhead to the routing process. Furthermore, it does not require the sending of additional packets, thus it does not consume the available bandwidth.

The underlying protocol used for the implementation of the intrusion detection component is the AODV routing protocol that recently became an Internet standard, coming one step closer towards being established as the main routing protocol to be used in ad hoc networking environments. Although some of the attacks that the system is designed to detect are specific to the AODV protocol.

The intrusion detection system runs locally in every participating node and it makes decisions upon the partial view of the traffic that it observes. Thus, it is a host-based network intrusion detection system. The operation of the system could be terminated when a malicious node is detected, however in order to provide a more complete solution the nodes upon an alarm take countermeasures to deal with the isolation of the detected misbehaving node and to keep the performance of the network within acceptable limits. The intrusion detection component has high rates of accuracy in detecting the malicious nodes and the countermeasures taken by the nodes individually enable the network to withstand aggressive attacks and keep the network operating within acceptable performance limits.

An enumeration of the objectives of the system will assist in the evaluation process of the intrusion detection component. Hence, the objectives of our system can be summarized in the following points:

- Create an intrusion detection model for wireless ad hoc networks that can be further extended to operate for many reactive or proactive routing protocols.
- Select some of the active attacks that a malicious node can perform against the AODV routing protocol and implement them.
- Formally describe the detection of the attacks with the use of timed finite state machines and fine tune them to achieve the maximum accuracy.
- According to the observed routing traffic more than one FSM (Finite State Machine) may be triggered simultaneously, however the system as a whole should not reach contradicting decisions.
- Upon detection of malicious activity the detecting node should be able to take countermeasures to hold the network performance in acceptable performance measures.
- The detected malicious nodes will be penalized for a finite period of time rather than being isolated forever in order to avoid the impact of possible false positive alarms.

Implement of Routing Attacks

In order to test the intrusion detection component we selected dropping attack and power consumption that have significant impact on network performance degradation when they are actively performed. These attacks can be applied to any routing protocol.

Implementation of the Dropping Routing Packets Attack

The rogue routing agent that was implemented to carry out the malicious behaviour of the dropping routing packets attack was added similarly to the SEQAODV routing agents. It was required to modify some files in order to add the new agent called DRPAODV routing agent. The internal files of the simulator that were modified along with the modifications made to add the DRPAODV routing agent.

In this attack the malicious node acts selfishly and drops all routing traffic that it is not destined for itself. Thus, upon of a RREQ packet it checks if the destination of the route discovery is itself and if this holds then it further processes the packet and sends a RREP. When it receives a RREP packet it checks if it has sent the original request for this route and if this holds it adds the new route to its routing table. The RERR packets are processed normally in all cases. In any other cases it drops the packets without further processing them. The attack is implemented in the methods recvRequest and recvReply.

The timed FSM developed to detect this attack was incorporated into the existing AODV routing agent that was previously described. It is essential for the detection of this attack to place the participating nodes in promiscuous mode, hence becoming able to overhear the forwarded traffic. Since AODV does not operate in promiscuous mode by default, some modifications had to be performed in the internal files along with some modifications in the AODV protocol implementation to add this functionality. The modifications that were performed are presented. The fact that promiscuous mode was enabled in AODV had no impact in the overall performance of AODV and the tap method that handles the overheard packets is only utilised in the detection of the dropping routing packets attack.

The FSM developed to detect this attack is triggered whenever a node forwards routing traffic to its neighbouring nodes. A structure called DRP_Node was developed to hold information necessary to monitor the neighbouring nodes that are suspected for malicious behaviour. The DRP_Node data structure holds the following information:

- node_id: the IP address of the node to which the routing traffic was forwarded.
- send_reply: a boolean value that becomes true whenever the offending node replies to a RREQ packet that was forwarded to it.
- pre_alarm: a boolean value that becomes true if the node does not respond as expected to the forwarded traffic.
- alarm: a Boolean value that becomes true whenever we decide that the offending node performs the dropping routing packets attack.
- time: a double variable that keeps the time where the offending node was added in the data structure.

Hence, whenever a node forwards routing traffic for which a neighboring node is not the destination it adds each neighboring node to the data structure and waits to observe their behaviour. Then in the tap method if it overhears that a neighboring node has replied to the forwarded RREQ, it means that it has acted appropriately and it can be removed from the monitoring list. If this is not the case and the packet was a RREP then the offending node has to forward the packet. If it fails to do so within the pre_alarm_time_threshold time period, which was determined by experiments to be 0.01 seconds, the pre_alarm state becomes true. The FSM remains in the pre_alarm state for 0.45 seconds which is the alarm_threshold time period. If the offending node fails to forward the routing packet within this time limit the FSM moves to the Alarm state. In case of an alarm the legitimate node marks this node as malicious and stops forwarding traffic to it for 2 seconds and it also sends a RERR message to all its upstream neighbors to inform them that all the routes that include this node are not valid any more.

Pseudo code of the implementation of the intrusion detection component for the FSM used to detect the dropping routing packets attack:

```
AODV
void AODV::tap(const Packet *p)
{
extract the header of the packet p;
if it is a RREQ packet
{
    if I am listening to my own RREQ packet
    {
    return;
    }
    if the source of the packet exists in the monitoring list
    {
    remove(source); // since it has forwarded the traffic
    }
}
else if it is a RREP packet
{
    if I am listening to my own RREP packet
    {
    return;
    }
    if the source of the packet exists in the monitoring list
    {
    remove(source); // since it has forwarded the traffic
    }
    if the node exists in the monitoring list
    {
    if the pre_alarm state is false
    {
     check the pre_alarm threshold against the time that this node
     was added in the list;
    if the time threshold has not expired
    {
     move the FSM to the pre_alarm state by setting the pre_alarm
     value to true;
    }
    else // has expired, so remove it
    {
    remove the node form the monitoring list;
    }
    }
    else // the pre_alarm state is true
    {
     check the alarm threshold time against the time that this node
     was in the alarm state;
   if the time threshold has not expired
    {
    set alarm value to true;
    send RRER packet to upstream neighbours;
     start timer for 2 seconds that the offending node will be pena-
     lised;
    }
    else
    {
    remove the node from the monitoring list;
    }
    }
    }
}
}
```

Implementation of the Resource Consumption Attack

In order to implement the resource consumption attack a new rogue routing agent that would realize this behavior had to be created. Similarly with the other routing agents, some internal files were modified. The new routing agent is called RCAODV.

The implementation of this attack was rather simple. The only modification that had to be made was a loop that sends frequent unnecessary routing traffic. One of the methods that is steadily and frequently used by the AODV routing protocol implementation is the sendRequest method. Thus, we decided that this loop that sends the unnecessary routing traffic should be implemented there. The destinations that are used in the unnecessary RREQ and RERR packets are nodes that do not exist in the network. In order for these packets not to be discarded automatically by the protocol implementation the destination nodes should be different each time. Hence, a function that returns pseudo random addresses was developed to realise this task. The number of the additional routing packets that are sent each time the malicious node initiates a route discovery process is randomly chosen between 2 and 10.

Implementation of the Resource Consumption Attack Detection

The timed FSM developed to detect the resource consumption attack was incorporated into the existing AODV routing agent that was presented in the sequence number attack detection section. To identify this attack two different data structures were developed. The first one is called RC_Node and holds the address of a node, a counter that denotes the number of routing packets that were received from the node and a time value that signifies the period in which the traffic was received. For every node that a legitimate node receives traffic from it keeps an associated counter and a timer. If within the time threshold of 7 seconds the counter of a node reaches the threshold value, which is 10, then it is removed from this list and it is added to the Alarm list. The Alarm list is realised by the second data structure called Alarm_Node that stores the time that the node was added in the list and the node's address. For the time threshold of 5 seconds all the traffic that origins from nodes that exist in the Alarm list is dropped without being further processed.

Pseudo code of the implementation of the intrusion detection component for the FSM used to detect the resource consumption attack:

```
AODV
void recvRequest(Packet p*)
{
// the normal operation of the method remains unchanged
check if the node's IP address is included in the alarm list
{
    if the timer has expired
    {
    it is safe to reset by removing the node from the alarm list;
}
else
{
drop the packet;
return;
    }
}
check if the node's IP address is included in the list
{
    if the timer has expired
    {
    remove node the node from the list;
    }
else
    {
if the counter is lower than the threshold value
{
increment the counter;
}
else if the counter has exceeded the threshold value
{
add the node to the alarm list;
remove it from the original list;
}
else it means that we do not have a entry for this node
{
    so we add the node to the original list;
}
}
// The method continues normally }
```

One of the most challenging parts in the development of the timed finite states machines was to fine tune them to achieve maximum accuracy. Both the timers and the threshold values that are included the FSMs had to be tested individually as well as combined all together in order to reach satisfactory results. In the following chapter were the evaluation and metrics that were used to determine the performance of our system are analyzed, the importance of the values presented here will become clearer.

Results

All intrusion detection systems suffer from false alarms that occur whenever the system incorrectly concludes in an alarm but there is no malicious behaviour present in the network. The knowledge-based intrusion technique that our system utilizes to detect malicious activity is less error prone than other intrusion techniques, like for example the behaviour-based intrusion detection approach. However, the traffic patterns that denote that an active attack is performed against the routing protocol can be realised when the AODV operates normally due to high application traffic and high node mobility. The system was tested in terms of detection accuracy and the percentages of detection accuracy for the three attacks are the following:

- Dropping routing packets attack detection accuracy: 71.5%.
- Resource consumption attack detection accuracy: 74.8%.

The detection accuracy of the system in all the three attacks can be considered high compared to results of other similar existing works.

Conclusion

Our intrusion detection system is a novel lightweight system that detects and takes countermeasures against active attacks that can be performed against the AODV routing protocol in mobile ad hoc networks. Despite that fact that there are many research papers that claim to have implemented similar active attacks using the network simulator, the information that is available in the papers and on the Internet is minimal raising suspicion on whether or not they have actually correctly implemented these attacks. Additionally, the network simulator and the CMU extensions for ad hoc networks that include the AODV are not flexible in usage and in modification. For that reason the malicious behaviours and the intrusion detection enabled AODV (ID-AODV) are implemented with rogue routing agents and many modifications in the internal files of the network simulator had to be made. Furthermore, the patterns that denote a specific malicious behaviour had to be first proven theoretically to decide whether it is feasible to design traffic patterns that can uniquely identify a specific malicious behaviour. This will proved to be harder than it first appeared to be since it is proven that some attacks that involve node impersonation (a node uses another node’s address to perform an attack) could not be identified without the use of a cryptographic mechanism. As it will present in the evaluation of the system, the developing intrusion detection mechanism manages to detect the active attacks that are specified with high accuracy and keeps the network performance within acceptable limits. The intrusion detection system operates currently for the AODV routing protocol, however it does not alter any of its fundamental operational functions. It also provides a lightweight mechanism to ensure protection from malicious activities performed against the routing fabric.

References

[1] Zeyad M. Alfawaer and Saleem Al_zoubi, “A proposed Security subsystem for Ad Hoc Wireless Networks”, IEEE 2009 International Forum on Computer Science-Technology and Applications, pp 253-256.

[2] Matthew Tan Creti, Matthew Beaman, Saurabh Bagchi, Zhiyuan Li, Yung-Hsiang Lu, “Multigrade Security Monitoring for Ad-Hoc Wireless Networks”, 978-1-4244-5113-5/09, IEEE 2009, pp 342-352.

[3] Wenchao Huang, Yan Xiong, Depin Chen, “DAAODV: A Secure Ad-hoc Routing Protocol based on Direct Anonymous Attestation”, IEEE 2009 International Conference on Computational Science and Engineering, pp 809-816.

[4] Wenjuan Li and Lingdi Ping and Xuezeng Pan, “Use Trust Management Module to Achieve Effective Security Mechanisms in Cloud Environment” IEEE 2010 International Conference on Electronics and Information Engineering (ICEIE 2010).

[5] Haider Abbas, Christer Magnusson, Louise Yngstrom and Ahmed Hemani, “A Structured Approach for Internalizing Externalities Caused by IT Security Mechanisms”, IEEE 2010 Second International Workshop on Education Technology and Computer Science.

[6] Wei Ren, Yoohwan Kim, Ju-Yeon Jo, Mei Yang3 and Yingtao Jiang, “IdSRF: ID-based Secure Routing Framework for Wireless Ad-Hoc Networks”, IEEE International Conference on Information Technology (ITNG'07).

[7] Anand Patwardhan and Michaela Iorga, “Secure Routing and Intrusion Detection in Ad Hoc Networks”, Proceedings of the 3rd IEEE Int’l Conf. on Pervasive Computing and Communications (PerCom 2005).

[8] A.H Azni, Azreen Azman, Madihah Mohd Saudi, AH Fauzi, DNF Awang Iskandar, “Analysis of Packets Abnormalities in Wireless Sensor Network”, IEEE 2009 Fifth International Conference on MEMS NANO, and Smart Systems, pp 259-264.

[9] Cuirong Wang, Shuxin Cai and Rui Li, “AODVsec: A Multipath Routing Protocol in Ad-Hoc Networks for Improving Security”, IEEE 2009 International Conference on Multimedia Information Networking and Security, pp 401-404.

[10] A Nagaraju and B.Eswar, “Performance of Dominating Sets in AODV Routing protocol for MANETs”, IEEE 2009 First International Conference on Networks & Communications, pp 166-170.

[11] Sheng Cao and Yong Chen, “AN Intelligent MANet Routing Method MEC”, 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp 831-834.

[12] WANG Xiao-bo ,YANG Yu-liang, AN Jian-wei, “Multi-Metric Routing Decisions in VANET”, 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp 551-556.