RESEARCH ARTICLE OPEN ACCESS

# Public Key Cryptosystem Approach for P2P Botnet Detection and Prevention

Anas Aliyu Usman[1], A Arokiaraj Jovith[2]
[1] Department of Information Technology, SRM University, Kattankulathur-603203
[2] Department of Information Technology, SRM University, India

**ABSTRACT**
Distributed (P2P) botnets have as of late been received by botmasters for their versatility against take-down endeavors. Other than being harder to bring down, p2p botnets tend to be stealthier in the way they perform vindictive exercises, making current discovery approaches ineffectual. In this paper, we simulate our proposal by detecting a gray hole attack in an Ad Hoc network using NS2.The detected malicious node is listed in a black hole list and notices all other nodes in the network to stop communicating with them. Our botnet location framework has been equipped for identifying stealthy P2P botnets (Gray Hole nodes) and can reduce packet loss caused by malicious nodes and have a better packet delivery ratio (PDR) within less period of time.

***Keywords* -** Peer-to-Peer (P2P), bot, Botnet, Bot-master

## I. INTRODUCTION

A BOTNET is a collection of compromised hosts that are remotely controlled by a malicious user usually called an attacker or a botmaster through a Command and Control (C&C) channel [1]. People with malicious intent make use of the botnet as a tool for various cybercrimes such as Distributed Denial of Service (DDoS) attack, Generation of spam emails, Click Fraud and so on. The C&C is used for issuing command to the bots and receiving information from them. For more than a decades there is an immense rise of the peer-to-peer computing paradigm [10]. This is due to the fact that traditional botnet represent a single point of failure [1]. Most of the security researches focuses on bringing down the central control because it is like taking the whole botnet down. Botnet Bot usually refers to software robots, which are used to automate tasks. Nowadays bot refers to an infected/compromised computer which can accept commands from remote controller (bot master). Botnet is a network of infected systems under the control of a bot master. The bot master can perform coordinated activities with these bots by issuing commands. In recent times, use of bots for nefarious activities pose serious threat. In P2P bots, commands are communicated through push/pull mechanism. Bot master publishes a command file over the P2P network. The bots then use the pull mechanism to obtain the command file [5]. P2P bots have to constantly communicate with its neighbours for commands and has to send KEEP ALIVE messages to other bots in the network. P2P botnets do not suffer from single point of failure but coordination of bots is difficult compared to the centralized architecture [10]. The advantages of P2P botnet over the centralised structure includes non-existence of C&C which reduced the dependency on the C&C server but it does not mean it is completely gone, bot may still decide to contact a C&C server under specific conditions. Example when there is stolen data to communicate back to the attacker. If they managed to completely remove the server then this can be considered a step to strengthening the botnet. If it only operates through P2P then it becomes nearly impossible to track the Guys behind it. P2P bot's life cycle consists of the following steps [10]:

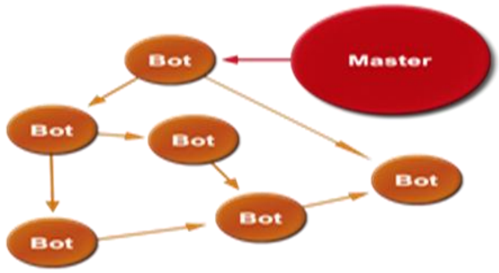


**Figure 1:** P2P Botnet structure

### *First Stage (Infection stage)*

The p2p bot's life cycle begins with the infection stage. The victim computer can be exploitation due to any one of the following reasons:

- Unpatched vulnerabilities
- Backdoors
- left by Trojans
- Password guessing and brute force attacks.

During infection period the bot spreads (this might happen through drive-by downloads, a malicious software being installed by end-user, and/or infected USB sticks, etc.).

***Second Stage (Rally stage)***

This is where the bot connects with a peer list in order to join the P2P networks.

***Third Stage (Waiting stage)***

During this stage, bots waits for the bot-master’s command

***Fourth Stage (Execution stage)***

This is the final stage in which it actually carries out a command, such as:

- Denial of-service (DoS) attack
- Generate spam emails
- Click fraud, etc.

The rest of the paper is organized as follows [4]. Section II discusses some related work on various approaches for detecting and prevention of P2P botnet. Section III discusses working principle of Ad hoc on Demand Distance Vector (AODV). Section IV discusses Secured AODV (SAODV). Section V discussed the proposed mechanism used the paper. Section VI the simulation result and finally section VII discusses the conclusion and future work.

## II. RELATED WORK

Yao Zhaoy proposed [8] has described a plan and execute a framework called Botgraph to distinguish the Web-record ill-use assault at a huge scale. We make two vital commitments. They first commitment is to propose a novel graph based methodology to distinguish the new Web-record misuse assault. This methodology uncovered the underlying associations among client login exercises by developing an extensive user diagram. Yao Zhaoy’s methodology is focused around the perception that bot-clients offer IP addresses when they log in and send messages.

Ms. Meenakshi et al [3] has simulate a gray hole attack as a malicious activity in ad hoc network using NS2.

Junjie Zhang et al presented [9] one of the most efficient method for detecting P2P botnet. Primarily their system identifies all host that are likely engaged in P2P communications and then derive a statistical fingerprints of the P2P communications to do the following task: Firstly, to Maintain and Monitor Profile of P2P traffic and further differentiate between P2P botnet traffic and legitimate P2P traffic.

Shalini Jain et al has presented [11] where a novel algorithm for detecting and prevention of cooperative black and gray hole attacks in a Mobile ad hoc networks.

Pratik Narang et al presented [10]. Peer shark is a novel methodology designed to detect p2p botnet traffic and differentiate it from benign p2p traffic in a network. It uses 2-tuple “conversation based” approach and it does not require deep packet inspection and can classify different p2p application with accuracy greater than ninety percent. Peer shark has the advantage of lack of single point-of failure and can categorize exact p2p application running on a host inside a network. Peer shark is limited to independent on deep packet inspection or a signature-based mechanism (uses by botnets/ application using encryption).

Jay dip et al [4] has proposed a mechanism for detecting a gray hole attack in a mobile ad hoc network.

## III. AD HOC ON-DEMAND DISTANCE VECTOR (AODV)

The Ad-hoc On-demand Distance Vector (AODV) routing protocol allows mobile nodes to quickly obtain routes for a new destinations, and it does not require nodes to maintain routes to the destinations that are in not in active communication [6]. It is classified under reactive protocol [3]. The main function of AODV is route discovery and route maintenance. The route discovery process begins with the creation of route request (RREQ) packet. To find a route to a particular destination node, the source node broadcast a RREQ to its immediate neighbors [12]. If one of these neighbors has a route to the destination, then it replies back with route reply (RREP) packet. Otherwise the neighbors in turn rebroadcast the request. This continues until the RREQ hits the final destination or a node with a route to the destination. In AODV, the routing protocol uses a destination sequence number for each route entry. The destination sequence is generated by the destination when a connection is requested to it [5]. The principle of this protocol is the greater the destination sequence number the fresher the route [14].In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it. As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route. As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

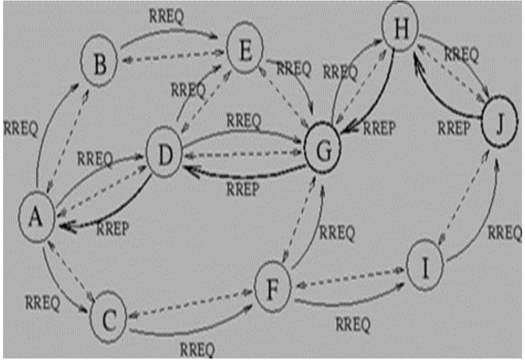


**Figure 2:** AODV Route Discovery Process

## IV. SECURE AD HOC ON DEMAND DISTANCE VECTOR (SAODV)

SAODV is an extension of the AODV routing protocol that protects the route discovery mechanism providing security features like integrity and authentication. It uses digital signatures to authenticate the non-mutable fields of the messages, and hash chains to secure the hop count information (the only mutable information in the messages).SAODV can use the Simple Ad hoc Key Management (SAKM) as a key management system. The Secure Ad hoc On-Demand Distance Vector Routing Protocol (SAODV) is an extension of the AODV routing protocol that can be used to protect the route discovery mechanism providing security features like integrity, authentication and non-repudiation. SAODV assumes that each ad hoc node has a signature key pair from a suitable asymmetric cryptosystem. Further, each ad hoc node is capable of securely verifying the association between the address of a given ad hoc node and the public key of that node. Achieving this is the job of the key management scheme. Two mechanisms are used to secure the AODV messages: digital signatures to authenticate the non-mutable fields of the messages, and hash chains to secure the hop count information (the only mutable information in the messages). This is because for the non-mutable information, authentication can be performed in a point-to-point manner, but the same kind of techniques cannot be applied to the mutable information. Route error messages are protected in a different manner because they have a big amount of mutable information. In addition, it is not relevant which node started the route error and which nodes are just forwarding it. The only relevant information is that a neighbor node is informing to another node that it is not going to be able to route messages to certain destinations anymore.

## V. PROPOSED MECHANISM

In this paper, we proposed a new approach for malicious detection and secure routing the detected malicious node is listed in the black hole list and notices all other nodes in the network to stop any communication with them**.** Introducing SAODV to put additional secure framework to control the attacks. These contributed in improving security in MANET there by reducing the packets loss caused by the malicious nodes and have better Packet Delivery Ratio (PDR) within less period of time [6]**.**

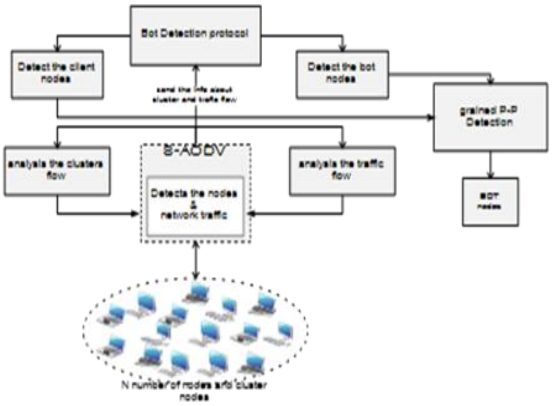

Bot Detection protocol
Detect the client nodes
Detect the bot nodes
send the info about cluster and traffic flow
grained P-P Detection
S-AODV
analysis the clusters flow
Detects the nodes & network traffic
analysis the traffic flow
BOT nodes
N number of nodes and cluster nodes

**Figure 3:** Architectural Design

The architecture in the fig. 3 above describe the full picture of how the secure SAODV works by analyzing the cluster flow and traffic flow in the Mobile Ad hoc Network (MANET). It first detect the nodes engaged in communication within the monitored network and then analyze the cluster of nodes in the network to determine the number of nodes communicating with each other in the cluster, it then determine type of traffic going in and out of the networks i.e. tcp or udp traffic. It then uses the information gathered at the first stage to detect or differentiate between legitimate client nodes and bot nodes in the monitored network. The secured Ad hoc on-demand distance vector routing protocol ensures integrity at the receiving node by calculating the hash of the packet received and stop communicating with that malicious node and let other communicating nodes on the monitored network to stop communicating with it further there by optimizing the good packet delivery.

### *1.1 GRAY HOLE ATTACK*

Our concern in this paper is to find the flaws of the security in the AODV protocol, and then give some insight explaining distinct types of the gray hole attack [4].Actually, in the AODV routing protocol, every mobile node maintains a routing table that stores the next hop node information for a route to a destination node. When a node wants to find a route to another one, it broadcasts a RREQ to all the network till either the destination is reached or another node is found with a fresh enough route to the destination (a fresh enough route is a valid route entry for destination whose associated sequence number is at least as great as that contained in the (RREQs**)**. Then a RREQ is sent back to the source and the discovered route is made available.

Nodes that are part of an active route may offer connectivity information by broadcasting periodically local Hello messages (special RREQ messages) to its immediate neighbors. If hello messages stop arriving from a neighbor beyond some given time threshold, the connection is assumed to be lost.

When a node detects that a route to a neighbor node is not valid it removes the routing entry and send a REER message to neighbors that are active and use the route; this is possible by maintaining active neighbor lists. This procedure is repeated at nodes that receive REER messages. A source that receives an REER can reinitiate a RREQ message. AODV does not allow handling of unidirectional links. Now we have to describe the gray hole attack on MANETS. The gray hole attack has two important phases [4]:

1. In the first phase, a malicious node exploits the security flaws of AODV protocol to advertise itself as having a valid route to a destination node, with the intention of intercepting packets, even though the route is spurious.
2. In the second phase, the node drops the intercepted packets with a certain probability. A gray hole may exhibit its malicious behavior in different ways. It may drop packets coming from (or destined to) certain specific node(s) in the network while forwarding all the packets for other nodes. Another type of gray hole node may behave maliciously for some time duration by dropping packets but may switch to normal behavior later. A gray hole may also exhibit a behavior which is a combination of the above two, thereby making its detection even more difficult.

### *1.2 RSA ALGORITHM*

In this paper we use the RSA algorithm for node to node verification. For the detection of malicious node (bot node), the mentioned algorithm is used in which each and every legitimate node is expected to have pairs of key i.e. public and private key. A node is considered and marked malicious node if it fails to have these two keys and is removed from the legitimate communicating node. Below is the detailed of the algorithm:

*Public Key:*
***n*** = Product of two primes, *p* and *q* (*p* and *q* must remain secret)
***e*** *relatively* prime to ($p$ - 1)($q$ - 1)
*Private Key:*
$d^{e-1}$mod (($p$ - 1)($q$ - 1))
*Encrypting:*
$c = me$ mod $n$
*Decrypting:*
$m = cd$ mod $n$

## VI. SIMULATION RESULTS

*Simulation Environment*

For simulation purpose, in this paper we have used NS-2 version 2.3 simulator on Fedora 8 operating system. The network simulator 2 is widely used tool in a network research and network industry. It is discrete event simulation and capable simulating various types of networks [3]. NS2 consist of two languages, C++ and Otcl. In the back end C++ which defines the internal mechanism of the simulation object, and the front end Otcl set up simulation by assembling and configuring objects as well as scheduling discrete events. To simulate NS2, a (.tcl) script file is required. After simulation it creates two types of file, one is trace file (tr) and another is (.nam) file.The trace file is used for calculation and statistical analysis, and that of .nam file is used to visualize the simulation process.

The implementation involves a network topology with 50 nodes in which we have different source nodes and destination nodes communicate with each other respectively. Below table shows the simulation parameters:

**TABLE I**
**SIMULATION PARAMETERS**

| **PARAMETERS** | **VALUES** |
|---|---|
| Simulator | NS-2 version 2.3 |
| Total number of Nodes | 50 |
| The Traffic model | Constant Bit Rate (CBR) |
| The routing protocol | AODV |
| Total number of malicious nodes | 3 |
| Number of source nodes | 3 |
| Simulation Time | 20 sec |

In this paper, there is 50 P2P nodes network depicted in the fig.4. The initial location of the respected nodes where set in two dimensional system (the z coordinate is assumed throughout to be 0) [8].

*Node communication between source and destination*

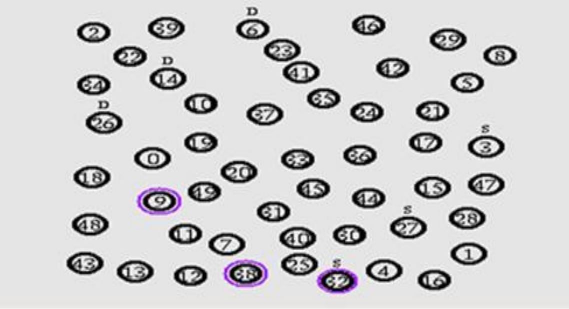


**Figure 4:** Scenario for basic topology of 50-nodes multiple Node communication

From the simulation result node 32 starts sending a number of packets to node 26 through node 38, 11 and 9 respectively at a speed of 3m/sec. Another node 27 starts sending a number of packets to node 14 through node 17, 42 and 41 respectively at the same speed. Lastly, another source node 3 sent additional 55 packets to the node 6 via 21, 42 and 46.

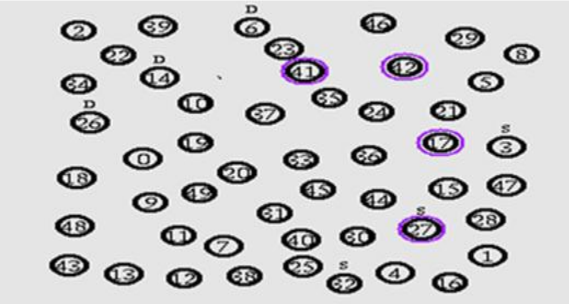


**Figure 5:** Snapshot of simulation in network animator (NAM)

*Attack after detection of the bot*

A bots was detected and broadcast a message to all other legitimate nodes to stop communicating with that node which was identified as node 36, 24 and 21.

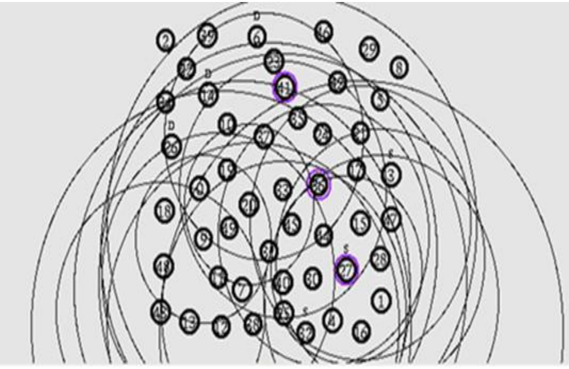


**Figure 6:** Snapshot of simulation in network animator (NAM)

*Malicious bot identification with Packet Delivery Ratio (PDR)*

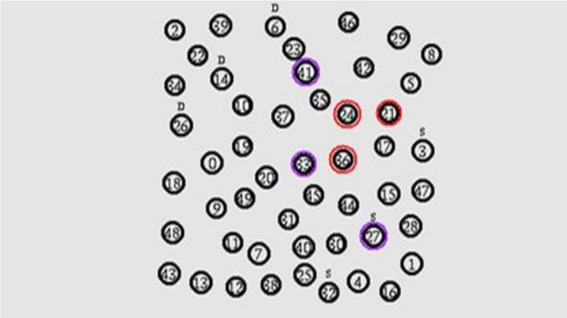
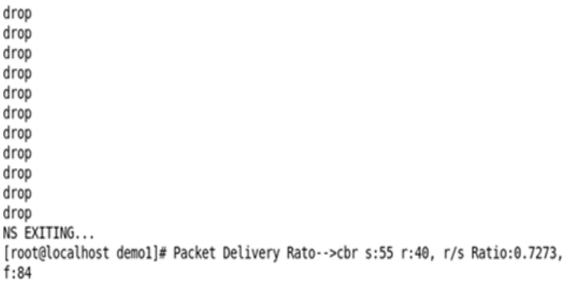


**Figure 7:** Snapshot of simulation in network animator (NAM)



```
drop
drop
drop
drop
drop
drop
drop
drop
drop
drop
drop
NS EXITING...
[root@localhost demo1]# Packet Delivery Rato-->cbr s:55 r:40, r/s Ratio:0.7273,
f:84
```

**Figure 8:** Snapshot of simulation in terminal

In this paper, packet delivery ratio is considered as a measurement of improving network security or network performance [2].In the simulation above the packet delivery ratio (PDR) is found to be 72% which means more than 25% of packet data were lost as a result of the malicious node between the source and the destination node.

**TABLE II**

| | |
|---|---|
| Source node | 27, 32, 3 |
| Destination node | 14, 26, 6 |
| Malicious node ( Bot node) | 36, 24 and 21 |
| Algorithm used | RSA |
| Packet Delivery Ratio | 72% |

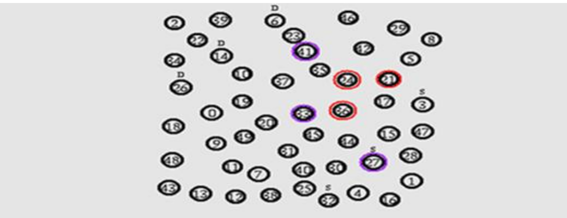
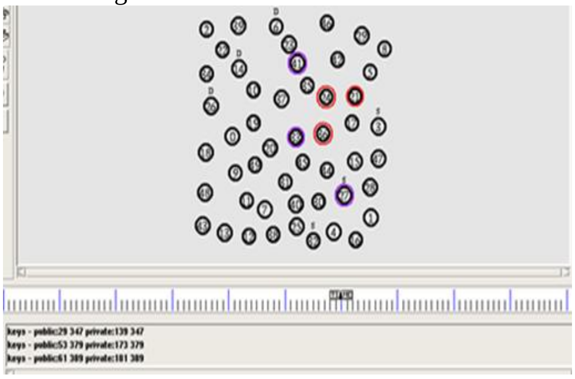


**Figure 9:** Simulation in network animator (NAM)

From the result in the Fig. 9 above source node i.e. node 3, 32 and 27 sent 55 packets at different time to the destination node 6, 26 and14 respectively but 28 packets were dropped by the bot nodes identified in red colored node. Hence only 40 packets were received by the receiving nodes.

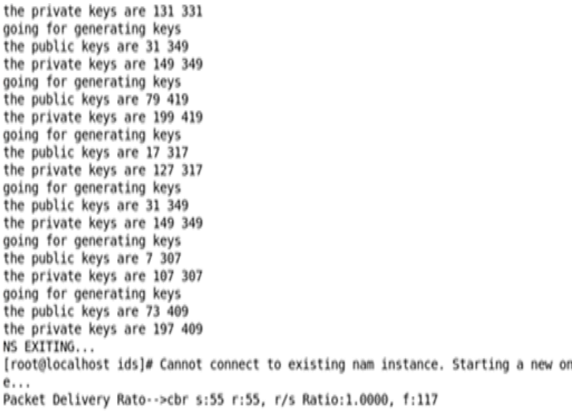
*Identifying malicious node and preventing the attack using RSA key exchange-(SAODV) implementation.*

IDS-After Attack detection and sending in different path with packet verification using RSA key exchange-(**SAODV**) implementation. After certain period of time the bot i.e. node 36, 24 and 21 was identified intercepting the communicating traffic between legitimate source/destination nodes.



```
keys - public:29 347 private:139 347
keys - public:53 379 private:173 379
keys - public:61 389 private:181 389
```

**Figure 10:** Simulation in network animator (NAM)



```
the private keys are 131 331
going for generating keys
the public keys are 31 349
the private keys are 149 349
going for generating keys
the public keys are 79 419
the private keys are 199 419
going for generating keys
the public keys are 17 317
the private keys are 127 317
going for generating keys
the public keys are 31 349
the private keys are 149 349
going for generating keys
the public keys are 7 307
the private keys are 107 307
going for generating keys
the public keys are 73 409
the private keys are 197 409
NS EXITING...
[root@localhost ids]# Cannot connect to existing nam instance. Starting a new on
e...
Packet Delivery Rato-->cbr s:55 r:55, r/s Ratio:1.0000, f:117
```

**Figure 11:** Simulation in terminal

From the simulation result obtain under Constant Bit rate (CBR) we have an improved secured network after implementing the RSA public key crypto-systems in the network. The table below indicate that an improve security were achieved.

**TABLE III**

| | |
|---|---|
| Source node | 27, 32, 3 |
| Destination node | 14, 26, 6 |
| Malicious node (Bot node) | 36,24 and 21 |
| Algorithm used | RSA |
| Packet Delivery Ratio | 90% |

## VII. Conclusion

In this paper a technique is proposed and applied to detect a malicious (Gary Hole) node in the ad hoc network. Our technique works well in detecting a bot node (Gray Hole node) and ensuring secured routing. This is ensured by measuring the network performance using packet delivery ratio (PDR). From the result of the simulation obtained, it shows that a lot of packets lost before securing the network resulting in loss of more than 25% of the packet being sent as a result of the malicious node. When the security was implemented it shows an improved secured network by having more than 90% of packets received at the receiving node resulted in high packet delivery ratio as shown in the Table III above. The simulation results using NS-2 shows that in a moderately changing network, most of the malicious nodes could be detected, the routing packet overhead was low, and the packet delivery ratio has been improved.

## References

[1] Junjie Zhang, Roberto Perdisci, Wenke Lee, Xiapu Luo, and Unum Sarfaz, "*Building a Scalable System for Stealthy P2P-Botnet Detection*", IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 1, JANUARY 2014

[2] J. K. Mandal and KhondekarLutful Hassan, "*A Novel Technique to Detect Intrusion in MANET*", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.5, September 2013

[3] Ms. Meenakshi et al, "*Simulation of Gray Hole Attack in Ad hoc Network Using NS2*", International Journal of Computer Science & Engineering Technology (IJCSET)

[4] Jaydip Sen, M. Girish Chandra, Harihara S.G., Harish Reddy, P. Balamuralidhar, "*A Mechanism for Detection of Gray Hole Attack in Mobile Ad Hoc Networks*"

[5] Nitesh Funde#1, P. R. Pardhi, "*Analysis of Possible Attack on AODV Protocol in MANET*", ***International Journal of Engineering Trends and Technology (IJETT) – Volume 11 Number 6 – May 2014***

[6] Patil V.P, "*Efficient AODV Routing Protocol for MANET with enhanced packet delivery ratio and minimized end to end delay*".

[7] David Dittrich, and Sven Dietrich "*P2P as botnet command and control: a deeper insight*"

[8] Yao Zhaoy, YinglianXie, Fang Yu, QifaKe, Yuan Yu, Yan Cheny, and Eliot Gillum, "*BotGraph: Large Scale Spamming Botnet Detection*"

[9] Junjie Zhang, Xiapu Luo, Roberto Perdisci, GuofeiGu, Wenke Leeand Nick Feamster , "*Boosting the Scalability of Botnet Detection Using Adaptive Traffic Sampling*"

[10] Pratik Narang et al PeerShark: “*Detecting Peer-to-Peer Botnets by Tracking Conversations*”, IEEE Security and Privacy Workshops 2014.

[11] Shalini Jain et al, “*Advanced algorithm for Detection and Prevention of Cooperative Black and Gray Hole Attacks in Mobile Ad Hoc Nteworks*”, International Journal of Computer Applications, 2010

[12] G. Acs, L. Buttyan, and I. Vajda, “*Provably Secure On-Demand Source Routing in Mobile Ad Hoc Networks*,” IEEE Trans. Mobile Computing, vol. 5, no. 11, pp. 1533-1546, Nov. 2006.

[13] Aad, J.-P. Hubaux, and E.W. Knightly, “*Denial of Service Resilience in Ad Hoc Networks*,” Proc. ACM MobiCom, 2004.

[14] C. Perkins. *“(RFC) request for Comments-3561*”, Category:Experimental, Network, Working Group, July 2003