I. J. Computer Network and Information Security, 2012, 7, 12-18
Published Online July 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.07.02

Protection Against DDoS and Data Modification Attack in Computational Grid Cluster Environment

Basappa B. Kodada[1] , Gaurav Prasad[2] , Alwyn R. Pais[2]
[1]Dept. of Computer Science & Engg - CEC, Bantwal, [2]Dept. of Computer Engineering - NITK Surathkal
[1]basappabk@gmail.com, [2]chgauravprasad@gmail.com, [2]alwyn.pais@gmail.com

Abstract — *In* the past decades, focus of computation has shifted to high performance computing like Grid Computing and Cloud Computing. In Grid computing, grid server is responsible for managing the all resources like processor, memory and CPU cycles. Grids are basically networks that pool resources, CPU cycles, storage or data from many different nodes used to solve the complex or scientific problem. However in this case, security is a major concern. Even most grid security researches focus on user authentication, authorization and secure communication. This paper presents DDoS and Data Modification attack scenario and also provides the solution to prevent it. In case of data modification attack, it shows how easy to read/forward/modify the data exchanged between a cluster head node and computing nodes. Therefore this paper provides the solution to protect the grid computing environment against Data Modification and DDOS attack.

Index Terms — *Grid* Security, Cluster Security, Denial of Service, Sync flood, ARP Poisoning

I. INTRODUCTION

Grid Computing Environment is a collection of cluster head nodes used to share the resources across the multiple domains or share resources among many computers to solve large-scale problems. The grid environment comprises of heterogeneous resources. The resource broker takes care of resource discovery, selection, aggregation, and data and program transportation; it initiates execution on a remote machine, and it gathers the results [1]. The Grid is a collection of cluster head nodes and cluster is a collection of computing nodes as shown in Figure 1. The job of Cluster head is to distribute the batch of jobs among the computing nodes; computing nodes execute those jobs and send the results back to cluster head. The job scheduler (GRAM) would be PBS, SGE or Candor.

In cluster, the Job selection or job scheduling is the process of picking which eligible job is placed into execution or initiated. To be eligible, the job must reside in an execution queue and be in the queued state. The Scheduler communicates with the Server via a socket based IPC, using the standard API.

This provides the capability of having the scheduler and the Server reside on different hosts. The Scheduler communicates with another process called the Machine Oriented Miniserver (MOM), it acts as a resource manager to obtain information about the resources which contains details of memory usage, CPU load average, and more. This information, taken as input to the scheduling program can be used to control job scheduling. The relationship between PBS, the Scheduler, and the Machine Oriented Miniserver can be seen in Figures 2 [2].

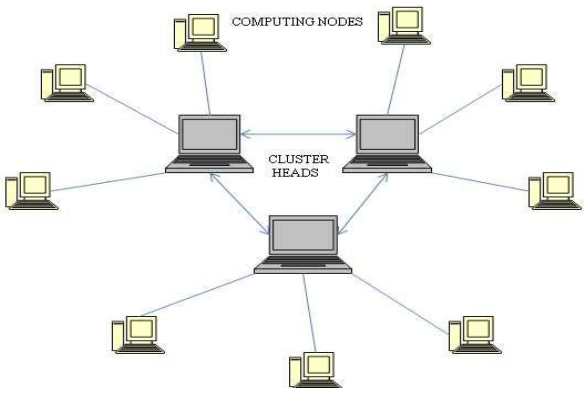

Figure 1: Generic view of Grid Cluster Computing Environment

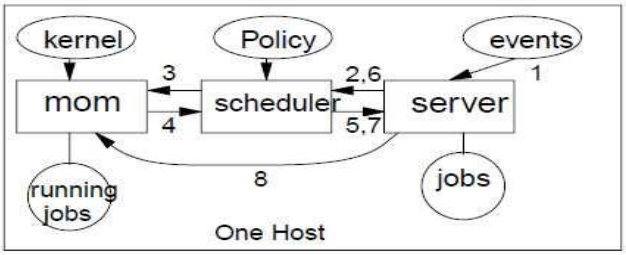

Figure 2: Batch Scheduling on a Single Host

1. Event tells Server to initiate a scheduling cycle.
2. Server sends scheduling command to Scheduler.
3. Scheduler requests resource info from MOM.
4. MOM retuns requested info.
5. Scheduler requests job info from server.
6. Server sends job status info to scheduler and Scheduler makes policy decision to run job.
7. Scheduler sends run request to server.
8. Server sends job to MOM to run.

Computational Grids are motivated by the desire to share processing resources among many organizations to solve large-scale problems [3, 4]. Very often, a Grid is used for executing a large number of jobs at dispersed resource sites. Each site executes not only local jobs but also jobs submitted from remote sites. Thus, job outsourcing becomes a major trend in Grid computing [5]. So while submitting the job from remote hosts or sites this becomes major security issue or vulnerability in Grid Computing Environment. Vulnerability is a flaw or weakness in a system's design, implementation, or operations that could be exploited to violate the system's security policy. There are many security issues in the computational grid environment mentioned in [6]. We have found data modification attack using ARP poising and DDoS. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks in the Computational Grid Environment. The main goal is to find or detect and mitigate the vulnerabilities in the Grid Computing Environment while submitting the job from grid client to cluster head node and then to computing nodes.

The rest of the paper is organized as follows. Section II contains related work on Grid Environment. Section III explains about attack scenario on Grid Computing environment. Section IV and V will explain about proposed solution for the found security issues DDOS attack and Data Modification attack in the grid computing environment. Section VI presents the Performance Evaluation of the proposed solution and Section VII concludes the paper.

II. RELATED WORK

DoS and DDoS attacks [7] are the most common attack today. A DoS attack involves sending large number of packets to a destination to prevent legitimate users from accessing information or services. The vulnerabilities of grid environment in the presence of DDoS have been presented in [4] and they have proposed a distributed defense system for Grid. A good classification of possible threats in grid computing can be found in [8], which is based on threats on different users associated with Grid. From which Distributed Denial of Service attack (DDoS) is an immense threat to grid computing. For example Sun's new on-demand grid computing service becomes a victim with a denial of service (DOS) attack on its first day of operation [9]. Author [10] uses entropy of distribution of source address for DDoS detection. The authors of [11] use entropy rate to discriminate the DDoS attack from legitimate traffic. Our objective in this paper is to protect the grid computing environment against DDOS and Data Modification attack.

III. ATTACK SCENARIO ON GRID COMPUTING ENVIRONMENT

In grid computing, we have a cluster setup underneath, which uses TCP protocol for communication on privileged ports. By default it uses port numbers 15001 to 15004 to communicate with the computing nodes. These ports are not blocked by the firewall since, using these ports only cluster head and computing nodes contact each other. In our attack scenario, instead of directly attacking the grid server we attack the cluster head node which acts as a scheduler in grid computing environment as shown in the Figure 3.

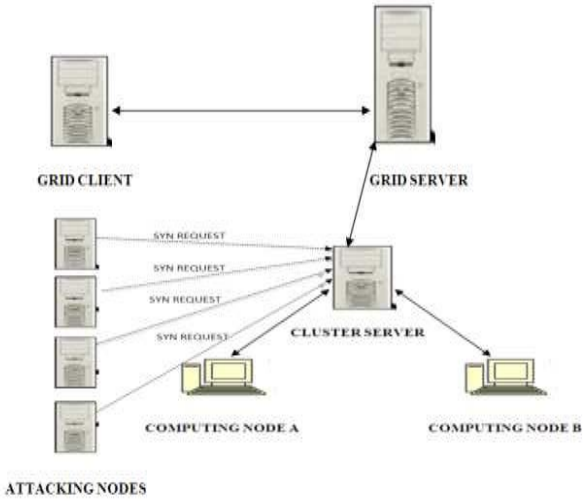

Figure 3: SYN Flooding attack

A. *SYN Flooding Attack*

To do SYN Flooding, any of the ports in the server should be open. In case of clusters, the attacker floods the cluster server with SYN request on privileged ports (15001:15004) using spoofed IPs. Spoofed IPs can be IPs which does not exists but in our scenario we take non existing IPs as well as IPs of the computing nodes. In response to sync packets from the attacker, the server sends acknowledgement which is part of three-way hand shake and allocates TCB memory to keep track the status of the connection. In case of the non existing IPs the cluster server will never get an acknowledgement there by filling up and retaining TCP status information in the TCB memory.

The server retransmits for few times and then the

TCB memory will be freed. In case of computing nodes, cluster server will be expecting an acknowledgement from the computing node .But acknowledgement will never be received, since the computing nodes have not initiated any connection. So in few microseconds, the rate at which the sync is received by cluster server from attackers will be much greater than the rate at which TCB memory is freed, which in due course of time exhausts the TCB memory. Due to lack of TCB memory subsequent request for TCP connection will be silently dropped, which may be legitimate connection from the computing nodes. This will deprive communication between cluster head and computing nodes which leads to a stage where in job will remain in the queue forever and cannot be scheduled to the computing nodes.

This results in unlimited delay in execution of job which was initiated by the grid client and sent to grid server which intern was sent to cluster server for execution. If this type of DoS/DDoS attack is done on every cluster head node lying below the grid server, will lead to Denial of Service with respect to the grid computing environment. So our aim is to provide the protection against the DDOS attack.

B. The ARP Poisoning (Man-In-The-Middle) Attack

The ARP Poisoning (Man-In-The-Middle) attack is also one of the security issues in the grid cluster environment. ARP Poisoning is a technique used to attack wired or wireless network. It may allow an attacker to sniff data frames on a local area network (LAN), modify the traffic, or stop the traffic altogether. The principle of ARP poisoning is to send fake, or "spoofed", ARP messages to an Ethernet LAN.

Generally, the aim is to associate the attacker's MAC address with the IP address of another node. Any traffic for that IP address would be mistakenly sent to the attacker instead. The attacker could then choose to forward the traffic to the actual default gateway or modify the data before forwarding it (man-in-the-middle attack). The “ettercap” tool is used to attack using the ARP Poisoning approach. Once the attacker spoof the IP address of the victim and target host, he can trace the all traffic passing from client to server and vice versa and also he can modify the data or job script between the cluster head and computing nodes. This is the major security issue in the grid cluster computing environment. So our aim is to provide the protection against the data modification attack in the grid cluster computing environment.

IV. PROPOSED MITIGATION FOR SYN FLOODING ATTACK IN GRID COMPUTING ENVIRONMENT

Grids are designed to make better utilization of resources by harvesting the resources that can work on complex problems which are being used when the peoples computer are free. Traditionally grids solve the problem of performance improvement in different ways. However, if it is vulnerable to DoS/DDoS attack, the true power and flexibility will be deprived. So we propose mitigation to DoS attack (SYN Flooding). The loophole used to launch the Sync flooding attack is usage of privileged port which is used for communication. One solution is to block the port which is ruled out, since it acts as an entry point for communication. The proposed mitigation algorithm validates the incoming packets which can be done by check each incoming packet and decide whether that packet is valid or a junk packet. The proposed algorithm for mitigation of sync flood attack can be divided into two parts, Part A and Part B. Part A is algorithm to generate valid IP list which is given below is called only once, whenever the cluster server starts and Part B will be running continuously until server stops.

C. Algorithm To Generate Valid IPs And MAC Address

Step 1: Get the number of computing nodes connected to the cluster server

Step 2: if count is greater than 1 and IP of the computing node is not equal to IP of cluster server then

a. Store the IP address of the computing node in IP file.

b. Store the Mac address of the computing node in MAC file

Step 3: Check validity of IP and MAC address pair and store it in IP MAC file

Step 4: Algorithm will return true if cluster server is having a valid computing node and stores the IP and MAC address of computing nodes else no computing nodes are available so no chances of SYN flooding attack

D. Mitigation Algorithm

Step 1: Listen to the privileged port

Step 2: Check the IP of each incoming packet and compare with ip list if equal then

Step 3: Check the mac address of the incoming packet and compare with macfile

Step 4: If equal then check ip and mac pair of the incoming packet and compare with ipmacfile. If equal then accept the packet else drops the packet, it might be spoofing of IP or MAC or both.

Step 5: else drop the packet and it indicates IP Address is been spoofed else drop the packet. Step 6: step 1 and 2 is repeated until cluster server stops

Step 6: step 1 and 2 is repeated until cluster server stops.

V. PROPOSED MITIGATION FOR DATA MODIFICATION ATTACK

In Cluster, it is know that server sends the job script to mom after getting the resource information from scheduler. That job script will be in the text format and it is easy to modify it when exchanged between a cluster head and computing nodes. The Figure 4 shows proposed Architecture for job scheduling on MOM host. Here we have used symmetric encryption/decryption (DES) algorithm with 64- bit key. We are encrypting job script by using secret shared key between client and server before transferring it to MOM as shown in Figure 4. In the final step the encrypted job script will be transferred to MOM. The MOM in the computing node will decrypt it before executing the job, then executes the job and sends the result back to the job submitter.

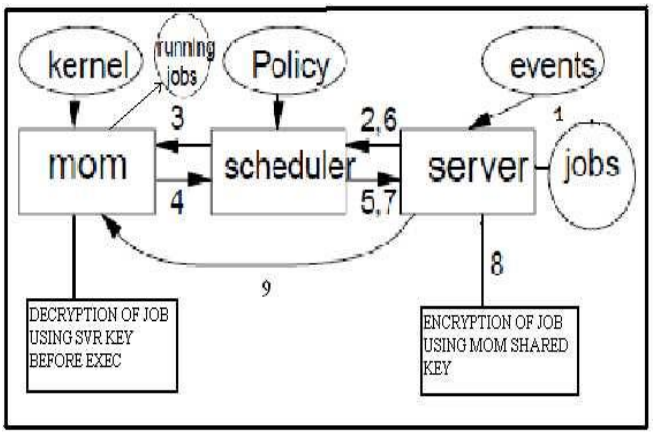

Figure 4: Proposed Architecture for job scheduling on a MOM Host

E. *Algorithm For Key Sharing*

We have used Diffie Hellman Key Exchange protocol to share secret key between MOM and SERVER.

Function:Key_Sharing_by_DHKE (hostaddr)

Step 1: MOM chooses a prime number '*p*' such that (*p*-1)/2 also a prime

Step 2: Find a generator *'g'* of '*p*' [12]. The algorithm to find the generator is given in section *F*

Step 3: The MOM then chooses 'α' a random number

Step 4: It then computes $mom = g^{\alpha} \bmod p$ and sends {*mom*, *g*, *p*} to PBS_SERVER

Step 5: PBS_SERVER chooses *'β'* a random number.

Step 6: PBS_SERVER computes $svr = g^{\beta} \bmod p$ and sends {*svr*} to MOM

Step 7: MOM Computes $Tmp\_Key = svr^{\alpha} \bmod p$

Step 8: PBS_SERVER Computes $Tmp\_Key = mom^{\beta} \bmod p$

Step 9: Compute *Original_Key* = MD5(*Tmp_Key*) to get 64-bit length key on both sides and store in the PBS_HOME/shared_keys/*hostname* directory

F. *Algorithm to select a prime 'p' number based on computing node IP Address*

Step 1: *Ip = A. B. C. D*

Step 2: $IpMapValue = 2^{24} * A + 2^{16} * B + 2^{8} * C + D$

Step 3: *qMapValue = IpMapValue* mod *Np* (Number of primes)

Step 4: *num = Np – qMapValue*

Step 5: Get the random number between (1, *num*)

Step 6: Update the *qMapValue = qMapValue + random_num*

Step 7: *p = primes*[*qindex*][*qMapValue*]

G. *Algorithm to encrypt and decrypt the job*

PBS_SERVER SIDE:

Step 1: Read the script; store it into buffer which is to be sent to PBS_MOM

Step 2: Get the key from \$PBS_HOME/shared_keys/*hostname*(MOM addr)

Step 3: Encrypt the script which is stored in a buffer using shared_key and send it to MOM

PBS_MOM SIDE

Step 1: Get the PBS_SEVER shared key from \$PBS_HOME/shared_keys/*hostname*

Step 2: Decrypt the script using shared key before executing it

VI. PERFORMANCE EVALUATION

Grid has proved to be better, it has some area to be focused .One such area is security. We have already addressed how a DDoS/DoS attack can be carried out on grid computing environment. So to do DDoS/DoS we have taken up Hping tool. The next stage we compare the stability of the proposed solution by comparing the results of the grid operation with proposed solution and with normal grid operation which is shown in the Figure 5.

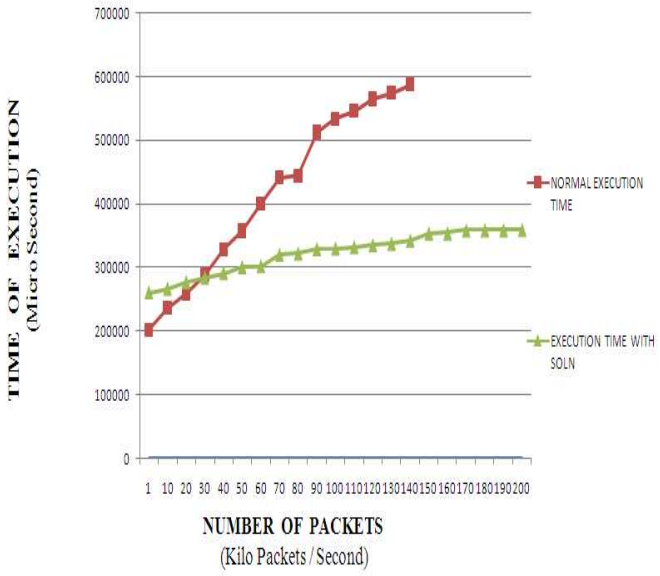

Figure 5: Performance of Proposed Solution in grid Vs Normal grid during attack scenario

The graph clearly specifies that the proposed solution works fine, even can handle more packets and the execution time for the job to get executed will remain almost same, even if the packet pumping rate increases drastically. The next issue to be handled is the performance of the proposed solution during normal scenario. To get the performance of the proposed solution, a set of jobs are submitted to the grid and its corresponding values i.e. job execution time are noted for both normal and the proposed solution. The Figure 6 is a graph with number of jobs is taken in X axis and time of execution is taken in Y axis.

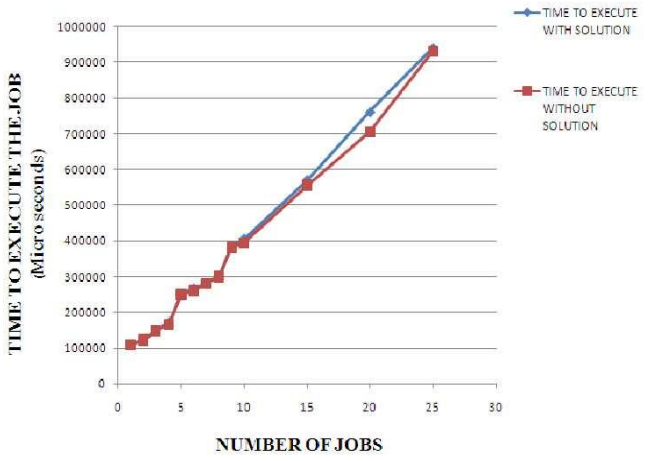

Figure 6: Performance of Proposed Solution in grid Vs Normal grid during normal scenario

As the number of jobs to be executed increase, time for job execution also increases. But the time taken by the by the server with the proposed solution is a little more when compared to normal grid server. The experimental results show an overhead of 1.99% in the performance of grid server with the proposed solution compared to the normal grid operation. We have also proposed and implemented the solution for ARP poisoning whose result is shown in the graph in Figure 7, shows the execution time for up to 25 jobs submitted to the grid system. The execution times for jobs with and without DES is almost similar up to 15 jobs but as the number of jobs increases the execution time with DES is more than the execution time without DES. Hence, the use of DES to encrypt the jobs doesn't add much of an overhead to the execution time of the jobs.

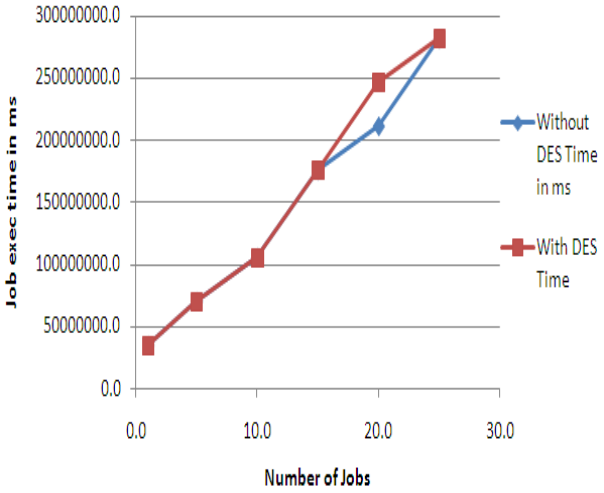

Figure 7: Job execution time with and without solution

The graph in Figure 8 shows the comparison between solution with random *p* and solution with *p* value based on IP address of the computing node. In this comparison, for execution of up to 20 jobs, the time taken to execute jobs is almost same and after that the time taken by *p* based on IP is less than random *p* with DES. So, we can say that there is no much over head involved while choosing a *p* value based on the IP address of the computing node.

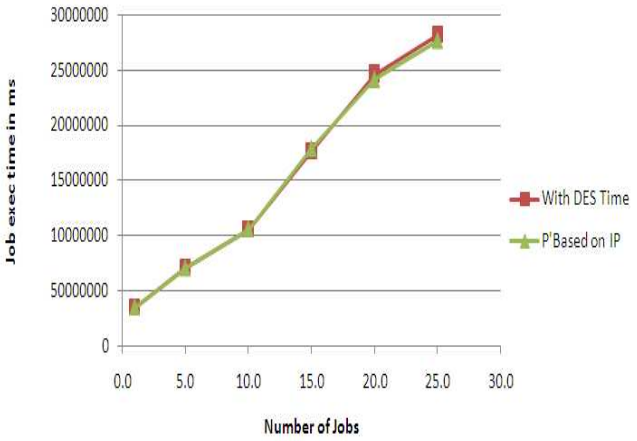

Figure 8: Comparison between solution with random p and solution with '*p*' value based on 'IP'

VII. CONCLUSION

In this paper we have presented that the grid computing environment is vulnerable to the SYN Flooding attack and data modification attack using ARP Poisoning approach. In case of data modification, the attacker can easily read, modify and forward the data between client (MOM) and server, because data exchanged between them is in plain text format. In case of DDoS, the attacker can easily attack the server using the existing network tools like Hping tool wherein it has the capability to pump large number of SYN packets there by leading to DDoS attack. We have proposed mitigation methods to overcome these security Vulnerabilities. Our experimental results show that encrypting the job doesn't add much overhead on the server. In case of Sync flooding attack, the proposed solution works fine in normal situation, consuming some extra micro second of time to execute when compared with the existing system. The performance of grid server with the proposed solution takes overhead of 1.99% compared to the normal grid operation which is quite reasonable considering the security benefits of the proposed solution.

REFERENCES

[1] Madhu Chetty, Rajkumar Buyya, *"WEAVING COMPUTATIONAL GRIDS: HOW ANALOGOUS ARE THEY WITH ELECTRICAL GRIDS?"*, IEEE 2002.

[2] Albeaus Bayucan, Robert L. Henderson, CasimirLesiak, Bhroam Mann, Tom Proett, Dave Tweten, *"Portable Batch System"*, Numerical Aerospace Simulation Systems Division, NASA Ames Research Center, Moffett Field, CA, 1999.

[3] F. Berman, G. Fox and T. Hey (eds.), *"Grid Computing: Making the Global Infrastructure a Reality"*. Wiley, 2003

[4] M. Cosnard and A. Merzky, *"Meta- and Grid-Computing"*, in Proceedings of the 8th International Euro-Par Confrence,August 2002, PP.861-862.

[5] Shanshan Song, Kai Hwang and Yu-Kwong Kwok, *"Trusted Grid Computing with Security Binding and Trust Integration"*, Internet and Grid Computing Laboratory, University of Southern California, EEB-212, 3740 McClintock Avenue, Journal of Grid Computing (2005) 3: 5373

[6] Yuri Demchenko, *"White collar Attacks on Web Services and Grids"*, Grid Security threats analysis and Grid Security Incident data model definition, Draft Version 0.2, August 12, 2004.

[7] J. Mirkovic, J. Martin and P. Reiher, *"A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms"*, ACM Computer Communications Review, Vol.34, No. 2, April 2004.

[8] N. Jiancheng, L. Zhishu, G. Zhonghe,S. Jirong, *"Threat analysis and Prevention for grid and web security services"*, SNPD, pp. 526-531, 2007.

[9] http://www.sun.com/service/sungrid/index.jsp

[10] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred. *"Statistical approaches to DDoS attack Detection and response"*. In Proc of DARPA Information Survivability Conference and Exposition, 2003

[11] S.Yu, W. Zhou, *"Entropy-based Collaborative Detection of DDoS attacks on Community Networks"*, In Proc. of IEEE international conference on pervasive computing and Communications, 2008.

[12] Avinash D., Radhesh M., Alwyn R. Pais, *"Throttling DDoS Attacks using Discrete Logarithm Problem"*, SECRYPT 2010, July 26-28, 2010 Athens, Greece.

Mr. Basappa B. Kodada, Asst. Professor, is a Faculty of Canara Engineering College, Mangalore at Vishweswaraya Technological University Belgaum, Karnataka. Prof. Basappa B. Kodada obtained his B.E (Computer Science and Engineering) from Vishweswaraya Technological University Belgaum, Karnataka and M.Tech (Computer Science & Engg – Information Security) from National Institute of Technology Karnataka in 2007 and 2010 respectively. His research areas include the Information and Network Security, Distributed Computing, Grid Computing, Cloud Computing and Mobile Computing. He has published 2 international journal and 3 international conference papers.

Mr. Gaurav Prasad, Lecturer, is a Faculty of NITK Surathkal, Mangalore. Prof. Gaurav Prasad obtained his M.Tech (Computer Science & Engg – Information Security) from National Institute of Technology Karnataka in 2010. His research areas include the Information and Network Security, Grid Computing and Cloud Computing. He has published 1 international journal and 2 international conference papers.

Mr. Alwyn R.Pais, Asst. Professor, is a Faculty of NITK Surathkal, Mangalore. Prof. Alwyn R.Pais obtained his B.Tech (Computer Science and Engineering) from Mangalore University, Karnataka and M.Tech (Computer Science & Engineering) from Indian Institute of Technology – Bombay. Currently He is pursuing PhD from NITK Surathkal. His research areas include the Information Security, Cryptography, Computer Algorithm and Computer Vision. He has published several journal and conference papers.

