World Applied Sciences Journal 23 (11): 1418-1424, 2013
ISSN 1818-4952
© IDOSI Publications, 2013
DOI: 10.5829/idosi.wasj.2013.23.11.950

DDoS Verification and Attack Packet Dropping Algorithm in Cloud Computing

Muhammad Zakarya
Department of Computer Science, Abdul Wali Khan University, Mardan, Pakistan

Submitted: Jun 8, 2013; **Accepted:** Jul 17, 2013; **Published:** Jul 25, 2013

Abstract: DDoS attacks on the World Wide Web in broad-spectrum and predominantly in modern cloud computing has become a noticeable issue for researchers in academia and industry related to the field of computer sciences. DDoS attacks are cool to provoke but their uncovering is a very challenging and dingy task and therefore, an eye-catching weapon for hackers. Hence DDoS torrents do not have familiar appearances; therefore currently existing IDS cannot identify and discover these attacks perfectly. Correspondingly, there implementation is a bamboozling task. In practice, gossip based detection machines are used to detect such types of attacks by exchanging stream of traffic over line but still results in network congestion and have upstairs of superfluous and bonus packets. Keeping the above drawbacks in mind, we have proposed a DDoS detection and prevention mechanism in [1], that has the attractiveness of being easy to adapt and more trustworthy than existing counterparts. We have introduced entropy based detection mechanism for DDoS attack detection. In [2] we have implemented the same algorithm to grids platform, where we obtain an accuracy of 90%. Our proposed solution has no overhead of extra packets, hence resulting in good QoS. In this paper we are going to implement the same algorithm on clouds.

Key words: DdoS · QoS · IDS · AS · ADS · CloudSim

INTRODUCTION

Cloud computing is a most modern and hottest buzzword nowadays, emerges as a key service of the utility or on-demand computing [1] which builds on decade of research in the ground of computer networking, World Wide Web and software services. People are looking for fastest, QoS, secure, efficient and reliable services and that's why a number of researchers are devoted to the distributed computing research including clusters, HPC, grids and clouds. Cloud computing put forwards a service oriented architecture, reduced information technology overhead for the end-user, enormous and huge flexibility and reduced total cost of ownership. Recent attacks on the clouds especially DDoS poses as a potential intimidation and danger to this key technology of the expectations and future. In this paper we are going to present a new DDoS attack confirmation and packet dropping algorithm for cloud environment. An entropy based ADS approach is presented to mitigate the attack which further improves network performance in terms of computational time, QoS and high availability. SaaS, PaaS, IaaS and IT foundation are four basic types of cloud computing. DDoS attacks are thrown through carriage of a large amount of packets to an objective machine, using instantaneous teamwork of numerous hosts which are scattered throughout the cloud computing environment. The rest of paper is organized as follows. In section 2 some related work is presented. Section 3 is about our previous work followed by existing problem in next section. Solution to the existing problem is discussed in section 5 followed by simulation results in next section. Performance evaluation are shown in section 7. Finally some concluding remarks and future work is discussed in section 8.

Related Work: According to [3, 4], any statements that have some shock and importance are called information. Some believe that information theory is to be a subset of communication theory, but we consider it much more. Entropy is a measure of the chaos of a group of particles i.e. 2nd law of thermodynamics. If there are a number of possible messages, then each one can be expected to occur after certain fraction of time known as probability of the message. In [9-15] Shannon proved that information content of a message is inversely related to its probability of occurrence. To summarize, the more unlikely a message is, the more information it contains. In [6, 16], Entropy H(X) is given by

Corresponding Author: Muhammad Zakarya, Department of Computer Science,
Abdul Wali Khan University, Mardan, Pakistan.

$$H(X) = -\sum_{x \in X}^{n} p(x) \log p(x) \qquad (1)$$

The log is to the base 2 and entropy is expressed in bits. To say uncertainty is directly proportional to entropy i.e. more accidental they are, more entropy is there. The value of entropy lies between 0 and log(n). The entropy value is smaller when the class distribution belongs to only one and same class while entropy value is larger when the class distribution is more even. Therefore, comparing entropy values of some traffic feature to that of another traffic feature provides a mechanism for detecting changes in the unpredictability. We use traffic distribution like IP address and application port number i.e. (IP address, Port). If we wants to calculate entropy of packets at a single or unique source i.e. destination, then maximum value of *n* must be 2^{32} for IPV4 address. Similarly if we want to gauge entropy at multiple application ports then value of *n* is the total number of ports [5, 7, 17, 18]. In similar way, p(x) where $x \in X$, is the probability that X takes the value x. We randomly examine X for a fix time window (w), then $p(x) = m_i/m$ Where, m_i is the total number we examine that X takes value x i.e

$$m = \sum_{i=1}^{n} m_i \qquad (2)$$

Putting these values in entropy equation 1, we get

$$H(X) = -\sum_{i=1}^{n} (m_i/m) \log (m_i/m) \qquad (3)$$

Similarly, if we want to calculate the probability p(x), then m is the entire number of packets, but m_i is the number of packets with value x at destination as source [8]. Mathematically given as

$$P(x) = \frac{\textit{Nunber of packets with } x_i \text{ as source (destination) (address)}}{\text{Total number of packets}} \qquad (4)$$

Again if we want to calculate probability p(x) for each destination port, then

$$P(x) = \frac{\textit{Nunber of packets with } x_i \text{ as source (destination) (port)}}{\text{Total number of packets}} \qquad (5)$$

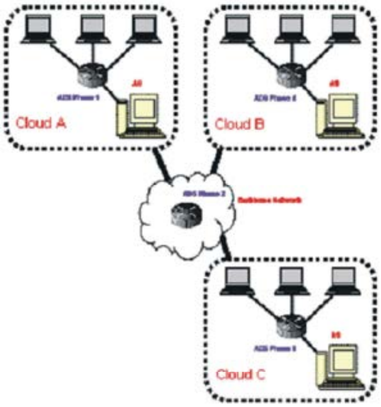

Fig. 1: Proposed Cloud Architecture [1]

Remember that total number of packets is the number of packets observed in a specific time slot (*w*). When this calculation finishes, normalized entropy is calculated to get the overall probability of the captured flow in a specific time window (*w*). Normalized Entropy is given by

$$\text{Normalized entropy} = (H/\log n_0) \qquad (6)$$

Where n_o is the number of dissimilar values of *x*, in a specific time slot (*w*). During the attack, the attack flow dominates the whole traffic, resulting in decreased normalized entropy. To confirm our attack detection, again we have to calculate the entropy rate i.e. growth of entropy values for random variables, provided that the limit exists and is given by

$$H(\chi) = \lim_{n \to \infty} \frac{1}{n} H(x_1, x_2 \ldots x_n) \qquad (7)$$

Proposed Solution and Results: In [1] the authors proposed a cloud architecture and a DDoS detection mechanism that has the beauty of being easy to adapt and more reliable than existing counterparts. The author's claims, that their proposed solution has no overhead of extra packets, hence resulting in good QoS. The architecture is shown in Fig 2. The whole cloud environment is divided into multiple sites either on geographical or administrative base. Every CS is under the control of a powerful AS. Our ADS is installed on every edge router. Our confirmation algorithm needs to be installed on subsequent and attached router to the edge router. Once DDoS [19-21] is detected at edge router, the flow is transferred to next neighboring router, where again the flow is checked against those information that were collected on edge router. If there is no change the attack is confirmed and the packet is discarded or dropped. Otherwise the packet is thrown to its destination on its way.

CloudSim was used for the evaluation of this approach. Results seen are of interest but high network access can lead to increase false positive rate. In next section we are going to propose a confirmation algorithm to limit these false positives. Our ADS can detect 100% DDoS attack only in case of good threshold value, which is one of the most challenging tasks in developing any ADS. We conclude our story that a threshold value of 0.97 results in good detection rate. A value greater than 0.97, results in good detection rate i.e. 100 % DDoS detection but generate more false positive alarms, as the value is increased from 0.97 to 1.0. In [2] we guessed a perfect threshold value of 0.95 while simulating in GridSim. Differences are due to the high number of packets. We conclude by examining small, medium and large flow of network packets, that where number of packets are increased in a platform, we have to set the threshold value a little bit larger.

The steps in algorithm are as under. Fig 5 shows the flow diagram of detection algorithm.

Existing Problem: We have proposed a DDoS detection and prevention mechanism in [1], that has the beauty of being easy to adapt and more dependable than existing counterparts. As, in service level security issues DoS, DDoS and network overcrowding, are most important. Solving the dispute of DDoS attack also results in network high availability as well as good QoS. The problem in that solution was that, in huge network usage or congested network flow our proposed detection algorithm will raise the attack alarm i.e. false positives, but it is not always be the case. To confirm the attack flow and decide to flush out or washout the flow, we are going to propose a confirmation algorithm, in this paper.

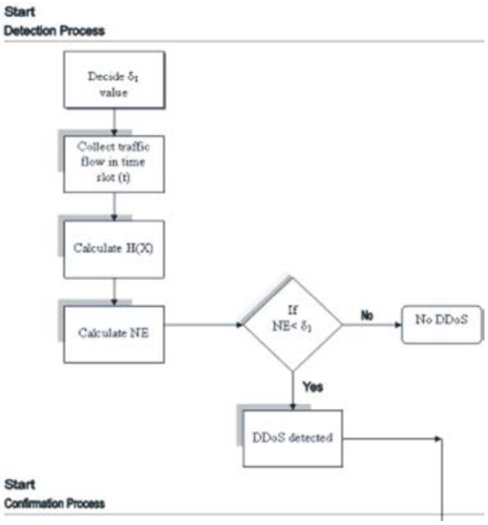

Fig. 2: Flow diagram [1]

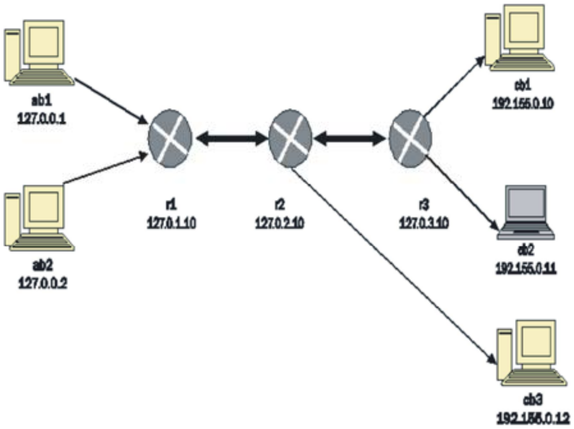

Fig. 3: Confirmation Algorithm

Proposed Packet Dropping Algorithm: In [1, 3, 22] the authors proposed entropy rate for confirmation of the attack flow, but still no exact solution was proposed. Entropy rate shows the increase or decrease ratio of distribution. We are going to extend our idea in this article and will propose and study a DDoS confirmation algorithm. Based on the results of such a confirmation algorithm the router will decide either to allow the flow of packets or to discard and drop that packet flow. We need such an algorithm because during high network access our DDoS detection algorithm will generate false positives and will alert the next edge router for DDoS attack, but it might not be the case. Our ADS is installed on each edging router. Our verification algorithm needs to be installed on consequent and attached router to the edge router [22-26]. Once DDoS is detected at edge router, the flow is transferred to subsequently adjacent router, where for a second time the flow is checked against those information that were claimed [27, 28] on edge router. If there is no alteration the attack is confirmed and the packet is superfluous one and hence needs to be dropped. Otherwise the packet is thrown to its target node or system on its own way. We have used CloudSim for simulation of our algorithm and have compared a number of cases to conclude performance evaluation.

A simple and straightforward solution is to run the same algorithm on receiver side router. But the problem is that we are going to detect and drop the packet flow early i.e. near the source. Suppose in Fig 6 below the user ab1

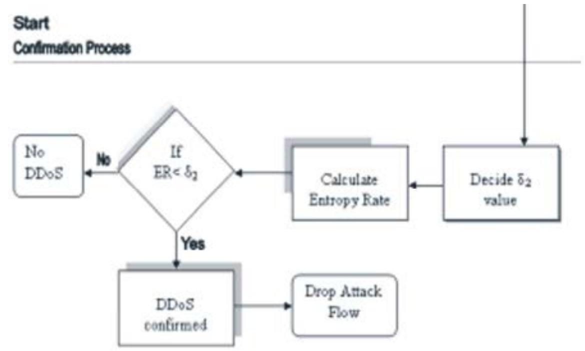

Fig. 4: Flow diagram

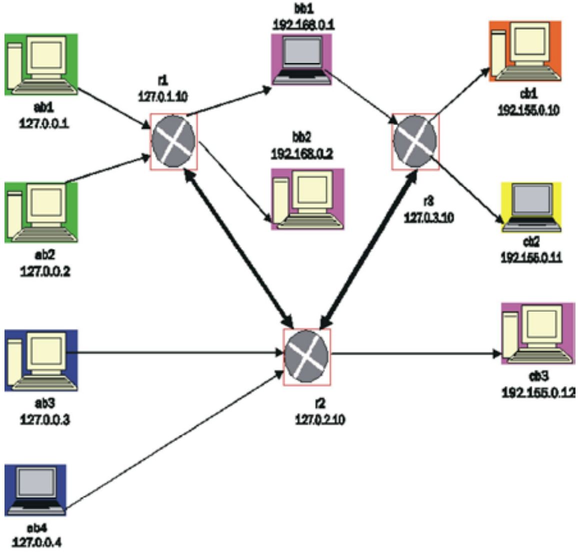

Fig. 5: Simulations Study

sends 90 packets to cb1, 91 packets to cb2 and 34 packets to cb3. When entropy is calculated on r1, the attack is detected. When this flow reaches to r2, the packets that were addressed to cb3 are directed on different way. Again if we calculate entropy of ab1 on r3, no attack is detected. It results in, if we calculate entropy i.e. if we run our detection algorithm two times on edge router to sender and receiver, then to some extent we will accurately measure DDoS and can drop only attack packets.

If the algorithm calculates same values, it means the attack is confirmed otherwise the packets are forwarded to its destination. The problem is that we need to detect and confirm the attack near to the source, so that the bandwidth is not wasted. The goal cannot be achieved in this solution. We can run the same detection algorithm on next edge router but still if the network is so large consisted upon 100 routers. There is the possibility that

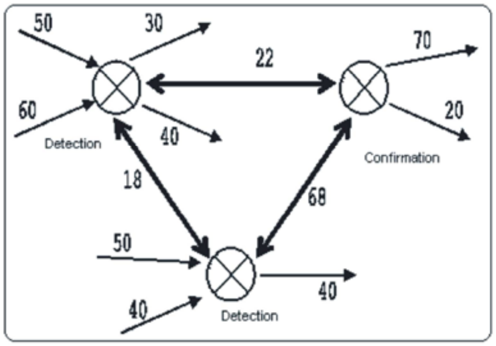

Fig. 6: Packet Flow Diagram

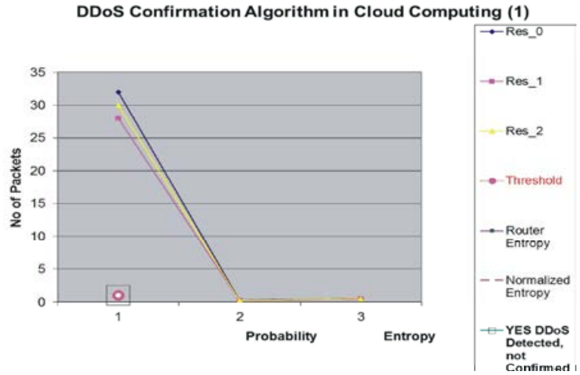

Fig. 7: DDoS Detection on Router 1

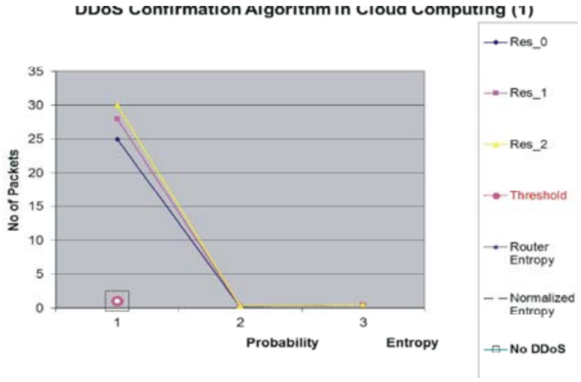

Fig. 8: DDoS Detection on Router 2

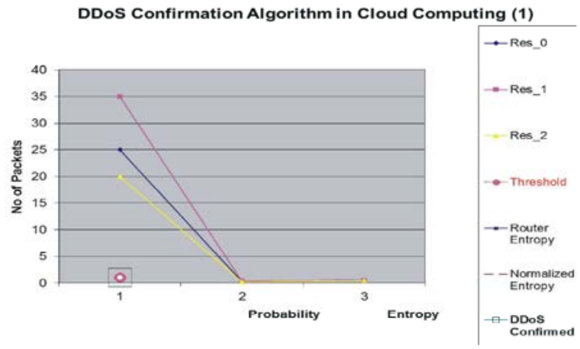

Fig. 9: DDoS Confirmation for Router 1 flow on Router 3

Table 1: Traffic at Router 1

Source node	Destination node	No of packets	R1	R3	Entropy (R1)	Entropy (R3)
AB1	CB1	20	12	8	0.35	0.27
AB2	CB1	20	4	16	0.17	0.40
AB1	BB1	30	15	15	0.39	0.39
AB2	BB2	40	32	8	0.52	0.28

Router entropy for R1 is 1.43 and normalized entropy for R1 is 0.90. Similarly router entropy for R3 is 1.35 and normalized entropy for R3 is 0.85.

Table 2: Traffic at Router 2

Source node	Destination node	No of packets	R1	R3	Entropy (R1)	Entropy (R3)
AB3	CB1	10	3	7	0.16	0.29
AB4	CB1	20	11	9	0.37	0.33
AB3	CB3	40	21	19	0.49	0.47
AB4	CB2	20	18	2	0.46	0.12

Router entropy for R1 is 1.49 and normalized entropy for R1 is 0.94. Similarly router entropy for R3 is 1.21 and normalized entropy for R3 is 0.77.

Table 3: Traffic at Router 3

Source node	Destination node	No of packets	Entropy (R1)
AB1	CB1	20	0.48
AB2	CB1	20	0.48
AB3	CB1	10	0.35
AB4	CB1	20	0.48
AB4	CB2	20	0.48

Router entropy for R3 is 2.28 and normalized entropy for R3 is 0.98.

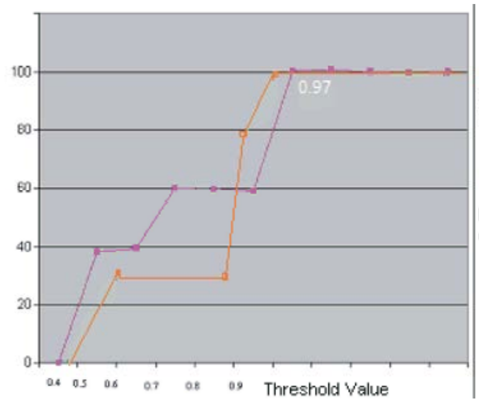

Fig. 10: DDoS detection and confirmation rate

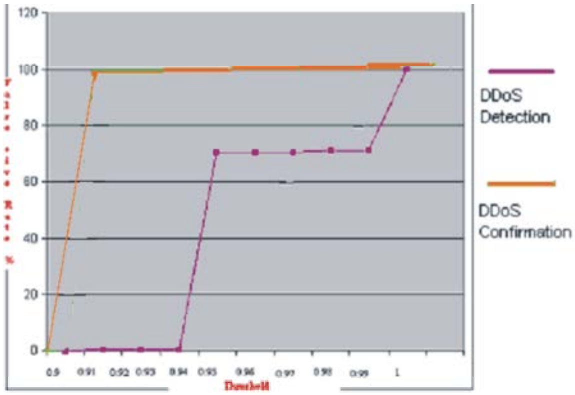

Fig. 11: DDoS false positive rate

the attack flow will remain on one path crossing over multiple routers. It will confirm the attack without any concern that in future the flow may be distributed over multiple paths.

Following are the steps for confirmation of the DDoS attack.

- Decide a threshold value δ_2
- Calculate entropy rate on edge router using Equation VII
- Compare entropy rates on that router, if =< δ_2, DDoS confirmed
- Drop the attack flow

Simulations Study and Results: Fig 7 shows the simulation environment that was created in CloudSim Simulator. To compare grids and clouds we have implemented the same scenario to extract more results [2]. The only difference is that the threshold value in higher than that was considered in [2]. The threshold value of 0.97 was always adjusted in [1], during the detection phase.

The above simulation environment was designed and developed in CloudSim simulation environment. Routers are connected to each other over a 10 Mbps link (◆), while all other connections are made at 1 Mbps link (↔). The reason behind this terminology is clear as router forward more data packets as compared to a single transmitting node.

Detection algorithm was executed on Router 1 and Router 2. On both routers attack was detected. The confirmation algorithm was executed on Router 3. The attack was not confirmed on this router; hence the flow was delivered to its destination nodes.

Fig 6 shows packets flows that were captured during the experiments. In our experiment, our detection algorithm shows that on routers 1 DDoS was detected but not confirmed. Similarly on Router 2, no DDoS flow was detected. During the conformation process on router 3, the flow was confirmed as an attack, hence packet drop mechanism was activated and the flow was successfully dropped.

Performance Evaluation: We observed that a threshold value of 0.97 results in good detection rate and a threshold value of 0.90 results in good confirmation.

A value greater than 0.97 and 0.90, results in good detection rate and confirmation i.e. 100 % DDoS detection and confirmation, respectively but generate more false positive alarms, as the value is increased from 0.97 to 1.0 i.e. false detection alarm or 0.90 to 1.0 i.e. false confirmation alarm. The reports are shown in figure 10 and figure 11, which are self-explanatory. Our experiments show that as more attacks are detected, more attacks are also confirmed and vice versa. In some situations that might not be the case, as its not assured that more network traffic will always cause DDoS. Still the topic needs researcher's attention for further exploration and solutions.

Conclusion and Future Work: In this paper, we have proposed a new solution and algorithm [25-27] to DDoS attack confirmation and attack packet dropping for cloud computing [7, 8]. In previous version of this article we introduced an ADS for recognition and early prevention of DDoS attacks in our suggested architecture. The problem of huge network access resulted false positive alarms. That issue was subject of this article. Our DdoS attack packet dropping algorithm will confirmthe attack flow, if it is an attack flow, the flow is discarded otherwise the flow is considered legitimate data packets and are forwarded to its destination, without any concern that it was targeted as a DDoS attack flow on the edge router. In future the proposed design and suggestion may be actually implemented over cloud computing platform to precisely detect DDoS attacks [28-30]. The idea may also be extended for recovery mechanism for DDoS attacks.

REFERENCES

1. Zakarya, M. and A.A. Khan, 2012. Cloud QoS, High Availability and Service Security Issues with Solutions. IJCSNS, 12(7): 71.
2. Zakarya, M. and S. Afzal, 2013. DDoS Confirmation and Attack Packet Dropping Algorithm in On-Demand Grid Computing Platform. VAWKUM Transaction on Computer Sciences, 1(1).
3. Claude Shannon, E., 1948. "A Mathematical Theory of Communication",
4. Claude Shannon, E., 1949. "Communication Theory of Secrecy Systems",
5. Cloud Security Alliance. "Top Threats To Cloud Computing". Technical Report, March 2010. http://www.cloudsecurityalliance.org/topthreats.html.
6. David Applebaum, "Probability and Information (An Integrated Approach)", Cambridge University Press, 2008.
7. Thomas M. Cover and Joy A. Thomas, 2006. Elements of Information Theory, Second Edition,
8. Dennis Arturo Ludeña Romaña and Yasuo Musashi, "Entropy Based Analysis of DNS Query Traffic in the Campus Network", Japan
9. George Nychis, 2007. "An Empirical Evaluation of Entropy-based Anomaly Detection", May 2007
10. Zakarya, M., AA. Khan and H. Hussain, "Grid High Availability and Service Security Issues with Solutions", ICIIT 2010, 978-1-4244-813 8-5/10 / $ 26.00 C 2010 IEEE
11. Zakarya, M. and I. Ur Rahman, 2013. A Short Overview of Service Discovery Protocols for MANETS. VAWKUM Transaction on Computer Sciences, 1(2).
12. Meenakshi, S. and S.K. Srivatsa, 2009. "A Comprehensive Mechanism to reduce the detection time of SYN Flooding Attack",
13. Preeti, Yogesh Chaba and Yudhvir Singh, 2008. "Review of Detection and Prevention Policies for Distributed Denial of Service Attack in MANET",
14. Thomas M. Cover and Joy A. Thomas, 2006. "Elements of Information Theory", Second Edition,
15. Manzur Murshed and Rajkumar Buyya, Using the GridSim Toolkit for Enabling Grid Computing Education, Monash University, Australia.
16. George Nychis, 2007. "An Empirical Evaluation of Entropy-based Anomaly Detection".
17. Yi-Chi Wu, Wuu Yang and Rong-Horg Jan, "DDoS Detection and Trace-back with Decision Tree and Gray Relational Analysis", National Chiao Tung University, Taiwan.

18. Zakarya, M., N. Dilawar, M.A. Khattak and M. Hayat, 2013. Energy Efficient Workload Balancing Algorithm for Real-Time Tasks over Multi-Core. World Applied Sciences Journal, 22(10): 1431-1439.
19. Zakarya, M., I.U. Rahman, N. Dilawar and R. Sadaf, An integrative study on bioinformatics computing concepts, issues and problems. International Journal of Computer Science (IJCSI), 8(6).
20. Zakarya, M. and I.U. Rahman, 2013. A Secure Packet Drop Defense Mechanism in Wireless Mobile Ad-hoc Networks, IJREAT 2013
21. Cha, B. and J. Kim, 2011. Study of Multistage Anomaly Detection for Secured Cloud Computing Resources in Future Internet. In Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference On, pp: 1046-1050. IEEE
22. Kar, S., 2009. An Anomaly Detection Scheme for DDoS Attack in Grid Computing (Doctoral dissertation).
23. Syed Navaz, S.A., V. Sangeetha and C. Prabhadevi, 2013. Entropy based Anomaly Detection System to Prevent DDoS Attacks in Cloud. International Journal of Computer Applications, 62(15): 42-47.
24. Khan, A.A. and M. Zakarya, 2010. Performance Sensitive Power Aware Multiprocessor Scheduling in Real-time Systems. Technical Journal UET Taxila (Pakistan).
25. Zakarya, M., I. Rahman and I. Ullah, 2012. An Overview of File Server Group in Distributed Systems.
26. Zakarya, M., I.U. Rahman and A.A. Khan, (2012, October). Energy crisis, global warming and IT industry: Can the IT professionals make it better some day? A review. In Emerging Technologies (ICET), 2012 International Conference on (pp: 1-6). IEEE.
27. Goyal, U., G. Bhatti and S. Mehmi, A Dual Mechanism for defeating DDoS Attacks in Cloud Computing Model.
28. Jeyanthi, N. and N.C.S.N. Iyengar, 2012. An Entropy Based Approach to Detect and Distinguish DDoS Attacks from Flash Crowds in VoIP Networks. International Journal of Network Security, 14(5): 257-269.