A Study on Securing Software Defined Networks

Chapter *in* Lecture Notes in Computer Science · October 2017

DOI: 10.1007/978-3-319-68786-5_38

Raihan Ur Rasool
National University of Sciences and Technology
43 PUBLICATIONS **114** CITATIONS
SEE PROFILE

Hua Wang
Victoria University Melbourne
194 PUBLICATIONS **1,157** CITATIONS
SEE PROFILE

Wajid Rafiq
National University of Sciences and Technology
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

A Study on Securing Software Defined Networks

Raihan Ur Rasool[1], Hua Wang[1], Wajid Rafique [2], Jianming Yong[3], Jinli Cao[4]

[1] Victoria University, Melbourne Australia
[2] National University of Sciences and Technology, Islamabad, Pakistan
University of Southern Queensland, Toowoomba, Australia
La Trobe University, Bundoora, Australia
raihan.rasool@live.vu.edu.au, hua.wang@vu.edu.au, rafiqwajid@gmail.com, Jianming.Yong@usq.edu.au,
j.cao@latrobe.edu.au

Abstract. Most of the IT infrastructure across the globe is virtualized and is backed by Software Defined Networks (SDN). Hence, any threat to SDN's core components would potentially mean to harm today's Internet and the very fabric of utility computing. After thorough analysis, this study identifies Crossfire link flooding technique as one of the lethal attacks that can potentially target the link connecting the control plane to the data plane in SDNs. In such a situation, the control plane may get disconnected, resulting in the degradation of the performance of the whole network and service disruption. In this work we present a detailed comparative analysis of the link flooding mitigation techniques and propose a framework for effective defense. It comprises of a separate controller consisting of a flood detection module, a link listener module and a flood detection module, which will work together to detect and mitigate attacks and facilitate the normal flow of traffic. This paper serves as a first effort towards identifying and mitigating the crossfire LFA on the channel that connects control plane to data plane in SDNs. We expect that further optimizations in the proposed solution can bring remarkable results.

Keywords: Network Security, Target Link Flooding, Software Defined Network

1. Introduction

SDN has proved to be one of the vastly adopted network paradigms, which has attracted a lot of attention from industry and academia. SDNs knit the fabric of today's computing power houses and most of the Internet. Hence ensuring SDN's security is of a paramount importance. There is a vast range of available SDN devices from manufacturers like CISCO, HP, and NEC, and OpenFlow [1] is a well-recognized protocol for SDN controller implementation. The control plane of SDN is regarded as the brain of the network and provides functions like network management, configuration, and exchange of routing table information. SDN's data plane is recognized as the forwarding plane, as the routers or switches here do as instructed by the control plane. The controller updates the flow tables of switches which contain the information of how to process the incoming packets.

Link Flooding Attacks (LFA) has emerged as one of the stealthiest attacks on the internet, these attacks consume resources of the target servers and cause a denial of service [16]. These attacks are usually implemented by bots, which send low rate legitimate traffic to the selected decoy servers that are not the target servers but lie in the path to the target server. In this way sending low rate traffic to these servers will cause the links going to the target server to be flooded and the target server becomes irresponsive. Figure 1 shows the depiction of one such attack. A recent example of botnet attacks is Mirai botnet attack which brought most of America's internet down and it was supposed to be one of the largest attacks in the history of America [2]. Traditional link flooding attacks consume resources of the targets, but these link flooding attacks use bots to deplete resources of selected links. These links are carefully chosen that lead to a selected server hence flooding these links will prohibit traffic to reach to the server and cause a denial of service. Precisely saying, it doesn't attack the target link directly, which makes the detection and defense of such attacks very difficult. In the past few years, various link flooding attacks are introduced, the attacks which are more critical are the Crossfire Attack [16], Coremelt Attack [17] and The Spamhaus Attack [18].

The adversary [16] makes use of bots and sends legitimate flows like TCP to the targeted link, like any other legitimate users using the network resources. Because of this indirect strategy, the targeted links don't receive any malicious traffic. In these attacks, it is also very difficult to differentiate between legitimate users and malicious bots because they are also using the valid IP addresses. In these attack first of all the adversary builds the network profile by sending traceroute packet, hence the adversary builds the network path and identifies the target server and critical links that are to be attacked. The adversary select the servers called decoy servers that are not the target servers, and analyze the bot decoy pairs that are required to perform the flooding operation which is sufficient to flood the links going to the target server. Hence the adversary will be able to send legitimate flows from bots to decoys and perform the flooding to the target link. The links will be flooded and traffic will not be able to access the target servers and denial of attack occurs.

In software defined networks, OpenFlow [15] is a reference implementation. It is a standard communication profile between data plane and control plane. In SDN, OpenFlow specifications define that packet routing is done by using traffic flows in the network. Each network device that is termed as switch needs to maintain a flow table, which contains flow rules installed by control plane to handle each incoming packet. Each OpenFlow switch also maintains a communication channel to an external controller in the control plane.

Following are the main contributions of this work:

- We present a detailed comparative literature review, analyzing and categorizing each one of the attacks and mitigation techniques relevant to SDN.
- We have identified a problem area in SDN where certain attacks can remain undetected, and can potentially disrupt the whole network.

- We propose and lay a foundation for a framework to detect and mitigate crossfire attacks on the control plane to data plane link in SDN. To the best of our knowledge, it is the first effort in this direction.

The remainder of this paper is organized as follows: In Section 2, we present the related work. In Section 3, we present the critical analysis of link flooding attacks. Section 4 proposes a methodology and conceptual framework and Section 5 presents the conclusions.

2. Literature Review

During recent years, link flooding attacks have gained enormous attraction from industry and academia. Various techniques have been proposed to mitigate link flooding attacks [21, 23]. Literature also presents different types of such attacks and points out Crossfire attacks [16] as the most difficult to identify and mitigate.

2.1 Types of Link Flooding Attacks

In the past few years, various link flooding attacks are introduced while other privacy and security challenges are studied in different domains [17, 18, 20, 22, 24-26]. The attacks which are more critical are: the Crossfire attack, Coremelt attack [17] and The Spamhaus attack [18]. The most critical and recent attacks are the crossfire attacks, which don’t attack directly to the intended server but congest the links that lead to the target. Hence targeting these links deplete the route to the server which becomes irresponsive. The link flooding and mitigation techniques can broadly be categorized into three categories

Table 1. Link flood mitigation technique types

	Technique Type	Basic Principle
1.	Traffic Engineering principles [4,6,7,13,14]	By rerouting traffic to different paths
2.	Deploying SDN approaches [3, 8, 9]	Using SDN principles to manage and control the traffic
3.	Link observation techniques [5, 10, 11, 12]	Works by observing the link for flood detection

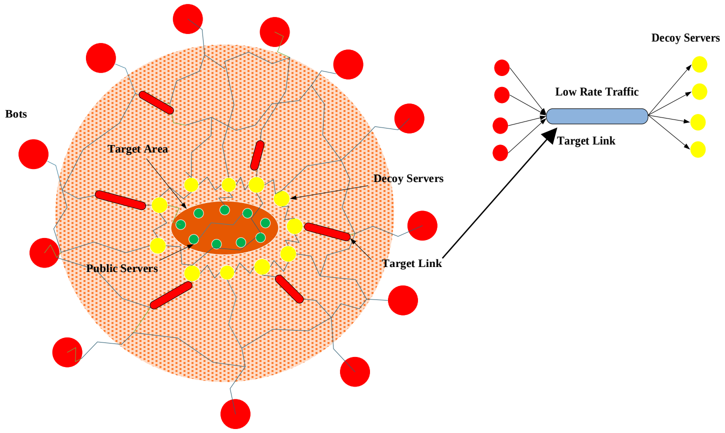

Figure 1: Cross Fire Attacks [16]

2.2 Traffic Engineering based approaches

Takayuki et. al. [4] proposed a proactive mechanism to mitigate link flooding attacks that make use of traceroutes packets. Their technique is based on the fact that traceroutes packets are increased in various regions of the network when the network is under link flooding attack. This technique uses the number of traceroute packets, so normal link congestion and target link flooding attacks can be differentiated because the behavior of increase of traceroute packets is independent of a link congestion, but the limitation of this technique is that it is difficult to distinguish between traceroute commands of legitimate users and adversaries.

In [6] Christos et. al. proposed a reactive traffic engineering method based on relational algebra principle to mitigate link flooding attacks, their technique is based on the network property of defending against flooding attacks i.e. when the flooding attacks occur the defender reroutes the traffic and after multiple such interactions between attacker and defender, it knows the sources that are consistently participating in flooding events, after the rerouting is performed. The sources that change their destination selection to adapt to re-routing are particularly suspicious. In [7] Dimitrios et. al. proposed reactive traffic engineering based method to mitigate link flooding attacks. Rerouting is performed when the defender realizes that there is an attack, the attacker recalculates the network path and identifies the critical links. Their work is based on destination based routing and the variable path which is effective against link flooding attacks. The limitation of this technique is that the detection speed is dependent on the routing rules modification that can cause legitimate traffic delays.

In [14] Aapo et. al. have proposed mechanism that combines normal traffic learning, external blacklist information, and elastic capacity invocation in order to provide effective load control, filtering and service elasticity during an attack. The black list comes from any Intrusion detection system or any previous knowledge repository. They have implemented their scheme in SDN network testbed. In [13] Fida et. al. proposed a technique called Agile Virtualized Infrastructure. This technique employs Virtualize Networks to dynamically reallocate network resources using VN placement and offers constant VN migration to new resources.

2.3 SDN based approaches

Wang et. al. [3] have proposed a technique called Woodpecker that makes use of incremental SDN deployment to mitigate link flooding attacks. Their technique is based on upgrading routers to SDN switches, which increase the network connectivity. They also use network probing approach to locating the congested links. At the end, Woodpecker makes use of cartelized traffic engineering to balance the traffic across the network and eliminate the bottlenecks that are caused by the adversary during the attack.

Previous techniques do extra header statistics, which increase cost but [8] Peng et. al. have used built-in SDN functionality of flow table inspection. It is based on bloom filters, and works in collaboration with a collector and detector module. When the utilization ratio of a link is not normal the flow tables are scanned and abnormal flows are extracted by the parameters of statistical features. The Collector system scans flow tables from the SDN network and collects traffic flows by IP header inspection. The Detector module extracts IP features from every packet that are important to link attack detection by using Bloom filter. In [9] Abdullah et. al. proposed an SDN based maneuvering technique to defend against link flooding attacks. During the link map construction phase, the links are obfuscated so it will be difficult for the attacker to launch the attack. The links are continuously changed so, it is difficult to always form the optimal path between links, so packets traveling time from source to the destination is increased.

2.4 Link observation based techniques

Qian et. al. [5] proposed active link obfuscation method, their technique is based on providing fake link map to the adversaries and prohibiting the adversary to accurately analyze the network and creating the network map of the underlying network to be attacked. The link map construction phase is one of the most important phases in link flooding attacks, so if an adversary is forced to construct a fake link map, then it will be very difficult for the adversary to locate the targets servers and the maintain the attack. They have used SDN testbed to perform the experimentation. Authors have exploited support vector machines (SVM) to distinguish legitimate users from adversaries, the unique flow features of the adversary are extracted from link map construction as well and link flooding phase and SVM is applied to differentiate legitimate users and adversaries. The limitation of this technique is that SVM is dependent on the training data if the volume of training data is high than its accuracy will also be high.

In [10] Lei et. al. proposed a technique called LinkScope is proposed, in this technique a system that employs both end to end and hop by hop network measurement mechanism to capture abnormal path performance degradation for detecting link flooding attacks. LinkScope learns the path metrics of normal traffic, so link flooding attacks can be differentiated from network failures. The other advantage of using this technique is that LinkScope can be deployed on one end of the path to perform the measurement instead of installing it on both sides of the link. In this attack, links are carefully chosen, the links with high flow density are selected and bots are used to send low rate traffic to these servers to congest these links. In [11] Soo et. al. proposed a mechanism called collaborative defense (CoDef), the links that are not attacked by the adversaries during the link flooding attacks, collaborate and legitimate traffic is rerouted to these links for successful network operation. An autonomous switch AS sends reroute the request to all ASes in the network to create a bypass path around the target area. A technique called SPIFFY is implemented in [12] by Min et. al. which relies on the principle of temporary bandwidth expansion and rate change measurement to detect adversaries from legitimate traffic. In their technique, the bandwidth of the network is increased for a specific time. And a measurement is performed, before and after the bandwidth expansion mechanism. The legitimate traffic expands the bandwidth when there is available bandwidth to be used, but the adversaries will not be able to increase the bandwidth so, they can be easily detected.

3. Critical analysis of LFA techniques

The general area of link flooding attacks has also been explored by many researchers and a lot of work has been done in this field (for example, [3, 4, 5, 6, 7]). Wang et. al. [3] has proposed centralized traffic engineering for limiting the flooding attacks effect. The limitation of their technique is that the attack prevention is reactive, it detects the attacks after it has occurred and reactive measures are taken after, link flooding has already done some damage. In [5] Wang et. al proposed active link obfuscation method to mitigate link flooding attacks, the link flooding attacks can be mitigated by providing fake link map to the adversaries and misfiring the target links. In this technique support vector machines (SVM) are used for classification of the adversaries, but SVM classifier is more accurate when there is a large amount of training data. If the training data is short than the classification process will not be accurate enough.

In [12] Kang et. al. has proposed a method for detection and prevention of link flooding attacks, their mechanism is based on temporary bandwidth expansion, in this mechanism bandwidth of the network is temporarily increased and in response to bandwidth expansion the legitimate users will also increase their bandwidth, but bots will be unable to increase their bandwidth because of consumption of bots. In the detection phase, these bots will be detected because of not increasing their bandwidth during bandwidth expansion phase. The limitation of this technique is that if the legitimate users are also not able to increase their bandwidth during bandwidth expansion phase than there will be confusion in differentiating legitimate users and attackers.

Many research works (for example, [3, 8, 9]) used SDN testbed to perform experiments and to mitigate link flooding attacks by using SDN techniques. However, the work on mitigating link flooding attacks on SDN control plane to data plane has not been explored as yet. The channel connecting control plane to data plane is very critical and if this link is flooded, the whole SDN network can malfunction. Therefore, this work aims to solve the issue of link flooding on control plane to data plane attacks. Reviewing literature reveals that it would be the first effort in this direction. The proposed work will result in techniques to secure large scale SDN based infrastructures from link flooding attacks and will enable ceaseless traffic for legitimate network flows. Table 2 gives a brief overview of the techniques for detection and mitigation link flooding attacks.

Table 2. Comparison of LFA mitigation techniques

Solution Name	Main Idea	Limitation
Incremental SDN Deployment [3]	• Hybrid SDN LFA detection using centralized traffic engineering based on SDN upgraded nodes	• It detects link flooding after the attack occurs.
Traceroute Packets Flow [4]	• Proactively detecting LFA using traceroute commands • Number of traceroute packets increase in regions when there occurs a link flooding attack.	• It is difficult to classify traceroute commands from legitimate users and attackers.
Active Link Obfuscation Method [5]	• Proactive solution, Linkbait which actively mitigates LFA by providing a fake link map to adversaries	• Depend on training data, accurately classify when training data is large
Framework for Mitigating LFA[6]	• Reactive traffic engineering solution, attacker defender interaction, • The sources that adapt to re-routing are classified as suspicious. • Bots are forced to adopt a suspicious behavior to remain effective, revealing their presence.	• Multiple attackers and defender interactions are required to reveal the identity of the attacker. • So it requires initial time for identifying the attackers.
Interplay of LFA and Traffic Engineering [7]	• Defender module perform rerouting it sniffs an attack, • the attacker update the link map and calculate critical links again • Works with traffic engineering features in a reactive manner	• Detection speed is dependent on the routing rules modification that can cause legitimate traffic delays
Bloom Filter in SDN [8]	• The reactive technique, Bloom filter that has collector and detector module. • Flow tables scanned for abnormal utilization ration. • No extra packet header statistics needs to be done	• Controlling false positive rate is a problem.
SDN approach for Moving Target Defense Attacks [9]	• Both proactive and reactive solution of LFA, Obfuscating the links at attack link map creation phase • By using SDN-based maneuvering techniques	• Delay in arriving packets from source to destination, routes are changed, new these paths may not be the optimal paths
LinkScope[10]	• Reactive solution based on learning path metrics of normal and detecting abnormal traffic • LinkScope can be installed on one end of the path for inspection	• Controlling false positive rate is a problem.
SPIFFY [12]	• Reactive solution, temporarily increases bandwidth, legitimate user's increase their bandwidth according to the expansion, bots will be unable to increase because of consumption of bots and can be detected.	• Legitimate users are unable to increase traffic flow in temporary bandwidth expansion phase. High false +ive.

4. Methodology and Conceptual Framework

In this research, a controller will be implemented that will reside independently of control plane or data plane. This controller will comprise of link listener module, flood detection module, and flood mitigation module. An algorithm in the link listener module will constantly observe the link from control plane to the data plane. The listener will alert the link detection module if it senses any congestion on the link which will analyze the link congestion and make a decision on whether it is normal traffic congestion by legitimate users or any flooding attack by adversaries. There will be a mechanism for detecting the link flooding attacks. After the realization of the attack, the attack mitigation module will mitigate the attack while not interrupting the normal traffic. At the end, it will facilitate the traffic to pass normally while constantly checking the link for further attacks. The following figure 2 gives an overview of the proposed framework.

4.1 Link Listener Module

Link listener will be directly connected with the link that connects control plane to the data plane. It will always be checking the link and will sense link congestion. If it finds any congestion it will invoke Flood Detection Module.

4.2 Flood Detection Module

Flood detection module will be invoked by the link listener. If the link listener finds any congestion on the link it will inform the flood detector module. The flood detector module will have two fold operation. First, it will check the type of congestion on the link, if the congestion is normal and due to normal traffic flooding than the link will be allowed to perform its normal flow of operation. If the congestion is caused by an attack, it will have to be mitigated. At this point, flood detection module will invoke flood mitigation module.

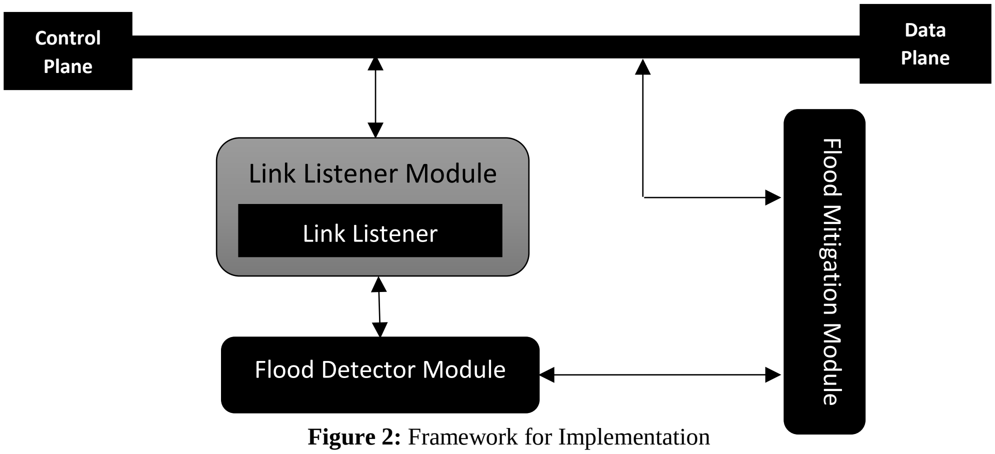

Figure 2: Framework for Implementation

4.3 Flood Mitigation Module

The flood mitigation module will mitigate the flooding of the link that will be causing the link to block the normal flow of operation by using flood mitigation technique and allow the network to carry out normal flow of operation. The three modules will be constantly interacting with each other to perform the flood mitigation operation. Figure 02 shows the flow chart of the proposed framework. Here all the components are shown and the components that invoke each other are also shown diagrammatically.

4.4 Design considerations for the framework

Following design, considerations are important for the implementation of proposed framework

- In mitigating LFA on control plane to data plane, a controller will be developed which will reside independently and will keep a check on the link.
- The independence of the proposed controller will pose less overhead in modifying the complex functionality of default SDN controllers.
- To minimize chances of a controller failure, the framework will be designed in a way that the controller will not interact with outside world, so there will be fewer chances of its failure.
- Traffic consistency will be random, so the controller will be scalable according to the incoming traffic.
- The controller will be able to gather network statistics at random time intervals.

5. Conclusion

This research presents a thorough literature analysis of link flooding attacks in SDNs. After comparative analysis, it identifies Crossfire link flooding technique as one of the lethal attacks that can potentially target the link connecting the control plane to the data plane in SDNs. In such a situation, the control plane may get disconnected, resulting in the degradation of the performance of the whole network and service disruption. This paper aims to establish a framework for mitigating flooding in the link that connects control plane to the data plane in SDN. The proposed framework comprises of three components, link listener module, flood detection module and link flood mitigation module. An algorithm is being designed to be used by the listener module, which will alert the flood detection module which will in-turn invigorate flood mitigation module to mitigate this attack and facilitate the normal flow of traffic. A Mininet testbed has been setup which uses Floodlight controller to mimic an SDN. Initial results are encouraging towards developing the first proof of concept. To the best of our knowledge, the presented problem and the proposed solution is unique and has not been discussed in the literature as yet.

6. References

1. ONF, “OpenFlow Switch Specification 1.5.0,” Open Networking Foundation, 2013.
2. DDoS attack using Mirai botnet https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.
3. Lei Wang, Qing Li, Yong Jiang, Jianping Wu. Towards Mitigating Link Flooding Attack Via Incremental SDN Deployment, 2016 IEEE Symposium on Computers and Communication (ISCC)
4. Takayuki Hirayama, Kentaroh Toyoda, and Iwao Sasase. Fast Target Link Flooding Attack Detection Scheme by Analyzing Traceroute Packets Flow. 2015 1EEE International Workshop on Information Forensics and Security (WIFS)
5. Qian Wang, Feng Xiao, Man Zhou, Zhibo Wang, and Hongyu Ding. Targets can be baits Mitigating Link Flooding Attacks With Active Link Obfuscation in arXiv:1703.09521v1 [cs.NI] 28 Mar 2017
6. Christos Liaskos, et al. A Novel Framework for Modeling and Mitigating Distributed Link Flooding Attacks. IEEE International Conference on Computer Communications. San Francisco, CA, USA, 2016
7. Dimitrios Gkounis, et al. On the Interplay of Link-Flooding Attacks and Traffic Engineering. ACM SIGCOMM Computer Communication, Volume 46 Issue 2, 2016 ACM New York, NY, USA
8. Peng Xiao,et al. An Efficient DDOS Detection with Bloom Filter in SDN. IEEE TrustCom / BigDataSE / ISPA, 2016
9. Abdullah Aydeger, et al. Mitigating Crossfire Attacks using SDN-based Moving Target Defense. IEEE 41st Conference on Local Computer Networks, 2016
10. Lei Xue, Xiapu Luo, Edmond W. W. Chan and Xian Zhan. Towards Detecting Target Link Flooding Attack. The 28th Large Installation System Administration Conference, 2014
11. Soo Bum Lee, Min Suk Kang, Virgil D. Gligor. CoDef Collaborative Defense Against Large-Scale Link Flooding Attacks. ACM CoNEXT’13, California, USA, 2013.
12. Min Suk Kang, Virgil D. Gligor, Vyas Sekar. SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks. NDSS ’16, 2016, San Diego, CA, USA
13. Fada Gillani, et al. Agile Virtualized Infrastructure to proactively Defend Against Cyber Attacks, IEEE Conference on Computer Communications (INFOCOM), 2015
14. Aapo Kalliola, et al. Flooding DDOS Mitigation and Traffic Management with Software Defined Networks. IEEE 4th International Conference on Cloud Networking, 2015.
15. OpenFlow whitepaper. https://www.opennetworking.org/sdn-resources/sdn-library/whitepapers.
16. Min Suk Kang, et al. The Crossfire Attacks, 2013 IEEE Symposium on Security and Privacy.
17. Ahren Studer, Adrian Perrig. The Coremelt Attack, ESORICS 2009
18. BRIGHT. Can a DDoS break the Internet? Sure... just not all of it. Ars Technicahttp://arstechnica.com/security/2013/ 04/can-a-ddos-break-the-internet-sure-just-not-all-of-it/, April 2013.
19. “Difference in control vs data plane in SDN”, http://sdntutorials.com/difference-between-control-plane-and-data-plane, June 2017.

20. Hua Wang, et al. A flexible payment scheme and its role-based access control. IEEE Transactions on knowledge and Data Engineering 17 (3), 425-436, 2005.
21. Xiaoxun Sun et al. A family of enhanced (L, α)-diversity models for privacy preserving data publishing, Future Generation Computer Systems 27 (3), 348-356, 2011.
22. Hua Wang, et al. Effective collaboration with information sharing in virtual universities, IEEE Transactions on Knowledge and Data Engineering 21 (6), 840-853, 2009.
23. ME Kabir et al. A conditional purpose-based access control model with dynamic roles, Expert Systems with Applications 38 (3), 1482-1489, 2011.
24. Xiaoxun Sun, et al. Injecting purpose and trust into data anonymization. Computers & Security 30 (5), 332-345, 2011.
25. ME Kabir, et al. Efficient systematic clustering method for k-anonymization, Acta Informatica 48 (1), 51-66, 2011.
26. Xiaoxun Sun, et al. Satisfying privacy requirements before data anonymization. The Computer Journal 55 (4), 422-437, 2012.