High Performance DDoS Attack Detection System Based on Distribution Statistics

Xia Xie, Jinpeng Li, Xiaoyang Hu, Hai Jin, Hanhua Chen, Xiaojing Ma, Hong Huang

High Performance DDoS Attack Detection System Based on Distribution Statistics

Xia Xie, Jinpeng Li, Xiaoyang Hu, Hai Jin, Hanhua Chen, Xiaojing Ma, and Hong Huang

National Engineering Research Center for Big Data Technology and System
Services Computing Technology and System Lab
Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China
shelicy@hust.edu.cn

Abstract. Nowadays, web servers often face the threat of distributed denial of service attacks and their intrusion prevention systems cannot detect those attacks effectively. Many existing intrusion prevention systems detect attacks by the state of per-flow and current processing speed cannot fulfill the requirements of real-time detection due to the high speed traffic. In this paper, we propose a powerful system TreeSketchShield which can improve sketch data structure and detect attacks quickly. First, we discuss a novel structure TreeSketch to obtain statistics of network flow, which utilizes the stepped structure of binary tree to map the distribution and reduces the complexity of the statistic calculation. Second, we present a two-level detection scheme that could make a compromise between the detection speed and detection accuracy. Experimental results show that our method can process more than 100,000 records per second. The false alarm rate can achieve 2% to 25% performance improvement.

Keywords: DDoS attack · Intrusion Prevention System · Sketch Data Structure · Real-Time

1 Introduction

People have enjoyed numerous high-quality services when the Internet technologies develop rapidly. Indeed, *Distributed Denial of Service* (DDoS) attacks have been mentioned more and more frequently since it adds huge burden to Internet services. In a DDoS attack, legitimate users' access to information or network resource are discarded, because the server cannot afford such numerous requests generated by a huge amount of compromised computers. Theses compromised computers can be controlled by attackers and ordered to perform some malicious tasks unintentionally. With the advancement of modern technology and the complexity of networking environment, this tendency is becoming more serious than before.

According to the Kaspersky lab research, more than 1/3 organizations in the world have suffered DDoS attacks in 2017, compared to just 17% in 2016. To detect DDoS attacks, many intrusion detection and prevention systems have been proposed. However, these systems usually make a compromise between scalability and accuracy. For example, fine grain traffic monitoring can increase the detection accuracy, but cannot scale well. There are many other schemes to detect anomalies based on the statistics of the traffic state, such as entropy-computing [1], deep learning [2]. These methods have high accuracy in detection but the calculation is too heavy to be applied. Since the Internet traffics increase rapidly every year, it is foreseeable that monitoring numerous network traffics in real-time is becoming more and more difficult in anomaly detection. Though dimensionality reduction may be an effective method to process such huge amount of data, it requires complicated computation, and thus can not be applied on a large scale in real-time detection.

Recently, a sort of methods based on sketch have been raised to deal with anomaly detection. Sketch is an effective data structure, which is used to store a summary of a large data set for space efficiency. However, these methods are either very computationally intensive or requiring a large amount of storage capacity, which limits their application in intrusion prevention systems.

In this paper, we address a new system *TreeSketchShield*, which can process the statistics of network traffic and can support a two-level detection to defend against DDoS attacks. The remaining parts are listed as follows: in section 2, we introduce the related works about how to prevent DDoS attacks; in section 3, a brief overview of the detection is discussed; in section 4, we introduce the architecture and implementation of our system. We evaluate the whole system in section 5 and draw a conclusion in section 6.

2 Related Work

The purpose of DDoS attacks is to make the Internet servers or network resource unavailables to its normal users. This is usually implemented by masquerading the normal flash crowd requests. Flash crowd means special situations many different users access a website at the same time, causing sudden huge access pressure on the website or database that could make the website inaccessible [3]. Based on these characteristics, most signature-based intrusion detection and prevention systems are difficult to effectively identify DDoS attacks. To distinguish the DDoS traffic and the normal flash crowds, researchers proposed a series of schemes. Xie et al. [4] observed that these attackers launched application-level DDoS attacks in the flow pattern similar to normal traffics. By using an access matrix, they could identify the spatial-temporal characteristics about the normal traffics. Besides that, a hidden semi-markov model was used to present the dynamics access pattern for detecting these DDoS attacks. Chonka et al. [5] proposed a model based on chaos theory to distinguish a normal traffic flow from the attack traffic flow. A novel system based on neural network was raised to detect anomalous traffic. In order to detect abnormal traffics, Rahmani et al.

[6] utilized joint entropy to record the characteristics of the traffic flows. They used connection coherence to identify the links of packets and the quantity of connections for both normal and abnormal traffic flow. Their result showed that the aggregated traffic with a DDoS attack was nearly doubled compared with normal traffics.

However, these schemes only considered about how to find the DDoS traffic in a flash crowds and ignored the issue that the detection speed could not meet the requirement of the increasing network traffic sometime. When a DDoS attack occurs suddenly and violently, in order to minimize the loss and mitigate the impact on normal users' access, a security system is in need to quickly discover and defend against those attacks. The sketch data structure can reduce the dimension of multidimensional data streams. With efficiently estimating the initial signal [7, 8], high-speed network links are very effective in detecting DDoS attacks with the sketch data structure when people deal with huge network traffic, especially under flooding attacks. Currently, there are many methods for detecting huge network traffic anomalies based on sketch [9–11].

Since sketches do not preserve the detailed information about the malicious hosts, we cannot use them to mitigate DDoS attacks. To solve this problem, Schweller et al. [12] proposed a reverse hashing scheme, which could be used to identify the keys of malicious flows from reversible sketches. Liu et al. [13] proposed an online DDoS detection scheme which adopted the sketch structure to cope with problems raised by DDoS attacks and used the distinction of IP addresses to pinpoint victims. Wang et al. [14] proposed an efficient system SkyShield to combine sketch data structure and special distance to detect DDoS attacks. Moreover, abnormal sketches are used to help identify malicious hosts with a DDoS attack.

These methods are based on the sketch data structure to make statistics, and they often use the distribution distance of the traffic attribution to detect the DDoS attacks. Even though these sketch-based schemes can well mitigate DDoS attacks, the high frequency of statistics cause not only high calculation cost but also the upping false alarm rate. Another problem needed to be solved is to meet the requirement of reducing computational complexity and decrease the time interval of detection.

Because the internet traffic flow grows rapidly, it is difficult to meet the challenge that DDoS detection technologies should handle requests as much as possible within affordable response time. Moreover, to achieve real-time detection, the time of calculating statistics needs to be set very short and this could result in the increasing of false alarm rate due to the insufficient statistic of network traffic.

3 Background

In this section, we give a brief overview of the detection based on the sketch data structure.

The process of the detection based on the sketch data structure is depicted in Fig. 1. First, the intput streaming passes through the filter layer. In the filter layer, a bloom filter is used to filter incoming requests. Malicious requests identified by the blacklist will be filtered and recorded. This can be explained by the fact that a large number of requests is needed to initiate a valid DDoS attack. Therefore, sketch-based detection can identify malicious hosts using the volume of the malicious requests. Second, the detection employs the divergence between $Sketch1$ and $Sketch2$ to signal an abnormal situation that are raised by a large number of requests from malicious hosts. In this step, sketch-based detection conducts the detection cyclically with a fixed time interval ΔT. During a detection cycle, any access received by this system will be aggregated into $Sketch1$, and their IP addresses are set as the input keys. Another $Sketch2$ is used to store the results of $Sketch1$ in the last normal mode. Finally, at the end of each detection cycle, the divergence $d(Sketch1, Sketch2)$ is calculated and compared to a threshold θ_t. If $d(Sketch1, Sketch2)$ exceeds θ_t, the system is considered to be suffering a DDoS attack and an alarm needs to be raised.

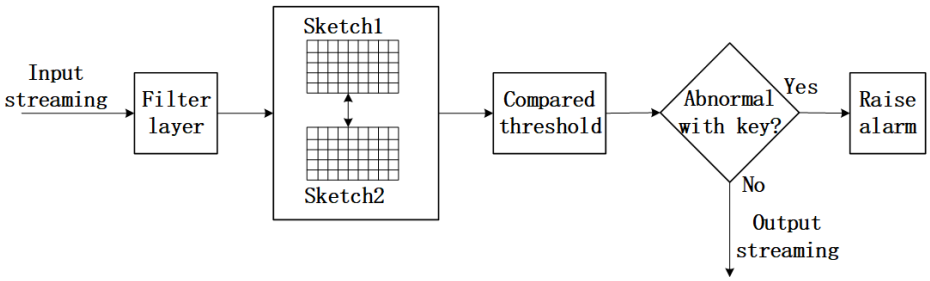

Fig. 1. Overview of sketch-based detection

Sketch-based schemes are based on the statistic of network traffic to detect DDoS attacks. However, the statistical time interval of each detection cycle cannot be set too short. On the one hand, the statistic of distribution will become incomplete due to the insufficiency of the time length, which leads high false alarm rate. The high frequency of statistics generate high calculation costs and cause long time delay. For above reasons, the speed of sketch-based detection cannot meet the requirement of real-time monitoring.

Besides sketch-based detection, other methods are also discussed. For example, the bloom filter can be used to add the requests from malicious hosts into the blacklist, which is implemented by its special data structure. Sketch is another data structure that can efficiently compute raw signals by reducing high dimensional data streams to low dimensions. We can deal with the statistics of traffic and then detect DDoS attacks by it. When DDoS attack occurs, the distribution of bucket values in a sketch is unstable compared with the normal situation due to the steadiness of the normal network traffic. Therefore, we can use the divergence between $Sketch1$ and $Sketch2$ to detect DDoS attacks efficiently.

4 System Design and Implementation

In this section, we introduce the structure of *TreeSketchShield* and describe the difference with sketch-based detection, then we explain how to implement it.

4.1 Process of TreeSketchShield

TreeSketchShield is depicted in Fig. 2. Similar to the sketch-based detection, the intput streaming first goes through the filter layer. But in the detection phase, the detection process is divided into the coarse grain detection and fine grain detection. The first one is used to quickly detect whether there is an abnormal traffic, while the last one determines whether the abnormal traffic is a real DDoS attack traffic. The principle of this scheme is that short statistical interval in coarse grain detection can reduce the computation time and the long statistical interval of fine grain detection can decrease the rate of false alarm.

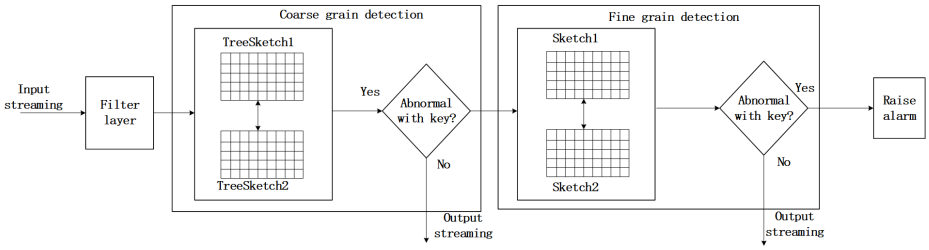

Fig. 2. Process of TreeSketchShield

In the step of coarse grain detection, a novel structure *TreeSketch* is employed, and the statistical time interval of detection cycle is set very short. All requests received and their corresponding IP addresses are saved as key-value pairs, which will be aggregated into $TreeSketch1$. Then $TreeSketch2$ is ready to store the results generated in $TreeSketch1$ during the last normal mode. As soon as the detection cycle is end, the divergence $d(TreeSketch1, TreeSketch2)$ will be calculated and compared to a threshold θ_t. If $d(TreeSketch1, TreeSketch2)$ exceeds θ_t, the fine grain detection will start. In the step of fine grain detection, the time interval of detection cycle is set longer and the fine grain detection uses the sketch data structure instead of the *TreeSketch* to deal with the statistic. If the distance exceeds the value of threshold, the alarm will be raised. After the detection, *TreeSketchShield* uses the volume of all buckets in fine grain detection to identify malicious hosts. Compared to the sketch-based detection, a special two-level detection and a novel structure are adopted to decrease the false alarm rate and improve the detection speed in $TreeSketchShield$.

4.2 TreeSketch

The *TreeSketch* data structure is shown in Fig. 3. *TreeSketch* is a special data structure with H rows of size K, while the source data stream is composed with key-value pairs. For every row in the *TreeSketch*, there is a binary tree and their leaf nodes are associated with different hash functions. When a key-value pair comes, the value will be added into the leaf nodes corresponding to the key. Besides, values in child nodes will be added up and stored in their parent nodes.

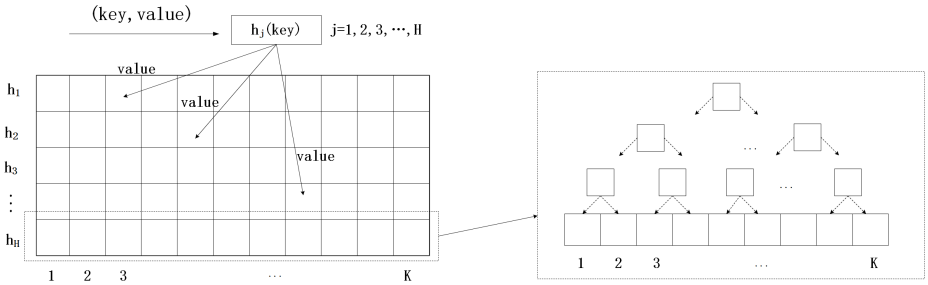

Fig. 3. TreeSketch data structure

In the sketch data structure, the attribute change is mapped to the sketch which represents the compact summaries of a data stream. The attribute is signed by each row of the sketch. It will result in a high computational cost. To avoid this problem, we use the trapezoidal data structure to map the attribute which is based on the *TreeSketch*. Different from the divergence calculation between sketches, when computing the divergence of TreeSketches, we do not need to calculate each button in every row of the *TreeSketch* in order. Denoting the attribution of *i-th* row in the *TreeSketch* as a vector $\langle n_{i1}, n_{i2}, ..., n_{iK} \rangle$, each row of *TreeSketch* is a binary tree and n_{ij} presents the first node in *j-th* layer of the binary tree. Let $N_i = \sum_{j=1}^{K} n_{ij}$ represents all the requests received. Denote $P_i = \langle p_{i1}, p_{i2}, ..., p_{iK} \rangle$ for the corresponding row, where $p_{ij} = n_{ij}/N_i$ means the probability that an incoming request is mapped into the *j-th* bucket of the *i-th* vector. Denote $Q_i = \langle q_{i1}, q_{i2}, ..., q_{iK} \rangle$ are the probability similar to P_i, then the distance $d(P_i, Q_i)$ can be calculated. The metric is that when the server cannot afford these numerous requests, the clients will receive little responses from the server, which could bring about the traffic attribute change. As shown in Fig. 3, each layer of an row in *TreeSketch* contains the next estimation of distribution. If there is any abnormity in the leaf layer, it will cause the upper node to be abnormal and the abnormality will continuously present to the root node.

4.3 Cycle Synchronization of Detection

In the detection of *TreeSketchShield*, the difference of statistical time between coarse grain detection and fine grain detection leads to the out-of-synchronization

problem at both detection points. As shown in Fig. 4, when a cycle of coarse grain detection is finished and an alarm is raised, a cycle of fine grain detection is still going on. To solve this problem, we employ a sliding time window scheme to keep the pace between two detections. Fig. 5 shows that in the fine grain detection, the interval time of each cycle is unchanged but the sliding distance of each cycle is set to a short time. This scheme makes the detection combine the sufficient statistic and the speedy computation, which decreases the false alarm rate and reaches the real-time detection at the same time.

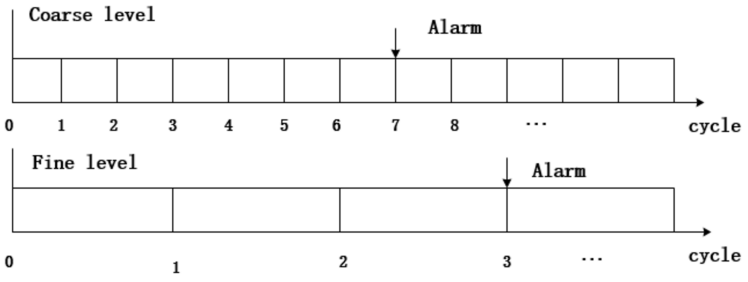

Fig. 4. Cycle of the two-level detection with different interval time

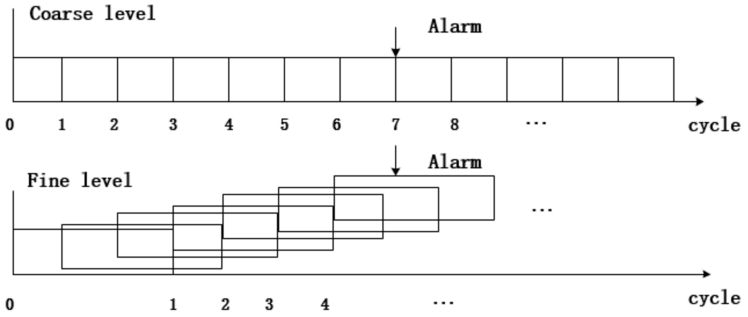

Fig. 5. Cycle of the two-level detection with sliding time window

5 Evaluation

In this section, we design a comparative experiment to evaluate the performance of *TreeSketchShield* about the detection speed and detection accuracy between $TreeSketchShield$ and sketch-based detection. The experiments are tested on a local machine equipped with 2.5GHz I5-7300HQ CPU, 24GB RAM.

Table 1. Summary of datasets

Dataset	Requests	Hosts	Attacks
Dataset0611	4,315,622	5,400	40
Dataset0630	5,332,901	5,880	40
Dataset0710	5,834,324	5,563	45

5.1 Datasets

At first, we introduce these datasets used in this experiment. These datasets are generated from WordCup98 [15], which records all requests from April 30, 1998 to July 26, 1998. The website received nearly 1.3 billion requests during this time. Each dataset in this experiment is composed of access logs of two days in the WordCup98 dataset. To assess the performance of the *TreeSketchShield*, the data from 1998/06/11 to 1998/06/12 is denoted as Dataset0611, the data from 1998/06/30 to 1998/07/01 as Dataset0630, and the data from 1998/07/10 to 1998/07/11 as Dataset0710. Table 1 presents a brief summary of the datasets.

5.2 Performance

We employ the *False Rejection Rate* (FRR) and the *False Acceptance Rate* (FAR) to evaluate the performance of DDoS attacks detection. FAR measures the proportion of normal requests that are mistakenly identified as DDoS attacks in all requests, and FRR is the probability of wrongly identifying normal requests as DDoS attacks in all attacks. We make a comparison between the performance of sketch-based detection and *TreeSketchShield* on three datasets, with the parameters set as: $\alpha = 0.3, \beta = 0.4, \lambda = 3, k = 16384, H = 8$.

Fig. 6 shows the FRR and the FAR on the three datasets with the sliding time window set as 5s. In Dataset0611, *TreeSketchShield* is equal to SkyShield in terms of FRR, which is 5%, but the FAR has decreased from 40% to 38%. In Dataset0630, *TreeSketchShield*'s FRR is 7.5%, the same as SkyShield's, and the FAR is much lower than that of SkyShield, from 44% to 27%. In Dataset0710, the FRR of both methods is 6.7%, but the *TreeSketchShield*'s FAR dropped from 46% to 33%. This is because the *TreeSketch* data structure decreases the sensitivity of attribute detection. The result shows that *TreeSketchShield* is better than sketch-based detection in the accuracy of DDoS attack detection.

Fig. 7 and Fig. 8 shows the results when the sliding time window is set as 10s and 15s respectively. According to the experimental results of these three datasets, the *TreeSketchShield* system reduces the FAR by $5\% - 20\%$ on the basis of ensuring the FRR.

We also employ the number of requests per second to evaluate the throughput of detection. Fig. 9 shows the performance of sketch-based detection and *TreeSketchShield* on three datasets. We can see the throughput of *TreeSketchShield* is more than 100,000 records per second. Moreover, compared with sketch-based detection, the average throughput of *TreeSketchShield* is 8 times faster in Dataset0611, 10 times faster in Dataset0630, and 10 times faster in Dataset0710

respectively. The result shows that *TreeSketchShield* is better than sketch-based detection in processing speed.

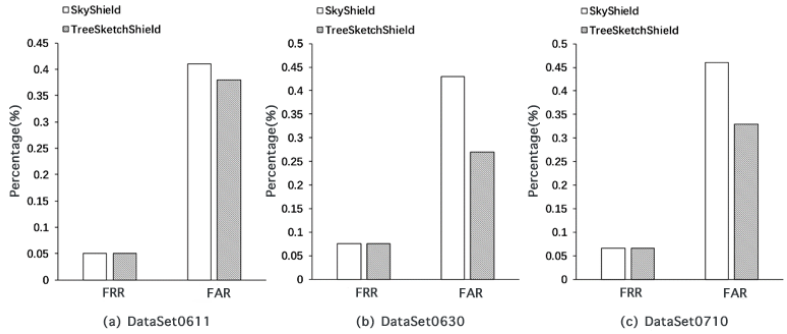

Fig. 6. The accuracy performance of detection while the sliding time window is 5s

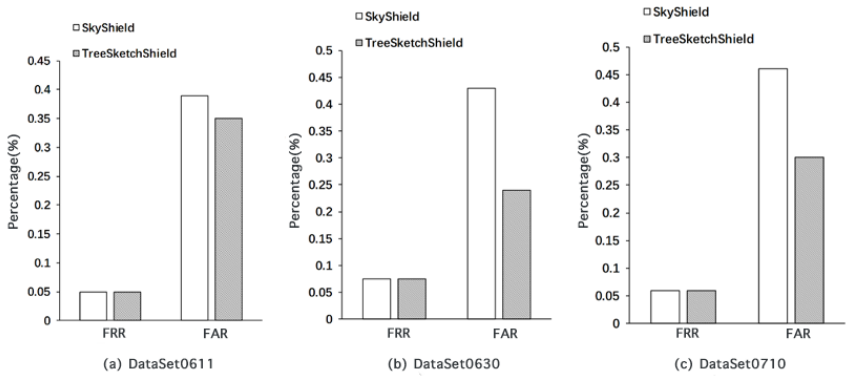

Fig. 7. The accuracy performance of detection while the sliding time window is 10s

6 Conclusion and Future Work

In this paper, we address a new defense system *TreeSketchShield*, which can detect DDoS attacks quickly. First, a novel structrue *TreeSketch* is addressed, which can greatly reduce the complexity of statistical calculations of network flow. Then a two-level detection scheme is designed to decrease the false alarm rate. *TreeSketchShield* still needs to be improved: first, although the FAR is decreased,

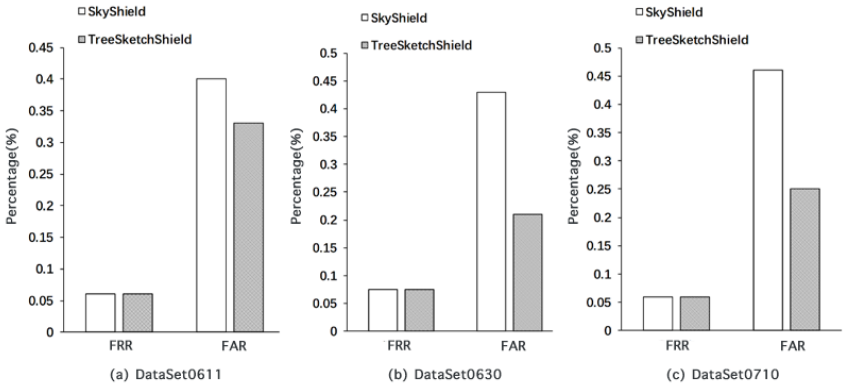

Fig. 8. The accuracy performance of detection while the sliding time window is 15s

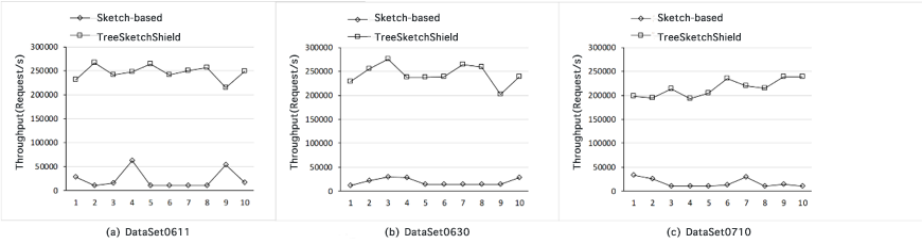

Fig. 9. The throughput performance of detection

it is still unacceptable. The reason is that the statistic of *TreeSketchShield* adopts the partial substitution strategy, which reduces the sensitivity of detection; Second, *TreeSketchShield* increases the memory consumption when obtaining statistics of net flow. In the future, we will continue to improve the structure to eliminate the rate of missing DDoS alarm and decrease the memory consumption of detection.

Acknowledgements

This work is supported in part by the National Key Research and Development Program of China under grant No.2016QY02D0302, the Fundamental Research Funds for the Central Universities (HUST No.3020210111).

References

1. Osanaiye, O., Choo, K.K.R., Dlodlo, M.: Distributed denial of service (ddos) resilience in cloud: Review and conceptual cloud ddos mitigation framework. Journal of Network and Computer Applications **67**, 147–165 (2016)
2. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. IEEE Communications Surveys Tutorials **15**(4), 2046–2069 (2013)
3. Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y., Tang, F.: Discriminating ddos attacks from flash crowds using flow correlation coefficient. IEEE Transactions on Parallel and Distributed Systems **23**(6), 1073–1080 (2012)
4. Xie, Y., Yu, S.: Monitoring the application-layer ddos attacks for popular websites. IEEE/ACM Transactions on Networking **17**(1), 15–25 (2009)
5. Chonka, A., Singh, J., Zhou, W.: Chaos theory based detection against network mimicking ddos attacks. IEEE Communications Letters **13**(9), 717–719 (2009)
6. Rahmani, H., Sahli, N., Kammoun, F.: Joint entropy analysis model for ddos attack detection. In: Proceedings of the 5th International Conference on Information Assurance and Security. pp. 267–271 (2009)
7. Ben, U., Bremler, A., Levy, H.: Vulnerability of network mechanisms to sophisticated ddos attacks. IEEE Transactions on Computers **62**(5), 1031–1043 (2013)
8. Tang, J., Cheng, Y., Hao, Y., Song, W.: Sip flooding attack detection with a multi-dimensional sketch design. IEEE Transactions on Dependable and Secure Computing **11**(6), 582–595 (2014)
9. Liu, Y., Chen, W., Guan, Y.: A fast sketch for aggregate queries over high-speed network traffic. In: Proceedings of the IEEE International Conference on Computer Communications. pp. 2741–2745 (2012)
10. Gangam, S., Sharma, P., Fahmy, S.: Pegasus: precision hunting for icebergs and anomalies in network flows. In: Proceedings of the IEEE International Conference on Computer Communications. pp. 1420–1428 (2013)
11. Wang, P., Guan, X., Zhao, J., Tao, J., Qin, T.: A new sketch method for measuring host connection degree distribution. IEEE Transactions on Information Forensics and Security **9**(6), 948–960 (2014)
12. Schweller, R., Li, Z., Chen, Y., Gao, Y., Gupta, A., Zhang, Y., Dinda, P., Kao, M., Memik, G.: Reverse hashing for high-speed network monitoring: algorithms, evaluation, and applications. In: Proceedings of the IEEE International Conference on Computer Communications. pp. 1–12 (2006)

13. Liu, H., Sun, Y., Kim, M.: Fine-grained ddos detection scheme based on bidirectional count sketch. In: Proceedings of the 20th International Conference on Computer Communications and Networks. pp. 1–6 (2011)
14. Wang, C., Miu, T.N., Luo, X., Wang, J.: SkyShield: a sketch-based defense system against application layer ddos attacks. IEEE Transactions on Information Forensics and Security **13**(3), 559–573 (2018)
15. Worldcup98. Website (2016), http://ita.ee.lbl.gov/html/contrib/WorldCup.html