ISSN 1566-2535
ELSEVIER
An International Journal on
INFORMATION FUSION
Multi-Sensor, Multi-Source, Multi-Process Information Fusion
Editor-in-Chief
Francisco Herrera
Available online at www.sciencedirect.com
ScienceDirect

Network Traffic Fusion and Analysis against DDoS Flooding Attacks with a Novel Reversible Sketch

Xuyang Jing , Zheng Yan , Xueqin Liang , Witold Pedrycz

PII: S1566-2535(18)30581-5
DOI: https://doi.org/10.1016/j.inffus.2018.10.013
Reference: INFFUS 1040

To appear in: *Information Fusion*

Received date: 20 August 2018
Revised date: 5 October 2018
Accepted date: 31 October 2018

Please cite this article as: Xuyang Jing , Zheng Yan , Xueqin Liang , Witold Pedrycz , Network Traffic Fusion and Analysis against DDoS Flooding Attacks with a Novel Reversible Sketch, *Information Fusion* (2018), doi: https://doi.org/10.1016/j.inffus.2018.10.013

Highlights

- Propose a novel sketch to fuse network traffic and recover anomalous sources;
- Design a multi-dimensional change-point method for flooding attack detection;
- Support self-adaptive and protocol independent detection
- Evaluate method efficiency, accuracy and adaptability with real-world datasets.

Network Traffic Fusion and Analysis against DDoS Flooding Attacks with a Novel Reversible Sketch

Xuyang Jing[1], Zheng Yan[1, 2*], Xueqin Liang[1,2],Witold Pedrycz[3]

[1]*State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China*
[2]*Department of Communications and Networking, Aalto University, Finland*
[3]*Department of Electrical & Computer Engineering, University of Alberta, Canada*

Abstract Distributed Denial of Service (DDoS) flooding attacks are one of the typical attacks over the Internet. They aim to prevent normal users from accessing specific network resources. How to detect DDoS flooding attacks arises a significant and timely research topic. However, with the continuous increase of network scale, the continuous growth of network traffic brings great challenges to the detection of DDoS flooding attacks. Incomplete network traffic collection or non-real-time processing of big-volume network traffic will seriously affect the accuracy and efficiency of attack detection. Recently, sketch data structures are widely applied in high-speed networks to compress and fuse network traffic. But sketches suffer from a reversibility problem that it is difficult to reconstruct a set of keys that exhibit abnormal behavior due to the irreversibility of hash functions. In order to address the above challenges, in this paper, we first design a novel Chinese Remainder Theorem based Reversible Sketch (CRT-RS). CRT-RS is not only capable of compressing and fusing big-volume network traffic but also has the ability of reversely discovering the anomalous keys (e.g., the sources of malicious or unwanted traffic). Then, based on traffic records generated by CRT-RS, we propose a Modified Multi-chart Cumulative Sum (MM-CUSUM) algorithm that supports self-adaptive and protocol independent detection to detect DDoS flooding attacks. The performance of the proposed detection method is experimentally examined by two open source datasets. The experimental results show that the method can detect DDoS flooding attacks with efficiency, accuracy, adaptability, and protocol independability. Moreover, by comparing with other attack detection methods using sketch techniques, our method has quantifiable lower computation complexity when recovering the anomalous source addresses, which is the most important merit of the developed method.

Keywords: DDoS flooding attacks; attack detection; Network traffic fusion; Reversible sketch; Chinese Remainder Theorem; Bloom filter; Multi-chart Cumulative Sum; Decision fusion

1. Introduction

Network attacks that are the main threats of security over the Internet have attracted special attention. The openness and interconnection of the network and the security vulnerabilities of protocols and software lead to multiple and multi-level network attacks. Distributed Denial of Service (DDoS) flooding attacks are one of the typical attacks over the Internet [1, 2]. They aim to flood a victim and occupy the victim's resources so that it cannot provide normal services for legitimate users. DDoS flooding attacks can be generated in two ways: direct flooding attacks and indirect flooding attacks. In direct flooding attacks, namely Network/Transport Layer and Application Layer DDoS flooding attacks [3], attackers usually spoof source IP address of attack packets and send them to the victim directly. In the case of indirect flooding attacks, namely Distributed Reflection DoS (DRDoS) [4, 5] and link flooding attacks [6, 7], attackers use many innocent intermediates to flood victim indirectly.

How to detect DDoS flooding attacks is a timely and important topic in the field of network security. DDoS flooding attacks usually flood a victim with massive attack packets. Thus, a dramatic increase in the number of packets is an indicator of DDoS flooding attacks. Moreover, current DDoS flooding attacks are often launched by a botnet. Each bot, which is infected by the same malicious program, generates attack packets of the same format. The attack packets share many similar characteristics, such as packet size, packet rate, destination address, destination port. But these characteristics are quite different from legitimate packets so that they can also be used to detect DDoS flooding attacks. There are three methodologies widely used to analyze traffic data to detect DDoS flooding attacks [1]. The first is statistical methods where observe network activities and generate a profile to represent normal network behaviors. By measuring the similarity between the profile that was extracted from current collected network traffic and the normal behavior profile, it is feasible to judge whether there are existing anomalies in the network. The second methodology concerns machine learning [8]. There are mainly three types of methods. Supervised learning ususally establishes the judging criterion of network traffic through training a large number of data and uses this criterion to analyze and determine whether the current collected traffic is abnormal. Unsupervised learning is often used to find potential network traffic anomalous patterns [9]. In semi-supervised learning, a portion of labeled data is mixed into a large amount of unlabeled data to generate the training dataset for unsupervised learning. Knowledge-based methods compare network events with pre-defined attack rules or attack patterns to detect network attacks.

Although many efforts have been made to detect DDoS flooding attacks, we are still facing several open issues. We summarize the existing issues in the field of DDoS flooding attack detection as follows: (1) Lack of effective network traffic compression and fusion methods. The increasing network traffic demands that a detection method must have the capability of processing big-volume network traffic, and at the same time ensures high detection accuracy with high efficiency. Otherwise, incomplete network traffic collection or non-real-time processing of big-volume network traffic will seriously affect the accuracy and efficiency of attack detection. Effective compression and fusion of network traffic, completely describing network traffic information in the form of a summary and accurately locating the abnormal traffic, can greatly eliminate the impact that the big-volume network traffic brings to the network attack detection [10]. Some excellent surveys on network traffic collection techniques were presented in [1, 11-14]. However, effective detection methods for DDoS flooding attacks that can compress and fuse big-volume network traffic with high accuracy are still missing in the literature; (2) Lack of self-adaptive and protocol independent detection methods. Most detection methods typically build a normal profile of network behaviors, and detect intrusion based on the deviation between the normal profile and a current network profile. But an attacker can elaborately launch attacks by intentionally training detection methods to make them gradually accept abnormal network behaviors as normal. Facing such an attack, it is hard to define what types of network behaviors are normal. If the patterns of normal network behaviors are wrong or incomplete, the detection methods will exhibit a high false alarm rate. Therefore, the detection methods should adapt to the changes of current network behaviors instead of mainly depending on a static profile. Moreover, there are many kinds of DDoS flooding attacks in the context of various networking protocols. We cannot predict when and which type of attacks will occur. A qualified detection method should be protocol independent for detecting DDoS flooding attacks. Both direct and indirect DDoS flooding attacks have their own attack characteristics and could exploit multiple protocols to launch attacks. So, a protocol-independent method is highly expected to detect a category of DDoS flooding attacks instead of detecting a specific protocol-related attack. As can be seen from the above discussion, a universal network attack detection method is urgently needed, which has the ability of dealing with big-volume network traffic, offers self-adaptive detection, and supports protocol independence.

In this paper, we propose a novel network traffic fusion and analysis method against DDoS flooding attacks that can support self-adaptive detection and protocol independence. First, with the purpose of gaining global network traffic information, we design a new Chinese Remainder Theorem based Reversible Sketch (CRT-RS) to effectively compress and fuse network traffic. Traditional sketch is a compact and constant-size data structure that summarizes network traffic by using hash functions to randomly aggregate traffic [15]. We replace the modulus in hash functions presented in traditional sketch that are selected from the family of *k*-universal hash functions with the modulus in Chinese Remainder Theorem (CRT) and employ the notion of modular coefficient description in number theory to keep the location in each row of sketch of a specific source address. By doing this, our CRT-RS not only retains the same features of the traditional sketch, but also has the ability of reversely discovering anomalous source addresses. Second, we design a multi-dimensional change-point detection method for DDoS flooding attacks by using a Modified Multi-chart Cumulative Sum (MM-CUSUM) algorithm. The MM-CUSUM algorithm is a sequential analysis technique used to detect irregular changes in traffic traces. By employing the CRT-RS and the MM-CUSUM algorithm to perform traffic fusion and attack detection, our method is capable of efficiently handling big-volume network traffic, discovering anomalous source addresses, realizing self-adaptive detection and supporting protocol independence. In summary, the following major contributions in this paper are worth stressing:

- We propose a novel Chinese Remainder Theorem based Reversible Sketch (CRT-RS) that has the abilities of effectively compressing and fusing network traffic and reversely recovering anomalous source addresses. CRT-RS solves the problem of large resource consumption when recovering the source addresses.
- We design a multi-dimensional change-point detection method that supports self-adaptive and protocol independent detection by utilizing the basic characteristics of DDoS flooding attacks and the MM-CUSUM algorithm. The proposed method takes sufficient traffic features into account and fuses detection results for each traffic feature into a final result in order to reduce false alarms.
- We evaluate the performance of the proposed method with two open source datasets in order to show its high advantages in terms of *efficiency, accuracy, adaptability* and *protocol independability*.

The paper is organized as follows. Section 2 gives a brief review on related work. In Section 3, we introduce the preliminaries of our proposed method. In Section 4, we present CRT-RS and the system architecture of attack detection and mitigation in detail. The performance evaluation results of the proposed method are showed in Section 5 followed by further discussions on its effectiveness. Finally, conclusions are covered in the last section.

2. Related Work

Distribution and cooperation are the main characteristics of DDoS flooding attacks. They aim to flood a victim and occupy the victim's resources so that it cannot provide normal services to legitmate users; see Figure 1. Numerous methods are presented to detect DDoS flooding attacks. Bellaiche and Gregoire [16] used the numbers of SYN, ACK, SYN-ACK and RST packets that drastically change in unusual TCP handshakes to detect SYN flooding attacks. Kim et al. [17] proposed a statistics-based malicious packet detection and filtering scheme named PacketScore to counter DDoS flooding attacks. PacketScore utilizes the notion of "Conditional Legitimate Probability" (CLP) based on Bayesian theorem to judge whether a packet is legitimate. Yu et al. [18] applied a similarity metric, called flow correlation coefficient to discriminate attack flows among suspicious flows (such as flow crowds). When a possible attack alarm goes off, the flow correlation coefficient between suspicious flows is calculated. If the correlation coefficient exceeds a certain threshold, this pair of flows is identified as attack. Alenezi et al. [19] utilized Cumulative Sum (CUSUM) algorithm to monitor the changes of congestion window (cwnd) value in packet headers to DDoS flooding attacks. Xiong et al. [20] applied synergetic neural networks and the catastrophe theory to analyze the dynamic characteristics of network traffic and then established two rules to monitor traffic changes to detect attacks.

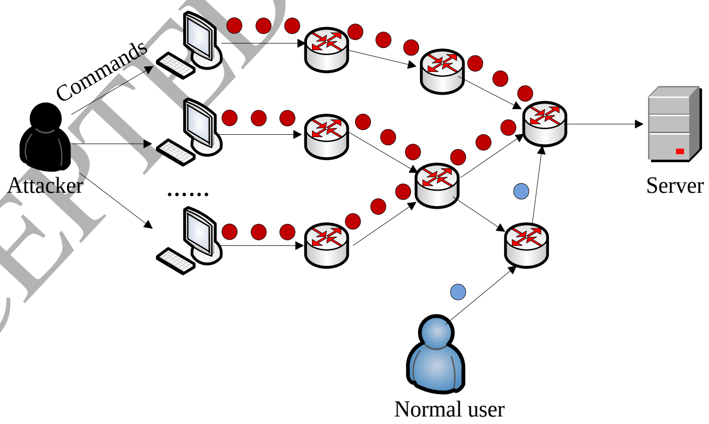


Figure 1. DDoS flooding attacks.

However, with the rapid growth of network traffic, traditional detection methods for DDoS flooding attacks become inefficient due to the lack of capacity of dealing with big-volume network traffic. Sketch data structures are the hope of solving this problem. Sketches are a family of probabilistic data structures that use data-oriented hashing [21] for data summarization. They differ in how they update buckets and estimate aggregated result for a specified query. There are some popular existing sketch data structures, such as *K-ary* Sketch [22], Count Sketch [23], Count-Min (CM) Sketch [24], Augmented Sketch [25] and Pyramid Sketch [26].

Researchers have proposed many methods for detecting DDoS flooding attacks using sketches. Tang et al. [27] developed a versatile detection method for the SIP flooding attacks. They designed a three-dimensional sketch data structure to separately summarize the number of SIP INVITE, SIP 200 OK, SIP ACK and SIP BYE packets. Based on sketch data structure, a probability distribution is established for each SIP attribute independently. A SIP flooding attack can be detected with a high probability by comparing the Hellinger distance among data distributions present in sketches. Salem et al. [28] proposed a sequential approach for DDoS attack detection. They recorded traffic information by using *K-ary* sketch. Based on the traffic record, they employed Least Mean Square Filter to estimate the current value of each bucket from previous values and Pearson Chis-quare divergence to measure the deviations between the current and estimated probability distribution. Callegari et al. [29] designed a method by combining sketch and wavelet analysis to detect attacks at the router level. However, the sketches used in these studies are not reversible. That is to say the sketches cannot report the set of keys that exhibit abnormal behavior.

How to design a reversible sketch and apply it into attack detection becomes an essential topic. Schweller et al. [30] proposed a reversible sketch along with reverse hashing algorithms to infer the abnormal keys. The basic idea is to hash intelligently by modifying the input keys and hashing functions. Li et al. [31] extended the work by designing efficient two-dimensional reversible sketches to distinguish different types of attacks. Salem et al. [32] proposed a detection method for DDoS flooding attacks by integrating multi-stage reversible sketch that proposed in [30]. Wang et al. [33] designed a reversible connection degree sketch for measuring and monitoring host connection degrees. The abnormal hosts that have a large connection degrees or significant changes in connection degrees can be located accurately and efficiently. Wang et al. [34] proposed an effective detection and defense system for application DDOS flooding attacks without the need of reverse calculation or storage of malicious hosts, named SkyShield. SkyShield monitors the divergence of distribution of packet amount recorded by sketch and exploits the detected abnormal sketch to identify malicious hosts directly.

There are numerous detection methods for DDoS flooding attacks using sketch techniques. However, few of existing studies considered the reversible characteristic of sketch and satisfied the detection requirements of self-adaptive detection and protocol independence. In order to address the problems discussed here and in Section 1, we propose a novel network traffic fusion and analysis method against DDoS flooding attacks by employing a reversible sketch based on CRT and a MM-CUSUM algorithm.

3. Preliminaries

In this section, we describe the theoretical preliminaries of the proposed detection method.

3.1. Sketch

A sketch is composed of H hash tables of size M, as shown in Fig. 2. It models network traffic as a stream of *(key, value)* pairs, where the key can be one or more fields in packet headers, and the value can be any accumulative feature, such as the size of a packet, the number of packets, etc. In the sketch, each bucket is represented as $T[i][j]$, i=1, 2, …., H, j=1, 2, …., M. Each row i is associated with an independent hash function h_i that maps the incoming keys into a hashing space of (1, 2, …., M). The hashed outputs are associated with their corresponding columns. For example, when a new pairwise item *(key, value)* arrives, the key will be hashed H times by $\{h_1, h_2, …., h_H\}$ and the value will be added to the corresponding bucket in each column, namely *T[i][hi(key)] += value*, *i=1, 2, …., H*. The purpose of applying H hash functions is to avoid the collisions between different keys. The probability that two keys are hashed in the same value is bounded if the function is selected from a kind of hash family. Usually, H hash functions in a sketch are chosen from the family of k-universal hash functions defined in the form,

$$h(x) = \sum_{i=0}^{k-1} \left(a_i x^i + b_i\right) \, mod \; r \; mod \; M \tag{1}$$

where r is an arbitrary prime, a_i (≠0) and b_i are randomly selected from the set of (0, 1, …., r-1), M is the width of sketch. Using k-universal hash functions, the probability that two keys are located in the same bucket over H hash tables is $(1/M)^{k*H}$.

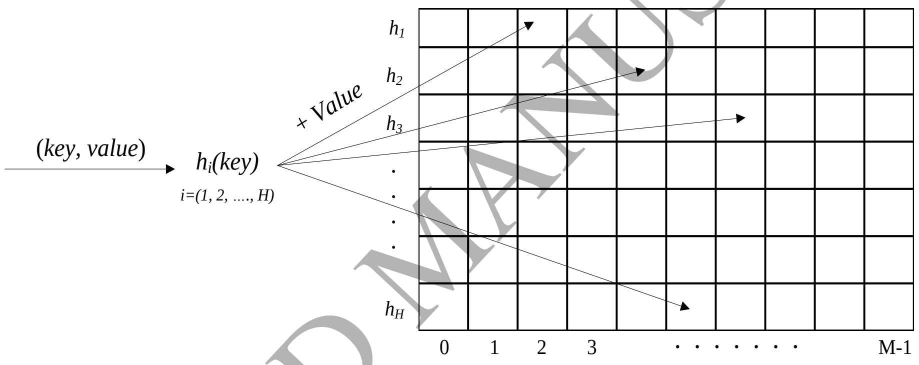


Figure 2. A diagram of the sketch.

3.2. Bloom Filter

Bloom filter is a space-efficient data structure for set membership queries. It is composed of an array of m bits with all initial value are zero and K independent hash functions h_i with a range *(0, 1, 2, …., m-1)*, *i=1, 2, …., K*, as shown in Fig. 3. For each element x in a data set S, the bits at the bucket indicated by $h_i(x)$, i=1, 2, …., K, are set to 1. To query whether another element y belongs to the data set S, the bits at the buckets indicated by $h_i(y)$, *i=1, 2, …., K*, are checked. If all checked bits are equal to 1, the element y belongs to the data set S with high probability. As shown in Fig. 3, y does not belong to S because there exists one bit equal to 0. Although the bloom filter has a little false positive rate caused by conflicts of hash functions, the advantage of fast query and little space consumption still makes it become a popular data structure for set membership queries.

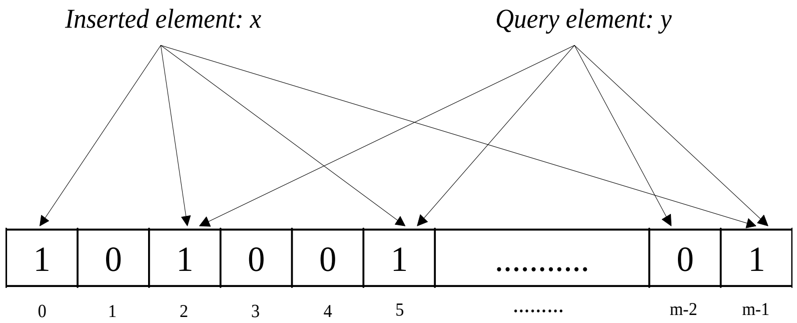


Figure 3. A diagram of the bloom filter.

3.3. Comparison

Sketches and bloom filters differ in their data representations. Sketches are mainly used to store information. A sketch consists of *H* hash tables of size M to summarize big-volume traffic information into a compact and constant-size data set. It models network traffic as a stream of *(key, value)* pairs, where the value is the storage unit determined by the application scenarios. Bloom filters are widely applied in set membership queries. They often do not store any statistical traffic information. The storage unit in a bloom filter is a bit and multiple true-value bits are used together to indicate the existence of an element in the set.

Sketches and bloom filters are similar in the usage of hash functions. Each arriving key will be hashed *H/K* times to determine the corresponding locations. The randomness of hash functions mainly affects the effectiveness and accuracy of sketches and bloom filters. The characteristics of space-efficiency make them popular in many fields.

4. The Proposed Attack Detection and Mitigation Method

In this section, we introduce the new reversible sketch based on Chinese Remainder Theorem. Then, we elaborate on the system architecture of attack detection and mitigation in detail.

4.1. Chinese Remainder Theorem based Reversible Sketch

When using traditional sketch to record network traffic information and carrying out detection based on it, we only get the information about whether there were any attacks. It is impossible to determine which *keys* exhibit abnormal status because the traditional sketch does not store any information about the *keys*. The irreversibility of hash functions makes the traditional sketch impossible to reversely compute the original keys. However, it is essential to accurately recover keys and identify malicious IP addresses for attack mitigation.

In order to maintain the abilities of sketch in dealing with network traffic and make it reversible, we take the advantages of Chinese Remainder Theorem (CRT) that is capable of solving simultaneous congruence to handle the problem of irreversibility of sketch. The proposed Chinese Remainder Theorem based Reversible Sketch (CRT-RS) is shown in Figure 4. The CRT-RS can find abnormal *keys* in an effective way instead of completing an exhaustive test of all possible *keys*. Comparing with traditional sketch discussed above, we replace the modulo of hash function in each row with the modulo in CRT so that we obtain unequal number of buckets in each row of CRT-RS. The purpose of applying hashing operation is to obtain the location of each arriving key in each row. In order to facilitate calculation, we set *r* to a very large arbitrary prime, set *k* to 2, set a to 1 and set b to 0 in (1). The hash functions in CRT-RS that we optimized are rather simple but efficient, as expressed below:

$$h_i(x) \equiv x \; mod \; p_i \quad i = (1, 2, \ldots., H) \qquad (2)$$

where $p_1, p_2, p_3, \ldots, p_H$ are selected as pair-wise coprime integers, $p_1 > p_2 > p_3 \ldots > p_H$. We use this descending order to improve the accuracy of reversely recovering the abnormal source addresses due to the probability that two keys are aggregated in the same bucket in each row is $\left(1/_{p_i}\right)^2$. The larger the p_i is, the lesser probability it is to collide.

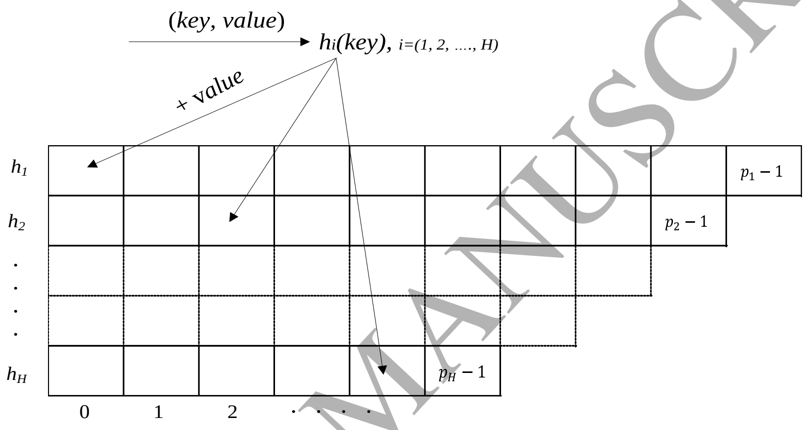


Figure 4. A diagram of the CRT-RS.

In the CRT-RS, each bucket is represented as $RS[i][j], i = 1, 2, \ldots, H,\ j = 0, 1, 2, \ldots, p_i - 1$. That is to say $RS[i][j]$ denotes the value in the bucket. There are four basic operations defined for CRT-RS:

(1) UPDATE: When a new pairwise item *(key, value)* arrives, the key will be hashed *H* times by $\{h_1, h_2, h_3 \ldots, h_H\}$ and the value will be added to the corresponding bucket in each column, namely

$$RS[i][h_i(key)] += value, i = (1, 2, \ldots., H) \qquad (3)$$

Update Time Analysis: for each row in CRT-RS, one needs to complete a single time hash calculation to update a bucket. The total number of hash calculations is H. Each bucket only requires *O(1)* running time. Therefore, the complexity of update time of CRT-RS is *O(H).*

(2) QUERY: The number of columns in each row of CRT-RS is different. To eliminate the affect which the distinct column number brings to the accuracy of query, the query of a specified *key* is defined as follow:

$$QUERY(key) = \frac{\sum_{i=1}^{H}(RS[i][h_i(key)])}{H} \qquad (4)$$

(3) COMBINE: Multiple CRT-RSs with the same sketch parameters can be combined into a single one by accumulating value in each bucket of each CRT-RS. For example, we combine a set of CRT-RSs, $\{RS_1, RS_2, \ldots., RS_l\}$:

$$RS_{combination}[i][j] = \sum_{n=1}^{l} RS_n[i][j] \qquad (5)$$

The COMBINE function of CRT-RS allows us to conduct a distributed collect ion of network traffic and then carry out a centralized analysis.

(4) RECOVER: The RECOVER functions of CRT-RS aims to reversely recover a set of keys that are abnormal. We first consider the simplest scenario. Assume one key exhibits abnormal behavior so that there is only one marked abnormal bucket in each row of CRT-RS, denoted as $c_1, c_2, \ldots, c_H$. Based on (2), we get the following simultaneous congruence:

$$key \equiv c_1 \; mod \; p_1; \; key \equiv c_2 \; mod \; p_2; \ldots.; \; key \equiv c_H \; mod \; p_H \qquad (6)$$

Based on CRT, key can be determined with the use of the following expression:

$$key \equiv \sum_{i=1}^{H} Q_i Q_i' c_i \; mod \; P \qquad (7)$$

where $P = p_1 p_2 \ldots p_H$, $Q_i = {}^{P}/_{p_i}$, and $Q_i Q_i' \equiv 1 \; mod \; p_i, i = (1, 2, \ldots., H)$.

Under the general situation, assume *w (w>1)* keys exhibit abnormality so that there are *w* marked abnormal buckets in each row of CRT-RS. If we use similar recover principle that used in a simplest scenario, we will get w^H different possible combinations of abnormal buckets come from different rows. This leads to heavy computational burden. In order to solve this problem, we employ the notion of modular coefficient description in number theory to record the remainders. For a *key*, its modular coefficient description is $\langle h_1(key), h_2(key), \ldots. h_H(key)\rangle$, $i = 1, 2, \ldots., H$. That is to say, modular coefficient description records the column number that a key locates in each row. We add a *flag* to the bucket $RS[i][h_i(key)]$ to record the modular information (column number) of the row *i+1* obtained from modular coefficient description, namely $flag = h_{i+1}(key)$, $i = 1, 2, \ldots., H-1$. Each bucket is represented as $RS[i][h_i(key)](flag)$. Therefore, when there are many marked abnormal buckets in each row, we match the modular information (column number) recorded by the *flag* from the first row to the last row with precalculated modular coefficient descriptions of all keys that have been hashed into the CRT-RS. After that, we can accurately recover all the abnormal keys using CRT based on the matched modular coefficient descriptions. The calculation process is the same as the calculation process completed in the simplest scenario. The rationality of this design is that if the *key* is malicious, it will exhibit abnormality in each row.

4.2. System Architecture

Figure 5 shows the system architecture of our detection and mitigation method. First, we compress and fuse network traffic with CRT-RS. Then, based on traffic record generated by CRT-RS, we apply multi-dimensional change-point detection method to find out the anomalies buckets in each row. Change-point is a concept encountered in statistics. It represents a point that abruptly changes in a stationary distribution process. By exploiting the reversible characteristic of CRT-RS, we obtain the abnormal keys and add them to a blacklist. A bloom filter is employed to verify whether new arriving traffic arives from the source addresses that existed in the blacklist. A detailed process is introduced below.

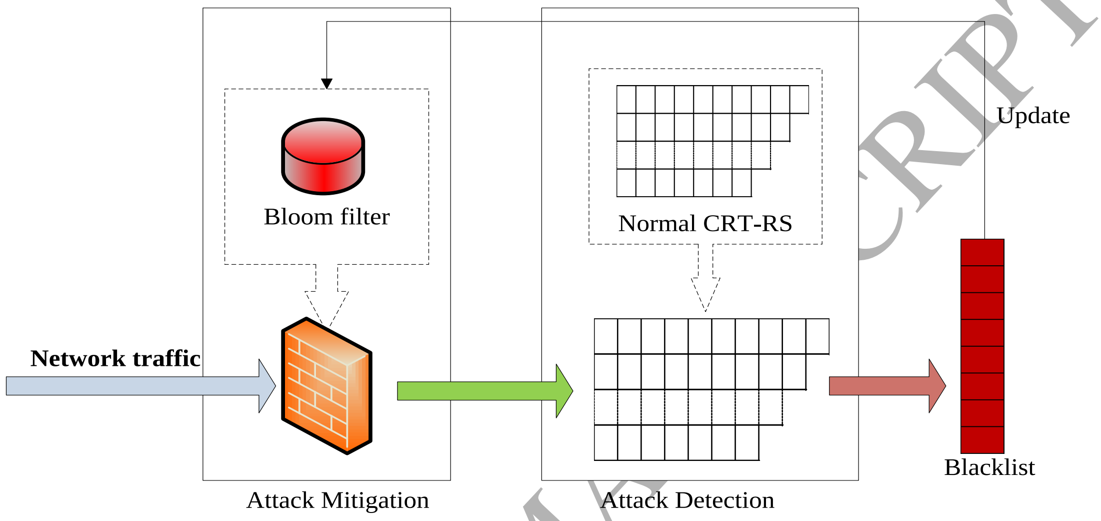


Figure 5. System Architecture.

4.2.1 Network Traffic Collection

The detection method exploits the basic characteristics of DDoS flooding attacks to detect and mitigate them. In CRT-RS, we choose the *key* as source address with the purpose of filtering attack packets come from abnormal source addresses. The ***Value*** = *{value*$_1$*, value*$_2$*, value*$_3$*, value*$_4$*, value*$_5$*}* in each bucket of CRT-RS is a multi-dimensional vector, where each dimension is the statistical information of network traffic, as described below:

(1) *value*$_1$: the number of TCP packets;
(2) *value*$_2$: the number of UDP packets;
(3) *value*$_3$: the number of ICMP packets;
(4) *value*$_4$: the distribution of packet size;
(5) *value*$_5$: the distribution of destination address.

We use three counters to record the traffic information (1)-(3), respectively. For (4) and (5), we first use histograms with the *x*-coordinate is packet size/destination address and the y-coordinate is the number of packets to record the traffic information. Then, we calculate the entropy based on each histogram. We use the above five traffic features to detect DDoS flooding attacks since they reflect the basic characteristics of this kind of attack [1]. Notably, more features of traffic can be involved into the CRT-RS to support attack detection if needed.

4.2.2 DDoS Flooding Attack Detection

The CUSUM algorithm is a commonly used change-point algorithm to detect irregular changes in traffic traces as it requires low computational and memory overhead. Moreover, when DDoS flooding attacks occur, a drastic changes will be monitored in the current traffic compared with the traffic present in the previous time interval. Modified Multi-chart Cumulative Sum (MM-CUSUM) is a variant of the CUSUM algorithm, which is designed to analyze multi-dimensional traffic feature vector. It is used to monitor the abrupt changes in each bucket upon the traffic record generated by CRT-RS. Denote $RS(n)$ is the traffic record collected in n^{th} time interval. $RS[i][j](n)$ is the ***Value*** of bucket in row *i* and column *j*. The result of MM-CUSUM algorithm $M_g[i][j](n)$ is expressed as follows:

$$\begin{aligned} M_g[i][j](n) \\ &= max\{0, M_g[i][j](n-1) + RS_g[i][j](n) \\ &- \left(\widehat{u_g}[i][j](n) + \alpha \cdot \widehat{\sigma_g}[i][j](n)\right)\} \end{aligned} \tag{8}$$

where $g = 1, 2, 3, 4, 5,$ is the traffic feature index described above, $M_g[i][j](0) = 0,$ α is a tunable parameter, $\widehat{u_g}$ and $\widehat{\sigma_g}$ are the mean value and standard deviation until n^{th} time interval that estimated by Exponential Weighted Moving Average (EWMA) method [35], $\widehat{u_g}$ and $\widehat{\sigma_g}$ are calculated as follows:

$$\widehat{u_g}\ [i][j](n) = \beta \cdot \widehat{u_g}\ [i][j](n-1) + (1-\beta) \cdot RS_g[i][j](n) \tag{9}$$

$$\begin{aligned} \widehat{\sigma_g}[i][j](n) \\ &= \ \beta \cdot \widehat{\sigma_g}[i][j](n-1) + \ (1-\beta) \\ &\cdot \sqrt{\left|RS_g[i][j](n) - \widehat{u_g}\ [i][j](n)\right|} \end{aligned} \tag{10}$$

where β is a tunable parameter. The calculation of the mean value and standard deviation using EWMA can counter seasonal variations. We discuss how to set the value of β in the next section.

$M_g[i][j](n)$ is 0 under normal conditions. But when there is a significant deviation, $M_g[i][j](n)$ will become a positive number, which can be considered as an attack alarm. Let $A_g[i][j](n)$ be the decision result at time *n* for traffic feature g in bucket $RS[i][j]$: "0" indicates normal bucket and "1 and 0.5" indicates abnormal bucket:

$$when\ g \in \{1,2,3\}, A_g[i][j](n) = \begin{cases} 1, if\ M_g[i][j](n) > \tau \\ 0, otherwise \end{cases} \tag{11}$$

$$when\ g \in \{4,5\}, A_g[i][j](n) = \begin{cases} 0.5, if\ M_g[i][j](n) > 0 \\ 0, otherwise \end{cases} \tag{12}$$

where τ is a positive numer, whose value setting will be discussed further in the next section. The reason that we set τ as positive numbers instead of zero in (11) is to offset network fluctuation. In (12), we set the threshold to zero as it is hard to decide an accurate deviation value between the distribution of packet size/destination address.

The final decision of bucket $RS[i][j]$ is the decision fusion of $A_g[i][j](n)$:

$$FA[i][j](n) = \begin{cases} abnormal, if \sum_{g=1}^{5} A_g[i][j](n) \geq \delta \\ normal, otherwise \end{cases} \tag{13}$$

where δ is a tunable threshold, which will be further discussed about how to select its value in the next section. The reason that we define this judge criterion is when DDoS flooding attacks occur, a drastical increase in the number of TCP/UDP/ICMP packets is an indicator of attacks. So we set $A_g[i][j](n) = 1$ if $M_g[i][j](n) > \tau$ in Equation (11). Moreover, the scanning behavior of a host that sends out a number of similar packets to many destination addresses/one destination address will cause anomalies in traffic features (4) and (5). We set $A_g[i][j](n) = 0.5$ if $M_g[i][j](n) > 0$ in (12). Setting $A_g[i][j](n) = 0.5$, when $g \in \{4,5\}$, is because the fourth and fifth features are not so strong attack indicator as the first three ones.

Notably, $M_g[i][j](n-1)$, $\widehat{u_g}\,[i][j](n-1)$ and $\widehat{\sigma_g}[i][j](n-1)$ are the results of the latest normal CRT-RS. That is to say, when an attack alarm has been raised, these values will no longer update until the alarm is free.

4.2.3 DDoS Flooding Attack Mitigation

After locating the abnormal buckets in each row, we can reversely compute the abnormal source addresses using the RECOVER function of CRT-RS. We get column number in each row (the c_i in (6) and use these numbers and (7) to gain source addresses. Then, the abnormal source addresses are added to a blacklist and a bloom filter is used to quickly look up whether the source address of new arriving traffic is present on the blacklist. Malicious traffic verified by the blacklist is filtered.

The general DDoS flooding attack detection and mitigation is described as Algorithm 1. We also describe the detection process of the proposed method in Algorithm 2.

Algorithm 1: DDoS Flooding Attack Detection and Mitigation

1. Use bloom filter to look up whether the source address of new arriving traffic is in the blacklist. If so, the traffic is filtered. Otherwise, transmit the traffic to a collector based on CRT-RS.
2. Use CRT-RS of the collector to record the statistical information of the traffic.
3. After a collection time, employ MM-CUSUM algorithm to locate abnormal buckets.
4. Call the RECOVER function of CRT-RS to compute abnormal source addresses and add the abnormal addresses to the blacklist.
5. Send the blacklist to bloom filter and update dataset that store the abnormal source addresses.
6. Return step 1.

Algorithm 2. The detection process of the proposed method

Define:
N is the number of new arriving packets;
H is the number of hash functions of CRT-RS;
SIP_s is Source IP address of the s-th new arriving packet;
1.1.1.1.1 h_i is the hash function of row i;

p_i is the modulo of row i;
1.1.1.1.2 $A[i][j]$ is the decision result for traffic features of bucket $RS[i][j]$;

1.1.1.1.3 $M_g[i][j]$ is the result of MM-CUSUM algorithm for g-th traffic feature in bucket $RS[i][j]$,

$abnormal\_list_i$ is the abnormal column number of row i;
$abnormal\_SIP$ is the list of abnormal source IP addresses detected by the method;
τ is a tunable threshold;
δ is another tunable threshold.

Input: New arriving packets, $p_s = <SIP_s>$, s =1, 2…N,
Output: $abnormal\_SIP$

```
Begin:
    For $s$=1 to $N$ do:
        For $i$=1 to $H$ do:
            $RS[i][h_i (SIP_s)]$ += 1
        end for
    end for
    For $i$=1 to $H$ do:
        For $j$=1 to $p_i$ do:
            $A[i][j]$= 0
            For $g$=1 to 3 do:
                If $M_g[i][j] > \tau$ do:
                    $A[i][j]$ +=1
                end if
            end for
            For $g$=4 to 5 do:
                If $M_g[i][j] > 0$ do:
                    $A[i][j]$ +=0.5
                end if
            end for
            If $A[i][j] \geq \delta$, do:
                add $j$ to the $abnormal\_list_i$
            end if
        end for
    end for
    $abnormal\_SIP$ = RECOVER($abnormal\_list_i$, $i$ = 1, 2, …, 5)
```

5. Experiments and Performance Evaluation

In this section, we perform a numer of experiments to evaluate the performance of our proposed method in terms of efficiency, accuracy (true positive rate and false positive rate), adaptability and protocol independability. We first give a discussion on parameter configurations. Then, we briefly introduce the evaluation criteria and datasets used in our experiments. At the end, a thorough experimental evaluation on our detection and mitigation method is reported.

5.1. Parameter Configurations

In this subsection, we discuss how to determine the proper configuration of each parameter because the effectiveness of the proposed method highly depends on the proper configurations of the parameters.

(1) **CRT-RS's parameters**: CRT-RS has two parameters, namely the number of hash functions H and the column number of each row $p_1, p_2, p_3, \dots, p_H$. In our proposed method, the *key* in CRT-RS is selected as source IP address. The key space is 2^{32}. In order to guarantee the accuracy and economic memory consumption of CRT-RS, we select *H*=5 and choose pair-wise coprime integers around 2^{12} as the values of $p_1, p_2, \dots p_5$ see also [30]. Thus, the largest memory consumption is 928 KB when *H=5* and $p_1, p_2, \dots p_5$ are around 2^{12} (i.e., each bucket contains five feature recorders and each bucket in the first four rows also contains one flag).

(2) **Bloom filter's parameters**: A Bloom filter has two parameters, namely the number of hash functions *K* and the hash table size *m*. How to properly set values of *K* and *m* is elaborately discussed in [34]. Through study, we set *K* = 15 and *m* = 2^{22}, herein.

(3) **MM-CUSUM's parameters**: There are two parameters in MM-CUSUM algorithm, namely scaling factor α in Equation (8) and EWMA's weighting coefficient β in Equation (9) and (10). The value of α limits the fluctuation range of $RS_g[i][j](n)$. The value of β determines the weight of current data. A larger value of β indicates that the current data is less important in the calculation of EWMA. However, the proposed method is not much sensitive to the values of α and β. Therefore, we set α as 2 and β as 0.1 as their default values because current data is more important than historical data in our method. We also provide experimental results of the effects of α and β on the detection method in next subsection.

(4) **Decision threshold:** There are two important threshold parameters directly influence the performance of the propoased method, namely decision threshold τ in Equation (11) and final decision threshold δ in Equation (13). They are all positive numbers with different scales and are used to determine whether a bucket is abnormal or not. If we select small values of τ and δ, we will get low false negative rate but high false positive rate. However, large values of τ and δ lead to high false negative rate and low false positive rate. Besides, δ is more important than τ. Thus, we evaluate the effects of them and select proper values in order to get high accuracy rate and lower false positive rate.

(5) **Detection time interval**: The detection time interval T is also an important parameter for attack detection and mitigation. It determines the response time of the system to attacks. A small value of T may cause frequent detection and reporting on attacks, which leads to high computation consumption. In addition,

we may obtain insufficient information of network traffic with a small value of T. However, a big value of T may result in response delay, and thus increase the false negative rate. Therefore, a trade-off must be made in selecting the detection time interval. We examine the effect of the detection time interval in our experiments.

Table I summarizes the parameters of the proposed method and their default and experimental values.

Table I. Summary of the parameters used in experiments

	Parameters	Description	Default Values
Sketch	H	Number of hash functions	5
	$p_1, p_2, \dots p_5$	Column number of each row	4139, 4129, 4111, 4093, 4079
Bloom filter	K	Number of hash functions	15
	m	Hash table size	2^{22}
MM-CUSUM algorithm	α	Scaling factor	2 (Tested from 1 to 5)
	β	Weighting coefficient	0.1 (Tested from 0.1 to 0.5)
Decision threshold	τ	Detection threshold	60 (Tested from 0 to 100)
	δ	Final decision threshold	1.5 (Tested from 0 to 4)
Detection time interval	T	Detection time	20s (Tested from 10s to 60s)

5.2. Evaluation Criteria and Datasets

1) **Evaluation criteria:** we employ True Positive Rate (TPR), False Positive Rate (FPR) as criteria to measure the effects of parameters on the performance of the proposed method. TPR represents the proportion of the attacking addresses that are correctly identified as malicious by the method. FPR represents the proportion of benign addresses that are mistakenly identified as malicious by the method. The effect of the parameter is evaluated by adjusting its value in a proper range (as shown in Table I) while keeping others as defaults. We prefer a high TPR and a low FPR. We also take F-score as a criterion in order to find the most suitable parameter setting. F-Score is calculated as follows,

$$\mathrm{F-Score} = \frac{2PR}{P+R} \tag{14}$$

where P is the precision and R is recall.

2) **Datasets:** Real-world traffic has more uncertainty and randomness than simulated network traffic. Therefore, using real-world network traffic instead of simulated one to test our method can sufficiently verify its quality attributes, e.g., effectiveness. Our experiments were carried out over two up-to-data traffic datasets.

- **M**easurement and **A**nalysis on the **WI**DE **I**nternet (MAWI) dataset [36]: The traffic traces in MAWI dataset are all collected from real-world networks. Each trace is a pcap file providing raw packets that were captured for 15 minutes everyday, since 2001 until now, on a trans-Pacific link between Japan and US. The dataset is daily updated to include new traffic traces. Moreover, each trace consists of a set of labels that indicate traffic anomalies. The labels are obtained by using an advanced graph-based methodology that compares and combines different and independent anomaly detectors. The detailed classification principle is described in [37, 38]. Table II gives the detailed information of traffic traces (IPV4 protocol) selected from the MAWI dataset. We used these traffic traces to evaluate the effects that the parameters bring to the performance of the proposed method.

- **C**anadian **I**nstitute for **C**ybersecurity (CIC) dataset [39]: CIC-IDS-2017 dataset contains benign and the most up-to-date common attacks, which resemble the true real-world data. It also includes the results of the network traffic analysis using CICFlowMeter with labeled flows based on time stamp, source and destination IPs, source and destination ports, protocols and attacks [40]. The data capturing period started at 9 a.m., Monday, July 3, 2017 and ended at 5 p.m. on Friday July 7, 2017, for a total of 5 days. We use Friday traffic trace that includes the traffic of DDoS flooding attacks to test the TPR and FPR of the proposed method, refer to Table II as to the details of CIC-Friday dataset.

Table II. Detailed information of traffic traces of two testing datasets

Dataset	Number of packets	Number of TCP packets	Number of UDP packets	Number of ICMP packets	Number of source addresses
MAWI-20150502	77,127,397	41,676,003 (52.78%)	6,814,548 (8.63%)	24,470,818 (30.99%)	4,901,419
MAWI-20150304	86,938,488	59,173,952 (63.92%)	2,802,067 (3.03%)	22,544,145 (24.35%)	5,063,104
CIC-Friday	9,915,173	9,192,354 (92.71%)	722,819 (7.29%)	0	8,278

As can be seen from Table II, our testing dataset contains a huge numer of packages delivered based on different networking protocols. Using these datasets to evaluate the performance of the method can verify its detection efficiency, accuracy, adaptability, and protocol independability.

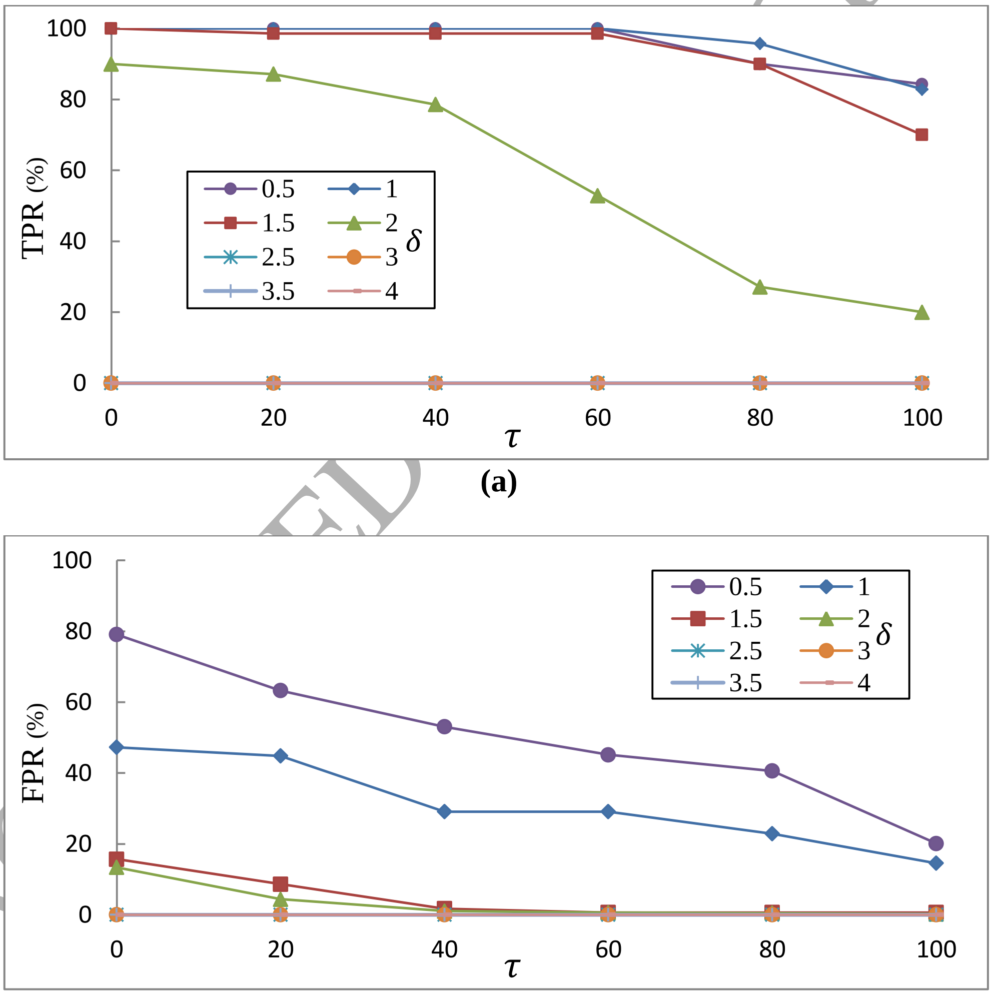
5.3. Settings of System Parameters

The experiments were conducted in a Windows Server 2012 with CPU E5-2630@ 2.2GHz and 16 GB RAM using python 2.7. We extracted the corresponding fields (source address, destination address, protocol, timestamp and length) from IPv4 packet headers and input the processed traffic information into our detection method.

5.3.1 Effects of Decision Threshold τ and δ

Decision threshold τ in (11) determines whether the traffic feature (the number of TCP/UDP/ICMP) is abnormal and final decision threshold δ in Condition (13) determines whether a bucket is abnormal. Figure 6 and Figure 7 show FPR and FNR evaluated by using different values of τ and δ based on the two MAWI traffic traces. We can see that the performance of the proposed detection method is highly sensitive to the changes of τ and δ. As indicated in these figures, when δ is small (0.5 and 1), TPR and FPR are high, no matter what values τ are taken. The mainly reason is that if δ is small, the anomalies in terms of the fourth and fifth traffic features (the distribution of packet size/destination address, as discussed in Subsection 4.2.1) will also trigger an alarm. But based on the analysis of the characteristics of DDoS flooding attacks, fourth and fifth features are not so strong to be considered as attack indicators if not measuring them together with the first three traffic features (the number of TCP/UDP/ICMP packets, as discussed in Subsection 4.2.1). Hence, a high FPR will be generated when setting δ as 0.5 and 1, as shown in Figure 6(b) and 7(b). When increase δ to 1.5 and 2, TPR also keeps high when $\tau \leq 60$ and becomes lower and lower with the increase of τ. At the same time, FPR decreases as τ increases. When increase δ to 2.5, 3, 3.5 and 4, TPR and FPR are almost zero because there are no anomalies can be detected by the method. An anomaly will be detected in the case of at least three features exhibiting abnormality under large values of δ. However, this situation usually will not happen. We also calculated the F-Score in order to clearly find the most proper setting of τ and δ under so many tests, as shown in Table III.

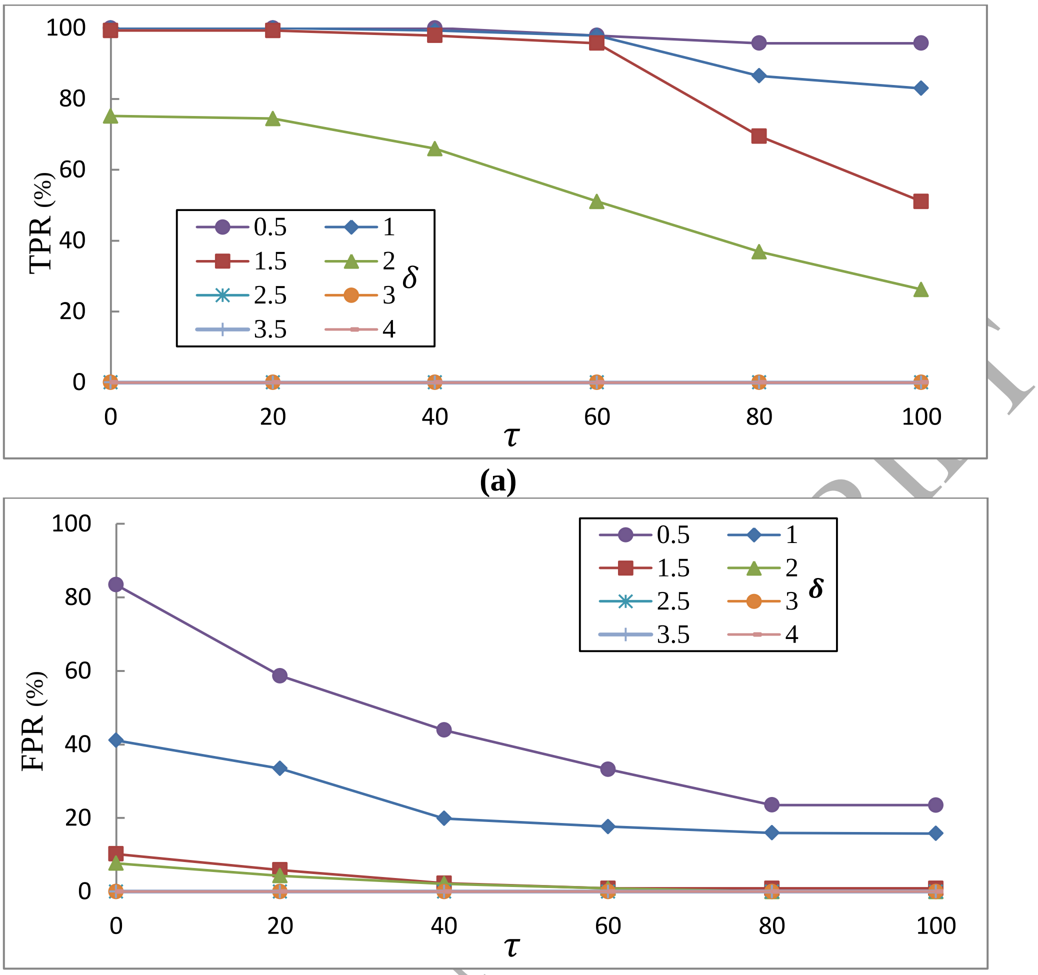
Based on the above discussion and the results indicated by Figure 6 and 7 and Table III, we select $\tau = 60$ and $\delta = 1.5$ as their proper values for achieving high TPR and low FPR.



(a)

(b)

Figure 6. Effect of τ and δ on MAWI-20150502. (a) TPR (B) FPR



(a)

(b)

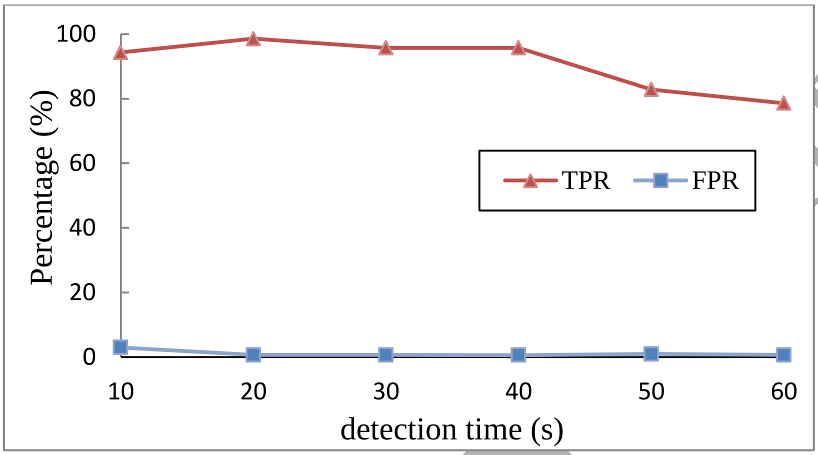
Figure 7. Effect of τ and δ on MAWI-20150304. (a) TPR. (B) FPR

Table III. The results of F-Scores ($\times 10^3$)

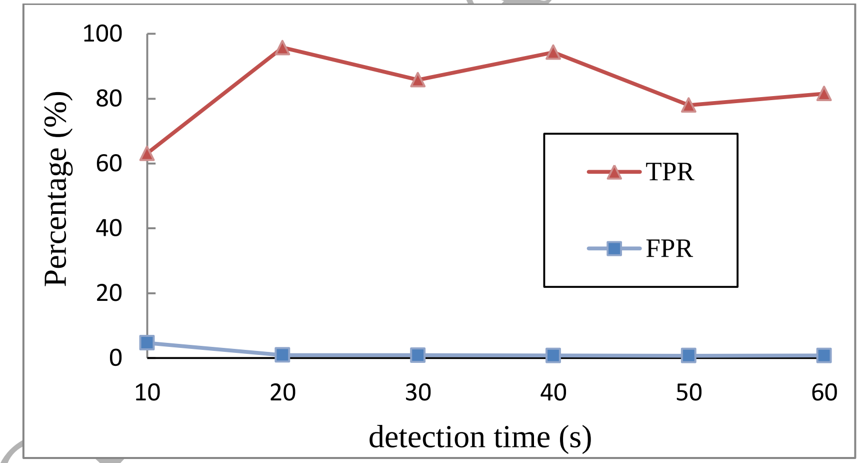
	MAWI-20150502						**MAWI-20150304**					
δ \ τ	0	20	40	60	80	100	0	20	40	60	80	100
0.5	0.036	0.045	0.053	0.063	0.063	0.120	0.067	0.095	0.127	0.164	0.226	0.227
1	0.060	0.063	0.098	0.098	0.119	0.162	0.135	0.167	0.278	0.309	0.301	0.292
1.5	0.181	0.326	1.613	**4.241**	3.871	3.010	0.541	0.941	2.396	**6.022**	4.382	3.22
2	0.192	0.56	2.142	2.565	1.707	2.291	0.543	0.961	1.737	3.316	2.597	0.251
2.5	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
3.5	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0

5.3.2 Effect of Detection Time *T*

Figure 8 shows the changes of TPR and FPR with different detection time *T* based on the two MAWI traffic traces. We observe that TPR keeps changing with the increase of *T* while FPR has a little change but within our acceptable range for both traffic traces. Hence, we select *T*=20s as a suitable setting.



(a)

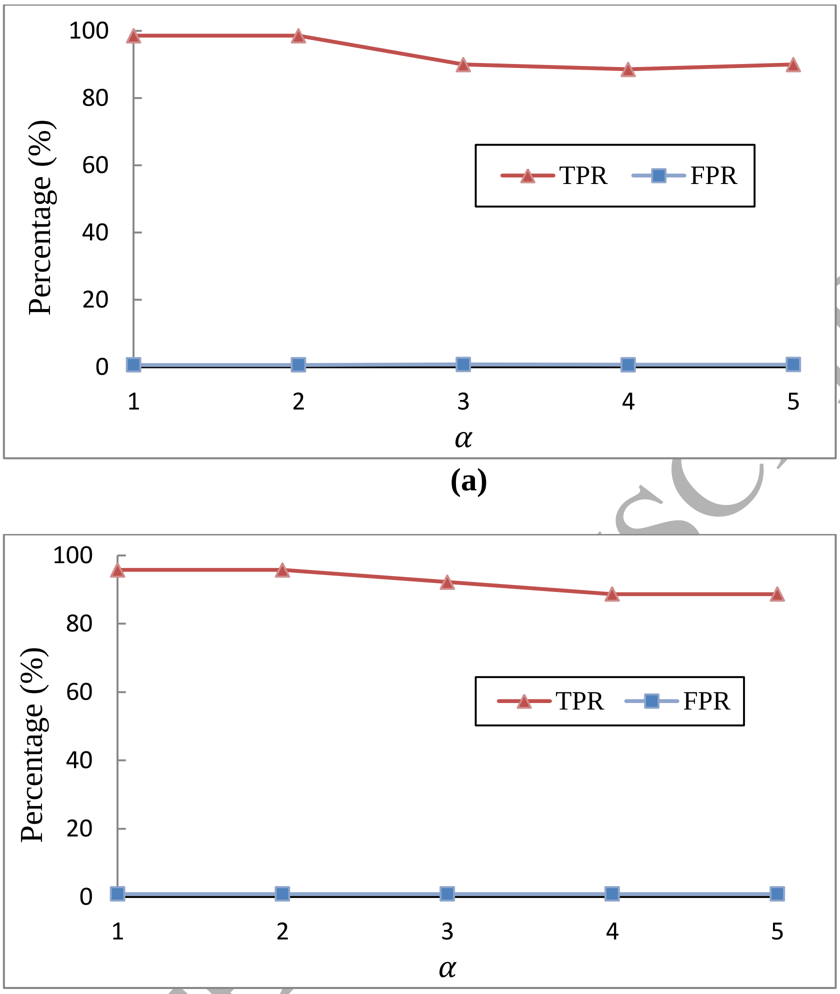


(b)

Figure 8. Effect of detection time *T*. (a) MAWI-20150502 (b) MAWI-20150304.

5.3.3 Effects of α and β

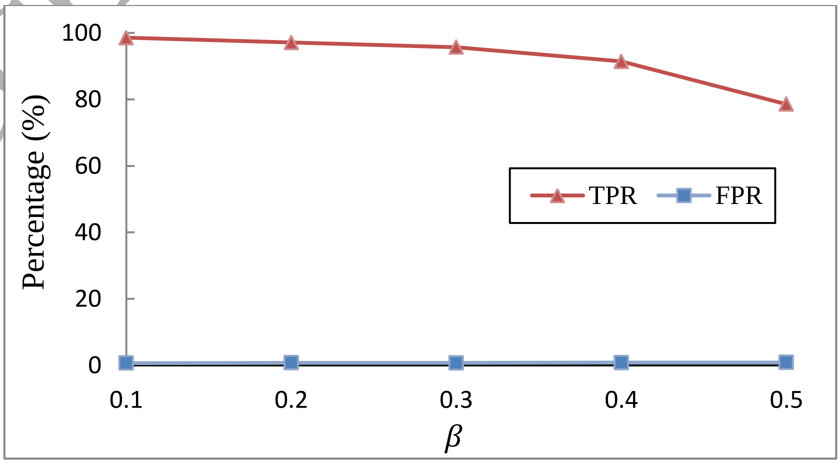
We also evaluate the impact of α and β, as shown in Figure 9 and Figure 10. The results indicate that the values of α and β has less impact on the performance of detection method. From (8)-(10), we can see that α determines fluctuation range of current data and β determines the weight of historical data in detection process. Usually, when DDoS flooding attacks is occurring, significant changes in the features of traffic will result in high deviation between current data and historical data and also break through the maximum fluctuation range of data. This is the main reason that the performance of the proposed method is not sensitive to the values of α and β. Hence, we set $\alpha = 2$ and $\beta = 0.1$.



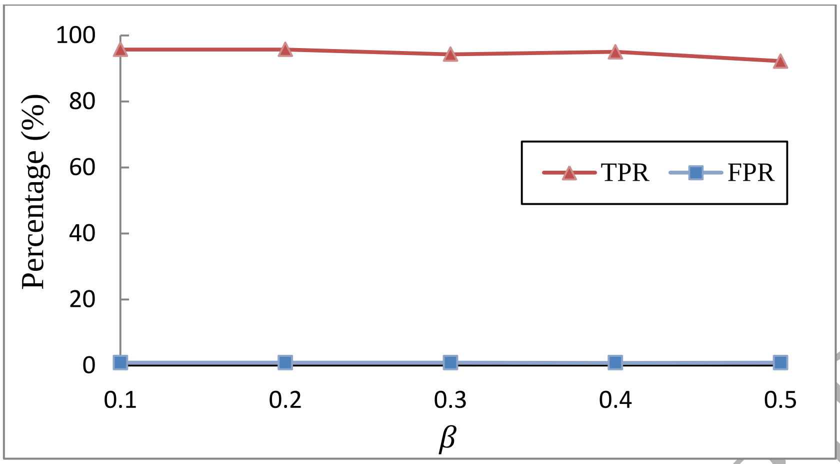
(a)

(b)

Figure 9. Effect of α. (a) MAWI-20150502 (b) MAWI-20150304.



(a)



(b)

Figure 10. Effect of β: (a) MAWI-20150502; (b) MAWI-20150304.

5.4. Effectiveness Evaluation

We compare the method with some existing detection methods using sketch techniques, as shown in Table IV. We can see that our method not only supports self-adaptability and protocol independability with reversible sketch, but also achieves the lowest complexity of reversible computation. Moreover, we also investigate the effects of system parameters on the detection performance, which makes our method reliable to network dynamics with easily configured parameters. Based on the comparison and the above experimental results, we take some discussion on the effectiveness of the proposed detection method for DDoS flooding attacks in terms of *efficiency, accuracy, adaptability* and *protocol independability*.

Table IV. Comparison with existing methods

	Reversible sketch	**Complexity of reversible computation**	**Self-adaptability**	**Protocol independability**	**Evaluation on the impact of each parameter**
[27]	Not Consider	-	Yes	No	No
[30]	*Reverse hashing –* based sketch	$O(qHM(\frac{n}{M})^{1/q})$	No	No	No
[41]	*Reverse hashing –* based sketch	$O(qHM(\frac{n}{M})^{1/q})$	No	Yes	No
[42]	Not Consider	-	Yes	No	No
[43]	Not Consider	-	No	No	No
[44]	Not Consider	-	No	Yes	No
[45]	Bitwise-based Sketch	$O(H*q)$	No	No	No
Our method	CRT-based sketch	$O(H)$	Yes	Yes	Yes

Yes: support the requirement; No: do not support the requirement;
H: the number of hash functions; M: the size of hash table; n: the size of key space;
q: the number of discrete segments of IP;

5.4.1 Efficiency

(1) Suitability for dealing with big-volume network traffic

As shown in Table II, there are a large number of different source addresses in each data trace, keeping a counter for each source address to record traffic information is either too expensive or too slow. By benefiting from the advance of the sketch techniques that compress and fuse network traffic into a compact and constant-size traffic summary, the proposed method has the ability of dealing with big-volume network traffic in an efficient way no matter how many source addresses exist. This ability mainly depends on the number of hash functions that determines the processing time and the number of columes in each row that determines the memory size. In order to evaluate the efficiency of the proposed method to handle big-volume network traffic, we take throughput defined as the number of packets processed by the CRT-RS per second as a measure [34] to test its performance. Figure 11 shows the relationship between the throughput and the number of hash functions in a single CPU core. However, the calculations of different hash functions are independent and can be paralleled with multiple cores. If we use multiple CPU cores for hashing, the total throughput will become higher. We also tested the processing time of the proposed detection method in each detection interval and show part of the results in Figure 12. The results indicate that our method can analyze traffic and then reversibly calculate abnormal source addresses always within 1 second. Therefore, based on the above discussion, the proposed method is suitable for handling big-volume network traffic.

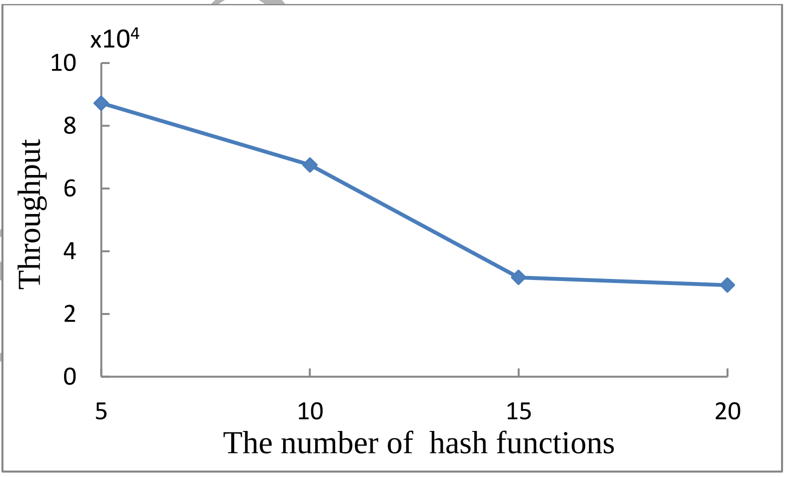


Figure 11. The throughput of the CRT-RS

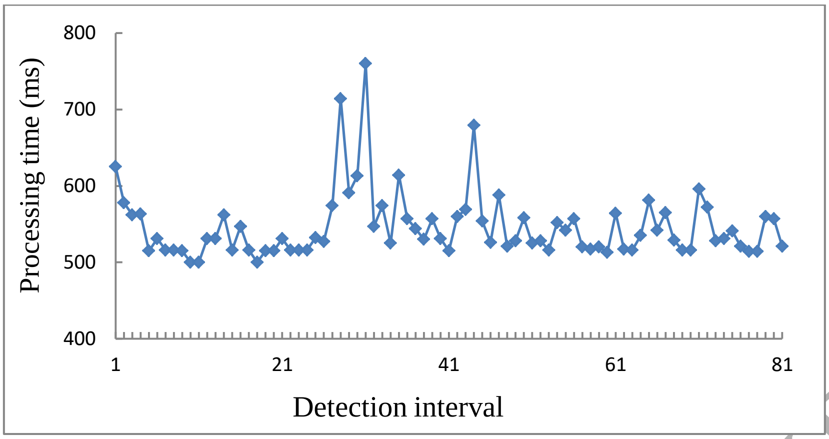


Figure 12. The processing time of the CRT-RS

(2) Low computation complexity in recovering process

As shown in Table IV, when compared with other reversible sketch techniques, CRT-RS has lower complexity of reversible computation. It is very important to have low computation complexity when reversely calculating the abnormal source addresses. The main step of CRT-RS is to utilize the solution to congruence equations in the Chinese Remainder Theorem to reversely compute the keys. The complexity of reversible computation depends on the number of hash functions, which will take *H* times modulus calculation. Thus, the computation complexity of CRT-RS is only *O(H)*, which is much lower than the complexity of other existing methods.

5.4.2 Accuracy

In order to demonstrate the accuracy of the proposed method, we conducted another experiment using CIC-Friday traffic trace. Table V provides the numerical experimental results of the method measured by two MAWI traffic traces and CIC-Friday traffic trace with proper parameter settings. In order to qualitatively compare our method with other state-of-the-art methods, we select two methods that both use sketch data structure to record network traffic information and also conduct experiments on MAWI dataset, namely the Multi-scale Principal Component Analysis (MSPCA) based detection method [44] and the information theory based detection method [41]. The results of comparison are shown in Table V. With proper configuration of the parameters, the TPR and FPR of MSPCA-based method are about 79.36% and 0.13%, respectively. The TPR and FPR of information theory based detection method are about 90% and 5%, respectively. The experimental and compared results indicate that the proposed detection method can accurately and reliable detect DDoS flooding attacks with relative high TPR and low FPR if the values of the parameters have been carefully selected.

Table V. Experimental results with proper parameters based on MAWI and CIC datasets

	Dataset	TPR (%)	FPR (%)
Our method	MAWI-20150502	98.57	0.66

	MAWI-20150304	95.74	0.88
	CIC-Friday	100	1.35
MSPCA based method [44]	MAWI	79.36	0.13
Information theory based method [41]	MAWI	90	5

5.4.3 Self-adaptability

The proposed method is self-adaptive for detecting DDoS flooding attacks. This is the benefit stemming from the CUSUM algorithm. CUSUM is one of the commonly used change point detection algorithms applied to monitor abrupt changes of network traffic with the smallest possible time delay. It only requires the distribution of traffic data before and after the changes rather than prebuilding normal profile of network activities. Thus, adopting CUSUM algorithm makes the detection process adaptive to recent traffic changes, there is no need to make judgement based on a previously established static profile. Such a self-adaptive property makes the proposed method more flexible and reliable to counter network attacks.

5.4.4 Protocol Independability

The method makes full use of basic attack characteristics of DDoS flooding attacks that cause a visible increase in the number of packets and an abnormal in the distribution of packet size/destination address. There are many protocols that can be exploited for launching DDoS flooding attacks. We cannot predict when and which type of DDoS flooding attacks will occur. A protocol-independent method is highly expected for evaluating the Internet security as a whole. In this study, we use a multi-dimensional change-point detection method to find out the anomalies buckets in each row of CRT-RS. This design can support handling abnormal attacks under different protocols. We also conducted the corresponding experiments to verify the protocol independability. As shown in Table II, MAWI and CIC traffic traces have a large number of packets with different protocols. The experimental results indicated by Table V show that the detection method achieves a high accuracy rate based on the datasets that contain a huge number of packets delivered with different types of protocols. Thus, the method is a generic solution that can combine different detection methods based on one uniform reversible sketch data structure and also achieves high detection efficiency.

6. Conclusions

This paper proposed a DDoS flooding attack detection method to alleviate the impact that the big-volume network traffic brings to attack detection and realize self-adaptive and protocol independent detection. We employed CRT-RS to record traffic information and used MM-CUSUM to carry out attack detection. Comparing with the previous methods used for detecting DDoS flooding attacks, the developed method is more efficient in terms of handling big-volume network traffic. It exhibits lower computational complexity when recovering the anomalous source addresses, and is more accurate than other detection methods with adaptability and protocol independency. Based on the completed study, we can conclude that the CRT-RS is capable of dealing with big-volume network traffic if system parameter configuration can be set properly. It is also very useful to support the mitigation of DDoS flooding attacks. In the future, one could envision studies focused on the design of multi-dimensional CRT-RS to perform traffic fusion for synthesis attack detection.

Acknowledgments

This work is sponsored by the National Key Research and Development Program of China (grant 2016YFB0800704), the NSFC (grants 61672410 and U1536202), the Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016ZDJC-06), Academy of Finland (grant No. 308087), and the 111 project (grants B08038 and B16037).

References

[1] X. Y. Jing, Z. Yan, W. Pedrycz, Security Data Collection and Data Analytics in the Internet: A Survey, IEEE Communications Surveys and Tutorials, 2018, doi: 10.1109/COMST.2018.2863942.

[2] S. T. Zargar, J. Joshi, and D. Tipper, A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks, IEEE Communications Surveys and Tutorials, 15 (4) (2013) 2046-2069.

[3] M. H. Jiang, C. X. Wang, X. P. Luo, M. T. Miu, and T. Chen, Characterizing the Impacts of Application Layer DDoS Attacks, in Proceedings of IEEE International Conference on Web Services, 2017, pp. 500-507.

[4] C. Rossow, Amplification Hell: Revisiting Network Protocols for DDoS Abuse, in Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium, 2014, pp. 1-15.

[5] F. J. Ryba, M. Orlinski, M. Wahlisch, C. Rossow, and T.C. Schmidt, Amplification and DRDoS Attack Defense - A Survey and New Perspectives, arXiv: 1505.07892, 2015, [Online], Available: https://arxiv.org/abs/1505.07892.

[6] C. Liaskos and S. Ioannidis, Network Topology Effects on the Detectability of Crossfire Attacks, IEEE Transactions on Information Forensics and Security, 13 (7) (2018) 1682-1695.

[7] C. Liaskos, V. Kotronis, and X. Dimitropoulos, A Novel Framework for Modeling and Mitigating Distributed Link Flooding Attacks, in Proceedings of IEEE International Conference on Computer Communications, 2016, pp. 1-9.

[8] A. L. Buczak and E. Guven, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection, IEEE Communications Surveys and Tutorials, 18 (2) (2016) 1153-1176.

[9] Y. Wang, Y. Xiang, J. Zhang, W. L. Zhou, G. Y. Wei and L. T. Yang, Internet Traffic Classification Using Constrained Clustering, IEEE Transactions on Parallel and Distributed Systems, 25 (11) (2014) 2932-2943.

[10] G. Q. Li, Z. Yan, Y. L. Fu, and H. L. Chen, Data Fusion for Network Intrusion Detection: A review, Security and Communication Networks, 2018 (2018).

[11] D. H. Zhou, Z. Yan, Y.L. Fu, Z. Yao, A Survey on Network Data Collection, Journal of Network and Computer Applications, 116 (2018) 9-23.

[12] H. Q. Lin, Z. Yan, Y. Chen, L.F. Zhang, A Survey on Network Security-Related Data Collection Technologies, IEEE Access, 6 (1) (2018) 18345-18365.

[13] G. Liu, Z. Yan, W. Pedrycz, Data Collection for Attack Detection and Security Measurement in Mobile Ad Hoc Networks: A Survey, Journal of Network and Computer Applications, 105 (2018) 105-122.

[14] L. M. He, Z. Yan, M. Atiquzzaman, LTE/LTE-A Network Security Data Collection and Analysis for Security Measurement: A Survey, IEEE Access, 6 (1) (2018) 4220-4242.

[15] Graham Cormode, Sketch techniques for approximate query processing, Foundations and Trends in Databases, 2011.

[16] M. Bellaiche and J. C. Gregoire, SYN Flooding Attack Detection Based on Entropy Computing, in Proceedings of Global Telecommunications, 2009, pp. 1-6.

[17] Y. Kim, W. C. Lau, M. C. Chuah, H. J. Chao, PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks, IEEE Transactions on Dependable and Secure Computing, 13 (2) (2006) 141-155.

[18] S. Yu, W. L. Zhou, W. J. Jia, S. Guo, Y. Xiang, and F. L. Tang, Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient, IEEE Transactions on Parallel and Distributed Systems, 23 (6) (2012) 1073-1080.

[19] M. Alenezi and M. J. Reed, Denial of Service Detection Through TCP Congestion Window Analysis, in Proceedings of World Congress on Internet Security, 2013, pp. 145-150.

[20] W. Xiong, H. Hu, N. Xiong, L. T. Yang, W. C. Peng, X. Wang, and Y. Qu, Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications, Information Sciences, 258 (2014) 403-415.

[21] L. H. Chi, X. Q. Zhu, Hashing Techniques, ACM Computing Surveys, 50 (1) (2017) 1-36.

[22] B. Krishnamurthy, S. Sen, Y. Zhang, Y. Chen, Sketch-based Change Detection: Methods, Evaluation, and Applications, in Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, 2003, pp. 234-247.

[23] M. Charikar, K. Chen, M. F. Colton. Finding frequent items in data streams, in Proceedings of International colloquium on Automata, Languages and Programming, 2002, pp. 693-703.

[24] G. Cormode and S. Muthukrishnan, An improved data stream summary: the count-min sketch and its applications, Journal of Algorithms, 55 (1) (2005) 58-75.

[25] P. Roy, A. Khan, G. Alonso, Augmented sketch: Faster and more accurate stream processing, in Proceedings of the 2016 International Conference on Management of Data, 2016, pp. 1449-1463.

[26] T. Yang, Y. Zhou, H. Jin, S. G. Chen , X. M. Li, Pyramid Sketch: a Sketch Framework for Frequency Estimation of Data Streams, Proceedings of the VLDB Endowment, 10 (11) (2018) 1442-1453.

[27] J. Tang, Y. Cheng, Y. Hao, and W. Song, SIP Flooding Attack Detection with a Multi-Dimensional Sketch Design, IEEE Transactions on Dependable and Secure Computing, 11 (6) (2014) 582-595.

[28] O. Salem, A. Makke, J. Tajer, A. Mehaoua, Flooding Attacks Detection in Traffic of Backbone Networks, in Proceedings of 2011 IEEE 36th Conference on Local Computer Networks, 2011, pp. 441-449.

[29] C. Callegari, S. Giordano, M. Pagano, T. Pepe, Combining sketches and wavelet analysis for multi time-scale network anomaly detection, in Proceedings of IEEE International Conference on Intelligent Computer Communication & Processing, 2010, pp. 313-319.

[30] R. Schweller, Z. C. Li, Y. Chen, et al., Reversible Sketches: Enabling Monitoring and Analysis Over High-Speed Data Streams, IEEE/ACM Transactions on Networking, 15 (5) (2007) 1059-1072.

[31] Z. C. Li, Y. Gao, Y. Chen, HiFIND: A high-speed flow-level intrusion detection approach with DoS resiliency, Computer Networks, 54 (2010) 1282-1299.

[32] O. Salem, S. Vaton, A. Gravey, A scalable, efficient and informative approach for anomaly-based intrusion detection systems: Theory and practice, International Journal of Network Management, 20 (5) (2010) 271–293.

[33] P. H. Wang, X. H. Guan,T. Qin, Q. Z. Huang, A Data Streaming Method for Monitoring Host Connection Degrees of High-Speed Links, IEEE Transactions on Information Forensics and Security, 6 (3) (2011) 1086-1098.

[34] C. X. Wang, T. T. Miu, X. P. Luo , J. H. Wang, SkyShield: A Sketch-Based Defense System Against Application Layer DDoS Attacks, IEEE Transactions on Information Forensics and Security, 13 (3) (2018) 559-573.

[35] D. C. Montgomery, Introduction to Statistical Quality Control, Hoboken, NJ, USA: Wiley, 2007.

[36] MAWI. MAWIWorking Group traffic archive, Accessed: August 19, 2018, [Online], Available: http://mawi.wide.ad.jp/mawi/.

[37] MAWILab, Accessed: August 19, 2018, [Online], Available: http://www.fukuda-lab.org/mawilab/.

[38] R. Fontugne, P. Borgnat, P. Abry, K. Fukuda, MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking, in Proceedings of the 6th International COnference, 2010, pp. 1-12.

[39] CIC dataset, Accessed: August 19, 2018, [Online], Available: http://www.unb.ca/cic/datasets/flowmeter.html.

[40] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, in Proceedings of International Conference on Information Systems Security & Privacy, 2018, pp. 108-116.

[41] C. Callegari, S. Giordano, M . Pagano, An information-theoretic method for the detection of anomalies in network traffic, Computers & Security, 70 (2017) 351-365.

[42] O. Salem, A. Makke, J. Tajer, A. Mehaoua, Flooding Attacks Detection in Traffic of Backbone Networks, in Proceedings 2011 Ieee 36th Conference on Local Computer Networks, 2011, pp. 441-449.

[43] A. P. Li , Y. Han, B. Zhou, W. H. Han, Y. Jia, Detecting Hidden Anomalies Using Sketch for High-speed Network Data Stream Monitoring, Applied Mathematics & Information Sciences, 6 (3) (2012) 759-765.

[44] Z. M. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, Detection of network anomalies using Improved-MSPCA with sketches, Computers & Security, 65 (2017) 314-328.

[45] F. Wang, X. F. Wang, X. F. Hu, J. S. Su, Bitwise Sketch for Lightweight Reverse IP Reconstruction in Network Anomaly Detection, in Proceedings of 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems, 2012, pp. 1-4.