A RESOURCE-EFFICIENT COLLABORATIVE SYSTEM FOR DDOS ATTACK DETECTION AND VICTIM IDENTIFICATION

Fei Wang, Zhenxing Li and Xiaofeng Wang

School of Computer, National University of Defense Technology, Changsha, China

ABSTRACT

Distributed Denial of Service (DDoS) attacks seriously threaten network security. Most countermeasures perceive attacks after the damage has been down. This paper thus focuses on the detection of DDoS attacks, and more importantly, victim identification as early as possible, so asto promote attack reaction in time. We present a resource-efficient collaborative DDoS detection system, called F-LOW. Profiting from bitwise-based hash function, split sketch, and lightweight IP reconstruction, F-LOW can defeat shortcomings of principle component analysis (PCA) and regular sketch. With a certain number of distributed detection nodes, F-LOW can detect DDoS attacks and identify victim IPs before the attack traffic arrives victim network. Outperforming previous work, our system fits all Four-LOW properties, low profile, low dimensional, low overhead and low transmission, of a promising DDoS countermeasure. Through simulation and theoretical analysis, we demonstrate such properties and remarkable efficacy of our approach in DDoS mitigation.

KEY WORDS

DDoS detection, victim identification, principle component analysis, split sketch, bitwise-based hash.

1. INTRODUCTION

As cloud computing becomes ubiquitous on the Internet, it opens the door to many serious attacks, particularly Distributed Denial of Service (DDoS) attack. As one of the most serious threats to cloud security, DDoS attack affects cloud and datacenter service even worse than to regular Internet service. Regarding cloud and datacenter as an inherent multi-tenant infrastructure, DDoS attack against a single customer is against all customers in the infrastructure [1]. In recent years, DDoS attacks evolve both in the quantity and the destructive power of a single attack. The peak bandwidth of the largest DDoS attack in 2021 exceeds 3Tbps [2]. DDoS attacks in such scale can easily breakdown arbitrary online services and incur huge financial losses. Cloud providers need to do more to ensure the availability of their cloud services.

Attention is thereby devoted to countermeasures against DDoS attacks towards online services such as could computing. This paper is concerned with detection, and more importantly, victim identification of network-wide DDoS attacks, so that outcomes can contribute to quick reaction to such devastating attacks. The challenging is, the two countermeasures suffer from the increasing link bandwidth of current Internet as well as inconspicuous sources of DDoS attacks. From this perspective, a promising DDoS countermeasure system must have the following Four-LOW properties. (1) low profile: the system should have the capability of detecting low-profile network anomaly, so that DDoS attacks can be detected as early as possible; (2) low dimensional: in order to identify victim IPs of DDoS attacks, dimensional reduction mechanism is necessary for

processing high-dimensional data, such as per-IP flow statistic; (3) low overhead: expensive computational cost and memory consumption should be avoided in the system; (4) low transmission: if the system is distributed, collaborative nodes transmit as little data as possible, not exacerbating network congestion caused by an ongoing DDoS attack.

Previous works make progress in DDoS detection or identification, but they do not fit all Four-LOW properties described above. Many of them are insensitive to low-profile DDoS attacks and only can detect anomalies when attack traffic is aggregated conspicuously near the victims [3][4][5]. Lakihina et al. [6] apply PCA on origin-destination (OD) flows, the aggregations of traffic from a source router to a destination router. Taking advantage of PCA, this method can figure out malicious OD flows that includes low-profile attack traffic. However, this method cannot result in particular victim IPs due to coarse-grained aggregation. Applying PCA on fine-grained per-IP flows may help to distinguish victim IPs, but the data dimensionality jumps sharply to a high degree that PCA cannot afford to. To tackle this problem, Li et al. [7] use sketch to randomly aggregate IP flows. Although this method keeps input data of PCA in low dimensional space, it therewith poses another high-overhead problem. That is to infer key IPs reversely from a number of particular buckets. This process always involves great space in memory for storing mapping tables, as well as high computational cost for calculate intersections of multiple large sets [8][9][10]. Another problem that obstructs practical application of [6] and [7] is too much data sharing among collaborative nodes. Original network measurements, such as OD flows statistic [6] and sketches [7] are transmitted over the network to a centralized device, challenging sparing bandwidth capability of DDoS-compromised links. Recently, software-defined DDoS detection methods are proposed to confront DDoS attacks [15, 16] for cloud and datacenters.

To conquer above challenges, this paper proposes a collaborative DDoS detection and victim identification system, called F-LOW. Our system can satisfy all Four-LOW properties, thus being a promising countermeasure against network-wide DDoS attacks. Our main contributions are summarized as follows.

- We pioneerly propose split sketch which divides a sketch structure into pieces and deploy them on distributed network devices. In this way, network measurements all over the network seems to be in a sketch and can be treated relevantly, even without being sent to a centralized device.
- We formally define bitwise-based hash function and adopt it into split sketch for traffic aggregation. The aggregation mechanism not just significantly reduces data dimensionality in local detection, but also makes victim identification easy-to-implement.
- We extend existing PCA-based anomaly detection algorithm to detect multiple anomalous flows, suiting the needs of practical situations. Heuristic rules are presented to improve computational cost.
- We present a lightweight IP reconstruction algorithm to identify victim IPs of DDoS attacks. It can accurately infer victim IPs or partial victim IP with very low computational cost. Both outcomes help to effectively filter attack traffic during reaction.
- We demonstrate Four-LOW properties of our approach through simulation and theoretical analysis. When compared with the most relevant work, our approach has much lower overhead and greater scalability. We experimentally show that outcomes of our system, even being partial victim IPs, can help to filter DDoS traffic in reaction. The false positive rate is less than 3%.

The rest of this paper is organized as follows: Section 2 briefly introduces background knowledge and practical issues.Section 3 presents overview of the F-LOW system and Section 4 anatomizes

the design of F-LOW. We evaluate our F-LOW system in Section 5 and draw a conclusion in Section 6.

2. BACKGROUND

In this section, we briefly introduce two building blocks, sketch structure and Principle Component Analysis (PCA), that are widely used in the domain of network anomaly detection. We also discuss practical issues that obstruct these two methods from achieving great progress in DDoS detection and victim identification.

2.1. Sketch

Sketch is a sublinear space data structure capable of summarizing high-dimensional data streams. A sketch structure, in the form of an$H \times M$ matrix, is composed of hash tables of H different hash functions. Normally, these functions are selected from a universal family of hash functions.$M$is the size of hash table. An element in a sketch is called bucket. If we refer to a sketch as SK, then SK[i][j]represents the bucket in i-th row and j-th column of SK. When summarizing a data stream using a sketch, a bucket is regarded as a counter for counting the number of items that are mapped to it. In a data stream, each item is a two-tuples (key, value). When an item arrives, the sketch updates as follows. Let $h_0, h_1, \dots, h_H$be H hash functions. For a particularh_i, calculate the bucket index $j = h_i(key)$. Then the bucket SK[i][j] is updated to SK[i][j]+ value. For each item, H buckets are updated, one in each hash table. After compressing a data stream into a sketch, fundamental queries about the data stream can be approximately answered very quickly based on the sketch [11].

2.2. Principle Component Analysis

PCA is an efficient dimension reduction method. It projects raw high-dimensional data into a lower dimensional space, while keeping as most relevant variances of original data as possible [12].

Let X be an $n \times m$ data matrix. Each row of X is viewed as a point in an m-dimensional space. PCA projects points in X into an r-dimensional space ($r < m$) and finds projections that maximize the variance of n points. The first principal component is the direction along which projections of these points have the largest variance. By analogy, the r-th principal component is the orthogonal direction that has maximum variance with respect to previous $r - 1$components. Before performing PCA, a mean of zero is needed for each column so as to find a basis that minimizes the mean square error. Let Y be the zero-meaned matrix of X. Then we can calculate the r-th principal component v_r as follow.

$$\mathbf{v}_r = \arg\max_{\|\mathbf{x}\|=1} \left\| \left(\mathbf{Y} - \sum_{j=1}^{r-1} \mathbf{Y}\mathbf{v}_j\mathbf{v}_j^T \right)\mathbf{x} \right\|$$

The vector $x=(x_1,x_2,\dots,x_m)^T$ in the above equation satisfies $\|x\|=1$. Let $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$ represent the first r principal components. The original data X can be approximately represented by the low-dimensional matrix $\mathbf{P} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r]$ that captures dominant variance of X. On the contrary, the last $m - r$ components contain most random fluctuation.

2.3. Practical Issues

Sketch is commonly used to summarize network traces for network anomaly detection. It compresses high-volume traffic into a relatively small space, thus reducing computational complexity in anomaly detection. However, due to sketch, one has great difficulty in reversely deriving original keys from a number of anomalous buckets. In terms of DDoS reaction, much more memory space (storing mapping tables for each buckets) and computations is needed to identify victims of DDoS attacks.

PCA can highlight differences in data, such as low-profile traffic anomaly caused by DDoS attacks. The sensitivity to data dimensionality yet obstructs PCA from being applied to identify victims of DDoS attacks. The computational complexity and spacer equirement of PCA are $O(nm^2)$ and $O(nm)$ respectively. For the purpose of victim identification, IP-based traffic isolation is necessary. In high-speed network, numerous distinct IPs appear even in a short time interval, causing an extremely large m. In this context, PCA yields to the expensive computational cost of anomaly detection in such large-scale data.

In this paper, we use a piece of sketch to aggregate traffic, thus reducing the data dimensionality, so that anomalous aggregation can be found in reasonable time using PCA. To conquer the drawback of sketch and derive victims of DDoS attacks, we design a special family of hash functions, which can be used to compose of a sketch and make the reversing mission fairly easy and lightweight.

3. F-LOW OVERVIEW

F-LOW is a distributed framework for DDoS detection and victim identification. It consists of global detectors (GD) and local detectors (LD). LDs are wildly deployed all over the network. For simplicity, we consider a classic distributed architecture as shown in Figure 1, which is adopted in many literatures [6][7]. In the F-LOW system, GDs communicate with each other and LDs using reliable channels.

In general, LDs detect traffic anomaly based on measurements of local traffic, while the GD confirms whether reported anomalies arise from network-wide DDoS attacks, and infers victim IPs (or partial victim IPs) based on local detection results. Particularly, an LD performs

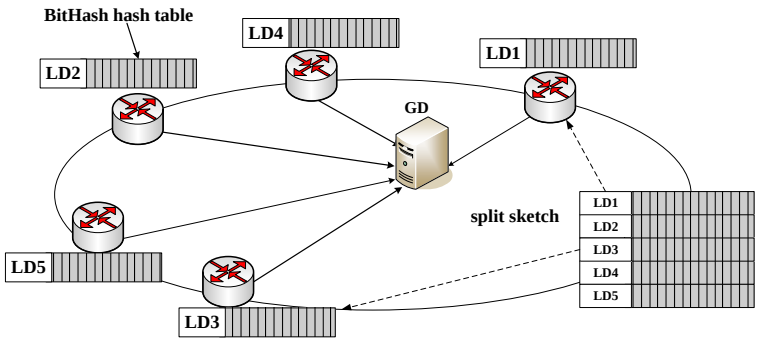

Figure 1. The architecture of FLOW system

a piece of a sketch, that is, a hash table of a special designed bitwise hash function; in this way the LD aggregates packets into bitwise flows according to certain bits in their destination IPs (dstIP). The aggregation significantly reduces the scale of time-series measurement data of

network traffic. LDs thus can detect when network anomalies occur and identify flows that involves these anomalies, using a PCA-based anomaly detection method. The method is like [6] and extended for identifying more than one anomalous flow. LDs send very little data to the GD including IDs of Anomalous Flows (AF). In view of the whole F-LOW system, all hash tables performed by LDs constitute a virtual sketch, called split sketch. The GD integrates flow IDs into split sketch and determines anomalies pervading all over the network as DDoS attacks. Taking advantage of bitwise hash functions, the GD can reconstruct (partial) victim IPs of DDoS attacks reversely with much lower computational cost and smaller space requirement than previous work [8][9][10].

There are three principles behind the F-LOW system, making it scalable and practical in network-wide DDoS detection and identification.

Principle 1: Flow aggregation based on bitwise information. Aggregating packets according to certain bits in their destination IPs, on one hand, reduces data dimensionality to yieldPCA-based method. On the other hand, since flow ID reifies part of victim IP, it benefits determination of victim IP.

Principle 2: Combination of distributed traffic monitoring and minimum data transmission. Considering the nature of DDoS attacks, F-LOW leverages pieces of a sketch on different LDs to gather network-wide measurements. Meanwhile, without sending original measurements, F-LOW greatly reduces the amount of transmitted data between nodes, respecting limited link bandwidth.

Principle 3: Lightweight reconstruction of victim IP based on popularity. As a profit from bitwise hash functions, F-LOW provides a lightweight way to infer victim IPs of DDoS attacks, having no use of storing mapping tables and calculating intersection of large sets of IPs like reverse sketch [10]. Determining a bit in victim IPs relies on the popularity of the bit being reified as “0” or “1”, in accordance with the multi-source feature of DDoS attacks.

4. ANATOMY OF F-LOW

Our F-LOW system achieves desirable Four-LOW properties through four essential components: bitwise-based flow aggregation, split sketch structure, PCA-based local detection method, and novel IP reconstruction algorithm. In this section, we present details of these components.

4.1. Bitwise-based Flow Aggregation

In F-LOW, LDs aggregate packets into flows according to certain bits in their dstIPs rather than whole dstIPs, thus obtaining much fewer flows. The principle of bitwise-based flow aggregation is, specifying k bit positions in an L-length IP address (e.g. L=32 in IPv4), packets whose dstIPs have identical bits in the same positions are assigned to the same flow.

Let us define a function F(ip,mask), where ip is an IP address and mask indicates which k bits are specified. mask is viewed as an L-length bit string in which specified bits are set to“1”while others are set to “0”. F(ip,mask) combines selected bits in ip orderly and results in a k-length string. Let ip_1 and ip_2 be two different IPs. Then we assign packets that are destined for ip_1 and ip_2 into the same bitwise flow if and only if

$$F(ip_1, mask) = F(ip_2, mask)$$

To identify flows, F(ip,mask) is viewed as flow ID. Figure 2 illustrates an example of bitwise-based flow aggregation. IP_1 ~ IP_4 are different dstIPs of packets. Specifying five bit positions in an IP address, packets destined for IP_1 ~ IP_4 thus are divided into three flows, identified by "010011", "011010" and "110110", respectively. Given a fixed k, there are at most 2^k bitwise flows. LDs thus can achieve an affordable space overhead by varying k.

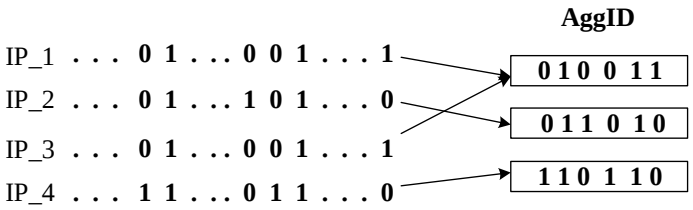

Figure 2. An example of bitwise-based flow aggregation

Considering functionality of bitwise-based flow aggregation, we then regard F(ip,mask) as a particular hash function, called bitwise hash function. Accordingly, mask is the seed. In the following, we explain how to use such hash function to generate a split sketch and show the remarkable benefits obtained from our novel design.

4.2. Split Sketch

In this section, we present the deployment of split sketch for network-wide traffic measurement. The split sketch is a virtual sketch structure that is divided into pieces and deployed on multiple LDs in F-LOW. In split sketch, bitwise hash functions are adopted instead of regular universal hash functions.

Imagine a sketch that consists of hash tables of N bitwise hash functions. Each hash table is a piece of the sketch puzzle and each LD only performs one piece. An LD generates its own bitwise hash function independently, by randomly choosing k integers from 0 to L-1. These integers correspond to selected bit positions in an IP address. Then the LD allocates memory space to keep a hash table of size2^k. Since our F-LOW system uses traffic volume as a feature to detect DDoS attacks, each bucket in the hash table is a counter for counting the number of packets that are mapped into the bucket. Although many other features can be adopted in our system, we choose the simplest but promising one, so as to simplify traffic measurement as well as reduce computational cost. When a packet arrives, the LD updates the bucket with index F(dstIP,mask) by adding one. Since each LD independently determines bitwise hash function, their choices may clash. However, the probability that two LDs select the same hash function is $(1/L)^k$. It is extremely small when k=10 and L=32. We thus say hash functions used in a split sketch are different.

Split sketch does not actually gather traffic measurements all over the network into a centralized device. In fact, each row of split sketch presents statistic of different traces. We thus let LDs detect network anomaly based on their own pieces of split sketch. On the other hand, local detection results (anomalous flow IDs) are integrated into a whole sketch and processed at GDs to confirm DDoS attacks and infer their victim IPs.

4.3. Extended PCA-based Local Detection

At LDs, we adopt PCA-based local detection method to detect potential DDoS attacks and identify abnormal traffic aggregates, which is similar with subspace method in [6]. Unfortunately,

subspace method in [6] only obtains the first AF that causes greatest anomalous change among all candidate flows. In practice, it is possible that more than one anomaly occurs simultaneously, legitimate or malicious. Furthermore, to reduce false negative, it is essential to capture as many AFs as possible. We thus extend identifying part in [6] to find a set of primary AFs.

Through a piece of split sketch performed by an LD, packets are naturally aggregated into bitwise flows. The LD obtains a time-series measurement of local traffic, forming an $n \times m$ matrix D, where n is the number of time intervals and m the size of hash table, $m = 2^k$.Applying PCA on **D** results in a low-dimensional matrix **P** that includes domain variance of **D**. Let **y** represent a row of D, which is viewed as the traffic fingerprint at a particular time. Like [6], we extract normal component y_n and abnormal component $\mathbf{y}_a$ of any **y** based on **P**.

$$\mathbf{y}_n = \mathbf{P}\mathbf{P}^{\mathbf{T}}\mathbf{y} = \mathbf{C}_n\mathbf{y}$$
$$\mathbf{y}_a = \left(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathbf{T}}\right)\mathbf{y} = \mathbf{C}_a\mathbf{y}$$

In general, the large change of $\mathbf{y}_a$ implies network anomaly. Thus an alert arises when the squared prediction error (SPE) of $\mathbf{y}_a$ exceeds a preset thresholdθ.

$$\text{SPE} \equiv \|\mathbf{y}_a\|^2 = \|\mathbf{C}_a\mathbf{y}\|^2 > \theta \quad (1)$$

Until now, we find an anomalous**y**, which corresponds to the time when the anomaly occurs or a DDoS attack starts. Next step is to identify AFs that are responsible for the detected anomaly.

We extend identifying part in [6] to find a set of primary AFs. The key principle is greedily choosing flows that cause greatest anomalous change in the residual ones. The first AF can be find as [6] does (assuming there is only one anomalous flow). The anomalous **y** can be represented by

$$\mathbf{y} = \mathbf{y}_i^* + \mathbf{\Delta}_i \quad (2)$$

where $\mathbf{y}_i^*$represents the sample vector for normal traffic conditionsand Δ_i representsthe amount of change due to flow F_i. See [6] for best estimates of Δ_i and $\mathbf{y}_i^*$. By minimizing the objection of $\mathbf{y}_i^*$ on abnormal space, we can obtain the first AF.

$$\text{AF}_1 = F_j, \quad j = \arg\min_i \|\mathbf{C}_a\mathbf{y}_i^*\| \quad (3)$$

To find the residual AFs, we eliminate the change caused by previous found AFs. Let **Ω** be the set of previous found AFs. Each flow in **Ω** corresponds to a four-tuple $\langle sn, F_i, \tilde{\mathbf{y}}_i^*, \tilde{\mathbf{\Delta}}_i \rangle$, where$sn$ represents the order of flow F_i being found. $\tilde{\mathbf{y}}_i^*$and $\tilde{\mathbf{\Delta}}_i$ are best estimate of $\mathbf{y}_i^*$ and $\mathbf{\Delta}_i$ and are calculated in the procedure to find F_i.(see details in [6]). Then we can construct a new traffic fingerprint$\boldsymbol{y}'$, getting rid of effects of AFs that have been verified to be anomalous.

$$\boldsymbol{y}' = \boldsymbol{y} - \sum_{\boldsymbol{F_i} \in \boldsymbol{\Omega}} \tilde{\boldsymbol{\Delta}}_{\boldsymbol{i}} \quad (4)$$

In terms of$\boldsymbol{y}'$, the next AF can be found using the same method as [6] does. The procedure is repeated until the change caused by any residual flow does not exceed a threshold.

$$\left\|\boldsymbol{y_n} - \tilde{\boldsymbol{\Delta}}_{\boldsymbol{i}}\right\| < \rho \quad (5)$$

In view of traffic increase that accompanies nearly all flooding-based DDoS attacks, we improve above local detection method in two greedy ways: 1) Further compress D to D'. The latter only consists of columns having increased amounts along the time axis in D. Accordingly, P is generated based on D'. 2) Measure the amount of packets that arrive in each time interval, and perform anomaly detection upon y, only if the increasing scale of packet amount in the corresponding interval exceeds certain threshold.

Local detection results in a set of AFs (or a set of flow IDs). According to the design of bitwise-based hash function, a flow ID reifies some bits in an IP address and mask indicates which bits they are. Namely, if an AF includes DDoS traffic, segments of victim IP of the DDoS attack is fixed by the flow ID. In the next subsection, we explain how the GD infers victim IPs based on such incomplete IP segments.

4.4. Alert-burst-based Global Detection

When a DDoS attack is launched, the malicious attack traffic accessing the attack target is relatively concentrated. From the point of view of distributed attack detection, the traffic abnormal alarm generated by each local detection device should also have time correlation. Therefore, when the global detection device finally determines the DDoS attack, it should consider the time when the abnormality occurs. If many abnormal traffic alarms occur in a short time, it can be considered that a large-scale DDoS attack has occurred in the network. As showed in Figure 3, a DDoS attack can be confirmed when LD0, LD1 and LD2 all reports local anomaly alerts nearly in the same time.

Assuming the number of abnormal alerts generated in an observation window T is N_{alert}, the abnormal alert density (DAA) is defined as the number of abnormal alarms generated in a unit time,

$$Dens(T) = \frac{N_{alert}}{T}$$

If the observation window t, the number of abnormal alarms generated by the local detection equipment LD is N_{alert}, and the abnormal alert density is $Dens(t)$ Expand the observation window along any direction of the time axis to obtain a new observation window t', let the abnormal alert density in t' be $Dens(t')$. If $Dens(t') < Dens(t)$ is always true, then the observation time t is called the abnormal burst period (BPA).

The determination process of abnormal burst period starts from the two abnormal alarms with the closest time interval. The initial observation window is set as τ and then slide the observation window to cover the above two abnormal alarms. Stretch the observation window along the two directions of the time axis then the first abnormal burst period of the local detection equipment can be obtained. As the network environment is complex and changeable, security threats occur all the time, each local detection device may have multiple abnormal outburst periods. The above operation is repeated until the initial observation window is greater than the threshold τ0. Threshold τ0 is the maximum relevant abnormal time interval, and 10 times the sampling interval is fine.

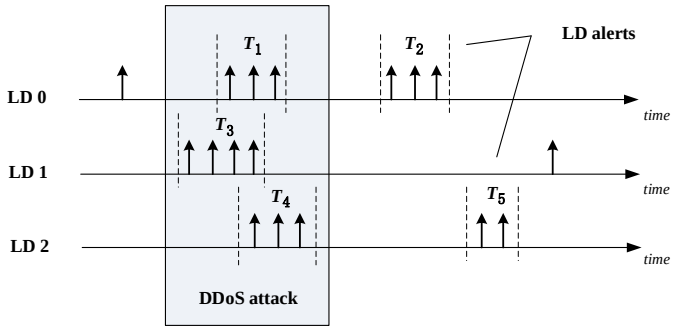

Figure 3. An outbreak period of local anomaly alerts

Based on the abnormal burst period of each local detection device, the global detection device finally determines DDoS attack in the network. DDoS attack sources are distributed all over the network, and the attack traffic triggers an abnormal alarm at multiple local detection devices. Therefore, a concentrated abnormal burst period of multiple local detection devices in a short time means a large-scale DDoS attack. By comparing the abnormal outbreak period of each local detection equipment, if the number of equipment whose abnormal burst period overlap in time is greater than a threshold n, then global detection confirm a DDoS attack in the network. As shown in Figure3, three local detection devices detect a large number of network traffic abnormalities at almost the same time (abnormal outburst periods T1, T3, T4), so it can be judged that there are large-scale DDoS attacks in the network. Local detection devices LD0 and LD2 also have anomaly alert burst in T2 and T5 respectively. However, since no global traffic anomaly is found by other detection devices in the network, these alerts do not conform to the distributed characteristics of DDoS attacks.

4.5. Victim IP Reconstruction

In order to confirm ongoing DDoS attacks and figure out victim IPs, GDs reconstruct anomalous dstIPs based on detection results obtained from LDs. In a DDoS attack, tens of thousands of hosts all over the network generate attack traffic simultaneously. The intuition thus is that a dstIP being widely regarded as anomalous indicates a network-wide DDoS attacks towards the dstIP. Following this lead, we show how GDs reconstruct victim IPs in details. Benefiting from bitwise hash function used in split sketch, our IP reconstruction algorithm has very low memory requirement and computational cost.

Let us begin with formatting outcomes of LDs. Through local detection, LDs obtain a number of AFs, which are sent to the GD in the form of (timestamp, flowID, mask). flowID is the ID of AF. timestamp represents the time when the anomaly occurs. mask implies the particular bitwise-based hash function adopted by an LD. If an LD report more than one AF to the GD, mask is only sent once. Combining flowID and mask, we can extract an L-length vector, in which bits specified by mask are set to “0” or “1”, according to flowID. We call the vector Discrete Segment of IP (DSIP). Figure 4 shows a simple example of DSIP which gives a partial segment of the destination IP address. L is the length of the IP address. If the IPv4 address is used in the network, L= 32.

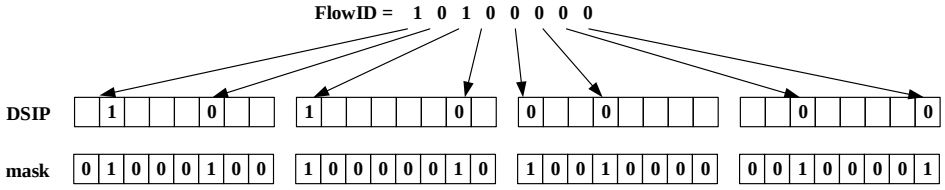

Figure 4. An example of DSIP

Regarding IP reconstruction as a vote for bits in victim IPs, then DSIP is the ballot. The GD first creates an empty vector B of length L to imitate an IP address (victim IP), and then deduces the IP bit-by-bit on the basis of amounts of DSIPs which set the bit to "0" and "1". IP reconstruction consists of three steps: eliminate noise DSIPs, decide bit statuses and reduce uncertainty. As a visual representation of IP reconstruction, we illustrate an example in Figure 5. In the example, DSIP3 is a noise DSIP and eliminated in the first step. After other two steps, a partial victim IP is obtained, in which two bits are undetermined. In the rest of this subsection, we present details of the three steps.

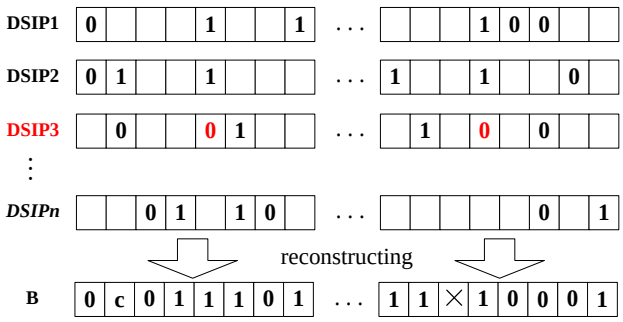

Figure 5. An example of IP reconstruction. Only the first and forth byte of an IP address are illustrated due to space limitation. Marks 0, 1 , c and × represent zero, one, amphibious and unknown statuses, respectively.

Eliminate noise DSIPs. We define DSIPs that comport with real victim IPs of DDoS attacks are majority DSIPs, and others are noise DSIPs. The latter DSIPs probably arise from some local network changes, rather than network-wide DDoS attacks. We thus consider a DSIP as a noise DSIP if it sets some bits just in contrast to what most other DSIPs do. Let Γ be the set of DSIPs received from LDs. For each bit in **B**, we count the numbers $q_{0,i}$ and $q_{1,i}$ of DSIPs that reify the bit as "0" and "1" respectively. The much smaller number between them implies corresponding bit ("0" or "1") is unpopular. Each DSIP reifies k different bits. If a DSIP reifies at least one unpopular bit, we consider it as a noise DSIP. We can obtain a more reliable set $\boldsymbol{\Gamma}'$ by eliminating noise DSIPs from Γ. Note that the input set Γ only includes DSIPs whose timestamps fall in the same time period (about several time intervals). Because network anomaly appears around the same time are more likely to be consequence of DDoS attacks.

Decide bit statuses. Based on the new set $\boldsymbol{\Gamma}'$ of DSIPs, we endow each bit in **B** with one of the following statuses: (1) zero: a great proportion of DSIPs reify the bit into "0"; (2)one: a great proportion of DSIPs reify the bit into "1"; (3) amphibious: the proportions of DSIPs that reify the bit into "0" and "1" both exceed a preset threshold, which may be caused more than one victim IPs; (4) unknown: the number of DSIPs reifying the bit is too small to determine the bit. Algorithm 1 shows the procedure of deciding bit statuses. The threshold T_q in Algorithm 1 varies

according to the number N of LDs. It is set to 1 when N is small to the extent that each bit in an IP address is probabilistically covered once by LDs.

Algorithm 1. Decide bit statuses

Input: $\mathbf{\Gamma}'$, T_q, α, $\beta\alpha > \beta$

Output: **B**

```
Initialize an empty vector B of length L.
Let b_i represent status of the i-th bit in B.
for i from 0 to L − 1
  calculate q_{0,i} and q_{1,i} based on Γ′
  if q_i = q_{0,i} + q_{1,i} < T_q
   set b_i to unknown
  else if q_{0,i}/q_i > α and q_{1,i}/q_i > α
set b_i to amphibious
        else if q_{1,i}/q_i < β
set b_i to zero
              else  set b_i to one
end for
```

Reduce uncertainty. To reduce the uncertainty in resultant victim IPs, we compare all DSIPs in $\mathbf{\Gamma}'$ with **B** and reify unknown and amphibious bits as many as possible. Let d_i and b_i represent the i-th bits in a DSIP and **B** , respectively. We say d_i matches with b_i under the followingconditions: (a) $d_i = 0$, $b_i = \text{zero}$; (b) $d_i = 1$, $b_i = \text{one}$; (c) $d_i = 0$ or $d_i = 1$, $b_i = \text{amphibious}$. If most reified bits in a DSIP match with **B**, we believe the DSIP is truely part of a victim IP. Other refied bits are unmatched only because no enough LDs select these bits in their hash functions. We thus can adjust statuses in **B** according to unmatched bits in the DSIP, reducing unknown bits. For amphibious bits, such mostly matched DSIPs help to decompose **B** intotwo or more vectors. Each vector corresponds to a distinct victim IP.

As a result, the GD obtains a number of victim IPs (or partial victim IPs), which also proves ongoing DDoS attacks. If there are enough LDs participating in the F-LOW system, the GD can identify whole victim IPs accurately. Even in lacking adequate LDs, partial victim IPs is still efficient as filtering rules in DDoS reaction. Our IP reconstruction algorithm infers victim IPs without storing mapping tables and calculating intersection of large sets, thus having low computational cost and memory requirement. We evaluate our algorithm and compare its overhead with previous work in the next section.

5. EVALUATION

In this section, we evaluate the performance of F-LOW system through simulation and theoretical analysis. The low-profile property of F-LOW is achieved through PCA-based anomaly detection method. Since the capability of the method has been thoroughly verified in many previous works [6] [7] [13], we lay particular emphasis on global detection of our approach. The focal point of evaluation is accuracy of IP reconstruction. We also measure the overhead of LDs and GDs, in order to demonstrate scalability of proposed approach.

5.1. Coverage Rate

We first measure an important metric, coverage rate of selected bits by LDs in an IP address, which is relevant to performance of IP reconstruction. IP reconstruction relies on flow IDs reported by LDs. Namely, GD deduces a victim IP based on LDs' reification on every bit of IP. Therefore, the more bits are selected by LDs, more precise the reconstructed victim IP is.

We define coverage rate as the proportion of bits in an IP address that are selected by LDs at least once. Each LD chooses k bits in an IP address to generate its own bitwise hash function. The probability of a bit being selected by an LD is k/L. Suppose N LDs exits in the network. Then, for a bit, it is not selected by any LDs with the probability $q = (1 - k/L)^N$.

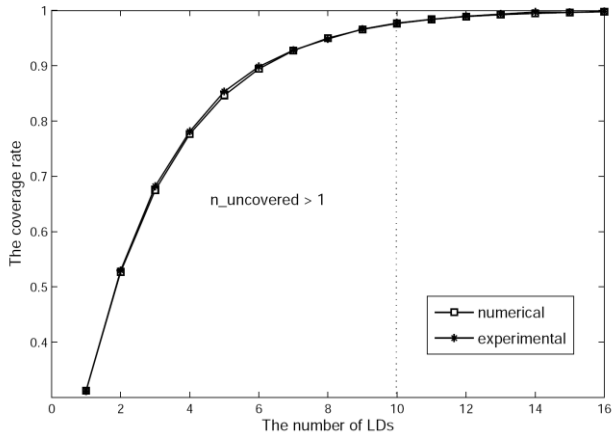

Figure 6. The numerical and experimental coverage rates with respect to the number of LDs

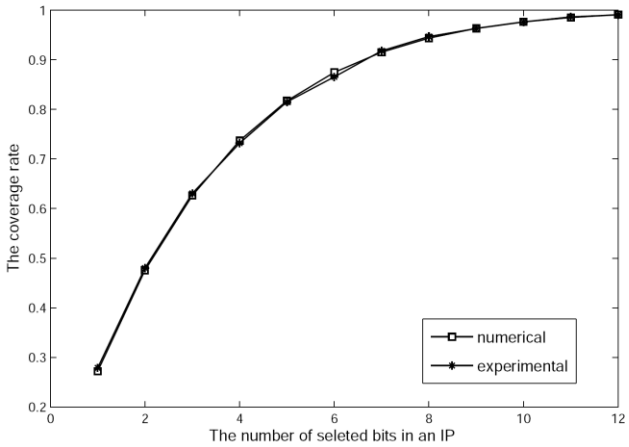

Figure 7. The numerical and experimental coverage rates with respect to the number of selected bits in an IP

Accordingly, the theoretical expected coverage rate is 1-q. We also calculate the coverage rate in the simulation. As we can see in Figure 4 and 5, the numerical and experimental coverage rates perfectly match with each other. When k=10, L=32, and N=10, the coverage rate is more than 97%. Uncovered bits in an IP address are no more than one. Namely, only 10 LDs participating in the F-LOW system can cover almost whole IP address, thus potentiating precise IP reconstruction.

5.2. Accuracy of IP reconstruction

To demonstrate the performance of our IP reconstruction algorithm, we first analyze the efficacy of noise DSIP elimination. In the simulation, we use settled $L = 32$ and $k = 10$, and vary N from 5 to 25. In order to see how the accuracy of local detection influences victim IP identification, we purposely increase false positive of local detections, varying the following proportion ω from 0.2 up to 5.

$$\omega = \frac{\text{the number of noise DSIPs}}{\text{the number of mojority DSIPs}}$$

The simulation result in Figure 8 leads us to two important conclusions. First, the accuracy of noise DSIP elimination increases with N. This suggests us to deploy LDs widely for better performance. Second, when N exceeds a certain threshold like 20, our algorithm maintains a high accuracy, even though there are five times noise DSIPs in the result of local detection. That means our approach is robust to false positive of PCA-based local detection. Therefore, adjusting parameters of local detection to report as many anomalous flows as possible, our F-LOW system can capture low-profile DDoS attacks.

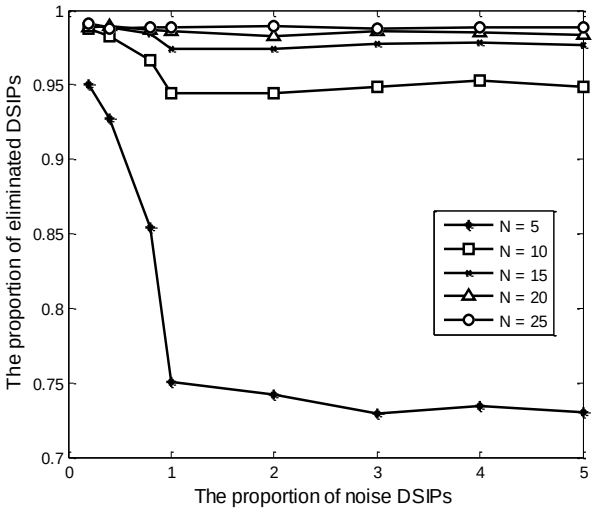

Figure 8. The accuracy of minority elimination

Since our approach may result in partial victim IPs, it is difficult to evaluate its accuracy in DDoS detection and victim identification. We thus incarnate the performance in an indirect but persuasive way. That is to evaluate the efficacy of outcomes of our F-LOW system in filtering traffic towards victims (distinguishing attack packets from normal packets towards victims is beyond the scope of this paper).

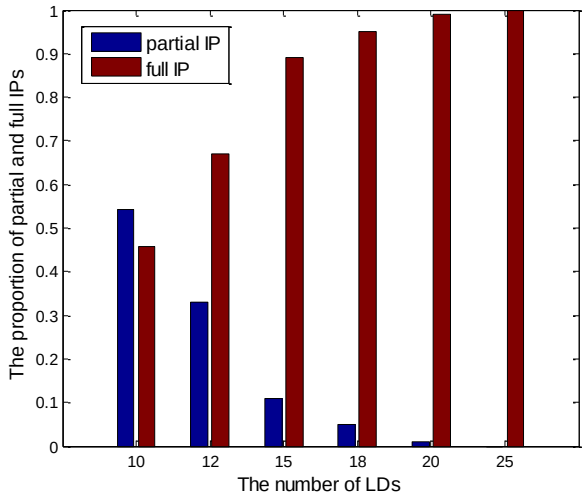

Figure 9. The comparison between partial and whole IPs

Figure 9 shows the proportions of partial IPs and whole IPs in the identification result. As N increases, the probability that our approach extracts whole victim IPs increases rapidly. Interestingly, most of partial victim IPs only has one indeterminate bit. Using them as rules, we can still successfully filter attack packets of DDoS attacks, with a false positive rate of less than 3%. We show the simulation result in Figure 10. In our simulation, all traffic toward victims can be captured as their destination IPs still matches with partial IPs.

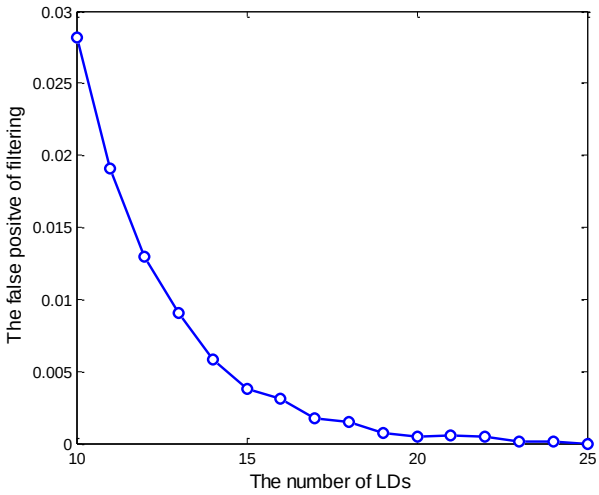

Figure 10. The false positive of filtering

5.3. Accuracy of DDoS Detection

We compare our F-LOW system with three existing approaches with respect to system overhead: sketch subspace (SS) [7], two-level sketch (TS) [8], and reverse sketch (RS) [10], which are closely related to our approach. These three methods all take the identification of the IP address (DDoS attack victim) causing the network anomaly as the final target of detection, and all reduce the statistical data dimension with the help of the summary data structure (sketch).

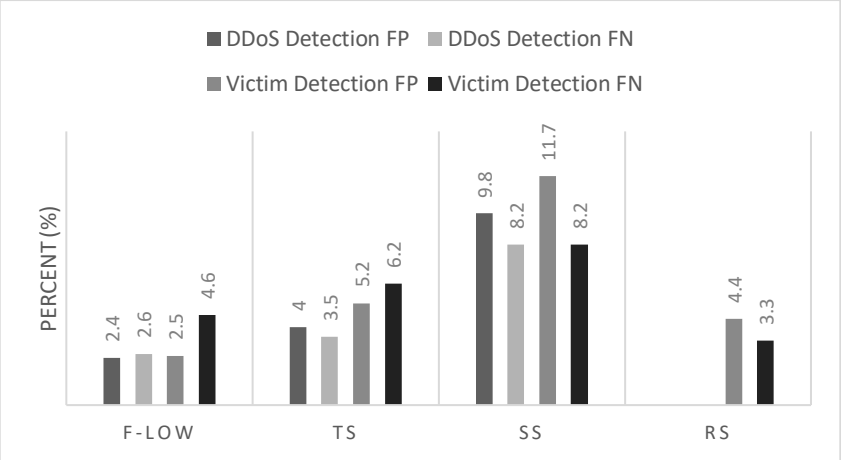

Figure 11. Accuracy of DDoS Detection

Figure 11 shows the experimental statistical results. Where FP represents false positive rate and FN represents false negative rate. Since RS is only a method of inversely deriving the IP address corresponding to the abnormal sketch entry, and there is no specific anomaly detection process, RS is used to replace bit hash based sketch in F-LOW system to compare the accuracy of RS and F-LOW in victim identification. In addition, the SS method is a single point detection algorithm, 10 points are selected in the experimental topology to deploy SS. As shown in Figure 11, F-LOW performs better than other methods in all aspects. The victim identification FN of F-LOW system is slightly higher than that of RS method because only the number of complete IP addresses is calculated when the experimental data are counted.

5.4. Overhead Comparison

Finally, we qualitatively compare our F-LOW system with three existing approaches [7][8][10], to demonstrate the advantages of our approach in computational complexity, space requirement and Communication overhead.

When comparing computational complexity, we consider two aspects: the cost to update corresponding structures, which can be approximately reflected through the Number of calculating Hash Value per Packet (NHVP); the cost to identify victim IPs after detecting anomaly (COIV). We leave the overhead of PCA for this moment, as three of four approaches adopt PCA to detect network anomaly. Since all compared approach use sketch, we thus separate space requirement for sketch structure (sketch size) with other storage in the comparison. Communication overhead, represented by the amount of transmitted data between nodes, exists only in collaborative systems. Thus, reverse sketch has no communication overhead.

In F-LOW, each collaborative node maintains only one hash table of size M. One of the major differences between our approach with the other three is the way to map packets in a sketch. Our approach uses special designed bitwise-based hash while others use random hash in sketches, which induces completely different method to identify victims. Most COIV of our approach arises from counting DSIPs and comparing every DSIP with the vector B, so the cost increases with the number n_{dsip} of received DSIPs proportionally.

Table 1. The overhead comparison.

Method	Comput. Complexity		Space Requirement		Commun.
	NHVP	*COIV*	sketch size	other storage	Overhead
F-LOW	1	$O(Ln_{dsip})$	M	0	$O(n_{dsip})$
SS [7]	$4H$	$O(n_{flow})$	$4HM$	$O(n_{flow})$	$O(HM)$
TS [8]	$H + H_{bc} + H_{bf}$	$O(1)$	$HM + H_{bc}M_{bc}$	$H_{bc}L_{bf}$	$O(n_{dip})$
RS [10]	H	$qHM(\frac{n}{M})^{\frac{1}{q}}$	HM	$H \log mn^{\frac{1}{q}}$	0

We summarize the comparison result in Table 1, in which n_{flow} represents the number of distinct flows that SS needs to store states, and n_{dip} represents the number of anomalous destination IPs.H is the number of hash functions in a sketch. L is the length of an IP address. The parameter q = 4 in [10]. L_{bf},H_{bf},$H_{bc}$and M_{bc} are defined in [8].Considering the number of DDoS attacks or victim IPs, the number $n_{dsip}$is much smaller than M and n_{flow}. In this context, our F-LOW system outperforms SS in all three aspects. Relatively, TS has lowest cost O(1) during victim identification at the expense of space. Even though approaches TS and RS are slightly better than F-LOW in some aspects, they all require much more memory to store statistic of network traffic.

6. CONCLUSION

In this paper, we study countermeasures against DDoS attacks and present F-LOW, a collaborative system that can detect DDoS attacks and identify victim IPs of these attacks. Innovative bitwise-based hash function and split sketch are designed to digest network traffic. Meanwhile, an efficient IP reconstruction is proposed to reversely calculate anomaly IPs. Benefitting from those innovations, F-LOW system can accurately detect network anomaly caused by DDoS attacks and infer victim IPs, thus contributing to quick and efficient DDoS reaction. In summary, F-LOW has Four LOW-characteristics: low profile, low dimensional, low overhead, and low transmission, while previous DDoS countermeasures only fit parts of them. The outcome of F-LOW can help to filter attack traffic with very low false positive. This paper proposes a promising approach against DDoS attacks. Since the F-LOW system is a collaborative system which involves plenty of cooperative detection nodes in the network, it is very difficult to apply and deploy. In the future, a flexible cooperation mechanism should be put on the agenda, to provide a practicable collaboration platform for F-LOW system.

REFERENCES

[1] Biswas, R., Kim, S., & Wu, J.,(2021)"Sampling rate distribution for flow monitoring and DDoS detection in datacenter", *IEEE Transactions on Information Forensics and Security*, Vol. 16, pp2524-2534.

[2] Toh, A.,(2022) "Azure DDoS Protection—2021 Q3 and Q4 DDoS attack trends".

[3] Wang, F., Wang, H., Wang, X., & Su, J.,(2012) "A new multistage approach to detect subtle DDoS attacks", *Mathematical and Computer Modelling*, Vol. 55, No. 1-2, pp198-213.

[4] Wang, H., Zhang, D., & Shin, K. G,(2002, June)"Detecting SYN flooding attacks"Proceedings,*Twenty-first annual joint conference of the IEEE computer and communications societies*, Vol. 3, pp1530-1539.

[5] Chen, Y., Hwang, K., & Ku, W. S., (2007) "Collaborative detection of DDoS attacks over multiple network domains", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 12, pp1649-1662.

[6] Lakhina, A., Crovella, M., & Diot, C., (2004) "Diagnosing network-wide traffic anomalies", *ACM SIGCOMM computer communication review*, Vol. 34, No.4, pp219-230.

[7] Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G., & Lakhina, A., (2006, October) "Detection and identification of network anomalies using sketch subspaces",*In Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pp147-152.

[8] Liu, H., Sun, Y., & Kim, M. S., (2011) "A Scalable DDoS Detection Framework with Victim Pinpoint Capability, *Journal of Communications*,Vol. 6, No. 9, pp660-670.

[9] Salem, O., Vaton, S., & Gravey, A., (2010) "A scalable, efficient and informative approach for anomaly‐based intrusion detection systems: theory and practice", *International Journal of Network Management*, Vol. 20, No. 5, pp271-293.

[10] Schweller, R., Gupta, A., Parsons, E., & Chen, Y., (2004, October) "Reversible sketches for efficient and accurate change detection over network data streams", *In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp207-212.

[11] Cormode, G., & Muthukrishnan, S., (2005) "An improved data stream summary: the count-min sketch and its applications", *Journal of Algorithms*, Vol. 55, No. 1, pp58-75.

[12] Smith, L. I., (2002) "A tutorial on principal components analysis",*Cornell University*.

[13] Huang, L., Nguyen, X., Garofalakis, M., Jordan, M., Joseph, A., & Taft, N., (2006) "In-network PCA and anomaly detection", *Advances in neural information processing systems*, Vol. 19.

[14] Wang, F., Wang, X., Hu, X., & Su, J., (2012, October) "Bitwise sketch for lightweight reverse IP reconstruction in network anomaly detection",*In 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*,pp1-4.

[15] Cui, Y., Qian, Q., Guo, C., Shen, G., Tian, Y., Xing, H., & Yan, L., (2021) "Towards DDoS detection mechanisms in software-defined networking", *Journal of Network and Computer Applications*, Vol. 190, pp103156.

[16] Wytrębowicz, J., (2018, October) "Software-defined anti-DDoS: Is it the next step?",*In Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018*,Vol. 10808, pp652-663.

AUTHORS

Fei Wang received the PhD. degree from the National University of Defense Technology, China. She is now an associated professor in School of Computer, National University of Defense Technology. Her current research interests are next generation networks, network security and network monitoring and forensics.

Zhenxing Li received the master's degree from Guilin University Of Electronic Technology, China, He is now an R&D Engineer in School of Computer, National University of Defense Technology. His Current research interests are Network Traffic Analysis and Network Security

Xiaofeng Wang is an assistant professor in School of Computer, National University of Defense Technology (NUDT), China. He completed his PhD at NUDT in 2009. His current research interests are in trust and security of networking systems, distributed and intelligent data processing. He has published several papers in renowned journals and conferences like IEEE/ACM CCGrid, AINA, IEEE Transactions on Services Computing and Elsevier FGCS etc.