Increasing Accuracy and Reliability of IP Traceback for DDoS Attack Using Completion Condition

Samant Saurabh and Ashok Singh Sairam
(Corresponding author: Samant Saurabh)

Department of Computer Science, Indian Institute of Technology
Patliputra Colony, Patna, Bihar 800013, India.
(Email: {ssaurabh, ashok}@iitp.ac.in)

(Received Apr. 2, 2015; revised and accepted July 4, 2015)

Abstract

Probabilistic Packet Marking (*PPM*) is one of the most promising schemes for performing IP Traceback. PPM reconstructs the attack graph in order to trace back to the attackers. Finding the Completion Condition Number (i.e. precise number of packets required to complete the traceback) is very important. Without a proper completion-condition, we might reconstruct a wrong attack-graph and attackers can evade detection. One presently being used works only for a single attacker based *DoS* attack and has an accuracy of just around 70%. We propose a new Completion Condition Number which has an accuracy of 95% and it works even for the multiple attacker based DDoS attacks. We confirm the results using detailed theoretical analysis and extensive simulation work. To the best of our knowledge, we are the first to apply the concept of Completion Condition Number to increase the reliability of IP Traceback for the *DDoS* attacks.

Keywords: Completion condition, coupon collector problem, DDoS attack, IP traceback, probabilistic packet marking

1 Introduction

In the recent years, the Internet world has seen an alarming increase in what we call Denial and Distributed denial of service attacks (Dos/DDoS). Dos/DDoS attacks are major threat to Internet today [24, 25, 30]. Even the fast emerging cloud infrastructure is at great risk due to the highly distributed *DDoS* attacks [9, 14]. They are possible due to IP spoofing and destination based routing. A number of approaches to mitigate these attacks have been suggested and probabilistic packet marking [1, 11, 22, 27] is one of the most promising among them.

In PPM, each router in the path marks the packet with probability $p < 1$ and the marks can be over-written by routers down the path from attacker to victim. Hence if a router is d hops away from the victim, the probability that packet mark from it reaches the victim is given by $p(1-p)^{d-1}$ [10, 26]. In PPM, the edge information is encoded in the packet mark. It is of the form (start, end, distance) where start and end are the IP addresses of the routers connected to the given edge [3, 12] and distance is the hop distance of the first router from the victim as shown in Figure 1.

To reconstruct the attack-path in order to trace the attacker, victim must collect sufficient number of packets so that she can get marks from all the routers in the path [18, 36]. We call the number of packets required to reconstruct the correct attack path as *Completion Condition Number (CCN)* [24, 35]. It turns out that *CCN* is very important because without correct completion condition number, victim might not perform successful traceback. The constructed path from victim to the attacker would remain incomplete as shown in Figure 1. This defeats the very purpose of traceback as we are not able to correctly identify the attacker.

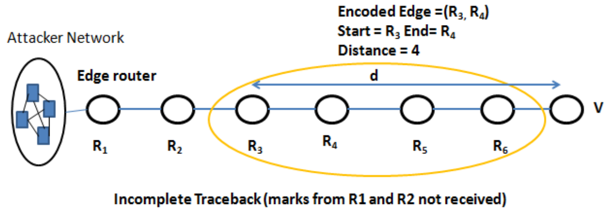

Figure 1: Probabilistic packet marking (PPM) algorithm - Each router in the path marks the packet with probability p and encode edge information (start, end, distance) in it. IP traceback might remain incomplete if the completion condition number is small.

We found that Completion Condition Number for IP traceback has not been extensively investigated in the present literature. The *CCN* presently being used is calculated based on mean of the number of packet marks required to complete the IP traceback [26, 31]. In our work, we show that the present *CCN* does not give correct results for 30 percent of cases. Computation of present *CCN* just takes expected number of packets into account. However, we show that distribution of *CCN* has got high variance as shown in Figure 3. Hence, if we ignore the high variance of the distribution of X, it can lead to inaccurate results.

Next, we argue that even if we have achieved a very precise Completion Condition Number for a single attacker based on a linear network, this would not work for multiple attackers scenario or a *DDoS* attack. As shown in Figure 2, attack graph is a tree with victim at the root of the tree and the attackers at the leaf nodes. Paths from the attackers to the victim have a lot of edges or routers in common as can be seen from Figure 2. For example, edges (R_2, R_1) is shared by paths from attackers R_8, R_9, R_{10} and R_{11} to the victim. These 4 paths share edge (R_2, R_1).

We know that Completion Condition Number improves the reliability and attacker-detection capability of *PPM*. However, in order to apply *CCN* in case of multiple attackers, we must know how many packets are contributed by each path individually. To find these numbers, we must separate the contribution of different paths for the routers that are shared in multiple paths. We propose a simple but elegant algorithm which finds the number of packets generated by each path separately in the attack graph. It distributes packets to their respective paths for the shared edges in the attack graph. Then using the Completion Condition Number, we can find if the *traceback* for these individual paths have been completed. Some of the important advantages of our proposed algorithm are

- It does not require any a priori knowledge of the attack graph.
- It can work for any attack graph and for any rate at which different attackers might be sending packets.

Tracing attackers in flooding based *DDoS* attacks becomes even more important because *DDoS* attack is a big threat for cloud computing [15] and number of well orchestrated *DDoS* attacks being conducted by botnets is increasing manifold each day [33].

We perform extensive simulation to evaluate the performance of our proposed work and algorithms. The results show that our proposed Completion Condition Number gives correct results for 98 percent of cases as compared to the present *CCN* which has a success rate of just around 70 percent and which cannot work for *DDoS* attack. The algorithm proposed to use *CCN* in case of *DDoS* attack also has a success rate of 95%.

In Section 2, we present a literature review of the related work and show why our work can improve the reliability and effectiveness of present *PPM* considerably. In

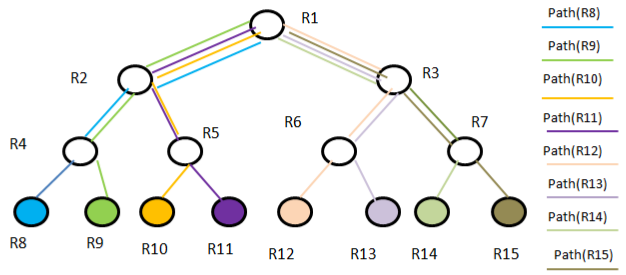

Figure 2: Shared edge or shared router problem for multiple attacker scenario or DDoS attack. This prevents us from using *CCN* in improving the reliability and accuracy of IP traceback in multiple attacker case for *PPM*.

Section 3, we outline the problem statement. We state the two major problems that we are going to solve in this paper. In Section 4, we derive a new completion condition number and theoretically prove the accuracy of this new CCN. In Section 5, we show how we can use CCN for multiple attacker case as well. In Section 6, we prove the utility and effectiveness of our work by means of rigorous simulation work. In Section 7, we conclude our work with some remark about the future work.

2 Related Work

As discussed in Section 1, the number of packets required to complete the traceback, called *CCN* is critical for increasing the reliability and accuracy of IP traceback using *PPM*. In this section, we perform a literature survey of *CCN* in *PPM* based IP *traceback*. In [17, 26], an estimate for number of packet-marks required to complete IP traceback for a given path length d and marking probability p was proposed. Let X be the number of marks required to complete the traceback. Then $E[X] < \frac{ln(d)}{p(1-p)^{d-1}}$ [23]. However, this number does not give accurate results for $25 - 30\%$ of cases because it does not take the high variance of distribution of X into account. Moreover, we cannot directly apply this estimate for multiple attacker scenario due to the problem of shared path that is explained in Figure 2 and Section 5.

In this section, we enumerate some of the major IP traceback mechanisms that use *PPM* to perform traceback. In Advanced and Authenticated marking scheme [29], they add authentication procedure to prevent problem of spoofed mark from the attackers and greatly enhance the performance of *PPM*. Similarly, in Dynamic probabilistic packet marking for efficient IP traceback [19], the marking probability is made dynamic and it depends upon the distance of the marking router. This too helps in in thwarting the spoofed marking problem of PPM and is more light-weight compared to [29]. To reduce the number of participating routers, in practical and robust inter-domain marking scheme for IP traceback [8], traceback is performed on the basis of autonomous systems (AS) instead of routers. This approach reduces the number of packets required for traceback as number of AS in a path is much less than the number of routers. Moreover, AS path are more stable. Still these methods lacked scalability. In Probabilistic Packet Marking for Large-Scale IP Traceback [10], they use randomize and link method along with check-sum chord to make IP traceback highly scalable.

However, none of the above mentioned schemes make explicit use of CCN to make IP traceback more reliable. Work by Wong [34, 35] was the first one to explicitly use CCN for increasing the reliability of IP *traceback*. Instead of finding the CCN for a single path, it finds the CCN for the complete attack graph treating it as a single unit. However, we observe that this approach has the following drawbacks:

1) **A-priori knowledge of attack-graph:** To compute the CCN, it needs to have a-prior knowledge of the complete attack graph. This defeats the very purpose of IP traceback because if we know the attack graph, then there is no need for performing IP traceback.

2) **Same Rate of packets from attackers:** Second, it assumes that all the attackers are sending traffic at the same rate. This assumption is very difficult to be true in the Internet due to its distributed nature.

3) **Bottleneck long paths:** Third, this paper finds the CCN for the complete attack graph as a single unit. This implies that IP traceback would start only when we have collected marks from all the routers in the attack graph. However, we can complete traceback for shorter paths much faster than those for longer paths. Hence, if we wait for longer paths, then starting of traceback for shorter path might get delayed unnecessarily. A few long paths might become the bottleneck in tracing to the attackers.

In our work, we decouple all the paths from the attacker to the victim and perform traceback individually for each path. This makes IP traceback much faster. In our work, we do not need any prior information or knowledge of the attacker graph. Moreover, our algorithm does not require the constraint of all the attackers sending at the same rate. We treat all attackers' path separately and start traceback for each of them individually and independently after victim has received completion condition number of packets for that particular path.

3 Problem Statement

In this section, we identify the two problems of our work. We first find a more accurate Completion Condition Number which makes PPM more reliable. Next, to use the concept of CCN in case of multiple attackers, we solve the problem of shared edges in the attack graph.

3.1 A More Accurate Completion Condition Number for a Single Attacker

Savage [26] in his preliminary paper on PPM had provided an estimation of the number of packets required before the victim can have a constructed graph that is the same as the attack graph under a single-attacker environment or a linear network. Let X represent the number of packets required to reconstruct the complete path from attacker to the victim. Let p be the marking probability and d be the number of routers between the attacker and the victim. In this scenario expected value of X is bounded by

$$E[X] < \frac{\ln(d)}{p(1-p)^{d-1}}. \tag{1}$$

. We will call this number the Savage Completion Condition Number ($SCCN$).

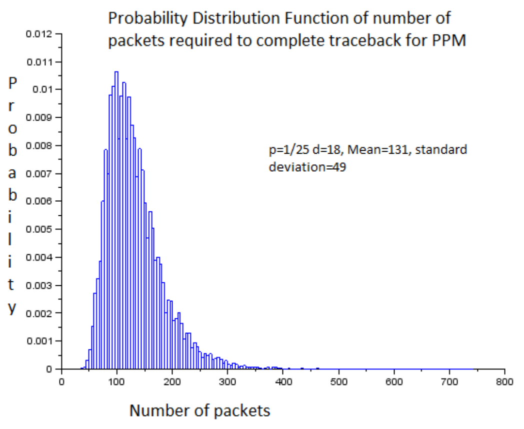

Figure 3: Probability Distribution Function for Number of Packets Required to Complete Traceback for $p = \frac{1}{25}$ and $d = 18$. We can observe the high variance of X from this figure.

The problem with using Equation (1) as completion condition is that on average around $25-30\%$ of cases, we cannot reconstruct the correct attack-graph within these number of packets and in worst case this can even increase upto 37% of cases as shown in Figure 3. The problem with Equation (1) is that it ignores the high variance of the distribution of X while calculating CCN. In our work, we calculate upper bound of variance of X and incorporate it in calculation of the completion condition number to make IP traceback algorithm more reliable and correct.

3.2 Shared Path Problem in Case of Multiple Attackers DDoS Attack

Even if we have calculated a very precise CCN for a single attacker case [13], there are some issues left in using completion condition number in making IP traceback more reliable for multiple attacker case. We consider the problem of shared edges as shown in Figure 2.

Let us assume that routers $R8$ to $R15$ are all attackers and we want to perform IP traceback for all these nodes. However, path $PATH1 = \{R8 - R4 - R2 - R1\}$ and path $PATH2 = \{R9 - R4 - R2 - R1\}$ have two shared edges $(R4 - R2)$ and $(R2 - R1)$. Similarly path $\{R8 - R4 - R2 - R1\}$ and $\{R10 - R5 - R2 - R1\}$ have a shared edge $(R2 - R1)$.

Hence, from this example we can observe that different paths from attackers to the victim would be sharing a lot of common edges in case of $DDoS$ attack because all these paths would be converging at the victim. However to use CCN, we need to know the number of packets received from each of these different paths individually. In our example of Figure 2, let's say that the victim collects 100 marks for edge $(R2 - R1)$, 50 for $(R4 - R2)$, 10 for $(R8 - R4)$ and 15 for $(R9 - R4)$. Then we need to separate the number of packets obtained for $PATH1$ and $PATH2$ from the shared edges otherwise we cannot apply the concept of CCN to find if traceback is complete for these paths. We need to find that out of 50 packets marks received for edge $R4 - R2$, how many of them belong to $PATH1$ (originating from router R_8) and how many belong to $PATH2$ and perform the same calculation for all other shared edges.

4 Completion Condition Number

As discussed in the previous section, we need to incorporate second moment of X in calculation of completion condition number to make it more accurate. Hence, like expectation, we calculate an upper bound on the variance of X and include it in out computation of the completion condition number.

4.1 Upper Bound on Expectation and Variance

From Equation (1), we know that if a router is d hops away from the victim, the probability that it marks the packet is given by $p(1-p)^{d-1}$. From this equation, it is clear that farther a router is from the victim, lower is the probability of its marked packet reaching the victim. Let $(p_1, p_2, p_3, \cdots, p_n)$ be the probability vector of en-route routers marking the packet, listed from the attacker to the victim side where the hop distance between the attacker and the victim is n. This implies that $(p_1 < p_2 < p_3 < \cdots p_n)$.

Let X be the random variable that represent the number of packets required to complete IP traceback. In this section we derive the upper bounds on expectation $E[X]$ and variance $Var[X]$. Without loss of generality, we can assume that on average, we will first receive packet-mark from the router nearest to the victim, then from hop distance two, then three and so on. On an average, we will get the mark from the farthest router in the end. Let t_i be a random variable that represents the average time required to collect mark from the i_{th} router after having collected marks from first $(i-1)$ routers. It can be seen that t_i is geometrically distributed [2]. For geometric distribution X with probability of success p, $E[X] = \frac{1}{p}$ and $VAR[X] = \frac{1-p}{p^2}$ [7].

Now for collecting the first mark, which can be any of n routers, probability of success is $p_1 + p_2 + p_3 + \cdots + p_n$. Then after receiving the first mark, we will wait to get the second mark. The probability of getting the second mark after receiving the first mark would be $p_1 + p_2 + p_3 + \cdots + p_{n-2} + p_{n-1}$. As we have already received the first mark, p_n is subtracted from probability of receiving second mark. This is because router nearest to the victim has the highest probability of generating the first mark. In a similar fashion, probability of receiving the third mark would be $p_1 + p_2 + p_3 + \cdots + p_{n-2}$ and that of last packet mark would be p_1. As there is always a chance of getting different packet-marks of lower probability also during these inter-arrival time t_i, hence, from linearity of expectation, $E[X]$ can be upper bounded by

$$\begin{aligned}
E[X] &< E(t_1) + E(t_2) + E(t_3) + \cdots + \\
&\quad + \cdots + E(t_{n-2}) + E(t_{n-1}) + E(t_n) \\
&< \frac{1}{p_1 + p_2 + \cdots + p_n} + \frac{1}{p_1 + p_2 + \cdots + p_{n-1}} \\
&\quad + \cdots + \frac{1}{p_1 + p_2 + p_3} + \frac{1}{p_1 + p_2} + \frac{1}{p_1} \\
&< \frac{1}{np_1} + \frac{1}{(n-1)p_1} + \frac{1}{(n-2)p_1} + \cdots + \\
&\quad + \frac{1}{3p_1} + \frac{1}{2p_1} + \frac{1}{p_1} \\
&< \frac{1}{p_1}\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right) \\
&= \frac{\ln(n)}{p(1-p)^{n-1}}
\end{aligned}$$

Similarly, from the linearity of variance, $VAR[X]$ can be upper bounded as

$$\begin{aligned}
VAR[X] &< VAR(t_1) + VAR(t_2) + VAR(t_3) + \cdots + \\
&\quad + \cdots + VAR(t_{n-1}) + VAR(t_n) \\
&< \frac{1 - (p_1 + p_2 + \cdots + p_n)}{(p_1 + p_2 + \cdots + p_n)^2} \\
&\quad + \frac{1 - (p_1 + p_2 + \cdots + p_{n-1})}{(p_1 + p_2 + \cdots + p_{n-1})^2} \\
&\quad + \cdots + \frac{1 - (p_1 + p_2)}{(p_1 + p_2)^2} + \frac{1 - p_1}{(p_1)^2} \\
&= \sum_{i=1}^{n}\left(\frac{1 - \sum_{k=1}^{i} p_k}{(\sum_{k=1}^{i} p_k)^2}\right)
\end{aligned}$$

$$\Rightarrow \sigma[X] < \sqrt{\sum_{i=1}^{n}\left(\frac{1 - \sum_{k=1}^{i} p_k}{(\sum_{k=1}^{i} p_k)^2}\right)}$$

Now we want to find CCN based on $E[X]$ and $\sigma[X]$ calculated above. We want our CCN to be of the form $E[X] + k\cdot\ \sigma[X]$. We need to find the value of k for which the probability of performing incomplete traceback after receiving CCN number of packets is negligible. We find the value of k by finding the tail estimate of the distribution of X [28]. Basically, we would like to find starting of the thin tail of distribution of X. After this point X has very little probability of occurrence.

4.2 Tail Estimate of Completion Condition Number

Let Z_k^r denote the probability that we do not receive any mark from the router that is k hops away from the victim after the victim receiving r packets. Probability of not receiving mark from the router at hop distance k after receiving a single packet (one attempt) would be given as $1 - p(1-p)^{k-1}$.

$$Z_k^r = (1 - (p(1-p)^{k-1}))^r \tag{2}$$

Probability that CCN would be greater than r packets would be the union of not receiving packet mark from any of the n routers in the path. It is calculated as follows

$$\begin{aligned} P(CCN > r) &= P(\cup_{i=1}^n z_i^r > a) \\ &\le (P(z_1^r) + P(z_2^r) + \cdots + P(z_n^r)) \\ &\le nP(z_n^r) \\ &\le n(1 - (p(1-p)^{n-1}))^r \end{aligned} \tag{3}$$

where $r = E[X] + k \cdot \sigma[X]$. For CCN to be 95% accurate, $P(CCN > r) \ge 0.95$. Now in the above equation, we already know p and n, hence we can find r from it. Next, when we know r, we can calculate k because $r = E[X] + k \cdot \sigma[X]$ and we know $E[X]$ and $\sigma[X]$ for given p and n. Therefore, we can compute CCN.

Therefore, given distance of attacker from the victim d, marking probability p and number of packets collected r, we can find if IP traceback has been complete for a given path by comparing the number of packets received for the given path with the completion condition number $CCN = E[X] + k \cdot \sigma[X]$ calculated above. In our result section, we show that value of $k = 3$ gives us an accuracy of almost 98%.

5 Shared Path

In Section 3, we derived a more accurate completion condition number for IP Traceback for a linear network. We argued that we need to take the high variance of the distribution of number of packets that are required for IP traceback into account. However, we cannot use the CCN calculated for a single attacker case to improve the correctness and reliability of PPM in case of multiple attackers because of the problem of shared edges which is shown in Figure 2. As paths from different attackers to the victim share edges or routers, we cannot find the number of packet-marks generated by each path individually. Therefore, we cannot apply CCN to find if traceback is complete for those paths.

In Figure 4, let us assume that routers $R8$ to $R15$ are all attackers and we want to perform IP traceback to all these nodes. However, path $PATH1 = \{R8 - R4 - R2 - R1\}$, $PATH2 = \{R9 - R4 - R2 - R1\}$, $PATH3 = \{R10 - R5 - R2 - R1\}$ and $PATH4 = \{R11 - R5 - R2 - R1\}$ have a one shared edge $(R2 - R1)$. Similarly, $PATH1$ and $PATH2$ have two shared edges $(R4 - R2)$ and $(R2 - R1)$. However, to use completion condition number equation, we need to know the number of packets received from each of these different paths individually.

Let us assume that the victim collected 100 packet-marks for edge $(R2 - R1)$. Then, we need to find, that out of the 100 packet-marks received for edge $(R2 - R1)$, how many of them are generated by $R8(PATH1)$, $R9(PATH2)$, $R10(PATH3)$ and $R11(PATH4)$. Then only we can use CCN to find the completion of traceback for these paths. Shared edge problem exists due to the convergence of these paths towards the victim.

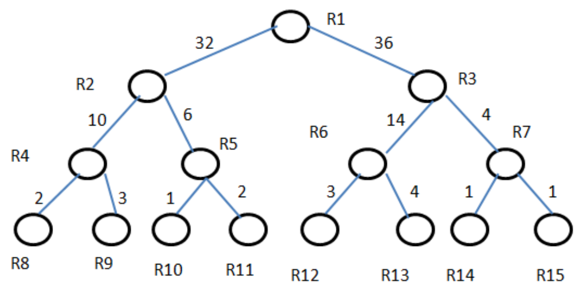

Figure 4: Problem of shared Path in IP Traceback. Marks for edge (R_2, R_1) are being generated by $R_8, R_9, R_{10} \text{and} R_{11}$. Weight of edges denote the number of marks generated for these edges. The 32 marks for (R_2, R_1) should be divided among $Path(R_8), Path(R_9), Path(R_{10})$ and $Path(R_{11})$.

5.1 Finding Contribution of Each Path in Shared Edges

In this section, we compute the number of packets generated by different attackers or basically the number of packets generated from each path individually.

In Figure 5, if N packets are generated by the attacker, then on an average, router R_d at hop distance d from the victim would generate $Np(1-p)^{d-1}$ marks for the edge (R_d, R_{d-1}) where p is the packet marking probability.

$$\begin{aligned} &\frac{\text{Number of marks from Router } R_{d+1}}{\text{Number of marks from Router } R_d} \\ &= \frac{Np(1-p)^{(d+1)-1}}{Np(1-p)^{d-1}} \\ &= (1-p)^{d-(d-1)} \\ &= (1-p). \end{aligned} \tag{4}$$

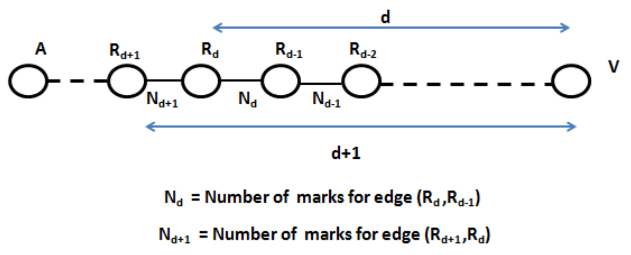
d
A
R_{d+1}
R_d
R_{d-1}
R_{d-2}
V
N_{d+1}
N_d
N_{d-1}
d+1
N_d = Number of marks for edge (R_d,R_{d-1})
N_{d+1} = Number of marks for edge (R_{d+1},R_d)

Figure 5: Calculation of ratio of number of packet-marks generated for neighboring edges from a single source

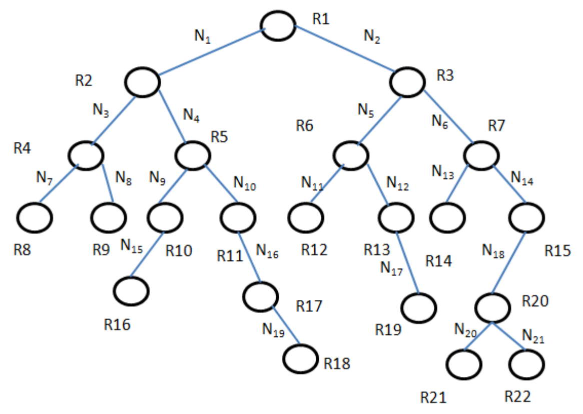

Figure 6: Calculating number of packets from different paths. Weight of the edges in the tree represent the number of packets encoded with that particular edge received by the victim (root R_1).

Let the number of packets from router R_{d+1} be NP_{d+1} and let the number of packets from router R_d be NP_d. Then $\frac{NP_{d+1}}{NP_d} = (1-p)$ or we can write that $NP_d = \frac{1}{(1-p)}NP_{d+1}$. As $(1-p)$ is smaller than 1, hence $NP_d > NP_{d+1}$. Let $\alpha = \frac{1}{1-p}$. Then $NP_d = \alpha NP_{d+1}$, $NP_{d-1} = \alpha^2 NP_{d+1}$ and in general

$$NP_{d-i} = \alpha^{i+1} NP_{d+1}. \quad (5)$$

Hence, the number of marked packets generated by routers as we go upstream increases in geometric progression as can be seen from Equation (5).

Let $N(R_i, R_j)$ represent the number of marked packets obtained for edge (R_i, R_j). In Figure 6, packets encoded with edge (R_4, R_2) would be coming from two sources, first part from node R_8 and another from node R_9. We need to find, how many encoded packets are from R_8 and how many are from R_9. Now contribution of number of packets by R_8 and R_9 in $N(R_4, R_2)$ would be in the ratio $\alpha N(R_8, R_4)$: $\alpha N(R_9, R_4)$. Hence, number of marked packets generated contributed by $R_8 = N_3 \cdot \frac{N_7\alpha}{(N_7\alpha + N_8\alpha)} = \frac{N_3 N_7}{N_7 + N_8}$. In Figure 6, $N(R_i, R_j)$s are denoted by weight of the edges in the graph.

In Figure 6, let us now try to work out the same process on number of marked packets from edge (R_2, R_1). Nodes R_8, R_9, R_{16} and R_{18} are generating packets which are reaching this edge. Now, we need to find the contribution of marked packets for paths originating from each of these nodes. The ratio in which packet marks $N(R_2, R_1)$ would be contributed by R_8, R_9, R_{16}, R_{18} would be in the ratio of $N_7\alpha^2$: $N_8\alpha^2$: $N_{15}\alpha^3$: $N_{19}\alpha^4$. This is because if N_7 marks are generated for edge (R_8, R_4), then using Equation (4), $N_7\alpha^2$ marks should be generated from the packets being generated by router R_8. Similarly, if N_{19} marks are generated for edge (R_{18}, R_{17}), then $N_{19}\alpha^4$ marks should be generated for the edge (R_2, R_1). Hence, we use these numbers as the ratio in dividing N_1 marks among different paths or routers. For example, contribution towards path from leaf R_8 would be

$$contr(R_{16}, (R_2, R_1)) = \frac{N_1 \cdot N_7\alpha^2}{N_7\alpha^2 + N_8\alpha^2 + N_{15}\alpha^3 + N_{19}\alpha^4}.$$

This concept of simple ratio and proportion forms the basis of our algorithm for finding number of marks obtained from different paths. Moreover, in a tree, path from a leaf to root node is unique because each node has a single parent. Let us denote the path from a leaf node R_l to the root node in our graph as $Path(R_l)$. This path would be unique. Let $N(R_i, R_j)$ denote the number of marks obtained for the edge (R_i, R_j) at the victim. Let us denote the contribution of $Path(R_l)$ in $N(R_i, R_j)$ as $contr(Path(R_l), (R_i, R_j))$. Let $d(R_i)$ denote the distance of router R_i from the victim in terms of number of hops. Let $leaf(R_i)$ denote the set of leaves in the sub-tree with root as R_i. Let $P(R_i)$ denote the parent of node R_i.

$$contr(Path(R_l), (R_i, R_j)) = \frac{N(R_i, R_j) \cdot N(R_l, P(R_l)) \cdot \alpha^{(d(R_l) - d(R_i))}}{\sum_{R \in leaf(R_i)} \cdot N(R, P(R)) \cdot \alpha^{(d(R) - d(R_i))}}. \quad (6)$$

5.2 Separating Number of Marks Generated by Each Leaf Node or Path

As discussed in Section 5.1, we can separate out number of marks generated for a given edge (R_i, R_j) among all the leaf routers whose packets are passing through the router R_i using Equation (6). We can safely assume our attack graph to be tree because paths in the Internet are mostly stable [20]. Hence, number of paths from victim to attackers would be equal to the number of leaf nodes in the attack graph.

Now, to solve the problem of shared edges, we need to distribute packets marked with edge (R_i, R_j) to each leaf node that are in the sub-tree rooted at R_i. The marks encoded with edge (R_i, R_j) are being generated by these nodes only. We need to perform a depth first search (DFS) traversal of the attack-graph [4]. For each edge (R_i, R_j), we need to distributed marks with encoding (R_i, R_j) to all the leaf nodes in the sub-tree rooted

Table 1: Symbols used in this paper

Symbols used in this paper and their Meaning	
N(R_i, R_j)	Number of packet marks encoding edge (R_i, R_j)
STLeaves(R_i)	Set of leaf nodes in sub-tree rooted at R_i
P(R_i)	Parent node of node R_i in the attack graph
d(R_i)	Distance of node R_i from root node or victim
p	Marking probability at each router
α	$\frac{1}{1-p}$
Path(R_l)	Path from leaf node R_l to root node
contri(Path(R_l),(R_i, R_j))	Number of marks generated by packets from R_l in $N(R_i, R_j)$.
$leaf(R_i)$	Set of leaf nodes in the sub-tree rooted at node R_i.
$noPktsInPath[R_i]$	Number of marked generated by packets from leaf node R_i
$leaf[j].noPkts()$	Number of marks generated for edge $(leaf[j], P(leaf[j])$
$leaf[j].noHops()$	Hop distance between $leaf[j]$ and victim node.
DMP	Distribute Marked Packets Algorithm 1

at node R_i. After traversing the whole graph, we would have calculated the number of marked packets obtained from each path in the attack graph. Then, we can use CCN equation to find if traceback has been completed for those particular paths.

5.3 Algorithm 1: Distribute Marked Packets

In Distribute Marked Packets (DMP), Algorithm 1, we distribute the number of marked packets to different paths. This helps us in finding the completion of traceback for different paths individually and independently. Table 1 lists the symbols used in this paper. Table 2 lists the functions and symbols used in DMP algorithm. In Lines $1-5$, if it is a root node (parent of node u is null), then we traverse to each of its child node in the for loop of Lines $2-4$. If u is not the root node, then in Line 6, we find the number of leaf nodes in the sub-tree rooted at u. In Lines $7-9$, if u is leaf node, then all packet-marks generated for edge $(u, P(u))$ are added to number of packets in path of u. In Lines $11-17$, all marks for edge $(u, P(u))$ are divided among all the paths that are originating from

Table 2: Functions and symbols used in DMP algorithm

Functions used in DMP Algorithm and their Meaning	
u.parent	Parent node of node u
adj(u)	All the nodes adjacent to node u in graph G
u.getSTLeaves()	get the list of all leaf nodes in the sub-tree with root as u
list $<node>$ leaf	list of leaf nodes
list.size()	size of the list
noPktsInPath[leaf[u]]	Number of marked packets generated by leaf node u
u.noPkts()	Number of packets encoded with edge $(u, P(u))$
leaf[i].noPkts()	Number of packets encoded with edge $(leaf[i], P(leaf[i])$

Algorithm 1 DMP(node u)

```
1: if u.parent == NULL then
2:     for all v ∈ adj(u) do
3:         do DMP(v)
4:     end for
5: end if
6: list < node > leaf = u.getSTLeaves();
7: if leaf.size()==0 then
8:     $noPktsInPath[u] += u.noPkts()$;
9:     return;
10: end if
11: for (i=1; i ≤ leaf.size(); i++) do
12:     for (j=1; j ≤ leaf.size(); j++) do
13:         D += leaf[j].noPkts()*$\alpha^{(leaf[j].noHops()-u.noHops())}$
14:     end for
15:     N = $leaf[i].noPkts()*\alpha^{(leaf[i].noHops()-u.noHops())}*$ $u.noPkts()$;
16:     noPktsInPath[leaf[i]]+= $\lfloor \frac{N}{D} \rfloor$;
17: end for
18: for all v∈ adj(u) do
19:     DMP(v)
20: end for
```

leaf nodes of sub-tree rooted at u. Finally, to traverse the whole attack-graph using *DFS*, we visit each child node of u in the for loop of Lines $18-20$.

Toy Example for DMP Algorithm

In this section, we give a toy example to illustrate how Algorithm 1 runs. In Figure 4, number of marked-packet received for each edge (R_i, R_j) is given. Now, Algorithm 1 starts with the call $DPM(R_1)$. As R_1 is the root node, it goes in the for loop of line 2-4. The we make a call to $DPM(R_2)$. At line 6 in call to $DPM(R_2)$, it finds the list of leaf nodes of the sub-tree rooted at R_2 with call to $u.STLeaves()$. This list contains R_8, R_9, R_{10} and R_{11}. These are the four nodes through which all traffic towards R_2 are coming.

Now, using Equation (4) Lines $10-16$, it calculates that out of those 32 marks for edge (R_2, R_1), how many of them belong to path from R_8, R_9, R_{10} and R_{11} respectively. It finds their share to be $8, 12, 4$ and 8 respectively. Hence, it adds them in noPktsInPath(R) for the leaf routers of its sub-tree. Therefore, noPktsInPath$(R_8) = 8$, noPktsInPath(R_9) is 12 and noPktsInPath(R_{10})=4 and noPktsInPath(R_{12})=8.

After visiting R_2, *DMP* visits node R_4. It has to distribute 10 marks of edge (R_4, R_2) to paths from R_8 and R_9. These are the leaf nodes in the sub-tree rooted at R_4. Using Equation (4), it divides marked packets and hence $noPktsInPath(R_8)$=$noPktsInPath(R_8) + 4 = 12$ and $noPktsInPath(R_9)$ is $noPktsInPath(R_9) + 6 = 18$.

Now from R_4 *DPM* next visits R_8. As R_8 has no child and hence, no leaf node in its sub-tree, all marks for (R_4, R_8) will be allotted to $noPktsInPath(R_8) = 12{+}2 = 14$. Next, in *DFS* traversal, R_9 would be visited and similar to R_8 all 3 marks for (R_9, R_4) would be allocated to $noPktsInPath(R_9)$ and total marks for $Path(R_9)$ would become 21.

After R_9, R_5 would be visited and 6 marks for edge (R_5, R_2) would be distributed. Two marks would be allocated to $Path(R_{10})$ and 4 to $Path(R_{11})$. Next, router R_{10} and R_{11} would be visited and 1 would be added to $Path(R_{10})$ and 2 to $Path(R_{11})$. Next, node R_3 would be visited and 36 marks for edge (R_1, R_3) would be shared as 12 to $Path(R_{12})$, 16 to $Path(R_{13})$, 4 to $Path(R_{40})$ and 4 to $Path(R_{15})$. This procedure would continue for routers R_6, R_{12},R_{13}, R_7, R_{14}andR_{15} and marks would be allocated for paths $Path(R_{12})$, $Path(R_{13})$, $Path(R_{14})$ and $Path(R_{15})$ respectively.

Finally, noPktsInPath$(R_8) = 14$, noPktsInPath$(R_9) = 21$, noPktsInPath$(R_{10}) = 7$, noPktsInPath$(R_{11}) = 14$, noPktsInPath$(R_{12}) = 21$, noPktsInPath$(R_{13}) = 28$, noPktsInPath$(R_{14}) = 7$ and noPktsInPath$(R_{15}) = 4$.

6 Results

In this section, we analyse the results obtained from our experiments. We have conducted our experiments in *Scilab* and *Omnet++* [32]. The results for completion condition for the more accurate *CCN* has been obtained by writing simulation code in *Scilab*. For the multiple attacker scenario and for different attack networks, we have used *Omnet++*. We have used Caida Internet Topology Generator for topology generation [21]. The code to perform traceback at victim was written in C++ in *Omnet++* framework itself.

In Figure 7, we study the number of packets X required to complete IP traceback in PPM for $p = \frac{1}{25}$ and hop distance d varying from $12-25$.

We plot expectation and standard deviation of X. We also plot the upper bound of expectation and standard deviation for X using the formula derived in Subsection 4.1. From Figure 7, we can verify that upper bound for both expectation and standard deviation are correct and they provide tight upper bounds.

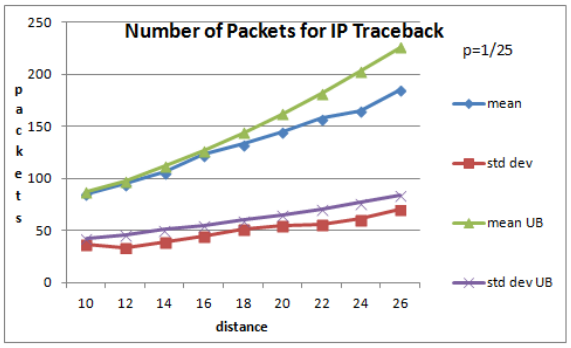

Figure 7: Statistics for number of packets X required for traceback. We show the expected value, standard deviation and upper bounds of mean and standard deviation of X. The upper bounds derived are correct and provide tight bound.

In Figure 8, we observe that our algorithm improves the correctness and reliability of PPM algorithm significantly. In this section, μ and σ represent the theoretically calculated upper bounds for mean and standard deviation of X in this paper. We observe that as we increase the value of completion condition number from μ to $\mu + 2\sigma$, the accuracy of IP traceback becomes more than 96% for all different value of d. On average, taking $\mu + 2\sigma$ as the completion condition number improves the correctness of attack-graph reconstruction by almost 26% and in the best case even by 35%.

However, this improvement in performance comes at the cost of increase in number of packets. As we can observe from Figure 9, percentage increase in number of packets required is around 80% for $CCN = \mu + 2\sigma$. However, we argue that this increase in number of packets is necessary for improving the accuracy of PPM from 70% to 98% on an average.

In Section 4.2, we find theoretical tail estimate of distribution of number of packets required for traceback,X. In Figure 10, we compare theoretical and experimental value of the tail of distribution of X for various hop dis-

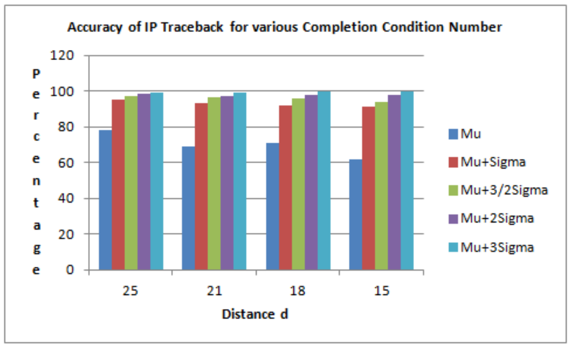

Figure 8: Performance Improvement of PPM in terms of accuracy of IP traceback achieved with our proposed completion condition number. We observe that as we increase the completion condition number, accuracy of PPM increases. Here Mu and Sigma represent the upper bound of mean and standard deviation of X.

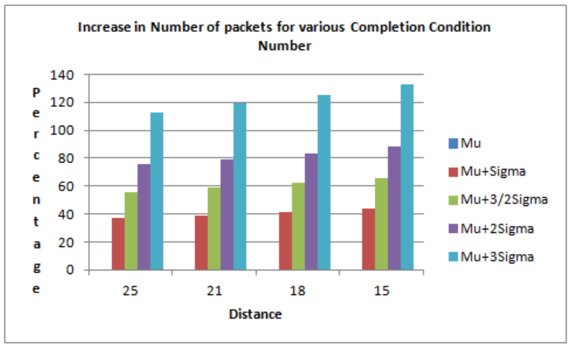

Figure 9: Increase in Number of Packets for PPM due to increase in completion condition number. We observe that for $X = \mu + 2\sigma$, we achieve an accuracy of 98% with an increase of 80% in number of packets. Here Mu and Sigma represent the upper bound of mean and standard deviation of X.

tance d and marking probability $p = \frac{1}{25}$. We observe that probability that IP traceback is incomplete decreases as the value of completion condition number increases. Theoretical tail estimate also confirms that for packet count above $\mu + 2\sigma$ probability of error in traceback decreases below 10% and in fact experimental results also confirm that error is less than 5%.

6.1 Results for Multiple Attacker based DDoS attacks

We use the dual router and AS Internet topology generator provided by Caida to generate Internet topologies used in our experiment for DDoS attacks. The main idea behind this dual Internet topology generator is to rescale a measured Internet topology to a given target size preserving some basic statistical characteristics of the measured topology.

Specifically, it first rescales the measured AS topology to any given target size using methods from the

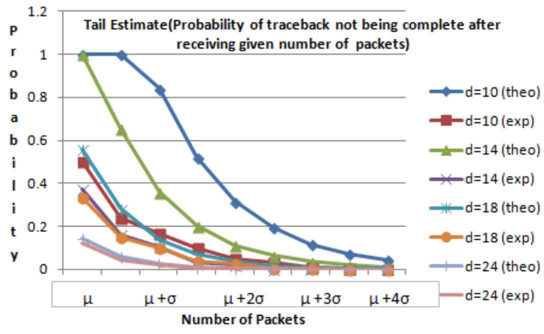

Figure 10: Theoretical Tail Estimate: Probability of IP traceback being incomplete even after getting given number of packets. μ is upper bound of mean and σ is upper bound of standard deviation proposed in this paper. As number of packets increases, tail estimate decreases. It becomes less than 5% for $\mu + 2\sigma$ and almost becomes negligible after $\mu + 4\sigma$ packets.

paper "Graph Annotations in Modeling Complex Network Topologies" [6], and then populate each AS with a router topology generated from scratch using methods from the paper "Hyperbolic Geometry of Complex Networks" [16] matching some average properties of per-AS router topologies.

We create 4 Internet topologies with 25,50,75 and 100 ASes. We randomly choose 100 leaf nodes which we assign them as attackers and generate attack traffic to chosen victim. All routers have PPM enabled in it. The victim collects packet marks from these attackers. Attack Packet generation from 25 of these attackers were deliberately stopped before all the marks could be obtained for these attack paths to test the accuracy of our algorithm. We found the completion condition for each of these 100 attack paths by running the Distribute Marked Packet (DMP) Algorithm. The results are shown for all the four topologies.

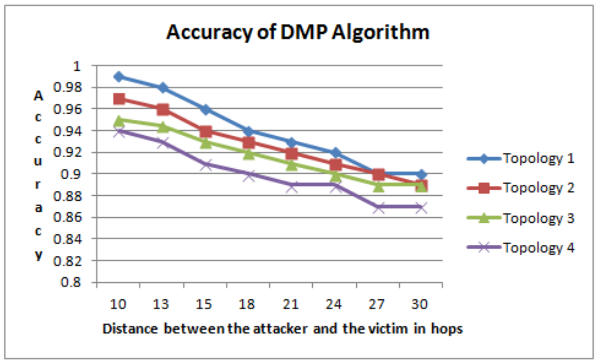

Figure 11: Accuracy of DMP Algorithm: Four different topologies with 25,50,75 and 100 AS were generated using Caida Topology generator and DDoS was simulated on these topologies using Omnet++; after which DMP algorithm was run to find the accuracy of IP traceback using the CCN number for each of these paths.

From Figure 11, it can be observed that for 90% of the cases, our DMP algorithm correctly identifies if the trcaeback has been completed for a given path using the CCN number. In fact, the accuracy is more than 95% for path lengths below 21. Even for path length from 25-31, accuracy of DMP identifying a complete traceback is above 90%.

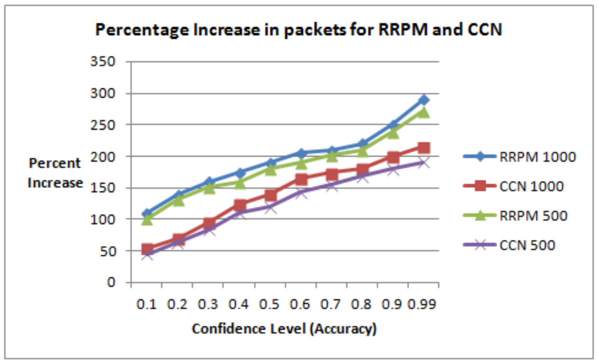

Figure 12: Percentage increase in number of packets for Rectified PPM (RRPM) and CCN: It can be seen that for the same confidence level, the number of packets required for CCN is far less compared to RPPM.

In Figure 12, we compare our work with Rectified PPM (RPPM)[35]. We generate two random tree graph of size 500 and 1000 nodes respectively It has a marking probability of 0.1, we perform traceback with different confidence levels. We compare the number of packets required for each confidence level ranging from 0.1 to 0.99 with a step of 0.1. We find that number of packets required for CCN is less than that required for RRPM for both the topologies and different confidence levels.

7 Conclusion

In this paper, we provide a framework, which makes *PPM* more reliable and accurate. First, we provide a more accurate completion condition number which increases the reliability of PPM based IP traceback from 70% to 98% for a single attacker case. However, *CCN* in its present form can only be used for a single attacker case. It cannot be used for *DDoS* attack because paths from different attackers to victim are not independent. They share edges in the attacker graph. Hence, we cannot separate out the number of packet marks generated for each path individually. To use *CCN* for such cases, we propose an algorithm which can find number of packet-marks generated for each path even if the paths have shared edges. This enables us to use the concept of *CCN* even for multiple attacker DDoS attack and hence makes PPM reliable in this case. Experimental results obtained from our simulation confirms that concept of *CCN* helps us improve the reliability of IP traceback to 95% with minimal number of packets. As far as we know, we are the first one to propose the use of *CCN* to improve the reliability of traceback in case of *DDoS* attack with an explicit algorithm to remove the problem of shared edges. *CCN* based framework can be applied to any flavour of *PPM* and can be used to make traceback more accurate and reliable.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] H. Beitollahi, G. Deconinck, "Analyzing well-known countermeasures against distributed denial of service attacks," *Computer Communications*, vol. 35, no. 11, pp. 1312–1332, 2012.

[2] P. Berenbrink, T. Sauerwald, "The weighted coupon collector?s problem and applications," in *Computing and Combinatorics*, pp. 449–458, Springer, 2009.

[3] Y. Chen, S. Das, P. Dhar, A. El-Saddik, A. Nayak, "Detecting and preventing ip-spoofed distributed dos attacks," *International Journal of Network Security*, vol. 7, no. 1, pp. 69–80, 2008.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, pp. 531–549, Cambridge: MIT press, 2001.

[5] D. Dean, M. Franklin, A. Stubblefield, "An algebraic approach to IP traceback," *ACM Transactions on Information and System Security*, vol. 5, no. 2, pp. 119–137, 2002.

[6] X. Dimitropoulos, D. Krioukov, A. Vahdat, G. Riley, "Graph annotations in modeling complex network topologies," *ACM Transactions on Modeling and Computer Simulation*, vol. 19, no. 4, article no. 17, Oct. 2009.

[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 2, John Wiley & Sons, 2008.

[8] Z. Gao, N. Ansari, "A practical and robust interdomain marking scheme for IP traceback," *Computer Networks*, vol. 51, no. 3, pp. 732–750, 2007.

[9] C. Gong, K. Sarac, "Toward a practical packet marking approach for ip traceback," *International Journal of Network Security*, vol. 8, no. 3, pp. 271–281, 2009.

[10] M. T. Goodrich, "Probabilistic packet marking for large-scale IP traceback," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 15–24, 2008.

[11] B. B. Gupta, R. C. Joshi, and M. Misra, "ANN based scheme to predict number of zombies in a DDoS attack," *International Journal of Network Security*, vol. 14, no. 2, pp. 61–70, 2012.

[12] M. S. Hwang, "Dynamic participation in a secure conference scheme for mobile communications," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1469–1474, 1999.

[13] M. S. Hwang, S. K. Chong, and Te-Yu Chen, "DoS-resistant ID-based password authentication scheme using smart cards," *Journal of Systems and Software*, vol. 83, no. 1, pp. 163–172, 2010.

[14] M. N. Ismail, A. Aborujilah, S. Musa, A. Shahzad, "Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach," in *Proceedings of the ACM 7th International Conference on Ubiquitous Information Management and Communication*, article no. 36, 2013.

[15] D. Jamil, H. Zaki, "Security issues in cloud computing and countermeasures," *International Journal of Engineering Science and Technology*, vol. 3, no. 4, pp. 2672–2676, 2011.

[16] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, M. Bogu, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, Sep. 2010.

[17] M. C. Lee, Y. He, and Z. Chen, "Towards improving an algebraic marking scheme for tracing DDoS attacks," *International Journal of Network Security*, vol. 9, no. 3, pp. 204–213, 2009.

[18] C. T. Li, M. S. Hwang, and C. Y. Liu, "An electronic voting protocol with deniable authentication for mobile ad hoc networks," *Computer Communications*, vol. 31, no. 10, pp. 2534–2540, 2008.

[19] J. Liu, Z. J. Lee, Y. C. Chung, "Dynamic probabilistic packet marking for efficient IP traceback," *Computer Networks*, vol. 51, no. 3, pp. 866–882, 2007.

[20] G. Malkin, *RIP ver. 2 - Carrying Additional Information*, RFC 1058, 1994.

[21] A. Medina, A. Lakhina, I. Matta, J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of the IEEE Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 346–353, 2001.

[22] S. Roy, A. Singh, A. S. Sairam, "IP traceback in star colored networks," in *IEEE Fifth International Conference on Communication Systems and Networks (COMSNETS'13)*, pp. 1–9), 2013.

[23] S. Saurabh, A. S. Sairam, "Linear and remainder packet marking for fast IP traceback," in *Fourth International Conference on Communication Systems and Networks (COMSNETS'12)*, pp. 1–8, 2012.

[24] S. Saurabh, A. S. Sairam, "A more accurate completion condition for attack-graph reconstruction in Probabilistic Packet Marking algorithm," in *IEEE 2013 National Conference on Communications (NCC'13)*, pp. 1–5, 2013.

[25] S. Saurabh, A. S. Sairam, "ICMP based IP traceback with negligible overhead for highly distributed reflector attack using bloom filters," *Computer Communications*, vol. 42, pp. 60–69, 2014.

[26] S. Savage, D. Wetherall, A. Karlin, T. Anderson, "Practical network support for IP traceback," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 295–306, 2000.

[27] D. Seo, H. Lee, A. Perrig, "APFS: Adaptive probabilistic filter scheduling against distributed denial-of-service attacks," *Computers & Security*, vol. 39, pp. 366–385, 2013.

[28] R. L. Smith, "Estimating tails of probability distributions," in *The Annals of Statistics*, pp. 1174–1207, 1987.

[29] D. X. Song, A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, vol. 2, pp. 878–886, 2001.

[30] Y. Tang, et al., "Modeling the vulnerability of feedback-control based internet services to low-rate DoS attacks," *IEEE Transactions on Information Forensics and Security*, vol. 9. no. 3, pp. 339–353, 2014.

[31] J. Udhayan and T. Hamsapriya, "Statistical segregation method to minimize the false detections during DDoS attacks," *International Journal of Network Security*, vol. 13, no. 3, pp. 152–160, 2011.

[32] A. Varga, "The OMNeT++ discrete event simulation system," in *Proceedings of the European Simulation Multi-conference (ESM'01)*, 2001.

[33] A. Welzel, C. Rossow, H. Bos, "On measuring the impact of DDoS botnets," in *Proceedings of the ACM Seventh European Workshop on System Security*, article no. 3, 2014.

[34] T. Y. Wong, J. C. S. Lui, M. H. Wong, "Markov chain modelling of the probabilistic packet marking algorithm," *International Journal of Network Security*, vol. 5, no. 1, pp. 32–40, 2007.

[35] T. Y. Wong, M. H. Wong, C. S. Lui, "A precise termination condition of the probabilistic packet marking algorithm," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, pp. 6–21, 2008.

[36] G. Yao, J. Bi, and A. V. Vasilakos, "Passive IP Traceback: Disclosing the locations of IP spoofers from path backscatter," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 471–484, 2015.

Samant Saurabh did his B.Tech in Electronics and Communication Engineering from IIT Guwahati, India and MS in department of Electrical and Computer Engineering, University of Massachusetts Amherst,USA. Currently he is doing his PhD in Computer Science and Engineering at Indian Institute of Technology, Patna, India. He is Assistant Professor at Birla Institute of Technology, Mesra. His areas of interest are Computer Networks, Operating Systems and Algorithms.

Ashok Singh Sairam obtained his B.Tech degree from National Institute of Technology Silchar, India in the year 1993. He obtained his M.Tech and Ph.D degree from Indian Institute Technology Guwahati, India in 2001 and 2009 respectively. Currently he is working as an Assistant Professor at Indian Institute of Technology Patna, India. His research interests include network security, wireless networks and Internet technologies.