International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 3, March - 2014

A Comparative Study over Various IP Traceback Schemes

Divya Ann Luke
Department of Computer Science and Engineering
SCT College of Engineering,
Trivandrum, India.

Dr. Jayasudha. J. S
Department of Computer Science and Engineering
SCT College of Engineering
Trivandrum, India.

***Abstract* — Defending against distributed denial-of service attacks is one of the hardest security problems on the Internet today. One difficulty to overcome these type of attacks is to trace the source of the attacks because they often use incorrect or spoofed IP source addresses to disguise the true origin. Because the Internet has been applied in various fields, many network security issues emerge and that catch people's attention. To disrupt the service of a server, the sophisticated attackers may launch a distributed denial of service (DDoS) attack. Based on the number of packets to deny the service of a server, DDoS attacks are categorized into flooding-based attacks and software exploit attacks. Various IP traceback schemes are compared in terms of low network and low router overhead and suitable scheme is recognized for identifying the source of attacker.**

Key words* — *DDoS, IP trace back, comparison

I. INTRODUCTION

IP traceback is a name given to any method that determining the origin of a packet on the Internet. Source IP address of a packet is not authenticated because of the trusting nature of IP protocol. As a result, the source address in an IP packet can be forged (IP address spoofing) allowing for Denial Of Service attacks (DoS). The problem of finding the source of a packet is called the IP traceback problem. IP Traceback can be defined as a method for identifying sources of attacks and instituting protection measures for the Internet. Existing approaches to this problem are based on DoS attack detection and their solutions require huge numbers of packets to converge on the attack path.

Most edge routers do not check the source address of packets when they arrive, so the core routers have difficulties in identifying the source of packets. When an attacker wants to hide himself from tracing, the attacker can spoof the source IP address in a packet (IP spoofing). IP spoofing makes hosts hard to defend against a DDoS attack. So a mechanism is needed to locate the real source of impersonation attacks has become an important issue now a days. IP Traceback is a method to find the source of a DoS or a DDoS attack. One solution is to mark the packets with path information as they pass through the routers and then identify the machine that directly generates attack traffic and also the network path the attack traffic follows.

The main goal is to stop attacks at the source for that we need to know where it is coming from, where the traceback comes in. It is possible to categorize the DDoS attacks into flooding-based attacks and software exploit attacks. Flooding based attacks includes ingress filtering, which allows only packets with legal source address to enter. The another one ICMP trace back, which generates ICMP packet with route information and send them along with incoming packet stream. Whenever an attack occurs the victim host can collect the ICMP packets to reconstruct the attack path and there by the source of attack. Another method packet marking scheme, which mark a border router's IP address on the passing packets is used to identify the source of attacks. Most current tracing schemes that are designed for software exploits can be categorized into three groups: single packet, packet logging and hybrid IP trace back. The basic idea of packet logging is to log a packet's information on routers.

II. IP TRACEBACK APPROACHES

Based on the number of packets to deny the service of a server, it is possible to categorize DDoS attacks into flooding-based attacks and software exploit attacks.

For Flooding-based attack, the following schemes are used to identify the source of attack.

1. Ingress Filtering
2. ICMP Trace Back
3. Packet Marking

In the case of Software exploit attack the following methods are considered to give the better results.

1. Single packet
2. Packet logging
3. Hybrid IP trace back

A. Ingress Filtering

By eliminating the ability to forge source addresses is the way to solve the problem of anonymous attacks. One such approach is called ingress filtering [1]. In this approach the router blocks the packets that arrive with illegal source addresses. Therefore the router must have sufficient power to examine the source address of every packet and sufficient knowledge to distinguish between legitimate and illegitimate addresses. It is useful in customer networks or at the border of Internet Service Providers (ISP) because their address ownership is relatively unambiguous. In the case of traffic from multiple ISPs, it is difficult to determine whether the packets arrived from legal source or not.

Even if ingress filtering were universally deployed at the client-to-ISP level, attackers could still forge addresses from the hundreds or thousands of hosts within a valid customer network. It is clear that wider use of ingress filtering would dramatically improve the Internet's robustness to denial-of-service attacks. At the same time it is sensible to assume that such a system will never be full proof and therefore traceback technologies will continue to be important.

B. ICMP trace back

A router sends ICMP traceback [2] messages to the destination of every 20,000th packet, traversing it. The traceback packet contains the IP address of the router sending it, a TTL (or hop limit) of 255, and information about the adjacent routers along the path to the destination. During an attack, a sufficient number of these packets will reach the target for it to reconstruct the path taken by the malicious data. The further improvement of this approach is to select a router with some probability instead of all to write its address on the packet. This is accomplished by reserving a 32 bit "node" field in the header of the packet for holding a router´s address. After receiving enough packets, the victim has at least one sample of every router in the attack path. From the relative number of samples per router (the distribution of samples) the attack path can be reconstructed. The ICMP message insertion and route trace back is shown in the Figure 2.1.

The problems with this trace back techniques can occur when the attacker compromises routers, which cause the packets to be marked with forged information. Besides, source addresses in attack packets cannot be trusted, since they are easy to forge. If all routers in the internet would implement source address filtering, tracing back packet routes would have been simplified.

C. Packet Marking

Packet marking is a technique that does not send additional packets, but rather modifies the IP identification field in each packet to carry information about addresses of routers that are traversed [3]. All marking algorithms contain two components, the marking procedure executed by routers and a path reconstructing procedure implemented by the host. A router marks packets by attaching additional small pieces of information about its path, so the victim can reconstruct the complete way back after observing marked packets.

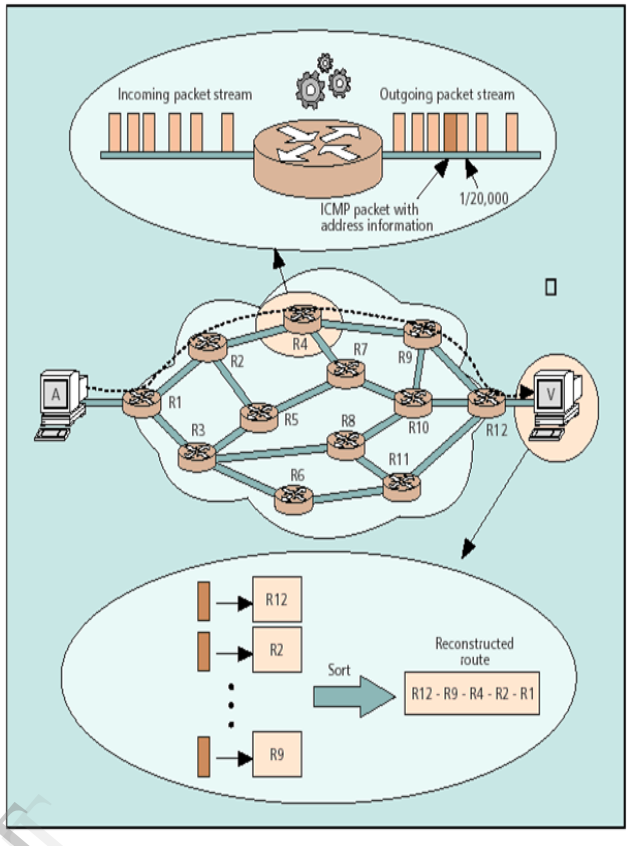

Fig. 2.1: ICMP trace back

The simplest marking algorithm is to append each router´s address to the end of the packet as it travels through the network. But since the length of the path is not known from the beginning, it is not possible to ensure that enough space is available for additional information. This approach can cause problems with IPv6 because of different header format. Deterministic and probability packet marking [4] are the two types of marking algorithms. Packet marking approaches are introduced to mark the router or path information. Mark packets deterministically or probabilistically. It can trace the attacks using marked packets.

1. Deterministic Packet Marking

In DPM trace back scheme [5], Packets are marked using border router's IP address. If the IP header's identification field is not enough to store the full IP address, divides its IP into several segments and digest it. Then randomly chooses a segment and digest to mark on its passing packets. The figure 2.2 represent the marking field of this scheme in IP packet header. When the destination host receives enough packets, it uses the digest value to assemble the different segments. These probability-based schemes require routers to mark partial path information on the packets which pass through them with a probability. If a victim collects enough marked packets, it can reconstruct the full attack path.

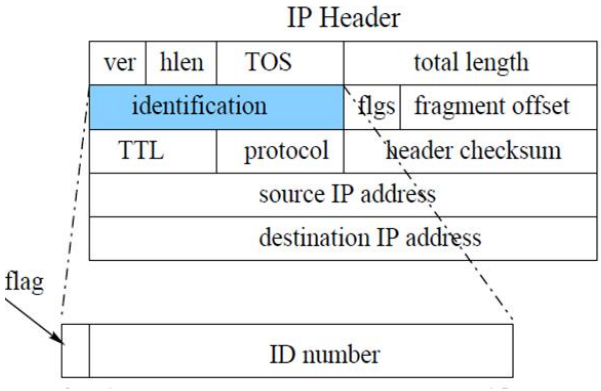

Fig. 2.2: Deterministic packet marking field in IP header

The marking procedure that is followed in this scheme is given below:

Marking procedure at router R,edge interface I:

For each incoming packet *w*
Let *x* be a random number from [0, 1)
If *x*<0.5 then
Write $I_{0\text{-}15}$ into *w*. ID_field
Write 0 into *w*. flags[0]
Else
Write $I_{16\text{-}31}$ into *w*. ID_field
Write 1 into *w*. flags[0]

The path reconstruction algorithm used is given below:

Ingress address reconstruction at victim V:

For each packet *w* from source S_x
If IngressTbl[S_x] = =NIL then
Create IngressTbl[S_x]
If *w*.flags[0]= = 0 then
IngressTbl[S_x]$_{0\text{-}15}$ =*w*.ID_field
Else
IngressTbl[S_x]$_{16\text{-}31}$ =*w*.ID_field

Edge interfaces on all edge routers will place either the first or the last 16 bits in every incoming packet in the ID field, and set the reserved flag to the appropriate value. At the victim, a table matches the source addresses to the ingress addresses is maintained. The victim would check the table entries to see if an entry for a given source already exists, and if an entry did not found then create it. Then depending on the value of the reserved flag, it would write appropriate bits into the ingress IP address value.

2. Probabilistic Packet Marking

It requires routers to mark partial path information on the packets which passes through them with a probability [6]. If a victim collects enough marked packets, it can reconstruct the full attack path. The figure 2.3 represent the marking field of this scheme in IP packet header.

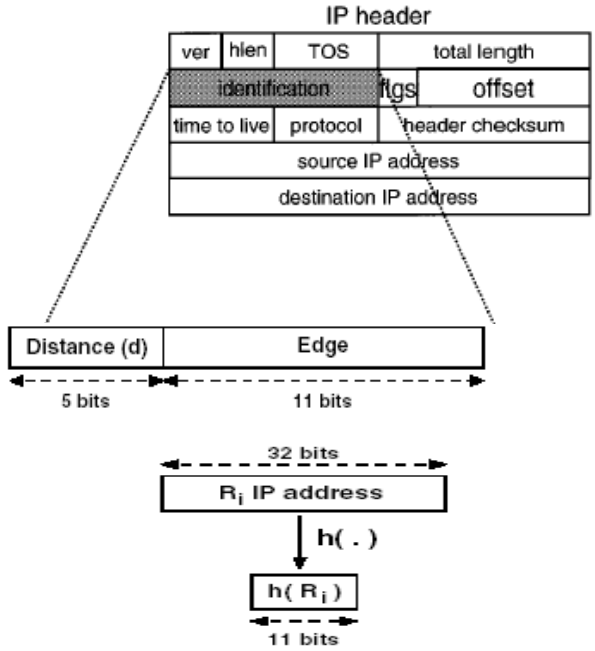

Fig. 2.3: Probabilistic packet marking field in IP header

The algorithm for marking is given below:

For each packet P
Let *u* be a random number from [0,1)
if $u \leq q$ then
P. distance=0
P. start=R
else
if (P. distance== 0) then
P. end=R
P. distance=P. distance + 1

As per the algorithm, certain routers have the probability to mark the packets. Those routers set its ID into p.start field and increase the distance by 1. The node next to the router having marking probability set the packets p.end field to its own ID and increment p.distance by 1 other routers only increment p.distance by 1. Along the path there exist routers having marking probability. The incoming packet information is stored on that route and forward the packet with p.start containing that node's ID, p.end with an initial value (suppose 'e') and p.distance to 0 and forward to next router and repeats the same methods along the path.

For trace back the source, victim decrease the distance field by 1 and forward packet to its neighbors. The neighbors then check the distance field, if it is 0 then check p.end field to check whether its ID matches with that field value. If that value matches, it set the p.end field to 'e' and then forward the packet to its neighbors otherwise drop the packet. Follow the same procedure until it finds its source.

D. Packet Logging

Packets are logged at key routers and then use data mining techniques to find the path that the packet traversed. This approach is useful to trace the attack path long after attack has completed [8]. It also has some drawbacks, including the need of large amount of resource

requirements (possibly addressed by sampling) and a large scale inter-provider database integration problem. The algorithm that is used for packet logging is given below:

```
For each packet p
 If  router ID number i carried by p is valid
        If  the logging flag bit in p is 0
            compute the digest of p
            store the digest in the digest table
              corresponding to i
            mark p with R's ID number
           set the logging flag bit in p to be 1
       else
            mark p with R's ID number
            set the logging flag bit in p to be 0
else
            mark p with R's ID number
            set the logging flag bit in p to be 0
```

The trace back enabled router can perform the packet marking and logging operations. Packet marking is performed by marking the packets with routers identification number and logging operation with record the packet digest and the mark carried by packet. Every router is assigned an ID number of 15 bits in length. The mark is represented by 16-bit identification field in IP header. The leftmost bit is a flag called as logging flag bit. It is set to 1 if the current router commits logging operation on the packet, otherwise set to 0. The remaining 15 bits is used to represent router identification. Figure 2.4 depicts the marking field in IP packet header. The storage of router ID numbers is implemented in a space-efficient fashion. Each router maintains a different digest table for each of its neighbor routers. When a router decides to commit logging operation on a packet, it examines the router ID number carried by the packet to get to know from which neighbor router the packet came, then stores the packet digest in the digest table corresponding to that neighbor. The digest table is paged out before being saturated. Each digest table is annotated with the time interval which the table covers, hash functions used to compute packet digests over that interval and the neighbor router's ID number. Each digest table stores the digests of the packets which are forwarded by the same router and carry the same router ID number.

For each arriving packet, the current router first examines the router ID number marked in the packet header in order to check whether it is *valid*. It is valid in the sense that ID in the packet is equal to the ID number of neighbor router of current router. If it is valid next operation is based on the logging flag bit in the packet, current router may choose only marking operation or both marking and logging operations. If logging flag is 1 which means the upstream router logged the packet and then the current router only mark the packet, otherwise the current router which both mark and log the packet. If the router ID number is not valid, which means that the arriving packet came directly from the sender host or an attacker sends packets with forged mark. In this case router chooses to commit only the marking operation.

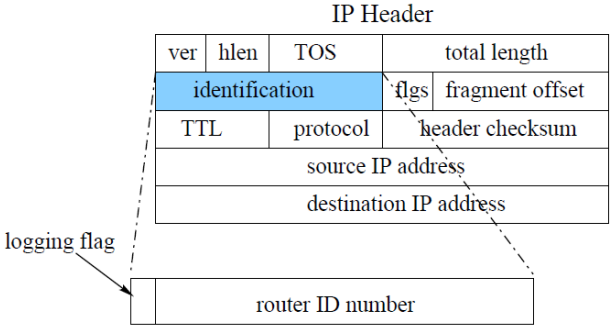

Fig. 2.4: Marking field of packet logging scheme in IP packet header

E. Single packet trace back

This method is used for the traceback of a single packet [9]. It is useful against non-DoS attacks or individualized attacks. It does not require too many system resources (storage, processor). It must be able to trace packets that undergo transformations. It includes two parts, a Source Path Identification Engine (SPIE) and a bloom filter. SPIE is used to digest the unchanged parts of a packet and bloom filter is used to store digest. A signature is stored in a packet digest for each packet that passes through a router. These digests cover a particular traffic flow in a router and cover a specific time interval. A recursive lookup of an attack packet's signature will reveal its route. The packets that are uniquely identified by the first 24 invariant bytes of the packet i.e., the 16 bytes of the header excluding the TOS, TTL, checksum, options and the first 8 bytes of the payload. Frequently modified fields (TOS, TTL, checksum, options) are masked out before digesting in order to ensure that a packet appears identical at all steps along its route.

Constructing a digest table containing packet digests corresponding to the traffic forwarded by a router for a given time interval is a challenging task. A naive technique that simply stored the digests themselves would require massive amounts of storage. Instead, SPIE implements digest tables using space efficient data structures known as Bloom filters. Bloom filter computes k distinct packet digests for each packet using independent uniform hash functions and uses the n-bit results to index into a 2^n-sized bit array. The array is initialized to all zeros and bits are set to one as packets are received.

Path Reconstruction procedure in single packet trace back includes request for trace back. The request contains the offending packet, the point it exited the local SPIE's domain and the time of the packet's arrival at the exit point. The SPIE then checks the appropriate digests and returns an attack graph that either indicates the host that the packet originated from or where it entered the SPIE's network. Figure 2.5 represents the attack graph.

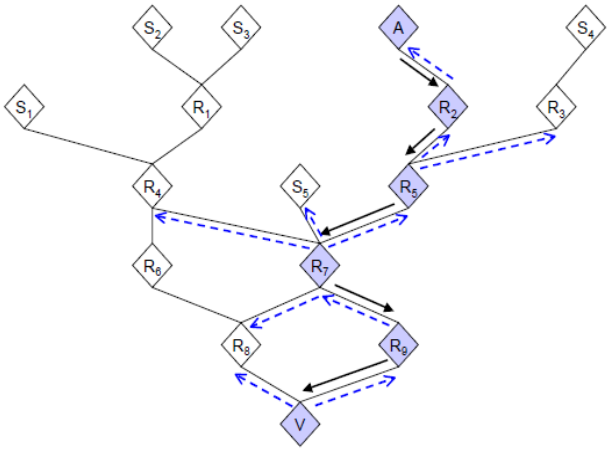

Fig. 2.5: Path reconstruction. Solid arrows represent the attack path; dashed arrows are SPIE queries. Starting at the victim's router *V* and proceeding backwards toward the attacker *A*. Queries are dropped by routers that did not forward the packet in question.

F. Hybrid IP trace back

This method is introduced to solve the storage problem in logging scheme [11]. It combines both marking and logging operations. It uses packet marking to reduce the number of routers required for logging. Marks interface numbers of routers on packet so as to trace the path of packets. The marking field on each packet is limited, this scheme may need to log the marking field into a hash table and store the table index on the packet. Repeat the marking/logging process until the packet reaches its destination. So it is possible to trace back to the origin of attack packets using path reconstruction scheme. The assumptions of this scheme are as follows.

- A router R_i creates an interface table and numbers the upstream interfaces (UI) from 0 to $D(R_i)$ in advance where $D(R_i)$ denotes the number of its neighbors.
- One router knows well whether a packet comes from a router or a local network.
- Such a trace back scheme is viable on every router.
- The traffic route followed by the packet and network topology may be changed, but not often.

The packet marking method is as follows: When a border router receives a packet from its local network, it sets the packet's marking field (P.mark) as zero and forwards the packet to the next core router. When a core router receives a packet P, it computes new value for marking field as $mark_{new}$ = P. mark* ($(D(R_i) + 1)$ UI_i + 1. If $mark_{new}$ is not overflow, the core router overwrites P.mark with $mark_{new}$ and forwards the packet to next core router. If $mark_{new}$ is overflow, the core router must log P.mark and UI_i. That is, it needs to compute hash H(P.mark) first and uses a quadratic probing algorithm to search P.mark and UI_i in Hash table. If P.mark and UI_i are not found there, the core router inserts them as a pair into the table.

Then it gets their index from the table and computes marknew=index * (D(Ri) + 1). Finally, it overwrites P.mark with marknew and the packet to the next router.

The path reconstruction procedure is as follows: When a victim is under attack, it sends to the upstream router a reconstruction request, which includes the attack packet's marking field, termed as *markreq*. When a router receives a reconstruction request, it tries to find the attack packet's upstream router. Firstly it computes *UIi= markreq % (D(Ri*)+1) - 1. If *UIi* ≠-1 which means this packet came from an upstream router along the upstream interface *UIi* , the requested router then restores the marking field to its premarking status. The router computes *markold* = *markreq* / *(D(Ri*)+1) so that we can get the packet's upstream router's *markreq* , i.e. *markold*. Then replace the request's *markreq* with *markold* and send the request to the upstream router. However, if *UIi =-1*, it means either the attack packet's marking field and its upstream interface number have been logged on the requested router, or the requested router itself is the source router. The requested router computes index= *markreq* / *(D(Ri*)+1),so that it is possible to decide whether the requested router is the source router or not. If index is not zero then the requested router has logged this packet, the router then uses index to access hash table *HT* and finds *markold* = *HT*[index]. *mark* and *UIi* = *HT*[index].*UIi*. Next, we use *markold* to replace the request's *markreq* and then send the request to the upstream router. However, if index is zero, this requested router is the source router and the path reconstruction is done.

III. COMPARISON AMONG VARIOUS APPROACHES OF IP TRACEBACK

IP Trace back approaches that have only one intension to locate the origin of an attack. For that purpose they trace back through the path travelled by the packet. Various trace backing schemes that use different kinds of information for locating the source of attack. IP Trace back approaches for flood based attack and software exploit attack are compared using various factors such as management overhead, network overhead, router overhead, distributed capability, post-mortem capability etc. and given in the Table. 3.1. Among various techniques, packet marking is best for tracing the source of flood based attack. Similarly in the case of software exploit attack hybrid IP trace back seems to be efficient and their comparison are given in Table 3.2.

TABLE 3.1
IP TRACEBACK TECHNIQUE COMPARISON FOR FLOOD BASED ATTACK

	Management overhead	Network overhead	Router Overhead	Distributed capability	Post-mortem Capability	Preventive/ reactive
Ingress Filtering	Moderate	Low	Moderate	N/A	N/A	Preventive
ICMP Trace back	Low	Moderate	Moderate	Good	Excellent	Reactive
Marking	Low	Low	Low	Good	Excellent	Reactive
Desirable properties	Low	Low	Low	Excellent	Excellent	Reactive

TABLE 3.2
IP TRACEBACK FOR SOFTWARE EXPLOIT ATTACK

	Management overhead	Network overhead	Router Overhead	Distributed capability	Post-mortem Capability	Preventive/ reactive
Logging	High	Low	High	Excellent	Excellent	Reactive
Hybrid IP trace back	Low	Low	Low	Good	Excellent	Reactive
Desirable properties	Low	Low	Low	Excellent	Excellent	Reactive

IV. CONCLUSION

The desired characteristics of an IP trace back mechanism are that it requires a relatively small number of packets for path reconstruction, low complexity of reconstruction, high robustness and low deployment overhead and cost. Consider all parameters equally important since a low evaluation of one of the parameters opposes using the method in practice. There is a need to identify these attacks and try to stop them. None of the methods possesses all the qualities of an ideal scheme.

Within the existing techniques, Packet marking scheme gives better results for identifying source in flooding based attack and Hybrid IP trace back scheme gives better results for identifying source in software exploit attack. Summarizing the all IP trace back schemes, that each of the suggested approaches is directed to solve one particular issue of the IP trace back problem but there is no technique to solve all of the issues.

V. REFERENCES

1. S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP trace back" IEEE/ACM Transactions on Networking, vol. 9, June 2001.

2. S. M. Bellovin, M. D. Leech, and T. Taylor, "ICMP trace back messages," *Internet Draft: Draft-Ietf-Itrace-04.Txt*, Feb. 2003.

3. Dong Yan, Yulong Wang, Sen Su and Fangchun Yang, "A Precise and Practical IP Trace back Technique Based on Packet Marking and Logging", Journal of Information Science and Engineering 28, 453-470 (2012).

4. Y.Bhavani, P.Niranjan Reddy, "An efficient IP trace back through Packet marking algorithm", International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.3, July 2010.

5. Andrey Belenky and Nirwan Ansari, "IP Trace back With Deterministic Packet Marking", IEEE COMMUNICATIONS LETTERS, VOL. 7,NO. 4, APRIL 2003.

6. Michael T. Goodrich*"*, Probabilistic Packet Marking for Large-Scale IP Trace back", IEEE/ACM Transactions on networking, VOL. X, NO. X, JANUARY 2007.

7. Swathy Vodithala , S. Nagaraju and V. Chandra Shekhar Rao, "A Resolved IP Trace back through Probabilistic Packet Marking Algorithm", International Journal of Computer Science and Telecommunications, Volume 2, Issue 7, October 2011.

8. S. Malliga and A. Tamilarasi, "A hybrid scheme using packet marking and logging for IP trace back," *Int. J. Internet Protocol Technol.*, vol. 5, No. 1/2, pp. 81–91, Apr. 2010.

9. A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer, "Single-packet IP trace back," IEEE/ACM Transactions on Networking, vol.10, December 2002.

10. H. Burch and B. Cheswick, "Tracing anonymous packets to their approximate source", Proc. of the 14th USENIX Systems Administration Conference, December 2000.

11. Ming-Hour Yang and Ming-Chien Yang, "RIHT: A Novel Hybrid IP Trace back Scheme", IEEE Transactions on information forensics and security, vol. 7, NO. 2, April 2012.