# On Improving an Algebraic Marking Scheme for Detecting DDoS Attacks

Moon-Chuen Lee[1], Yi-Jun He[2] and Zhaole Chen[3]

The Chinese University of Hong Kong, Dept. of Computer Science & Engineering,
Shatin, NT, Hong Kong SAR
{*mclee, yjhe, zlchen*}@*cse.cuhk.edu.hk*

***Abstract*:** Distributed Denial of Service (DDoS) attacks could be considered as one of the most serious security problems to the Internet today. To locate the sources of the attack packets, we usually need to find the paths through which the attack packets traversed from the sources to the victim. In this paper, we identify the weaknesses of an existing algebraic marking scheme for detecting DDoS attacks, and propose an improved version of the marking scheme. Simulation experiment results show that the proposed marking scheme could achieve a high success rate in identifying the attack sources. When compared with other marking schemes, it requires fewer packets for attack paths reconstruction. Further, it is characterized by generating no false positives, creating no additional traffic to the network, having a relatively low packet marking and attack path reconstruction overhead, and being backward compatible.

***Keywords*:** Distributed Denial of Service (DDoS), IP Traceback, Probabilistic Packet Marking, Attack Graph Reconstruction**.**

## 1. Introduction

Denial of service (*DoS*) or Distributed DoS (*DDoS*) attacks have become one of the most severe network attacks today. Though relatively easy to be executed [1], it could cause devastating damages. By consuming a huge amount of system resources, *DoS* attacks can render the normal services to the legitimate users unavailable. While email has become the most popular form of communication, the DDoS attack is a common mode of attack to cripple a mail server [2]. Lee and Fung [3] indicate that a DoS attack could be carried out during an authentication process involving public-key based operations. Many different approaches have been proposed to defend against *DoS* attacks:[4]-[8], [27]-[28]. To mitigate the damage from DDoS attack, Su *et al*. [6] proposed an online approach which involves first identifying the approximate sources of the an attack traffic, and then applying packet filtering as near the attack sources as possible. Huang *et al*. [7] proposed an incentive based approach using cooperative filtering and cooperative caching for defeating DDoS attacks. The most effective approach against *DoS* attack is to isolate the attackers from the victim's network. Thus, locating the attack source would be most important. We cannot rely on the source address in the *IP* header of an attack packet since the source address is not authenticated in the current protocol when a router forwards a packet; so the attacker can spoof the source *IP* address while launching an attack. Locating the attack source usually involves finding the paths of the relevant packets. Because of the stateless nature of Internet routing, it is quite difficult to identify such paths. Finding the attack traffic paths is known as the *IP traceback* problem [9].

Dean *et al.* proposed an algebraic marking scheme for *IP* traceback [10], which is based on *linear algebra* and *coding theory*. One main drawback of this marking scheme is its ineffectiveness in dealing with multiple attacks. We propose in this paper an improved algebraic marking scheme, which has greatly enhanced the existing algebraic marking scheme. The proposed approach uses a new packet marking method, and simplifies significantly the paths reconstruction procedure. It can perform *IP* traceback more efficiently even in the presence of multiple attacks.

The rest of this paper is organized as below. In section 2, we introduce the related traceback techniques proposed in the literature. Section 3 presents our enhanced algebraic marking scheme and section 4 gives a detailed performance analysis of our method. After showing the experiment results in section 5, we conclude this paper in section 6.

## 2. Related Work

In general, traceback techniques can be grouped into two major categories—one based on tracing a single packet, and the other based on using a large number of packets for tracing back to the attackers. Hash-based traceback [11], the representative of the former technique, digests and logs some specific information of every packet on the routers. The victim could query the routers whether a certain packet was forwarded by them. There are two obvious problems: each router requires a large-scale database to store and manage the packets information. Furthermore, the queries must be done before the relevant packet records in database are overwritten. The marking scheme proposed in this paper belongs to the category based on using large number of packets for traceback. In the literature, different approaches, based on using a large number of packets, have been proposed for IP traceback, such as link testing, ICMP traceback, and some marking schemes. However, all of them have some drawbacks and cannot be easily applied in practice. In the following subsections, we give a brief overview of several existing methods.

### 2.1 Link Testing

The link testing technique starts from the routers one hop away from the victim and then checks recursively and interactively the upstream links, until the attack source is found or the ISP's border is reached. There are two variants of this technique—input debugging [12] and controlled flooding

Received October 16, 2008 

[13]. Input debugging involves a heavy management overhead—it requires much attention and cooperation of both the remote network administrators and the victim [9]. Controlled flooding assumes that during DoS attacks the links of the attack path would be heavily loaded. By measuring the incoming traffic to the victim and flooding the links, one at a time, of the suspected path, a drop in the attack packets should be observed. The process is repeated for the next upstream link and so on until the attack source is reached. Controlled flooding itself is a DoS attack and will heavily debase the performance of the routers, so it is not practical. Further, it is not suitable for tracing DDoS attacks, since it is very difficult to discern the set of links when multiple attack paths exist. In addition, link testing can only handle ongoing attacks.

### 2.2 Probabilistic Packet Marking (PPM) Scheme

Many existing traceback methods are based on probabilistic packet marking (PPM). For instance, Savage *et al*. [9] propose several marking schemes based on PPM; such marking schemes are also referred to as probabilistic marking schemes. A marking scheme consists of two basic components: the marking algorithm executed by the routers, and the reconstruction algorithm deployed by the victim. In their marking schemes, each router marks the packets with path related information defined by the IP address and the distance (from the router to the victim) with a low marking probability. After having received enough packets, the victim would employ the reconstruction algorithm to reconstruct the attack paths. This approach does not require any interactive cooperation from the ISPs and can therefore avoid the high management overhead of input debugging. When compared to controlled flooding, it has the advantage of being able to cope with DDoS attacks and would not create any additional network traffic. Another advantage of this marking scheme is that it can be used to trace attacks *post mortem*—long after the attacks have stopped.

Savage *et al.* proposed three kinds of marking schemes—node sampling, edge sampling and compressed edge fragment sampling. Node sampling only records one IP address in the packet according to a marking probability, and cannot cope with multiple attacks. Edge sampling records the IP addresses of two adjacent routers and the distance of the further router to the victim; therefore, it needs more than 70 bits for the marking, which are obviously not available in a normal IP header; so it is not backward compatible. The third one uses compressed edge fragments to overcome the storage problem. However, it has two major serious drawbacks when the number of attack paths increases: one is the high computation overhead because it needs to test many combinations of the edge fragments due to the difficulty in grouping the relevant fragments together; another is the large number of false positives because of the large number of collisions of the encoded values.

There are numerous other works exploring the use of PPM to trace the source of a DoS attack [12], [14]-[20]. Park and Lee [18] indicate that while PPM has advantages of efficiency and implementability over other approaches, it has a potential weakness that an attacker may impede traceback by sending packets with forged markings. Alder [19] studied the tradeoff between the number of IP header bits used and the number of packets required for reconstruction.

### 2.3 ICMP Traceback

In addition to the probabilistic marking schemes, there is another similar proposal, namely Bellovin's ICMP traceback method [14]. This method involves using each router to pick a packet with a low probability (1/20,000) and generate an ICMP traceback message or iTrace directed to the same destination as the packet. The iTrace message itself keeps the next and previous hop information. The time to live (TTL) field is set to 255, and is then used to identify the attack path. Under DoS attacks, the victim will get all the addresses of the routers on the attack path that implements iTrace. The addresses, sorted by the TTL fields, can be used to reconstruct the attack path hop by hop. This approach could have significantly less false positives than certain PPM based schemes. However, there are several inherent problems in the current design [9]: an iTrace message might be filtered in a network under attack; iTrace messages rely on an input debugging capability which may not be available in some router architectures; the attackers may send fake iTrace messages to make the victim more difficult to perform attack paths reconstruction; and iTrace messages may give rise to extra network traffic. It is indicated in [21] that even with a few improvements made, the ability of ICMP traceback to handle major DDoS attacks is still poor. Moreover it cannot cope with a DDoS attack with a large number of reflectors.

### 2.4 Algebraic Marking Scheme

Dean, Franklin and Stubblefield proposed an algebraic marking scheme [10] for marking the packets and reconstructing the attack paths. The marking procedure writes two values in the packets, which correspond to $f(x)$ and $x$ of the following polynomial

$$\begin{aligned} f(x) &= a_n + a_{n-1}x + a_{n-2}x^2 + \cdots + a_0 x^n \\ &= a_n + (a_{n-1} + (a_{n-2} + \cdots + (a_1 + a_0 x)x \cdots)x)x \end{aligned} \tag{1}$$

They used *Fullpath* and $x$ to denote the two values. In general, an attack packet will pass through a number of routers before reaching the victim. The first router that decides to make a marking assigns a value for $x$ to the packet and let *Fullpath* be the value of its *IP* address represented by $a_0$. Then the next router computes its *Fullpath* value by multiplying the *Fullpath* value (from the packet) by $x$, and adding its *IP* address (represented by $a_1$). The following routers mark the packet in a manner similar to what the second one did. When the packet arrives at the victim, it records a *Fullpath* value related to a path formed by a number of routers. In fact, it is the value of the above polynomial with the routers' *IP* addresses represented by $a_i$'s and the highest power (i.e. $n$) of $x$ unknown. Note that there is no way for a router to know whether it is the "first" participating router on a particular path; so it has to adopt a coin-flipping method—*random full* (or *partial*) *path encoding* to solve this problem. The router flips a coin and if it comes up tails the router will assume it is not the first router and simply follows the algorithm as presented above; otherwise the router will select an $x$ for the marking of this packet and do the marking in the capacity as the first router. With this packet marking method, each marked

Received October 16, 2008 1554-1010 $03.50 © Dynamic Publishers, Inc

packet received by the victim represents a polynomial. Each polynomial represents one suffix of the whole path. Because the selection of the first marking router is random, the degree of the polynomial is not fixed. They pointed out that with the recent advances in *coding theory* such mixed data problem could be solved to identify the paths if there are enough marked packets.

Further advancement of the underlying mathematical techniques could improve various aspects of their reconstruction algorithm. However, their approach is not powerful enough for dealing with distributed DoS attacks because at present there is not an effective means to find out those packets which have traversed to the victim from the same path; it also requires a huge number of packets to reconstruct the multiple paths.

### 2.5 Advanced Authenticated Marking Scheme

Song and Perrig improved the probabilistic marking scheme and proposed an advanced and authenticated marking scheme [15]. Their method requires a map of the upstream routers with *IP* addresses. By using the router map they made a significant improvement on the performance as measured by the number of packets needed to reconstruct each path, the reconstruction time, the number of false positives, and the ability to deal with distributed DoS attacks. Furthermore, their marking scheme provides a mechanism to authenticate the marking information, which is not available in other previous marking schemes. Therefore, to our knowledge, it could be the best proposal among the current IP traceback methods. Nevertheless, there are still some false positives when the number of attack paths is large, and the design of effective hash functions, which are used in their marking scheme, is not an easy task.

## 3. Proposed Marking Scheme

In this section, we introduce our improved algebraic marking scheme in detail. Unlike the algebraic marking scheme of Dean, *et. al.*, our proposal does not require the use of sophisticated mathematical techniques for paths reconstruction, because we have improved the underlying packets marking procedure. We exploit the idea of probabilistic packets marking (i.e. to mark the packets with a low probability) scheme [9] to reduce the marking overhead of the participating routers. Before presenting further details of our method, we first introduce some relevant definitions and the basic assumptions behind the design of the algorithms. Some of the definitions and assumptions are similar to those presented in [10, 12, 22].

### 3.1 Definitions and Assumptions

An *upstream routers map* describes the topology of the upstream routers of a single host. We assume the upstream routers map captures the *IP* addresses of the routers. Figure 1 depicts an *upstream* routers map with respect to the victim. We use the symbols *V*, *R*, and *A* to denote the victim, router, and attacker respectively. Here *upstream* is used to describe routers viewed from the victim. For example, $R_9$ and $R_{10}$ are the upstream routers of $A_2$. In this graph, there are two attack paths represented by the dotted lines: one is ($A_1$ $R_6$ $R_3$ $R_2$ $R_1$), and the other is ($A_2$ $R_3$ $R_2$ $R_1$). The *distance* between two hosts means the number of routers in the attack path between them. For example, in the attack path ($A_1$ $R_6$ $R_3$ $R_2$ $R_1$), the distance between router $R_6$ and the victim is 3. Some routers might be compromised by the attacker and they would mark fake information in the packets. Therefore, we limit the traceback problem to finding a candidate attack path that contains a suffix of the real attack path, and such a suffix is called *valid suffix* of that path. For example, the path ($R_3$ $R_2$ $R_1$) is a valid suffix of the real attack path ($A_1$ $R_6$ $R_3$ $R_2$ $R_1$). We say a traceback technique is *robust* if the attackers cannot prevent the victim from finding the candidate paths containing the valid suffixes of the attack paths. We say that a router is a *false positive* if it is in the reconstructed attack path but not in the real attack path.

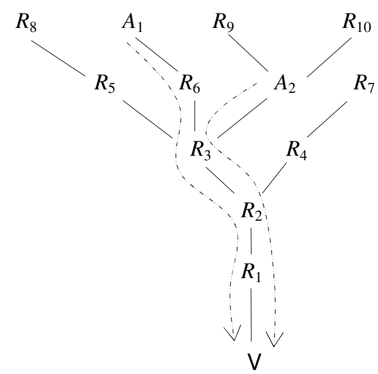


**Figure 1.** An upstream routers map as seen from the victim *V*. There are two attack paths indicated by the dotted lines.

For practical considerations, we make the following assumptions, some of them being similar to those outlined in [10, 12, 22] in the design of our marking scheme.

1) Attackers are able to generate and send any number of packets to a target destination.
2) Multiple attackers may coordinate their attack.
3) Packets may be reordered or lost.
4) The routes between the attack sources and the victim are fairly stable.
5) The routers have limited CPU and memory resources and cannot do too much processing per packet.
6) Attackers might be aware that they are being traced.
7) The markings in a packet may be modified by the attacker.
8) The source address of a packet may be forged.
9) Routers are not compromised widely and the routers adjacent to the victim should not be compromised.
10) The packet size should not grow as a result of tracing.

Assumptions 1 to 8 reflect the ability of the DoS attackers and the weakness of the current network infrastructure. Sophisticated attackers could detect that they are being traced and might send fake packets to confuse the victim. So any *IP* traceback algorithm designer should be aware of such a potential ability of the attackers. Similar to the probabilistic marking scheme proposed in [9], our method marks packets with a low probability; therefore, it requires a good number of packets, sent by the attacker, to reconstruct the attack paths. If some routers are compromised, we might only trace the source back to the compromised router which could tamper the information marked by its upstream routers. Therefore, we use

Received October 16, 2008
1554-1010 $03.50 © Dynamic Publishers, Inc

a *valid suffix* instead of the entire attack path to assess the robustness of a traceback technique. Note that the nearest routers should not be compromised; otherwise they could tamper any information marked by the upstream routers and the victim might not be able to reconstruct any attack paths correctly. Therefore, assumption 9 is a realistic one.

The last assumption concerns avoiding the growth of packet size. There are a number of protocols today which support the packet size to grow. However, increasing the packet size could create the MTU problem and consume additional bandwidth. Thus, we try to avoid designing a traceback system which requires the packet size to grow

## 3.2 Improved Algebraic Marking Scheme

Our proposed marking scheme is presented below. Before introducing the packets marking algorithm, and the attack paths reconstruction algorithm, we first introduce the underlying basic mathematical theory.

### 3.2.1 Basic Mathematical Theory

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \cdots \\ A_n \end{bmatrix} = \begin{bmatrix} Fullpath_1 \\ Fullpath_2 \\ \cdots \\ Fullpath_n \end{bmatrix} \quad (2)$$

The above is a matrix equation (or system of equations) with Vandermonde matrix coefficients. In linear algebra, there is a theorem stating that the above matrix equation, with $A_i$'s unknown, has a unique solution if and only if the $x_i$'s are distinct [23]. By applying field theory to the above theorem, we can obtain a similar theorem over $GF(p)$, where $GF$ denotes Galios Field and $p$ is a prime number if the $x_i$'s and $Fullpath_i$'s are elements in $GF(p)$ [24].

In the context of algebraic marking scheme proposed by Dean, *et. al.*, the above matrix equation represents a sub-path or full path along which the attack packets traversed. Each full path value $Fullpath_i$ is represented by $n$ *IP* addresses $A_1 \ldots A_n$ of the routers which form the attack path. The markings in each marked packet include the *Fullpath* value and the corresponding value of $x$. So each *Fullpath* value captures the information of a path represented by the *IP* addresses of the underlying routers. The reconstruction of an attack path would involve using *Fullpath* markings for $n$ routers from $n$ packets each with a distinct value of $x$. The $n$ *Fullpath* markings correspond to $n$ polynomials/equations for $n$ unknown *IP* addresses of $n$ routers. Mathematically, the $n$ unknown router IP addresses can be solved with $n$ relevant equations.

Instead of encoding the whole attack path, the algebraic marking scheme proposed in this paper encodes only one edge of a path in a packet. An edge consists of two adjacent routers on an attack path through which the packet traversed to the victim. In order to reduce the number of bits for a *Fullpath* marking, each *IP* address is split into 4 fragments. In our proposed marking scheme, the above matrix equation has been modified to the following form:

$$(3) \quad \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^7 \\ 1 & x_2 & x_2^2 & \cdots & x_2^7 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_8 & x_8^2 & \cdots & x_8^7 \end{bmatrix} \begin{bmatrix} A_{1,1} \\ A_{1,2} \\ \cdots \\ A_{2,4} \end{bmatrix} = \begin{bmatrix} Fullpath_1 \\ Fullpath_2 \\ \cdots \\ Fullpath_8 \end{bmatrix}$$

The matrix equation now represents 8 polynomials which encode an edge formed by two adjacent routers, referred to as the first (or start) router and second (or end) router; where $A_{1,1} \ldots A_{1,4}$ and $A_{2,1} \ldots A_{2,4}$ represent the four *IP* address fragments of the first router and the four *IP* address fragments of the second router respectively; $x_1 \ldots x_8$ represent 8 distinct random integers, one for each marked packet.

### 3.2.2 Packet Marking

Similar to other marking schemes, our method involves writing partial path information into the packets' *IP* headers by the routers and reconstructing the attack paths by the victim. The information recorded in each marked packet includes three integer values: $x$, *distance* and *Fullpath*; $x$ is a packet related value; *distance* is the distance between the start router of the edge in the marking and the victim. To reduce the value of *Fullpath*, we split a router $R_i$'s *IP* address into $c$ identical fragments, and use $A_{i,j}$ ($j$ = 1, 2, …, $c$) to denote the value of each fragment. For example, if router $R_1$'s *IP* address is 137.189.89.101 and we split it into 4 fragments, then $A_{1,1}$ = 137, $A_{1,2}$ = 189, $A_{1,3}$ = 89, and $A_{1,4}$ = 101. Using $c$ equal to 4 is an eclectic choice while considering the bits needed to store the *Fullpath* value, and the reconstruction time. The idea behind the proposed packet marking is similar to edge sampling. Consider a packet being marked respectively by any two consecutive routers $R_i$ and $R_j$; that is, $R_i$ and $R_j$ would become the start router and end router of the edge respectively in the marking. Router $R_i$ may compute the *Fullpath* as follows:

$$Fullpath = (A_{i,1} + A_{i,2}x + A_{i,3}x^2 + A_{i,4}x^3) \bmod p$$

Then router $R_j$ may compute the *Fullpath* for the edge as follows:

$$Fullpath = (Fullpath + A_{j,1}x^4 + A_{j,2}x^5 + A_{j,3}x^6 + A_{j,4}x^7) \bmod p$$
$$= (A_{i,1} + A_{i,2}x + A_{i,3}x^2 + A_{i,4}x^3 + A_{j,1}x^4 + A_{j,2}x^5 + A_{j,3}x^6 + A_{j,4}x^7) \bmod p$$

where $p$ is the smallest prime number larger than 255 ($2^8 - 1$), i.e. 257. If $R_i$ is adjacent to the victim, the last 4 terms of *Fullpath* for $R_j$ would be omitted. The aim of mod $p$ in the above formulae is to reduce the value of *Fullpath* so that it would occupy fewer bits in the *IP* header.

Figure 2 depicts the packets marking algorithm, with $c$ equal to 4; we also assume $c$ equal to 4 in the following sub-sections.

Figure 3 illustrates the marking procedure; $F$ and $d$ denote *Fullpath* and *distance* respectively; $v$ represents the value of $A_{2,1} + A_{2,2}x + A_{2,3}x^2 + A_{2,4}x^3$ for router $R$, where $A_{2,i}$'s( i = 1, 2, 3, 4) are the 4 fragments of the *IP* address of $R$. When router $R$ receives a packet from its upstream router $R'$, it first generates a random number $u$ and performs packet marking depending on the value of $u$, and the distance $d$ from the packet.

Received October 16, 2008 1554-1010 $03.50 © Dynamic Publishers, Inc

```
Marking procedure in router R
for each packet P {
   generate a random number u [0, 1) ;
     if (u ≤ q ) {
        // q is the marking probability of each router
        P.distance = 0;
        randomly select an integer x in the range 0..7;
        P.x = x ; // each packet P is assigned one value of x
        Fullpath = (A1,1 + A1,2x + A1,3x^2 + A1,4x^3) mod p;
     }
     else {
        if (P.distance == 0) {
           Fullpath = (Fullpath + A1,1x^4+ A1,2x^5
                      + A1,3x^6 + A1,4x^7) mod p;
         // x is from a packet marked by an upstream router
           P.distance = P.distance + 1;
        }
        else if (P.distance > 0) P.distance =P.distance+1;
           else call error_handler;
     }
}
```

**Figure 2.** Packet Marking Algorithm.

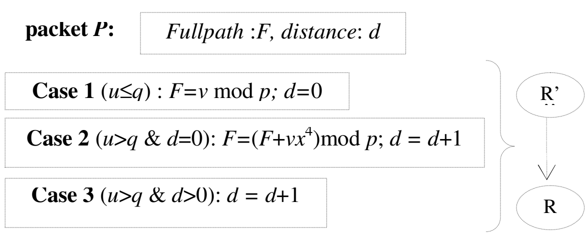


**Figure 3.** Packet Marking illustration at router $R$. $F$ and $d$ denote *Fullpath* and *distance* respectively, $v = A_{2,1} + A_{2,2}x + A_{2,3}x^2 + A_{2,4}x^3$, $R$' is an upstream router of $R$.

As an example, let the *IP* address of router $R$ be 192.168.10.5 and the values of ($F$, $d$, $x$) from the packet being marked are (133, $d$, 2). Router $R$ would first generate a random number $u$. Then the marking algorithm would produce one of the 3 possible outcomes:

**Case 1**($u \le q$): suppose the randomly selected $x$ is 3.
Then, $F = (192 + 168*3 + 10*3^2 + 5*3^3) \bmod 257 = 150$, $d = 0$.

**Case 2**($u>q$ & $d = 0$): Assume $d$ from packet is 0.
$F = (133 + (192 + 168*2 + 10*2^2 + 5*2^3) *2^4) \bmod 257 = 95$, $d = 1$.

**Case 3**($u > q$ & $d > 0$): Increment $d$ by 1.

When 8 (or 4) packets with distinct $x$'s arrive at the victim, the victim can solve the relevant matrix equation in section 3.2.1 to obtain the *IP* addresses (or address) of two adjacent routers (or the nearest router to the victim) in the attack path. Therefore, we use a set of 8 distinct $x$'s (0-7) to do the marking. The inclusion of the *distance* field ensures the *robustness* of our scheme. We can use a method similar to edge sampling [9] to reconstruct the attack path hop by hop.

### 3.2.3 Attacks Paths Reconstruction

There is no simple means to group the packets coming from the same path. It will involve a high computation overhead if we check all possible combinations of the marked packets similar to the probabilistic marking scheme [9]. Therefore, we resort to using an upstream routers map of the victim to simplify attacks paths reconstruction. As pointed out by Song and Perrig, it is quite easy to obtain and maintain such an upstream routers map[15]. After receiving enough marked packets, the victim can reconstruct all the attack paths by using the algorithm as presented in Figure 4.

Figures 5 and 6 are used to illustrate the reconstruction algorithm. Figure 5 shows the initial stage of the attack paths reconstruction, starting from the routers adjacent to the victim. The algorithm first identifies the nearest routers in layer 1 (its distance from the victim is 0). The routers in layer 1 can be found by using the packets from the packet set $P_0$ (for $d$=0) since all packets are grouped by distance $d$. The table on the left side of Figure 5 depicts the packets in each subset $P_{0,x}$ of $P_0$. For each adjacent upstream router $R_i$ of $V$ in the upstream routers map $M$, and for each packet subset $P_{0,x}(x = 0..7)$, a *path* value can be computed; for instance, the path value for $R_1$ can be computed as

$$path = (A_{1,1} + A_{1,2}x + A_{1,3}x^2 + A_{1,4}x^3) \bmod p.$$

**Figure 4.** Attack paths reconstruction algorithm.

```
Reconstruction algorithm
/* Let M denote the upstream routers map;
Let G denote the reconstructed attack graph and be
initialized with one node V for the victim;
Let Pd denote a set of packets with distance d (0 ≤ d ≤
maxd) and Pd,k denote a subset of Pd with x = k;
maxd is the distance from the furthest attack source to the
victim; */
for each direct upstream router R of V in M {
   count = 0; k = 0;
   while (count <4 && k < 8) { x = k;
     path = (A1,1 + A1,2x + A1,3x^2 + A1,4x^3) mod p
      // A1,j (j = 1, 2, 3, 4) form the IP address of R
      // x and Fullpath are from the packet
      for each packet in P0,k {
        if (path ==Fullpath){
           count=count+1;  quit this loop; }
       k=k+1;  }
   if (count == 4) insert R into G next to V;
}
for d = 1 to maxd
   for each router R inserted into G in the last loop {
      for each upstream router R' of R in M{
         k = 0;
         while (k < 8){ x = k; found = false;
           path = (A1,1 + A1,2x + A1,3x^2 + A1,4x^3 + A2,1x^4
                 + A2,2x^5 + A2,3x^6 + A2,4x^7) mod p
           // A1,j (j = 1, 2, 3, 4) form the IP address of R'
           // A2,j (j = 1, 2, 3, 4) form the IP address of R
           for each packet in Pd,k {
              if (path == Fullpath ) {k = k + 1;
                 found = true; quit the present for loop} }
            if not found {quit while loop};
            if (k == 8) insert R' into G next to R;   }
   }}
Output the reconstructed attack paths from graph G
```

Then search for a packet from $P_{0,x}$ with *Fullpath* equal to the computed *path* value. If there are 4 packet subsets each having at least one packet with *Fullpath* equal to the *path* value, we can conclude that the selected router is on one of the attack paths and insert it in the reconstructed attack graph.

Received October 16, 2008

| $P_{d,x}$ | $d$ | $F$ | $x$ |
|---|---|---|---|
| $P_{0,0}$ | 0 | $F_{01}$ | 0 |
| | … | … | … |
| | 0 | $F_{0a}$ | 0 |
| $P_{0,1}$ | 0 | $F_{11}$ | 1 |
| | … | … | … |
| | 0 | $F_{1b}$ | 1 |
| … | … | … | … |
| $P_{0,7}$ | 0 | $F_{71}$ | 7 |
| | … | … | … |
| | 0 | $F_{7h}$ | 7 |

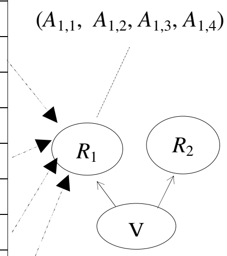


**Figure 5.** Reconstruction illustration 1. *F* and *d* denote *Fullpath* and *distance* respectively. $R_1$ , $R_2$ are upstream routers of *V*.

| $P_{d,x}$ | $d$ | $F$ | $x$ |
|---|---|---|---|
| $P_{i,0}$ | $i$ | $F_{01}$ | 0 |
| | … | … | … |
| | $i$ | $F_{0a}$ | 0 |
| $P_{i,1}$ | $i$ | $F_{11}$ | 1 |
| | … | … | … |
| | $i$ | $F_{1b}$ | 1 |
| … | … | … | … |
| $P_{i,7}$ | $i$ | $F_{71}$ | 7 |
| | … | … | … |
| | $i$ | $F_{7h}$ | 7 |

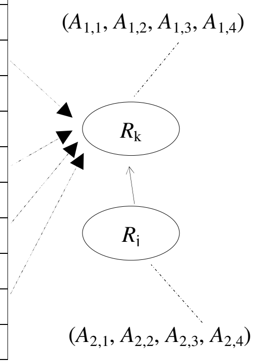


**Figure 6.** Reconstruction illustration 2. *F* and *d* denote *Fullpath* and *distance* respectively. $R_k$ is an upstream router of $R_j$

Figure 6 shows how to reconstruct the attack paths by identifying the routers in other layers after finding the routers in the first layer. Suppose an attack path has been reconstructed from the victim to router $R_j$ in layer *i* (whose distance to the victim is *i*-1). Now, we need to identify its upstream router $R_k$ in layer *i*+1 by using the packets from the set $P_i$. The table on the left side of Figure 6 depicts the packets in each packet subset $P_{i,x}$. For each upstream router $R_k$ next to $R_j$ in *M*, and for each packet subset $P_{i,x}$ (*x*=0..7), the value for *path* can be computed as follows:

$$path = (A_{1,1} + A_{1,2}x + A_{1,3}x^2 + A_{1,4}x^3 + A_{2,1}x^4 + A_{2,2}x^5 + A_{2,3}x^6 + A_{2,4}x^7) \bmod p$$

If *path* is equal to *Fullpath* from any packet in $P_{i,x}$, we move to another packet subset $P_{i,x+1}$. If there is no single packet in $P_{i,x}$ having a *Fullpath* value equal to *path*, we can declare that the selected router is not on the attack paths involving routers in this layer (it could be on the paths involving other layers). If each of the 8 packet subsets has at least one packet with its *Fullpath* value equal to *path*, we can conclude that the selected router is on one of the attack paths and insert it into the reconstructed attack graph.

With the proposed reconstruction algorithm, we can reconstruct multiple attack paths by examining the routers on the victim's upstream routers map, starting from the routers adjacent to the victim, and adding routers to the reconstructed attack graph hop by hop until the ends of the paths have been reached. Note that to identify each router nearest to the victim on an attack path, four packets are used; whereas to identify two adjacent routers, eight packets are used.

## 4. Analysis

The evaluation of a marking scheme for *IP* traceback is normally based on a number of parameters, including number of false positives, minimum number of packets needed to reconstruct each path, marking and reconstruction overheads, backward compatibility, etc. In the following sub-sections, we analyze our proposed *IP* traceback method based on the above-mentioned parameters.

### 4.1 Number of Positives

The most prominent strength of our marking scheme is that no false positives are generated by the attack paths reconstruction algorithm. Any two routers with distinct *IP* addresses cannot yield the same *Fullpath* value for their packets having the same set of values for *x*'s; in addition, any two edges formed by a router *R* and any two of its immediate upstream routers $R_1$ and $R_2$ will not have same *Fullpath* value in their packets. Therefore, the reconstruction algorithm will never include any irrelevant router in an attack path. Moreover, the unique paths traced by the proposed method can be proved mathematically because a *Vandermonde* matrix equation has a unique solution as long as distinct values of *x*'s are used in solving the equation (section 3.2.1). Many other marking schemes produce a certain amount of false positives; for instance, some of them employ hash functions for encoding purpose, which could have a collision problem; that is, they could have the same hash value for two different *IP* addresses.

### 4.2 Minimum Number of Packets

| Version | H.Len | Service Type | Total Length | |
|---|---|---|---|---|
| Identification (16-bit) | | | (1-bit) Flags (total 3-bit) | Fragmentation Offset |
| Time to Live | | Protocol | Header Checksum | |
| Source *IP* Address | | | | |
| Destination *IP* Address | | | | |

**Figure 7.** IPv4 header. The shaded fields (17bits) are little used in current network design.

As the minimum number of packets required to reconstruct an attack path is path independent, it can be analyzed based on a single attack path. Suppose we split an *IP* address into *c* identical chunks and the distance from the attacker to the victim is *d*. As mentioned above, we need *c* packets to identify each router adjacent to the victim and 2*c* packets to identify each upstream edge formed a pair of routers. For each edge, the victim should receive at least 2*c* packets with markings of the edge for attack path reconstruction. If the marking probability is *q*, we need at least $2c/(q(1-q)^{d-1})$ packets. For example, with *c*, *d,* and *q* equal to 4, 20, and 0.01 respectively, the minimum number of packets needed would be 968.

We can also evaluate an upper bound for the expected number of packets for path reconstruction. The probability that a router receives a packet having a marking with a distance

Received October 16, 2008

$d$ is $q(1-q)^{d-1}$. Suppose the attack path length is $D$. We can conservatively estimate the probability of a packet marked with a distance $d < D$ to be $q(1-q)^{D-1}$. Since the victim needs at least $2c$ packets marked with distinct values of $x$ and *distance* from 0 to $D$-1 for reconstructing the entire path, based on the well-known *coupon collector problem* [25], we have

$$E(N) < \frac{2c\ln(2cD)}{q(1-q)^{D-1}} \quad (4)$$

where $E(N)$ denotes the expectation of the number of packets needed for attack path reconstruction.

For example, with $c = 4$, $D = 20$, $q = 0.01$, the upper bound expectation of the number of packets needed for path reconstruction would be 4242. The experimental results presented in section 5 show that, for this case ($c$=4, $D$=20, $q$=0.01), the number of packets needed for path reconstruction, with a success probability of 95%, is around 3500, which is smaller than the expectation.

It is obvious that a larger value for $q(1-q)^{D-1}$ implies a smaller value for $E(N)$. In addition, it can be shown that when $q$ is $1/D$, $E(N)$ reaches a minimum; and as long as $q$ is smaller than $1/D$, the value of $E(N)$ differs by only a small amount, and $q$ should not be smaller than 1%.

### 4.3 Multiple Attacks

A distributed DoS attack normally involves a huge number of packets being sent from multiple attack sources under the control of the attacker. The proposed packets marking algorithm performs packets marking in such a way that the attack paths reconstruction algorithm does not need to discern the packets by the paths through which they traversed to the victim. With the help of the victim's upstream routers map, it can uniquely identify any upstream edge formed by two adjacent routers on each path during attack paths reconstruction. Therefore, the proposed marking scheme is effective for tracing multiple attacks. The algebraic approach marking scheme proposed by Dean *et al.* does not make use of the upstream router map, and the markings do not indicate through which path a packet is from; it is not efficient for multiple attacks. Moreover, the number of packets needed to reconstruct the attack graph is quadratic to the number of attack sources; whereas the number required by our enhanced algebraic marking scheme is linear to the number of attack sources.

### 4.4 Marking and Reconstruction Overheads

The packet marking algorithm as shown in Figure 2 takes only a constant time to execute. Each router marks the packets with a small marking probability. When marking a packet, it computes a *Fullpath* value for a single router or for an edge involving two adjacent routers. To reduce the overhead on the computation of such *Fullpath* values, we can keep possible pre-computed *Fullpath* values in a table for each router. Then any required *Fullpath* value can be obtained by table lookup; thus, the marking overhead would become very small.

The complexity of the reconstruction algorithm as shown in Figure 4 depends on a number of parameters including the number of attack paths, the number of direct upstream edges of each router on an attack path, the number of packets collected in each packet set for a certain distance from the victim, the time to compute *path* values during the reconstruction process, etc. The reconstruction is done hop by hop, starting from the routers closest to the victim. To check if a certain edge is on an attack path, we need to compute 8 *path* values; overall, it is quite fast. Compared to the probabilistic marking scheme of Savage *et al.*[9], checking each direct edge (from the upstream routers map) of a router already found to be on a reconstructed path is much more efficient than checking all possible combinations of *IP* fragments. In addition, we can further speed up the reconstruction process by storing in a table the *path* values based on different values of $x$ for each router. Then, instead of computing the *path* values, the reconstruction algorithm can search from the table the *path* values for any upstream router being examined; so much computation time could be reduced. Overall, the proposed paths reconstruction algorithm is quite efficient.

### 4.5 Backward Compatibility

Backward compatibility is an important issue concerning whether the proposed method can be put into practice. As our marking scheme involves writing some information to the *IP* header of a packet, we should find out the maximum number of bits available in an *IP* header that can be used to store the markings.

The total number of bits $b$ needed to store the markings can be estimated by: $\log_2(p) + \log_2(d) + \log_2(n)$; the three terms estimate the bits to store *Fullpath* (a value less than $p$), *distance*, and $x$ respectively. In practice, we can set $c$, $d$, $p$, and $n$ as follows:

$$c = 4,\ d = 32,\ p = 257,\ n = 2c = 8$$

Then the total number of bits $b$ would be 17. The reason for setting $n$ equal to $2c$ is that each *Fullpath* value is related to $2c$ fragments of two *IP* addresses. As long as there are $2c$ packets with distinct values of $x$, the next hop router can be identified. Therefore, 3 bits have been used to represent 8 distinct values of $x$.

There is a tradeoff between the number of packets needed for paths reconstruction and the number of bits for the markings, which depends partly on the number of *IP* address fragments, $c$. A smaller $c$ implies: i) fewer packets and a shorter time would be required for attack paths reconstruction; ii) more bits would be needed since the value of each *IP* address fragment would be larger. Though the range of distinct values for $x$ would be smaller, the total number of bits needed would be larger.

As the number of bits available in the *IP* header that can be used to store the markings is very limited, we eclectically choose $c$ equal to 4 in our implementation. Since almost any packet can reach its destination through no more than 32 hops [26], allocating 5 bits for *distance* should be sufficient. In summary, we need only 17 ($>\log_2(257) + \log_2(32) + \log_2(8)$) bits to store the markings in our marking scheme.

Figure 7 shows the structure of the IPv4 header. The 16-bit *Identification* field is used to allow the destination host to determine which datagram a newly arrived packet fragment belongs to. Stoica and Zhang pointed out that less than 0.25% of the entire network traffic is fragments [22]; we consider that overloading the Identification field can be backward compatible. There is also one out of three bits of the *Flags* field, which is little used in the current implementation [10].

Received October 16, 2008 1554-1010 $03.50 © Dynamic Publishers, Inc

These 17 bits could be used in the proposed marking scheme. Therefore, our marking scheme is backward compatible with current protocols and could be considered for practical use.

Nonetheless, the proposed marking scheme could not be applied directly to *IP*v6, where the *IP* header does not have the *Identification* field and the *IP* address is 128 bits. However, it is possible that there could be similar space available in the *IP* header of *IP*v6; if the space available is not sufficient, we need to partition the *IP* address into more fragments.

## 5. Simulation Results

We have performed a good number of simulation experiments to examine the feasibility and to assess the performance of our marking scheme. The primary objective of the experiments is to examine the following parameters related to the performance of the marking scheme: the number of false positives, the minimum number of packets needed for reconstruction, the reconstruction time, etc.

We prepare for the simulation experiments an upstream routers map with over 2000 routers. The routers are assigned some real IP addresses obtained from the Internet by using the *traceroute* technique. The attack paths are randomly chosen from the paths in the map; and different numbers of packets are generated and transmitted along each of these paths respectively. Each router simulates marking any packets it receives, according to our packet marking algorithm. After collecting sufficient number of marked packets, the victim simulates reconstructing the attack paths according to our proposed reconstruction algorithm.

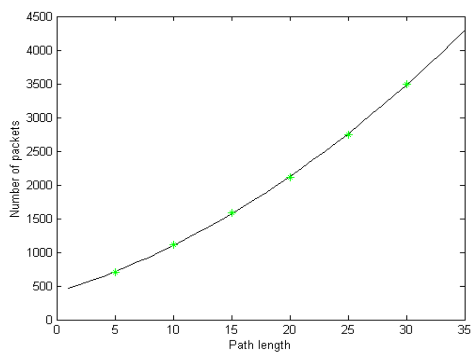


**Figure 8.** Minimum number of packets required for attack paths reconstruction ($q$ = 4%).

The experiment results show that the proposed marking scheme is feasible and the performance is satisfactory. They also confirm that the attack paths reconstruction algorithm yields no false positives.

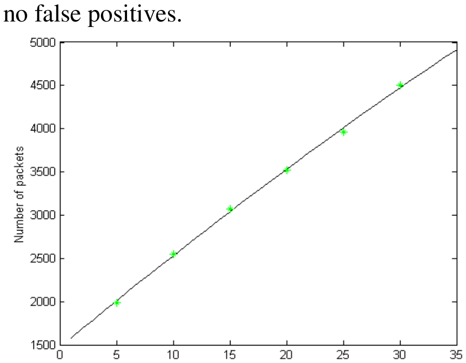


**Figure 9.** Minimum number of packets required for attack paths reconstruction (q = 1%).

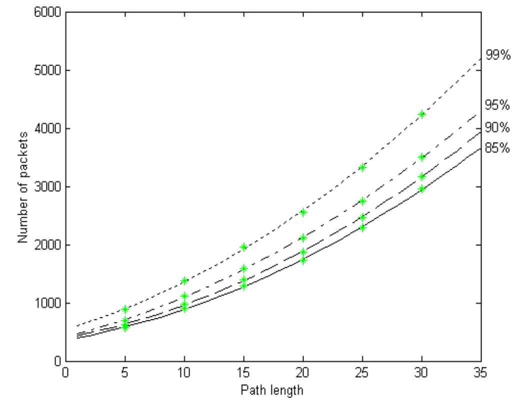


**Figure 10.** Minimum number of packets, with $q$ = 4%, required for reconstruction for different success probabilities: 85%, 90%, 95%, and 99% respectively.

Figures 8 and 9 present two plots showing the minimum number of packets, required for reconstruction, sent by the attacker along any single path for two different marking probabilities 4% and 1% respectively, assuming the reconstruction success probability being 95%. As expected, with a smaller marking probability $q$, more packets would be needed for attack paths reconstruction. Each data point in each of the plots corresponds to an average of the data values obtained from over 300 independent experiments for a certain path length. The experiment results on the minimum number of packets needed for paths reconstruction have been compared with those presented in FMS [9] and the advanced marking scheme [15] respectively. Note that such results are independent of the platforms of the experiments. When compared with FMS [9], and scheme 1 of the advanced marking scheme [15], our marking scheme requires significantly less packets for attack paths reconstruction. Our scheme is also fairly better than scheme 2 with $m$>7 (the case of the minimum number of false positives), and not worse than scheme 2 with $m$>6 (the case of the second least number of false positives) of the advanced marking scheme [15]. In addition, if the number of false positives is used as a performance metric, our marking scheme outperforms all schemes of the advanced marking scheme [15] since our reconstruction algorithm does not generate any false positives.

We have also performed experiments to investigate how the number of packets needed for reconstruction varies for different successful reconstruction probabilities. Figure 10 shows the results based on a marking probability of 4%; the solid line, dashed line, dash-dotted line and dotted line represent the number of packets for reconstruction with a success probability of 85%, 90%, 95%, and 99% respectively. As shown in Figure 10, for a given path length the number of packets for reconstruction increases geometrically as the success probability is increased. For example, for the path length of 25, the number of packets increases from 2290 to 2470, 2740, and 3330 as reconstruction success probability increases from 85% to 90%, 95%, and 99% respectively. The figures show that the number of packets for path reconstruction increases non-linearly with the success

Received October 16, 2008 1554-1010 $03.50 © Dynamic Publishers, Inc

probability. The increase in the number of packets will be more acute as the success probability approaches 100%.

In summary, if an attacker sends out more than 3000 packets along a single path whose length is generally no more than 30, the attack path could most likely be traced by our marking scheme.

Concerning the speed of attack paths reconstruction, our algorithm can reconstruct 50 distributed attack paths (path lengths ranging between 20 and 30) within 3 seconds on a 500MHz Pentium III Linux workstation. It is obviously much faster than FMS [9]. A good portion of the reconstruction time is spent on grouping the packets. When the number of received packets becomes very large, say, more than 300,000, the proposed reconstruction algorithm might take more time than does the advanced marking scheme [15]. However, in practice, the victim can simply use a subset of received packets for reconstruction if the reconstruction time is crucial; moreover, if necessary, the overhead on grouping the packets could be much reduced by using sophisticated sorting algorithms and implementation techniques.

## 6. Conclusion

The algebraic marking scheme proposed in this paper improves on the algebraic marking scheme proposed by Dean *et al*. [10] by using an innovative packet marking technique which records probabilistically in each packet markings related to at most two adjacent routers' *IP* addresses. The attack paths can be reconstructed with the help of the victim's upstream routers map, which allows the reconstruction algorithm to be simplified and speeded up significantly. With the inclusion of a *distance* value in the packet, a compromised router cannot arbitrarily forge a wrong marking in a packet to mislead the victim. Therefore, the *distance* field improves the robustness of the markings. Another advantage of the proposed marking scheme is that it can trace multiple attacks efficiently. The reconstruction algorithm is not required to identify packets coming from the same path; it simply examines efficiently all upstream edges of any reconstructed router by using the upstream routers map to reconstruct the attack paths hop by hop, starting from the router closest to the victim. When compared to other IP traceback schemes, the proposed method has the advantage of being able to effectively eliminate the false positives, and to perform paths reconstruction with fewer packets from the attackers.

One fundamental disadvantage of the proposed method is that it does not authenticate the markings. Therefore, a compromised router might tamper the markings of its upstream routers and make the victim reconstruct wrong paths. As a result, our marking scheme can reconstruct only a valid suffix of the real attack path, though the compromised router could be regarded as an attacker to a certain extent. In our future work, we shall develop a technique to authenticate the markings so that the compromised routers could be identified. While the proposed marking scheme is backward compatible with the present IP network protocols, it cannot be applied directly to IPv6. However, we believe that it could be modified to suit the future network protocol.

## Acknowledgment

The work reported in this article has been supported in part by CUHK under RGC Direct Grant with Project ID 2050347.

## References

[1] Computer Emergency Response Team (CERT). "Denial of Service". Feb. Tech Tips, 1999. http://www.cert.org/tech_tips/denial_of_service.html.

[2] D. Nagamalai, C. Dhinakaran, and J.K. Lee. "Multi Layer Approach to Defend DDoS Atttacks Caused by Spam". In *Proceedings of 2007 International Conference on Multimedia and Ubiquitous Engineering*, pp. 97-102, 2007.

[3] Moon-Chuen Lee and C.K. Fung. "A Public-Key Based Authentication and Key Establishment Protocol Coupled with a Client Puzzle", *Journal of the American Society for Information Science and Technology,* 54(9), pp. 810-823, 2003.

[4] P. Karn and W. Simpson (1999). Photuris: "Session-key management protocol". RFC 2522.

[5] Garg, A., and Reddy, Narasimha A.L. "Mitigation of DoS attacks through QoS regulation", In *Proceedings of the Tenth International Workshop on Quality of Service*, May, pp. 45-53, 2002.

[6] Wei-Tsung Su, Tzu-Chieh Lin, Chun-Yi Wu, Jang-Pong Hsu, and Yau-Hwang Kuo. "An On-line DDoS Attack Traceback and Mitigation System Based on Network Performance Monitoring". In *Proceedings of the 10th International Conference on Advanced Communication Technology (*ICACT 2008*),* pp. 1467-1472, 2008.

[7] Yun Huang, XianJun Geng, and REW B. Whinston. "Defeating DDoS Attacks by Fixing the Incentive Chain", *ACM Transactions on Internet Technology*, Vol. 7, No. 1, Article 5, 2007.

[8] Shui Yu and Wanlei Zhou. "Entropy-Based Collaborative Detection of DDOS Attacks on Community Networks". In *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 566-571, 2008.

[9] S. Savage, D.Wetherall, A. Karlin, and T. Anderson. "Practical Network Support for IP Traceback". In *Proceedings of 2000 ACM SIGCOMM Conference*, pp. 295-306, 2000.

[10] Drew Dean, Matt Franklin, and Adam Stubblefield. "An Algebraic Approach to IP Traceback", *ACM Transactions on Information and System Security, Vol .5 No.*2, pp. 119-137, 2002.

[11] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. "Hash-Based IP Traceback". In *Proceedings of ACM SIGCOMM 2001*, pp. 3-14, August. 2001

[12] R. Stone. CenterTrack. "An IP overlay network for tracking DoS floods". In *Proceedings of 2000 USENIX Security Symp*., pp. 199–212, Jul. 2000.

[13] H. Burch and B. Cheswick. "Tracing Anonymous Packets to Their Approximate Source". In *Proceedings of 2000 USENIX LISA Conference*, pp. 319–327, 2000.

[14] Steve Bellovin. "ICMP Traceback Messages". *Internet Draft, AT&T Labs Research*, Jan. http://www.ietf.org/internet-drafts/draft-ietf-itrace-03.txt . 2003.

[15] D. Song and A. Perrig. "Advanced and Authenticated Marking Schemes for IP Traceback", In *Proceedings of* IEEE INFOCOM 2001, pp.878-886, 2001.

[16] Z. Gao and N. Ansari. "Enhanced Probabilistic Packet Marking for IP Traceback", In *Proceedings of IEEE Global Communications Conference*, pp. 1676-1680, 2005.

[17] S.C. Ke and Y.W. Chen. "An Edge Router-based Fast Internet Traceback", In *Proceedings of 2007 IEEE Region 10 Conference*, pp. 1-4, 2007.

[18] K. Park and H. Lee. "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack". In *Proceedings of IEEE INFOCOM'01,* 2001.

[19] M. Adler. "Tradeoffs in Probabilistic Packet Marking for IP Traceback", In *Proceedings of the 34th ACM Symposium Theory of Computing*, pp. 407-418, 2002.

[20] Marcel Waldvogel. "GOSSIB vs. IP Traceback Rumors". *In Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC)*, pp. 5-13, 2002.

[21] S.F. Wu *et al*.. "On Design and Evaluation of 'Intention-Driven' ICMP Traceback". *In Proceedings of 10th International Conference on Computer Communications and Networks*, pp. 159-165, 2001.

[22] I Stoica and H. Zhang. "Providing guaranteed services without per flow management". *In Proceedings of ACM SIGCOMM '99*, pp. 81–94, Cambridge, MA. 1999.

[23] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, Cambridge University Press, pp. 83-84, 1992.

[24] Thomas W. Judson. *Abstract algebra: theory and applications*, Boston, MA: PWS Pub. Co., page 379, 1994

[25] Motwani and Rajeev. *Randomized algorithms*, Cambridge, New York, N.Y.: Cambridge University Press,1995.

[26] Wolfgang Theilmann and Kurt Rothermel. "Dynamic distance maps of the internet". In *Proceedings of IEEE INFOCOM Conference*, Vol.1, pp. 275-284, 2000.

[27] Vrizlynn L. L. Thing, Morris Sloman, Naranker Dulay. "Non-Intrusive IP Traceback for DDoS Attacks". In *Proceedings of 2007 ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS'07)*, pp. 371-373, 2007.

[28] Haining Wang, Cheng Jin, Kang G. Shin. "Defense Against Spoofed IP Traffic Using Hop-Count Filtering", *IEEE/ACM Transactions on Networking*, Vol. 15, No. 1, pp. 40-53, 2007.

## Author Biographies

**Moon-Chuen Lee** He, born in Hong Kong, received the B.Sc. degree from University College London, the M.Sc. degree from the Imperial College of Science and Technology, London, and the Ph.D. degree in computer science from the University of London. He is currently Associate Professor at the Department of Computer Science and Engineering of the Chinese University of Hong Kong. His research interests lie in the areas of content-based image retrieval, multimedia applications, video compression algorithms, network security, and mobile positioning.

**Yi-Jun He** Born at Jinshi (Hunan, China) in 1981. She received B.Sc. degree in computer science from Hunan University of Science and Techonology, Hunan, China in 2004, and the M.S. degree in computer science from Central South University, Hunan, China in 2007. She was a Research Assistant at the Department of Computer Science and Engineering of the Chinese University of Hong Kong in 2007. She is currently a PhD student in the Department of Computer Science of the University of Hong Kong. Her research interests include network security, wireless communications and cryptography.

Received October 16, 2008