Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 866–882
Computer Networks
www.elsevier.com/locate/comnet

Dynamic probabilistic packet marking for efficient IP traceback ☆

Jenshiuh Liu [a,*,1], Zhi-Jian Lee [a], Yeh-Ching Chung [b]

[a] Department of Information Engineering and Computer Science, Feng-Chia University, Taichung 407, Taiwan, ROC
[b] Department of Computer Science, National Tsing Hua University, HsingChu 300, Taiwan, ROC

Received 23 May 2005; received in revised form 20 December 2005; accepted 28 June 2006
Available online 28 July 2006

Responsible Editor: M. Smirnow

Abstract

Recently, denial-of-service (DoS) attack has become a pressing problem due to the lack of an efficient method to locate the real attackers and ease of launching an attack with readily available source codes on the Internet. Traceback is a subtle scheme to tackle DoS attacks. Probabilistic packet marking (PPM) is a new way for practical IP traceback. Although PPM enables a victim to pinpoint the attacker's origin to within 2–5 equally possible sites, it has been shown that PPM suffers from uncertainty under spoofed marking attack. Furthermore, the uncertainty factor can be amplified significantly under distributed DoS attack, which may diminish the effectiveness of PPM. In this work, we present a new approach, called dynamic probabilistic packet marking (DPPM), to further improve the effectiveness of PPM. Instead of using a fixed marking probability, we propose to deduce the traveling distance of a packet and then choose a proper marking probability. DPPM may completely remove uncertainty and enable victims to precisely pinpoint the attacking origin even under spoofed marking DoS attacks. DPPM supports incremental deployment. Formal analysis indicates that DPPM outperforms PPM in most aspects.

Keywords: Denial-of-service; IP; Network security; Probabilistic packet marking; IP traceback

1. Introduction

In recent years, denial-of-service (DoS) attack has become a pressing problem on the Internet [1–6]. As opposed to other types of attacks, DoS attacks do not alter, delete or steal information stored on victims' systems, but prevent legitimate access to services normally provided by victims. In February 2000, Yahoo, was held back by DoS attacks for over 12 h. Many DoS attacks (e.g., eBay,

☆ A preliminary version of this paper has appeared in the Proc. of the 11th IEEE Intl. Conf. on Network, pp. 475–480, September 2003.

* Corresponding author. Tel.: +886 4 2451 7250.
E-mail address: liuj@fcu.edu.tw (J. Liu).

[1] This work was sponsored in part by a grant from the National Science Council of ROC under contract no. NSC91-2213E-035-031.

1389-1286/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2006.06.009

Amazon, and other .com sites) occurred before and after the Yahoo's incident. DoS attacks have become more prevalent recently due to the lack of an efficient method to locate the real attackers and ease of launching an attack with readily available source codes on the Internet. Research work observes that there were 12,805 attacks on over 5000 distinct hosts belonging to more than 2000 distinct organizations during a three-week period [7]. Even worse, recent reports indicate that hackers have developed tools to launch an attack simultaneously from many separate sources . This is the so called distributed denial-of-service(DDoS) attack [8,5,9]. As the Internet attracts more and more applications, coping with DoS has become an important issue.

Most work on solutions to DoS attack has followed two directions: One is to tolerate the attack by mitigating its effect on the victim [10–12]. The other is to attempt to locate the origin of the attack and, hopefully, to stop the attack at the source. The process of identifying the machines that directly generate attack packets and the network paths these packets follow is called the *traceback* problem [13]. Traceback is a subtle scheme to tackle DoS attacks. If it can provide us with a precise attack origin, then we may apply some proper actions to stop attacks completely; even incomplete or approximate information is valuable, since the closer packet filtering is applied to the attack source the more we are able to control and contain attacks.

It is surprisingly difficult to determine the origin of attacks on the Internet due to the characteristics of IP routing: each packet is routed independently to its destination, and moreover, attackers routinely disguise their origin using an incorrect or ''spoofed'' address in the IP source address field. Much research work has been done on the traceback problem [13–17]. Recently, Savage et al. [13] have proposed the *probabilistic packet marking* (PPM) as a network support for practical IP traceback. In their work, each router probabilistically marks packets with path information as they pass by. By collecting a certain number of packets, a victim is able to identify the network path(s) traversed by attack traffic without requiring interactive assistance from outside network operators. Traceback is a game between the victim and the attacker. Under PPM, the victim may raise the marking probability in order to collect path information with the least number of packets. On the other hand, the attacker may choose to spoof marking field and/or source address to lessen the effectiveness of PPM. Park and Lee [18] have shown that PPM suffers from uncertainty under spoofed marking attack, which may impede traceback by the victim. Their interesting findings are as follows: With PPM, the victim can pinpoint the attacker's address to within 2–5 equally possible sites under a single source DoS attack. However, under DDoS attack, the uncertainty introduced by the attacker will be significantly amplified, which may diminish the effectiveness of PPM.

In this work, we present a new approach, called *dynamic probabilistic packet marking* (DPPM), to further improve the effectiveness of PPM. By deducing the traveling distance of each packet and then assigning a proper marking probability, DPPM may completely remove uncertainty even under spoofed marking DoS attack. With support from hosts and routers in the Internet community, DPPM enables any victim to precisely pinpoint the attacking origin under spoofed marking DoS attacks. The number of packets required by DPPM to reach its conclusion is much less compared to that of PPM. Moreover, our proposed DPPM can be applied to DDoS attacks and still leaves a very limited uncertainty.

The rest of this paper is organized as follows: Section 2 contains an introduction to PPM. We present our DPPM and its implementation issues in Section 3. Performance analysis of DPPM is given in Section 4. In Section 5, we examine DPPM under attacks with spoofed TTL values. Section 6 compares the performance of PPM and DPPM in the presence of legacy routers. Related work is discussed in Section 7. Finally, a summary and remarks are given in Section 8.

2. Preliminaries

One feature of the Internet Protocol (IP) is that the source host itself fills in the IP source address field before it sends any packet. It has been long understood that this permits anonymous attacks. Packet marking [15,13] is one way to enable IP traceback, whereby each router puts some path information as packets are forwarded to their destinations. By collecting a certain number of packets, a victim is able to identify the network path(s) traversed by attacking packets. Probabilistic packet marking (PPM) is one of the most prominent methods for traceback in DoS attack. In this section, we will briefly review PPM.

2.1. Probabilistic packet marking

A traceback can be divided into *marking* and *reconstruction* phases. During the marking phase, each router marks packets with partial path information with a probability p, as they pass by. A victim performs the reconstruction, where it uses the path information recorded in the packets to create a network graph leading back to the source or the sources of an attack. Node append, node sampling, and edge sampling are three different schemes for recording path information. Savage et al. [13] proposed to use edge sampling and distance as path information, where an *edge* instead of the individual node in the path and the *distance* measured from the marking router to the victim are recorded in packets. Interested readers should refer to [13] for more details regarding packet marking and path reconstruction procedures.

2.2. Issues in choosing probability

Traceback is a game between attackers and victims. Attackers may use spoofing and may limit the number of attacking packets to hide their identity. On the other hand, victims may choose a proper marking scheme to pinpoint the attacker(s). In the following, we will argue that it is very difficult for victims to determine a proper marking probability for efficient PPM, since many issues are involved.

Consider an attack path $\mathscr{A} = (a, r_1, r_2, \ldots, r_D, v)$, where a and v denote the attacker and the victim of a DoS incident, and $r_i\,(i = 1,2,\ldots,D)$ denote D routers in the attack path.

2.2.1. At-least-one-marking per router

Let p_i represent the marking probability of router r_i. Define the *leftover* probability for router r_i, denoted by α_i, to be the probability that an attacking packet has lastly been marked at router r_i and nowhere further down the path. For victim v, α_i is the probability that allows v to learn that router r_i is on the attack path by examining this arriving packet. It can be shown that

$$\alpha_i = \begin{cases} p_i \prod_{j=i+1}^{D} (1-p_j) & \text{for } 1 \leqslant i < D, \\ p_D & \text{for } i = D. \end{cases} \quad (1)$$

All routers have a fixed marking probability p under PPM. Through Eq. (1), we have

$$\alpha_i = p(1-p)^{D-i} \quad \text{for } 1 \leqslant i \leqslant D. \quad (2)$$

Therefore, the leftover probability is geometrically smaller the closer it is to the attacker, *i.e.*,

$$\alpha_1 < \alpha_2 < \cdots < \alpha_D.$$

In other words, route r_1 has the least chance whereas route r_D has the biggest chance to pass its marking to victim v. The victim v has no way to find out that route r_i is on the attack path if none of the arriving packet carries a marking left by r_i. Therefore, the victim must collect at-least-one-marking from each router along the attack path in order to construct the attack path. Let N denote the total number of attack packets (attack volume) from an attacker to a victim. To satisfy the requirement of at-least-one-marking per router, any successful traceback by PPM requires that

$$N\alpha_1 = Np(1-p)^{D-1} \geqslant 1. \quad (3)$$

Fig. 1 shows the values of α_1 (leftover probability for r_1) with respect to p and D. It can be seen that α_1 is a bell shape function of p. Moreover, one can show that its peak value occurs at $p = 1/D$. Since D (the number of routers between a and v) is usually not known to victims, it is difficult for users to determine the optimal marking probability in advance. One approach is to choose a small p as suggested by Fig. 1. However, a subtle attacker may reduce his/her attack volume in order to shrink the range of feasible p for a successful traceback.

2.2.2. Spoofed packets

The probability that a packet arriving at the victim without any marking inscribed by any router along the path (*unmarked probability*) is

$$\alpha_0 = (1-p)^D, \quad (4)$$

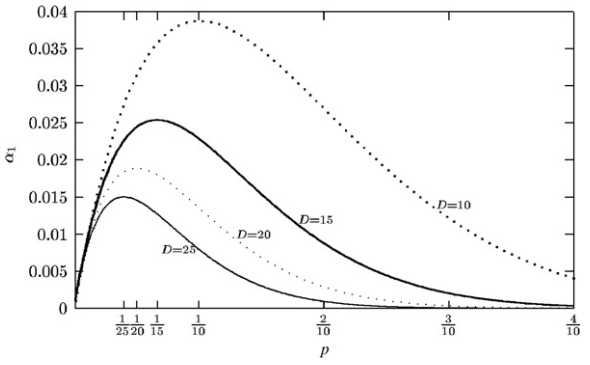

Fig. 1. Leftover probability (α_1) for r_1.

which is obtained by

$$1 - \sum_{i=1}^{D} \alpha_i.$$

In addition to spoof source address, attackers may spoof the marking field with a false value in order to hide themselves or the attack path. This is referred to as *spoofed marking attack*. If a packet is not marked by any router along the path, the marking field spoofing may cause false information during the traceback. The unmarked probability is a measure of the effect of spoofed marking attack. Fig. 2 shows the unmarked probability (α_0) for a packet with respect to p and D. It is clear that α_0 is a decreasing function of p.

2.2.3. Uncertainty

Packets with spoofed marking field also introduce *uncertainty* in traceback, which was first studied by Park and Lee [18]. Fig. 3 helps us to grasp the idea of uncertainty. Consider an attack path $\mathscr{A} = (a, r_1, r_2, \ldots, r_D, v)$. Attacker a may choose to

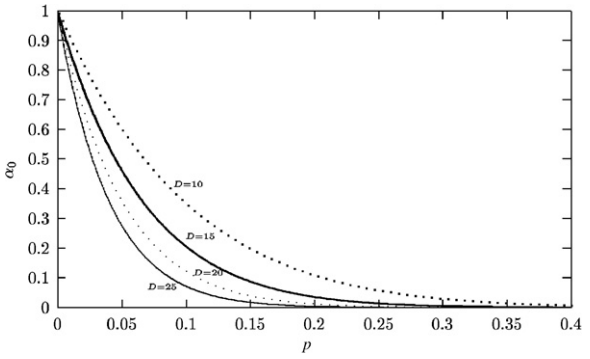

Fig. 2. Unmarked probability (α_0).

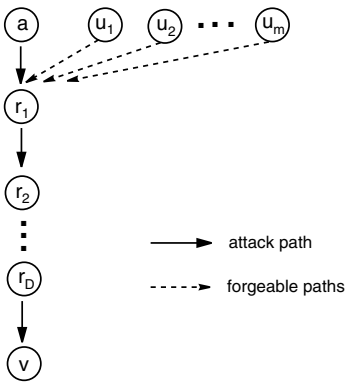

Fig. 3. Forgeable paths.

spoof the marking field with the edge (u_1, r_1). If this spoofed packet is not marked by any router along the attack path, victim v may conclude that u_1 is the source of attack, since the packet is indistinguishable from the real packets originating at u_1 and arriving at v. Similar conclusions may be drawn for $u_2, u_3, \ldots$ and u_m.

Hence, a traceback may give m false sources of attack in addition to the real one (*i.e.*, attacker a). The *uncertainty* factor (denoted by m) is the number of false attack sources (other than the real one) identified by a traceback [18]. To maximize entropy, attacker a sends spoofed packets with marking field (u_i, r_1), where $1 \leqslant i \leqslant m$, as if they are from m different sources, each with the same probability.

We have learned that router r_1 has the least leftover probability among all routers. The best scenario that attack a can expect is that all spoofed packets are unmarked and these packets appear as marked by route r_1 when they arrive at victim v. This results in

$$m\alpha_1 = \alpha_0. \tag{5}$$

Therefore, for a given marking probability p, the maximal uncertainty factor of a DoS attack has been shown [18] to be

$$m = \frac{1}{p} - 1, \tag{6}$$

which is obtained by solving Eq. (5).

We can see that uncertainty decreases as p increases. Therefore, users tend to choose a large p in order to diminish the effect of spoofed packets. In fact, the victim may choose $p = 1$ to try to completely eliminate the uncertainty. However, traceback by PPM would not work under this condition, since under such a condition all packets arriving at the victim would bear the marking of the last router (r_D) on the attack path.

The uncertainty is a key issue in a successful traceback. A determinant factor of uncertainty is the marking probability (p). The victim tries to minimize uncertainty with a large p, while the attacker tries to maximize uncertainty with various spoofed packets. Moreover, as mentioned above, the at-least-one-marking per router constraint (Eq. (3)) also affects the victim's choice of p. The higher the value p the victim chooses, the higher a value N is required for a successful traceback. However, the attacker controls the value N.

Analyzing the uncertainty of PPM under DDoS attack is more complicated. Given a desired attack

volume N, the attacker may mount M separate attacks each with N/M packets in a DDoS attack. Even without spoofed packets, the victim needs to detect M attack paths. With spoofed packets, uncertainty will be amplified through DDoS attack. It has been shown in [18] that a large M results in a high amplification. Thus, PPM has a very limited application in the case of DDoS attack due to the amplification of uncertainty.

3. Dynamic probabilistic packet marking

PPM uses a fixed probability in marking packets. As we have seen in Section 2.2, a small p would enable a traceback with a low attack volume (N). However, a small p would lead to a large uncertainty. The major cause of this conflict is the uneven leftover probability for routers along the attack path. In the following, we will present a new packet marking scheme based on dynamic probability, called dynamic probabilistic packet marking (DPPM), where the marking probability of a packet is determined dynamically as a function of traveling distance of the packet.

One approach to minimize the number of packets required for a successful traceback is to maintain a uniform leftover probability for all routers on the attack path. In addition, the uncertainty introduced by spoofed marking may be removed completely if every packet is marked at least once along the attack path. Our proposed DPPM meets both conditions.

To achieve a uniform leftover probability, routers should decrease the marking probability as a packet travels along the path. Instead of a fixed p, under DPPM, each router uses a different marking probability to mark packets. A router chooses a high marking probability if the packet is just sent out from its source. On the other hand, a route chooses a low marking probability if the packet is far away from its source. More precisely, our proposed DPPM works in the following way: For a given attack path, let i ($1 \leqslant i \leqslant D$) be the traveling distance of a packet w from its source. Router r_i chooses its marking probability

$$p_i = 1/i$$

to mark packet w. One question that needs to be answered is: for each arriving packet, how can a route determine its traveling distance from its source? We will answer this in the following section.

3.1. Determination of distance

Our dynamic marking scheme is based on the traveling distance of a packet from its source (origin). One challenge we have to face is how to determine the traveling distance of each packet. Our solution lies in the Time-to-live (TTL) value in the IP header. The TTL serves two purposes. It limits the lifetime of an IP datagram, and it also terminates internet routing loops. The source node of a packet sets the TTL value to a default initial number, which is system and protocol dependent. Some default TTL values for various systems and protocols are shown in Appendix A. They were compiled from [19,20] and from some of our experiments.

As a packet travels through routers on the network, each router decrements TTL by one [21]. Routers drop any packet with a value of zero in its TTL field. If a router knows the initial TTL value of a packet, then the traveling distance of that packet could be calculated accordingly. One question remains: how can a router find out the initial TTL value for every packet passing by? One solution to this is to require all hosts to use the same initial TTL value as defined by assigned numbers [22]. For all systems to comply with this request may take time. However, a further look at Appendix A leads us to a quick solution, which also enables incremental deployment of our scheme. Although systems may use different TTL values in different protocols, Appendix A shows that most initial TTL values fall in the set of $\mathscr{S} = \{32, 64, 126, 255\}$. Recent studies [23–25] have shown that very few packets travel more than 25 hops (routers). Hence, the most likely initial TTL value for a packet with a TTL value of 47 is 64. In addition, this packet is most probably at a distance of 17 ($64 - 47$) from its origin. Therefore, a router can deduce the initial TTL value of any packet in the following way: Let t be the TTL value of a packet arriving at a router. Its initial TTL value should be the *least* value in set $\mathscr{S}$ that is equal to or greater than t. More precisely, let s be the size of $\mathscr{S}$. We sort all members in $\mathscr{S}$ in an increasing order, and express $\mathscr{S}$ as

$$\mathscr{S} = \{v_i | 1 \leqslant i \leqslant s \text{ and } v_i < v_j \text{ for } 1 \leqslant i < j \leqslant s\}.$$

Then, the most probable initial TTL value (denoted by t_0) for a packet with a TTL value of t arriving at a route is

$$t_0 = v_i \text{ such that } v_i \in \mathscr{S} \quad \text{and} \quad v_{i-1} < t \leqslant v_i. \tag{7}$$

3.2. Leftover probability and uncertainty

Leftover probability is a good measure for effectiveness of a traceback. We now proceed to evaluate it for DPPM. Recall that the marking probability of a packet at route r_i on attack path $\mathscr{A}$ is $p_i = 1/i$ where i is the traveling distance of that packet from its source for $i = 1, 2, \ldots, D$. It can be seen that

$$\alpha_D = 1/D, \tag{8}$$

since no router follows r_D along the path. The leftover probability for other routers can be computed by Eq. (1):

$$\begin{aligned}\alpha_i &= p_i \cdot \prod_{j=i+1}^{D} (1 - p_j) \\ &= p_i \cdot (1 - p_{i+1}) \cdot (1 - p_{i+2}) \cdots (1 - p_D) \\ &= \frac{1}{i} \cdot \left(1 - \frac{1}{i+1}\right) \cdot \left(1 - \frac{1}{i+2}\right) \cdots \left(1 - \frac{1}{D}\right) \\ &= \frac{1}{D} \quad \text{for } 1 \leqslant i < D. \end{aligned} \tag{9}$$

Therefore, by Eqs. (8) and (9), each router along the attack path has the same probability to leave its marking in every packet that arrives at the victim. In other words, the victim has an equal probability to obtain each router's information along the path despite their distance from the victim. This is a subtle feature of our DPPM, which will be referred to as *constant leftover probability*. Fig. 4 compares the leftover probabilities of PPM with various p and DPPM where $D = 20$. We have seen in Section 2 that spoofed marking may introduce uncertainty in PPM. There are D routers in the attack path. Each one has a leftover probability of $1/D$. Therefore, the unmarked probability for any packet under DPPM is zero, *i.e.*,

$$\alpha_0 = 0. \tag{10}$$

The uncertainty of DPPM is obtained by solving Eq. (5). Since the unmarked probability is zero under DPPM, we conclude that the uncertainty is also zero under DPPM. This means that there is no uncertainty in DPPM, since each packet got a legitimate marking. Therefore, the effectiveness of our DPPM will not be affected by packets with spoofed markings.

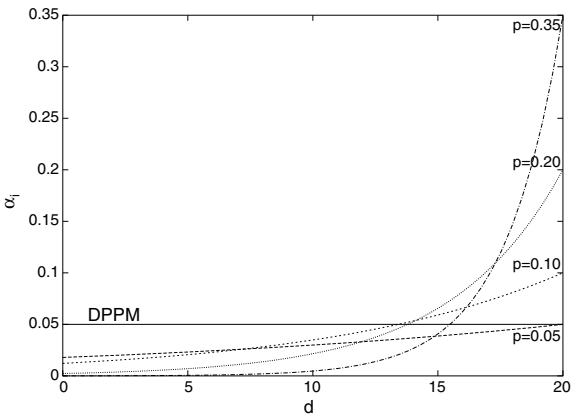

Fig. 4. A comparison of leftover probability for PPM and DPPM.

3.3. Implementation

The only difference between DPPM and PPM is their marking probability. Since implementation issues for PPM have been extensively studied, for example, in [15,13,26,27], we will move our focus to the IPv6 environment and point out one feasible way to implement DPPM.

The traveling distance of a packet is determined by its TTL value. The TTL in IPv6 has a different name: *hop limit*, which serves the same purpose and works the same way as in IPv4. Since IPv6 supports variable header length, we may choose to record all routers traveled in the IP header. However, this would impose a large overhead for traceback. Therefore, we propose to follow PPM by employing a fixed length marking. Savage et al. [13] proposed to overload the *identification* field in the IP header for the marking. There is no identification or similar field available in the IPv6 basic header to carry the marking. However, IPv6 supports extension headers for additional functionality [28]. The hop-by-hop header is one choice to carry the marking information. Options in the hop-by-hop extension header are represented in type-length-value (TLV) format [28]. Once we define a new option type for DPPM, a marking information can be encoded into the TLV format.

4. Performance analysis

In this section, we will further compare the performance of DPPM to PPM. Almost all of our findings indicate that DPPM is superior to PPM.

4.1. Minimal number of packets required for a traceback

To satisfy the requirement of at-least-one-marking per router, a victim needs to collect a certain

number of packets. The minimal number of packets required for a successful traceback by both PPM and DPPM, denoted by N_{ppm} and N_{dppm}, respectively, depends on the leftover probability. To fulfill at-least-one-marking per router, PPM imposes (cf. Eq. (3)) that $N_{\text{ppm}}p(1-p)^{D-1} \geqslant 1$. Thus, for a fixed p and D, PPM requires

$$N_{\text{ppm}} \geqslant \frac{1}{p(1-p)^{D-1}}. \tag{11}$$

On the other hand, we learned from Eqs. (8) and (9) that $\alpha_i = 1/D$ for all routers on the attack path. Therefore, we conclude that

$$N_{\text{dppm}} \geqslant D \tag{12}$$

for DPPM. In fact, there is a difference between the minimal number of packets (N_{ppm}) and its expected value. Savage et al. [29,13] have shown that the expected value of N_{ppm} is

$$E(N_{\text{ppm}}) \leqslant \frac{\ln D}{p(1-p)^{D-1}},$$

which is derived by using the least leftover probability (α_1) and the model of the *coupon collecting problem* [30, Chapter 8] . When using DPPM, we have a fixed leftover probability for all routers. Therefore, we can also apply the coupon collecting model and obtain the expected value of N_{dppm} to be

$$E(N_{\text{dppm}}) \approx D \ln D.$$

It can be seen that the difference between the minimal number of packets and its expected value is a factor of $\ln D$ for both PPM and DPPM. For simplicity, we will in the following consider the minimal number of packets instead of its expected value. Table 1 displays some numerical values of N_{ppm} and N_{dppm} for certain setups. The table clearly shows that DPPM always needs a lower number of packets to get its job done. The difference between PPM and DPPM increases if a lower uncertainty is preferred when employing PPM.

4.2. Uncertainty

It has been shown in [18] that the maximal uncertainty of PPM is $m = 1/p - 1$. We have seen in Section 3.2 that the unmarked probability is zero (*i.e.*, $\alpha_0 = 0$) for DPPM, and hence there is no uncertainty. In other words, we have

$$m = 0 \tag{13}$$

Table 1
Minimal numbers of packets required by PPM and DPPM

PPM	D					m
p	10	15	20	25	30	
0.01	110	116	122	128	134	99
0.02	60	67	74	82	90	49
0.04	37	45	55	67	82	24
0.06	30	40	55	74	101	14.67
0.08	27	41	61	93	141	11.5
0.10	26	44	75	126	213	9
0.20	38	114	347	1059	3232	4
0.30	83	492	2925	17,400	103,523	2.33
0.35	138	1189	10,247	88,311	761,106	1.86
DPPM	10	15	20	25	30	0

for DPPM under the spoofed marking attack. This suggests that DPPM enables us to pinpoint the exact attacker under DoS attack. On the other hand, under the spoofed marking attack PPM may give a few sites for the possible attacker. The last column in Table 1 displays uncertainty values for some PPM setups.

4.3. Overhead on routers

Each marking poses some cost to a router. We now proceed to compare the overhead (with respect to no marking) of PPM and DPPM. It is expected that a marking generated by DPPM costs more than one generated by PPM, because DPPM needs to find the traveling distance of the packet in order to determine its marking probability. However, this is not necessarily true, because routers have to examine and decrease the TTL value by one for each arriving packet. The traveling distance of a packet (hence, marking probability) can be obtained by a table lookup when routers examine the TTL. Therefore, the cost difference between a marking generated by DPPM and one generated by PPM is very small. Both can be achieved in about the same amount of time. For simplicity, we use number of markings performed as our measurement for overhead.

Let us consider a DoS attack with D routers between the attacker and the victim. We examine two types of overhead: The *individual overhead* is the cost experienced by a route along the attack path. The *total overhead* is the total cost summed over all D routers.

Let o_{ppm} and o_{dppm} denote the individual overhead of PPM and DPPM, respectively. Under

PPM, each router uses a fixed probability p to mark packets. If there are N packets in a DoS attack, the individual overhead for each router is

$$o_{\text{ppm}} = Np. \tag{14}$$

On the other hand, under DPPM, router r_i uses a probability of $1/i$ to mark packets, where $i = 1, 2, \ldots, D$. Therefore, the individual overhead for router r_i is

$$o_{\text{dppm}} = N/i. \tag{15}$$

Fig. 5 compares the individual overheads of PPM and DPPM, where $N = 100{,}000$, $D = 25$ and $p = 0.35$. It can be seen that all routers under PPM have the same individual overhead. On the other hand, the first two routers under DPPM suffer a high overhead. However, it drops very rapidly. The high overhead for the first two routers can be viewed as an advantage of DPPM, since any router experiencing a sudden surge of workload may indicate some sort of DoS attack. Hence, proper action can be taken at the first place. In general, for a fixed marking probability, the individual overhead of PPM remains a constant value; however, the individual overhead of DPPM is inversely proportional to its traveling distance from the attacker as indicated by Eqs. (14) and (15).

The total overhead is the price paid by all routers along the attack path. In general, the total overhead is a cost index of the Internet community to perform marking for traceback. Let O_{ppm} and O_{dppm} denote the total overhead of PPM and DPPM, respectively. There are D routers in an attack path. Therefore, we have

$$O_{\text{ppm}} = NpD. \tag{16}$$

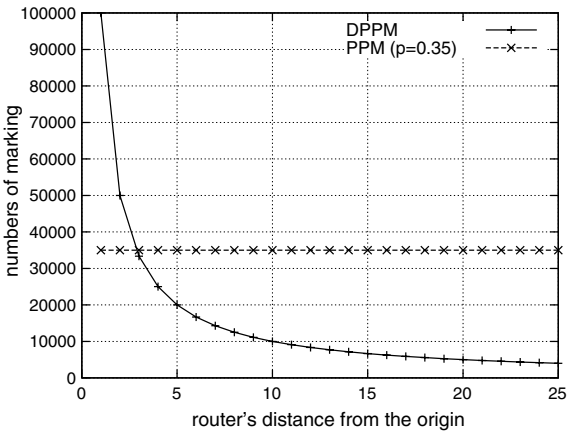

Fig. 5. A comparison of individual overheads.

For DPPM, the total overhead is obtained by summing over all D terms:

$$O_{\text{dppm}} = N\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{D}\right) = NH_D, \tag{17}$$

where H_D is the Dth harmonic number (see, for example, [30, Section 6.3]). Table 2 compares the total overhead of PPM and DPPM, where the overhead has been normalized to the number of times (frequency) each packet has been marked in the network. The normalized total overhead is defined as the ratio of the total overhead (O_{ppm} or O_{dppm}) to the attack volume (N). This is because the value of N is controlled by the attacker, which is not known to the network routers. The uncertainty factor of PPM also depends on p. The last column in Table 2 shows the uncertainty for our reference.

Some interesting points deserve our attention. First, for a fixed attack path length (D), the total overhead of PPM (O_{ppm}) increases linearly with p, whereas the total overhead of DPPM (O_{dppm}) remains fixed. This is captured by Eqs. (16) and (17). Second, in order to lower the total overhead, one tends to choose a small p. However, this also introduces a large uncertainty, which becomes a major challenge to the application of PPM. Finally, O_{ppm} will be larger than O_{dppm} if we want to keep uncertainty to be less than 2.33 (*i.e.*, $p \geqslant 0.3$).

4.4. Distributed DoS attack

Given a desired attack volume N, an attacker may mount a DDoS attack from M different sites each with a volume of N/M. Although this may be considered as M separate DoS attacks, two issues

Table 2
Normalized total overhead and uncertainty of PPM and DPPM

PPM	D					m
p	10	15	20	25	30	
0.01	0.1	0.15	0.2	0.25	0.3	99
0.02	0.2	0.3	0.4	0.5	0.6	49
0.04	0.4	0.6	0.8	1	1.2	24
0.05	0.5	0.75	1	1.25	1.5	19
0.0667	0.67	1	1.33	1.67	2	13.99
0.10	1	1.5	2	2.5	3	9
0.20	2	3	4	5	6	4
0.30	3	4.5	6	7.5	9	2.33
0.35	3.5	5.25	7	8.75	10.5	1.86
0.40	4	6	8	10	12	1.5
0.50	5	7.5	10	12.5	15	1
DPPM	2.93	3.32	3.60	3.82	4.00	0

deserve our further attention. In general, users of PPM will choose a large p to ease the uncertainty problem. For a crafty attacker, he/she would choose to mount a DDoS attack from a large number of sites in order to hide himself/herself. Under PPM, with a fixed probability p and attack volume N, the attack volume from each site (N/M) may fall below the constraint of at least one marking per router (N_{ppm}), which makes the DDoS attack become untraceable. Therefore, PPM may not be able to get its job done if its marking probability is not adjusted accordingly and properly. Adjusting p properly to the network conditions is a major challenge to the application of PPM. On the other hand, DPPM performs much better, since its N_{dppm} is much less than that of PPM (cf. Table 1) and no human attention is needed to adjust marking probability.

As we just saw when using PPM one needs to lower p in order to combat DDoS attacks. This also causes an amplification effect [18], where the uncertainty induced by a DDoS with a total of N attacking packets from M different sites is amplified by some factor with respect to a DoS attack with N packets. It has been shown in [18] that the amplification could be up to 20 if the attack path length is sufficiently large. Since there is no uncertainty in DPPM, DPPM suffers no amplification under DDoS attacks.

5. Challenge on spoofed TTL value attack

We have learned that the performance of DPPM will not be affected by spoofed marking field. A subtle attacker may observe one possible weakness of the DPPM: the marking probability of any packet is completely determined on its traveling distance from its origin, which is equivalent to the TTL value of that packet. By spoofing the initial TTL value of a packet, any attacker may have a chance to defeat the DPPM. For example, by sending a packet with a TTL value of 129, a crafty attacker would definitely get away without any trace, since the router would deduce that the packet is at a distance of 126 ($= 255 - 129$) from its origin and marks the packet with a probability of 1/126. In the following we will discuss how to prevent this new *spoofed TTL value attack*.

5.1. Unified initial TTL

A router examines and decreases the TTL value by one when it forwards a passing packet [21]. To thwart the spoofed TTL value attack, we propose a *unified* initial TTL value, denoted by T_0. Each machine should set the TTL value to T_0 when it sends an IP packet. If a router sees a packet with a TTL value greater than T_0, it should rewrite the TTL value as T_0 and mark this packet as it is at a distance of 1 from its origin. It is clear that only attackers or compromised routers could set a TTL value greater than T_0 if all systems comply with the unified initial TTL value rule. The best way to handle such a packet is to view it as one hop away from its attacker and mark it with a probability of 1. A good choice for T_0 would be 32, since studies have shown that most applications on the Internet are within this limit [23–25]. In the following, we will show that DPPM still provides a uniform leftover probability and suffers very little uncertainty under the spoofed TTL value attack.

5.2. Minimal number of packets required for traceback

With a unified initial TTL value, attackers still may try to beat DPPM by sending packets with spoofed TTL values. However, they should choose TTL values smaller than T_0, otherwise the attack will be detected and corrected as we mentioned before. Assume that an attacker sends a packet with a spoofed TTL value that is by z smaller than T_0 through an attack path $\mathscr{A}$, whereby $0 \leqslant z < T_0$, *i.e.*, $\text{TTL} = T_0 - z$. No router can distinguish such an attacking packet from normal ones. Router r_1 views this packet as originated at $1 + z$ hops away and consequently marks it with a probability of $p'_1 = 1/(1+z)$. Similarly, router r_i ($1 \leqslant i \leqslant D$) will mark this packet with a probability of

$$p'_i = \frac{1}{i+z}. \tag{18}$$

The leftover probability for routers can be calculated similarly as in Eq. (9). Hence, we have

$$\alpha'_i = \frac{1}{D+z}. \tag{19}$$

Eq. (19) indicates that all routers still have a uniform leftover probability under the spoofed TTL value attack. Moreover, the expected minimal number of packets required for successful traceback, denoted by N'_{dppm}, can be obtained immediately from Eq. (19). Thus, we have

$$N'_{\text{dppm}} \geqslant D + z, \tag{20}$$

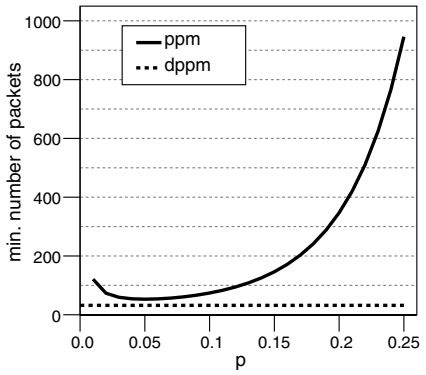

Fig. 6. A comparison of minimal numbers of packets required by PPM and DPPM.

which is an increase of z compared to Eq. (12). Fig. 6 compares the minimal numbers of packets required for traceback by PPM and by DPPM, where we fix D to 20 and z to 12. The figure clearly shows that DPPM still needs a lower number of packets to get its job done even under the spoofed TTL value attack. By Eqs. (18) and (19), both the marking and the leftover probability decrease under the spoofed TTL value attack. This would be an advantage for DPPM, since both individual and total overhead decrease. However, we have to pay such an advantage with the price of a little uncertainty. In the following, we will show that DPPM suffers an uncertainty of z under the spoofed TTL value attack.

5.3. Uncertainty and overhead

The unmarked probability is no longer zero under DoS attack with spoofed TTL values. With the new leftover probability (Eq. (19)), the unmarked probability is

$$\alpha'_0 = 1 - \sum_{i=1}^{D} \alpha'_i = \frac{z}{D+z}, \tag{21}$$

which is no longer zero. This suggests that DPPM will suffer from the spoofed TTL value attack. With leftover and unmarked probabilities (Eqs. (19) and (21)), the uncertainty can be obtained by solving Eq. (5), *i.e.*, by solving

$$m' \frac{1}{D+z} = \frac{z}{D+z}.$$

Therefore, we have

$$m' = z. \tag{22}$$

In other words, DPPM may suffer an uncertainty of z under a spoofed TTL value attack combined with spoofed marking, where z is the difference between T_0 and the spoofed packet's TTL value. It is clear that an upper bound for uncertainty is T_0. However, a tighter upper bound for uncertainty is $T_0 - D$, since no attacking packet can reach the victim if its initial TTL value is set below D. On the other hand, the lower bound for z is 0. Fig. 7 compares the uncertainty of PPM and DPPM, where we fix D to 20 and z to 12. We see that DPPM outperforms PPM for p less than 0.077 even in the worst case ($z = 12$) of DPPM. With Eq. (18), we can show that the total overhead for DPPM (denoted by O'_{dppm}) under a spoofed TTL value attack is

$$O'_{\text{dppm}} = N(H_{D+z} - H_z). \tag{23}$$

Fig. 8 compares the total overhead of PPM and DPPM, where we fix D to 20, z to 0, 6 and 12. It can be seen that the total overhead of DPPM decreases as the value z increases. Furthermore,

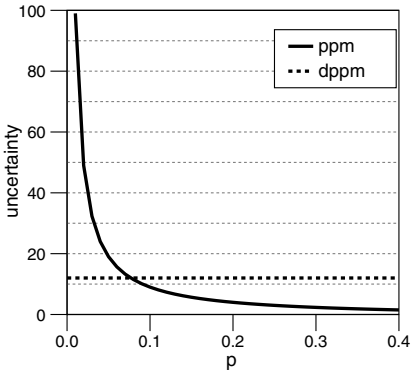

Fig. 7. A comparison of uncertainty of PPM and DPPM under spoofed TTL value attack.

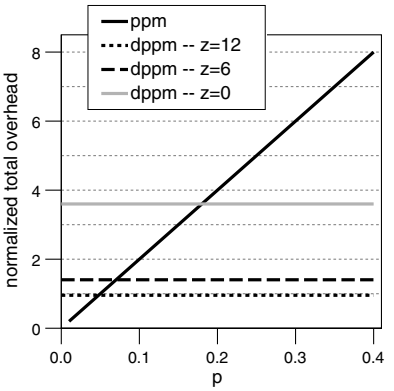

Fig. 8. A comparison of total overheads of PPM and DPPM.

for p greater than 0.05, PPM suffers more total overhead compared to DPPM with z being 12.

6. Legacy routers

Any proposed packet marking scheme should allow incremental deployment and backward compatibility. In this section, we consider the robustness of DPPM in the presence of legacy routers that do not support packet marking. New routers implementing our DPPM scheme will most likely be deployed in clusters. In the following discussion, we assume that each attack path consists of several clusters interleaved with routers that do or do not support packet marking. We will examine unmarked probability and total overhead.

6.1. Unmarked probability

We first consider the case where an attack path consists of two clusters of routers with DPPM mechanism. Fig. 9 displays a schematic view, where routers $d_0, d_0+1, \ldots, d_1$ form cluster C_1 and routers $d_2, d_2+1, \ldots, d_3$ form cluster C_2. Routers in C_1 and C_2 support packet marking whereas other routers do not support packet marking. To account for unmarked probability, we compute leftover probability first. It can be seen that

$$\alpha_{d_3} = \frac{1}{d_3},$$

since no router supports marking afterwards. The leftover probability for router $d_3 - 1$ is

$$\alpha_{d_3-1} = \frac{1}{d_3-1}\left(1-\frac{1}{d_3}\right) = \frac{1}{d_3}.$$

It can be shown that all routes in cluster C_2 have the same leftover probability, which is

$$\alpha_i = \frac{1}{d_3} \quad \text{for } d_2 \leqslant i \leqslant d_3. \tag{24}$$

The leftover probability for router d_1 is

$$\begin{aligned}\alpha_{d_1} &= \frac{1}{d_1}\left(1-\frac{1}{d_2}\right)\left(1-\frac{1}{d_2+1}\right)\cdots\left(1-\frac{1}{d_3}\right)\\ &= \frac{d_2-1}{d_1}\frac{1}{d_3}.\end{aligned}$$

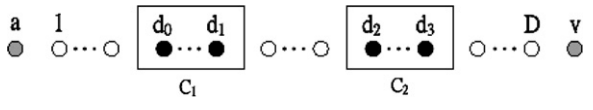

Fig. 9. Incremental deployment: 2 supporting clusters C_1 and C_2.

Similarly, leftover probabilities for all other routers in cluster C_1 can be derived as

$$\alpha_i = \frac{d_2-1}{d_1}\frac{1}{d_3} \quad \text{for } d_0 \leqslant i \leqslant d_1. \tag{25}$$

From Eqs. (24) and (25), we can see that a router in cluster C_1 has a larger chance to leave its marking than any other router in cluster C_2, since $d_2 > d_1$. The main reason behind this is that none of the legacy routers between d_1 and d_2 overwrites the marking. There are $(d_1 - d_0 + 1)$ and $(d_3 - d_2 + 1)$ routers in cluster C_1 and C_2, respectively. Therefore, the expected marking probability (denoted by α) for any packet arriving at the victim is

$$\alpha = (d_1 - d_0 + 1)\frac{d_2-1}{d_1}\frac{1}{d_3} + (d_3 - d_2 + 1)\frac{1}{d_3}.$$

Consequently, the unmarked probability is

$$\alpha_0 = \frac{d_0-1}{d_1}\frac{d_2-1}{d_3}, \tag{26}$$

which is obtained by $1 - \alpha$.

Let us consider a special case where cluster C_1 starts with router 1, *i.e.*, $d_0 = 1$. From Eq. (26), the unmarked probability for any packet is exactly 0. This is the same feature we have come to know in Section 3.2. An implication of this is that every packet arriving at the victim carries a marking to reveal its path, even though there are some routers that do not support packet marking. This is quite an advantage of DPPM over PPM. However, DPPM will suffer from some unmarked probability if cluster C_1 does not start with router 1, *i.e.*, router 1 does not support packet marking.

Fig. 10 shows the case where an attack path consists of three clusters (C_1, C_2 and C_3) of routers with DPPM mechanism. We can show (in accordance with what we did in the case of two clusters) that the leftover probability is

$$\alpha_i = \begin{cases} \frac{1}{d_5} & \text{for } d_4 \leqslant i \leqslant d_5,\\ \frac{d_4-1}{d_3}\frac{1}{d_5} & \text{for } d_2 \leqslant i \leqslant d_3,\\ \frac{d_2-1}{d_1}\frac{d_4-1}{d_3}\frac{1}{d_5} & \text{for } d_0 \leqslant i \leqslant d_1. \end{cases} \tag{27}$$

The unmarked probability can be computed accordingly as

$$\alpha_0 = \frac{d_0-1}{d_1}\frac{d_2-1}{d_3}\frac{d_4-1}{d_5}. \tag{28}$$

For an attack path consisting of n clusters of routers that support the DPPM mechanism, we can show that the unmarked probability will be

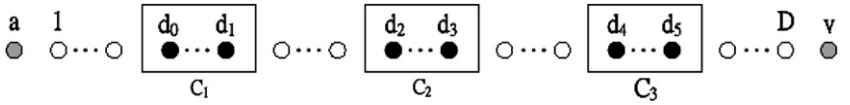

Fig. 10. Incremental deployment: 3 supporting clusters C_1, C_2 and C_3.

$$\alpha_0 = \prod_{i=1}^{n} \frac{d_{2i-2} - 1}{d_{2i-1}},$$

where d_{2i-2} and d_{2i-1} are the indices of first and last router for cluster i, respectively. Furthermore, we can show that DPPM provides the following subtle features: (a) all routers in the same cluster have the same leftover probability; (b) routers in a cluster closer to the attacker have a higher leftover probability than routers in a cluster farther away from the attacker; and (c) the unmarked probability is caused by legacy routers in front of the first cluster, other legacy routers do not affect the unmarked probability. Feature (a) indicates that, even in the presence of legacy routers, DPPM maintains the property of constant leftover probability (Eqs. (8) and (9)). Feature (b) enables us to have a higher chance to identify routers that are closer to the attacker compared to those that are farther away. This would allow us to immediately take some proper action. Feature (c) suggests that boundary routers should be our first choice for incremental deployment of DPPM.

Recall that the marking probability for each router in PPM is a fixed value. It can be shown that the unmarked probability of PPM in the presence of legacy routers is

$$\alpha_0 = (1-p)^l, \tag{29}$$

where l is the total number of routers supporting PPM in the attack path. Tables 3 and 4 display the unmarked probabilities of DPPM and PPM in the presence of legacy routers, where the attack path consists of 3 supporting clusters that are of (almost) equal lengths as depicted in Fig. 10. We have learned that DPPM may achieve a zero unmarked probability in the best case, *i.e.*, if the first supporting cluster starts with router r_1. However, more general cases should also be considered. In Table 3, we consider an attack path of length 18. The notation $6(2,2,2)$ means that there are 6 routers that support packet marking and there are 2 routers in each of C_1, C_2 and C_3. The remaining 12 routers are evenly divided into 4 clusters that do not support packet marking. Similarly, in Table 4, we consider an attack path of length 30. The notation $20(7,7,6)$ means that there are 20 routers that support packet marking and the number of routers in C_1, C_2 and C_3 are 7, 7, and 6, respectively. The remaining 10 routers are divided into 4 unsupporting clusters containing 2, 2, 3, and 3 routers, respectively. We can see that, in general, DPPM results in a smaller unmarked probability compared to PPM with p being less than 0.1. This indicates that DPPM will suffer less by spoofed packets compared to PPM. On the other hand, PPM may reduce the effect of spoofed

Table 3
Unmarked probability with legacy routers: 3 supporting clusters with $D = 18$

PPM	l		
p	6(2,2,2)	9(3,3,3)	12(4,4,4)
0.01	0.9415	0.9135	0.8864
0.02	0.8858	0.8337	0.7847
0.04	0.7828	0.6925	0.6127
0.06	0.6899	0.573	0.4759
0.08	0.6064	0.4722	0.3677
0.1	0.5314	0.3874	0.2824
0.15	0.3771	0.2316	0.1422
0.2	0.2621	0.1342	0.0687
0.25	0.178	0.0751	0.0317
0.3	0.1176	0.0404	0.0138
0.35	0.0754	0.0207	0.0057
DPPM	0.416	0.224	0.09

Table 4
Unmarked probability with legacy routers: 3 supporting clusters with $D = 30$

PPM	l		
p	10(4,3,3)	15(5,5,5)	20(7,7,6)
0.01	0.9044	0.8601	0.8179
0.02	0.8171	0.7386	0.6676
0.04	0.6648	0.5421	0.442
0.06	0.5386	0.3953	0.2901
0.08	0.4344	0.2863	0.1887
0.1	0.3487	0.2059	0.1216
0.15	0.1969	0.0874	0.0388
0.2	0.1074	0.0352	0.0115
0.25	0.0563	0.0134	0.0032
0.3	0.0282	0.0047	0.0008
0.35	0.0135	0.0016	0.0002
DPPM	0.426	0.2138	0.1056

packets when a large p is chosen. However, we have to pay for this with a higher overhead and a larger volume number (N_{ppm}) in order to get the job done.

6.2. Total overhead

The total overhead of DPPM can be obtained by summing the overheads of all supporting routers, which can be expressed as

$$O_{\text{dppm}} = N\sum_{i=1}^{n}(H_{d_{2i-1}} - H_{d_{2i-2}-1}),$$

where d_{2i-2} and d_{2i-1} are the indices of the first and the last router for cluster i, respectively. On the other hand, the total overhead of PPM can be expressed as

$$O_{\text{ppm}} = Npl,$$

Table 5
Normalized total overhead with legacy routers: 3 supporting clusters with $D = 18$

PPM	l		
p	6(2,2,2)	9(3,3,3)	12(4,4,4)
0.01	0.06	0.09	0.12
0.02	0.12	0.18	0.24
0.04	0.24	0.36	0.48
0.06	0.36	0.54	0.72
0.08	0.48	0.72	0.96
0.1	0.6	0.9	1.2
0.15	0.9	1.35	1.8
0.2	1.2	1.8	2.4
0.25	1.5	2.25	3.0
0.3	1.8	2.7	3.6
0.35	2.1	3.15	4.2
DPPM	0.799	1.335	2.040

Table 6
Normalized total overhead with legacy routers: 3 supporting clusters with $D = 30$

PPM	l		
p	10(4,3,3)	15(5,5,5)	20(7,7,6)
0.01	0.1	0.15	0.2
0.02	0.2	0.3	0.4
0.04	0.4	0.6	0.8
0.06	0.6	0.9	1.2
0.08	0.8	1.2	1.6
0.1	1.0	1.5	2.0
0.15	1.5	2.25	3.0
0.2	2.0	3.0	4.0
0.25	2.5	3.75	5.0
0.3	3.0	4.5	6.0
0.35	3.5	5.25	7.0
DPPM	0.859	1.430	2.050

where l is the total number of routers supporting PPM in the attack path. Tables 5 and 6 display the total overhead of DPPM and PPM in the presence of legacy routers with 3 supporting clusters of (almost) equal lengths. We can see that DPPM results in less total overhead compared to PPM with p being greater than 0.1. Recall that a large p is required by PPM in order to achieve a smaller unmarked probability. Therefore, PPM still faces the challenge of how to select a proper marking probability under legacy routers. This is similar to what we saw in Section 4.

7. Related work

DoS attack is one of the hardest network security problems [31], because it is simple to implement and hard to prevent. Recently, Douligeris and Mitrokotsa [9] have done an excellent survey on attack and defense mechanisms for DoS and DDoS. In the following, we briefly review some other related work. PPM for IP traceback is one subtle scheme to thwart DoS attacks. The idea of DPPM, to adjust the marking probability dynamically based on the number of hops traveled (hop count), was first proposed by Lee and Liu [32,33]. Independently, Peng et al. [34] have introduced three schemes for adjusting marking probability, one of which is exactly the same as DPPM. However, they proposed to add a special hop count field in order to deduce the hop count, which obviously would introduce another risk of spoofing attack. Jin et al. [35] have proposed a hop-count filtering scheme against spoofed packets in DoS attacks, where the hop-count information was also deduced from the TTL field.

Recent work on defending DoS attacks can be classified into two approaches: router-based (intermediate system) and end-system-based. In the router-based approach, defence mechanisms are installed at the routers in order to block, detect and/or trace back attacking packets. On the other hand, defence mechanisms are installed at the site of the victim in order to block or thwart attacking packets in the end-system-based approach.

Filtering and traceback are two common schemes proposed for the router-based approach. Filtering mechanisms rely on routers' enhancements to detect abnormal traffic pattern or malicious packets and then to foil attacks. This solution operates in an on-line fashion. Ingress filtering [36] is one way to defend DoS by filtering malicious packets, where routers are configured to block arriving packets

with spoofed and/or illegitimate source addresses. This is done most effectively on boundary routers, where address ownership is relatively unambiguous. It is known that attackers can get around the ingress filtering by choosing legitimate border network addresses at random. A more serious problem in ingress filtering is that its effectiveness depends on widespread deployment. However, most ISPs are reluctant to implement this scheme due to administrative burden, router overhead and lack of benefits from their immediate customers. Given the reachability constraints imposed by routing and network topology, the router-based distributed packet filtering (DPF) [37] utilizes routing information to determine if an incoming packet is valid with respect to its source and destination addresses. Experimental results reported [37] that a significant fraction of spoofed packets may be filtered out, whereas those that escape the filter net can be localized to within five candidate sites, which should be easy to trace back. A much stronger source address checking scheme was proposed in SAVE [38], a source address validity enforcement protocol, where routers establish and maintain valid incoming source addresses. With this knowledge, routers can filter out all packets with improper source addresses. Similar to the ingress filter, the success of SAVE hinges on its widespread deployment.

Traceback mechanisms attempt to establish a procedure to track down attack source(s). This can be done in either on-line or off-line fashion. Input debugging and controlled flooding [14] are two variations of link testing for on-line traceback. Input debugging is a feature supported by many routers, which allows a router to filter and identify the input port of packets with a certain attack signature. The victim under attack must develop an attack signature and send it (mostly via some other channels, *e.g.*, telephone) to its upstream router, where the input debugging is applied. This process is repeated recursively on the upstream routers. Traceback by input debugging can be performed either manually or automatically. It is known that the input debugging suffers from considerable management overhead. With a pregenerated map of the Internet, the controlled flooding [14] does not require any support from network operators. This scheme tries to flood links with large bursts of traffic and observes how this perturbs traffic from the attacker. By observing changes in the rate of attack packets, the victim can therefore deduce which link the packets are coming from. Controlled flooding faces two shortcomings: it is a DoS by itself and it requires any victim to have a good map of large sections of the Internet.

ICMP traceback [16] is another proposal for traceback. The main idea of this scheme is as follows: When forwarding packets, each router with a low probability (*e.g.*, 1/10,000) generates a special traceback message that is sent along to the destination. The traceback message contains information regarding the router that generates this message and the adjacent router along the path to the destination. With sufficient traceback messages from sufficient routers along the path, the (attack) packet source and (attack) path can be determined by the destination (victim). One disadvantage of this approach is that the ICMP traceback message may itself be filtered in the Internet and never reach the destination when under attack. To remedy this disadvantage Savage et al. [29,13] proposed PPM for IP traceback, where the marking is embedded in the packet itself. More precisely, they proposed to overload the 16-bit IP(v4) identification field with a path edge sampling marking. One implementation issue of PPM is how to put the 64 bits edge sampling into the IP identification field. Compressed edge fragment sampling scheme has been proposed in [29,13], where the marking was fragmented into multiple packets in order to fit into the IP identification field. By employing the technique of hashing on routers' addresses, Song and Perrig [27] presented an advanced PPM scheme that can cut the marking length to 16 bits, if every victim knows the map of its upstream routers. One should note that our work can be applied to both the original PPM [29,13] and the advanced PPM proposed by Song and Perrig [27].

In contrast to the router-based approach, the end-system-based approach can be deployed immediately, since it does not require any support from the Internet routers. This creates much incentive for network servers (end systems) to implement this approach. Some end-system-based approaches rely on sophisticated resource management mechanisms that must provide accurate resource accounting in real-time. Escort [11] is a new architecture built on the Scout operating system [39] with an end-to-end resource accounting mechanism, which provides protection against resource based DoS attack, such as SYN floods. It works in the following three steps: accounting for all resources consumed by every principal, detecting which principal exceeds the pre-defined limit, and reclaiming the illegitimately

consumed resource. It can be seen that how to reclaim consumed resource is the most crucial step of a successful defence, since removal of the legitimately consumed resource becomes a DoS attack in its own right. To distinguish the illegitimately consumed resource is a major challenge to Escort. The HCF (hot-count filtering) [35] is a second scheme for end-system-based approach, which detects and discards spoofed traffic without any router support. It utilizes the information of source address and TTL value in the IP header for filtering. There are two phases in HCF: the end-system builds an IP-to-hop-count mapping table in the initial phase, then the end-system filters out any arriving packet whose source address does not match with the hop-count stored in the IP-to-hop-count mapping table. The idea to deduce the hop-count is similar to what we utilize in DPPM. HCF faces two shortcomings: the problem of building an accurate IP-to-hop-count table in every end-system, and the possibility of severely suffering from spoofed TTL value attack.

Recently, Yaar et al. [40] have presented the Pi (Path identifier) marking scheme to defend against DDoS attacks. This is a combination of route and end-system-based approaches, where the Pi marking is done by routers and the end-system does the filtering based on the Pi marking. Each router leaves some bits in the Pi marking field, which are part of the router's address. The TTL value in each packet is used to determine what part of the router's address will be used in the marking. Hence, the Pi marking in each packet contains a part of each router's address along the path when it reaches its destination. Filtering is based on the assumption that all the attack packets coming from the same source should go through the same path and hence carry the same Pi markings. To distinguish between good and malicious identifiers, a learning phase is first exercised on the victim. Then the victim applies the filtering mechanism to defend the attacks. The success of Pi marking hinges on the ability to distinguish between good and bad packets (identifiers), which is exactly the goal of the learning phase. However, it is known that Pi marking suffers from a TTL wrapping attack [40].

8. Concluding remarks

Traceback is a subtle scheme to thwart DoS attacks. PPM opens a new avenue for practical IP traceback. Although PPM enables a victim to pinpoint the attacker's origin to within 2–5 equally possible sites, it has been shown that PPM suffers from uncertainty under the spoofed marking attack. The uncertainty can be amplified significantly under distributed DoS attack, which may diminish the effectiveness of PPM.

In this work, we presented a new approach, called DPPM, to further improve effectiveness of PPM. DPPM may completely remove uncertainty under a spoofed marking attack and enable a victim to precisely pinpoint the attacking origin under a DoS attack. Our proposed DPPM can be applied to DDoS attacks with a very limited uncertainty. A subtle feature of DPPM is that it allows incremental deployment. Implementation issues on IPv6 is addressed. Formal analysis indicates that DPPM outperforms PPM in most aspects. We believe that DPPM combined with some end-system-based approach would foil most DoS attacks.

Appendix A. Some default TTL values

OS	ICMP Reply	ICMP Request	TCP SYN	TCP FIN	TCP Data	UDP
Windows 95	32	32	32	32	32	64
Windows 98	128	32	128	128	128	128
Windows 98SE	128	32	128	128	128	128
Windows ME	128	32	128	128	128	128
Windows NT4.0	128	32	128	128	128	128
Windows 2000	128	128	128	128	128	128
Windows XP	128	128	128	128	128	128
FreeBSD 3.4	255	255	64	64	64	64
FreeBSD 4.0	255	255	64	64	64	64
FreeBSD 4.3	255	255	64	64	64	64

Appendix A (*continued*)

OS	ICMP Reply	ICMP Request	TCP SYN	TCP FIN	TCP Data	UDP
Redhat Linux 6.2	255	64	64	64	64	64
Redhat Linux 7.2	255	64	64	64	64	64
Debian Linux 2.2	255	64	64	64	64	64
Linux Kernel 2.0.x	64	64				
Solaris 2.5.1	255	255				
Solaris 2.6	255	255				
Solaris 2.7	255	255				
Solaris 2.8	255	255				
OpenBSD 2.7	255	255				
OpenBSD 2.6	255	255				
NetBSD	255					
BSDI BSD/OS 4.0	255					
BSDI BSD/OS 3.1	255					
HP-UX v10.20	255	255				
HP-UX v11.0	255					
Compaq Tru64 v5.0	64					
Irix 6.5.3	255					
Irix 6.5.8	255					
AIX 4.1	255					
AIX 3.2	255					
ULTRIX 4.2–4.5	255					
OpenVMS v7.1–2	255					

References

[1] Computer Emergency Response Team, CERT Advisory CA-1996-01 UDP Port Denial-of-Service Attack, February 1996. Available from: <http://www.cert.org/advisories/CA-1996-01.html>.

[2] Computer Emergency Response Team, CERT Advisory CA-1996-26 Denial-of-Service Attack via ping, December 1996. Available from: <http://www.cert.org/advisories/CA-1996-26.html>.

[3] Computer Emergency Response Team, CERT Advisory CA-1997-28 IP Denial-of-Service Attacks, December 1997. Available from: <http://www.cert.org/advisories/CA-1997-28.html>.

[4] Computer Emergency Response Team, CERT Advisory CA-1998-01 Smurf IP Denial-of-Service Attacks, January 1998. Available from: <http://www.cert.org/advisories/CA-1998-01.html>.

[5] Computer Emergency Response Team, CERT Advisory CA-2000-01 Denial of Service Developments, January 2000. Available from: <http://www.cert.org/advisories/CA-2000-01.html>.

[6] Computer Emergency Response Team, CERT Incident Note IN-2000-04 Denial of Service Attacks using Nameservers, April 2000. Available from: <http://www.cert.org/incident_notes/IN-2000-04.html>.

[7] D. Moore, G.M. Voelker, S. Savage, Inferring internet denial-of-service activity, in: Proceedings of the 10th USENIX Security Symposium, August 2001, pp. 9–22.

[8] Computer Emergency Response Team, CERT Incident Note IN-99-07 Distributed Denial of Service Tools, July 1999. Available from: <http://www.cert.org/incident_notes/IN-99-07.html>.

[9] C. Douligeris, A. Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-art, Computer Networks 44 (2004) 643–666.

[10] G. Banga, P. Druschel, J. Mogul, A new facility for resource management in server systems, in: USENIX/ACM Symposium on Operation System Design and Implementation, February 1999, pp. 45–58.

[11] O. Spatscheck, L. Peterson, Defending against denial of service attacks in scout, in: 1999 USENIX/ACM Symposium on Operating System Design and Implementation, February 1999, pp. 59–72.

[12] Cisco Systems, Configuring TCP Intercept (Prevent Denial-of-Service Attacks), Cisco IOS Documentation, 1997.

[13] S. Savage, D. Wetherall, A. Karlin, T. Anderson, Network support for IP traceback, IEEE/ACM Transactions on Networking 20 (2) (2001) 226–237.

[14] H. Burch, B. Cheswick, Tracing anonymous packets to their approximate source, in: 2000 USENIX LISA Conference, December 2000, pp. 319–327.

[15] T. Doeppner, P. Klein, A. Koyfman, Using router stamping to identify the source of IP packets, in: Proceedings of the 7th ACM Conference on Computer and Communications Security, November 2000, pp. 184–189.

[16] S.M. Bellovin, ICMP Traceback Messages, March 2000. Available from: <http://www.research.att.com/~smb/papers/draft-bellovin-itrace-00.txt>.

[17] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, B. Schwartz, S.T. Kent, W.T. Stayer, Single-packet IP traceback, IEEE/ACM Transactions on Networking 10 (December) (2002) 721–734.
[18] K. Park, H. Lee, On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack, in: 2001 IEEE INFOCOM Conference, June 2001.
[19] The Swiss Education and Research Network, Default TTL Values in TCP/IP, 1999. Available from: <http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html>.
[20] O. Arkin, S.-S. Group, ICMP Usage in Scanning, 2001. Available from: <http://www.sys-security.com/html/projects/icmp.html>.
[21] J. Postel, RFC 791: Internet Protocol, September 1981.
[22] J. Reynolds, J. Postel, RFC 1700: ASSIGNED NUMBERS, October 1994.
[23] R. Carter, M. Crovella, Server selection using dynamic path characterization in wide-area networks, in: IEEE INFOCOM Conference, April 1997.
[24] W. Theilmann, K. Rothermel, Dynamic distance maps of the Internet, in: Proceedings of the 2000 IEEE INFOCOM Conference, March 2000.
[25] Cooperative Association for Internet Data Analysis, Skitter analysis, 2000. Available from: <http://www.caida.org/tools/measurement/skitter/>.
[26] D. Song, A. Perrig, Advanced and authenticated marking schemes for IP traceback, University of California at Berkeley, Tech. Rep. UCB/CSD-00-1107, June 2000.
[27] D. Song, A. Perrig, Advanced and authenticated marking schemes for IP traceback, in: 2001 IEEE INFOCOM Conference, April 2001.
[28] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) specification, December 1998, RFC 2460.
[29] S. Savage, D. Wetherall, A. Karlin, T. Anderson, Practical network support for IP traceback, in: ACM SIGCOMN, August 2000, pp. 295–306.
[30] R.L. Graham, D.E. Knuth, O. Patashnik, Concrete Mathematics, Addison-Wesley, 1989.
[31] L. Garber, Denial-of-service attacks rip the Internet, Computer (April) (2000) 12–17.
[32] Z.-J. Lee, Efficient dynamic probabilistic packet marking, Master's thesis, Department of IECS, Feng Chia University, June 2002 (in Chinese).
[33] J. Liu, Z.-J. Lee, Y.-C. Chung, Efficient dynamic probabilistic packet marking, in: Proceedings of the 11th IEEE International Conference on Network, September 2003, pp. 475–480.
[34] T. Peng, C. Leckie, K. Ramamohanarao, Adjusted probabilistic packet marking for IP traceback, in: Proceedings of the 2nd IFIP Networking Conference, May 2002, pp. 697–708 [Online]. Available from: <citeseer.nj.nec.com/557355.html>.
[35] C. Jin, H. Wang, K.G. Shin, Hop-count filtering: an effective defense against spoofed traffic, in: Proceedings of the ACM Conference on Computer and Communications Security, October 2003 [Online]. Available from: <citeseer.nj.nec.com/561060.html>.
[36] P. Ferguson, D. Senie, Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing, May 2000, rFC 2827.
[37] K. Park, H. Lee, On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets, in: Proceedings of the ACM SIGCOMM Conference on ACM, August 2001, pp. 15–26 [Online]. Available from: <citeseer.nj.nec.com/article/park01effectiveness. html>.
[38] J. Li, J. Mirkovic, M. Wang, P. Reiher, L. Zhang, Save: source address validity enforcement protocol, in: Proceedings of IEEE INFOCOM, June 2002 [Online]. Available from: <citeseer.nj.nec.com/li01save.html>.
[39] A.B. Montz, D. Mosberger, S.W. O'Malley, L.L. Peterson, T.A. Proebsting, Scout: A communications-oriented operating system, in: Proceedings of Hot OS, May 1995.
[40] A. Yaar, A. Perring, D. Song, Pi: A path identification mechanism to defend against DDoS attacks, in: Proceedings of the IEEE Symposium on Security and Privacy, 2003.

Jenshiuh Liu received his B.S. and M.S. degrees in Nuclear Engineering from National Tsing Hua University, also M.S. and Ph.D. degrees in Computer Science from Michigan State University in 1979, 1981, 1987 and 1992, respectively. Since 1992, he has been an associate professor in the Department of Information Engineering and Computer Science at Feng Chia University, Taiwan. His research interests include parallel and distributed processing, computer system security, and computer algorithms.

Zhi-Jian Lee received his B.S. and M.S. degrees in Information Engineering and Computer Science from Feng Chia University in 2000 and 2002, respectively. His is now with the 3Probe Technologies in Hsin-chu, Taiwan.

Yeh-Ching Chung received the BS degree in information engineering from Chung Yuan Christian University in 1983, and the M.S. and Ph.D. degrees in computer and information science from Syracuse University in 1988 and 1992, respectively. He joined the Department of Information Engineering at Feng Chia University as an associate professor in 1992 and became a full professor in 1999. From 1998 to 2001, he was the chairman of the department. In 2002, he joined the Department of Computer Science at National Tsing Hua University as a full professor. His research interests include parallel and distributed processing, pervasive computing, embedded software, and system software for SOC design. He is a member of the IEEE Computer Society and ACM.