Mitigating Distributed Denial-of-Service Attack with Deterministic Bit Marking

Yoohwan Kim[1], Ju-Yeon Jo[2], Frank Merat[3], Mei Yang[4], and Yingtao Jiang[4]

[1]School of Computer Science, University of Nevada
Las Vegas, NV, 89052-4019, USA
Email: yoohwan@cs.unlv.edu

[2]Computer Science Department, California State University
Sacramento, CA, 95819-6021, USA
Email: joj@ecs.csus.edu

[3]Electrical Engineering and Computer Science Department,
Case Western Reserve University
Cleveland, OH, 44106-7071, USA
Email: flm@po.cwru.edu

[4]Department of Electrical and Computer Engineering, University of Nevada
Las Vegas, NV, 89052-4019, USA
Email: {meiyang, yingtao}@egr.unlv.edu

Abstract

The Distributed Denial-of-Service attack is a serious threat in Internet and an effective method is needed for distinguishing the attack traffic from the legitimate traffic. We propose the concept of bit marking to identify and drop the attack packets. Bit marking is a variation of packet marking technique that modifies the packet header at each router. However bit marking differs from packet marking in its process and the purpose. Instead of storing the router information in the packets, bit marking alters one or more bits in the marking field at each router. The bit positions for each ingress line card are selected randomly only once at the initialization. Such bit marking is performed to all the packets, resulting in a common path signature in the marking field upon arriving at a destination for all the packets originating from the same location. Since the packets traversing different paths are likely to have different path signatures, the bit marking process generates quite unique path signature for different sources, roughly emulating the source IP. Such Path Signature allows an easy identification and blocking of the DDoS attack.

We show that the PS becomes more diverse as the lengths of the distinct path increases. From an artificial and real Internet topology we observe that the sources are uniformly distributed over the path signature space. In our experiments, the attack traffic can be blocked up to 99.6% using PS. DBM can mitigate most known attack types, such as SYN flooding, reflection attack, UDP flooding, etc. and it is robust to various attack patterns. DBM can also be extended to source address traceback with the topology information of participating routers. This method is simpler to implement than PPM and only small number of routers needs to be upgraded. The deployment can be done gradually without any impact on non-participating routers.

Keyword: Denial-of-Service Attack, Internet, Network Security

I. Introduction

A. *Distributed Denial-of-Service Attack*

In Denial-of-Service (DoS) attack, large amount of packets are bombarded to a destination to disable the victim server or the communication link toward the victim [18]. In Distributed DoS (DDoS) the attack is originated from large number of attack agents. DDoS attack is becoming more threatening with newer type of attacks and attack patterns [31]. It is known that a lot of Internet traffic is caused by dormant DDoS attack agents [29]. As well as wasting the network bandwidth, such dormant attack agents can concentrate the attack packets to any destination any time, materializing into a successful attack.

Under DDoS packet flooding, it is difficult to identify the attacking packets or to identify the source of the attack since the source IP addresses of the attack packets are usually spoofed. Especially in reflector attack, the real source of attack is very difficult to track.

B. *Previous Approaches*

There have been many proposals to identify the attack packets and block them. For the packet flooding attack, rate limiting is commonly used to lower the bandwidth to the victim. In some rate-limiting schemes, the both attack and legitimate packets towards the victim destination are simply dropped [20][23][41][43] Pushback scheme [21] can reduce the attack traffic rate greatly, but it requires a network-side collaboration and may not work under all different attack patterns or types. Other methods use techniques to differentiate the attack packets from legitimate packets. [28] is based on measurement. URPF [13] examines the source address and drops the packets that are coming from unlikely ingress ports. [25] blocks TCP packet flooding.

Another area of research focuses on traceback of the source address, including logging [34], packet marking [33] and new ICMP [6][40]. However, a disadvantage of traceback is that it cannot identify and drop the attack packets. Besides, if the attack is highly distributed, finding all the sources may not be possible or meaningful.

There are other methods for blocking specific attack types. TCP SYN packet flooding attack incapacitates the destination server with only small bandwidth. There are methods specifically designed to block SYN flooding attack, such as SYN cookie in Linux, puzzle [3], host/connection pricing/timing, SYN intercept [12] and etc [11][17][27][38].

Many denial-of-service attacks can be prevented if the source IP address is trustworthy. First of all, the attack can be more easily detected. Once the attack is detected, it is straightforward to limit the rate for the attack traffic by examining the source IP. Ingress filtering [16][24] guarantees the reliability of the source IP. But there are technical and economical barriers, such as mobile IP and extra hardware. There are trials to identify the packet with spoofed address, but they are mostly based on heuristics and may not be applicable to all networks [37].

C. *Our Approach*

In this paper, we propose an innovative and simple method that can significantly mitigate most types of the packet flooding attack. Our method is based on the concept of packet marking, where the packet header is modified as it passes through a router. In probabilistic packet marking (PPM), another similar technique, the router ID information is inserted in the packet to enable source traceback. In contrast, we change one or more bits in the packet header in randomly selected bit positions, which is called bit marking. By performing bit marking at all the routers along the path, we generate virtually unique Path Signature (PS) for all the packets originating from the same location. While the PPM is used for source traceback, bit marking is mainly used for identifying and dropping the DDoS attack packets.

A path signature is uniquely mapped to by less than several source subnets or Autonomous Systems (AS), so it can be treated as semi-source IP address when the source IP address is unreliable. When there is a DDoS attack, only the path signatures containing the attack show a surge

of traffic. Therefore the attack traffic can be easily rate-limited, while the legitimate traffic can be passed without harm.

Bit marking can be performed either deterministically, or probabilistically. In Deterministic Bit Marking (DBM), all the routers along the path flips the selected bits (bitwise exclusive-OR operation) for all incoming packets. In Probabilistic Bit Marking (PBM), the bit marking is performed only in selected routers, where each router makes the decision by probability once at the initialization time. Those selected routers perform the bit marking for all the incoming packets, but differently from DBM, i.e., they overwrite the randomly selected bits with 1 (bitwise OR operation). In our analysis DBM is far superior to PBM in generating a unique path signature, so we focus on DBM in this paper although we present a brief analysis of PBM.

Bit marking has several advantages over packet marking. Generally it is simpler to implement, and can achieve both packet dropping and source traceback. The dropping decision and traceback is done for individual packet, so the response time is short. The number of routers required for full support is much less.

In the next sections, we show how unique the path signature is, how the path signatures are distributed in artificial and real Internet topologies. We then demonstrate the effectiveness of DBM in blocking extremely distributed DoS attack. In addition, we briefly mention the deployment issues, the performance of PBM and the possibility of source traceback using DBM.

II. Deterministic Bit Marking

A. Background in Packet Marking

It is a safe assumption that all the attack packets spoof the source IP address to hide the origin of the attack packets. Currently IP network has no provision for verifying the legitimacy of the source IP, although a few methods have been proposed to solve this problem. The packet marking technique is one of such approaches. In packet marking, each router along the path puts the router information in the packet header, so the receiver can traceback the path of the packet toward the ingress point. One bruteforce approach is attaching all the router IPes for each packet, which is called deterministic packet marking (DPM) since it stores all the path information in every packet. But Deterministic Packet marking is impractical due to the large and variable size.

Probability packet marking (PPM) solves this problem [19][33]. Instead of attaching all the router IPs, it allows only one fixed marking space. Since this small space is not sufficient for storing all the router information, partial path information is written over multiple packets. Each router overwrites its router information in some number of packets selected with a probability. Then the receiver can reconstruct the full path after collecting sufficient number of packets.

Packet marking requires usable bits for marking purpose in IP header and it is suggested to use the16-bit ID field in the IP packet header [33]. ID field is used for handling IP packet fragmentation, but only 0.25% of the Internet packets are fragmented. The performance of PPM depends on the probability of marking and length of the path [30]. As the length of the path increases, the required number of packets for reconstructing the path is also increased. PPM is effective under a single-source DoS attack because a lot of packets are received from one source during the attack and the path for such traffic can be uniquely reconstructed. However as the number of attack sources increases, it becomes more difficult to identify the paths.

The purpose of PPM is tracing back the source of the attack, but it cannot be used for isolating the attack traffic and proactively dropping the attack packets as packet-by-packet because one packet alone does not contain sufficient information. PPM has these additional limitations.

- It is not effective in highly distributed attack and in handling various attack patterns
- For each router, it must compute a new random number for each incoming packet in order to decide whether to mark it or not, which may be an expensive operation.
- If any of the router along the path does not participate, the path may not be reliably reconstructed.

To resolve the aforementioned problems, we propose a mechanism called Deterministic Bit Marking.

B. *Overview of Bit Marking*

The Bit marking, proposed in this paper, is based on the idea of packet marking. Bit marking is similar to packet marking in that each router changes the marking field in the packet header. However, the marking processes and the purposes are different. In Bit marking, the routers do not store any router-specific information, but instead they alter one or more bits in the marking field. This process is done for all the packets, not only for selected packets, so all the packets going through the same routers from the source to destination have the same bit marking pattern. This pattern reflects the unique path for a packet, so it is called *Path Signature* (PS). Since the bit positions are randomly selected, the packets traversing different paths are very likely to have different PS upon arriving at the destination. Due to this uniqueness, PS can be used in isolating the flooding attack traffic even if the source IP is spoofed. When there is a DDoS attack, the traffic size bearing a specific PS that accompanies an attack will jump up, and we can easily apply rate limiting algorithms for such a PS. Therefore the Bit Marking scheme can be used for identifying the attack traffic and dropping those packets, although we later augment it for source tracking. Unlike in PPM, each packet is treated independently in Bit Marking for packet dropping. So there is no need to collect many packets. These concepts depart sufficiently from the concept of packet marking, so we refer this new concept to *Bit Marking* to avoid any confusion.

There are two methods, deterministic bit marking (DBM) and probabilistic bit marking (PBM). In deterministic bit marking, the bits are flipped (= Exclusive OR with 1) and the marking is done for all the packets. On the other hand, in probabilistic bit marking, the bits are flipped in the same way, but the marking decision is made by probability. However, unlike PPM, the marking decision is made once per router, not packet by packet basis. We concentrate on DBM in this paper, but we will briefly introduce the probabilistic bit marking, too.

C. *Deterministic Bit Marking Algorithm*

The bit marking strategies can be different, in terms of number of bits and bit marking method. In this section, we explain the algorithm for Deterministic Bit Marking with b-bits.

Let $\{b_1, b_2, \ldots b_i, \ldots, b_k\}$ be the *marking bits* in a packet header. These bits are initialized to 0 upon entering the ISP's network through the ingress router. From then on, each router selects b bit positions among k bit position ($1 \leq b << k$). The bit positions are randomly chosen at the time of hardware initialization, or they are periodically reselected. In any case, they do not get changed for a prolonged time. (e.g., minutes or hours) A DBM scheme that uses b bit positions for the number of marking bits is called b-bit DBM.

For each incoming packet, the bit values at the selected position are flipped, which means the value 1 is changed to 0, and value 0 is changed to 1. It is equivalent to an exclusive-OR operation with 1. After flipping the bits, the IP checksum must be updated accordingly, which can be done incrementally [9][26]. This marking operation is performed for all the packets at all routers; hence it is called *deterministic* in contrast to *probabilistic*. Figure 1 shows the example of 1-bit DBM.

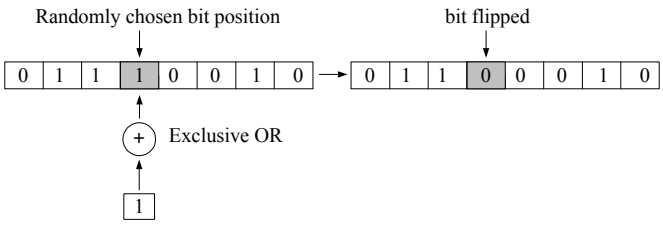


Figure 1: Bit Marking Process (1-bit DBM)

The Path Signature is constructed as follows:

1. Each ingress line card chooses *b* bit positions randomly at the initialization.
2. When a packet enters the ingress line card at ingress router, the line card resets the marking bits to 0
3. All the ingress line card, including the one at the ingress router, perform exclusive OR on b bits with 1 that it has selected in step 1, update checksum and forward the packet to the next router.
4. The last router, after performing its own DBM, finishes constructing PS.

Figure 2 shows how the Path Signature is constructed along the path.

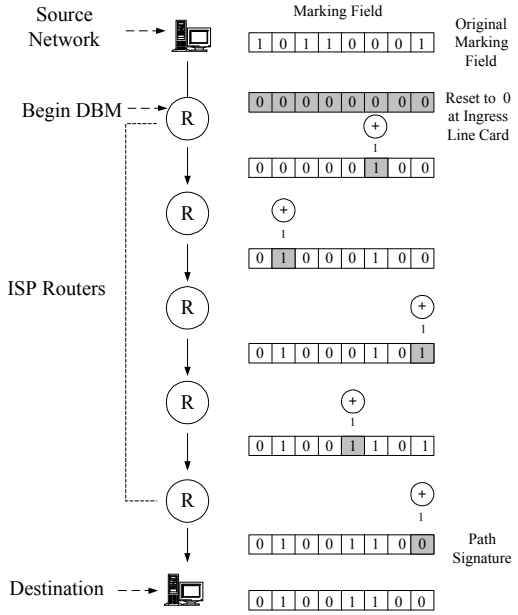


Figure 2: Path Signature Construction

D. *Path Signature Diversity*

The key issue in DBM scheme is whether the packets from different sources to the same destination ended up having different PS. If many different sources are funneled into the same PS, the effectiveness of DBM scheme will be very limited. It is clear that the PS distinction cannot be made when the packets traverse the same path, so the distinction can be introduced only before the paths of packets get merged. However, note that any existing distinction between PSes will be maintained because the existing bits are only flipped rather than being overwritten.

We examine the probability that two packets, *A* and *B*, will have the same PS before their paths get merged. We assume that the 16-bit ID field of IP packet header is used for the marking field throughout this paper. However different number of bits can be used without loss of generality.

We denote their marking bits as $\{a_1, a_2, .. a_i , .. a_{16}\}$ for packet A and $\{b_1, b_2, .., b_i, .. b_{16}\}$ for packet B. Let *d* be the number of routers that the two packets traverse independently before they get merged to the same path. Let's start from the case of 1-bit DBM. Their marking bits are initialized to 0 upon entering the first router. At each router, the probability that a particular bit is selected is 1/16. For a bit value to be 0, the bit must be flipped even number of times because a bit value is initialized at the ingress router. The bit flipping may be done at any router among *d* routers. This is summarized by the following equations. After *d* distance, the probability for a bit to be 0 is

$$P(a_i = 0) = P(b_i = 0) = P_0 = \sum_{k=even} {}_dC_k \left(\frac{1}{16}\right)^k \left(\frac{15}{16}\right)^{d-k} \quad (k < d)$$

Similarly, the probability for a bit to be 1 is

$$P(a_i = 1) = P(b_i = 1) = P_1 = \sum_{k=odd}^{d} {}_dC_k \left(\frac{1}{16}\right)^k \left(\frac{15}{16}\right)^{d-k} \quad (k < d)$$

For a_i and b_i to have the same value, they must be either $a_i = b_i = 0$ or $a_i = b_i = 1$. Since $P(a_i = 0) = P(b_i = 0)$ and $P(a_i = 1) = P(b_i = 1)$,

$$P(a_i = b_i = 0) = P_0 * P_0$$
$$P(a_i = b_i = 1) = P_1 * P_1.$$

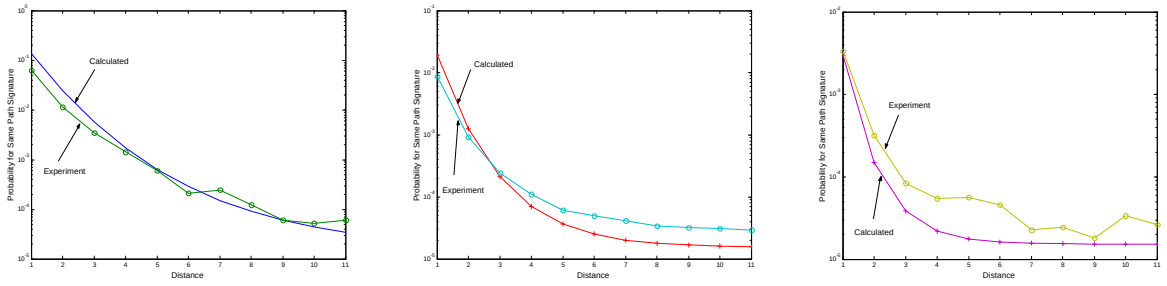
Therefore the probability that two bit values are same after d hops is

$$P(a_i = b_i) = \left(P_0^2 + P_1^2\right)$$

For packet A and B to be same, all the 16 bits must be same, so the probability that both packets are same is

$$\left(P_0^2 + P_1^2\right)^{16}$$

For the case of $d = 0$ and 1, clearly the above probabilities are 0 and 1/16 respectively. The above argument can be generalized for k-bit marking, by replacing the bit selection probability to k/16 from 1/16. We have plotted the probability in the following figures. The graph also shows the experimental results of 5-experiments average from applying DBM to 10 million pairs for each distance.



(a) 1-bit DBM **(b) 2-bit DBM** **(c) 3-bit DBM**

Figure 3: The probability that two PS from different sources are same after *d* routers

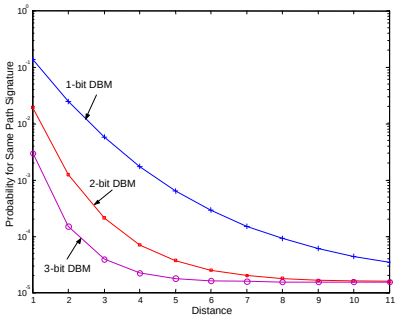


Figure 4: Comparison of 1,2 and 3-bit DBM with calculated values

The length of the distinct portion between two packets' paths is most important in diversifying the PS. As shown in Figure (a), the diversity increases as d increases. By using multiple bits, the diversity can increase much quicker as shown in Figure 3 (b) and (c). But after about 10 hops, the probability converges in all cases. Therefore multiple-bit DBM is useful when the length of distinct path is short. The relative effectiveness of multiple-bit DBM schemes are compared in Figure 4.

III. Path Signature Distribution

Now we would like to find out whether there is sufficient PS diversity in a network where many paths get merged. We performed two experiments, one with an artificial topology and the other with a real Internet topology.

A. Artificial Topology

We have created an artificial topology based on some Internet statistics. There are currently about 20,000 ASs and about 140,000 subnets registered worldwide [1].

We have constructed a tree structure to reflect the statistics as shown in Figure 5. Note that although the real Internet connectivity is in graph form, it becomes a tree form from a particular destination's point of view. The destination subnet is at the root of the tree and the packets are delivered to the root from the leaves. In this topology, each router is assumed to have 3 ports that are connected to two upstream routers and one local subnet except at level 16. The furthest upstream routers at level 16 are connected to one subnet. The number of hops from the destination router is called distance *d*. At *d* distance, there are 2^d routers and subnets, and 2^d subnets. The maximum number of hops is 16, which gives us total $2^{17}-1$ (=131,071) subnets.

This model encompasses the case of many merged paths in all different levels. It also closely reflects the actual number of connected subnets to the Internet, and the realistic hop counts of 1 to 16.

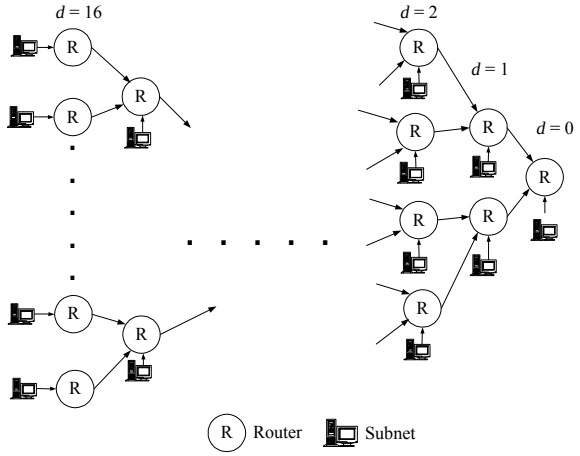


Figure 5: Artificial Topology

As a packet enters a router, its ID field in the IP packet header is reset to 0 and DBM process begins. We run the experiment 5 times, where the bit positions in each line cards are initialized differently. TABLE I shows the summary of how the subnets are distributed among the possible 2^{16} (=65,536) Path Signatures.

TABLE I: PS DISTRIBUTION

	Case 1	Case 2	Case 3	Case 4	Case 5
Total occupied PS	48826	47743	48654	49080	47625
Maximum subnets per PS	19	25	19	19	27
Average subnets per PS	2.68	2.75	2.69	2.67	2.75
Median subnets per PS	2	2	2	2	2

The results show that about 75% of PS values are utilized and less than 3 client subnets are assigned to each PS on the average. We observe that even though many paths are merged, the PS values are distributed very evenly. Moreover, a server is rarely connected by all the subnets at the same period, so the actual number of subnets per PS will be lower in reality.

B. The Real Internet Topology

The artificial topology shows an excellent diversity of path signature with realistic hop counts and number of subnets. But the actual Internet topology is not as uniform as the artificial topology and exhibits the power-law topology [10][14]. In the power law Internet, the paths from Internet hosts tends to get merged into a limited number of points. This reduces the number of hop counts and has an adverse effect on the PS diversity. To investigate the PS distribution under real Internet topology we analyzed the actual Internet AS connectivity data [2]. Although it is not complete, it has the basic topology information on the Internet The ASs are either transient AS or Stub AS. Stub ASs do not forward to packets, but only serves the connected clients. Transient AS collects the packets from other AS and forwards them to other ASs through peering points. A basic relationship between ASs is shown in Figure 6.

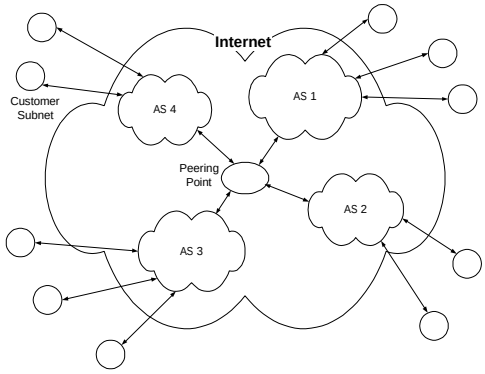


Figure 6: Illustration of AS Connectivity

There are total 11,174 ASs in the data set and its connectivity status is shown in TABLE II. It clearly shows that a small number of ASs are connected to large number of ASs, i.e., the power-law Internet connectivity. The 3 most-connected ASs are, ALTERNET-AS (2389), SprintLink (1334), and ATT-INTERNET4 (1042).

We first constructed AS connectivity graph, then for each stub AS (which has less than 6 connections) we have derived a BFS tree. This guarantees that all the AS can reach a particular destination AS via the shortest path. This is a rather conservative assumption because in real network the path may be longer than the shortest path and thus the PS diversity may be stronger.

TABLE III sows the number of ASes required to reach a destination AS. We have randomly chosen 5 ASes that are connected to 1,2,3,4, and 5 other ASes respectively.

It shows that the destination AS can be reached via only 2 to 5 other ASes in most cases. We need to investigate whether this is enough for achieving usable PS diversity.

TABLE IV shows how the ASes are distributed among the PS. In case of 3-bit DBM, less than two ASs are co-assigned to the same PS. This is a very encouraging discovery because we can use the PS in place of source IP when the source IP is not trustworthy.

We have further explored whether it is possible to distinguish the attack packets with smaller unit than an AS, i.e. subnet level. When the packets from different subnets enter the same ingress router, they go through different DBM process via different line cards. We have assumed that each AS is connected by 10 subnets, simulating the case of 111,740 networks that is close to the actual number of networks connected to the Internet (about 130,000).

TABLE II: AS CONNECTVTY

Number of Connected AS	Number of AS	%	Cumulative %
1	3866	34.60%	34.60%
2	4484	40.13%	74.73%
3	1207	10.80%	85.53%
4	466	4.17%	89.70%

5	240	2.15%	91.85%
6	177	1.58%	93.43%
7	94	0.84%	94.27%
8	87	0.78%	95.05%
9	66	0.59%	95.64%
10	40	0.36%	96.00%
11-20	246	2.20%	98.20%
21-50	123	1.10%	99.28%
51-100	43	0.38%	99.69%
100-1000	32	0.29%	99.97%
1000-2000	2	0.02%	99.99%
2000-3000	1	0.01%	100.00%
Total	11174	100%	100%

TABLE III: NUMBER OF HOPS TO REACH AN AS

AS ID / # hops	Test 1	Test 2	Test 3	Test 4	Test 5
	7803	851	18701	9940	13374
1	1	2	3	4	5
2	31	49	3376	3206	2708
3	1479	3417	5159	5336	5702
4	6858	5162	2279	2255	2351
5	2408	2189	325	339	368
6	374	321	30	32	38
7	21	32	1	1	1
8	1	1	0	0	0

TABLE IV: PATH SIGNATURE DISTRIBUTION
(When one subnet is connected to each AS)

		Test 1	Test 2	Test 3	Test 4	Test 5
No. assigned PS	1-bit	861	1041	1033	1008	1566
	2-bit	2436	2611	2683	2586	3897
	3-bit	6154	6723	6692	6483	7871
Average no. AS per PS	1-bit	12.98	10.73	10.82	11.08	7.13
	2-bit	4.59	4.28	4.16	4.32	2.87
	3-bit	1.82	1.66	1.67	1.72	1.42
Maximum no. AS per PS	1-bit	429	293	310	301	204
	2-bit	219	168	190	211	98
	3-bit	28	23	26	28	13
Median no. AS per PS	1-bit	2	2	2	2	3
	2-bit	3	3	2	2	1
	3-bit	1	1	1	1	1

TABLE V: PATH SIGNATURE DISTRIBUTION
(When 10 subnets are connected to each AS)

		Test 1	Test 2	Test 3	Test 4	Test 5
No. assigned PS	1-bit	1532	1881	1747	1790	2935
	2-bit	6352	7508	7536	7211	10735
	3-bit	23126	25476	25618	24971	34017
Average no. subnets per PS	1-bit	72.93	59.40	63.96	62.42	38.07
	2-bit	17.59	14.88	14.83	15.49	10.41
	3-bit	4.83	4.39	4.36	4.47	3.28
Maximum no. subnets per PS	1-bit	4345	3042	2951	3150	2194
	2-bit	664	710	634	628	474
	3-bit	106	73	87	80	53
Median no. subnets per PS	1-bit	8	9	11	9	5
	2-bit	2	2	2	2	3
	3-bit	3	3	3	3	2

TABLE V shows that on the average about 5 subnets share the same path signatures in case of 3-bit DBM. The similar result is obtained from the ASs with high connectivity. In case of SprintLink, which has 1,334 ASs connected, the total number of PS is 24,306, the maximum number of subnets per PS is 76, and average number of subnets per PS is 4.6.

In real Internet, 1-bit DBM is probably sufficient because the packets actually go through multiple hops within an AS. That is, 3-bit DBM simulates the situation where there are 3 router hops within an AS, each router with 1-bit DBM. This is a realistic assumption as the average number of hops in TABLE III will become 3 to 24, which is very close to the number of hops in the Internet. Therefore with 1-bit DBM, we can differentiate the attack traffic with the same performance of 3-bit DBM in TABLE V in real Internet. However we will need precise subnet connectivity information for deriving more accurate PS diversity data.

IV. Defense against a DDoS Attack with Path Signature

A. *Attack Detection*

Detecting whether there is an attack is the first step in coping with DDoS attack. During DDoS attack, the traffic amount for a specific type of traffic toward a victim server or subnet is greatly increased, such as total amount of traffic, number of SYN packets, number of ICMP packets, etc. Such a sudden increase in traffic amount is often considered an attack in many DDoS defense schemes.

However the increased traffic could be due to legitimate traffic. The sudden increase of legitimate traffic is called *flash crowd*. There are studies for distinguishing an attack from flash crowd by observing the subnet prefix distribution for source IP addresses [5][22][32]. However, this method has two challenges. First it is necessary to pre-establish the normal traffic pattern, which is cumbersome and may not be stable for some sites. Further if the attacker generates similar subnet prefix distribution, rather than random distribution, the detection mechanism fails. Second, even if the attack is identified, it does not let us decide which packet is attack packet.

Attack detection with PS offers some advantages. For a particular destination, its clients are usually well distributed over different ASes or networks. The incoming traffic amount (total traffic amount, number of SYN packets, number of ICMP packets, etc) per AS or a network should be within a certain range under normal conditions. But during the DDoS attack, the incoming traffic for the PS that contains the attack traffic jumps up. Further, the number of source IPs per PS will increase if the source is spoofed. Even if the attacker mimics the source IP distribution pattern for the destination, it works only for the whole packets, not for the packets per PS. So DDoS attack can be better detected by monitoring per-PS traffic characteristics. If the surge of traffic is determined indeed due to an attack, we can use rate-limiting algorithms for the particular PS as described in the next section

B. *Rate Limiting for Attack Traffic*

To reduce the attack rate, we use per-PS rate control algorithm similar to per-flow rate control (Fair Queuing [15][35][36]). We show a simple rate control algorithm for bandwidth flooding attack. But any rate-limiting algorithm can be used independent of DBM. Let's denote that

- Number of observed PS = P
- The maximum link bandwidth = B
- Number of PS containing attack = P_a
- The bandwidth from attack PS = B_a
- The bandwidth from the legitimate PS = B_l
- $B << B_a$ under attack

We limit the bandwidth per PS as

- The bandwidth per PS = B/*P*

Then the allowed bandwidth for the attacking PS is

$$P_a * B/P = B * P_a/P$$

while the allowed bandwidth for the legitimate PS is

$$(P-P_a) * B/P = B * (1 - P_a/P)$$

In other words, no matter how big the attack bandwidth B_a is, only $(P_a * B/P)$ can pass the router. That is, as the attack intensifies, the blocking rate becomes higher as long as the number of attack PS stays same. This is substantially better than the case without PS. Without PS, the allowed bandwidth for the attack traffic is

$$B * B_a/(B_a + B_l)$$

and the allowed bandwidth for legitimate traffic is

$$B * B_l/(B_a + B_l)$$

That is, as the attack traffic increases, the less legitimate traffic can pass.

Both detection and dropping can be done using a token bucket array. For each flooding attack type and destination, we allocate a token bucket array with one bucket per PS. The packets are drained at a reasonable rate, e.g., *B/P*. If an overflow over a threshold in a bucket is monitored, the packets for the PS are analyzed for attack. If it is due to an attack, the packets causing overflow will be dropped.

One problem with per-PS rate control is that the legitimate packets coming from the same subnet as the attacker's or from the subnets whose PS is same as the attacker's suffer from the packet dropping. This is called collateral damage. The packets coming from the same subnet as the attacker are always sacrificed. However, we believe that such a case must be handled by the subnet itself, as it is responsible for the attack. In case of accidental co-assignment of the same PS, we can reduce the damage by periodically reassigning the marking bit positions at routers. The resulting PS for the subnet that suffered coincidental sharing of PS then has a new PS. That way, the collateral damage may be shared among many subnets in rotation.

C. *Attack Detection and Packet Dropping Location*

The attack detection can be best done at a closer location to the victim, such as egress router to the victim, where the PS diversity is maximally achieved. But once the attack is detected, the filtering can be done either locally at the detection site or at upstream routers. In case of local dropping, the attack detection is done in egress port, and the attacking PS is given to the ingress ports. Each egress port performs reverse DBM for the PS and drops the packets for the PS. This procedure is described in Figure 7.

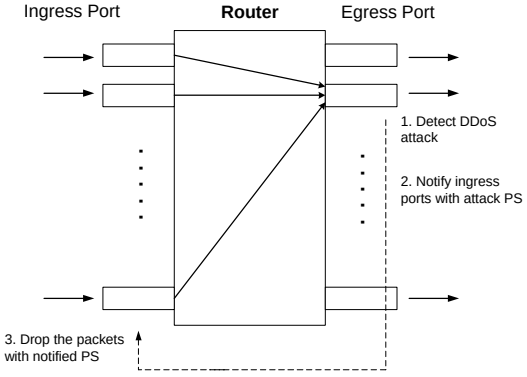


Figure 7: Router Ingress Port and Bit Marking

For remote dropping, the detected PS is given to the upstream routers. They perform reverse bit marking along the upstream path to get the correct PS, and drop the packets with the PS

V. Effectiveness of DBM in Blocking Attack Traffic

A. *Attack Conditions*

To find out the attack detection and blocking effectiveness, we perform the following experiment. From the actual AS topology, we take the same 5 stub ASs which were taken in section **Error! Reference source not found.**. For each stub AS, we randomly choose 1,000 client ASes. To simulate a heavily distributed attack, we again choose 10 and 100 attack ASes randomly out of the 1,000 ASs. This is considered an extremely distributed attack because the chance of compromising so many hosts at the same time is very slim. We use 3-bit DBM and assume these conditions.

- The destination subnet can receive 200 Mbps of traffic
- Each AS sends 0.1 Mbps of traffic to the victim's AS, resulting in 100 Mbps for legitimate traffic
- Attacker ASs send 500 Mbps combined traffic evenly distributed among them, i.e., 5 Mbps with 100 attack ASs and 50 Mbps with 10 attack ASs.
- The victim's AS can receive up to 1 Gbps of incoming traffic, so the traffic increase can be handled at AS level.

The total incoming traffic during attack is 600 Mbps during the attack but only 200 Mbps can be delivered. Since the packets are dropped equally for attack packets and legitimate packets without distinction, only 33 % of legitimate traffic can be delivered. We examine how DBM can help to mitigate this situation.

B. *Attack Traffic Identification and Blocking*

TABLE VI and TABLE VII show the results of the experiment. In the experiment of attack from 10 ASes, summarized in TABLE VI, we can see that PS can effectively identify the attack traffic with small collateral damage, while passing the majority of legitimate traffic (99%). More than 99.5% of the attack traffic is dropped.

TABLE VI: RESULTS FOR 10 AS ATTACK (3-BIT DBM)

	Test 1	Test 2	Test 3	Test 4	Test 5
Number of Assigned PS	883	923	922	904	954
Number of PS containing attack traffic (Attacking PS)	10	10	10	10	10
Number of AS sharing the PS with attack traffic	15	10	14	13	10
Total traffic from Attacking PS	501.50	501.00	501.40	501.30	501.00
Ratio of attack traffic bandwidth in attacking PS	99.70%	99.80%	99.72%	99.74%	99.80%
Allowed bandwidth per PS	0.23	0.22	0.22	0.22	0.21
Allowed bandwidth for attacking PS	2.27	2.17	2.17	2.21	2.10
Total attack traffic passed	2.26	2.16	2.16	2.21	2.09
Ratio of attack traffic passed	**0.45%**	**0.43%**	**0.43%**	**0.44%**	**0.42%**
Total legitimate traffic passed	98.51	99.00	98.61	98.71	99.00
Ratio of legitimate traffic passed	**98.51%**	**99.00%**	**98.61%**	**98.71%**	**99.00%**
Total combined traffic passed	100.77	101.17	100.77	100.91	101.10

TABLE VII: RESULTS FOR 100 AS ATTACK (3-BIT DBM)

	Test 1	Test 2	Test 3	Test 4	Test 5
Number of Assigned PS	883	923	922	904	954
Number of PS containing attack traffic (Attacking PS)	96	100	97	98	100
Number of AS sharing the PS with attack traffic	137	120	119	117	112
Total traffic from Attacking PS	513.70	512.00	511.90	511.70	511.20

Ratio of attack traffic bandwidth in attacking PS	97.33%	97.66%	97.68%	97.71%	97.81%
Allowed bandwidth per PS	0.23	0.22	0.22	0.22	0.21
Allowed bandwidth for attacking PS	21.74	21.67	21.04	21.68	20.96
Total attack traffic passed	21.16	21.16	20.55	21.19	20.51
Ratio of attack traffic passed	**4.23%**	**4.23%**	**4.11%**	**4.24%**	**4.10%**
Total legitimate traffic passed	86.88	88.51	88.59	88.80	89.26
Ratio of legitimate traffic passed	**86.88%**	**88.51%**	**88.59%**	**88.80%**	**89.26%**
Total combined traffic passed	108.04	109.67	109.14	109.98	109.76

The experiment of attack from 100 ASes, summarized in TABLE VII, shows the effectiveness of DBM under extremely distributed attack. In all 5 test cases, the simple rate-limiting algorithm can pass about 88% of legitimate traffic while blocking 96% of the attack traffic. The total bandwidth is only slightly increased. Although the amount of collateral damage is increased inevitably due to the wide distribution of attack sources, still the denial-of-service situation can be prevented for the 88% of legitimate traffic.

VI. Deployment

A. *Implementation at Routers*

In most line cards, only simple bit flipping operation needs to be added by firmware upgrade. The checksum update is already a standard function in line cards. No extra storage or other hardware is necessary. Unlike PPM, no random number generation is needed for each packet, which is an expensive operation.

The egress routers that need to protect the customer subnets may need to implement the per-PS rate control algorithm. Since many routers already have per-flow rate control, small modification of firmware will suffice.

B. *Deployment Location*

For achieving good PS diversity, only several router hops are necessary. Given the Internet topology, deployment only at AS level routers is necessary, such as border routers running BGP. The routers within a subnet or internal routers within the ISP need not be upgraded, although it is recommended for protection from intra-AS attacks. This reduces the number of routers requiring the upgrade significantly. Given the number of AS, the number of routers need to be upgraded is only in the order of 10,000 for the entire Internet.

C. *Gradual Deployment*

Not all the routers can be upgraded in one day. Most likely the routers will be upgraded gradually over several years. However, the gradual upgrade does not impact the Internet operation. For the routers that do not support DBM, the marking bits are simply ignored and there is no impact to them.

Until every ISP employs DBM, it is necessary to protect those DBM-enabled networks from an external attack. This can be achieved by extending marking field to support a marking indicator. In addition to the ID field in the IP header, we further assume that the IP fragment offset field (13 bits) is available because this field is unused if IP fragmentation is disabled. This results in 29 bits usable for bit marking. We divide the 29 bits into two groups; 21 bits for the actual bit marking, and 8 bits for checksum for marking indicator as shown in Figure 8. The checksum value is obtained by applying CRC-32 hashing to the sum of path signature and TTL (Time to Live) values in the IP packet header.

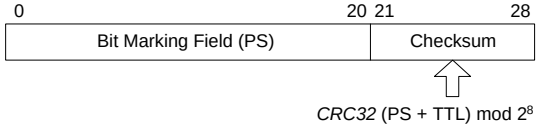


Figure 8: Checksum as a Marking Indicator

When a packet arrives from an end-user network, the edge routers initialize the marking bits to 0, perform DBM and write a proper checksum. If the next hop router supports DBM, it also writes a recalculated checksum. Therefore the receiving router can determine whether a packet is marked or not. Even if an attacker initializes the checksum to a correct value, it will be invalidated as the packet travels non-DBM routers that decrease only the TTL values. This relation is described in Figure 9. The existing PS value is preserved through the DBM-participating networks, which is called trusted domain.

In the core network, if a packet coming from another ISP is unmarked, the receiving router initializes the marking bits to 1's and starts the bit marking. This helps source traceback to distinguish between the packets originating from end-user networks and from the core networks. If the reverse bit marking yields all 1's, the packet is originated from a non-DBM-enabled network.

The set of DBM-enabled routers that are mutually reachable via other DBM-enabled routers *implicitly* form a trusted domain. Initially there will be islands of trusted domains. Legitimate packets from untrusted domains will be also treated as potential attack traffic. If a DDoS attack is launched from an untrusted domain, all the traffic bearing the same PS, including the legitimate traffic, is subject to rate limiting, in order to protect the DBM-enabled network. The packets originating from a trusted domain but traveling through untrusted domains will be treated as unmarked packets. But as more ISP's support DBM, the trusted domains will become merged.

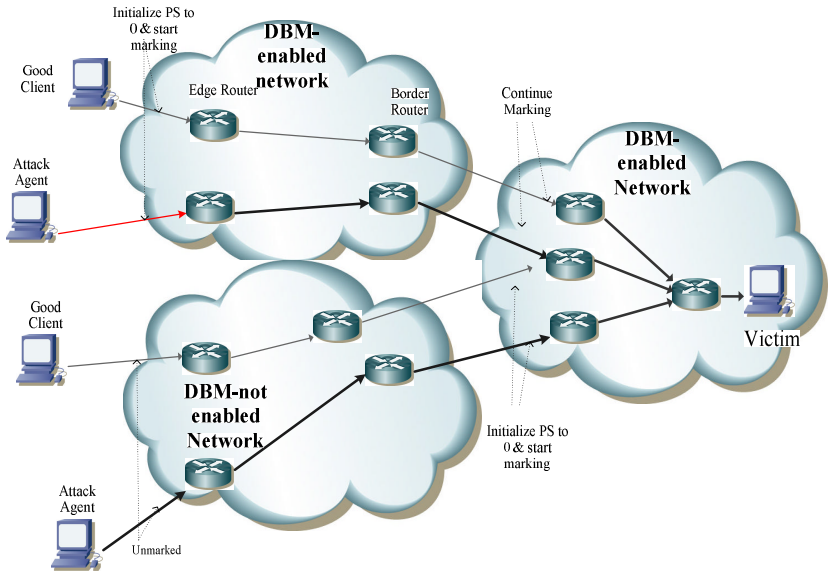


Figure 9 Protecting Trusted Domain

VII. Probabilistic Bit Marking

We have also explored the concept of *Probabilistic Bit Marking (PBM)*, where the selected bit is overwritten instead of being flipped. The bit selection process is same as in DBM, but the decision whether to perform bit marking or not is made with the probability *p*. However, unlike in PPM, that decision is made per line card, not per packet, and only once at the initialization time. Thus, all the packets passing through a router undergoes the same sequence of bit marking. Such process generates a unique path signature as in DBM. The PBM algorithm is as follows:

1. At the initialization, each ingress line card at all routers decides with probability *p* whether to perform bit marking or not. If it decides to do so, it chooses *b* bit positions randomly.
2. Regardless of the decision, the ingress line card at ingress router resets the marking bits to 0 when a packet enters.

3. At each router including the ingress router, if it was decided to do bit marking, the ingress line card performs OR operation with 1 on the *b* bits that it has selected in step 1, and update checksum. If not, it simply forwards the packet to the next router.

The last router finishes constructing the path signature.

A. PS value Bias in PBM

PBM tends to have a bias toward 1 or 0 depending on the distance and choice of *p*. At small *d* and low *p*, the marking bit values tends to be 0. The marking field value converges to 1 as *d* increases because the bits are overwritten with 1. So the choice of *p* and the estimation of the distance are important. The number of bits marked after *d* routers is calculated as follows. The probability of a bit to remain as 0 after one hop is

$$P_0(0) = [(1-1/m) + (1/m) * (1-p)] = 1 - p/m \qquad (m\text{: number of marking bits})$$

After *d* hops, it must be still 0,

$$P_0(d) = [1-p/m]^d$$

Therefore the number of marked bits is

$$m * P_1(d) = m * (1 - P_0(d))$$

This result is shown in Figure 10 for $m = 16$ and $p = 0.2, 0.4, 0.6, 0.8$ and 1.0. After 30 hops, almost 14 bits are marked as 1 when $p = 1.0$.

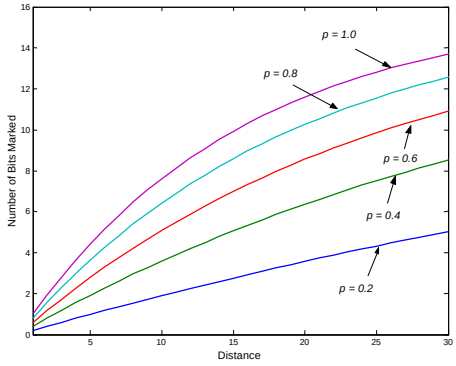


Figure 10: Number of Bits Marked

B. PS Diversity in PBM

The probability of sharing the same PS for two packets after *d* hops can be calculated as following.

$P_u = P$ (a bit remains unmarked after d hops) $= (15/16)^d$
$P_m = P$ (a bit is marked after d hops) $= 1 - (15/16)^d$

For one particular position, it must be either marked together or unmarked together. The probability is,

$$(P_u * P_u + P_m + P_m)$$

This must happen for all 16 bits.

$$(P_u^2 + P_m^2)^{16}$$

Figure 11 shows the PS diversity as the distance increases when $p = 1$ for both the calculated value and the actual experiment. Since the bits are overwritten, the PS converges to 1, so the diversity decreases. In addition, once the two paths are merged, the existing differences decrease further because the bits are overwritten.

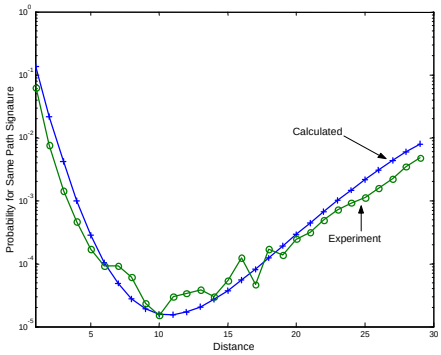


Figure 11: Comparison of Calculated Value and Experiment (p = 1)

By lowering the marking probability, the high diversity can be maintained to farther distances. Figure 12 shows the PS diversity with varying probabilities.

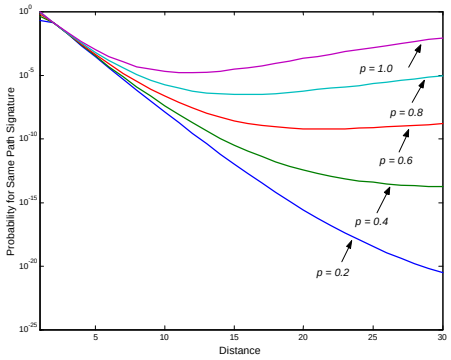


Figure12: PS Diversity with Various p Values

C. *PS Distribution under Artificial Topology*

We have tested the PS diversity under the same artificial network topology as in section III.A. In addition to number of marking bits and distance, the diversity performance of PBM depends on p. TABLE VIII shows the result when p = 1 with 1-bit PBM.

TABLE VIII: PS DIVERSITY WITH 1-BIT PBM (P = 1.0)

	Case 1	Case 2	Case 3	Case 4	Case 5
Total occupied PS	19489	15241	18203	20488	15747
Maximum subnets per PS	87	76	68	80	63
Average subnets per PS	6.73	8.60	7.20	6.40	8.32
Median subnets per PS	4	6	5	4	6

Compared with TABLE I showing the result of DBM under the same condition, DBM has much higher diversity with the number of occupied PS around 48,000, which is more than twice of PBM's.

In addition to lower PS diversity, PBM has another drawback. Source trace back, which is discussed in next section, is not possible with PBM because we don't know whether a bit is overwritten or not in a particular router.

VIII. Source Traceback

With DBM, it is possible to identify the origin of the attack with the resolution of AS or subnet, which is same as in Probabilistic Packet Marking. Although we can't identify the actual host, the AS or ISP is at the best position to directly control the attack source machines, so AS-level identification is sufficient for a practical purpose.

Source traceback is done by constructing a tree of participating routers with the destination router as the root, then applying the bit marking process in reverse. If the marking field becomes 0 after applying the bit marking in reverse, it is possibly the source of the attack. An example is shown in Figure 13.

To achieve source traceback, the topology of participating routers and their bit marking status must be known to the routers that want to investigate the source of a packet. By frequently exchanging the DBM router topology, a router can locally calculate the source of a packet. To exchange the bit marking status information, a protocol will have to be developed.

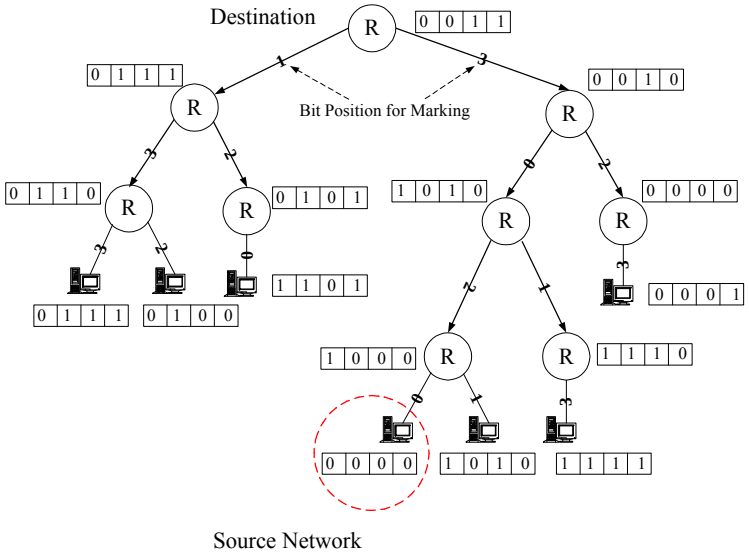


Figure 13: Reverse Bit Marking for Source Traceback

There are a few issues to be resolved. Although very low, the co-assignment of a same PS prevents us from pinpointing the exact source. (In our simulation, the co-assignment rate is less than 0.1% of the subnets.). Further the marking bits may become 0 in some intermediate router prematurely. Another issue is the combinatorial explosion problem because the routers have multiple line cards and the reverse bit marking must be performed for the all line card combinations. These issues require further study.

Nevertheless, this method has a few advantages over PPM. PPM can't resolve the source very well under distributed attack. But DBM-based traceback is not affected by the distributed attack. PPM also requires large number of packets to reconstruct the attack path, but DBM-based traceback can decide the source for each individual packet. However, DBM has a disadvantage over PPM because PPM requires no additional topology information since the incoming packets contain the router information.

It is known that probabilistic packet marking (PPM) is vulnerable to the GOSSIB attack [38] that injects false route information in the marking field. DBM is not affected by that because DBM initializes the marking field at the ingress routers.

In comparison with hash-based traceback [34], DBM offers simpler solution because no packet storing is needed in each router. Hash-based traceback requires similar topology information as in DBM and it requires successive queries among the participating routers. However hash-based traceback has an advantage over DBM that it can resolve to a unique source where as DBM resolves to multiple candidates.

IX. Discussions

A. *Defense Capability against Different Attacks*

DBM can block most flooding-type attacks, including SYN flooding, ICMP flooding or reflector attack. No matter how evenly the attack is distributed, the number of attack sources is typically smaller than the number of legitimate clients. The SYN flooding or ICMP flooding can be filtered similarly as general bandwidth consumption attack with DBM.

Attacks are also launched in variety of patterns. The attack may burst for a certain period of time and stop, or the attack may be originated from different points over time. DBM is quick in responding for such a changing attack. As soon as the attack pattern changes, DBM can detect and block the surge of the attack traffic on new PSes. In comparison, such changing attacks are difficult to block with other methods because they need to monitor the traffic pattern for prolonged time. By the time they respond, the attack traffic may be gone or the pattern may be changed. For example, in PPM if the attack pattern changes before not enough packets are collected, it can't reliably resolve the attack path.

B. *Attack within an AS or a subnet*

Our scheme requires only AS-level routers to perform DBM operation. So it will be difficult to block the traffic within an AS. However an AS that wants a complete protection may install DBM function in its internal routers.

C. *ID field Modification*

A critical drawback of DBM and PPM is that they require modification of the ID field in the IP header. Although the idea of using the ID field is gradually being accepted, the receiving host may interpret packets with same ID as fragmented packets. To prevent that, a DBM-enabled egress router may replace the PS values with incremented values for fragmented packets so that they can be reassembled at the receiving host. Of course this scheme won't work if the packets arrive out of order, but it can significantly reduce the chance of losing fragmented packets.

D. *Related Work*

A benefit similar to PS can be obtained by assigning a unique bit pattern only at the ingress routers without marking the bits at the intermediate routers. [6][7] However without updating the checksum field at the intermediate routers, it is difficult to verify whether a packet is properly marked. By performing bit marking at intermediate routers, the marking field remains reasonably free from attackers' manipulation. Yaar et. al [42] implements a similar scheme using the concept of Path Identifier (PI). A summary of DBM-like schemes are summarized in [3].

X. Conclusions

Denial-of-Service is a serious threat in Internet. The Deterministic Bit Marking (DBM) scheme presented in this paper offers a simple and comprehensive protection against the attack under a widely distributed attack. DBM changes one or more bits in the packet header at each router along the path of a packet creating a virtually unique path signature (PS) for each source network. Using the PS's in place of source IP addresses, it is possible to detect and drop the attack packets with high accuracy. DBM can handle variety of DDoS attack types, such as UDP flooding and SYN attack, changing attack patterns, and massively distributed attacks.

DBM can be easily implemented in routers and gradually deployed. It can defend DBM-enabled networks from the attacks from non-DBM networks. Upgrading the border routers at AS can achieve excellent path signature diversity. This is advantageous because upgrading all the routers in Internet is impractical. With some extra information, DBM can be also used for source traceback.

Our simulation results show that the attack traffic may be isolated within less than 5 subnets in current Internet topology. The simulation results also show that over 99% of the DDoS attack traffic can be dropped.

References

[1] American Registry for Internet Numbers IP network dump, Feb. 2003. ftp://rs.arin.net/netinfo.

[2] AS Graph Data Sets, "The Origin of Power-Laws in Internet Topologies Revisited," http://topology.eecs.umich.edu/data.html

[3] B. Al-Duwairi and Tom E. Daniels, "Topology Based Packet Marking", In *Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN 2004),* 2004

[4] T. Aura, P. Nikander, and J. Leiwo, "DOS-resistant Authentication with Client Puzzles," Lecture notes in Computer Science Series, April 2000, Springer.

[5] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2002.

[6] A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," *IEEE Communications Letters*, vol. 7, no. 4, pp. 162-164, April 2003.

[7] A. Belenky and N. Ansari, "Accommodating Fragmentation in Deterministic Packet Marking for IP Traceback," In *Proceedings of IEEE Global Telecommunications Conference*, vol. 22, no. 1, Dec 2003, pp. 1374-1378.

[8] S. Bellovin, M. Leech, and T. Taylor, "ICMP Traceback Messages," draft-ietf-itrace-01.txt, *Internet draft, work in progress*, October 2001.

[9] R. Braden, D. Borman, and C. Partridge, "Computing the Internet Checksum," RFC 1071, September 1988.

[10] Q. Chen, H. Chang, R. Govindan, S. Jamin, S.J. Shenker, W. Willinger, "The Origin of Power Laws in Internet Topologies Revisited," In *Proceedings of IEEE Infocom,* June 2002.

[11] Y. W. Chen, "Study on the Prevention of SYN Flooding by Using Traffic Policing," In *Proceedings of Network Operations and Management Symposium,* 2000.

[12] Cisco IOS Security Configuration Guide, "Configuring TCP Intercept (Preventing Denial-of-Service Attacks)," pp. SC-213 ~ SC-218, http://www.cisco.com.

[13] Cisco IOS Security Configuration Guide, "Configuring Unicast Reverse Path Forwarding," pp. SC-429 ~ SC-446, http://www.cisco.com.

[14] C. Faloutsos, P. Faloutsos, and M. Faloutsos, "On Power-Law Relationships of the Internet Topology," In *Proceedings of the ACM SIGCOM*M, Sep. 1999.

[15] W.-C. Feng, D.D. Kandlur, D. Saha, and K.G. Shin, "Stochastic Fair Blue: A Queue Management Algorithm for Enforcing Fairness," In *Proceedings of IEEE Infocom*, April 2001.

[16] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing," RFC 2827, 2000.

[17] C.-K. Fung and M.C. Lee, "A Denial-of-Service Resistant Public-key Authentication and Key Establishment Protocol," In *Proceedings of IEEE International Performance, Computing, and Communications*, 2002.

[18] L. Garber, "Denial-of-Service Attacks Rip the Internet", *IEEE Computer*, April, 2000, pp. 12-17

[19] M. T. Goodrich, "Efficient Packet Marking for Large-Scale IP Traceback," In Proceedings of ACM CCS, Nov. 2002.

[20] Y. Huang and J.M. Pullen, "Countering Denial-of-Service Attacks Using Congestion Triggered Packet Sampling and Filtering," In *Proceedings of Computer Communications and Networks,* 2001.

[21] J. Ioannidis and S.M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks," In *Proceedings of Network and Distributed System Security Symposium,* Feb. 2002.

[22] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," In *Proceedings of the International World Wide Web Conference*, May 2002.

[23] D. Kashiwa, E.Y. Chen and H. Fuji, "Active Shaping: A Countermeasure against DDoS Attacks," In *Proceedings of European Conference on Universal Multiservice Networks,* 2002.

[24] G. Kim, T. Bogovic, and D. Chee, "Active Edge-Tagging (ACT): An Intruder Identification & Isolation Scheme in Active Networks," In *proceedings of 6th IEEE Symposium on Computers and Communications,* 2001.

[25] Y. Kim, J.-Y. Jo, H. J. Chao, and F.L. Merat, "High-speed router filter for blocking TCP flooding under Distributed Denial-of-Service attack", International performance, Computing and communications Conference, phoenix, Arizona, April 2003.

[26] T. Mallory and A. Kullberg, "Incremental Updating of the Internet Checksum," RFC 1141, January 1990.

[27] D. Mankins, R. Krishnan, C. Boyd, J. Zao, M. Frentz, "Mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing", In *proceedings of 17th Annual conference on Computer Security Applications,* 2001.

[28] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the Source," In *Proceedings of 10th IEEE International Conference on Network Protocols*, November 2002.

[29] D. Moore, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," In *Proceedings of 10th USENIX Security Symposium,* Aug. 2001.

[30] 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*, April 2001.

[31] V. Paxson, "An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks," *Computer Communication Review* vol. 31 no. 3, July 2001.

[32] M. Poletto, A. Gorelik, and R. Morris, "Practical Approaches to Dealing with DDoS Attacks," *NANOG22*, May 2001.

[33] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network Support for IP Traceback," *IEEE/ACM Transactions on Networking,* vol. 9, no. 3, June 2001

[34] A. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, and W.T. Strayer, "Hash-based IP Traceback," In *Proceedings of ACM SIGCOMM,* August 2001.

[35] I. Stoica, S. Shenker, and H. Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High Speed Networks," In *Proceedings of ACM SIGCOMM,* September 1998.

[36] B. Suter, T.V. Lakshman, D. Stiliadis, and A. Choudhury, " Design Considerations for Supporting TCP with Per-flow Queueing," In *Proceedings of IEEE Infocom,* March 1998.

[37] S. Templeton, "Detecting Spoofed Packets," Seminars - UC Davis Computer Security Laboratory, May 2002.

[38] Waldvogel, M. "GOSSIB vs. IP Traceback Rumors," In *Proceedings of 18th Annual Computer Security Applications Conference,* December 2002.

[39] H. Wang, D. Zhang, and K.G. Shin, "Detecting SYN Flooding Attacks," In *Proceedings of IEEE Infocom,* June 2002.

[40] S.F. Wu, W. Huang, D. Massey, A. Mankin, C.L. Wu, X.L. Zhao, and L. Zhang, "Intention-Driven ICMP Trace-Back," *Internet Draft, draft-wu-itrace-intention-02.txt*, Nov. 2001.

[41] Y.Xiong, S. Liu, and P. Sun, "On the Defense of the Distributed Denial of Service Attacks: An On-Off Feedback Control Approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* Vol. 31, No. 4. pp. 282-293, July 2001.

[42] A. Yaar and D. Song, "Pi: A Path Identification Mechanism to Defend against DDoS Attacks," In *Proceedings of IEEE Symposium on Security and Privacy 2003*.

[43] D.K.Y. Yau, J.C.S. Lui, and F. Liang, "Defending Against Distributed Denial-of-Service Attacks with Max-min Fair Server-centric Router Throttles," In *Proceedings of IEEE International Workshop on Quality of Service*, pp. 35-42, 2002

Yoohwan Kim is an Assistant Professor at School of Computer Science at the University of Nevada, Las Vegas. He received Ph.D. degree from Case Western Reserve University in 2003. His research interests include switch design, network traffic measurement, and network security.

Ju-Yeon Jo is an Assistant Professor at Department of Computer Science at California State University, Sacramento. She received her Ph.D. degree from Case Western Reserve University in 2003. Her research interests include Internet traffic management, network security and real-time embedded software.

Frank Merat is an Associate Professor at Electrical Engineering and Computer Science Department at Case Western Reserve University. He received a M.S. and Ph.D. in electrical engineering from Case Western Reserve University in 1975 and 1979 respectively. He is a Senior Member of the IEEE. His technical interests include RF electronics, wireless computer networks, micro-opto-mechanical devices, lasers and optics, image processing, intelligent sensors and machines, 3-D computer vision.

Mei Yang received her Ph.D. degree in Computer Science from the University of Texas at Dallas in 2003. She is an Assistant Professor in the Department of Electrical and Computer Engineering at University of Nevada, Las Vegas. Her current research interests include wireless sensor networks, network survivability and security, switch scheduling and control, computer architectures, and embedded systems.

Yingtao Jiang received the Ph.D. degree in Computer Science from the University of Texas at Dallas, in 2001. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas. His research interests include algorithms, VLSI architectures, and circuit level techniques for the design of DSP, networking, and telecommunications systems, computer architectures, and sensors.