2013 IEEE Security and Privacy Workshops

On Evaluating IP Traceback Schemes: A Practical Perspective

Vahid Aghaei-Foroushani
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
vahid@cs.dal.ca

A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, NS, Canada
zincir@cs.dal.ca

***Abstract*—This paper presents an evaluation of two promising schemes for tracing cyber-attacks, the well-known Deterministic Packet Marking, DPM, and a novel marking scheme for IP traceback, Deterministic Flow Marking, DFM. First of all we explore the DPM in detail and then by investigating the DFM, we analyze the pros and cons of both approaches in depth in terms of practicality and feasibility, so that shortcomings of each scheme are highlighted. This evaluation is based on CAIDA Internet traces October 2012 dataset. The results show that using DFM may reduce as many as 90% of marked packets on average required for tracing attacks with no false positives, while it eliminates the spoofed marking embedded by the attacker as well as compromised routers in the attack path. Moreover, unlike DPM that traces the attack up to the ingress interface of the edge router close to the attacker, DFM allows the victim to trace the origin of incorrect or spoofed source addresses up to the attacker node, even if the attack has been originated from a network behind a network address translation (NAT), firewall, or a proxy server.**

Keywords—Flow Base IP Traceback; DDoS Attacks; Deterministic Flow Marking; Authenticated Flow Marking

I. INTRODUCTION

The appearance of Denial of Service (DoS) attacks and its advanced derivative, Distributed Denial of Service (DDoS) attacks, quickly changed the perspective of network security. Even the network servers with high performance capacity can be overwhelmed easily by these types of attacks. In a DoS/DDoS attack, an offender may bombard a victim with thousands of attack packets in a short period of time. Because of the stateless nature of the Internet and prevalent attack tools, it is easy for an attacker to run an attack against a network resource without concern of being caught. It is the reason why DoS/DDoS attacks have been widespread. Due to the complexity of today's Internet and the trusting nature of the IP protocol, which originally did not include security as a design principle and so the source IP address of a packet is not authenticated, it is difficult for a victim to determine the source of DoS/DDoS attack.

So far, several approaches have been proposed to counter DoS/DDoS attacks. These approaches may be categorized into 4 groups: intrusion prevention, intrusion detection, intrusion mitigation, and intrusion response. This paper focuses on IP traceback, which belongs to fourth group, intrusion response. IP traceback techniques neither prevent the attack nor stop the attack; instead, they can be used to identify the source of violating packets during or after the attack. IP traceback is not only limited to DoS/DDoS attacks, the main purpose of IP traceback is to identify the real origin of any type of packets regarding the fact that the IP address of packets can be spoofed.

The main objective of this article is to evaluate and compare two promising schemes for tracing cyber-attacks, the well-known Deterministic Packet Marking, DPM, and a novel marking scheme for IP traceback, Deterministic Flow Marking, DFM, from the perspective of practicality and feasibility. We have employed the CAIDA Internet traces October 2012 dataset, and used a number of metrics to evaluate the performance of disparate traceback schemes. The metrics employed in this work are: the computational overhead, the memory overhead, the bandwidth overhead, the traceback rate, the false positive rates, mark spoofing by attackers or subverted routers in the attack path, the number of required packets for traceback, the percentage of marked packets, Internet service providers (ISP), the ability to handle fragmentation, the ability to handle major DDoS attacks, and the maximum traceback ability.

Our contributions are fourfold: (i) DFM reduces the number of required packets to be marked for tracing back. To measure this, we define a metric—the ratio of marked packets by the edge router to the total number of packets, to evaluate both traceback schemes. Our results show that using DFM may reduce the number of marked packets by 90%. (ii) DFM totally eliminates the threat of mark spoofing, not only if spoofed marking is inscribed by the attacker, but also if it is incurred by the compromised routers in the attack path. We show that this can be accomplished by using optional authenticated flow marking. (iii) DFM outperforms DPM in that it can handle larger scale DDoS attacks, because the maximum number of concurrent attackers in DPM is limited, whereas there is no such limitation in DFM. Finally, unlike DPM that traces the attack up to the ingress interface of the edge router close to the attacker, DFM allows the victim to trace the origin of the incorrect or the spoofed source addresses up to the attacker node, even if the attack has been originated from a network behind a NAT, firewall, or a proxy

DOI 10.1109/SPW.2013.13

IEEE computer society

server.

The rest of this paper has the following structure: Section 2 summarizes the related work on IP traceback and various traceback schemes are classified from multiple aspects. The actual schemes of DPM and DFM are presented, and implications and challenges associated with each of them are discussed from the perspective of practicality and feasibility in sections 3 and 4, respectively. Finally, we provide a comprehensive comparison table of both methods and present our conclusions in section 5.

II. Related Work

So far, many traceback approaches have been proposed. According to [1], [2] and [22], we classify existing approaches from multiple viewpoints. Three aspects are selected to classify existing traceback schemes into several categories. They include the basic principle, processing mode and location.

According to classification by the basic principle, Most of the existing traceback methods categorize into Logging and Marking groups. In logging methods, the routers keep some specific information of travelling packets [5]. For example, Snoeren et al. [3] have suggested generating a fingerprint of the packet, based upon the invariant portions of the packet (source, destination, etc.) and the first 8 bytes of payload. During the traceback, the routers can verify if a suspicious packet has been forwarded or not. Further improvement in terms of logging only a small portion of each travelling packet at the transient routers have been proposed in [4]. One of the major problems of the logging method is the requirement for high amount of memory and CPU usage on the routers in the attack paths [6]. In marking methods, some or all routers in an attack path send specific information along with traveling packets. The destination may use this information to trace the attacker even if the source IP has been spoofed. This information could be either embedded in the packet's IP header or sent by generating new packets and consume extra bandwidth [7], [8], [9], [10]. In particular, Savage et al. [11] have described a technique for tracing anonymous packet flooding attacks on the Internet back toward their source. This traceback can be performed after an attack is identified. While each marked packet represents only a sample of the path it has traversed, by combining a modest number of such packets, a victim can reconstruct the entire attack path. Dean et al. [12] have presented a scheme for providing traceback data by having routers embedding specific information into packets randomly. This is similar to the technique used by Savage et al. [11], with the major difference being that it is based on algebraic techniques. On the other hand, Song et al. [13] present two new IP marking techniques to solve the IP traceback problem: The Advanced Marking Scheme and the Authenticated Marking Scheme. The Authenticated Marking Scheme supports authentication of routers' markings. This prevents a compromised router from forging other uncompromised routers markings. Doeppner et al. [14] identify the source of Denial of Service attacks, provided that a significant percentage of packets are sent from one subnet. In this method, each router marks its own IP address to the travelling packet with a determinable probability. Moreover, Tseng et al. [15] have proposed a modification to the PPM [11] to ensure that the probability of receiving the mark is equal to the original marking probability. Yaar et al. [16] have proposed a method of encoding path identification by marking packets with path fingerprints. They have also another research [17] based on the PPM [11] with further improvements such as 1-bit distance. Victims can identify attack paths after receiving tens of packets encoding. It detects the distance of the attacker by changing the TTL field and storing 1 bit in the IP header. Goodrich et al. [18] have proposed to use relatively large, randomized messages to encode router information. The main idea is to have each router fragment its message into several words, then include a large checksum cord on the entire message randomly in the reusable bits of such a word fragment. Instead of the recovery of the full paths, Belenky et al. [20] and [21], proposed to only record the IP addresses of ingress edge routers. Their scheme, Deterministic Packet Marking (DPM), is simple and easy to implement, and has a little overhead on routers and the victim. Aghaei-Foroushani et al. [22] proposed the Deterministic Flow Marking (DFM) approach, which allows the victim to traceback the origin of an incorrect or spoofed source IP address up to the attacker node, even if the attack has been originated from a network behind a NAT or a proxy server. This scheme has low processing and memory overhead at the victim machines and edge routers. Additionally, DFM provides an optional authentication, so that a compromised router cannot forge markings of other uncompromised routers. Yang et al. [19] take advantage of both marking and logging methods and combines both approaches at routers in an attack path. Most marking methods [11], [13], [15], [16], [17], [22] use 16 bits of identification field. However, some other works propose to use 17 bits (identification field and reserved flag) [20], [23], 25 bits (identification and TOS fields plus reserved flag) [12], [18], [24], or 32 bits (identification field, flag and fragment offset) [9], [19], [22].

From the perspective of the processing mode based classification, traceback schemes can be categorized into two groups: deterministic and probabilistic. In deterministic methods, regardless of the marking or logging, every packet should be processed at both the source and the destination end. In comparison to the probabilistic methods, these methods require more processing overhead but higher accuracy. For example, Belenky et al. [20] embed the upper or the lower half of the IP address of the ingress interface into the fragment id field of the packet with a probability of 0.5. Then, they set a reserve bit indicating which portion of the address is contained in the fragment field. Aghaei-Foroushani et al. mark the first K packets of each flow by the combination of the egress interface IP address, the ingress interface ID, and the Host ID [22]. Most of the current traceback methods are probabilistic. While the required bandwidth and processing time in these methods are less than the ones required by the deterministic methods, the complexity for reconstruction at the destination side is more. Some well-known examples of probabilistic methods are PPM [11] and many of its variants [15] [17], ATA [12], iTrace [7] and others such as [13], [14], [16], [17], [18], [9].

From the perspective of the location based classification, existing traceback methods are divided into two types: those that send traceback information by the edge routers closest to the source (source group), and those that send traceback information by some or all routers in the attack path on the

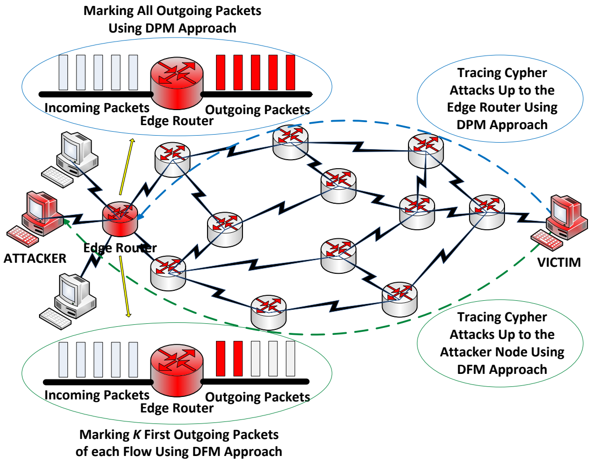

Fig. 1 A Schematic Illustration of both DPM and DFM Approaches

network (network group), respectively. Most of the current traceback methods belong to the network group [11], [12], [13], [14]. The purpose of these methods is to identify the attack path entirely or partially [15], [16], [17], [9]. The drawbacks of these methods are the requirement of involvement of all the routers along the paths and high resource consumption in terms of the processing time and memory [3], [19], [24]. While the goal of source group methods is to identify the attack source, they do not identify the attack path [18], [20], [22].

Furthermore, the proposed methods in [9], [24] and [25] trace up to the autonomous system (AS) level, while the other aforementioned works trace up to the edge router of the attack source. Song et al. [13] and Goodrich et al. [18] have proposed authentication marking methods, while the other aforementioned works send their marking information in clear text that are susceptible for alteration in the case of existing compromised routers in the network path.

III. Revisiting DPM

DPM is a well-known IP traceback approach and possesses several attractive features such as its ease of implementation, low computational and memory overhead on participating routers as well as the victim machines. Among all previous works described above, only DPM falls into the same category of classification as DFM. Properties of DPM are as follows: Basic principle: Marking, Processing modes: Deterministic, Location: Near the source. Therefore, we evaluate our approach, DFM, against DPM in the following sections.

A. DPM Scheme

Basic DPM was proposed by Belenky and Ansari [20]. They later improved their method in [21]. As it is shown in fig. 1, only the ingress interfaces of the edge router marks the packets, and the rest, including the backbone routers, are exempt. DPM uses 17 bits of the IP header, including 16 bits Identification field and 1 bit reserved flag, to embed the marking information to every packet. The 32 bits ingress interface IP address is split into two segments, 16 bits each: segment 0 – bits 0 through 15, and segment 1 – bits 16 through 31. When a packet passes through an edge router, one segment is selected with equal probability and inserted in the Identification field. The victim maintains a table matching the source addresses to the ingress addresses. When the victim gets both segments of an edge router, then it is able to reconstruct the whole ingress interface IP address of that router. One bit reserved flag plays the rule of a sign for the victim to identify which part of IP address is carried by the current packet. It should be noted that only incoming packets are marked, and outgoing packets are not marked. This ensures that the egress router will not overwrite the mark in a packet placed by an ingress router.

DPM has two key features: First of all, DPM only marks the closest ingress edge router to the attacker, and secondly, DPM marks all packets at the ingress interface of the edge routers.

Although the basic DPM approach can handle DoS attacks, it has high false positive rates under DDoS attacks. The reason behind this is that the victim associates segments of the ingress address with the source address of the attacker and the source IP addresses may be spoofed. Under such attacks, there are at least two cases when the edge router IP address reconstruction may not be ineffective under DPM. Firstly, two or more hosts that have the same source IP address attack the victim and secondly, (D)DoS attackers simply change the source address field for every packet they send. In these cases, the basic DPM is unable to reconstruct any valid ingress addresses [20]. To solve this problem, they improved their basic DPM approach to use a hash function to produce digests or hash values of the ingress address [21]. They proposed that all packets belonging to the ingress interface of an edge router carry the same hash value. Using this hash value, the victim is able to match the correct mark information to form a valid ingress IP address. Therefore, the marking information is formed by 3 parts: a segment of ingress address *a*, the index of segment *d*, and digest of ingress address *k*. They claimed that the best tradeoff for the size of each of these parameters are *a=4*, *d=3*, and *k=10*, all together 17 bits.

B. Analysis of DPM

1) Computational Overhead: The CPU overhead of DPM is lower than the previous IP traceback approaches like the well-known Probabilistic Packet Marking scheme, PPM. Because unlike PPM, in DPM only the closest edge router to the attacker is responsible for marking (not all routers in the attack path). Moreover, in DPM, there is no decision process for marking each packet. However, there are other computational overheads such as preparing marking information and upgrading marking fields. Having said this, in DPM, reconstructing the ingress interface IP address of the edge router is much simpler than the attack path reconstruction process of PPM approach. Therefore, in the face of DDoS attacks, the victim is able to traceback to the edge router in real time, if DPM is in use. Furthermore, the hash values of the ingress address may be used as a guide to effectively prevent the combinatorial explosion problem of PPM.

2) Memory Overhead: The memory overhead on the routers is negligible; and the victim keeps only a small reconstruction table. It is because DPM requires only *32/a* packets to reconstruct the ingress address (i.e. with the suggested *a=4* [21], DPM requires only 8 packets to traceback to the ingress interface address of edge router close to the attacker).

3) False Positive Rate: As discussed earlier, basic DPM method has a significant limitation to deal with multiple attackers at the same time with the same source IP address. In this situation, the victim cannot recognize which marked fragment should be concatenated together to form a valid mark, this causes high false positive rates. To counter this problem, they propose another method to use a hash function to produce hash values of the ingress interface, called single-digest DPM technique, or to use a family of hash functions to produce multiple digests of an ingress address, called multiple-digest DPM technique. In these techniques, these hash values are sent along with marked bits to effectively prevent the combinatorial explosion problem. This modification to DPM guarantees the false positive rate not to go over 1%, if the number of concurrent attackers in a DDoS attack is not more than a limited number. For example, using 55 datagrams to be marked by the DPM-enabled interface, the maximum number of simultaneous attackers that can be traced back with the false positive rate not exceeding 1% in the single-digest DPM technique are 45, and in the multiple-digest DPM technique are 2296 [21].

4) Mark spoofing by attackers: In DPM approach, each packet is marked when it enters the network. In this case, even if an attacker tries to spoof the mark, the spoofed mark will be overwritten with a correct mark.

5) Mark spoofing by subverted routers: DPM assumes that a mark remains unchanged for as long as the packet traverses the network. As DPM does not have any mechanism to authenticate the packet marking, this assumption automatically obviates the issue of mark spoofing by subverted routers in the attack path. Thus, in an untrusted network such as Internet, and in the case of a compromised router on the attack path, the marking information could be changed and the destination would be unable to identify the origin of the traffic.

6) Number of required packets for traceback: *32/a* packets are required to reconstruct the ingress address. By the suggested *a=4* [21], DPM requires 8 packets to traceback to the ingress interface address of the edge router close to the attacker, where *a* refers to the number of bits in a segment of ingress address field.

7) ISP Involvement: In this case, involvement of the Internet Service Providers (ISPs) is very limited. Only the edge routers have to be upgraded to support the function of deterministic packet marking. Unlike previous IP traceback approaches like PPM, the other routers in the attack path and the network backbone do not need to be responsible for any function of DPM traceback process.

8) Fragmentation: DPM uses the ID field in the IP header of packets, which is generally used for fragmentation, as well as 1 bit reserved flag to embed marking information. If only a single packet of a fragmented datagram is marked, then the datagram reassembly will fail.

C. *Motivations*

Although DPM has some good traceback features in IP traceback approaches, however DPM has still some problems such as the following:

- To keep the false positive rate not exceeding 1%, DPM cannot scale under heavy DDoS attacks as discussed above [21].
- DPM is able to traceback up to the ingress interface of edge router close to the attacker, not the exact attacker node.
- Although DPM has higher traceback accuracy in comparison to probabilistic marking approaches, this accuracy is achieved by marking all the packets in the network.
- DPM assumes that the marking information remains unchanged for as long as the packet traverses the network. Unfortunately, such an assumption is not realistic given the issue of mark spoofing by forged routers.

Thus, the aforementioned four problems were the motivation of proposing DFM approach [22].

IV. Revisiting DFM

A. *DFM Scheme*

DFM [22] is a promising IP traceback approach proposed by the authors.Unlike DPM, DFM marks every flow[1], (i.e. *K* first packets of each flow), instead of every packet, to have both advantages of “high traceback accuracy of DPM” and “marking only some packets” of probabilistic packet marking approaches like PPM. Moreover, DFM aims to trace the attack up to the source node(s) located on a LAN behind the edge routers. To this end, DFM uses three identifiers to mark a flow: (i) the IP address of the egress interface of the edge router; (ii) the NI-ID, which is an identifier assigned to each interface of either the MAC address of a network interface on the edge router or the VLAN ID of a virtual interface if the edge router uses VLAN interfaces; and (iii) Node-ID, which is an identifier assigned to each source MAC address observed on incoming traffic from local networks.

As it is shown in fig. 1, only the ingress interfaces of the edge router marks the packets, and the rest, including the backbone routers, do not involve in packet marking. DFM Marks each flow by 60 bits identification data including 32 bits IP address of the egress interface, 12 bits NI-ID, and 16 bits Node-ID, to distinguish the traffic of particular node from the other nodes. The 60 bits identification data is divided into *K* fragments; therefore the mark contains *M=60/K* bits of the identification data and *S=log2(K)* bits required to identify a fragment. DFM also takes advantage of one flag bit to identify marked and unmarked packets in a flow. The first *K* packets of every flow carry the mark fragments including *M* bits for

[1] Flow is a unidirectional sequence of packets between two networks with no more than 600 milliseconds inters packet delay time. A TCP/IP flow can be uniquely identified by source and destination IP address, source and destination port, and L4 protocol (TCP/UDP). An ICMP flow can be recognized by source and destination IP address, L4 protocol (ICMP), ICMP type, ICMP code and ICMP ID [22] [26].

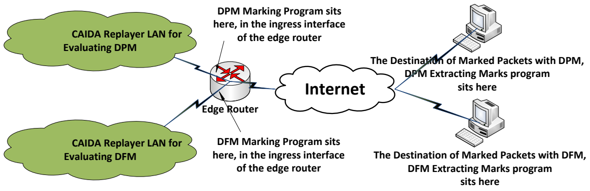
CAIDA Replayer LAN for Evaluating DPM
DPM Marking Program sits here, in the ingress interface of the edge router
The Destination of Marked Packets with DPM, DPM Extracting Marks program sits here
Internet
Edge Router
CAIDA Replayer LAN for Evaluating DFM
DFM Marking Program sits here, in the ingress interface of the edge router
The Destination of Marked Packets with DFM, DFM Extracting Marks program sits here

Fig. 2 Network testbed to analyze DFM and DPM techniques for IP Traceback

identification data fragment, *S* offset bits to represent 2^S possible fragments and one bit flag *F*, which should be set to “1” for the marked packets and “0” for the rest.

Each destination maintains a table matching the flow ID and *K* possible mark fragments. Flow ID has been defined as the five tuples of source IP address, destination IP address, L4 protocol type (TCP/UDP), source port and destination port for TCP and UDP flows. Moreover, it has been defined as the six tuples of source IP address, destination IP address, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID for ICMP flows. When a packet that belongs to an unseen flow arrives at the destination node, the node extracts the marking bits of this flow from the marked packets, identified by one bit flag. After all fragments corresponding to a flow reach the destination, the source node for the given flow becomes recognizable to the destination. Using DFM, the destination is able to distinguish the traffic of different nodes behind an edge router. As a result, when abnormal traffic is observed, the destination can filter the traffic of each node individually.

Unlike DPM that does not have any solution to counter to malfunctioned routers in the attack path, DFM has an optional authenticated flow marking feature to ensure that the marking information have not been changed in the network path. To this end, DFM utilizes Elliptic Curve Digital Signature Algorithm (ECDSA) [27]. The edge router creates 42 bytes signature value (size of ECDSA digital signature with 160 bits elliptical curve key) by applying ECDSA signing algorithm to 60 bits identification data plus 13 bytes flow ID. DFM adds this 42 bytes signature to the end of the K^{th} packet payload of each flow. Suppose an edge router wants to send a signed mark to a destination, it can use its own private key to do so. For each flow, it produces a 42 bytes signature value and sends it with the K^{th} packet of the flow. When the destination gets the signed flow, it uses the sender’s public key to authenticate the sender. If the two agree, the destination knows that the author of the mark was in possession of the edge router's private key, and that the mark is in fact valid, otherwise; it would reject the flow.

B. Practical Results

To evaluate DFM and compare the result with DPM, we have employed both approaches on the CAIDA anonymized Internet traces October 2012 dataset [28]. This dataset contains anonymized passive traffic traces from CAIDA's Equinix-Sanjose monitor on high-speed Internet backbone links. In other words, CAIDA data set we employed in this work is a standard tcpdump file. The Equinix-Sanjose Internet data collection monitor is located at an Equinix datacenter in San Jose, CA, and is connected to a backbone link of a Tier1 ISP between San Jose, CA and Los Angeles, CA. Currently it is a 10GigE link. This ISP has multiple links between these cities. Load balancing is done per flow. The infrastructure consists of 2 physical machines. Both machines have a single Endace 6.2 DAG network monitoring card. A single DAG card is connected to a single direction of the bi-directional backbone link. Both machines have 2 Intel Dual-Core Xeon 3.00GHz CPUs, with 8 GB of memory and 1.3 TB of RAID5 data disk, running Linux 2.6.15 and DAG software version dag-2.5.7.1. On the testbed network, both machines dropped less than 1% of the replayed packets with snaplen 48 at 100% OC192 line utilization, using a Spirent X/4000 packet generator sending packets with a quadmodal distribution, with peaks at 40, 576, 1500 and 4283 bytes [28]. We have chosen CAIDA anonymized Internet traces dataset, because it is publicly available, so it provides the possibility for other researchers to compare their own methods with our results, using the same dataset.

To analyze and evaluate our proposed method, we implemented a network, as it is shown in fig. 2. In this case, two different LANs, one for evaluating DPM and the other for evaluating DFM, have been setup. For replaying CAIDA dataset, we took advantage of Tcpreplay and Tcprewrite free applications. In addition, we implemented two real time programs for each approach, using winpcap library by C++, one for marking the flows running at the LAN side on the ingress interfaces of the edge router, and the other for tracing back the source of packets running at the destination side. The marking programs run at edge router and only mark those flows travelling from the network inside to the network outside. At the same time, the traceback programs run at destination node and try to detect the source of the marked traffic.

As described before, the mark inserted in each packet contains *M=60/K* bits of the identification data, S=*log2(K)* bits to identify a fragment and 1 bit flag to identify the marked and the unmarked packets in a flow. In [22], we have shown that the best tradeoff for the size of each of these parameters are either *M=30*, *S=1*, and *F=1* altogether 32 bits to embed in the identification field, flag and fragment offset of the IP header; or *M=12*, *S=3*, and *F=1* altogether 16 bits to embed only in the identification field of the IP header. To find the best tradeoff, we took advantage of *TR*, the ratio of the number of successful traced back packets to the total number of packets, and *MR*, the ratio of the marked packets to the total number of packets. Table I shows the evaluation of both DPM and DFM approaches on the CAIDA dataset, using the same *TR* and *MR* metrics. Note that *TR* for the DPM approach is less than 100% because fragmented traffic will be corrupted by the DPM, and there is some fragmented traffic in the CAIDA dataset. If a single fragment of the original datagram is marked, the reassembly function would fail at the destination.

The results show that marking the first 2 packets of every outgoing flow using DFM makes it possible to correctly determine the origin of ~93% of the packets (*TR*) or ~97% of the traffic in terms of bytes, while it only marks ~10% of all the packets (*MR*). Moreover, DFM correctly determines the

TABLE I. EVALUATING BOTH DPM AND DFM ON THE CAIDA DATASET, USING THE SAME METRICS

Comparison Metrics	**DPM**	**DFM with *K*=2**	**DFM with *K*=5**
Number of Marked Packets	241,589,706	24,059,752	32,470,758
MR	100%	9.96%	13.44%
Traced traffic in term of Number of Packets	23,948,7875	22,445,5742	219,912,254
TR in term of Number of Packets	99.13%	92.91%	91.03%
Traced traffic in term of Size (byte)	72,019,480,296	70,178,321,305	69,615,277,632
TR in term of traffic Size	99.26%	96.72%	95.95%

origin of 91.03% of packets (*DR*) or 95.95% of traffic in term of size by marking 13.44% of all packets (*MR*), if the first 5 packets of every outgoing flow are marked. Although the detection rate of DFM is less than DPM (i.e. the detection rate of DPM is 99.13%), however using DFM may reduce as many as 90.04% of marked packets on average with *K=2*, or it may reduce 86.56 % of marked packets on average with *K=5* with no false positives.

1) Memory Usage of DFM approach in the Edge Router: The requiring space for running DFM on an edge router is equal to the sum of required space of three tables including flow table, NI-ID table and Node-ID table [22]. In our practical analysis, the total required space for running our method with CAIDA dataset was less than 26 KB. Below we explain the details of each of these tables and the space occupied by them.

a) NI-ID table: For every edge router interface and in case of existence of VLANs, for every VLAN 9 bytes table record including 12 bits for NI-ID, 48 bits for MAC address and 12 bits for VLAN ID is stored. Because the implemented evaluation network has assigned one interface for evaluating DFM approach (fig. 2), this table only occupied 9 bytes in the edge router.

b) Node-ID table: For every record in the NI-ID table, DFM stores a separate Node-ID table. For every new observed source MAC address, a 60 bits record including 12 bits NI-ID and 48 bits MAC address should be stored. Thus the size of this table varies and is based on the number of unique observed source MAC addresses. Our DFM implemented program utilizes a memory management algorithm, so when it does not observe a source MAC address for a specific period of time, it removes its record from the Node-ID table. In our experimental results, the required space for storing Node-ID table was 890 bytes.

c) Flow table: In addition to NI-ID and Node-ID tables, DFM utilizes another table called the flow table. Each row in this table belongs to an observed flow. DFM stores 180 bits for each flow including the following 3 items:

- Flow ID, 13 bytes: For TCP and UDP flows, the flow ID is the sum of five tuples including 4 bytes source IP addresses, 2 bytes source port numbers, 4 bytes destination IP addresses, 2 bytes destination port numbers, and 1 byte protocol. That makes 13 bytes in total. For ICMP flows, the flow ID is the sum of 6 tuples including 4 bytes source IP addresses, 4 bytes destination IP addresses, 1 byte protocol, 1 byte ICMP type, 1 byte ICMP code and 2 bytes ICMP ID, altogether 13 bytes.
- Flow Mark, 60 bits as described earlier.
- Packet Number, 2 bytes: The edge router increases this number by one in the corresponding flow record for every transmitted packet. In other words, this number indicates the number of packets in a flow. DFM uses this number for keeping track of *K* first packets of every flow.

DFM no longer needs keeping the record of a flow when the flow is over. End of a flow is detected by an inter packet delay that is more than 600 ms. Therefore the space required for flow table varies and is based on the number of concurrent flows. Since the maximum number of concurrent flows in CAIDA dataset was 1131, the maximum required space to store flow table was about 25 KB.

2) Memory Usage of DFM approach at the Victim side: The victim maintains a reconstruction table, matching the flow ID and *K* possible mark fragments. For every observed flow, a 13 bytes flow ID and 60 bits identification data should be stored. Like the flow table in the edge router, the victim no longer needs keeping the record of a flow when a flow is over. Therefore, the space required for reconstruction table varies and is based on the number of concurrent flows. Since the maximum number of concurrent flows in CAIDA dataset was 1131, the maximum required space to store reconstruction table was about 23 KB.

3) Memory Usage of DPM approach in the Edge Router: Since marking process on the edge router by the DPM approach only stores the hash value of the ingress IP address, router's memory overhead in DPM algorithms is negligible.

4) Memory Usage of DPM approach at the Victim side: The Reconstruction Table consists of *f* parts, and each of those parts is a 2^{17} bit structure (2^{d} areas, *k* segments in every area, and 2^{a} bits in every segment) [21]. *f* refers to the number of hash value functions. We implemented DPM approach with the suggested 4 hash value functions [21]. Therefore the required space for reconstruction table was 64 Kb.

5) Bandwidth Usage of Authenticated DFM: Enabling the optional edge router authentication increases the network bandwidth usage given that an extra 42 bytes signature data is embedded to the end of the K^{th} packet of each flow. The amount of this increase can be observed by comparing the size of the transmitted traffic with and without the flow signing, Table II. This comparison shows that performing the optional edge router authentication has only about 0.2% bandwidth overhead with *K=2,* and about 0.08% bandwidth overhead with *K=5.*

6) Computational Cost of Authenticated DFM: To investigate the processing overhead on the edge router, we estimated the ability of DFM by computing the signing and verification of a 164 bits message including 60 bits identification data and 13 bytes flow ID. This estimation is

TABLE II. THE SIZE OF TRANSMITTED TRAFFIC WITH AND WITHOUT THE SOURCE EDGE ROUTER AUTHENTICATION IN BYTES

K	Marking without Authentication	Marking With Authentication	Increment
2	72,556,397,639	72,701,839,187	0.2%
5	72,556,397,639	72,618,776,795	0.08%

based on running our algorithm on a PC with 3.4 GHZ processor and Ubuntu 10.04 operating system. Our experimental results show that signing a flow takes less than half a millisecond. Moreover, if the optional authentication process is used, then verifying the digital sign of a flow takes less than one millisecond.

7) Memory Usage of Authenticated DFM: Since the digital signature of a flow is created and embedded to the flow at the time of sending the flow, no signature data is stored at the edge router. Thus, performing the authenticated flow marking method does not need any extra memory.

C. Analysis of DFM

1) Computational Overhead: Like DPM, only the closest edge router to the attacker is responsible for marking, and there are some computational overhead such as preparing marking information and upgrading marking fields. However unlike DPM, DFM does not require to calculate the hash value of the ingress address. Moreover, unlike DPM that extracts hash values of ingress addresses, the victim uses the flow ID as a guide to prevent the combinatorial explosion problem [21] in DPM. In addition, unlike DPM, DFM only marks *K* first packets of each flow, not all packets, and the victim extracts marking information from only those packets that the flag bit is set, again not all packets. Therefore, DFM has lower computational overhead than DPM. Moreover, as previously discussed, signing a flow takes less than half a millisecond and verifying the digital sign of a flow takes less than one millisecond by the authenticated DFM algorithm.

2) Memory Overhead: Like DPM, memory overhead on the routers in DFM approach is negligible (about 25 Kb), and at the victim side, DFM requires a small reconstruction table (23 Kb). This is even less than the DPM requirements (64 Kb).

3) False Positive: As discussed earlier, the DPM algorithm uses a limited number of bits for storing a hash value to prevent the combinatorial explosion problem, which then results in false positives in tracebacking an IP address under DDoS attacks when the number of attacks are more than the DPM can handle. On the other hand, the DFM algorithm uses the flow ID to prevent the combinatorial explosion problem. Therefore it does not face the false positive rates problem under the DDoS attacks.

4) Mark spoofing by attackers: DFM marks each flow when it enters to the edge router. In this case, even if an attacker tries to spoof the mark, the spoofed mark will be overwritten with a correct mark, once the flow passes through the edge router. Therefore mark spoofing by the attacker is not an issue of the DFM.

5) Mark spoofing by subverted routers: Unlike DPM that does not have any solution to counter the malfunctioned routers in the attack path, the DFM has an optional authenticated flow making feature to ensure that the marking information have not been changed in the network path.

6) The number of required packets for traceback: With the suggested *NB=32*, the DFM requires 2 packets and with *NB=16*, DFM requires 5 packets to traceback up to the attacker node [22]. This is lower than 8 packets required in DPM to traceback to the ingress interface address of the edge router.

7) ISP Involvement: Like DPM, the involvement of ISPs is very limited. Only the edge routers have to be upgraded to support the function of deterministic packet marking and the other routers in the attack path and the network backbone do not need to be responsible for any function of the DPM traceback process.

8) Fragmentation: Like the DPM, the DFM uses the ID field in the IP header of the packets, which is generally used for fragmentation. Thus if only a single packet of a fragmented datagram is marked, then the datagram reassembly will fail.

In addition to all the advantages of the DFM that are discussed above, there is one more unique feature that does not exist in any other traceback method. This is to enable the victim to trace the attack source, not only up to the source edge routers, but also to the exact source network interface of the edge router and then, to the source node(s) located in a LAN behind the edge routers. DFM assumes that each node in a local network may change its IP address, and the MAC filtering is enabled in the edge router. Moreover, the attacker may change its MAC address. However, in these cases, if the attacker changes his MAC address, DFM is still able to trace three levels up to the attacker node. Only in a case when the attacker spoofs his MAC address with several existing MAC addresses in the white list regularly, then the DFM can trace two levels up to the source network interface of the edge router.

Finally, as discussed earlier, using the proposed authenticated flow marking method is optional for the destination in the DFM approach. In a situation when the victim is under attack, it may use the signature to validate the mark to find the attacker node, otherwise the destination is not forced to consume its CPU and memory resources to verify ECDSA signature.

V. CONCLUSION

In this work, we performed an evaluation and a comparison of two IP traceback techniques, the well-known Deterministic Packet Marking (DPM), and a novel marking scheme for IP traceback proposed by the authors, Deterministic Flow Marking (DFM), from the perspective of practicality and feasibility. We employed the CAIDA Internet traces October 2012 dataset, and used a number of metrics to evaluate the performance of disparate traceback schemes, including the computational overhead, the memory overhead, the bandwidth overhead, the traceback rate, the false positive rate, mark spoofing by attackers or subverted routers in the attack path,

TABLE III. COMPARISON OF DPM AND DFM

Comparison Metrics	DPM	DFM
Percentage of marked packets	100%	If $K = 2$: 9.96% If $K = 5$: 13.44%
Mark Spoofing by subverted routers	Yes	No
Maximum traceback ability	Up to the ingress interface of the edge router	Up to the attacker node
Mark Spoofing by Attacker	No	No
Computational Overhead on routers	Low	Fair
Computational Overhead on victim	Low	Fair
Memory Overhead on routers	Low	Low
Memory Overhead on victim	Low	Low
Bandwidth Overhead	None	Low
Tracdeback Rate	Good	Fair
False Positive Rate	Low, but the number of concurrent attackers is limited	Low
Number of required packets for traceback	8	2 or 5
ISP Involvement	Low	Low
Ability to handle Fragmentation	No	No
Ability to handle major DDoS attacks	Fair, The Maximum Number of Concurrent Attackers is limited	Good
Number of Marking bits	17	If $K = 2$: 16 If $K = 5$: 32

the number of required packets for traceback, the percentage of marked packets, ISP involvement, the ability to handle fragmentation, the ability to handle major DDoS attacks, and the maximum traceback ability. Table III provides a summary of the evaluation and offers a comparison of two IP traceback techniques. The results show that DFM reduces the required number of packets for tracebacking accurately by 90% on average with no false positives. Moreover, DFM eliminates the spoofed marking embedded by the compromised routers in the attack path, and traces the attack source up to the attacker node, even if the attack has been originated from a network behind a NAT, firewall, or a proxy server. Future work will explore how to embed an IP traceback scheme such as DFM into existing security systems and frameworks.

ACKNOWLEDGMENT

This research is supported by the Natural Science and Engineering Research Council of Canada (NSERC) grant, and is conducted as part of the Dalhousie NIMS Lab at http://projects.cs.dal.ca/projectx/.

REFERENCES

[1] Z. Gao and N. Ansari, "Tracing Cyber Attacks from the Practical Perspective," IEEE Communications Magazine, vol. 43 , no. 5, pp. 123–131, May 2005.

[2] T. Subbulakshmi, I. A. A. Guru and S. M. Shalinie, "Attack source identification at router level in real time using marking algorithm deployed in programmable routers," ICRTIT 2011, pp.79-84, June 2011.

[3] A.C. Snoeren et al., "Single-packet IP traceback," IEEE/ACM Transactions on Networking, vol. 10 , No 6, pp. 721-734, Dec 2002.

[4] J. Li et al., "Large-scale IP traceback in highspeed Internet: practical techniques and theoretical foundation," IEEE/ACM Transactions on Networking, Vol. 16, no. 6, pp. 1253-1266, December 2008.

[5] S. Matsuda et al., "Design and implementation of unauthorized access tracing system," SAINT 2002, pp. 74–81, January/February 2002.

[6] A. Belenky and N. Ansari, "On IP traceback," IEEE Communications Magazine, Vol. 41, no. 7, pp. 142-153, July 2003.

[7] S.M. Bellovin, "ICMP traceback messages," IETF Draft, March 2000.

[8] S. Savage et al., "Network support for IP traceback," IEEE/ACM Transactions on Networking, Vol. 9, No 3, pp. 226-237, June 2001.

[9] Z. Gao and N. Ansari, "A practical and robust inter-domain marking scheme for IP traceback," The International Journal of Computer and Telecommunications Networking, Vol. 51, no. 3, pp. 732-750,Feb 2007.

[10] S.F. Wu, L. Zhang, D. Massey and A. Mankin, "On design and evaluation of intention-driven ICMP traceback," Proc. ICCCN2001, pp. 159-165, October 2001.

[11] S. Savage, D. Wetherall, A. Karlin and T. Anderson, "Network support for IP traceback," IEEE/ACM Transactions on Networking, Vol. 9, no. 3, pp. 226-237, June 2001.

[12] D. Dean, M. Franklin and A. Stubblefield, "An algebraic approach to ip traceback," TISSEC 2002, Vol. 5, no. 2, pp. 119-137, May 2002.

[13] D.X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback"INFOCOM 2001,Vol.2,pp.878-886, Apr 2001

[14] T.W. Doeppner, P.N. Klein and A. Koyfman, "Using router stamping to identify the source of IP packets," CCS 00, pp. 184-189, 2000.

[15] Y. Tseng, H. Chen and W. Hsieh, "Probabilistic Packet Marking with Non-Preemptive Compensation," IEEE Communications Letters, vol. 8, no. 6, pp. 359–361, June 2004.

[16] A. Yaar, A. Perrig and D. Song, "Pi: a path identification mechanism to defend against DDoS attacks" Proc. Symposium on Security and Privacy, pp. 93–107, May 2003.

[17] A. Yaar, A. Perrig and D. Song, "FIT: fast Internet traceback," INFOCOM 2005, Vol. 2, pp. 1395–1406, March 2005.

[18] M.T. Goodrich, "Efficient packet marking for large-scale IP traceback," CCS'02, pp. 117–126, November 2002.

[19] M. Yang, "RIHT: A Novel Hybrid IP Traceback Scheme," IEEE Transactions on Information Forensics and Security, Vol. 7, no 2, pp. 789-797, April 2012.

[20] A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," IEEE Commun. Lett., vol. 7, no. 4, pp. 162–64, April 2003.

[21] A. Belenky and N. Ansari, "On deterministic packet marking," Computer Networks: The International Journal of Computer and Telecommunications Networking,Vol.51. No.10, pp.2677-2700,Jul 2007

[22] V. Aghaei-Foroushani and N. Zincir-Heywood, "Deterministic and Authenticated Flow Marking for IP Traceback", The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013), March 2013.

[23] S. K. Rayanchu, and G. Barua, "Tracing Attackers with Deterministic Edge Router Marking (DERM)," ICDCIT'04, pp. 400–409, Dec 2004.

[24] M.D.D. Moreira, R. P. Laufer, N. C. Fernandes and O. C. M. B. Duarte, "A Stateless Traceback Technique for Identifying the Origin of Attacks from a Single Packet,", ICC 2011, pp. 1-6, June 2011.

[25] T. Hongcheng and B. Jun, "An Incrementally Deployable Flow-Based Scheme for IP Traceback," IEEE Communications Letters, Vol. 16, no. 7, pp. 1140-1143, July 2012.

[26] R. Alshammari, A. N. Zincir-Heywood, "Can Encrypted Traffic be identified without Port Numbers, IP Addresses and Payload Inspection?," Journal of Computer Networks, Elsevier, 2011.

[27] D. Johnson, A. Menezes and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom Corporation, Dept. of Combinatorics & Optimization, University of Waterloo, Canada,
http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf

[28] The CAIDA Anonymized Internet Traces 2012 Dataset,
http://www.caida.org/data/passive/passive_2012_dataset.xml,