A SECURITY FRAMEWORK BASED ON FLOW MARKING IP-TRACEBACK TECHNIQUES

by

Vahid Aghaei Foroushani

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
July 2016

I dedicate this thesis to my family. Without their unconditional love and ever-lasting support, nothing of this would have been possible.

Table of Contents

List of Tables

List of Figures

Abstract

Distributed-Denial-Of-Service (DDoS) attacks are one of the more challenging security issues on the Internet today. They can easily exhaust the resources of the potential victims. The problem is even more exacerbated since the attackers often forge their IP addresses to hide their identity. The existing defence mechanisms against DDoS attacks usually filter the attack traffic at the victim's side. In this case, even if the attacking traffic can be filtered by the victim, the attacker may reach the goal of blocking access to the victim by consuming the victim's computing resources or bandwidth. To address this issue, a modular security framework is proposed which consists of three main components: Detection, Traceback and Traffic Control. These three components can work independently as standalone systems, as well as collectively, bound by the proposed framework which aims to facilitate the replacement or addition of security modules without affecting the operation of the system as a whole. The Detection component aims to detect unusual changes of the incoming traffic to identify DDoS attacks. For the Traceback component five different approaches to IP-Traceback are proposed: Deterministic Flow Marking (DFM), Probabilistic Flow Marking (PFM), Unique Flow Marking (UFM), Deterministic Flow Marking for IPv6 Traceback (DFM6) and Autonomous System-based Flow Marking (ASFM). This component enables the identification of the origin of the traffic traversing through the Internet on a per flow basis, regardless of source IP address spoofing. The above five IP-Traceback approaches are designed for different network environments with varying network requirements. They all embed a fingerprint in the packets, but each one of them has some specific features and performances that make them suitable for various situations. For the traffic control component, Traceback-based Defence against DDoS Flooding Attacks (TDFA) is proposed. TDFA aims to place the packet filtering as close to the attack source as possible. In doing so, the traffic control component employs the IP-Traceback component to locate the origin of the attack and then sets up a limit on the packet forwarding rate to the victim. TDFA effectively reduces attack forwarding rate and improves the throughput of the legitimate traffic.

List of Abbreviations Used

AS	Autonomous System
AS-SPT	AS-level Single-Packet Traceback
ASBR	Autonomous System Boundary Router
ASFM	Autonomous System-based Flow Marking
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CA	Certificatee Authority
DDoS	Distributed Denial of Service
DFM	Deterministic Flow Marking
DFM6	Deterministic Flow Marking Encoding Module for IPv6
DFMD	Deterministic Flow Marking Decoding Module
DFMD6	Deterministic Flow Marking for IPv6
DFME	Deterministic Flow Marking Encoding Module
DFME6	Deterministic Flow Marking Decoding Module for IPv6
DGA	Data Generation Agents
DH	Diffie-Hellman
DOH	Destination Options Header
DoS	Denial of Service
DPM	Deterministic Packet Marking
EAST	Efficient AS DoS Traceback
FAST	Fast Autonomous System Traceback
FDPM	Flexible Deterministic Packet Marking

FIT	Fast Internet Traceback
FP	False Positives
GBF	Generalized Bloom Filter
IANA	Internet Assigned Numbers Authority
ICMP	Control Message Protocol
IR	Increasing Rate
ISP	Internet Service Providers
MAC	Media Access Control
MR	Marking Rate
NAT	Network Address Translation
NB	Number of Bits
NGN	Next Generation Network
NI-ID	Network Interface Identifcation
PF	Packet Filtering
PFM	Probabilistic Flow Marking
PFMD	Probabilistic Flow Marking Decoding Module
PFME	Probabilistic Flow Marking Encoding Module
Pi	Path Identifferer
PMTU	Path Maximum Transmission Unit
PPM	Probabilistic Packet Marking
PPPoE	Point-to-point over Ethernet
RIR	Regional Internet Registry
RL	Rate Limit

STM	SPIE Traceback Manager
TA	Traffic Adjustment
TCP	Transmission Control Protocol
TOS	Type Of Service
TR	Traceback Rate
TTL	Time To Tive
UDP	User Datagram Protocol
UFM	Unique Flow Marking
UFMD	Unique Flow Marking Decoding Module
UFME	Unique Flow Marking Encoding Module
VLAN	Virtual Local Area Network
WT	Waiting Time

Acknowledgements

Foremost, I would like to express my sincere and utmost gratitude to my supervisors, professors Nur Zincir-Heywood. Without her crucial guidance, genuine care and paramount support, I would have not been where I am today. As I move forward in my career, I promise her to abide with what she has taught me from the academic as well as the personal perspectives.

I would also like to express my appreciation to the staff of the faculty of Computer Science − Dalhousie University. I thank them for providing crucial aid and constant support throughout the five years that I have spent at Dalhousie University.

Finally, I would like to thank my wife Fariba Haddadi for her love and care and my family for their ever-lasting support.

Chapter 1

Introduction

a cyber-attack is any type of offensive maneuver employed by individuals or whole organizations which targets computer information systems, infrastructures, computer networks, and/or personal computer devices by means of various malicious acts usually originating from an anonymous source which steals, alters, or destroys a specified target by hacking into a susceptible system. These can be labeled as a Cyber campaign, cyberwarfare or cyberterrorism in different contexts. Cyber-attacks can range from installing spyware on a PC to attempts to destroy the infrastructure of entire nations [3].

Currently, the Internet is designed with functions in which security is not considered as the main goal. The Internet offers users fast, easy and cheap communication mechanisms at the network level which provide efficient approaches for different services on the network. These services are implemented at the end points: the sender and the receiver. With this implementation, malicious end users can easily violate the described policy of the different protocols and act to damage the other party.

1.1 Context and Motivation

Denial of Service attacks, DoS, or Distributed Denial of Service attacks, DDoS, form one of the major sources of attacks on today's computer networks and have become a serious threat on the Internet due to the available variety of DoS forms, sizes, types, and used tools. Typically, in a DoS attack an attacker can flood a victim's connection with random packets to prevent the legitimate traffic from getting through, causing congestion and resource consumption on the intermediate routers, as well as the end systems. A DDoS attack is a variant of the DoS attack in which the attacker can increase the machines involved in the attack from one causing a classic DoS attack to many compromised machines causing a DDoS attack. A DDoS attack has the capability of exhausting a victim's computing and communication resources within a short period of time. DDoS attacks mainly affect the consumption of computational resources such as bandwidth, disk space or CPU time. Sometimes, they may cause disruption of network configuration information and may also disrupt physical network components. Such attacks are among the hardest security problems to address, because they are simple to implement, hard to prevent, and very difficult to trace. In the last several years DDoS attacks have increased in frequency, intensity and sophistication. This has become more widespread recently due to the ease of execution and near untraceability.

Two features of the DDoS attacks are challenging for security (defence) systems. Firstly, the DDoS packets tend to appear as legitimate packets, so filtering them with no impact on the legitimate traffic is challenging. There is usually no explicit attack pattern to distinguish legitimate packets from malicious ones. Secondly, identifying the sources of large-scale DDoS attacks is a challenging task. The reasons for this are threefold:

1. IP routing is based on the destination IP address, not the source IP address.

2. Due to the trusting nature of the IP protocol, which originally did not include security as a design principle, the source IP address of a packet is not authenticated. Attackers are usually interested in hiding their identity with fake addresses. The attackers use spoofed or incorrect IP addresses by filling the source IP header fields with randomized values, known as IP spoofing, or by using a NAT or a proxy device and therefore, the real origin of such attacks remains hidden.

3. Usually no information about packet forwarding is kept at the intermediate routers. The combination of these issues guarantees the anonymity of the attackers.

IP spoofing enables the attacker to make the attacking traffic to appear as if it comes from a totally different network. When the administrator of the victim network tries to solve the problem by blocking all traffic from the apparent attack source, he/she is in fact taking action against innocent systems, thereby contributing to the (D)DoS attack. In fact, simple detection/prevention tools which limit the traffic from such attacks already exist. However, such tools are not useful when the source addresses are spoofed.

Besides the DoS/DDoS attacks, other forms of attacks, such as network intrusions, are also typically carried out with spoofed IP addresses. According to the IP spoofer project at MIT, 13% of source IP address were spoofable and 4.6% were inconsistent, totalling 17.6% of all announced IP addresses as vulnerable. [1]. CAIDA also demonstrated that the United States and China are the two major victim countries of IP spoofing attacks [2].

To limit source IP address spoofing, some routers employ a mechanism called ingress filtering [22], but this mechanism requires a router with sufficient knowledge to distinguish between legitimate and illegitimate addresses. As traffic is aggregated from multiple ISPs into transit networks, there is no longer enough information to determine if a packet arriving on a particular interface has a legal source address. Furthermore, modifying the source IP address is employed (one can say legal spoofing) on the Internet by network address translation (NAT) servers, proxies, mobile IP servers, and other unidirectional link technologies such as hybrid satellites.

IP-Traceback is an alternative mechanism which aims to identify the true source of an IP datagram despite source IP address spoofing.

IP-Traceback is a critical ability for identifying the sources of attacks and instituting protection measures on the Internet. Most existing approaches to this problem have been tailored toward DoS attack detection. However, most of the state-of-the-art IP-Traceback approaches have the following problems:

- Even though the traceback rate of Deterministic IP packet-based Traceback approaches is high, their marking rate is very high, too.

- On the other hand, the marking rate of Probabilistic IP Packet-based Traceback approaches is low. However, their traceback rate is low, too.

- Both of the above approaches are able to traceback only up to the edge router of the attacker network, not the attacker node, because proxies and NATs make it difficult to differentiate activities from distinct hosts. In this case, to defend against attacks, the victim filters out all traffic belonging to the attacker network after identifying the origin of the attack, in fact taking action against innocent systems, thereby contributing to the (D)DoS attack.

- The victim needs to reconstruct the exact path that the packets follow. To do so, the victim must receive a large number of the marked packets, and be aware of the network map and routing in advance.

- The routers among the attack path should perform some extra operations to mark or log the packets or to reconstruct the path to the attacker.

- Previous work on IP-Traceback generally requires deployment over all routers (Probabilistic Packet Marking -PPM- approaches) to rebuild the complete path taken by attacker traffic, or all ISP edge routers (Deterministic Packet Marking - DPM- or Deterministic Flow Marking -DFM- approaches) to discover the source network of the attacker traffic. Naturally, these requirements limit the potential deployment of IP-Traceback solutions on the Internet.

- Although the IP-Traceback approaches allow the victim to traceback to the source of an attack, in general they do not have the ability to decrease the impact of the attack while the attack is in progress.

1.2 Objectives and Contributions

The main objective of this thesis is to explore how far one can push a defence/security system to mitigate the impact of the IP spoofing attacks, which are mainly used in DDoS attacks, while the attack is in progress. To achieve such a solution a modular security framework is presented to address the above challenges. The proposed framework uses the strategy of detecting the attack and finding the source of the attack at the victim-end and responding to the attack by controlling the traffic forwarding as away as possible from the victim. This security framework consists of three main components: Detection, Traceback and Traffic Control, Figure 1.1.

The Detection component aims to detect unusual changes in the incoming traffic to identify DDoS attacks. There are several algorithms and tools for detecting DDoS attacks which can be used for this component (such as Snort [13]). The intention is

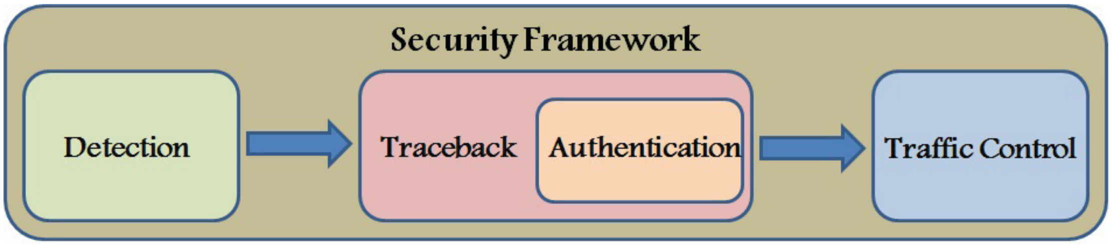


Figure 1.1: A Schematic Illustration of the Proposed Security Framework

not to propose a new DDoS detection algorithm, but to use the proposed Traceback component with existing detection systems.

For the Traceback component, five different novel flow-based IP-Traceback approaches, namely Deterministic Flow Marking (DFM), Probabilistic Flow Marking (PFM), Unique Flow Marking (UFM), Deterministic Flow Marking for IPv6 Traceback (DFM6) and Autonomous System-based Flow Marking (ASFM) is proposed. This component permits the identification of the origin of the traffic traversing through the Internet on a per flow basis, regardless of the source IP address spoofing. The five IP-Traceback approaches are designed for different network environments with varying network requirements. They all embed a fingerprint in the packets, but each one of them has different features and performances which make them suitable for various situations. To this end, DFM marks every flow, PFM marks some of the flows which are chosen randomly, and UFM marks only the unique flows in the network traffic. The main concept behind the UFM approach is that once the victim finds the origin of a packet in a flow, then the origin of any other packet in that flow and all packets in any other flow with the same flow identity are discovered as well. Given that most of the current IP-Traceback approaches, as well as DFM, PFM and UFM, are based on IPv4, they are not suitable to be applied directly on IPv6 networks. Therefore the use of DFM6 is proposed, being designed mainly for IPv6 networks. Although some security threats were taken into consideration in the IPv6 design, DDoS attacks still exist in IPv6 networks. Finally, the ASFM traceback method operates on the Boarder Routers of Autonomous Systems. ASFM identifies some key points in the path where attacker packets are being forwarded. This enables efficient countermeasures to be taken in a distributed mechanism to block the ongoing attack (e.g., at the closest traceback collaborative ASs with respect to the sources of a DDoS attack, or at the AS which forwards more traffic). Additionally, the Traceback component provides an optional Authentication sub-component for DFM, PFM, UFM and ASFM IP-Traceback approaches, so that a compromised router cannot forge markings of other uncompromised routers. The main characteristics of the proposed IP-Traceback methods that distinguish them from other methods include the following:

- The main concept behind the proposed flow-based IP-Traceback approach is that once the victim finds the origin of a packet in a flow, then the origin of any other packet in that flow is also discovered. This results in lower marking, processing, bandwidth and memory overhead in the flow-based IP-Traceback approaches compared with the packet-based IP-Traceback approaches. The definition of a flow is accepted as a unidirectional sequence of packets between two endpoints within a certain time period that have five-tuple information in common, including the source IP address, the destination IP address, the L4 protocol type (TCP/UDP), the source port number and the destination port number. An ICMP flow is defined as a unidirectional sequence of ICMP packets between two networks that have the six-tuple information in common, including the source IP address, the destination IP address, the L4 protocol type (ICMP), the ICMP type, the ICMP code and the ICMP ID.

- Most of the traceback methods assume 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. The proposed flow-based IP-Traceback approaches eliminate 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. This can be accomplished by using the Authentication sub-component.

- Finally, unlike the other traceback methods that are able to traceback only up to the AS level or at best, the edge router of the attacker network, DFM, PFM, UFM and DFM6 allow 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 server.

The traffic control component aims to mitigate the impact of the attack. The proposed Traffic Control component defends against DDoS attacks by coordinating between the defence systems at the source and victim ends. This necessitates communication between the victim and the filtering routers so that the filtering can be located as far away as possible from the victim. Once a DDoS attack is detected by the Detection component, the Traceback component finds the source of the attack. Then the Traffic Control component sends traffic control messages to the edge router of the attack network. When the edge router of the attack network receives the control messages, it will be triggered to adjust the packet forwarding rate to the victim. TDFA filters attack traffic at the source end to eliminate the consumption of computing resources and the bandwidth of the victim. In return this improves system performance for the legitimate services and users. In summary, the objectives of the Traffic Control component are:

- To minimize the burden of filtering on the participating routers; and
- To maximize the survival rate for the legitimate traffic under an intensive attack.

These three components can work independently as standalone systems, as well as collectively, bound by the proposed framework. The proposed framework aims to facilitate the replacement or addition of security modules without affecting the operation of the system as a whole.

1.3 Organization

The rest of this thesis is organized as follows. Chapter 2 summarizes the related work on IP-Traceback and DDoS defence systems, and also various traceback schemes are classified from multiple aspects. In Chapter 3, the implications and the challenges associated with two well-known IP-Traceback approaches, Deterministic Packet Marking (DFM) and Probabilistic Packet Marking (PPM) are presented to be able to compare the proposed flow-based IP-Traceback approaches with them. Most of the current IP-Traceback techniques are variants of these two approaches. Then, the five proposed schemes for the Traceback component including DFM, PFM, UFM, DFM6 and ASFM and the Authentication sub-component are introduced and discussed from the perspective of practicality and feasibility in Chapters 4, 5, 6, 7 and 8 respectively. In Chapter 9, the Traceback-based Defence against Flooding Attacks (TDFA) is presented as an example of the proposed framework where all three components work together. Finally, Chapter 10 concludes this thesis, summarizes its contributions and highlights some new research directions for future work.

Chapter 2

Literature Review

This chapter presents a survey of previous works on the IPv4 and IPv6 Tracebacks, as well as a review of previous works on DDoS defence systems.

2.1 IPv4 Traceback

So far, several different traceback approaches have been proposed. In accordance with Gao and Ansari [57] and Subbulakshmi and Shalinie [84]], existing approaches are classified from multiple viewpoints. As shown in Fig. 2.1, five aspects are selected to classify existing traceback schemes into different categories. They include the basic principle, processing mode, location, level of operation and operational mode.

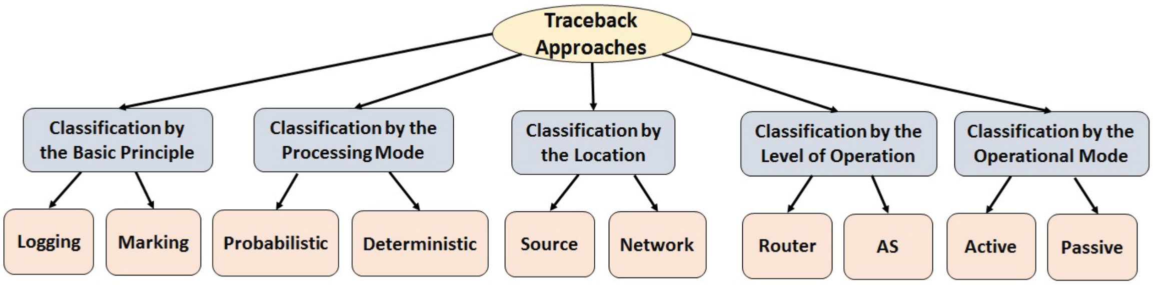

Traceback Approaches
Classification by the Basic Principle
Classification by the Processing Mode
Classification by the Location
Classification by the Level of Operation
Classification by the Operational Mode
Logging
Marking
Probabilistic
Deterministic
Source
Network
Router
AS
Active
Passive

Figure 2.1: Categorizing IP traceback schemes

2.1.1 Basic Principle

According to classification by the basic principle, most of the existing traceback methods are categorized into Logging and Marking groups. In logging methods, the routers keep some specific information of the travelling packets [49]. For example, Snoeren et al. [47] have suggested generating a fingerprint of the packet,-based upon the invariant portions of the packet (source, destination, etc.) and the first eight bytes of the payload. During the traceback, the routers can verify if a suspicious packet has been forwarded or not. Further improvements in terms of logging only a small portion of each travelling packet at the transient routers have been proposed in [48]. One of the major problems of the logging method is the requirement for a high amount of memory and CPU usage on the routers in the attack paths [37]. In marking methods, some or all routers in an attack path send specific information along with the traveling packets. The destination may use this information to trace the attacker even if the source IP has been spoofed. This information could be embedded either in the packet's IP header or sent by generating new ICMP packets [39], [50], [58], [78]. Marking in the headers of the packets does not require extra bandwidth, while the number of bits that may be used for marking is rather limited. On the other hand, there are far more bits available by generating new ICMP packets than marking in the header of the packets. However, marking by new ICMP packets needs extra bandwidth, which may further aggravate the performance of the network being attacked. Moreover, a new ICMP message must be introduced into the Internet with no ICMP filtering. Otherwise, the ICMP message may be blocked due to ICMP filtering.

Bellovin et al. proposed an ICMP traceback method called iTrace [39]. In this approach, each router selects one packet per 20,000 packets and then generates an ICMP message. The ICMP message has the same destination IP address as the traced packet. The ICMP message also contains the IP header of the traced packet, the IP addresses of the incoming interface and the outgoing interface of the current router. As long as the victim receives sufficient ICMP messages, it should be possible to recover the whole attack path. In ICMP traceback, the TTL field in the IP header of the ICMP message is set to 255 so that the TTL value may be used as a clue to reconstruct an attack path correctly. Wang et al. proposed an ICMP Caddie Messages scheme (Caddie) [89], a variant of the ICMP messaging technique for IP traceback. The scheme provides a DoS-resistant solution for various network security issues. A Caddie message is an extra ICMP message generated by a router, called the Caddie initiator, attached with the entire packet routing history of one randomly selected packet, called the Ball packet, forwarded by the router. In other words, while a router is forwarding packets, it selects one of the packets randomly as a ball packet, and then generates a Caddie message following the ball packet. The Caddie message will collect the path information about the sequence of the router's identities along the way toward the ball packet's destination. Accordingly, the destination can construct a traffic source tree structure by simply composing the Caddie travel paths inscribed in the Caddie messages received. When a host receives an abnormal number of Caddie messages from some sources, the network administer should consider initiating an attack path construction process to identify the sources.

Savage et al. [50] have described a technique for tracing anonymous packet flooding attacks on the Internet toward their source by embedding the marking information in the header of the packets. 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. [42] have presented a scheme for providing traceback data by having routers embed specific information into packets randomly. This is similar to the technique used by Savage et al. [50], with the major difference being that it is based on algebraic techniques. On the other hand, Song et al. [79] 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 the authentication of the routers' markings. This prevents a compromised router from forging other uncompromised routers' markings. Doeppner et al. [86] can identify the source of Denial of Service attacks, provided a significant percentage of packets are sent from one subnet. In this method, each router adds its own IP address to a travelling packet with a determinable probability. Moreover, Tseng et al. [91] have proposed a modification to Probabilistic Packet Marking (PPM) [50] to ensure that the probability of receiving the mark is equal to the original marking probability. Goodrich et al. [60] have proposed the use of relatively large, randomized messages to encode router information. The main idea is to have each router fragment its message into several words and to 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. [36] and [38], propose only to record the IP addresses of the ingress edge routers. Their scheme, Deterministic Packet Marking (DPM), is simple and easy to implement, and has little overhead on the routers and the victim. This scheme has low processing and memory overhead for the victim machines and edge routers. Additionally, DFM provides an optional authentication, preventing a compromised router from forging the markings of other uncompromised routers. Yang et al. [94] and Dong Yan et al. [44] take advantage of both marking and logging methods, combining both approaches at the routers in an attack path. Most marking methods use 16 bits of identification field such as [50], [79], [91], [26] and [28]. However, some other works propose to use 17 bits (identification field and reserved flag) [38], [76], or 25 bits (identification and type of service (TOS) fields plus reserved flag) [42], [60], [67], [92], or 32 bits (identification field, flag and fragment offset) [58], [94].

2.1.2 Processing Mode

From the perspective of classification based on the processing mode, traceback schemes may 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 side. In comparison to the probabilistic methods, these methods require more processing overhead but their advantage is providing higher accuracy. Deterministic processing may be indispensable for more advanced security services, such as nonrepudiation.

Yaar et al. [28] have proposed a probabilistic IP-Traceback approach based on PPM [50], called Fast Internet Traceback (FIT). Victims can identify attack paths after receiving tens of marked packets. It detects the distance of the attacker by changing the time to live (TTL) field and storing 1 bit in the IP header. The proposed method allows all FIT-enabled routers in path to be identified. The FIT traceback mechanism consists of two major parts: a packet marking scheme to be deployed at the routers and path reconstruction algorithms used by end hosts receiving the packet marking. The victim has a map of the upstream router and their IP addresses using the packet marking received before the attack itself occurs. The packet marking scheme itself consists of three elements: a fragment of the hash of the marking router's IP address, the number of the hash fragment marked in the packet, and a distance field. Based on the distance field and the TTL of a given packet, the attack victim can determine from how many hops away the marking is generated. The victim uses the hash fragment and distance calculation from the marking in the malicious packets in conjunction with its router map to identify a candidate set of marking routers. After a number of different hash fragments matching a particular router arrive at the victim, that router is added to the reconstructed attack path. Liu et al. proposed an IP-Traceback method, called ERPPM, which marks the packets with a dynamic optimal marking probability to ensure that the victim receives all the marked packets with equal probability [87]. It can greatly reduce the number of packets needed to reconstruct the attacking path. This model can reduce the possibility that a marked packet is remarked again by the router nearer to the victim. The marking probability in each router is adjusted according to the distance between the router and the victim. This minimizes the total number of packets required for path reconstruction, hence minimizing the time needed to reconstruct the attacking path.

There is some research being done on deterministic marking as well. For example, the suggested idea by Belenky and Ansari [38] is to store, with a random probability of 0.5, the upper or the lower half of the IP address of the ingress interface into the fragment id field of the packet, and then set a reserve bit indicating which portion of the address is contained in the fragment field. The proposed method by Rayanchu and Barua [76] is similar to Belenky and Ansari [38], but the difference is that they do not embed the IP address in the IP header; instead they only embed the hash of the edge router's IP address. Yaar et al. [26] proposed a Path Identification Mechanism Pi (Path Identifier), a deterministic packet marking approach in which a path fingerprint is embedded in each packet, enabling a victim to identify packets traversing the same paths through the Internet on a per packet basis, regardless of source IP address spoofing. In this approach an identifier is embedded in each packet, based on the router path that a packet traverses. Since the packet marking is a per packet deterministic mechanism, each packet traveling along the same path carries the same identifier. The victim needs only determine one packet as malicious to be able to filter all subsequent packets with the same marking. This allows the victim to take a proactive role in defending against a DDoS attack by using the Pi mark to filter out packets matching the identifiers of the attackers' on a per packet basis. Xiang et al. [92] proposed Flexible Deterministic Packet Marking (FDPM) to perform a large-scale IP traceback to defend against Distributed Denial of Service attacks. The flexibilities of FDPM are twofold: one is that it can adjust the length of the marking field according to the network protocols deployed; and the other is that it can adjust the marking rate according to the load of the participating routers.

When an IP packet enters the protected network, it will be marked by the interface close to the source of the packet on an edge ingress router. The mark will not be changed when the packet traverses the network. At least two packets are needed to carry a 32-bit source IP address. The reconstruction process includes two steps, one is mark recognition, and the other is address recovery. As with DPM, their scheme records only the IP addresses of the ingress edge routers, but unlike DPM, FDPM changes its marking rate according to the load of the participating router.

Most of the recent traceback methods are probabilistic. While the required bandwidth and processing time in these methods are less than those required by the deterministic methods, the complexity for reconstruction at the destination side is higher. Some well-known examples of probabilistic methods are PPM [50] and many of its variants [91] [28], ATA [42], iTrace [39] as well several others [79], [86], [26], [28], [60], [58].

2.1.3 Location

From the perspective of classification by locations, existing traceback methods are divided into two types: those that send traceback information by the edge routers closest to the source (source group), and in the network by some or all routers in the attack path (network group). DPM is an example which performs marking near the source (edge routers closest to the source). Most of the current traceback methods belong to the network group [50], [42], [79], [86]. The purpose of these methods is to identify the attack path entirely or partially. That is, the routers (some or all) in the network perform certain processing (marking or logging), either stochastically or deterministically, and inscribe the required information into the packets [91], [26], [28], [58]. The drawbacks of these methods are the involvement of the routers along the paths and the cost of their processing times and memories for this purpose [47], [94], [67]. While the goal of source group methods is to identify the attack source, they do not identify the attack path. One benefit of the source group approaches is that the victim is greatly relieved from the heavy computational and storage burden [60], [38], [76], [92].

2.1.4 Level of Operation

Most of the current IP-Traceback approaches, and all approaches described in the previous sections, are operating at the router level. However, there are numerous techniques which work at the autonomous system (AS) level as opposed to individual routers. Inter-domain traceback has some advantages over the router-level traceback. Using the AS in traceback will not reveal the topology, which is advantageous for network operators. Additionally, the number of AS hops is significantly less than the number of router hops. Therefore, it is easier to traceback through ASs than routers.

Authenticated AS traceback [72] is proposed to traceback the attack on the AS level. The technique is based on PPM [50] with a difference in the marking probability and the encoded IP address. This technique uses a fixed marking probability ($P = 1/6$) and encodes the AS number (ASN) instead of the IP address. The technique was proposed before the migration for the ASN from 16 bits to 32 bits. Since January 2010, Regional Internet Registries (RIRs) [11] have assigned a 32-bit AS number to new ASs instead of the 16-bit AS number.

Korkmaz et al. [59] propose an AS-level single-packet traceback (AS-SPT) mechanism which could operate in a partial deployment scenario and is based on packet logging. The victim AS asks its neighbouring ASs to check their logs and identify which of them sent the packet. When a neighbour replies positively, the victim AS would then repeat the same process recursively with the neighbors of that AS until arriving at the source AS. Among the disadvantages of this approach is that the success rate drops quickly as the number of ASs deploying this approach goes down. Also, it generates a rather large number of messages during traceback. Moreover, the traceback process has to be done for every single packet being traced. It also requires previous knowledge of the network topology.

Durresi et al. proposed the Fast Autonomous System Traceback (FAST) [71], an inter-domain traceback system in which the first five ASs mark the packet. The marking field is divided into five subfields to accommodate the five marks. The marking is made by the border router of each AS. The border routers insert the hash of their AS number (ASN) into the forwarded packets. FAST can only record the information of 5 ASs; The path reconstruction process needs more packets when the attacker is only 1 or 2 hops away from the victim. Victims have to know the AS-level network topology and routing information in advance. Furthermore, it cannot identify mark forging.

In Castelucio et al. [41], packets are marked when they pass through an AS border router taking part in the traceback system. This packet marking is performed by the insertion of information related to the AS into a Generalized Bloom Filter (GBF) present in the header of the packet. Basically, the information inserted into the GBF is the result of a hash function between the Autonomous System Number (ASN) combined with the TTL of the packet at the moment that the packet passed through the router. At the destination AS, the victim checks which of its neighboring ASs have their mark present in the filter. It would then check to see which of the identified neighboring ASs have their marks in the packet and so on until it reaches the source AS. In short, with this method marking data is added as an IP option which significantly slows down router performance.

Efficient AS DoS traceback (EAST) [32] is a PPM [50]-based approach and is proposed for identifying the AS of the attacker by probabilistically inserting 25-bit marking information into the packet headers. The results show better performance than PPM. However, this scheme still has a very similar computational overhead to that of PPM at the destination.

2.1.5 Operational Mode

All approaches described in the previous sections are Active IP-Traceback approaches. However, there are several research on Passive IP-Traceback (PIT) [55, 56, 68, 66, 63], which simply means an IP-Traceback approach that bypasses the challenges of deployment. All PIT variants are based on the same underlying assumption that routers may fail to forward an IP spoofing packet due to various reasons, e.g., TTL exceeding. In such cases, the routers may generate an ICMP error message (named path backscatter) and send the message to the spoofed source address. In this case, if the routers are close to the spoofers, the path that the backscatter messages take may disclose the locations of the spoofers. PIB takes advantage of these backscatter messages to find the attackers location. However all of the above assumptions may not happen. In addition, while most of existing IP-Traceback approaches focus on rapid attacks, PIT is interested in events related to slow stealthy attacks. Yao et al. [55] discussed that their proposed mechanism certainly does not work in all the attacks and cannot capture all the spoofers, but it does tell something about the spoofing attacks. At least, the luckiest victims are able to locate some of the spoofers.

2.2 IPv6 Traceback

The current IPv6 traceback approaches can be categorized into either logging methods or marking methods. As an example of the logging method, Strayer [82] modified the original SPIE algorithm [47] to make it work for IPv6 networks. It consists of three components: SCAR (SPIE- Collection and Reduction Agents), DGA (Data Generation Agents) and STM (SPIE Traceback Manager). The STM should be in direct communication with an external intrusion detection system (IDS) and its task is to control the IP-Traceback process. Once an attack occurs, the IDS informs the STM. The STM then sends a message to all the SCARs, which collect the data from the DGAs, and receives the reply. DGA produces a packet digest of each packet and stores the digest in a time-stamped digest table. The STM verifies the data with the information received by the IDS and creates the attack graph. Finally the attack graph is sent to the IDS. The main limitation of this approach is that to generate a sufficient entropy and to describe a packet to be identified uniquely, the IPv6 algorithm requires 1232 bytes of digest input data, whereas the SPIE IPv4 algorithm needs 28 bytes.

In the marking group, some approaches employ the 20-bit flow label field in the IPv6 header as the marking field and embed the marking information in that field. In particular, Xuan [45] proposed a probabilistic packet marking approach for the IPv6 networks. The marking data has two parts: a 14-bit edge fragment field and a 6-bit distance field. The outgoing packets are selected randomly by intermediate routers. Once a router selects a packet, it writes the hash of its own IP address in the edge fragment field and marks the distance field by zero. If the next router selects the same packet for marking, it XORs the hash of its own IP address with the previous value of the edge fragment field, then overwrites the new value in the edge fragment field, and increases the distance value by one. Otherwise, the router only increases the distance value by one. Yang et al. [95] proposed a link signature-based IP-Traceback algorithm for the IPv6 networks. Each link in the network has a unique signature. Each router uses this signature for marking the travelling packets. Marking involves XORing the packet signature area with the signature of the egress link, and storing the new value in the flow label field of the IPv6 header. They assume that the victim should have the entire network topology and the link signature of each network link. The main problem of this group of IPv6 traceback approaches is that overwriting the flow label field interferes with the main function of this field. However, the flow label field is used primarily for identifying different flows and serves as a hint to routers and switches with multiple outbound paths that these packets should stay on the same path so that they should not be reordered [25, 23].

There are some other IPv6 approaches that fall into the marking category, but instead of using the Flow label filed, they take advantage of the extension header and store the marking information in either the Hop-by-Hop Option or the Destination Option extension (DOH) headers. To the best of the author's knowledge, there are four previous works on this category that aim to adapt the Deterministic Packet Marking IP-Traceback approach (DPM) [38] on the IPv6 networks. In particular, Obaid et al. [33] uses a 24-byte hop by hop extension header to store the marking data into every outgoing packet. The interface of the router closest to the source of the attack marks the packets. You-ye et al. [96] use the same concept that Obaid et al. [33] applied, but employs the Destination Option Header to store the 24-octet marking data. They also introduce two thresholds terms: L_min' and L_max terms. Only when the load is between L_min and L_max, are the packets marked. For reconstruction, the victim finds the marked packets by looking at the DOH field, and extracts the ingress IP address of the marking router. Animesh et al. [85] divide the 128-bit ingress address into K segments, and marks every packet with the K bits IPv6 fragment, the d bits hash of the IPv6 address and the 2^a bits for the fragment offset. With the suggested number of $k = 16$, $d = 11$ and $a = 2$, the marking data for each packet is 8 octets which should be stored in the Destination Option Header. The reconstruction procedure consists of mark recording and ingress address recovery. The mark recording process indicates which mark fragment arrived at the destination. The address recovery reconstructs the IP address segments and determines which ones are valid. Ashwani et al. [69] have introduced a modified Deterministic Packet Marking approach for IPv6 networks which uses 40-octet hop by hop extension header to store the ingress IP address of the edge router as well as its hash, into every outgoing packet.

Kim et al. [65] propose an IPv6 traceback approach for the next generation network (NGN) which is a combination of the marking and the logging traceback methods. The option field of the Hop by Hop Option header is employed for embedding the marking information. The intermediate routers probabilistically mark the path information in the travelling packets. The hash of the network identification, the router IPv6 address and the source IPv6 address are used for marking. The intermediate routers store the logs of the forwarded packets and the victim reconstructs the attack paths using the logs stored in the routers. All intermediate routers should process the Hop-by-Hop options header. This scheme requires the participation of several intermediate routers. It also increases the computation and the memory overhead of the participating routers.

2.3 DDoS Defence Systems

Several defence mechanisms have been proposed to counter DDoS attacks. They can be divided into two main categories: Prevention defences and Reaction defences [35]. A prevention defence aims to eliminate the possibility of an attack, or to enable the victim to tolerate an attack without blocking legitimate traffic. Prevention defence mechanisms usually require many changes to software and client/server protocols.

Reaction defence mechanisms [61] [98] aim to stop the attacks while they are in progress and to reduce the impact of the attacks on the legitimate traffic. In doing so, they require that the attack be detected as quickly as possible and to respond to it by blocking the malicious traffic. These defence mechanisms need to discriminate between the legitimate and the attack traffic, so that the traffic filtering can be performed selectively. To this end, first they identify the attack source or the attack path and then they enable the filtering to block the malicious traffic coming from the identified source. Traffic classification is an important component of the reaction defence mechanisms, but when anonymized or spoofed IP addresses are employed in the attack packets, it becomes more complicated. Although ingress filtering [54] and route-based packet filtering [97] are two well-known mechanisms against IP spoofing, their performance is based mostly on the distribution over the network. These defence mechanisms require a router to check the source IP address of each packet against a database of legitimate IP addresses. This operation causes a large overhead on the filtering routers and therefore is a challenge to employ in practice. In addition, by spoofing the source IP addresses of the attack packets with the IP addresses of a legitimate host, the attacker can make these mechanisms ineffective.

Another approach for detecting and responding to spoofed attacks is to use IP tunnelling between all the edge routers and several tracing centres [81]. Once a DDoS attack is detected, the edge router forwards the victim traffic to the tracking centre. In doing so, the point of the attack traffic entry to the Internet can be detected. Besides the considerable overhead of IP tunnelling on the edge routers, the tracking centre processes a large amount of traffic during the attack.

There are several attempts to integrate the traceback mechanisms with the real-time DDoS defence systems to react to the attacks while they are in progress [93] [83] [80] [51] [29] [27] [90] [53]. Sung and Xu [83] have suggested a mechanism in which the IP-Traceback and the packet filtering are integrated and work together. Thus, the victim's resources are preserved, even before the traceback is completed. In this approach, two types of marks are employed: the signaling mark for the traceback and the data mark for filtering. The victim employs a traceback method to identify the attack path. While the traceback algorithm is in progress, the victim identifies the data marks of suspicious packets and selects the filtering probability of the marked packets. This decision is based on the rate of the incoming packets and the attack path. This algorithm is performed regularly and the results sent to the filtering router that is located near the victim. There are several challenges related to this approach. First of all, filtering is limited to the packets with the data mark, so the data marking rate has to be adjusted by the attack intensity. However, such a feature does not exist in the current traceback approaches. Secondly, the victim sends the identified attack path to all of the filtering routers. To avoid the extra communication overhead between the victim and the filtering router, the number of filtering routers and consequently, the distance between the defence systems and the victim has to be low (in terms of the number of hops). However the proximity of the defence systems and the victim may cause the filtering routers to be overwhelmed by an intense attack in which the attackers are highly distributed over the network and can generate a high volume of attack traffic. More importantly, if the defence system is located near the victim, when a legitimate packet reaches the filtering router, it most likely carries the marking information of an infected router, and therefore there is a low probability of it escaping the filtering.

Chen and Song [80] have changed the aforementioned defence system [83] to rely only on the edge router near the victim for both IP-Traceback and packet filtering purposes. This defence scheme is interesting but if the attack is distributed widely over the Internet, this approach cannot differentiate between the legitimate and the attack traffic appropriately. A good example of this kind of attack is the Code Red Worm which orchestrated 300,000 bots to initiate a DDoS attack against the White House website in 2001 [51].

Yaar et al. [29] have proposed the Pi approach which is a deterministic packet marking (DPM) mechanism and embeds the path fingerprint into every packet. This defence approach does not work in a situation in which the purpose of the attack is to consume the victim's bandwidth, because it relies only on the DDoS packet filtering at the victim's side. To address this problem, they proposed a capability-based defence system [27]. However, their approach requires fundamental and essential modifications to current network protocols. This prevents their approach to be employed easily over the Internet. Chen et al. [90] have proposed another approach to DDoS attack detection and filtering based on packet marking which is very similar to the Pi approach [29]. Instead of tracking the attacking packets, they have proposed creating a table of legitimate paths and eliminating those packets which come from the other paths. Since the filtering operation is performed at the victim's side, this approach suffers from the same problems as the Pi approach.

2.4 Summary

There is no elixir in IP traceback. Different tracing schemes make different assumptions and attempt to solve different problems. In general, each tracing scheme has to make some trade-offs between performance and overhead. For example, in the marking schemes, factors which need to be taken into account include marking every packet or marking at a certain probability, the number of bits used for marking, the place to store the marking information, as well as those parts of the networks (the routers, the victim, or both) which bear the incurred overheads. In the logging schemes, the issues to be addressed include the content to be logged, the frequency of logging, the place in which to store the logging information, and an efficient approach for communicating between the victim and the routers where the logging information is stored. In deterministic approaches, some features which need to be considered are the traceback rate and the degree of involvement of the Internet Service Providers (ISPs). In the probabilistic methods, the important factors are the complexity of constructing the attack path, the number of concurrent DDoS attackers that the system can handle, and the number of required marked packets to construct the attack path.

Considering all the strengths and weaknesses of the state of the art approaches in the literature, there are some gaps among all the groups of IP-Traceback schemes which have not been considered yet. For example, they all operate at the packet level. If they could work at the flow level, once a victim finds the origin of a packet that belongs to a flow, then the origin of any other packet in that flow discovered as well. Using this concept may result in lower marking, processing, bandwidth and memory overhead in the flow-based IP-Traceback approaches compared to the packet-based IP-Traceback approaches. In addition, the approaches introduced in the literature are able to traceback only up to the AS level or at best, up to the edge router of the attacker network, not to the attacker node.

This thesis presents a modular security framework for addressing the above challenges, and exploring how far one can push a defence/security system to mitigate the impact of the IP spoofing attacks. This framework employs the strongest features of the previous deterministic and probabilistic marking methods, as well as the router-based and the AS-based approaches, either for IPv4 or IPv6 networks, and addresses the gaps which are not considered in the previous approaches.

Chapter 3

Two Principal IP-Traceback Approaches, an Empirical Evaluation

This chapter presents a brief survey of two promising schemes for tracing cyber-attacks: the well-known Probabilistic Packet Marking (PPM) and the Deterministic Packet Marking (DPM) approaches. Most of the current IP-Tracebacks are variants of these two approaches. PPM and DPM are explored in detail and the advantages and disadvantages of both approaches are analyzed in depth in terms of practicality and feasibility so that the shortcomings of each scheme are highlighted.

3.1 Probabilistic Packet Marking, PPM

In this section, the Probabilistic Packet Marking method called PPM is described. Based on the various IP-Traceback approaches described in Chapter 2, PPM falls into the following categories: Basic principle–Marking; Processing modes–Probabilistic; Location–Network group.

This approach is based on the idea that all routers in the attack path select the packets that pass through them randomly, with a constant probability, and then mark the selected packets with their own IP address (i.e. write a portion of their own IP address in the packet IP header). Once the victim gets a large amount of marked packets the attack path can reconstructed, even if the IP addresses of the packets have been spoofed. This approach, Figure 3.1, had been introduced by Burch and Cheswick [40] and later improved by Savage et al. [50]. Assume that there are d routers in an attack path and the marking probability of each of these routers is a constant number p. The optimal value for p is $1/d$. However from the viewpoint of the victim, the marking probability of router R^i $(1 => i <= d)$ is $p(1-p)^{d-i}$ which is different from P [91], [70]. This happens because subsequent routers may override (re-mark) the packets which have been marked by the previous routers. In other words, the routers which are further away from the victim are more likely to be overridden by subsequent routers. Thus, the closest router to the victim has the best chance of delivering its marks in the attack path.

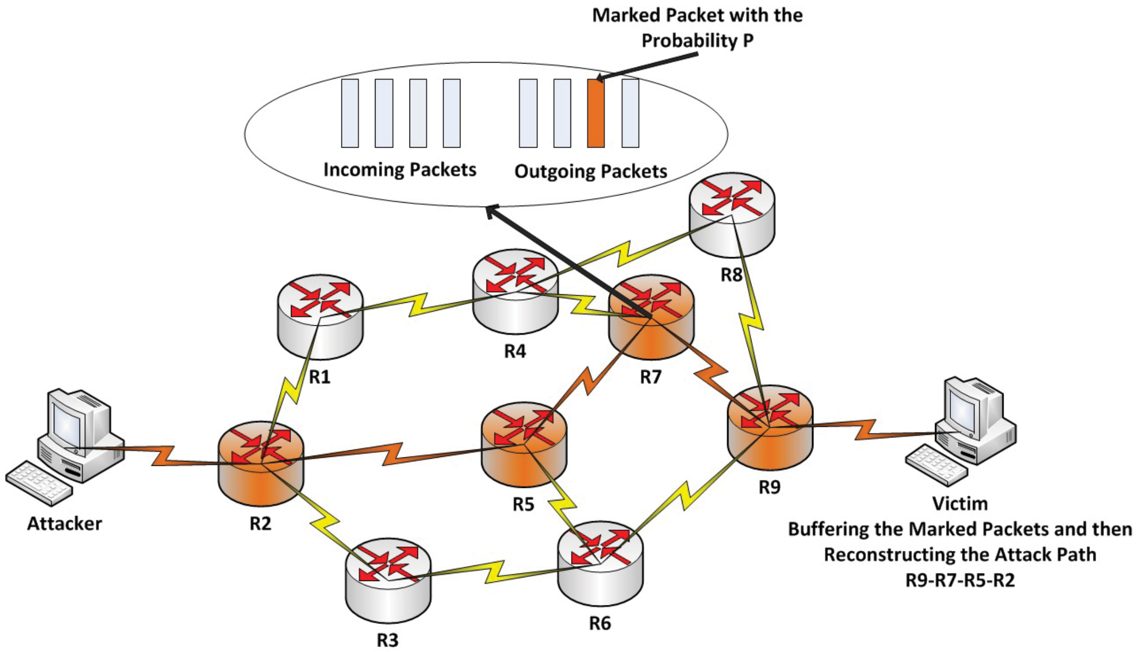

Marked Packet with the Probability P
Incoming Packets
Outgoing Packets
R1
R2
R3
R4
R5
R6
R7
R8
R9
Attacker
Victim
Buffering the Marked Packets and then Reconstructing the Attack Path R9-R7-R5-R2

Figure 3.1: A Schematic Illustration of the PPM Approach - All Routers in the Attack Path Take Part in the Traceback.

3.1.1 PPM Scheme

In this approach, there are two fields, *addr* and *dist*, which play the main role in the packet marking. Both of these fields have been embedded in the $identification$ field of packet IP header. Once a router decides to mark a packet (i.e. this decision is independent of other routers), it writes its own IP address to the *addr* field and zero in the *dist* field. In other words, if the router gets a packet with a *dist* field of zero, it indicates that this packet has been marked by the previous router. In this case, the router would XOR its own IP address with the *addr* field of the marked packet and would override the result into the *addr* field again. Finally, if the router does not mark the packet, it always adds one to the *dist* field.

Figure 3.2 shows the marking process in an attack path with 3 routers $R1$, $R2$, and $R3$. In the $R1$ router, there are 2 cases. The box on the left shows the case in which the $R1$ marks the packets (i.e. it writes its own IP address in *addr* field and sets the *dist* field to zero), and the box on the right shows an unmarked case by the

$R1$ router (i.e. it just adds one to the *dist* field). In the $R2$ router, there are 4 cases, two boxes on the left show a situation in which the $R1$ router has previously marked the packets. Of these two boxes, the box on the left shows the case in which the $R2$ router marks the packet again (i.e. it writes its own IP address in the *addr* field and sets the *dist* field to zero), and the box on the right shows the case in which the $R2$ router does not mark the packet (i.e. it XORs its own IP address with the *addr* field, overwrites the result into the *addr* field, and adds one to the *dist* field). On the other hand, two boxes on the right show a situation in which the $R1$ router has not previously marked the packets. Of these two boxes, the box on the left shows the case in which the $R2$ router marks the packet (i.e. it writes its own IP address in the *addr* field and sets the *dist* field to zero), and the box on the right shows the case in which $R2$ router does not mark the packet (i.e. it just adds one to the *dist* field). Using the same process for the $R3$ router, the victim would get eight results. However, some of these eight marking results are similar. For example, the boxes numbered 1, 3, 5, and 7 have the same marking information. By eliminating the duplicate results, four non-repetitive cases remain for the victim.

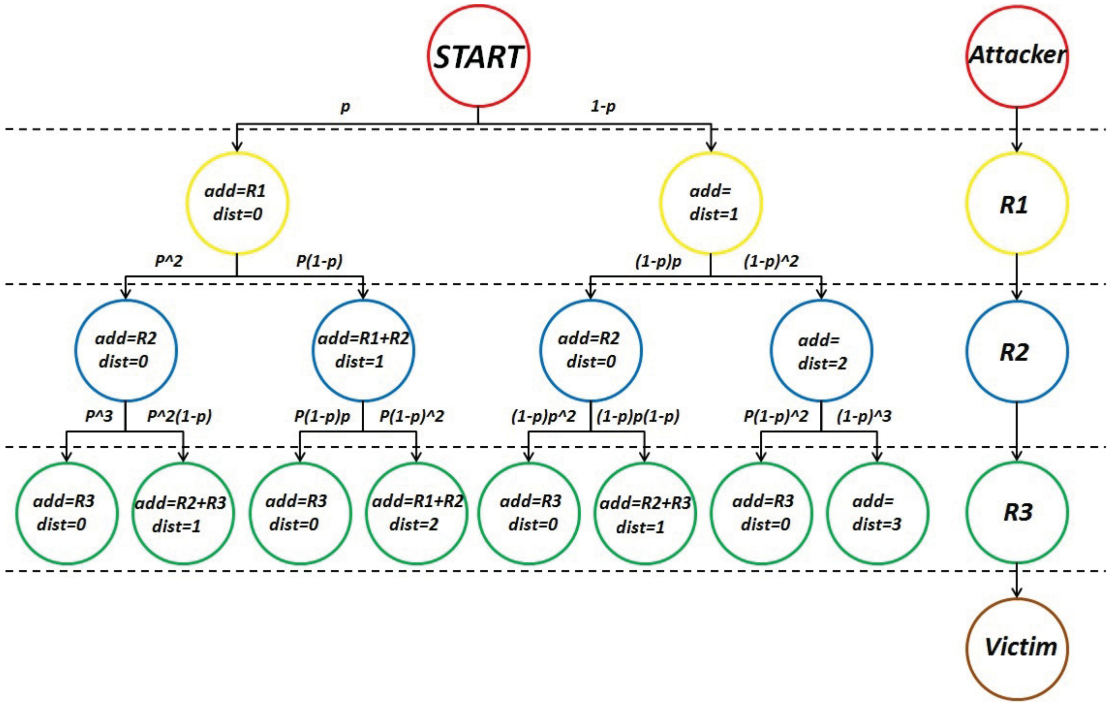


Figure 3.2: The PPM Marking Process - The Marking Process in an Attack Path with Three Routers $R1$, $R2$, and $R3$

For path reconstruction the victim first locates the closest router to itself (i.e. $R3$ in Figure 3.2) by looking for the packet in which the *dist* field is equal to zero. Secondly, because $R3 \oplus (R2 \oplus R3) = R2$, the victim can locate the $R2$ router by looking for the packet in which the *dist* field is equal to one, and XOR its *addr* field with the IP address of the $R3$ router. The victim continues this process until locating the router which is the most far away.

3.1.2 PPM Analysis

Although PPM has several good advantages such as zero bandwidth overhead (i.e. all marking information is stored in the packet IP header), PPM has serious weaknesses in the face of DDoS attacks, as discussed below.

Computational overhead

For each packet, there is a computational overhead to decide whether the packet should be marked or not. In addition, if the packet is marked, there is further computational overhead involved with preparing the marking information and upgrading the *addr* and the *dist* fields. However in comparison with the computational overhead on the victim's side for path reconstruction, the computational overhead of the routers in the attack path is negligible. [79] shows that when there are 25 concurrent attacks on a victim, path reconstruction may take several days with thousands of false positives, while the current DDoS attackers may orchestrate thousands of attack zombies at the same time. In this situation, the victim may never reconstruct the attack path.

Memory overhead

Memory overhead on routers is highly undesirable, because it reduces network performance and requires hardware upgrades. Since the marking process on routers does not store anything, the router's memory overhead in PPM algorithms is negligible. However, on the victim's side, a large memory structure for the attack path reconstruction process is required. The victim could be forced to store millions of records in the data structure, and then search on it, to reconstruct the attack path. However, memory overhead on the victim machine is more tolerable than on the routers.

False Positives

PPM has a high false positive rate in the face of DDoS attacks. This problem originates from the basis of the reconstruction algorithm. In this case, the victim should perform two processes: first of all, it should obtain the IP addresses of all routers in the attack path and secondly, using the routers' IP addresses, reconstructs the attack path. In PPM, 8 packets marked by the same router need to be identified and combined to resume the IP address of that router [58]. Since there is no sign other than the *dist* field, in a situation in which there are several attacks paths, it is difficult for the victim to identify which marked packets belong to which router because there are several routers the same distance from the victim. This issue may prevent the attack path reconstruction process.

Mark Spoofing by attackers

If the attacker is aware of an existing PPM marking process in the network, he/she may send fake marked packets to the victim. In this situation, the victim may not be able to reconstruct the attack path correctly because the victim cannot differentiate between the fake and the genuine marked packets.

Mark spoofing by subverted routers

There are two kinds of malfunctioning routers which may disrupt the traceback operation by the victim. First of all, the incorrectly-configured routers which participate in the PPM packet marking may confuse the path reconstruction process. Secondly, the compromised routers can prepare and send fake marked packets which will most likely prevent the victims traceback to the attack source.

Awareness of the attack path length in advance

As described above, the optimal value for the marking probability, p, is $1/d$. However once a router decides to mark a packet, it does not have any idea about its path length, d, so it cannot set the p to the optimal value. [50] suggests using the constant number 0.04 for p. However if the victim is under several attack with different attack path lengths, using a predetermined constant number for p strongly reduces the efficiency of the path reconstruction process.

Awareness of the network map and the routing in advance

The PPM algorithm works based on an assumption that the victim should be aware of the network map and the routing in advance to be able to reconstruct the attack path using the IP addresses of the routers along the path extracted from the received marked packets. So the concern in this case is how to keep the victim updated about the network map and routing; otherwise whenever a new router is added to the network, the path reconstruction process will not work correctly.

The number of required packets for traceback

In the first implementation of the PPM algorithm, the victim required thousands of packets to reconstruct an attack path [50]. Later this was improved to less than 1000 packets by [79]. However, the number is still high and therefore is a serious drawback of the PPM algorithm.

Fragmentation

PPM uses the ID field in the IP header of packets to embed marking information, which is generally used for fragmentation. If only a single packet of a fragmented datagram is marked, then the datagram reassembly will fail.

ISP Involvement

The path reconstruction process needs to get marked packets from all routers along the attack path. To this end, the marking process should be activated on all routers in the network. However, what the Internet Service Providers (ISPs) need to do is limited to updating the router's IOS and enabling PPM on the routers. Having said this, ISPs should do this on all routers, including either edge or backbone routers. Indeed, the involvement of all of the routers is a major problem in using this method. Given that, in practice it may cross the boundaries of both ISPs and countries.

3.1.3 Discussion

So far, several variations of PPM have been proposed [50][91][28]. For example, to counter the "awareness of the attack path length in advance" problem, there have been some attempts [58] [67] [62] to set d as the number of Autonomous Systems (AS), rather than the number of routers, between the current network and the victim. However this solution cannot reconstruct the attack path accurately, which is the main goal of the PPM approach.

Song et al. [79] have proposed an advanced and authenticated marking scheme for IP-Traceback. Their approach decreases the high computational overhead and false positive rate, as the number of the required marked packets for path reconstruction is less than one thousand in their approach, and it partially covers the mark spoofing problem. However, their approach cannot resist against the compromised routers in the attack path, since a compromised router may be reconfigured to mark the packets incorrectly and it is still authenticated by the victim. Unfortunately, there is still no approach to cover the problem of malfunctioning routers. Note that the computational overhead and the false positive rate are in direct proportion: as the high computational cost increases so does the false positive rate.

One possible solution to counter the fragmentation problem is to mark the fragmented packets with a lower probability. Therefore, the fragmented datagrams have more of a chance to survive. However, this approach will definitely increase the number of required marked packets for path reconstruction.

As described earlier, as the number of hops between a router in the attack path and the victim increases, the mark information of that specific router is less likely to survive, so from the perspective of the victim, the farthermost router has the lowest chance of delivering its mark in the attack path. One solution to this problem is to use variable marking probability for each router, based on the distance between the current router and the victim [43]. However, the hard part is how to find the number of hops between the two ends.

Based on the above PPM specifications, PPM is a good solution for small scale DoS attacks. However, most of the current attacks are large scale DDoS attacks with hundreds or thousands of concurrent attack sources. In these cases, PPM is not a good choice. Therefore, researchers have proposed another approach, called DPM

[36], to overcome some of the problems of the PPM approach.

3.2 Deterministic Packet Marking, 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 on the victim machines. Based on the various IP-Traceback approaches described in Chapter 2, DFM falls into the following categories: Basic principle–Marking; Processing modes–Deterministic at packet level; Location–Source group.

3.2.1 DPM Scheme

The main goal of DPM, which was first proposed by Belenky and Ansari [36] and later was improved by [38], was to focus on an obvious issue of PPM. That issue was that each packet in a datagram network is being routed individually, so even if the sources and the destinations of the packets are the same, they may be routed along different paths. This feature of packet networks may prevent the attack path reconstruction by the victim, using the PPM algorithm. Since each packet may travel a different route from the same source to the same destination, the only address in the network path that is surely the same for all packets is the ingress interface IP address of the closest router to the source of packets. The main idea behind the DPM approach is that the ingress interface IP address of the closest router to the source of the packet is enough to find the attacker network. It should be noted here that in the current Internet network, packet routing is mostly stable. However, there is still this potential to route the packets from different paths.

As shown in Figure 3.3, 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 the 16-bits $Identification$ field and the 1-bit *reserved* flag, to embed the marking information to every packet. The 32-bit 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 to 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. The 1-bit *reserved* flag plays the role of a sign for the victim to identify which part of the IP address is carried by the current packet.

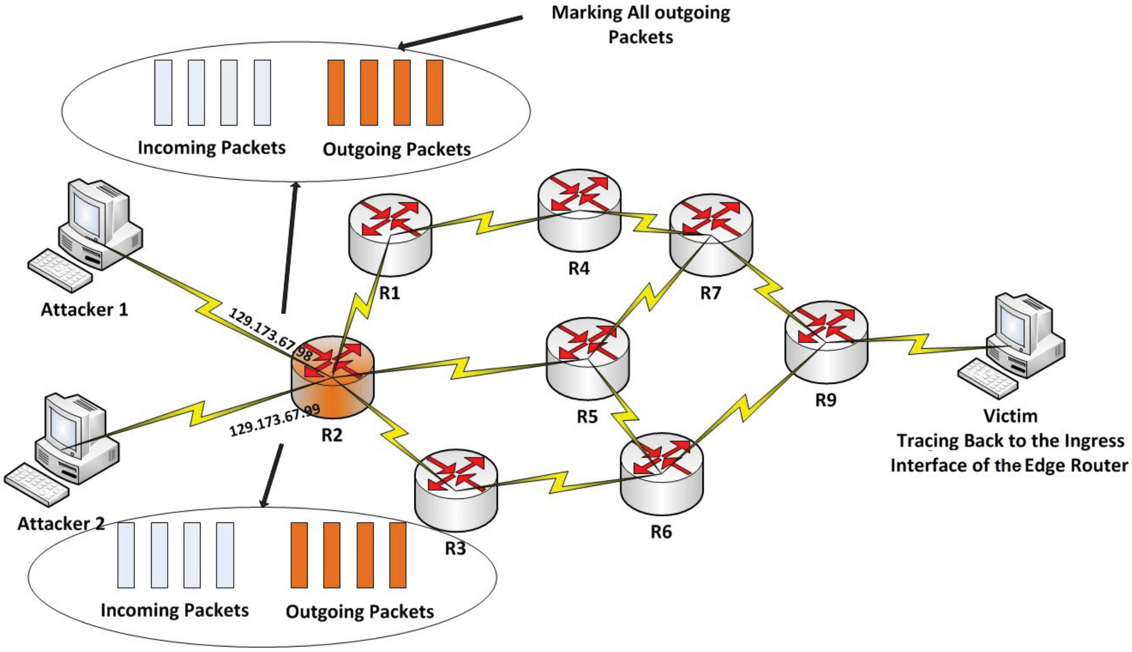


Figure 3.3: A Schematic Illustration of DPM Approach - Only the Edge Router Takes Part in Traceback.

DPM has two key features: firstly, 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 . However, it is a well known fact that the source IP addresses may be spoofed. Under such attacks, there are at least two cases in which the edge router IP address reconstruction may not be effective: firstly, when two or more hosts that have the same source IP address attack the victim and secondly, when (D)DoS attackers simply change the source address field for every packet they send. In these cases, basic DPM is unable to reconstruct any valid ingress addresses [36]. 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 [38]. 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 three parts: a segment of ingress address a, the index of segment d, and a digest of ingress address k. They claimed that the best trade-off for the size of each of these parameters happens when $a = 4$, $d = 3$, and $k = 10$, all together 17 bits.

3.2.2 DPM Analysis

For analyzing DPM, the same evaluation metrics were employed which were used to analyze PPM as summarized below.

Computational Overhead

The CPU overhead of DPM is lower than the PPM approach. Unlike PPM, in DPM only the edge router closest to the attacker is responsible for marking (not all the routers along the attack path). Moreover, since DPM marks every packet, there is no need for a 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 the 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.

Memory Overhead

As with the PPM approach, the memory overhead on the routers is negligible. However unlike PPM where the victim requires a large memory structure to store millions of records, in the DPM approach, the victim keeps a smaller Reconstruction-Table. It is because in PPM, the victim needs almost 1000 packets to reconstruct an attack path, while in DPM, only $32/a$ packets are required to reconstruct the ingress address (i.e. with the suggested $a = 4$ [38], DPM requires only 8 packets to traceback to the ingress interface address of the edge router close to the attacker).

False Positives

As discussed earlier, the basic DPM method has a significant limitation when dealing 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 of using a hash function to produce hash values of the ingress interface, called the single-digest DPM technique, or to use a family of hash functions to produce multiple digests of an ingress address, called the multiple-digest DPM technique. In these techniques, hash values are sent along with marked bits to effectively prevent the combinatorial explosion problem. This modification to DPM guarantees that the false positive rate will not 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 is 45, and in the multiple-digest DPM technique is 2296 [38].

Mark spoofing by attackers

In the 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. This automatically obviates the issue of mark spoofing which PPM has to account for.

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 increases the issue of mark spoofing by subverted routers in the attack path. Thus, in an untrusted network such as the 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.

Awareness of the attack path length in advance

As the process of ingress interface IP address reconstruction does not need the path length, awareness of the attack path length is not an issue in DPM.

Awareness of the network map and the routing in advance

As the goal of DPM is not reconstructing the attack path, instead it reconstructs the ingress interface IP address of the edge router, so awareness of the network map and the routing is not an issue in DPM.

The number of required packets for traceback

As discussed earlier, $32/a$ packets are required to reconstruct the ingress address. With the suggested $a = 4$ [38], 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 the ingress address field.

Fragmentation

Like PPM, DPM uses the ID field in the IP header of the packets as well as the 1-bit reserved flag to embed the marking information. If only a single packet of a fragmented datagram is marked, then the datagram reassembly will fail.

ISP Involvement

In DPM, the involvement of the ISPs is very limited. Only the edge routers have to be upgraded to support the function of the deterministic packet marking. Unlike PPM, 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.

3.3 Summary

In this chapter, a survey of two schemes for IP-Traceback, the well-known Probabilistic Packet Marking, PPM, and the Deterministic Packet Marking, DPM, approaches have been presented. In summary, DPM mitigates some of the problems of PPM. These are: its CPU and Memory burden are far less, it improves the false positive rate, limits sending the mark-spoofed packets by the attackers, does not require an awareness of the attack path length, the network map and the routing in advance, decreases the number of the required packets for the traceback from almost 1000 packets to 8 packets, and the involvement of the ISPs is limited only to the edge routers. However DPM still has several problems, including those listed below.

- To keep the false positive rate below 1%, DPM cannot scale under heavy DDoS attacks as discussed above [38].

- DPM is able to traceback up to the ingress interface of the edge router close to the attacker, but not the exact attacker node.

- Although DPM has higher traceback accuracy in comparison to the 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.

The aforementioned four problems were the motivation for proposing the flow-based IP-Traceback approaches that will be described in the next chapters.

Chapter 4

Deterministic Flow Marking: DFM

This chapter describes the first proposed IP-Traceback method called Deterministic Flow Marking, DFM. Based on various IP-Traceback approaches described in Chapter 2, DFM falls into the following categories: Basic principle–Marking; Processing modes–Deterministic at flow level; Location–Source group.

4.1 Assumptions

1. An attacker may generate any packet,
2. Attackers may be aware they are being traced,
3. Attacker may spoof the source MAC and IP addresses,
4. Packets may be lost or reordered,
5. An attack may consist of just a few packets,
6. Packets of an attack may take different routes,
7. Routers are both CPU and memory limited,
8. The edge routers have been MAC filtering enabled.

4.2 DFM Modules

The DFM algorithm consists of two modules: one is a DFM encoding module (DFME), which runs at the attacker-end edge router, and the other one is a DFM decoding module (DFMD), which runs at the victim-end. The following sections describe these two modules in detail.

4.2.1 DFM Encoding Module (DFME)

The DFM encoding module (DFME) marks every flow, instead of every packet, to take advantage of both the high traceback accuracy and the low packet marking rate. DFME uses three identifiers to mark a flow in order to trace up to the attacker node. These three identifiers are described below.

1) The IP address of the egress interface of the edge router (32 bits): The edge router is the closest router to the attacker node with at least one valid assigned IP address to its egress interface. Some previous researches on IP-Traceback such as [38] and [75] have proposed the use of the ingress interface IP address of the first router in the attack path as an identifier for traceback. However since DFM should be able to trace up to the attacker node even if the attacker is behind a NAT, the ingress interface IP address will be useless in this case because most of the time the ingress interface IP address of a NAT device has an invalid IP address which is untracable.

2) Network Interface Identification (NI-ID) (12 bits): This is an identifier assigned to each interface of either the MAC address of a network interface on the edge router, or the virtual local area network (VLAN) ID of a virtual interface if the edge router uses VLAN interfaces. The NI-ID specifies which subnet a traffic flow comes from. Currently, a router can equip 802.1Q VLAN interfaces, which means that multiple virtual interfaces can be used on one physical network interface. Although VLAN interfaces, which share one physical interface, have the same MAC address, each VLAN interface has a unique VLAN ID, so that a router distinguishes VLAN interfaces by their VLAN IDs. If a network interface is a VLAN interface, an NI-ID is assigned to its VLAN ID instead of the MAC address shared with another VLAN interface. A 12-bit NI-ID, expressed in the range from 0 to 4095, is sufficient to represent all possible network interfaces and VLANs on an edge router. An edge router keeps an NI-ID table and numbers the interfaces from 0 to 4095. Each NI-ID table entry consists of an NI-ID and the MAC address of a network interface card. If VLAN interfaces are used, then the entry in the NI-ID table consists of an NI-ID and a VLAN ID. Table 4.1 shows the NI-ID table of an edge router which uses VLAN interfaces on a physical interface whose MAC address is “C”.

3) Node-ID (16 bits): An identifier assigned to each source MAC address observed from incoming traffic from the local networks. Each MAC has a unique Node-ID.

NI-ID	MAC address of the connected interface of the edge router to the local network	VLAN ID
1	A	
2	B	
3	C	101
4	C	102

Table 4.1: An Example of an NI-ID Table

Representing the Node-ID with 16 bits seems to be sufficient as it makes it possible to address all nodes on a LAN connected to each interface of the edge router, even if each LAN is as big as a class B network (the maximum number of nodes in a class B network is $2^{16} - 2$). An edge router keeps multiple Node-ID tables, a table for each NI-ID, and numbers the source MAC addresses of the observed incoming traffic from 0 to 65535. An entry in the Node-ID table is composed of an NI-ID, source MAC addresses of the observed incoming traffic and a Node-ID. Table 4.2 shows two examples of a Node-ID table.

NI-ID	Node-ID	Source MAC Addresses of incoming packets
1	1	F
	2	E
	3	H
	4	K

NI-ID	Node-ID	Source MAC Addresses of incoming packets
2	1	B
	2	S
	3	Q

Table 4.2: Two Examples of a Node-ID Table for NI-ID 1 and 2

Marking each flow with a combination of the IP address of the egress interface [(32 bits) + NI-ID (12 bits) + Node-ID (16 bits) = 60 bits identification data], distinguishes the traffic of a particular node from the other nodes.

The definition of a flow is accepted as a unidirectional sequence of packets between two endpoints that have a Flow-ID in common with no more than a specific inter packet delay time. Flow-ID is the five information tuples including the source IP address, the destination IP address, the L4 protocol type (TCP/UDP), the source port number and the destination port number. While this definition is able to define Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) flows, it is unable to define Internet Control Message Protocol (ICMP) flows, as ICMP does not use a port number to establish a session. For a good traceback method, it is very important to trace ICMP flows as well because some (D)DoS attacks employ ICMP flooding attacks. To this end an ICMP flow is defined as a unidirectional sequence of ICMP packets between two networks which have six tuples: source IP address, destination IP address, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID in common with no more than a specific inter-packet delay time. As used in [73], 600 milliseconds is set for the inter-packet time delay to terminate a flow.

Algorithm 1 describes the marking procedure of the proposed traceback method. 60 bits of identification data needs to be passed to the destination for each flow. The identification data is divided into K fragments. Therefore, the mark contains $M = 60/K$ bits of the identification data and $S = log_2 K$ bits required to identify a fragment. Also, advantage is taken of the one flag bit to identify marked and unmarked packets in a flow. The experimental results section describes how to store the mark bits in some IP header fields which are used rarely.

Figure 4.1 depicts the choice for partitioning the 60 bits in the first K packets of each flow. The first K packets of every flow carry the mark fragments including M bits for identification data fragment, S offset bits to represent 2^S possible fragments and one flag bit which should be set to “1” for the marked packets and “0” for the rest.

4.2.2 DFM Decoding Module (DFMD)

Algorithm 2 describes the mark decoding procedure. The DFMD module at the destination maintains a table matching the Flow-ID and K possible mark fragments. See the ”Reconstruction-Table”, Table 4.3. As described above, the Flow-ID of TCP and UDP flows is defined by five tuples while the Flow-ID of ICMP flows is defined by six tuples. When a packet belonging to an unseen flow arrives at the destination, the DFMD module creates a new table entry in the Reconstruction-Table. Then, it extracts the marking bits of this flow from the marked packets, identified by the one bit F-flag, and writes them in the corresponding fields, Table 4.3. 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

Algorithm 1 Mark Encoding Procedure

```
if $Packet.protocol = tcp$ or $udp$ then
  FlowID $\Longleftarrow$ 5 tuples
else
  if $Packet.protocol = icmp$ then
    FlowID $\Longleftarrow$ 6 tuples
    Mark $\Longleftarrow$ Packet.EdgeIP + Packet.NI-ID + Packet.Host-ID
  end if
end if
if $Flow[FlowID].Packet\# < K$ then
  switch Flow[FlowID].Packet#
    case 1:
      Packet.MarkField $\Longleftarrow$ 1th Frag
      Packet.Frag# $\Longleftarrow$ 0
    case 2:
      Packet.MarkField $\Longleftarrow$ 2th Frag
      Packet.Frag# $\Longleftarrow$ 1
      .
      .
      .
    case $K$:
      Packet.MarkField $\Longleftarrow$ $K$th Frag
      Packet.Frag# $\Longleftarrow$ $K - 1$
  end switch
  Packet.Flag $\Longleftarrow$ 1
else
  Packet.Flag $\Longleftarrow$ 0
end if
FixCheckSum
SendPacket
```

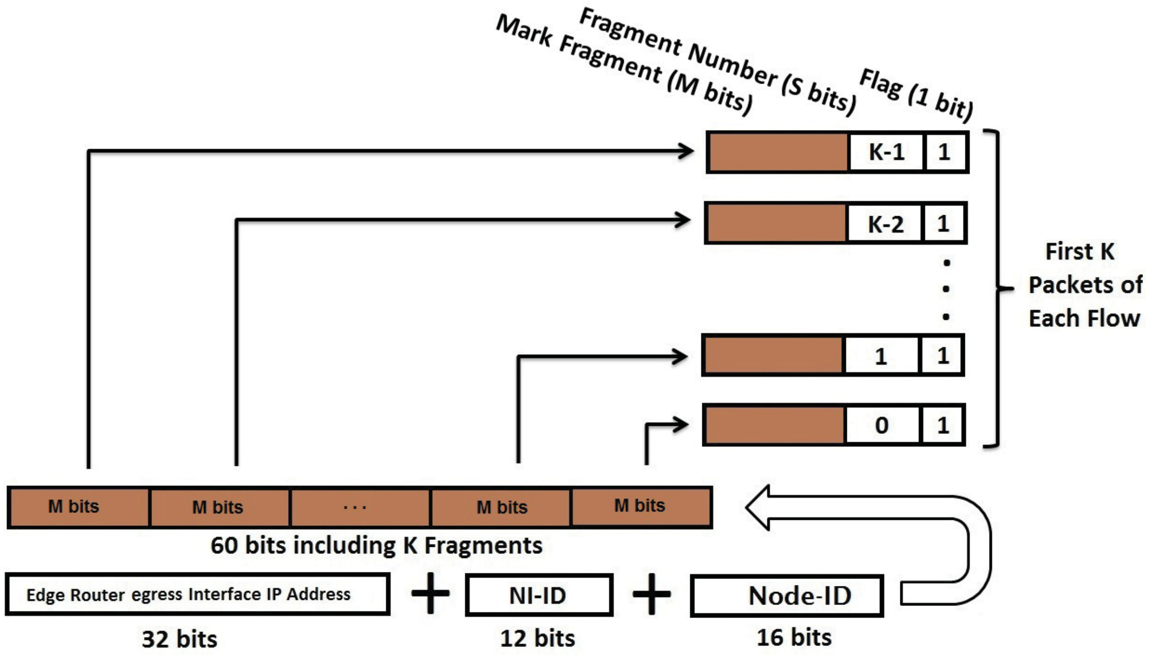


Figure 4.1: Partitioning Identification Data in the First K Packets of Each Flow

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.

4.3 Discussion

As discussed earlier, some previous traceback research uses hash functions and sends the hash values along with the marked packets to counter the concurrent attacks. In particular, Belenky and Ansari [38] propose to insert the ingress interface IP address into outgoing packets by the first router of the attack path. Since the 32 bits that compose an IP address do not fit into the available marking space, they propose to split the router IP address into K fragments. After receiving all K address fragments, the victim can recover the address by reassembling the received fragments. Their basic DPM method has limitations when dealing with multiple attackers at the same time with the same source IP address. To counter this problem, they propose producing hash values of the ingress interface and sending this hash value along with the marked bits.

Under the same conditions, the proposed DFM method does not need to use hash functions, because DFM first detects the flow to construct a valid mark. Thus, DFM

Algorithm 2 Mark Decoding Procedure

```
if $Packet.protocol = tcp$ or $udp$ then
    FlowID $\Longleftarrow$ 5 tuples
else
    if $Packet.protocol = icmp$ then
        FlowID $\Longleftarrow$ 6 tuples
    end if
end if
if $Packet.flag = 1$ then
    switch Packet.flag#
        case 0:
            Frag1 $\Longleftarrow$ 1th Frag
        case 1:
            Frag2 $\Longleftarrow$ 2th Frag
            .
            .
            .
        case $K - 1$:
            Frag$K$ $\Longleftarrow$ $K$th Frag
    end switch
end if
if $Flow[FlowID].\#Frags = K$ then
    extract EdgeIP
    extract NI-ID
    extract Host-ID
end if
```

Flow-ID	First Fragment	Second Fragment	Identification

AC100C14055A5585B02A005006
Srcip = 172.16.112.20
SrcPort = 1370
Dstip = 85.133.176.42
DstPort = 80
Protocol = tcp
	AC100001	4004005F	Edge router IP =
			172.16.0.1
			NI-ID = 1
			Host-ID = 23

AC10715410925585B817001511
Srcip = 172.16.113.84
SrcPort = 4242
Dstip = 85.133.184.23
DstPort = 21
Protocol = udp
	AC100001	40080037	Edge router IP =
			172.16.0.1
			NI-ID = 2
			Host-ID = 13

Table 4.3: An Example of the Reconstruction-Table with K=2, M=30 and S=1

is safer when countering the problem of multiple attackers with the same IP addresses at the same time.

Furthermore, in the proposed system, it is assumed that the attacker can change its MAC address. Consider the following four potential scenarios when an attacker changes his MAC address.

1. The attacker spoofs his MAC address with a random MAC address: in this case, it is assumed that MAC filtering is enabled in the edge router and the attacker cannot access the network.

2. The attacker has access to the white list of MAC addresses and he spoofs his MAC address with an active MAC address: the current switches and routers reject concurrent access of more than one node with the same MAC address, so the attacker cannot access the network.

3. The attacker has access to the white list of MAC addresses and he spoofs his MAC with an inactive MAC address: in this situation, after the attack is detected by the victim, he/she can block the attacker node, using DFM to distinguish the attacker traffic from the rest of traffic, while the other nodes will still have access to the destination (victim).

4. The attacker spoofs his MAC address with several existing MAC addresses in the

white list regularly: after detecting the attack using DFM, the victim assumes that several source nodes from the same network belonging to one interface of the edge router are try to send malicious traffic. At this point, the victim traces up to level two of the traceback (the edge router interface) and only cuts off the access of all nodes belonging to this interface, not all the nodes which are connected to this edge router. It should be noted here that the other traceback methods in the face of such a situation are only able to trace up to the edge router and therefore, they would cut off the access of all the nodes forwarded by the edge router.

As discussed in the first three MAC address changing scenarios, DFM is able to trace three levels up to the attacker node. Only in scenario 4, does DFM trace two levels up to the source network interface of the edge router. However this is still much better than the current traceback methods, where they at the best can detect up to the source edge router.

It should be noted here that DFM is able to traceback to the source of the traffic one step behind the ingress interface of the edge router. Every router with a valid IP address on its egress interface can potentially act as an edge router. So if a valid IP address is assigned to the egress interface of the closest router to the local network, then DFM would be able to traceback up to the source node. However, if the network administrator defines the farthest connected router to the Internet as the edge router, then it is likely that there are some subnets behind that edge router. In this case, DFM is able to traceback up to the sub-networks and therefore, fewer routers are required to participate in the DFM marking scheme. This is a trade-off between the accuracy and the number of participant edge routers in the DFM marking scheme.

4.4 Experimental Results

As described earlier, based on the various IP-Traceback approaches described in Chapter 2, DFM and DPM fall into the same categories of classification, but PPM falls into the other categories. In addition, the goal of PPM is entirely different from that DPM and DFM. The purpose of PPM is to identify the attack path, while the goal of DPM and DFM is to identify the attack source. Consequently, the performance of PPM cannot be compared directly with those of DPM and DFM under the same conditions on the same network: only the performances of DFM and DPM were compared under the same conditions and on the same network platform [31].

Figure 4.2 is a schematic illustration of both DPM and DFM approaches and is a comparison of the two methods. To evaluate DFM and compare the result with DPM, both approaches have been employed on six discrete network traces including the MAWI (Measurement and Analysis on the WIDE Internet) traffic archive December 2012 [8], the CAIDA DDoS attack 2007 [17], the CAIDA anonymized Internet traces October 2012, December 2012 [14], March 2014 and July 2014 [15] data sets.

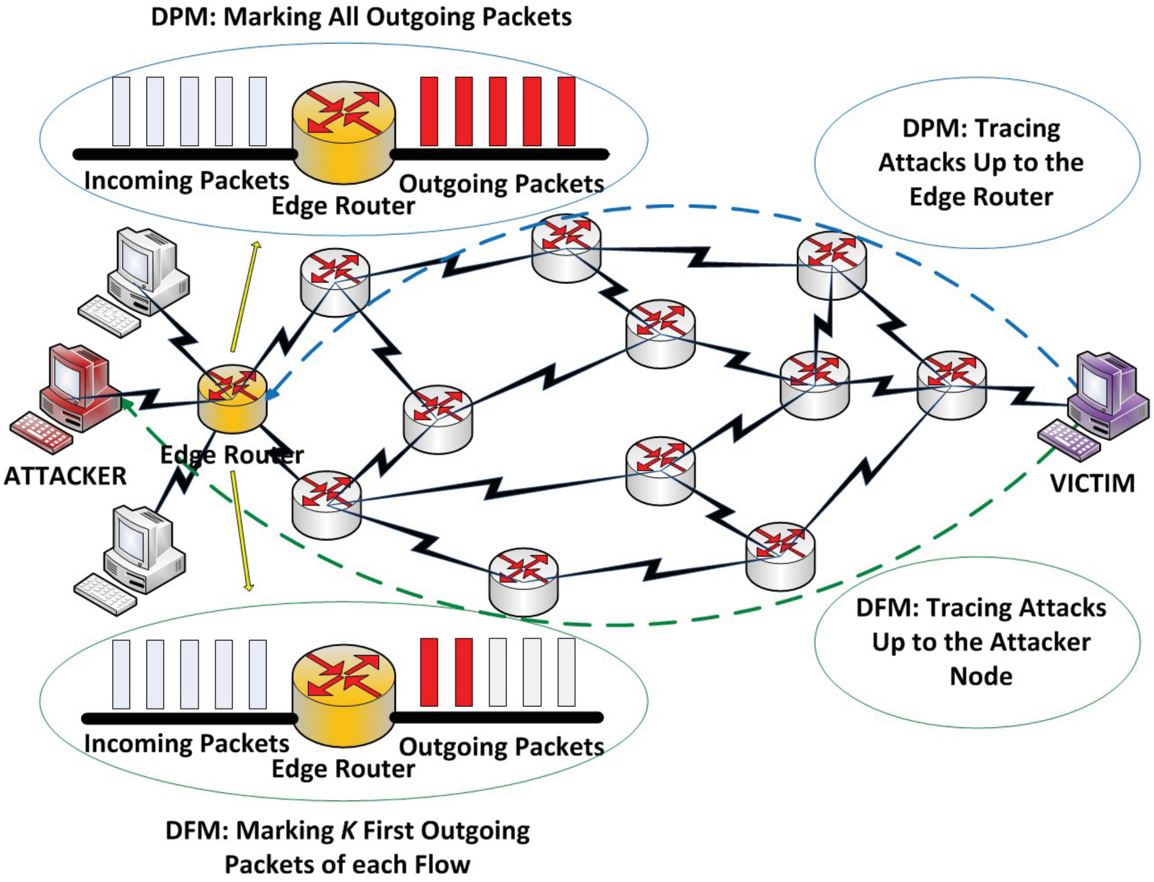

DPM: Marking All Outgoing Packets
Incoming Packets
Outgoing Packets
Edge Router
DPM: Tracing Attacks Up to the Edge Router
ATTACKER
Edge Router
VICTIM
DFM: Tracing Attacks Up to the Attacker Node
Incoming Packets
Outgoing Packets
Edge Router
DFM: Marking *K* First Outgoing Packets of each Flow

Figure 4.2: A Schematic Illustration of both DPM and DFM Approaches. This Figure Compares the Marking and the Traceback Procedures of both DPM and DFM Approaches. Blue Lines: DPM. Green Lines: DFM.

- **MAWI** network traces are from a traffic data repository maintained by the MAWI Working Group of the WIDE Project [8]. These traffic traces are in tcpdump format, and the IP addresses in the traces are scrambled for privacy reasons.

- **The CAIDA DDoS attack 2007 data set** contains approximately one hour of anonymized traffic traces of DDoS attacks captured on August 4, 2007. This data set represents DDoS attacks where the attackers attempt to block access to the targeted victim by consuming the victims computing resources and by consuming all of the bandwidth of the network connecting the victim to the Internet.

- **The CAIDA Anonymized Internet Traces** contain anonymized passive traffic traces from CAIDA's Equinix-Sanjose monitor on high-speed Internet backbone links. In this work the following CAIDA anonymized Internet traces data sets have been used: October 2012, December 2012 [14], March 2014 and July 2014 [15].

The data sets employed in this work are real-life tcpdump files and are all available publicly.

Table 4.4 represents statistical information of all aforementioned network traffic tracess.

Data set	**Number of Packets**	**Number of Flows**	**Avg. # of Packet/Flow**	**size (byte)**
MAWI 2012	11,6737,412	6,635,840	17.59	36,678,015,695
CAIDA DDoS 2007	333,072,991	1,229,427	270.92	20,832,940,519
CAIDA Oct. 2012	103,487,076	10,268,919	10.08	30,433,317,702
CAIDA Dec. 2012	189,333,786	9,930,125	19.07	41,539,282,881
CAIDA Mar. 2014	281,944,583	17,393,190	16.21	78,087,247,181
CAIDA Jul. 2014	234,383,915	19,234,385	12.19	85,832,026,129

Table 4.4: Statistical Information on the nine Evaluation Data Sets

A testbed network was established in the research lab, Figure 4.3. As shown in this figure, one local network for evaluating the DFM technique on all of the 6 data sets given above is implemented. For this purpose, these data sets were replayed on the testbed network using the tcpreplay and tcprewrite open source applications [1]. In addition, two real time programs were implemented using the Winpcap library by

C++ [12], one for marking and the other for tracing back the source of the flows for DFM. The marking program runs at the source edge router and marks only those flows travelling from the inside of the network to the outside. At the same time, the traceback program runs at the destination nodes and tries to trace the source nodes of the marked traffic. As I do not have access to modify the kernel of marking router, I set up a bridge system between the marking router and the local network, which is transparent to both, to enable the marking module and to mark the outgoing traffic on the fly.

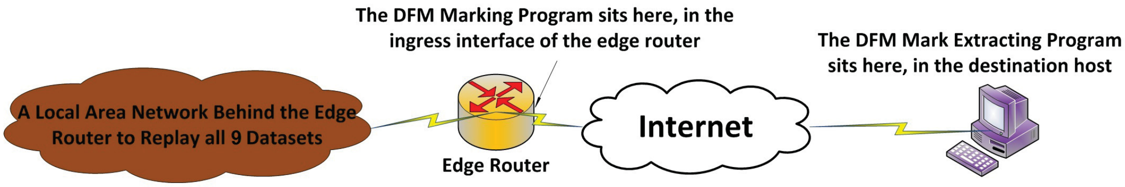

The DFM Marking Program sits here, in the ingress interface of the edge router
The DFM Mark Extracting Program sits here, in the destination host
A Local Area Network Behind the Edge Router to Replay all 9 Datasets
Internet
Edge Router

Figure 4.3: The Testbed Network for Analyzing DFM.

4.4.1 Traceback and Marking Rates

As described above, the marking method divides 60 bits of identification data for each flow into K fragments and passes it to the destination by the first K packets of each flow. Therefore, the mark inserted into each packet contains $M = 60/k$ bits of the identification data, $S = log_2 K$ bits for identifying a fragment and a 1-bit flag for identifying the marked and the unmarked packets in a flow. However finding the best number for K is an important issue. There are three metrics which are important for choosing the best value for K as listed below.

- **TR:** Traceable Rate: the ratio of the number of successfully traced back packets to all packets.
- **MR:** Marking Rate: the ratio of the packets marked by the edge router to all packets.
- **NB:** Number of Bits: the total number of bits that are embedded in the IP header of each mark-carrying packet.

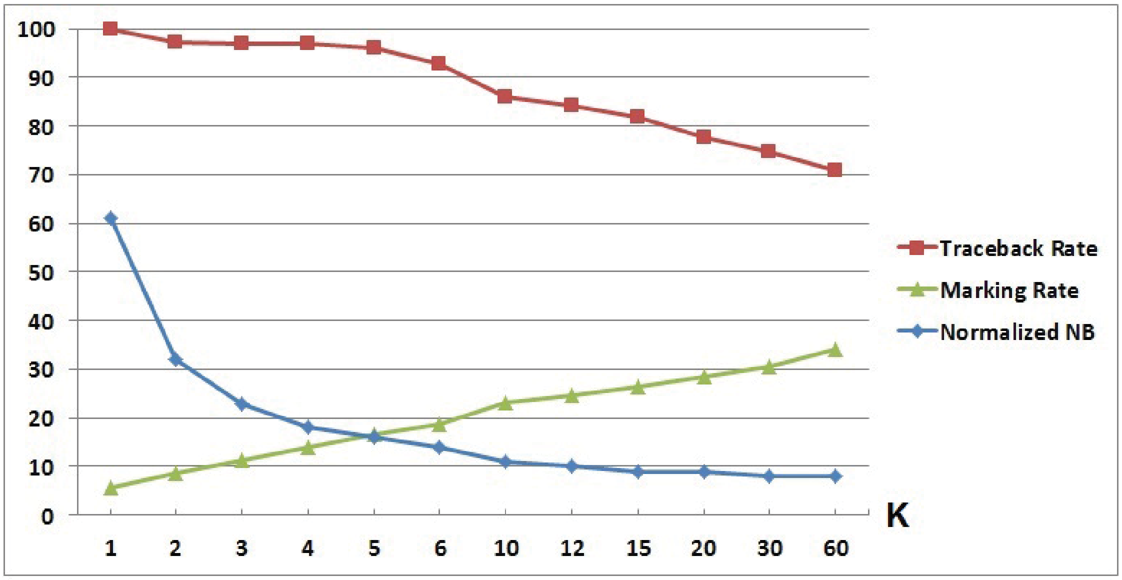
A desirable K should result in higher values of TR and lower values of MR and NB. Tables 4.5 presents the relation between K, M, S, NB, TR, and MR for all nine data sets on the DFM approach. For a better understanding of the topic, Figure 4.4 presents nine charts, one chart for each data set, showing the values of TR, MR and NB for several values of K for the proposed DFM approach.

				MAWI Dec2012		**CAIDA DDoS2007**		**CAIDA October2012**	
K	M	S	NB $(M+S+1)$	TR	MR	TR	MR	TR	MR
1	60	0	61	100	5.68	100	0.37	100	9.52
2	30	1	32	97.23	8.6	99.84	0.58	91.48	11.32
3	20	2	23	96.96	11.38	99.75	0.58	91.06	11.82
4	15	2	18	96.84	14.12	99.69	0.6	90.03	12.43
5	12	3	16	96.2	16.7	99.66	0.62	89.31	13.36
6	10	3	14	92.88	18.62	99.63	0.63	89.11	13.74
10	6	4	11	86.06	23.17	99.56	0.68	88.82	13.96
12	5	4	10	84.27	24.75	99.52	0.71	88.48	14.12
15	4	4	9	81.74	26.55	99.46	0.77	88.08	14.28
20	3	5	9	77.84	28.37	99.31	0.89	87.64	14.98
30	2	5	8	74.73	30.66	98.64	1.51	87.12	16.15
60	1	6	8	70.76	34.14	96.47	3.58	85.95	18.91

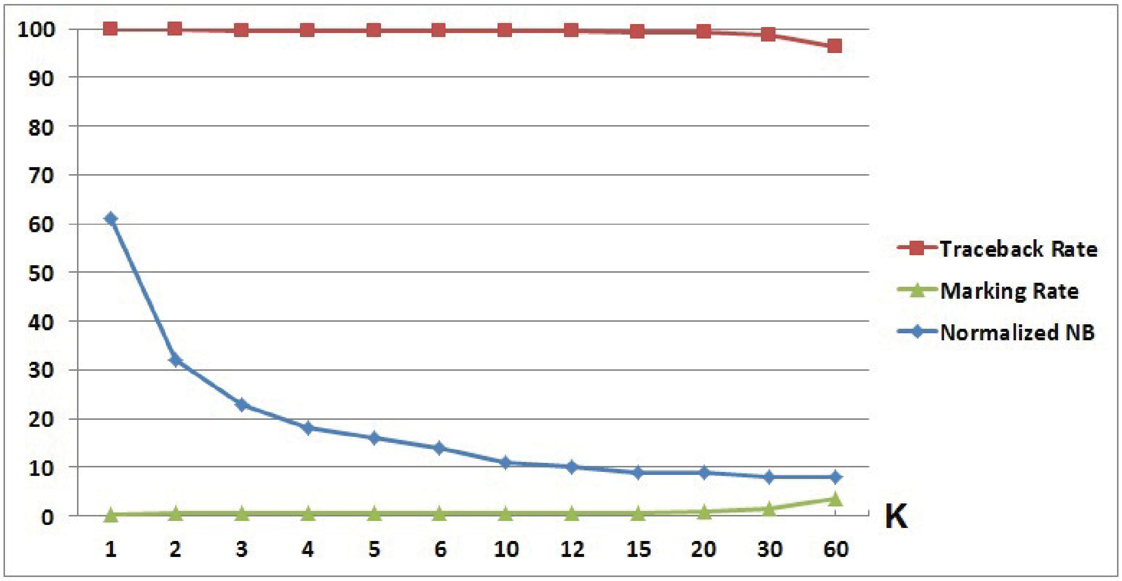
				CAIDA December2012		**CAIDA March2014**		**CAIDA July2014**	
K	M	S	NB $(M+S+1)$	TR	MR	TR	MR	TR	MR
1	60	0	61	100	5.24	100	6.17	100	8.21
2	30	1	32	95.75	6.24	94.57	6.91	92.7	9.11
3	20	2	23	95.24	6.24	94.08	6.9	92.05	9.11
4	15	2	18	94.6	6.45	93.75	7.02	91.61	9.26
5	12	3	16	94.09	6.71	93.47	7.15	91.24	9.44
6	10	3	14	93.83	6.87	93.28	7.27	91.03	9.57
10	6	4	11	93.16	7.34	92.49	7.83	90.23	10.14
12	5	4	10	92.89	7.56	92.22	8.05	89.98	10.34
15	4	4	9	92.48	7.92	91.95	8.27	89.67	10.6
20	3	5	9	92.06	8.28	91.67	8.52	89.37	10.87
30	2	5	8	91.66	8.65	91.26	8.9	88.97	11.24
60	1	6	8	87.5	12.63	90.43	9.7	88.14	12.03

Table 4.5: The Relationship between K, M, S, NB, TR and MR on all six Evaluation Data Sets for the DFM Approach

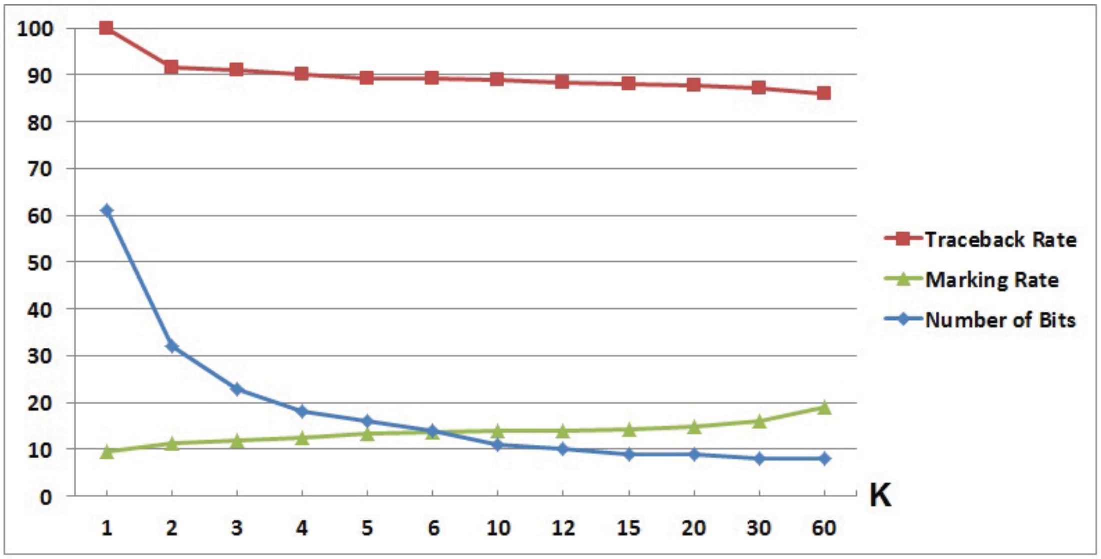
The most interesting thing that can be observed in Figures 4.4 is that unlike the existing traceback methods in which reducing MR reduces TR, in DFM reducing MR increases TR. It means that DFM can achieve a high traceback rate (TR) while marking a lower number of packets, i.e. a low marking rate (MR). The reason is that obviously the flows with fewer than required K packets are unable to carry all the K marked fragments, so there are untraceable. Therefore, decreasing K results in having more traceable flows (a higher traceback rate, TR). Also, it results in the marking of fewer packets in each flow (a lower marking rate, MR).



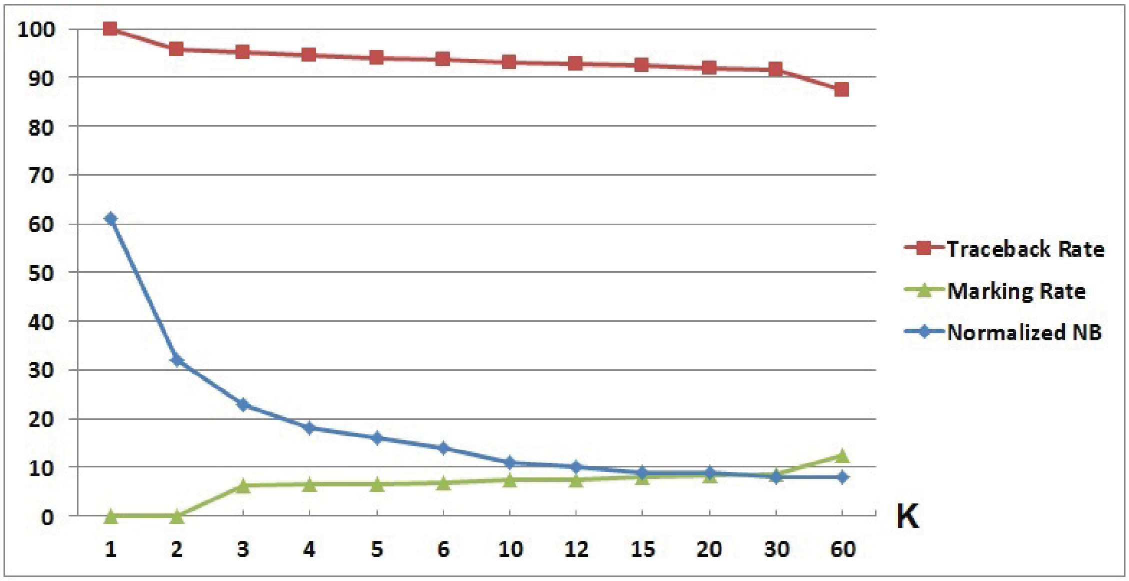
(a) MAWI December 20102



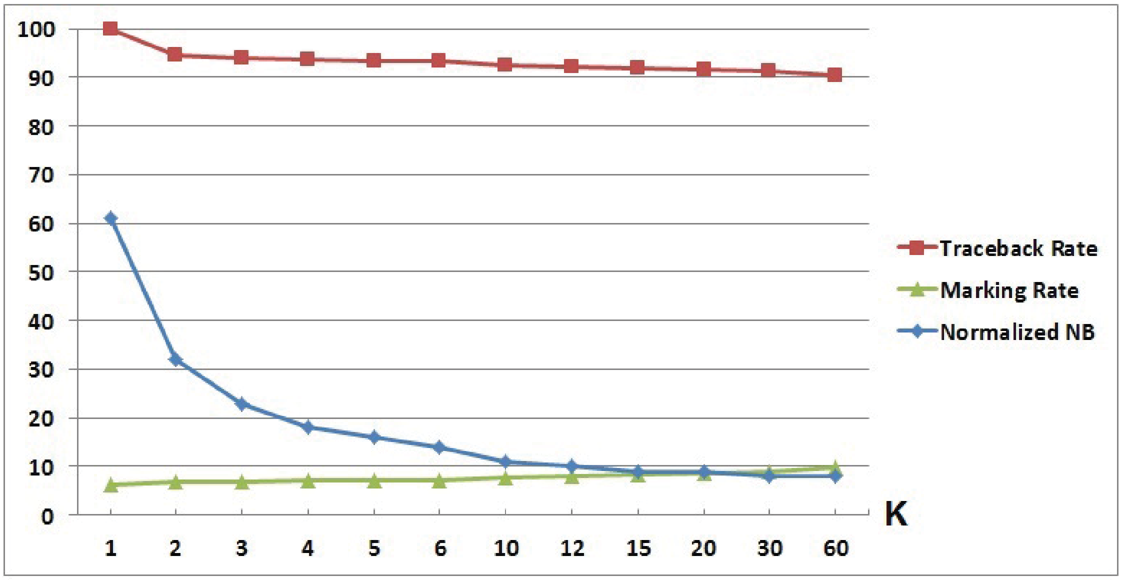
(b) CAIDA DDoS Attack 2007



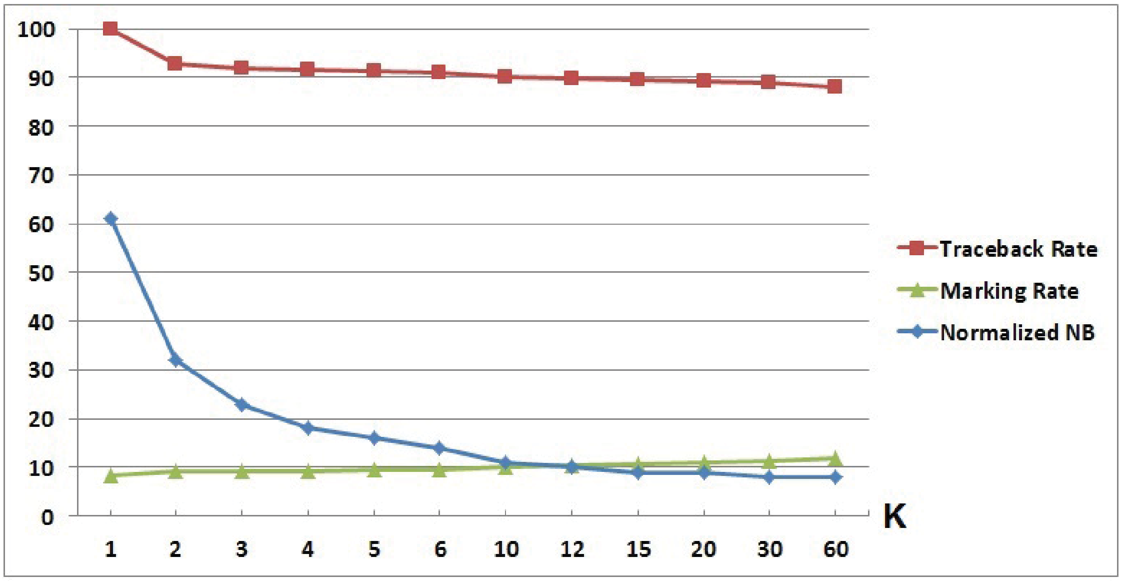
(c) CAIDA October 2012



(d) CAIDA December 2012



(e) CAIDA March 2014

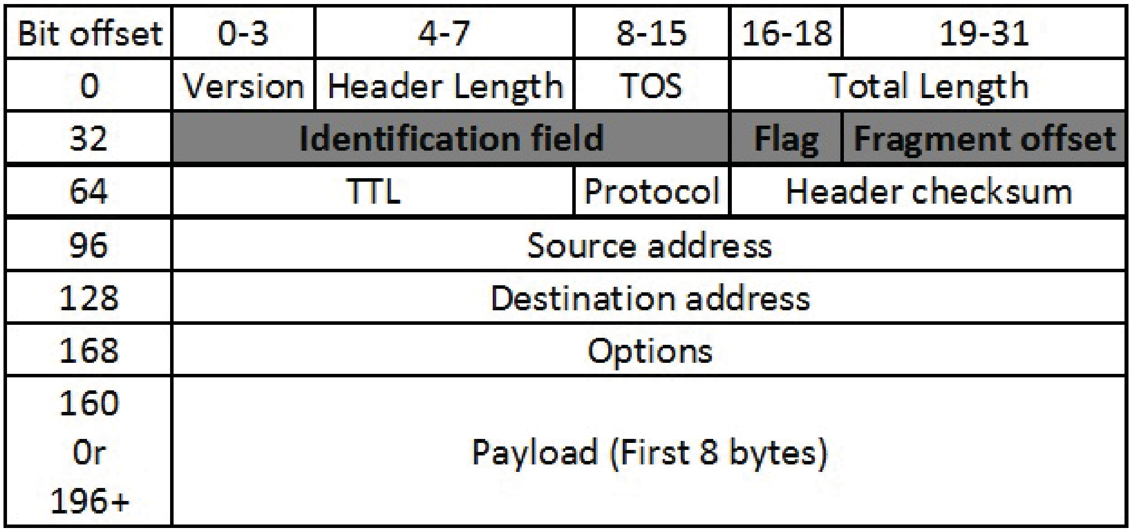


(f) CAIDA July 2014

Figure 4.4: Traceback Rate, Marking Rate, and Number of Bits with different values of K for all 9 Evaluation Data Sets.

However, while decreasing K results in a higher TR and a lower MR, which are both desirable; it also increases NB, which is undesirable. Since TR is a very important factor in DFM, focus has been on values of $K \leq 5$ which have a reasonable traceback rate (around 90 to 99%). On the other hand, $K = 1$ cannot be an option because in that case the required NB becomes 61 bits. It should be noted here that finding 61 rarely usable bits in an IP header to embed marking bits is almost impossible. Selecting K values from 2 to 5 is a trade-off between MR and NB. While lower K values have better MR and worse NB, higher K values have worse MR and better NB. As most traceback methods utilize 16 bits of the identification field in the IP header to embed the marking bits [50], [79], [91], [26], [28], $K = 5$ seems to be a good option. It requires 16 bits of IP header and around 12 to 33% of all packets end up being marked. However in this implementation it is possible to set $K = 2$. This case has a better MR compared to $K = 5$, Table 4.5, but more NB (32 bits). Based on previous work on IP-Traceback [58], [94], it is possible to use the $identification$, $flag$ and $fragment$ offset fields of the IP header as a 32-bit marking field, Figure 4.5. Fortunately the use of the $fragment$ and the $identification$ fields in the IP header affect only 0.06% of legitimate packets [58], [94].

It should be noted here that each flow requires at least K packets to carry the entire marking fragmentations. So if the number of packets in a flow is less than K, then this flow cannot be tracebacked by DFM. This is the reason why changing the K changes the TR in Table 4.5. As a result, it can also be concluded that if any of the marked packets get lost and does not arrive to the destination, the specific flow that the lost marked packet belongs to cannot be traced back. However, the victim can traceback to the attacker using the other flows from the same attacker.



Bit offset	0-3	4-7	8-15	16-18	19-31
0	Version	Header Length	TOS	Total Length	
32	Identification field			Flag	Fragment offset
64	TTL		Protocol	Header checksum	
96	Source address				
128	Destination address				
168	Options				
160 Or 196+	Payload (First 8 bytes)				

Figure 4.5: Using the Gray Fields as Marking Field in the IP Header for $K = 2$

The results show that marking the first two packets of every outgoing flow using DFM makes it possible to determine correctly the origin of 92 to 99% of all packets (TR), while it requires that only 0.5 to 11% of all packets to be marked (MR). Moreover, DFM correctly determines the origin of 90 to 99% of packets (TR) by marking 0.6 to 17% of all packets (MR) if the first 5 packets of every outgoing flow are marked (gray rows in Table 4.5).

Table 4.6 shows the evaluation of the DPM approach on all six evaluation data sets, using the same TR and MR metrics used to evaluate the DFM approach. Although it is expected to have a 100% traceback rate using the DPM approach, in fact, the TR for the DPM approach is less than 100% because fragmented traffic will be corrupted by the DPM. If a single fragment of the original datagram is marked, the reassembly function would fail at the destination.

Data Set	TR	MR
MAWI December 2012	99.42	100
CAIDA DDoS attack 2007	99.88	100
CAIDA October 2012	98.77	100
CAIDA December 2012	99.13	100
CAIDA March 2014	99.08	100
CAIDA July 2014	99.03	100

Table 4.6: The TR and MR of the DPM Approach for All Nine Evaluation Data sets

By comparing Table 4.5 and Table 4.6, it can be seen that DPM has a higher traceback rate compared to DFM (TR), however, this accuracy is achieved by marking all of the packets in the network (MR).

4.4.2 Memory Usage

This section describes the space required for marking the traffic by the DFME module and the space required by the DFMD module to extract the marking data.

Memory Usage of the DFME module at the Edge Routers

The space required for running the DFME module on an edge router, $DFME_{mem}$, is equal to the sum of the required space of three tables, namely the Marking-Table, the NI-ID-Table and the Node-ID-Table:

$$DFME_{mem} = NT + NDT + MT \tag{4.1}$$

Where NT is the size of the NI-ID-Table, NDT is the size of the Node-ID-Table and MT is the size of the Marking-Table.

1. **NI-ID-Table:** As described in Section 4.2.1, the NI-ID is assigned to each interface of either the MAC address of a network interface on the edge router, or the virtual local area network (VLAN) ID of a virtual interface if the edge router uses VLAN interfaces. The NI-ID specifies which subnet a traffic flow comes from. A marking router keeps an NI-ID-Table and assigns an NI-ID to each interface from 0 to a maximum of 65535. Each NI-ID-Table entry

consists of an NI-ID, the MAC address of the network interface and the Vlan ID. Therefore NT is calculated by:

$$NT = N_i \times (NI_{ID} + I_{MAC} + V_{ID}) \tag{4.2}$$

Where N_i is the number of physical and virtual interfaces on the marking router, NI_{ID} is the size of the egress interface IP address of the marking router, I_{MAC} is the size of the router interface MAC address and V_{ID} is the size of the Vlan ID.

Since the NI_{ID}, the I_{MAC} and the V_{ID} are 1.5 bytes, 6 bytes and 1.5 bytes respectively, the size of NI-ID-Table (equation 4.2) can be summarized as:

$$NT = 9N_i \tag{4.3}$$

2. **Node-ID-Table:** As described in Section 4.2.1, the Node-ID is an identifier assigned to each source MAC address observed from the incoming traffic from the local networks. For every newly observed source MAC address, there is a Node-ID-Table record which consists of the NI-ID of the interface that the traffic comes from and the observed source MAC address on the incoming packet.

$$NDT = N_{MAC} \times (NI_{ID} + S_{MAC}) \tag{4.4}$$

Where N_{MAC} is the number of observed unique MAC addresses still available in the Node-ID-Table and S_{MAC} is the size of the source MAC address. Thus, the size of this table varies and is based on the number of observed unique source MAC addresses still available in the Node-ID-Table. The DFME module 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.

Since the size of S_{MAC} is 6 bytes, the size of Node-ID-Table (equation 4.4) can be summarized as:

$$NT = 7.5N_{MAC} \tag{4.5}$$

3. **Marking-Table:** In addition to the NI-ID-Table and the Node-ID-Table, DFME utilizes another table called the Marking-Table to keep track of the packets and their Flow-ID, as well as the marking data. Each row in this table stores a Flow-ID, its flow marking data and the packet number. The edge router increases the packet number by one in the corresponding flow record for each transmitted packet. In other words, this number indicates the number of packets in a flow. DFM uses this number for keeping track of the first K packets of every flow. The DFME module does not keep the record of a flow when the flow is over. The end of a flow is detected by a four-way connection termination handshake or a pre-defined time-out duration for the TCP flows and a pre-defined time-out duration for the UDP and ICMP flows. Consequently, the number of rows in the Marking-Table varies and depends on the number of concurrent flows, which can be calculated by:

   $$MT = N_{CF} \times (F_{ID} + MD + P_n) \tag{4.6}$$

   where N_{CF} represents the number of concurrent flows in the traffic, F_{ID} represents the size of the Flow-ID, MD represents the size of the marking data and P_n is the size of the packet number (which is 2 bytes).

   As described in Section 4.2.1, Flow-ID is defined as the five tuples of the source IP address, the destination IP address, the L4 protocol type (TCP/UDP), the source port number and the destination port number for the TCP and UDP flows. ICMP flows are defined as the six tuples of the source IP address, destination IP address, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID. Therefore for TCP or UDP flows:

   $$F_{ID} = S_{IP} + S_P + D_{IP} + D_P + Pro \tag{4.7}$$

   where S_{IP}, D_{IP}, S_P, D_P and Pro are the size of the source IP address, destination IP address, source port, destination port and protocol respectively.

   For ICMP flows:

$$F_{ID} = S_{IP} + D_{IP} + Pro + T_{ICMP} + C_{ICMP} + ID_{ICMP} \tag{4.8}$$

where T_{ICMP}, C_{ICMP} and ID_{ICMP} are the size of the ICMP type, ICMP code and ICMP ID respectively.

Since the S_{IP} and the D_{IP} are 4 bytes each, the S_P, D_P and ID_{ICMP} are 2 bytes each, and the Pro, T_{ICMP} and C_{ICMP} are 1 byte each, then the F_{ID} for either the TCP/UDP (equation 4.7) or the ICMP flows (equation 4.8) is 13 bytes.

As described above, the marking data consists of the egress interface IP address of the marking router, the NI-ID and the Node-ID. Therefore:

$$MD = E_{IP} + ND_{ID} + NI_{ID} \tag{4.9}$$

where E_{IP}, ND_{ID} and NI_{ID} are the size of the egress interface IP address of the marking router, the size of the Node-ID and the size of the NI-ID, respectively.

Since E_{IP} is 4 bytes and ND_{ID} is 2 byte, then MD is 7.5 byte. Now equation 4.6 can be summarized as:

$$MT = 22.5N_{CF} \tag{4.10}$$

Therefore the space required for running the DFME module on an edge router, $DFME_{mem}$ (equation 4.1) is sum of equations 4.3, 4.5 and 4.10:

$$DFME_{mem} = 9N_i + 7.5N_{MAC} + 22.5N_{CF} \tag{4.11}$$

The marking router in the testbed network, Figure 4.3, has two interfaces (N_i), and the number of simulated source MAC addresses (N_{MAC}) is 105. Table 4.7 represents the space required for running the DFME module at the edge router for all six evaluation data sets. It shows that the maximum required space for running the DFME module in the testbed network is 42.42 KB for the CAIDA July 2014 data set.

Memory Usage of the DFMD module at the victim's side

The space required for running the DFMD module at the victim's side, $DFMD_{mem}$, is equal to the space required for the Reconstruction-Table. The victim maintains this table for matching the Flow-ID and K possible mark fragments. For every observed flow, a table entry including 13 bytes of Flow-ID and K fragments of marking data should be stored. Like the Marking-Table in the edge router, the victim no longer needs to keep the record of a flow when it is over. Therefore, the space required at the victim's side varies and is based on the number of concurrent flows and K, which can be calculated by:

$$DFMD_{mem} = N_{CF} \times (F_{ID} + (K \times NB)) \tag{4.12}$$

where K and NB can be found from Table 4.5. Since F_{ID} is 13 bytes, then equation 4.12 for $K = 2$ can be summarized as:

$$DFMD_{mem} = 77N_{CF} \tag{4.13}$$

and for $K = 5$ can be summerized as:

$$DFMD_{mem} = 93N_{CF} \tag{4.14}$$

Table 4.7 represents the space required for running the DFMD module at the victim's side for all six evaluation data sets. It shows that the maximum required space for running the DFMD module in the testbed network is 172.1 KB for the CAIDA July 2014 data set when $K = 5$.

Data Set	N_{CF}	$DFME_{mem}$	$DFMD_{mem}$	
			K=2	K=5
MAWI December 2012	930	21.22	69.93	84.46
CAIDA DDoS attack 2007	754	17.35	56.7	68.48
CAIDA October 2012	1131	25.64	85.05	102.72
CAIDA December 2012	1043	23.7	78.43	94.73
CAIDA March 2014	1724	38.67	129.64	156.57
CAIDA July 2014	1895	42.42	142.5	172.1

Table 4.7: The Space Required by the DFME and the DFMD Modules in KB for the Nine Evaluation Data Sets, when $N_i = 2$ and $N_{MAC} = 105$.

4.5 Authenticated Flow Marking

Using DFM, the destination is able to trace up to the source node of the received traffic by extracting the edge router's IP Address, NI-ID and Node-ID from the marked packets of each flow. Although DFM has promising results, in cases of a compromised router on a network path, the marking data could be changed and the destination would be unable to identify the origin of the traffic.

To address this problem, adding an optional authentication mechanism to DFM is proposed. This means that even if a compromised router managed to break into the connection, it would not be able to decrypt and change any of the marking data which passes between the DFME and the DFMD modules.

A question which has been raised is why make the authentication feature optional? Clearly, authentication consumes resources such as CPU time in a participating router. However, in a highly secure network in which authenticity is important, it is essential to ensure that the marking information cannot be changed by the compromised routers on a network path.

To have a secure connection between two ends an authentication mechanism is proposed which relies on the https secure connection algorithm. Using the https secure connection algorithm has the two main purposes listed below.

1. Authentication: verifying that the victim is talking directly to the marking router that the victim thinks it is talking to.
2. Data Integrity: ensuring that the marking data is delivered exactly as intended.

A hybrid cryptographic system is proposed which makes use of Asymmetric Cryptography for authentication and sharing the symmetric cryptography secret key, as well as Symmetric Cryptography for encrypting the identification data. In fact, the encrypted identification data works as the digital signature of the marking router. Digital signing is used to ensure that the marking data has not been changed along the path, and the mark is in fact valid.

4.5.1 Asymmetric Key Encryption

Asymmetric or public key encryption is a type of cryptographic system in which each party has both a private and a public key, which are mathematically linked to each other. The public key is used for encrypting plain text to cipher text, while the private key is used for decrypting that cipher text back into plain text.

Once a message has been encrypted by a public key, it can only be decrypted with the corresponding private key. Neither key can perform both functions by itself. The public key can be published freely without compromising the security of the system, but the private key must not be revealed to anyone who is not authorized to decrypt messages. One of the benefits of asymmetric key cryptography is that two parties with no prior knowledge of each other can create a secure connection while initially communicating over an open, insecure connection.

The authenticated flow marking should be initiated by the DFMD module on the destination network. Similar to what we have in https handshake, the handshake begins with the DFMD sending a ClientHello message. This contains all the information the DFME needs in order to connect to the DFMD, including the various cipher suites. The DFME on the marking router responds with a ServerHello, which contains similar information required by the client, including a decision based on the client's preferences about which cipher suite will be used.

The DFME and DFMD modules use asymmetric key cryptography to agree upon a shared secret key for the session. That means that even if someone is sitting in between the marking router and the victim, and watches the connection happen, they still cannot determine the secret key for the session. One of the more common ways this key exchange is performed is by using a Diffie-Hellman (DH) key exchange. This process allows the DFME and the DFMD modules to agree upon a shared secret, without having to transmit that secret over the connection. Once the initial DH exchange takes place, the resulting shared secret can be used to encrypt further communications in that session using a much simpler symmetric key encryption. The math behind it is actually fairly simple to calculate one way, but essentially impossible to reverse. Wikipedia has a great image, Figure 4.6, involving mixing colors to explain how two parties agree upon a shared secret key [4].

Notice how the starting color (yellow) ends up getting mixed with both Alice's color and Bob's color. That is how it ends up being the same for both parties at the end. The only thing that is sent over the connection is the half-way-done mixture which is meaningless to anyone watching the connection.

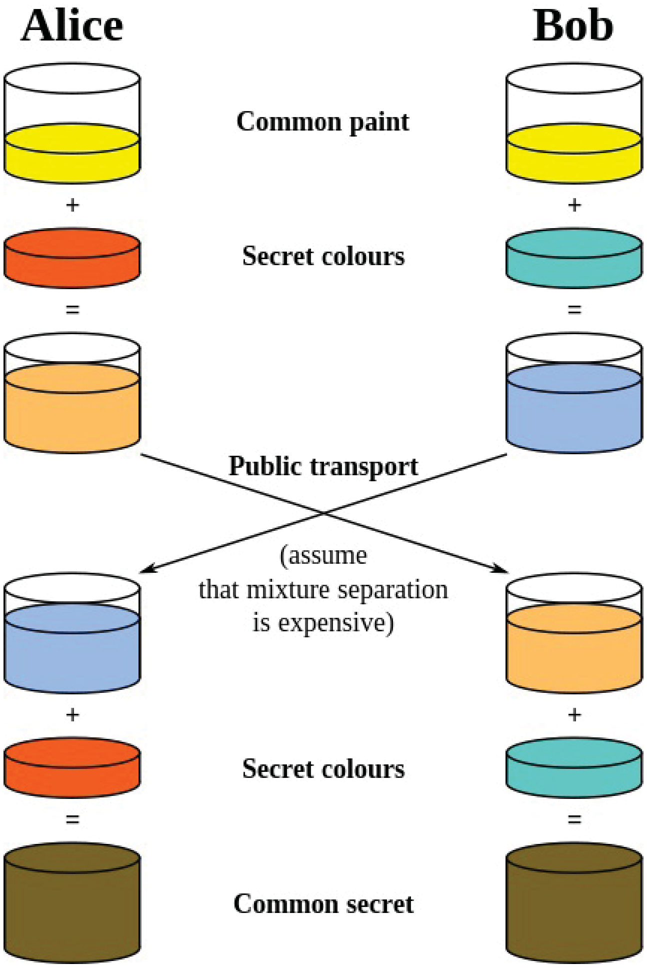


Figure 4.6: Illustration of the Diffie-Hellman Key Exchange

4.5.2 Authentication

The Diffie-Hellman key exchange allows two parties to create a private, shared secret. However, how do the two parties know they are talking to the correct entity?

In order to be trusted by the end users (victims), the certificates of the marking routers have to be signed by a trusted Certificate Authority (CA). CAs are companies that perform manual inspection and review to make sure that the applying entity is a real person or business that exists in the public record, and is in control of the domain for which they are applying. Once the CA verifies that the applicant is real and really owns the domain, the CA will sign the network's certificate, essentially putting their stamp of approval on the fact that this network's public key really belongs to them and should be trusted.

4.5.3 Symmetric Key Encryption

The public key exchange only needs to happen once per session, the first time a DFME module and a DFMD module connect. Once they have agreed on a shared secret, the DFME module can communicate to the DFMD module using a symmetric-key crypto system which is much more efficient to communicate on since it saves an extra round-trip each exchange. With the agreed shared secret the DFME encrypts the marking data, and DFMD decrypts the marking data securely, with a compromised router seeing just cypher text.

AES, the Advanced Encryption Standard, is currently one of the more popular algorithms used in symmetric key cryptography (for example, as used for actual data transmission in SSL and TLS). This thesis proposes the use of AES for encrypting the 60 bits of identification data and sending it along with the plain text marking data from the marking router to the destination. Since a compromised router does not know the AES secret key, it cannot forge flow markings. The flow-specific information is necessary to prevent a replay attack, because otherwise, a compromised router can forge other routers' markings simply by copying their marking data and encrypted identification data into other flows. For flow specific information, the flow-ID could be used.

AES is specified with block and key sizes which may be any multiple of 32 bits. The most popular AES block and key sizes are 128, 192 and 256 bits. A block and key size of 192 bits (24 bytes) is used in order to be able to apply the AES algorithm on the 60 bits of identification data plus the 13-byte Flow-ID (28 bits padding for 192 bit blocks is required). This 24 bytes of encrypted data should also transfer to the destination along with the plain text marking data embedded in the header of the first K packets of each flow. This work proposes the addition of 24 bytes of AES output to the end of the K^{th} packet payload of each flow. In fact this 24 bytes of encrypted data acts as a digital signature to ensure that the marking data is embedded in the flow by the marking router and not changed along the path by a compromised router.

4.5.4 Verification by Destination

As described earlier, each destination maintains a Reconstruction-Table (Table 4.3). In cases of authenticated flow marking, each destination should add two more fields to this table: one field for the digital signature, and another field for the verification status, Table 4.8. When the destination gets a signed flow, it uses the shared secret key to decrypt the flow-ID and the 60 bits identification data. Then the destination validates the flow by comparing the decrypted data with the real flow-ID and the extracted identification data from the header of first K packets of the flow. If the flow passes the verification, the destination knows that the mark has not been changed and is in fact valid; otherwise, it would reject the flow.

Flow-ID	**First Mark Frag**	**Second Mark Frag**	**Digital Sign**	**Identification**	**Sign Verification**
AC100C14055A5585 B02A005006 Srcip=172.16.112.20 SrcPort=1370 Dstip=85.133.176.42 DstPort=80 Protocol=tcp	AC100001	4004005F	24 bytes sign data	EdgeRouter IP= 172.16.0.1 NI-ID=1 Host-ID=23	Verified
AC10715410925585 B817001511 Srcip=172.16.113.84 SrcPort=4242 Dstip=85.133.184.23 DstPort=21 Protocol=udp	AC100001	40080037	24 bytes sign data	EdgeRouter IP= 172.16.0.1 NI-ID=2 Host-ID=13	Failed

Table 4.8: An Example of the Reconstruction-Table for the DFM Authenticated Flow Marking Method

One advantage of the proposed authenticated flow marking method is that it is optional. Authenticated flow marking needs only to be initiated in a situation when the victim is under attack. In the networks where security is critical, it is possible to assign a specific system outside the critical network for initiating the authenticated flow marking to ensure that it will be initiated even if the network is under a heavy DDoS attack.

4.6 Experimental Results of the Authenticated Flow Marking

To evaluate the proposed authentication DFM method, this method was applied on the same six data sets as used in Section 4.4. To this end, the original DFM implementation was modified in both the DFME and DFMD modules to include the authentication component. The DFME module runs at the edge router and only marks and signs those flows travelling from the inside of the network to the outside. At the same time, the DFMD module runs at the destination node and tries to detect the source of marked traffic and verify the marking data.

4.6.1 Bandwidth Usage

It is clear that enabling authenticated flow marking increases network bandwidth usage given that an extra 24 bytes of signature data is embedded at 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 authenticated flow marking, Table 4.9. This comparison shows that performing the authenticated flow marking has less than a 0.81% bandwidth overhead.

Data Set	**Marking without Authentication**	**Marking with Authentication**	**Increment**
MAWI December 2012	898,704,130	908,659,114	1.11%
CAIDA DDoS 2007	20,832,940,519	20,862,446,767	0.14%
CAIDA October 2012	30,433,317,702	30,679,771,758	0.81%
CAIDA December 2012	41,539,282,881	41,777,605,881	0.57%
CAIDA March 2014	78,087,247,181	78,504,683,741	0.53%
CAIDA July 2014	85,832,026,129	86,293,651,369	0.53%

Table 4.9: The Size of Transmitted Traffic with and without the Authenticated Flow Marking in bytes

4.6.2 Memory Usage

Since the digital signature of a flow is created and embedded in the flow at the time of forwarding, no signature data is stored on the marking routers. Thus, performing the authenticated flow marking does not require any extra memory.

In summary, DFM can traceback the traffic to its origin with high accuracy in practice. This is achieved by marking approximately 0.58% (for $K = 2$ in the CAIDA DDoS 2007 data set) to 16.7% (for $K = 5$ in the MAWI December 2012 data set) of all the transmitted packets. In addition, this proposed authenticated flow marking method guaranties that the marking data is valid and unchanged in the network transmission path by a forged router.

4.7 DFM Analysis

To analyze DFM the same evaluation metrics used to analyze PPM and DPM were used as indicated below.

- **Computational Overhead:** As with DPM, in DFM only the closest edge router to the attacker is responsible for marking, and there is some computational overhead such as preparing the marking information and upgrading the marking fields. However unlike DPM, DFM does not require the calculation of the hash value of the ingress address. Moreover, unlike DPM which extracts the hash values of the ingress addresses, the victim uses the Flow-ID as a guide to prevent the combinatorial explosion problem [38] in DPM. In addition, unlike DPM, DFM marks only the first K packets of each flow, not all the packets, and the victim extracts marking information from only those packets in which the flag bit is set, again not for all the packets. Therefore, DFM has a lower computational overhead than DPM.

- **Memory Overhead:** Like PPM and DPM, memory overhead on the routers in DFM is negligible, and at the victim's side, DFM requires a small Reconstruction-Table. Table 4.7 represents the pace required for running the DFM approach. This is even less than the DPM requirements [38].

- **False Positives:** As discussed earlier, the DPM algorithm inserts the ingress interface IP address into outgoing packets by the first router of the attack path. Since the 32 bits that compose an IP address do not fit into the available marking space, Belenky and Ansari [38] propose to split the router IP address into K fragments. After receiving all K address fragments, the victim can recover the address by reassembling the received fragments. Their basic deterministic packet marking method has limitations when dealing with multiple attackers at the same time with the same source IP address. To counter this problem, they propose to produce hash values of the ingress interface and sending this hash value along with the marking data.

  Under the same conditions, the proposed method does not need to use hash functions because DFM first detects the flow to construct a valid mark. In other words, DFM uses the Flow-ID to prevent the combinatorial explosion problem. Thus, DFM is safer for countering the problem of multiple attackers with the same IP addresses at the same time, and there is no false positive under the DDoS attacks.

- **Mark spoofing by attackers:** As described above, DFM takes the advantage of the 1-bit flag to identify marked and unmarked packets in a flow. This flag should be set to “1” for the marked packets and “0” for the others. In this case, even if an attacker aims to spoof the mark, the flag bit will be overwritten once the flow passes through the marking router. Therefore, mark spoofing by the attacker is not an issue for the DFM approach.

- **Mark spoofing by subverted routers:** Unlike PPM and DPM, which do not have any solution for countering the malfunctioned routers in the attack path, DFM has an optional authenticated flow marking feature to ensure that the marking information has not been changed in the network path.

- **Awareness of the attack path length in advance:** Unlike PPM, the process of trackback in the DFM approach does not need the path length; therefore awareness of the attack path length is not an issue.

- **Awareness of the network map and the routing in advance:** Unlike PPM, the goal of DFM is not to reconstruct the attack path, so awareness of the network map and the routing is not an issue for DFM.

- **The number of required packets for traceback:** With the suggested $NB = 32$, DFM requires two packets and with $NB = 16$, DFM requires five packets to traceback up to the attacker node. This is lower than the eight packets required by DPM to traceback to the ingress interface address of the edge router [38], and much lower than the at least 1000 packets required by PPM to find the attack path [79].

- **Fragmentation:** Like PPM and DPM, 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.

- **ISP Involvement:** Unlike the probabilistic IP-Traceback approaches in the literature such as PPM [50] and all of its variants [91] [28] that require the participation of several routers in the attack path, DFM only requires the participation of one router, and it can be any router along the attack path. DFM is able to traceback the source of the traffic to the marking router, and also to the source network interface of the marking router, and even to the source node located behind the marking router. However, the ideal location for enabling DFM is the edge router. This enables traceback not only up to the source edge router, but also to the source network interface of the edge router and even one step further to the attacker node located in a LAN behind the edge router. In this case, the requirement on the ISP side would only involve upgrading their edge router's IOS and enabling the DFM feature on the router.

4.8 Discussion

In addition to all of the advantages of 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 local area network behind the edge routers. DFM assumes that each node in a local network may change its IP address, and 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, will DFM only 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 in which 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 the ECDSA signature.

4.9 Summary

In this chapter, an IP-Traceback technique called Deterministic Flow Marking (DFM), which is a novel real time three-level IP-Traceback method, is presented. For this evaluation six different network traffic traces, including the MAWI (Measurement and Analysis on the WIDE Internet) traffic archive December 2012, the CAIDA DDoS attack 2007, the CAIDA anonymized Internet traces October 2012, December 2012, March 2014 and July 2014 data sets, are employed and several metrics to evaluate the performance of the traceback schemes are used. Table 4.10 provides a summary of the evaluation and offers a comparison among DFM, DPM and PPM. The results show that DFM reduces the marking rate up to 88% on average with no false positives. Furthermore, the optional authentication scheme for DFM provides efficient authentication of routers' markings such that even a compromised router cannot forge or tamper markings from other uncompromised routers. Moreover, DFM traces the attack source up to the attacker node, even if the attack has been originated from a network behind a NAT server.

Comparison Metrics	**PPM**	**DPM**	**Basic DFM**	**Authenticated DFM**
Packet Marking Rate (MR)	low	100%	9% to 33%	0.58% to 33%
Mark Spoofing by Subverted Routers	Yes	Yes	Yes	No
Maximum Traceback Ability	Edge router	Ingress interface	Attacker node	Attacker node
Mark Spoofing by Attacker	Yes	No	No	No
Computational Overhead on Routers	Low	Low	Low	Fair
Computational Overhead on Victim	High	Low	Low	Fair
Memory Overhead on Routers	Low	Low	Low	Low
Memory Overhead on Victim	High	Low	Low	Low
Bandwidth Overhead	No	No	No	Low
Traceback Rate (TR)	Low	Good	Fair	Fair
False Positive Rate	High	Low, except heavy DDoS attacks	Low	Low
Number of required packets for traceback	1000	8	2 or 5	2 or 5
Awareness of the Attack Path Length in Advance	Yes	No	No	No
Awareness of the Network Map and Routing in Advance	Yes	No	No	No
Ability to handle Fragmentation	No	No	No	No
Ability to Handle Major DDoS Attacks	Poor	Fair	Good	Good
Number of Marking bits	16	17	If K=2: 16 If K=5: 32	If K=2: 16 If K=5: 32

Table 4.10: Comparison Table of PPM, DPM and DFM

Chapter 5

Probabilistic Flow Marking: PFM

PFM is an IP-Traceback approach which aims to find the origin of the network traffic, given that the source IP address of the packets is not authenticated and can be changed (can be spoofed or can be changed by intercepting devices like a NAT server). Although DFM, described in Chapter 4, shows a high traceback rate, low marking rate and no bandwidth overhead, this performance is achieved by marking all of the traffic flows. DFM marks every flow at the attacker side, and the victim needs to traceback each flow individually, even if the origins of all the flows are the same. This causes an overhead on the victim's resources. The approach in this chapter aims to minimize this overhead by proposing the Probabilistic Flow Marking approach (PFM). To this end, unlike the DFM that marks every flow by embedding the source identity data in the first K packets of each flow, PFM selects the packets randomly and marks them based on the flows they belong to. In fact, PFM aims to drop the marking rate of the DFM approach significantly without any notable change in the traceback rate.

Indeed, there is a trade-off between the marking rate and the traceback rate in such techniques. To this end, a sensitivity analysis is performed on this trade-off and the results compared with DFM. The goal is to find an acceptable range of flow marking probability for the new PFM approach. The new scheme encodes the identifications of the real source (origin) of the traffic and then embeds partial information into the forwarded packets (a randomly chosen few) destined to the victim on the marking router. Later the victim can infer the origin of the traffic by extracting the marking data from a small number of marked packets.

Based on the various IP-Traceback approaches described in Chapter 2, PFM falls into the following categories: Basic principle-Marking; Processing modes-Probabilistic at packet and flow levels; Location-Source group. The same assumptions have been applied to the PFM approach as were introduced for the DFM approach.

5.1 PFM Modules

PFM has been designed using two modules, the PFM Encoding module (PFME), which runs at the attacker-end edge router, and PFM Decoding module (PFMD), which runs at the victim-end. In the following sections these two modules are described in detail.

5.1.1 PFM Encoding Module (PFME)

The PFME module aims to mark the outgoing packets randomly with a probability of P. This module can be embedded as the traceback module on the routers, however only one router among all the routers in an attack path is responsible for the packet marking. Therefore, only one router along the attack path enables this feature and the rest are not involved in the traceback process. As with DFME 4.2.1, the PFME module uses three identifiers to select and mark the outgoing packets of a participating router randomly (see below).

1. The valid IP address of the egress interface of the marking router (32 bits).

2. The network interface identifier, NI-ID, (12 bits): this is an identifier assigned to each interface of either the MAC address of a network interface on the router, or the VLAN ID of a virtual interface if the router uses VLAN interfaces. The NI-ID specifies which subnet a traffic flow comes from.

3. Node-ID (16 bits): an identifier assigned to each source MAC address observed on incoming traffic. Each MAC has a unique Node-ID. In [30], it has been shown that spoofing the source MAC does not affect on the traceback process.

Using the above three identifiers to mark the packets, PFM is able to traceback the source of the traffic to the marking router as well as to the source network interface of the marking router, and even to the source node located behind the marking router. To make the best of the PFM scheme, the ideal location to enable the PFME module is on the edge routers. Edge routers are the closest routers to the end networks with at least one valid assigned IP address in their egress interfaces. In this case, PFM is able to traceback not only up to the source edge router, but also to the exact source network interface of the edge router and even one step further to the source node located in a LAN behind the edge routers. If instead of the edge router, the PFME module is enabled on any other router among the path, then the PFM approach is able to traceback up to one hop beyond the PFME-enabled router.

PFM takes advantage of the fact that all packets in a flow have the same source. Thus, if the origin of some packets in a flow can be found, then the origin of all of the other packets in the same flow is also discovered. PFM uses this fact to decrease the marking rate and to increase the traceback rate simultaneously. In doing so, the PFME module at the source side selects and marks the packets with a probability of P. At the same time, it keeps track of the flows to which the selected packets belong because all of the randomly-marked packets of one flow should carry different portions of the same identification data.

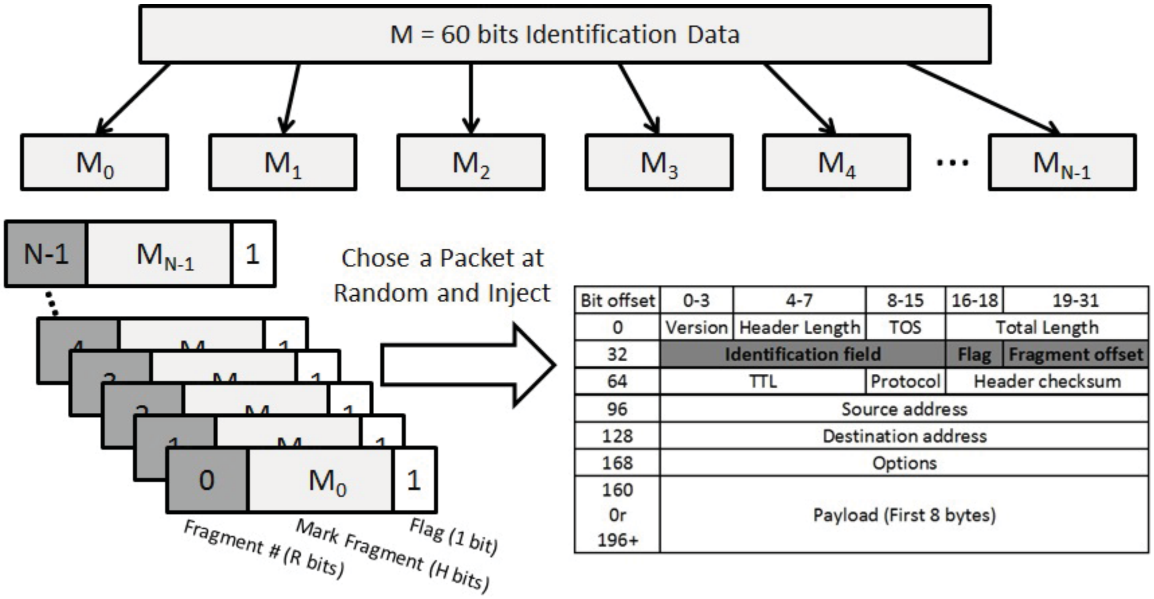
The PFM identification data consists of the IP address of the egress interface (32 bits) + the NI-ID (12 bits) + the Node-ID (16 bits) = 60 bits for distinguishing the traffic of a particular node from the other nodes. This identification data should be divided into K fragments, and each randomly-selected packet in a flow should carry one of these fragments. Therefore, each randomly selected packet carries $H = 60/K$ bits of the identification data fragment and $R = log2(K)$ offset bits to represent 2^R possible fragments of the identification data. PFM also embeds a one-bit flag, F, which should be set to "1" for the marked packets and "0" for the rest, Figure 5.1.

The PFME module maintains a table to keep track of the randomly selected packets and their Flow-ID, and the embedded marks in them, called the FragTable (the PFM has borrowed the definition of Flow-ID from the DFM, Section 4.2.1). Once the PFME module selects an outgoing packet with a probability of P, it extracts its Flow-ID first and then looks for the existence of a table record for this Flow-ID in the FragTable. If this Flow-ID is not in the FragTable, it means that this flow is new to the PFME modules, so the PFME module performs the following steps:

1. creating a table record for the new Flow-ID in the FragTable;

2. calculating the 60 bits of identification data;

3. dividing the identification data into K fragments and inserting them into the FragTable;



Bit offset	0-3	4-7	8-15	16-18	19-31
0	Version	Header Length	TOS	Total Length	
32	Identification field			Flag	Fragment offset
64	TTL		Protocol	Header checksum	
96	Source address				
128	Destination address				
168	Options				
160 0r 196+	Payload (First 8 bytes)				

Figure 5.1: Partitioning the Identification Data into K Fragments, Choosing the Packets Randomly and Injecting into the IP Header

4. marking the selected packet with one of the H bits identification data fragments, R bits fragment identifier, and one bit flag, F.

However, if this Flow-ID is in the FragTable, it means that this flow has already been seen by the PFME module, so the PFME module marks the selected packet by one of the fragments already available in the FragTable.

According to the results in Section 4.4.1, the best value of K is either 2 or 5. It means that to gain the best performance, the identification data should be divided into 2 or 5 fragments and each randomly selected packet should carry one fragment. If $K = 2$, then the size of each fragment is 32 bits (it uses 32 bits of the header to embed the marking data), includes the 30 bits of the identification data fragment, H, the 1-bit fragment offset, R and the 1-bit flag, F. If $K = 5$, then the size of each fragment is 16 bits (it uses 16 bits of the header to embed the marking data), including 12 bits the identification data fragment, H, three bits of the fragment offset, R, and the one-bit flag, F. Based on several previous studies on IP-Traceback [58] [94], it is possible to use the identification and flag fields of the IP header as a 16-bit marking field, or to use the identification, the flag and the fragment offset fields of the IP header as a 32-bit marking field, Figure 5.1. The use of the fragment and identification fields in the IP header affects only 0.06% of the legitimate packets [58] [94]. Therefore, in this proposed PFM traceback scheme, these three fields are employed to embed the marking data into the IP header of the packets.

Selecting a value of K between 2 or 5 is a trade-off between the marking rate and the number of required marking bits in the IP header. Selecting the lower value of K causes a lower marking rate (which is desirable) but needs more space in the IP header (which is undesirable). In the experimental results section, the sensitivity of the marking and traceback rates are evaluated to the value of K.

5.1.2 PFM Decoding Module (PFMD)

The PFMD module is located at the destination network and its goal is to infer the origin of the incoming traffic, even if the source IP addresses of the incoming packets are spoofed. This module maintains a table for matching the Flow-ID and K possible identification data fragments of a flow, called the ReconTable, in order to reconstruct valid source identification data. Once the source identification data of a flow is extracted, then the source of all packets in that flow can be found. The PFMD module checks for the flag field (which is set by the PFME module) in the packet's IP header of the incoming packets to find the marked packets. Once the PFMD module finds a marked packet, it extracts its Flow-ID and then checks for the existence of a table record for this Flow-ID in the ReconTable. If this Flow-ID is not in the ReconTable, it means that this flow is new to the PFMD module. So the PFMD module creates a table record for the new Flow-ID in the ReconTable and inserts the extracted identification data fragment into the ReconTable. However, if this Flow-ID was already in the ReconTable, it means that this flow has been already seen by the PFMD modules, so the PFMD module inserts only the extracted identification data fragment into the ReconTable. The order of different fractions of the same source identification data can be identified with the R field in the packet's IP header (for K=5, R is 3 bits and for K=2, R is 1 bit). Once all the fractions of the same source identification data of a flow have been received by the PFMD module, the origin of all the packets in that flow is apparent to the victim, even if the source IP address of those packets is incorrect or has been spoofed.

5.2 Experimental Results

To evaluate the proposed PFM approach and compare it with the DFM and DPM approaches, six network traces have been employed, including the MAWI (Measurement and Analysis on the WIDE Internet) traffic archive December 2012 [8], the CAIDA DDoS attack 2007 [17], the CAIDA anonymized Internet traces October 2012, December 2012 [14], March 2014 and July 2014 [15] data sets. The description of these data sets is presented in Section 4.4. The data sets employed in this work are real-life tcpdump files and are available publicly. Table 4.4 represents statistical information of all of aforementioned network traffic tracess.

Figure 5.2 shows the testbed network implemented in the research lab. As shown in this figure, one local network for evaluating the PFM approach on all 6 data sets given above is implemented and is connected directly to a marking router. The evaluation traffic is sent from the local area network behind the marking router and directed to the destination. For this purpose, the data sets were replayed on the testbed network using the tcpreplay and tcprewrite open source applications [1]. In addition, two real-time programs were implemented using the Winpcap library by C++ [12], one for implementing the PFME module to select and mark the packets randomly, and the other for implementing the PFMD module to traceback the source of the packets. The PFME module runs at the egress interface of the source edge router and only marks outgoing packets. At the same time, the traceback program runs at the destination network and tries to trace the origin of the marked traffic.

5.2.1 Traceback and Marking Rates

The most important metrics for evaluating a traceback approach are the Traceback Rate, TR, and the Marking Rate, MR. The definitions of these two metrics are provided in Section 4.4.1.

Naturally, the desired outcome is to have higher values for the traceback rate and lower values for the marking rate. Tables 5.1 and 5.2 and Figures 5.3 and 5.4 represent the trade-off between the traceback rate and the probability of selecting and marking the packets, P, on all six evaluation data sets for $K = 2$ and $K = 5$ (If $K = 2$, then the marking data uses 32 bits of the packets' header. If $K = 5$, then the marking

	Traceback Rate for K=2					
P	**CAIDA Oct2012**	**CAIDA Dec2012**	**CAIDA DDoS2007**	**MAWI Dec2012**	**CAIDA Mar2014**	**CAIDA Jul2014**
0.001	71.58	77.58	93.83	58.87	80.48	74.27
0.002	75.85	80.77	94.97	61.04	83.55	78.68
0.003	77.93	82.24	95.42	62.42	85.01	80.74
0.004	79.14	83.12	95.67	63.42	85.92	82.01
0.005	79.91	83.76	95.83	64.23	86.55	82.85
0.006	80.56	84.26	95.94	64.95	87.029	83.49
0.007	81.00	84.66	96.03	65.56	87.41	83.10
0.008	81.41	85.05	96.12	66.10	87.72	84.41
0.009	81.67	85.36	96.19	66.59	87.99	84.76
0.01	83.49	87.46	96.26	67.05	88.22	85.06
0.02	84.43	88.78	96.83	70.32	89.65	86.94
0.03	85.03	89.75	97.31	72.53	90.40	87.90
0.04	85.44	90.48	97.74	74.30	90.85	88.47
0.05	85.78	91.05	98.09	75.83	91.17	88.84
0.06	86.06	91.47	98.38	77.17	91.40	89.10
0.07	86.29	91.81	98.62	78.37	91.59	89.29
0.08	86.48	92.08	98.81	79.46	91.74	89.45
0.09	86.67	92.30	98.97	80.49	91.87	89.59
0.1	86.83	92.48	99.09	81.44	91.99	89.71
0.2	88.06	93.50	99.55	88.15	92.78	90.56
0.3	88.90	94.07	99.64	91.99	93.27	91.10
0.4	89.54	94.48	99.69	94.29	93.61	91.49
0.5	90.03	94.81	99.73	95.68	93.87	91.79
0.6	90.41	95.08	99.75	96.47	94.06	92.04
0.7	90.73	95.31	99.78	96.89	94.22	92.24
0.8	91.01	95.49	99.80	97.08	94.35	92.42
0.9	91.25	95.64	99.82	97.17	94.47	92.57
1	91.48	95.75	99.84	97.23	94.57	92.70

Table 5.1: The Trade-off between the Traceback Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 2$

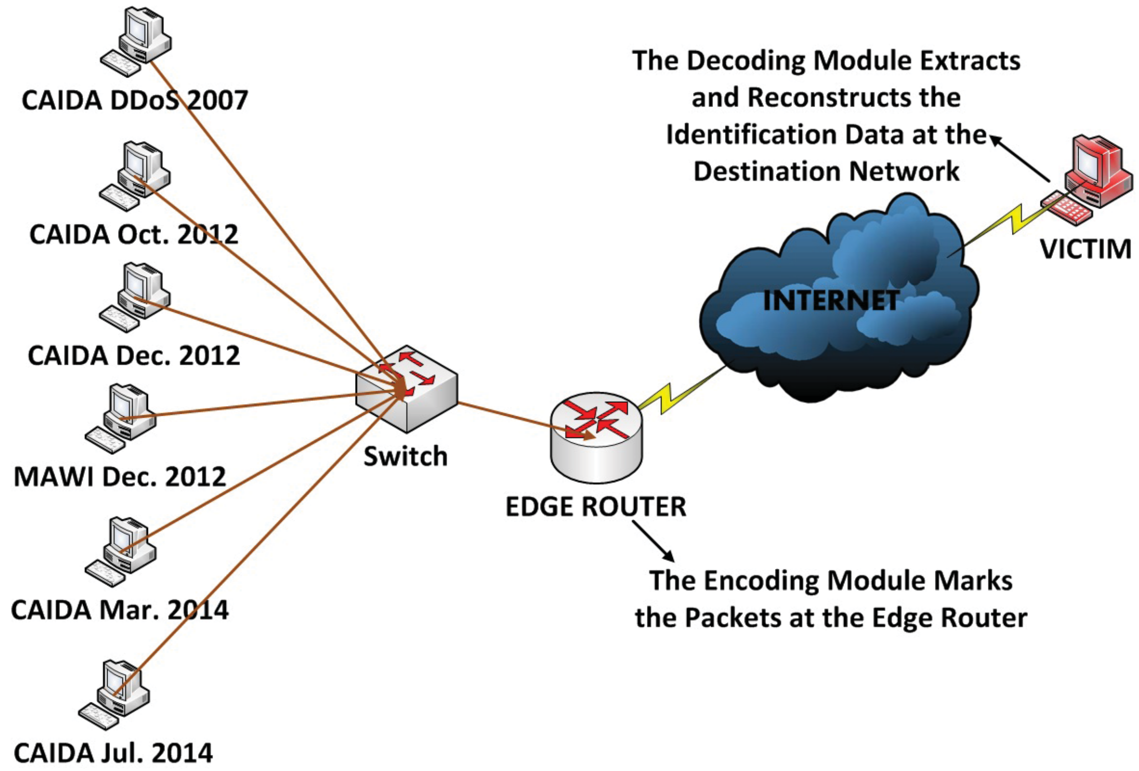

CAIDA DDoS 2007
CAIDA Oct. 2012
CAIDA Dec. 2012
MAWI Dec. 2012
CAIDA Mar. 2014
CAIDA Jul. 2014
Switch
EDGE ROUTER
The Encoding Module Marks the Packets at the Edge Router
INTERNET
VICTIM
The Decoding Module Extracts and Reconstructs the Identification Data at the Destination Network

Figure 5.2: The Testbed Network for Analyzing PFM

data uses 16 bits of the packets' header). The DFM (Section 4) and DPM (Section 3.2) approaches have been already evaluated twith the same evaluation data sets in Section 4.4.1, in Tables 4.5 and 4.6. In Table 5.3, those results are presented again in order to compare them with the new PFM approach.

The results show that as the probability of the marking is increased, the traceback rate increases (which is desirable), but increasing the marking rate itself is undesirable. In addition, as expected if the probability of the packet marking is set to 1 (marking all packets by the PFME module), the traceback rate of the PFM approach is exactly equal to the traceback rate of the DFM approach, because by default DFM deterministically marks all of the traffic flows. This can be seen clearly by comparing the results of $P = 1$ in Tables 5.1 and 5.2 with the results of evaluating the DFM approach in Table 5.3.

Furthermore, it was observed that although increasing P in the PFM approach increases the traceback rate, this increase is not linear. For example, when $K = 2$, Table 5.1, increasing the packet marking rate from 0.01 to 1 results in only about

	Traceback Rate for K=5					
	CAIDA	**CAIDA**	**CAIDA**	**MAWI**	**CAIDA**	**CAIDA**
P	**Oct2012**	**Dec2012**	**DDoS2007**	**Dec2012**	**Mar2014**	**Jul2014**
0.001	48.02	60.23	87.64	51.45	65.06	53.19
0.002	62.07	70.85	91.38	55.08	74.38	65.25
0.003	68.70	75.66	93.22	57.07	78.44	71.17
0.004	72.21	78.10	94.14	58.37	80.64	74.53
0.005	74.14	79.47	94.66	59.27	82.05	76.70
0.006	75.41	80.38	94.97	59.97	83.10	78.14
0.007	76.39	81.04	95.18	60.62	83.86	79.23
0.008	77.21	81.55	95.33	61.17	84.46	80.03
0.009	77.78	81.98	95.45	61.65	84.92	80.69
0.01	80.68	84.14	95.53	62.07	85.30	81.21
0.02	81.86	85.31	95.94	64.88	87.25	83.77
0.03	82.65	86.13	96.11	66.65	88.18	84.99
0.04	83.21	86.78	96.26	67.95	88.86	85.90
0.05	83.70	87.38	96.41	68.98	89.39	86.64
0.06	84.03	87.93	96.58	69.86	89.81	87.23
0.07	84.34	88.46	96.77	70.64	90.14	87.67
0.08	84.60	88.95	96.97	71.33	90.39	87.99
0.09	84.81	89.41	97.18	71.97	90.58	88.23
0.1	85.00	89.82	97.39	72.55	90.74	88.41
0.2	86.04	91.90	98.89	77.33	91.54	89.24
0.3	86.61	92.49	99.36	81.13	91.96	89.68
0.4	87.08	92.89	99.50	84.25	92.29	90.02
0.5	87.52	93.20	99.56	86.76	92.57	90.31
0.6	87.93	93.44	99.59	88.87	92.81	90.55
0.7	88.31	93.64	99.62	90.81	93.01	90.76
0.8	88.68	93.81	99.63	92.66	93.20	90.95
0.9	89.02	93.96	99.65	94.48	93.35	91.11
1	89.31	94.09	99.66	96.20	93.47	91.24

Table 5.2: The Trade-off between the Traceback Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 5$

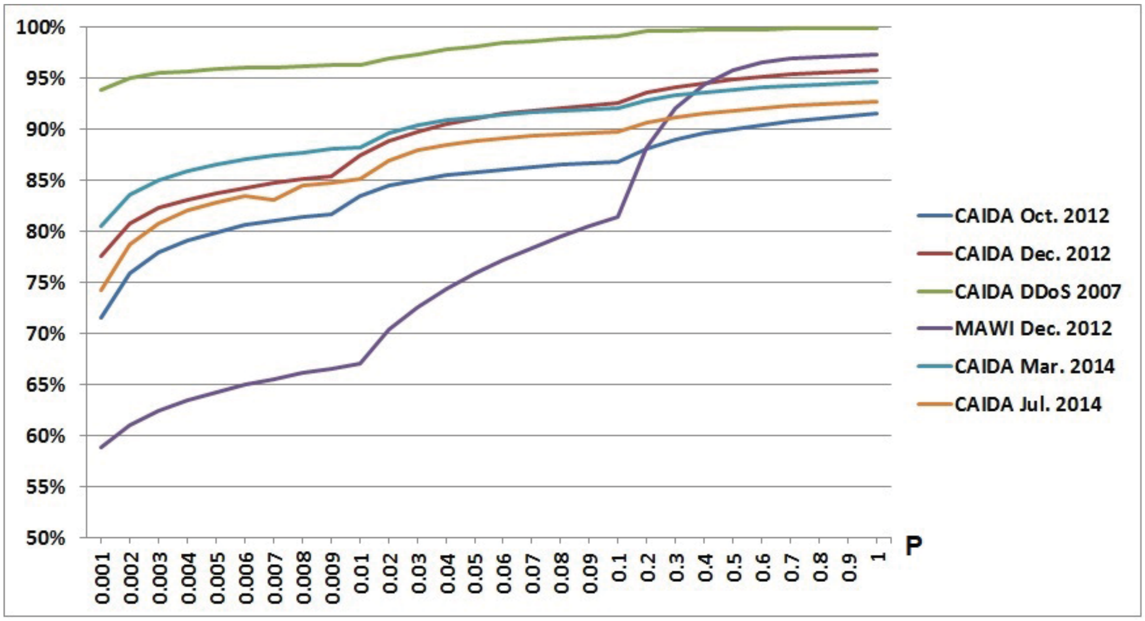


Figure 5.3: The Trade-off between the Traceback Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 2$

a 7% increase in the traceback rate for the attack-free data sets (except the MAWI data set), and about a 3.5% increase for the DDoS traffic. Also when $K = 5$, Table 5.2, increasing the packet marking probability from 0.01 to 1 results in only about a 10% increase in the traceback rate for the attack-free data sets (except the MAWI data set), and around a 4% increase for the DDoS traffic. This demonstrates that by selecting and marking the packets randomly, most times a significant decrease in the marking rate can result without losing much from the traceback rate (the above examples show that this approach can usually reduce the marking rate by 99% while losing only 3.5 to 10% of traceback rate. For the MAWI data set, the same results are obtained by reducing the marking rate by 70%).

The reason why the traceback rate of the PFM approach never reaches 100% or the reason why changing the K changes the TR is that to enable the PFMD module to traceback the source of all packets in a flow, at least K packets of that flow should be marked randomly by the PFME module. If the number of randomly-selected packets of a flow is less than K (it may happen if the number of packets in a flow is less than K, or if the marking probability, P, is low), then the source of that specific flow cannot be traced back by the PFMD module. It can also be concluded that if losing the marked packets results in arriving less than K marked packets to the destination

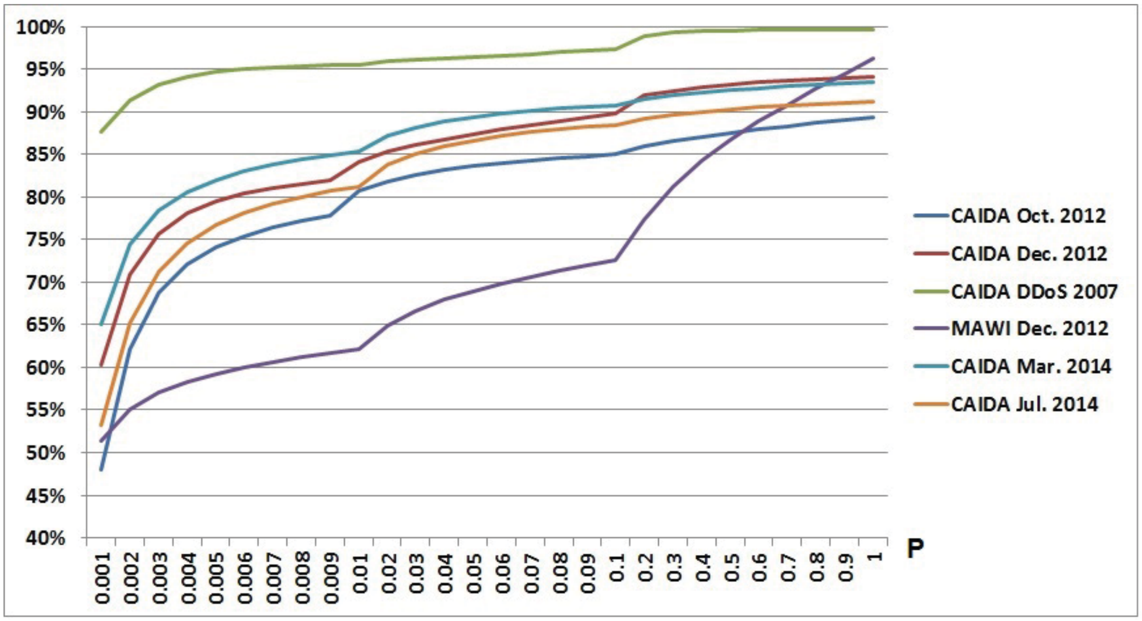


Figure 5.4: The Trade-off between the Traceback Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 5$

for a specific flow, that specific flow cannot be traced back. However, the victim can traceback to the attacker using the other marked flows from the same attacker.

	Traceback Rate			**Marking Rate**		
	DFM		**DPM**	**DFM**		**DPM**
Data Set	**K=2**	**K=5**		**K=2**	**K=5**	
CAIDA Oct2012	91.48	89.31	98.77	11.32	13.36	100
CAIDA Dec2012	95.75	94.09	99.13	6.24	7.91	100
CAIDA DDoS2007	99.84	99.66	99.88	0.58	1.03	100
MAWI Dec2012	97.23	96.20	99.42	8.60	16.70	100
CAIDA Mar2014	94.57	93.47	99.08	6.91	8.10	100
CAIDA Jul2014	92.70	92.24	99.03	9.11	10.48	100

Table 5.3: The Traceback and Marking Rates of the DFM and DPM Approaches on Six Evaluation Data Sets

5.2.2 Reducing the Marking Rate

As shown by the experimental results, the advantage of the PFM approach is that PFM can significantly decrease the marking rate without a significant decrease in the traceback rate. For example, looking at Table 5.1, decreasing the traceback rate by only 3% (from 95% to 92%) for the CAIDA December 2012 data set, the PFM approach reduces the marking rate from 100% to 10%. However, the weakness of PFM is that when the maximum traceback rate is required, the marking rate increases dramatically. For example, to increase the traceback rate by 8% (from 87% to 95%) for the same data set, the marking rate would be increased by 99%, from 1% to 100% (from $P = 0.01$ to $P = 1$). The reason is that the PFMD module at the destination network only needs to observe K marked packets (2 or 5 packets) per flow to infer the origin of all the packets in that flow, while the PFME module at the source-end does not check whether or not the number of randomly selected packets is more than the actual required marked packets.

To address this issue, the design of PFME module was modified to keep track of the number of marked packets in each flow. So, if the number of randomly selected and marked packets by the PFME module in each flow exceeds the number of required marked packets, the PFME module does not mark the next randomly selected packet from that specific flow. In doing so, the PFME module adds a sign bit for each identification data fragment in the FragTable. Once the PFME module selects an outgoing packet randomly, first, it extracts its Flow-ID and then checks for the existence of a table record for this Flow-ID in the FragTable. If this Flow-ID is not in the FragTable, it means that this flow is new to the PFME module, so the PFME module performs the following steps:

1. creating a table record for the new Flow-ID in the FragTable;

2. calculating the flow identification data;

3. cividing the flow identification data into K fragments and inserting them into the FragTable;

4. carking the selected packet by one of the identification data fragments;

5. changing the sign of the selected fragment to “1”.

However, if this Flow-ID is in the FragTable, it means that this flow has already been seen by the PFME module, so the PFME module marks only the selected packet where its sign is “0”, and changes its sign to “1”. If the sign bit of all fragments are already set to “1”, then the PFME module does not mark the selected packet.

This improved version of the PFM approach was evaluated on the same data sets and the same testbed network to analyze whether the performance would increase or not. Tables 5.4 and 5.5 and Figures 5.5 and 5.6 present the trade-off between the marking rate and the probability of selecting the packets, P, on all of the evaluation data sets. As can be seen, the improved PFM approach addresses the problem of the high marking rate of the original PFM approach, so now the marking rate is decreased dramatically. For example, to obtain a 90% traceback rate with K=2 in the CAIDA December 2012 data set, the marking rate is reduced from 4% in the PFM approach to only 0.67% in the improved PFM approach, Tables 5.1 and 5.4. Moreover, to obtain a 93% traceback rate with K=5 for the CAIDA December 2012 data set, the marking rate is reduced from 50% in the PFM approach to only 4.7% in the improved PFM approach, Tables 5.2 and 5.5.

Furthermore, comparing the performance of the DFM and the improved PFM approaches shows that it can decrease the marking rate significantly without any notable changes in the traceback rate by employing the new PFM scheme. For example, when $K = 2$, DFM requires to mark 58 packets out of every 10000 packets to gain a 99.84% traceback rate on the CAIDA DDoS attack 2007 data set. However, when employing the improved PFM approach, the number of marked packets can be decreased to less than a half, from 58 packets to 26 packets (out of every 10000 packets), and can still achieve a 99.09% traceback rate.

Finally, although the PFM approach shows a low marking rate and a high traceback rate on the attack-free data sets, the performance of the scheme on the CAIDA DDoS attack data set is even better. In this case, to have more than a 99% traceback rate on the DDoS traffic, less than 0.26% of the packets need to be marked for $K = 2$. This is much better than the results for the attack-free traffic. This happens because the average number of packets in a flow in the DDoS traffic is much more than the attack-free traffic. Since PFM does not mark the traffic with more than two or five packets per flow (depending on the value of K), having more packets in a flow results in the lower marking rate and the higher traceback rate. The average number of packets per flow for the CAIDA DDoS data set is 270.92, for the CAIDA October data set is 10.08 and for the CAIDA December data set is 19.07.

According to the experimental results, marking flows with a probability of $P = 0.1$

	Marking Rate for K=2					
	CAIDA	**CAIDA**	**CAIDA**	**MAWI**	**CAIDA**	**CAIDA**
P	**Oct2012**	**Dec2012**	**DDoS2007**	**Dec2012**	**Mar2014**	**Jul2014**
0.001	0.10	0.08	0.04	0.10	0.07	0.09
0.002	0.13	0.10	0.05	0.14	0.09	0.12
0.003	0.15	0.12	0.05	0.18	0.10	0.14
0.004	0.17	0.14	0.06	0.22	0.12	0.15
0.005	0.19	0.15	0.06	0.25	0.13	0.17
0.006	0.21	0.17	0.06	0.29	0.15	0.19
0.007	0.23	0.19	0.07	0.32	0.16	0.21
0.008	0.25	0.20	0.07	0.36	0.17	0.22
0.009	0.27	0.22	0.08	0.39	0.18	0.24
0.01	0.44	0.35	0.08	0.42	0.20	0.25
0.02	0.60	0.47	0.11	0.74	0.31	0.39
0.03	0.75	0.57	0.14	1.03	0.41	0.52
0.04	0.90	0.67	0.17	1.30	0.50	0.64
0.05	1.04	0.76	0.19	1.55	0.59	0.75
0.06	1.18	0.85	0.21	1.79	0.68	0.86
0.07	1.32	0.94	0.22	2.02	0.77	0.97
0.08	1.45	1.02	0.24	2.24	0.85	1.08
0.09	1.59	1.10	0.25	2.45	0.94	1.19
0.1	1.72	1.17	0.26	2.65	1.02	1.29
0.2	2.99	1.88	0.32	4.25	1.79	2.30
0.3	4.17	2.52	0.37	5.36	2.51	3.24
0.4	5.29	3.12	0.40	6.17	3.20	4.15
0.5	6.36	3.69	0.44	6.79	3.86	5.03
0.6	7.40	4.24	0.47	7.26	4.50	5.89
0.7	8.41	4.76	0.50	7.66	5.13	6.73
0.8	9.39	5.27	0.53	7.99	5.74	7.54
0.9	10.37	5.76	0.55	8.30	6.32	8.33
1	11.32	6.24	0.58	8.60	6.91	9.11

Table 5.4: The Trade-off between the Marking Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 2$

	Marking Rate for K=5					
	CAIDA	**CAIDA**	**CAIDA**	**MAWI**	**CAIDA**	**CAIDA**
P	**Oct2012**	**Dec2012**	**DDoS2007**	**Dec2012**	**Mar2014**	**Jul2014**
0.001	0.15	0.12	0.08	0.12	0.11	0.14
0.002	0.20	0.16	0.09	0.17	0.14	0.18
0.003	0.23	0.18	0.10	0.21	0.16	0.21
0.004	0.26	0.21	0.10	0.26	0.18	0.24
0.005	0.29	0.23	0.11	0.30	0.20	0.26
0.006	0.31	0.25	0.11	0.34	0.22	0.29
0.007	0.34	0.27	0.12	0.38	0.24	0.31
0.008	0.36	0.29	0.12	0.42	0.25	0.33
0.009	0.38	0.31	0.13	0.45	0.27	0.35
0.01	0.58	0.47	0.13	0.49	0.28	0.37
0.02	0.77	0.62	0.18	0.86	0.42	0.54
0.03	0.95	0.76	0.22	1.20	0.54	0.70
0.04	1.11	0.90	0.25	1.53	0.66	0.84
0.05	1.28	1.03	0.29	1.85	0.77	0.98
0.06	1.44	1.15	0.33	2.16	0.87	1.11
0.07	1.60	1.27	0.36	2.46	0.97	1.24
0.08	1.75	1.38	0.39	2.75	1.07	1.36
0.09	1.90	1.49	0.42	3.04	1.17	1.48
0.1	2.06	1.59	0.45	3.32	1.26	1.60
0.2	3.49	2.48	0.63	5.86	2.15	2.72
0.3	4.85	3.26	0.72	8.01	2.99	3.79
0.4	6.17	4.00	0.78	9.84	3.79	4.82
0.5	7.45	4.70	0.83	11.41	4.57	5.82
0.6	8.69	5.37	0.87	12.78	5.32	6.81
0.7	9.90	6.03	0.91	13.99	6.05	7.77
0.8	11.08	6.67	0.95	15.03	6.75	8.69
0.9	12.24	7.30	0.99	15.94	7.43	9.59
1	13.36	7.91	1.03	16.70	8.10	10.48

Table 5.5: The Trade-off between the Marking Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 5$

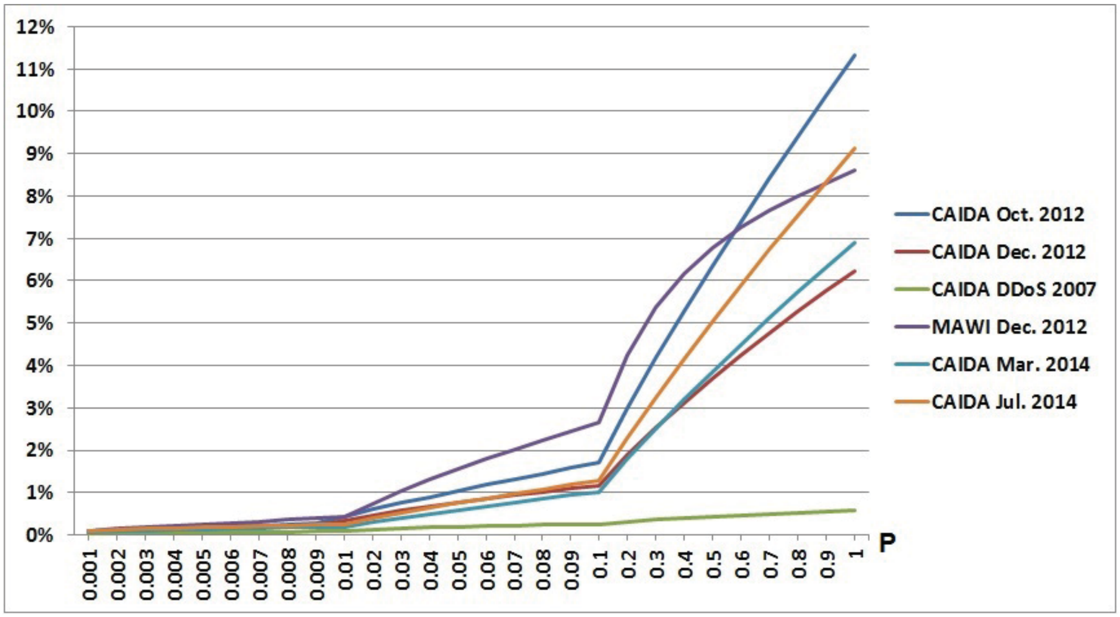


Figure 5.5: The Trade-off between the Marking Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 2$

is a good trade-off between the TR and MR, since in comparison with the DFM, $P = 0.1$ reduces the marking rate of PFM significantly without a major change in the traceback rate. Figures 5.7 and 5.8 compare the traceback rate and the marking rate of DFM and PFM when $K = 2$ and $P = 0.1$. Also, Figures 5.9 and 5.10 compare the traceback rate and the marking rate of DFM and PFM when $K = 5$ and $P = 0.1$. As an example, selecting $P = 0.1$ for the CAIDA December 2012 data set when $K = 2$, results in an 81.25% drop in the marking rate (dropping the number of selected packets from 624 to 117 out of every 10,000), while it only reduces the traceback rate by 3.27% (from 95.75% to 92.48%). However, the value of P should be set by the network administrators based on their expected performance of the employed IP-Traceback mechanism, as well as their internal policies.

5.2.3 Memory Usage

PFM has been designed with two separate modules, PFME and PFMD. Each of these modules has its own memory usage as described below.

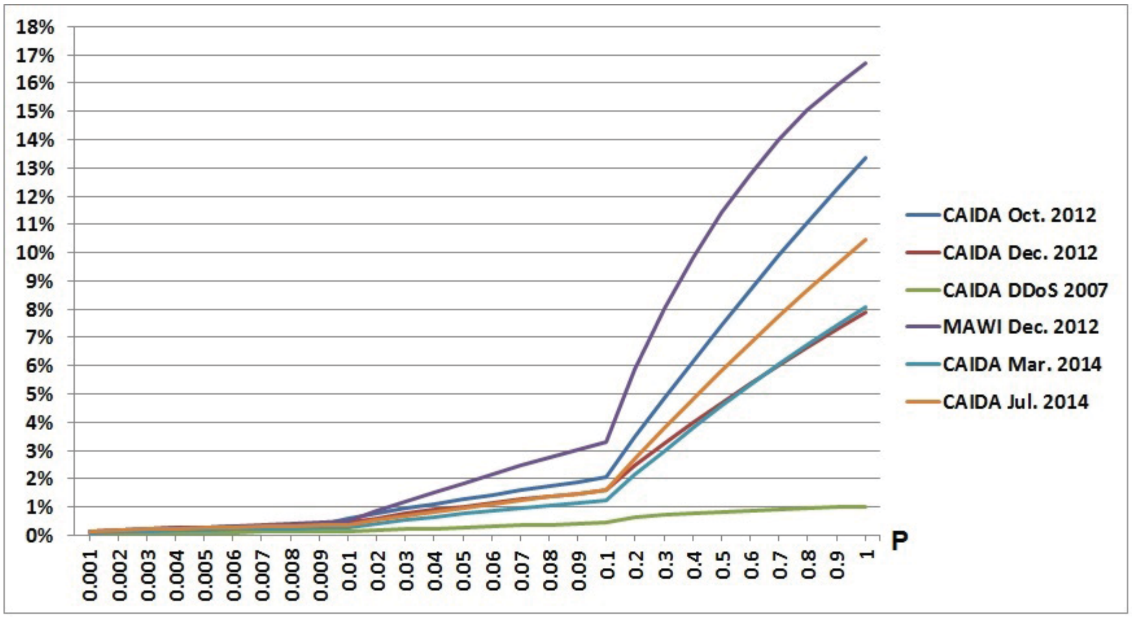


Figure 5.6: The Trade-off between the Marking Rate and the Probability of Selecting and Marking the Packets, P, on Six Evaluation Data Sets for $K = 5$

Memory Usage of the PFME Module at the Marking Router

The space required for running the UFME module on an edge router, $PFME_{mem}$, is equal to the sum of the space required by three tables namely the Marking-Table, the NI-ID-Table and the Node-ID-Table:

$$PFME_{mem} = NT + NDT + MT \tag{5.1}$$

where NT is the size of the NI-ID-Table, NDT is the size of Node-ID-Table and MT is the size of the Marking-Table.

Section 4.4.2 showed that equation 5.1 can be summarized to:

$$PFME_{mem} = 9N_i + 7.5N_{MAC} + 22.5N_{CF} \tag{5.2}$$

where N_i is the number of physical and virtual interfaces on the marking router, N_{MAC} is the number of observed unique MAC addresses still available in the Node-ID-Table, and N_{CF} represents the number of concurrent flows in the traffic.

Similar to the testbed network used for evaluating the DFM approach, Figure 4.3, the testbed network for evaluating the PFM approach has two interfaces (N_i), Figure 5.2, and the number of simulated source MAC addresses (N_{MAC}) is 105 as

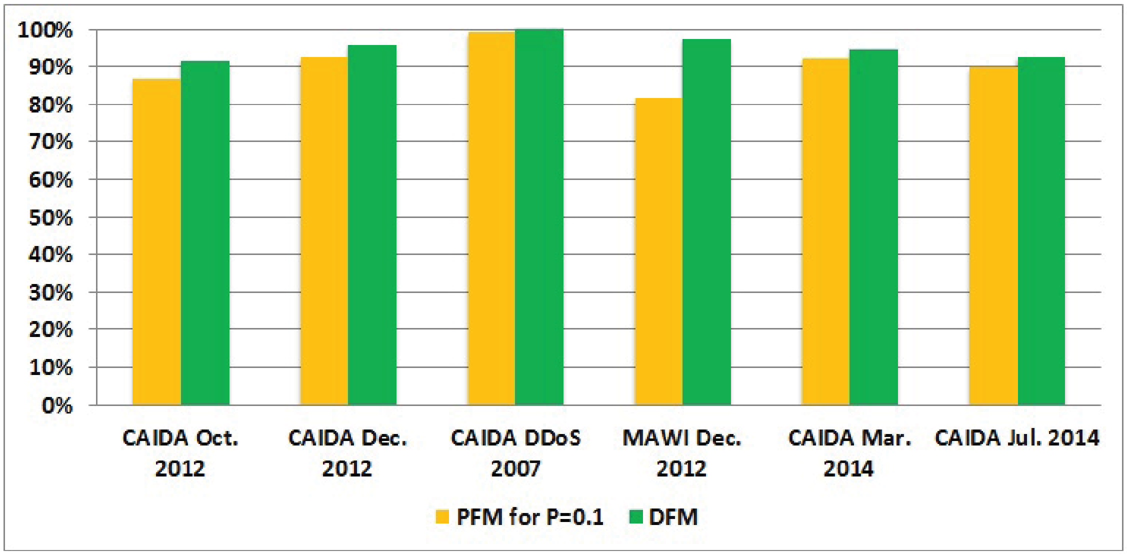


Figure 5.7: Comparison between the Traceback Rates of PFM for $P = 0.1$ and DFM for K=2

well. Therefore, in this experience the space required for running the PFME module is equal to the space required for running the DFME module, Table 4.7. As can be seen, the maximum required space for running the DFME module in the testbed network is 5.3 MB for the CAIDA July 2014 data set.

It should be noted here that the NI-ID and the Node-ID can be extracted from the ARP table of the marking router. In addition, Flow detection is an embedded feature in almost all manageable routers (eg. Cisco has NetFlow, InMon has sFlow, and Juniper uses JFlow). MAC filtering is also an embedded feature in the routers. Therefore implementing the PFME module has a minimum burden on the routers, as most of the implementation has already been done on the routers.

Memory Usage of the PFMD Module at the victim's side

The space required for running the PFMD module at the victim's side, $PFMD_{mem}$, is equal to the space required for the Reconstruction-Table. As described above, the PFMD module maintains the Reconstruction-Table for matching the Flow-ID and the K possible identification data fragments of a flow in order to reconstruct valid source identification data. For every record in the ReconTable, a Flow-ID and K possible identification data fragments should be stored. The victim no longer needs to keep the record of a flow when it is over. The $PFMD_{mem}$ is equal to $DFMD_{mem}$. In Section

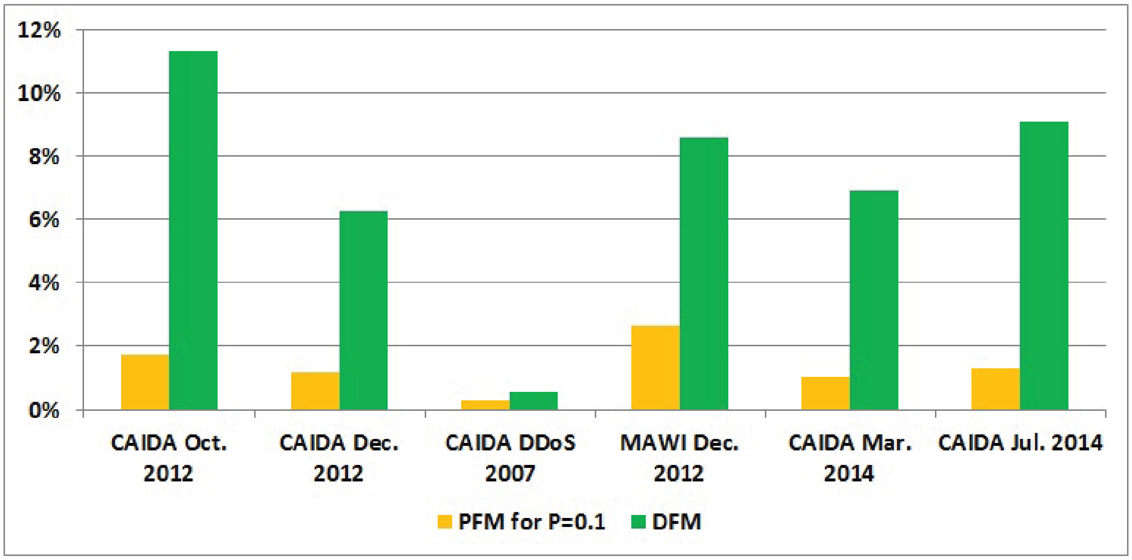


Figure 5.8: Comparison between the Marking Rates of PFM for $P = 0.1$ and DFM for K=2

4.4.2 it was shown that the required space for $DFMD_{mem}$ can be summarized to:

$$DFMD_{mem} = N_{CF} \times (13 + (K \times NB)) \tag{5.3}$$

where K is the number of mark fragments (can be either 2 or 5), and NB is the total number of bits that are embedded in the IP header of each mark-carrying packet. For $K = 2$, the NB would be 32 and for $K = 5$, the NB would be 16. As can be seen from Table 4.7, the maximum required space for running the PFMD module in the testbed network is 4.74 MB for the CAIDA July 2014 data set.

It should be noted here that the ReconTable can be placed on the destination host, destination router, or another host in the destination network.

5.3 Authenticated Flow Marking

The authenticated flow marking for the PFM approach follows the same authentication algorithm already introduced for the DFM approach in Section 4.5. The authenticated flow marking verifies that the victim is talking directly to the marking router that the victim thinks it is talking to and ensures that the marking data is delivered exactly as intended. The experimental results of evaluating the authenticated flow marking are presented in Section 4.6 and Table 4.9.

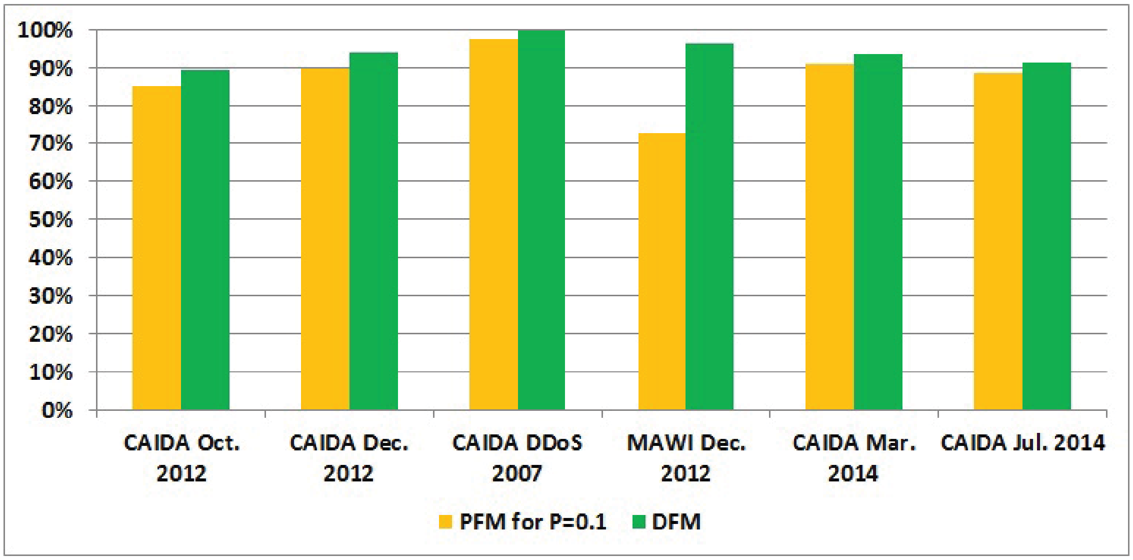


Figure 5.9: Comparison between the Traceback Rates of PFM for $P = 0.1$ and DFM for K=5

5.4 Summary

In this chapter, a new IP-Traceback method, called Probabilistic Flow Marking (PFM) was presented. The motivation was to decrease the marking rate of the DFM approach while keeping the traceback rate as high as possible. This has been performed by selecting the packets randomly and marking them based on the flows they belong to. PFM consists of two modules, the PFM encoding module, PFME, and the PFM decoding module, PFMD. The PFME module runs at the egress interface of the marking router, and selects and marks the packets randomly. The PFMD module runs at the destination network and tries to infer the source of the traffic by analyzing the marked packets and extracting the source identification data from them. If the probability of selecting and marking the packets, P, is set to 1, then PFM works like DFM in terms of traceback and marking rates. However, results show that if P is reduced, then PFM can decrease the number of selected and marked packets up to 99% without considerable change in the traceback rate. It has been demonstrated as well that PFM does not need to mark the traffic with more than two or five packets per flow. Therefore, having more packets in a flow results in a lower marking rate and a higher traceback rate. As the average number of packets per flow in the DDoS attack data set is much more than in the attack-free data sets, the performance of

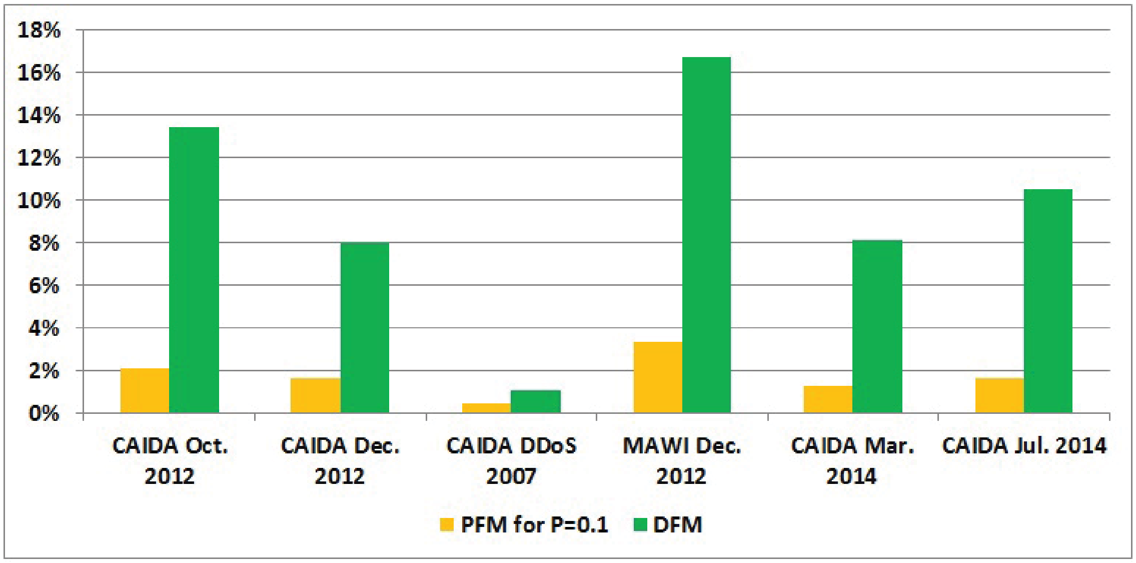


Figure 5.10: Comparison between the Marking Rates of PFM for $P = 0.1$ and DFM for K=2

the proposed PFM on the DDoS data set is far better than even on the attack-free data sets.

It should be noted that since the PFM approach inherits most of its fundamental characteristics from the DFM approach, all analyses described in Section 4.7 for the DFM approach are also valid for the PFM approach.

Chapter 6

Unique Flow Marking: UFM

This chapter proposes a new approach to IP-Traceback called Unique Flow Marking (UFM) to improve upon the previous work on Deterministic Flow Marking (DFM) 4. Although DFM shows good traceback performance such as high traceback rate, low marking rate and low bandwidth usage, this performance is achieved by marking all the flows in the traffic. DFM marks every flow at the attacker side, and the victim needs to traceback every single flow. To reduce this overhead at the victim's side and on the edge router at the attacker side, Unique Flow Marking (UFM) IP-Traceback approach is proposed. Similar to DFM, UFM is an IP-Traceback approach which aims to find the origin of network traffic, even if the source IP address of the packets is hidden by a proxy, or a NAT device or spoofing. However, unlike DFM which marks every flow, UFM marks only the unique flows in the network traffic.

The main concept that leads us to the UFM approach is that it has been noticed that usually there are some flows in the real traffic traces that share the same Flow-ID. In other words, TCP/UDP flows share the same five tuples of information: source IP address, destination IP address, L4 protocol type (TCP/UDP), source port and destination port numbers. Additionally, ICMP flows share the same six tuples of information: source IP address, destination IP address, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID. If one of these flows at the source edge router can be found and marked, then the victim is able to traceback all of the packets in all of the flows that share the same 5 or 6 tuple information, i.e. the Flow-ID. In this case, UFM may reduce the processing and memory overheads of IP-Traceback at both sides of the source and the destination networks. In fact, unlike the DFM which marks every flow by embedding the source identity data in the first several packets of each flow, UFM aims to find and mark only one flow out of all the flows that have the same Flow-ID. In comparison with the DFM and the packet-based traceback approaches such as DPM (Section 3.2), this method aims to minimize the number of marked flows and maximize the traceback rate. Indeed, there is a trade-off between the marking rate and the traceback rate in such techniques. To this end, a sensitivity analysis was done on this trade-off and it was compared with the existing techniques in the literature including DFM.

Based on the various IP-Traceback approaches described in Chapter 2, UFM falls into the following categories: Basic principle-Marking; Processing mode-Deterministic at flow level; Location-Source group. Similar assumptions were made for the UFM approach to those introduced for the DFM approach in Section 4.1.

6.1 UFM Modules

UFM has been designed with two modules, one in the UFM Encoding module (UFME), which runs on an edge router near the attacker network, and the other one is the UFM Decoding module (UFMD), which runs at the victim-end. The following sections describe these two modules in detail.

6.1.1 UFM Encoding Module (UFME)

This module is located on the edge routers and its goal is to mark the flows, Figure 6.1. In UFM, only the edge routers mark the packets, and the others, including the core routers, are not involved in the marking process. As with the DFME module (4.2.1) and the PFME module (5.1.1), the UFME module uses three identifiers to mark the packets in order to trace up to the attacker node. These three identifiers are described below.

- The IP address of the egress interface of the edge router (32 bits).
- The network interface identifier, NI-ID, (12 bits): This 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. The NI-ID specifies from which subnet a traffic flow comes.
- Node-ID (16 bits): An identifier assigned to each source MAC address observed on the incoming traffic from local networks. Each MAC has a unique Node-ID.

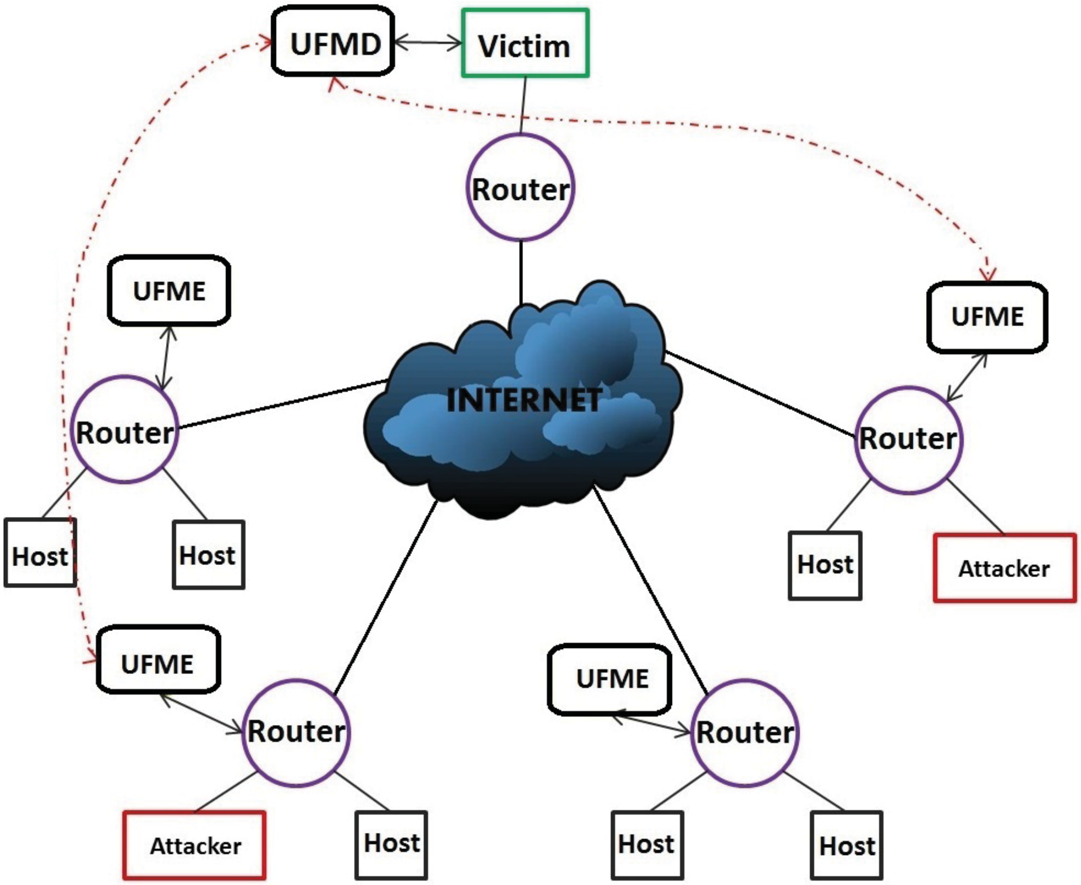


Figure 6.1: The Locations of the UFMD and UFME Modules on a Sample Network. The UFMD Module at the Victim end is in Communication with the UFME Module on the Edge Routers of the Attacking Networks

The UFM identification data consists of the IP address of the egress interface (32 bits) + the NI-ID (12 bits) + the Node-ID (16 bits) = 60 bits, to distinguish the traffic of a particular node from the other nodes. The UFME module at the source side marks each unique flow (in terms of its Flow-ID). In doing so, the identification data of a flow should be divided into K fragments, and the first K packets of the flow should carry these fragments. Once a flow is marked, then any other flow with the same Flow-ID does not need to be marked. Therefore, each selected packet carries $M = 60/K$ bits of the identification data and $S = log2(K)$ bits required to identify a fragment of the identification data. UFM also embeds a one-flag bit, F, in each selected packet to distinguish between marked and unmarked packets. Therefore, each identification data fragment, by which each selected packet should be marked, consists of: M bits for the identification data fragment, S offset bits to represent 2^S

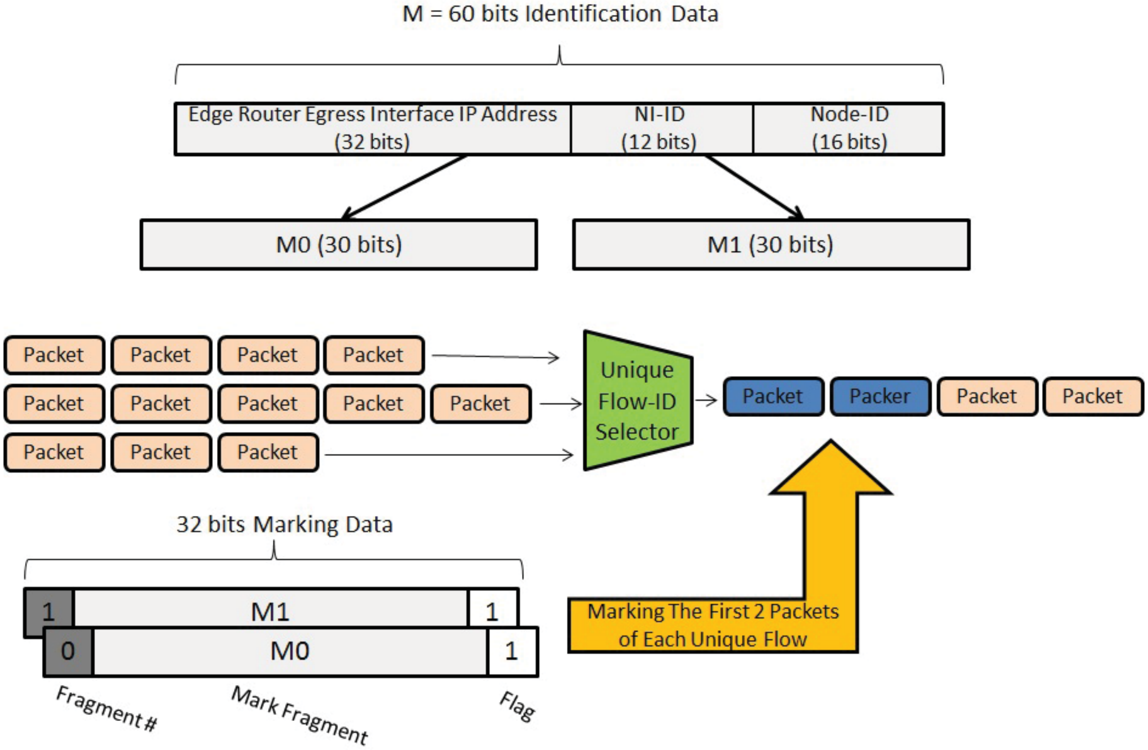


Figure 6.2: An Example of Marking a Unique Flow by the UFME Module. In this Example, $K = 2$, $M = 30$ bits, $S = 1$ bit, and flag=1 bit

possible fragments and a one-bit flag F, which should be set to "1" for the marked packets and "0" for the rest, Figure 6.2.

The UFME module maintains a table to keep track of the marked packets, their Flow-IDs, and their embedded marks. This table is called the FragTable. Once the UFME module observes an outgoing packet, it first extracts its Flow-ID and then looks for the existence of a table record for this Flow-ID in the FragTable. If this Flow-ID is not in the FragTable, it means that this flow is new to the UFME module, so the UFME module takes the following steps:

- creates a table record for the new Flow-ID in the FragTable;
- calculates the flow identification data;
- divides the flow identification data into K fragments and inserts them into the FragTable;
- marks the selected packet with one of the identification data fragments.

However, if this Flow-ID is already in the FragTable, it means that this flow has been already seen by the UFME module. In this case, if all of the identification data fragments of this flow have already been sent, then the UFME module just forwards the packets. Otherwise, the UFME module marks the packet with one of the unsent fragments in the FragTable and then forwards it.

As described above, the identification data (60 bits) should be divided into K fragments, and each fragment contains $M = 60/K$ bits of the identification data, $S = log2(K)$ bits to address each fragment, and a 1-bit flag to differentiate between the marked and the unmarked packets. Based on the results in Section 4.4, the best value of K is either 2, making the size of each fragment 32 bits, or 5, making the size of each fragment 16 bits. As shown in Section 4.4.1 and Figure 4.5, it is possible to use the identification and flag fields of the IP header as a 16-bit marking field, or to use the identification, flag and fragment offset fields of the IP header as a 32-bit marking field. Selecting a value of K between 2 or 5 is a trade-off between the marking rate and the number of required marking bits in the IP header. Selecting the lower value of K results in a lower marking rate (which is desirable) but needs more space in the IP header (which is undesirable). If K=5 is selected, the UFME module uses 16 bits of the IP header to mark the selected packets, but if K=2 is chosen, then the UFME module marks 32 bits of the IP header.

6.1.2 UFM Decoding Module (UFMD)

Similar to the DFME (4.2.1) and PFME (5.1.1) modules, the UFMD module is located at the destination network and its goal is to infer the origin of the incoming traffic, even if the source IP addresses of the incoming packets are spoofed, Figure 6.1. This module maintains a table for matching the Flow-ID and K possible identification data fragments of a flow, called ReconTable, in order to reconstructs valid source identification data. The UFMD module checks for the flag field (which is set by the UFME module) in the packet's IP header to find the marked packets. Once the UFMD module finds a marked packet, it extracts its Flow-ID and then checks for the existence of a table record for this Flow-ID in the ReconTable. If this Flow-ID is not in the ReconTable, it means that this flow is new to the UFMD module. The UFMD module creates a table record for the new Flow-ID in the ReconTable and inserts the extracted identification data fragment into the ReconTable. However, if this Flow-ID was already in the ReconTable, it means that this flow has been already seen by the UFMD modules, so the UFMD module only inserts the extracted identification data fragment into the ReconTable. The order of different fractions of the same source identification data is recognizable to the UFMD module by the fragment number field which has S bits (for K=5, S is 3 bits and for K=2, S is 1 bit) in the packet's IP header.

Once all the fractions of the same flow source identification data get to the UFMD module, the origin of all the packets in that flow, as well as the origin of all the packets in any other flow with the same Flow-ID is apparent to the victim, even if the source IP address of those packets is hidden in one shape or form as discussed earlier. Using UFMD, the destination is able to distinguish the traffic of different nodes behind an edge router. As a result, when an abnormal traffic is observed, the victim is able to distinguish between the attack and the legitimate traffic and infer the source of an attack, even if it is behind a NAT or a proxy device.

6.2 Experimental Results

To evaluate the proposed UFM approach and compare it with the DFM (Section 4) and DPM (Section 3.2) approaches, the same six evaluation network traces were used that were already used for evaluating the DFM, PFM and DPM approaches, including the MAWI (Measurement and Analysis on the WIDE Internet) traffic archive December 2012 [8], the CAIDA DDoS attack 2007 [17], the CAIDA anonymized Internet traces October 2012, December 2012 [14], March 2014 and July 2014 [15] data sets. The description of these data sets are presented in Section 4.4. The data sets employed in this work are real-life tcpdump files and are available publicly. Table 4.4 represents statistical information of all aforementioned network traffic tracess.

A testbed network was implemented in the research lab similar to the PFM testbed network, Figure 5.2. As shown in this figure, the aforementioned traces are replayed from the local area network behind the edge router and directed to the destination. For this purpose, the tcpreplay and tcprewrite open source applications were used [1]. In addition, two real-time programs were implemented using the Winpcap library by C++ [12], one for implementing the UFME module to mark the packets, and the

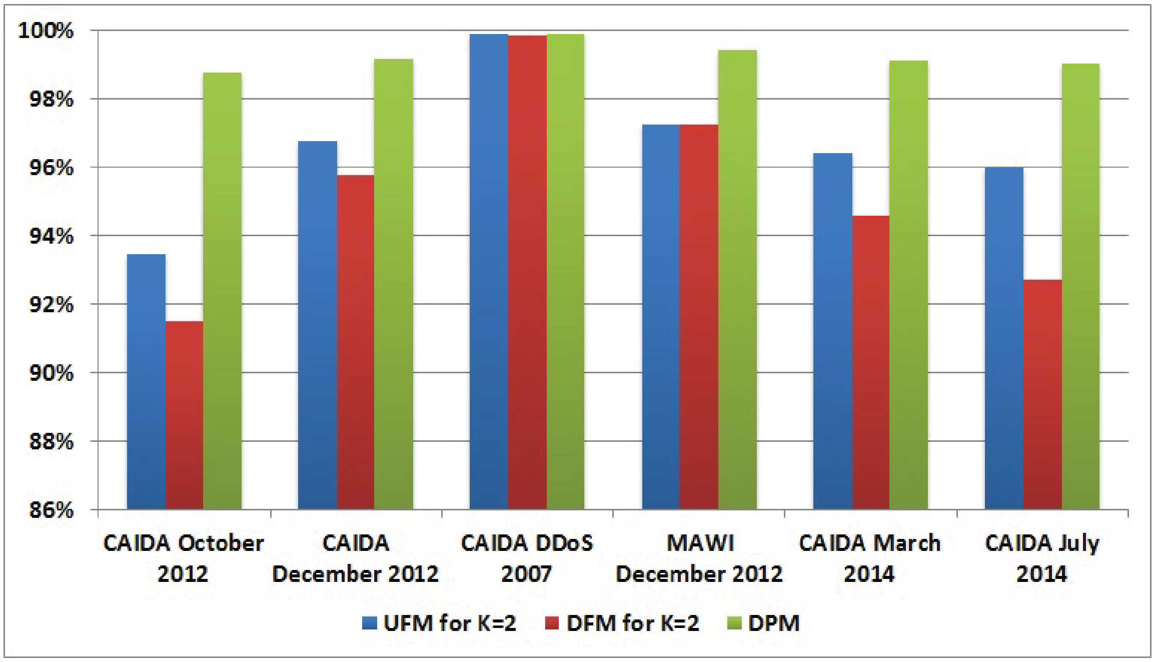


Figure 6.3: Comparison among DPM, DFM and UFM Approaches in Terms of Traceback Rate for $K = 2$

other for implementing the UFMD module to traceback the source of the packets. The UFME module runs at the egress interface of the source edge router and marks only the outgoing packets. At the same time, the traceback program runs at the destination network and aims to trace the origin of the marked traffic.

6.2.1 Traceback and Marking Rates

Two important metrics for evaluating a traceback method are the Traceback Rate, TR, and the Marking Rate, MR. The definition of these two metrics are provided in Section 4.4.1.

Obviously, the desired outcome is to have higher values for TR and lower values of MR. As described above in Section 4.4.1, the best value for K (the best number of fragments for each flow identification data) is either 2 or 5. The UFM approach has been evaluated by both of these two best values of K, using all six evaluation data sets.

Table 6.1 and Figures 6.3, 6.4, 6.5, 6.6 present the comparison among the UFM and the other approaches, using TR and MR metrics, on all six evaluation data sets. As can be seen, DPM has a higher traceback rate compared with UFM and DFM,

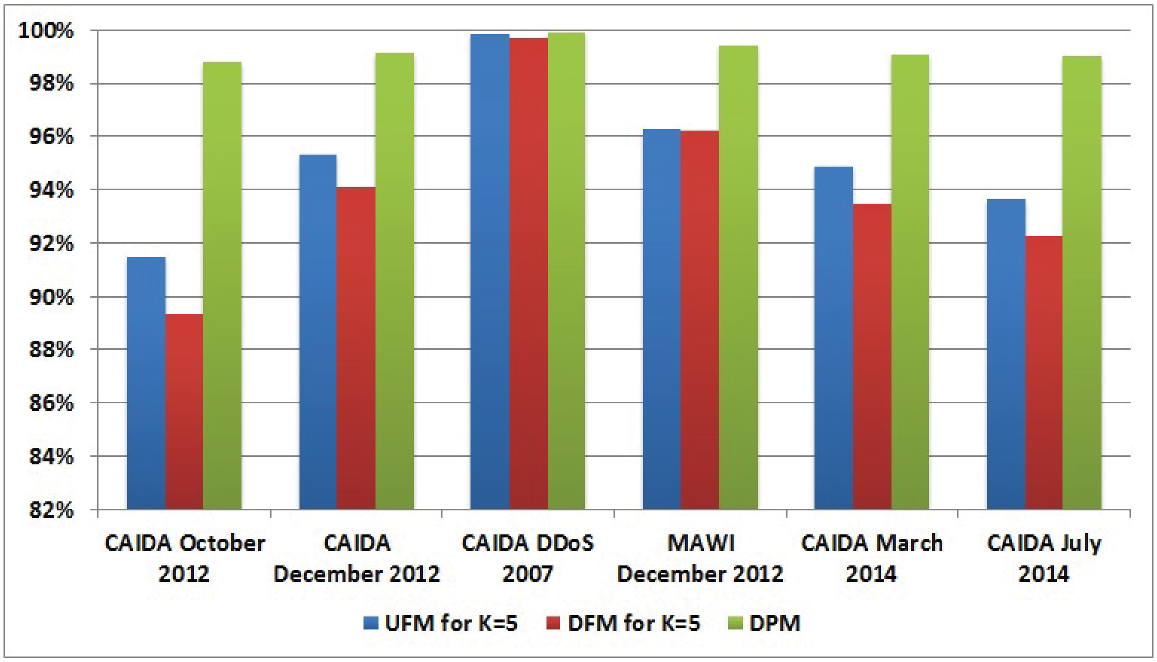


Figure 6.4: Comparison among the DPM, DFM and UFM Approaches in Terms of Traceback Rate for $K = 5$

but this accuracy is achieved by marking all packets in the network ($MR = 100\%$), which is not acceptable. In contrast to DFM, UFM increases the traceback rate and decreases the marking rate on all six evaluation data sets without exception. The best results are gained from the CAIDA July 2014 data set for $K = 2$, where UFM improved the Traceback Rate by 3.28% while the Marking Rate is decreased by 2.67%.

			CAIDA Oct2012	CAIDA Dec2012	CAIDA DDoS2007	MAWI Dec2012	CAIDA Mar2014	CAIDA Jul2014
TR	K=2	UFM	93.44	96.76	99.86	97.23	96.38	95.98
		DFM	91.48	95.75	99.84	97.23	94.57	92.70
	K=5	UFM	91.45	95.34	99.82	96.27	94.87	93.64
		DFM	89.31	94.09	99.66	96.20	93.47	92.24
		DPM	98.77	99.13	99.88	99.42	99.08	99.03
MR	K=2	UFM	8.97	4.64	0.31	8.45	5.20	6.44
		DFM	11.32	6.24	0.58	8.60	6.91	9.11
	K=5	UFM	10.39	5.50	0.89	16.34	6.15	8.12
		DFM	13.36	7.91	1.03	16.70	8.10	10.48
		DPM	100	100	100	100	100	100

Table 6.1: Comparison among the DPM, DFM and UFM Approaches in Terms of Traceback and Marking Rates

Moreover, as expected, $K = 2$ obtains better results than when $K = 5$. However, as described in Section 4.4.1, the required number of bits in the IP header for $K = 2$ is 32 bits. This is two times more than when $K = 5$ is set. The results confirm that UFM provides better performance than the other approaches.

It should be noted that if any of the marked packets get lost and does not arrive to the destination, the specific flow that the lost marked packet belongs to, and all other flows that share the same flow ID with the lost packet cannot be traced back. However, the victim can traceback to the attacker using the other flows from the same attacker.

6.2.2 Memory Usage

UFM has been designed with two separate modules, UFME and UFMD. Each of these modules has its own memory usage as described below.

Memory Usage of the UFME Module at the Marking Router

Section 4.4.2 shows that the space required for running the UFME module on an edge router, $UFME_{mem}$, can be summarized to:

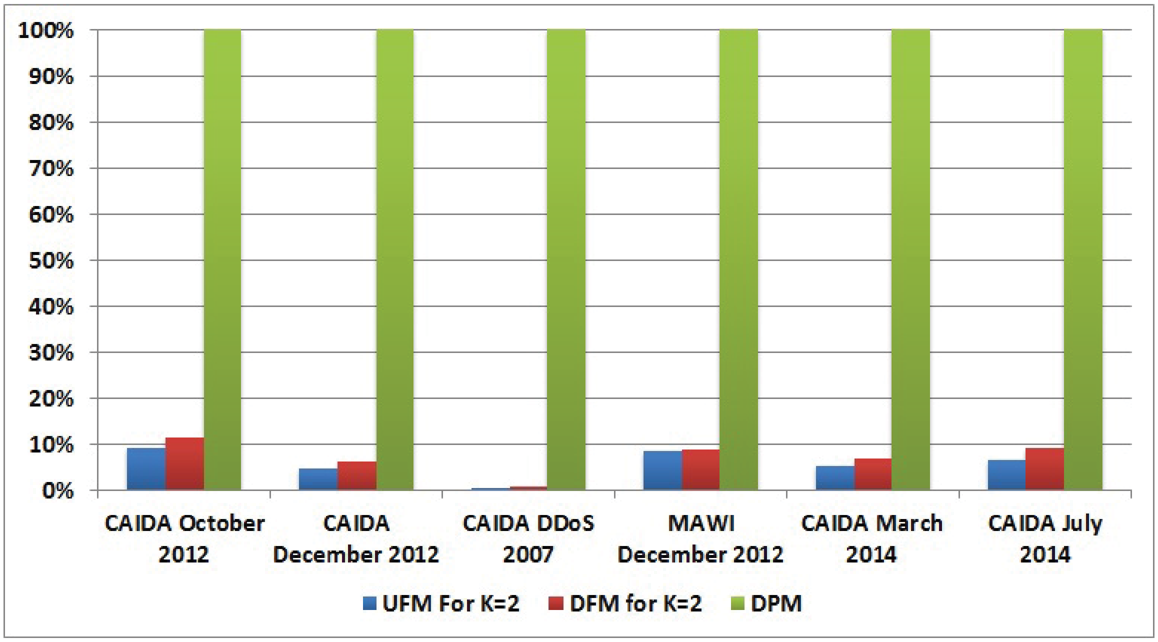


Figure 6.5: Comparison among DPM, DFM and UFM Approaches in Terms of Marking Rate for $K = 2$

$$UFME_{mem} = 9N_i + 7.5N_{MAC} + 22.5N_{CF} \tag{6.1}$$

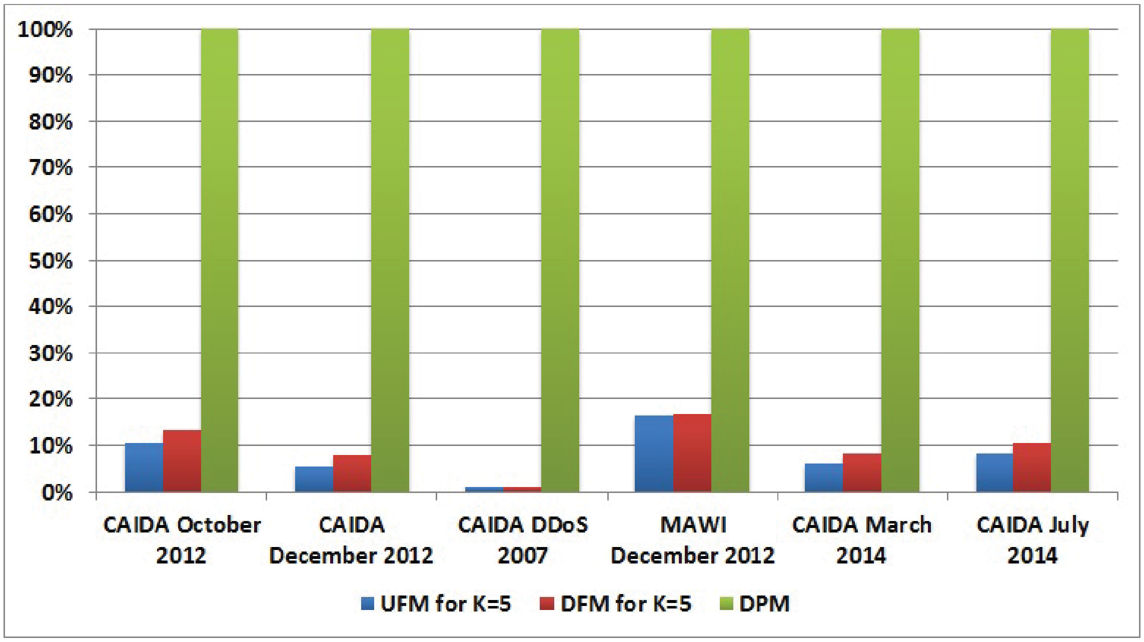


Figure 6.6: Comparison among DPM, DFM and UFM Approaches in Terms of Marking Rate for $K = 5$

where N_i is the number of physical and virtual interfaces on the marking router, N_{MAC} is the number of observed unique MAC addresses still available in the Node-ID-Table, and N_{CF} represents the number of concurrent flows in the traffic.

Similar to the testbed network used for evaluating the DFM approach, Figure 4.3, the testbed network for evaluating the UFM approach has two interfaces (N_i), Figure 5.2, and the number of simulated source MAC addresses (N_{MAC}) is 105 as well. Therefore, in this experience the space required for running the UFME module is equal to the space required for running the DFME or PFME modules, Table 4.7.

Memory Usage of the UFMD Module at the Destination-end

The space required for running the UFMD module at the victim's side, $UFMD_{mem}$, is equal to the space required for the Reconstruction-Table. In Section 4.4.2 it is shown that the required space for $UFMD_{mem}$ can be summarized to:

$$UFMD_{mem} = N_{CF} \times (13 + (K \times NB)) \tag{6.2}$$

where K is the number of mark fragments (can be either 2 or 5), and NB is the total number of bits that are embedded in the IP header of each mark-carrying packet. For $K = 2$, the NB would be 32 and for $K = 5$, the NB would be 16. Therefore, in this experience the space required for running the UFMD module is equal to the space required for running the DFMD or PFMD modules, Table 4.7.

Null Hypothesis

In inferential statistics, the term “null hypothesis” refers to a general statement that there is no relationship between two measured phenomena, or no difference among groups [52]. Rejecting or disproving the null hypothesis, and thus concluding that there are grounds for believing that there is a relationship between two phenomena, gives a precise criterion for rejecting a hypothesis [9].

There are two different tests that can be used for rejecting or disproving the null hypothesis. The first one is called the “Student’s t-test” and is most commonly applied when the test statistic would follow a normal distribution. The second one is called the “Mann-Whitney U test” and unlike the t-test it does not require the assumption of normal distributions.

Both tests were run on the traceback and marking rates of the DPM, DFM and UFM approaches to prove whether there is a relationship between these approaches or whether they are significantly different, Table 6.2. A returned value of less than 5% indicates that the test rejects the null hypothesis at the 5% significance level (evidence of a statistically significant difference). By contrast, a returned value greater than 5% indicates that the test does not reject the null hypothesis at the 5% significance level (evidence of no statistically significant difference). It should be noted that the results of the t-test in Table 6.2 are based on the assumption that the populations have unequal variances.

		K=2		**K=5**	
		TR	MR	TR	MR
DPM vs. DFM	T-test	2.52	1.76	2.07E-06	1.9E-05
	U-test	4.11	4.11	0.21645	0.21645
DPM vs. UFM	T-test	2.90	1.81	8.75E-07	1.23E-05
	U-test	4.11	4.11	0.21645	0.22
DFM vs. UFM	T-test	39.40	52.68	47.65	73.25
	U-test	32.90	39.39	30.95	69.91

Table 6.2: The T-test and U-test results on the DPM, DFM and UFM approaches.

As can be seen, the results for “DPM vs. DFM” and “DPM vs. UFM” are all less than 5%, indicating that there is statistically significant difference between DPM and the proposed DFM and UFM approaches. On the other hand, as expected, all results for “DFM vs. UFM” are greater than 5%. This indicates that there is no statistically significant difference between DFM and UFM. As mentioned above, UFM is derived from the DFM approach which improves its traceback rate and reduces its marking rate.

6.3 Authenticated Flow Marking

The authenticated flow marking for the UFM approach follows the same authentication algorithm already introduced for the DFM approach in Section 4.5. The authenticated flow marking verifies that the victim is talking directly to the marking router that the victim thinks it is talking to and ensures that the marking data is delivered exactly as intended. The experimental results of evaluating the authenticated flow marking are presented in Section 4.6 and Table 4.9.

6.4 Summary

In this chapter, a new IP-Traceback approach was presented called Unique Flow Marking, UFM. The main concept behind UFM is to find and mark only some of the packets of a flow among those that have the same Flow-ID. Once the source of some packets in a flow can be found, the source of any other packets in that flow, as well as the source of any other packets in any other flows with the same Flow-ID can also be found. UFM utilizes this fact to decrease the marking rate and to increase the traceback rate (compared to other approaches) simultaneously. UFM consists of two modules, the UFM encoding module, UFME, and the UFM decoding module, UFMD. UFME runs at the egress interface of the edge router at the source network and aims to mark some packets of one flow out of all the flows that have the same Flow-ID. UFMD runs at the destination network and aims to infer the source of traffic by analyzing the marked packets and extracting the source identification data from them.

In should be noted that since the UFM approach inherits most of its fundamental characteristics from the DFM approach, all analyses described in Section 4.7 for the DFM approach are also valid for the UFM approach.

Chapter 7

Deterministic Flow Marking for IPv6: DFM6

On the parts of the Internet where network growth is still continuing, there is more and more usage of IPv6 [5]. The IETF proposed using IPSec as an integral part of IPv6 when formulating the IPv6 standards [20]. A common myth of IPv6 is that IPv6 is secure because IPsec is mandated. However, this is not really true:

- Although the IETF mandates that all IPv6 nodes have IPsec available, the actual use of IPsec is optional.
- If all communications between two IPv6 nodes are encrypted then the network (which is usually trusted because it is centrally managed) becomes blind and cannot inspect the traffic or enforce a security policy.

In short, IPsec on IPv6 is usually reserved for the same cases as in IPv4: remote access virtual private networks (VPNs) or site-to-site VPNs. Therefore, spoofing IPv6 addresses is as easy as spoofing IPv4 addresses [7].

IP-Traceback is an alternative mechanism which aims to identify the true source of an IP datagram. However, as many current IP-Traceback schemes are designed for IPv4 networks, they cannot be used directly in IPv6 networks. Implementing those techniques for IPv6 networks requires modifications because of technological differences, such as those in the IP header.

To address this problem, a new traceback approach under IPv6 is proposed (DFM6) which helps network administrators to traceback actively and effectively to the source of attacks in a short time when suffering from an attack with spoofed source IP addresses. The proposed architecture is derived from Deterministic Flow Marking (DFM) 4, which is an IP-Traceback technique used for IPv4 networks. DFM has two unique features which have lead to the proposed IPv6 traceback approach. First, unlike other marking methods which mark the traffic at the packet level, DFM marks the traffic at the traffic flow level. This feature gives DFM a low marking rate. It requires a small amount of marked packets to find the source of the DDoS attack at the victim's side. Second, it can traceback up to two levels behind the marking router, which most of the times enabling it to determine not only the attacking network, but also the attacker node behind the network address translation (NAT) or proxy devices. In order to verify the validity of this proposed algorithm, DFM6 was evaluated using the CAIDA IPv6 data set under an IPv6 environment network and comparing the results with four other deterministic marking IPv6 traceback approaches.

7.1 DFM for IPv6 Networks: DFM6

The aim is to design DFM6 with enough flexibility so that it can be used to find the source of any kind of IPv6 traffic. This led to the assumptions listed below.

1. Attackers may generate any IPv6 packet.
2. Attackers may be aware they are being traced.
3. Attackers may spoof the source MAC and IPv6 addresses.
4. Packets may be lost or reordered.
5. An attack may consist of just a few packets.
6. Packets of an attack may take different routes.
7. Routers are both CPU and memory limited.
8. Routers are not compromised. It is assumed as well that medium access control (MAC) filtering is enabled at the edge routers. Such an assumption is realistic since most of the routers have this function enabled.

DFM6 has been designed with two modules, the DFM6 Encoding (DFME6) and DFM6 Decoding (DFMD6) modules, which will be described below.

7.1.1 Selecting the Marking Field

To apply DFM in IPv6 networks, an appropriate field in the IPv6 header for embedding the marking data should be selected. IPv6 has two types of headers, basic and extension headers. The IPv6 basic header contains eight fields, which is bigger, but simpler that the IPv4 header. As described in Section 2.2, there are some previous approaches which use the 20-bit flow label field as the marking field in the basic IPv6 header. The main problem for this group of IPv6 traceback approaches is that overwriting the flow label field interferes with the main function of this field. Even if one assumes that this 20-bit field can be employed for traffic marking, it will require remarkable amounts of computational overhead to encode the 128-bit IPv6 address into the 20-bit flow label. This leads to the inevitable conclusion that the IPv6 basic header is not a suitable option for storing the marking data.

Several extension headers are defined in IPv6 to support a broad range of applications, and new extension headers may be defined in the future [19]. Among these extension headers, only the Destination Options Header (DOH) is designed to be examined and processed at the destination node [19]. In addition, DOH can provide a large enough space for storing the entire marking data. Therefore, for the DFM6 approach, the DOH is selected for storing the marking data in the IPv6 networks. Employing the DOH for storing the marking data eliminates the marking data fragmentation overheads that are experienced in the previous DFM design for IPv4 networks. The DOH is identified with a Next Header value of 60 in the immediately preceding header [19].

7.1.2 Mark Encoding

The mark encoding task is assigned to the DFM6 Encoding module (DFME6), which runs at the marking routers. The architecture of DFME6 is based on the fact that all packets in a flow have the same source. Thus, if the origin of even one packet in a flow can be found, then the origin of all of the other packets in the same flow is also discovered. Flow detection is an embedded feature in almost all manageable routers (i.e. Cisco has NetFlow, InMon has sFlow, and Juniper uses JFlow). The DFME6 module marks the outgoing IPv6 flows by selecting and marking only the first packet of each flow in order to have a low marking rate while achieving a high traceback rate. In DFM6, only the edge routers mark the flows, and the rest, including the core routers among the traffic path from the source to the destination, are not involved in the marking process.

The DFME6 module uses three identifiers to mark the flows in order to trace up to the attacker node. These three identifiers are described below.

- The IPv6 address of the egress interface of the edge router (16 octets): an edge router is the closest router to the attacker node with at least one public IPv6 address to its egress interface.

- Node-ID (6 octets): an identifier assigned to each source MAC address observed from the incoming IPv6 packet to the edge router. If the edge router is the closest router to the end network, then the source MAC address would be the MAC address of the source of the packet; otherwise, the source MAC address would be the MAC address of the previous hop (usually the previous router in the path).

- The network interface identifier, NI-ID (2 octets): this 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. The NI-ID specifies from which subnet a traffic flow comes. A two octet NI-ID, expressed with a range from 0 to 65535, is sufficient representing all possible network interfaces and VLANs on an edge router.

Using the above three identifiers (24 octets of marking data) to mark the first packet of each IPv6 flow, DFM6 is able to traceback not only up to the source edge router, but also to the exact source network interface of the edge router and even one step further to the source node located in a LAN behind the edge routers.

However, the question arises here is how to determine the flows in the IPv6 traffic to mark them. Traditionally, flow exporters (generators) have been based on the 5-tuple source and destination IPv4 addresses, ports, and the transport protocol type. ICMP flows have been defined with 6-tuple source and destination IPv4 addresses, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID. However, determining the flows in the IPv6 traffic is simpler than IPv4 networks since IPv6 includes a 20-bit flow label field which can be used by hosts to label a unidirectional sequence of packets uniquely from a host to a particular unicast, anycast, or multicast destination. The tuple of source and destination IPv6 address and the flow label is intended to identify a particular flow uniquely during its lifetime (plus a subsequent quarantine period) [24]. Since transport-layer port fields may be located at a variable offset within a packet due to the IPv6 extension headers, the flow label provides a fixed field in the IPv6 header to facilitate flow generation in routers [24]. The rules for the generation and use of the flow label values that are defined in RFC6437 [23] and are used in the DFM6 architecture include the following.

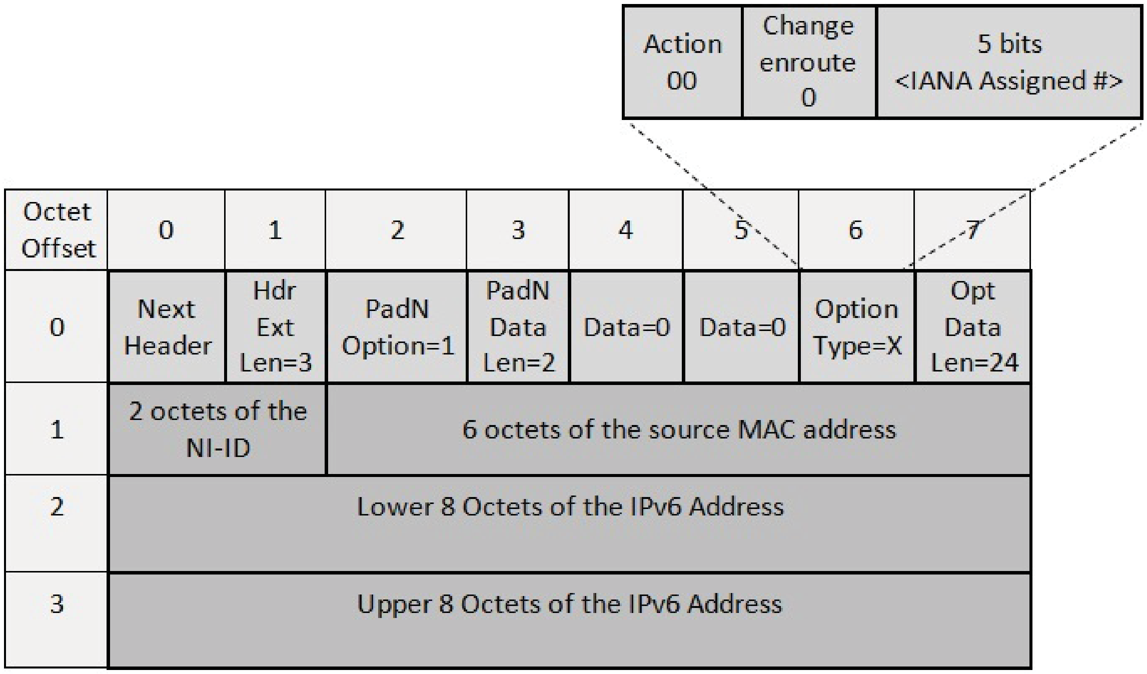
- Flow label values used previously with a specific pair of source and destination addresses must not be assigned to new flows with the same address pair within 120 seconds of the termination of the previous flow.
- The flow label value that is set by the source node must be delivered unchanged to the destination node(s).

This work defines the 3-tuple source and destination IPv6 addresses and the flow label as the Flow-ID. The DFME6 module maintains a table to keep track of the marked packets and their respective Flow-IDs. This table is called the Marking-Table. Once the DFME6 module observes an outgoing packet, first it extracts its Flow-ID and then looks for the existence of a table record for this Flow-ID in the Marking-Table. If this Flow-ID is not in the Marking-Table, it means that this flow is new to the DFME6 module, so the DFME6 module takes the steps below.

1. Creates a table record for the new Flow-ID in the Marking-Table;
2. Calculates the 24 octets of flow marking data;
3. Marks the selected packet with the marking data. If the selected packet does not have a Destination Option Header, creates one and stores the marking data; otherwise adds the marking data to the available Destination Option Header.

However, if this Flow-ID is already in the Marking-Table, it means that this flow has already been marked by the DFME6 module. In this case, the DFME6 module just forwards the packet without any change. It should be noted that because a flow label of zero indicates that the packet is not part of any flow, therefore the DFME6 module should mark every packet with a flow label of zero. To eliminate mark-spoofing attacks, the DFME6 module needs to monitor the Destination Option

Header of the incoming packets. If it faces a packet with the same DFM6 identifier in the Destination Option Header, this packet should be considered as a mark spoofing attack. In this case, the DFME6 module overwrites the Destination Option Header with the correct marking data.



Action 00	Change enroute 0	5 bits <IANA Assigned #>

Octet Offset	0	1	2	3	4	5	6	7
0	Next Header	Hdr Ext Len=3	PadN Option=1	PadN Data Len=2	Data=0	Data=0	Option Type=X	Opt Data Len=24
1	2 octets of the NI-ID		6 octets of the source MAC address					
2	Lower 8 Octets of the IPv6 Address							
3	Upper 8 Octets of the IPv6 Address							

Figure 7.1: The Proposed Format of the Destination Options Header for Storing the Marking Data

7.1.3 Marking Data Format

As described above, the IPv6 Destination Option header is selected for storing the marking data. The proposed format of the Destination Options header for storing the marking data is shown in Figure 7.1, which is based on the formatting guideline described in RFC2460 [19], and has the fields listed below.

- Next Header (One octet): Identifies the type of header immediately following the Destination Options header.

- Hdr Ext Len (One octet): Length of the Destination Options header in 8-octet units, not including the first 8 octets. As the total size of the proposed Destination Option Header is four 8 octets, the value of this field should be 3.

- The PadN option is used to align subsequent options and to pad out the containing header to a multiple of 8 octets in length. For N octets of padding, the PadN Data Len field contains the value N-2, followed by N-2 zero-valued octets. In the proposed Destination Option Header, 4 octets of padding are needed, so the PadN Data Len should be 2, followed by 2 zero-valued octets.

- Option Type (One octet): The Option Type is internally encoded into three fields, such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type. The third-highest-order bit specifies whether or not the Option Data of that option can change en-route to the packet's final destination. The remaining five bits along with these three bits produce a unique identifier for the option. By zeroing the first two bits, this scheme shows that nodes not recognizing this option type should skip over this option and continue processing the header. Setting the third bit to 0 shows that this option must not change en-route

- Option Data Len (One octet): Length of the Option Data field, in octets. In this scheme, there are 24 octets of marking data, so this field is set to 24.

- Option Data: Variable-length field. The 24 octets of marking data are stored in this field.

7.1.4 Calculating the Path Maximum Transmission Unit

In IPv6 networks, as intermediate routers do not support fragmentation, the source nodes should use the Path Maximum Transmission Unit (PMTU) algorithm prior to sending the packets [19]. For IPv6, Path MTU discovery works by initially assuming the path MTU is the same as the MTU on the link layer interface through which the traffic is being sent. If any of the packets sent on that path are too large to be forwarded by some node along the path, that node will discard them and return "ICMPv6 Packet Too Big" messages, allowing the source host to reduce its Path MTU appropriately. The process is repeated until the MTU is small enough to traverse the entire path without fragmentation [19].

The problem arises when the sender sends a packet that is equal to the size of the path MTU; then the DFME6 module cannot add the Destination Option Header to the packets to mark the flows. Since marking the packet with an extra 32 bytes will increase the size of the packet to greater than the path MTU, and given that the intermediate routers cannot do fragmentation, the packets will be dropped.

There has been some previous research trying to avoid this problem [34]. This work modifies the previous algorithms to adapt them to the proposed flow-based IPv6 traceback approach. In this new approach the path MTU needs to be reduced by 32 bytes, so that the marking router is able to add an extra 32 bytes destination option header to the first packet of each flow. To this end, the proposed DFME6 module on the edge router reduces the MTU value of the "ICMPv6 Too Big" packets by 32, and returns the packet back to the sender. The sender then sends the packet according to the size of this modified MTU, as shown in Figure 7.2.

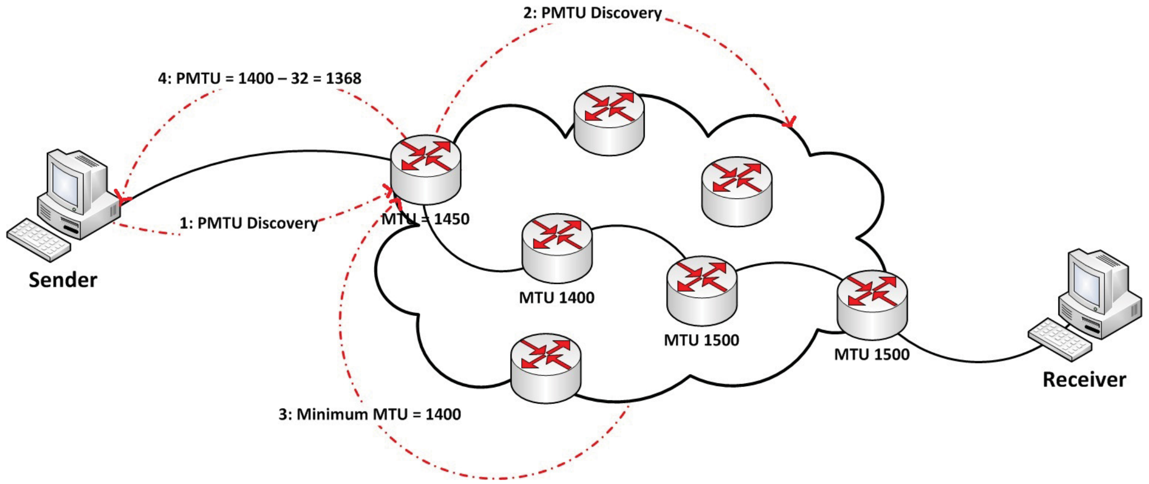


Figure 7.2: Modified PMTU Algorithm

According to [19], an IPv6 device cannot have less than a 1280 bytes MTU. Therefore, the minimum MTU size received by the DFM6 must be 1312 bytes (1280 plus 32 bytes for the Destination Option Header). This restriction does not affect the proposed system. The reasons listed below.

- Nowadays, the MTU size of the IPv6 routers is greater than 1,400 bytes (the usual MTU of Ethernet, Point-to-point over Ethernet (PPPoE), and Asynchronous Transfer Mode is 1500, 1492, and 1492 bytes, respectively, and could be more) [34]

- It is recommended by [19] that the IPv6 routers be configured with an MTU of

1500 bytes or greater (The default MTU size of the IPv6 routers is 1500 bytes or greater).

- Even though the probability of receiving a MTU size of less than 1312 bytes from an intermediate router is low, it might happen. As the DFME6 module reduces this value by 32 bytes, the originating IPv6 node may receive an "ICMPv6 Packet Too Big" message reporting a Next-Hop MTU less than 1280. In that case, the IPv6 node can either reduce the size of subsequent packet it sends or perform IPv6 fragmentation on packets by breaking the packet into N roughly equal-length pieces (where N is minimized and the length of each piece is smaller than the Next-Hop MTU).These fragments will be reassembled at the destination [21].

7.1.5 Mark Decoding

The DFM6 Decoding module (DFMD6) is located at the destination network and its goal is to infer the origin of the incoming traffic, even if the source IPv6 addresses of the incoming packets are spoofed. As the complete marking data is stored in the Destination Option Header, mark decoding consists of the simple process of extracting the marking data from the DOH header.

Given that the origin of all packets in a flow is the same, once the marking data of a flow is extracted from one packet, then the origin of all other packets in the same flow is discovered as well, even if the source IPv6 address of those packets is incorrect or has been spoofed. Using DFMD6, the destination is able to distinguish the traffic of the different nodes behind an edge router. As a result, when abnormal traffic is observed, the victim is able to distinguish between the attack and legitimate traffic and infer the source of an attack, even if it is behind a NAT or a proxy device.

7.1.6 Discussion

The location of the marking router, the IPv6 tunnelling and the rapid changes of the Flow-ID values are discussed below.

The Location of the Marking Router

Every router with a valid IPv6 address on its egress interface can potentially act as a marking router. DFM6 is able to traceback to the source of the traffic one step behind the ingress interface of the making router. Therefore, the ideal location for enabling the DFME6 module is the closest router to the end networks (edge router), as DFM6 would be able to traceback not only up to the source edge router, but also to the exact source network interface of the edge router and even one step further to the source node located in a LAN behind the edge routers. In addition, by enabling the DFME6 module at the edge routers, this approach ensures that the MTU values of all "ICMPv6 Too Big" packets are modified. Placing the DFME6 module on routers other than the edge router might result in some missing "ICMP6 Too Big" packets that choose to take return routes other than the initial route.

The IPv6 Tunnelling

There are several kinds of tunnelling in IPv6 networks such as IPv6 over IPv4 or router to router tunnelling. Usually, these kinds of tunnelling are problematic in logging and probabilistic-based traceback techniques in which the intermediate routers need to have access to the IPv6 header. However, as these techniques keep the original IPv6 header and the original header can be extracted at the destination, they do not cause any interruptions to the proposed traceback algorithm. It should be noted here that the above discussion is about tunnelling between the intermediate routers, not between end-to-end devices, as there is no need of traceback if the two ends establish a tunnel between each other.

Rapid changes to the Flow-ID value

An attacker could perform rapid cycling of flow label values in such a way that the packets of a given flow will each have a different Flow-ID value. In this scenario, DFM6 considers each packet as the first packet of a new flow and marks every packet. This does not affect the 100% traceback rate of DFM6, but may increase the marking rate. To address this issue, the flow exporters can switch to the classic flow identifiers (5-tuple for TCP and UDP flows and 6-tuple for ICMP flows, described in Section

7.1.2), instead of using the 3-tuple.

7.2 Experimental Results

To evaluate the proposed DFM6 approach and compare its performance with the other previous deterministic methods, it was implemented along with the four DPM approaches described in the literature [33, 96, 85, 69], and they were evaluated on a testbed network. To the best of the author's knowledge, these four approaches are the only previous works which aim to adapt the Deterministic Packet Marking IP-Traceback approach (DPM) [38] on the IPv6 networks. The CAIDA IPv6 5 June 2012 Anonymized Internet Traces were employed as the evaluation traces. This data set contains anonymized passive traffic traces from the equinix-sanjose monitor taken during the IPv6 Launch Day on 6 June 2012. This data set consists of standard tcpdump traffic and is publicly available [16]. The evaluation traffic is sent from a local area network behind a marking router and directed to the destination. For this purpose, the data sets were replayed on the testbed network using the tcpreplay and tcprewrite open source applications [1]. In addition, mark encoding and mark decoding programs were implemented using the Winpcap library by C++ [12] to mark the packets at the edge router and traceback the source of traffic at the destination.

7.2.1 Evaluation Metrics

To evaluate the DFM6 approach, and compare it with the other deterministic marking approaches, the four evaluation metrics listed below were employed.

- Traceback Rate: the ratio of the number of successfully traced back packets to all packets.

- Marking Rate: the ratio of the marked packets to all packets.

- Bandwidth Overhead: the ratio of the volume of the overhead traffic to the volume of the original traffic.

- Number of Required Packets: the number of required marked packets at the destination to complete the traceback process.

Naturally, the desired outcome should have a higher traceback rate, a lower marking rate, lower bandwidth overhead and a lower number of packets needed to complete the traceback process. Table 7.1 and Figure 7.3 present the evaluation results of the proposed DFM6 approach and the four previous deterministic approaches on IPv6 traceback from the literature [33, 96, 85, 69]. These results show that all of the deterministic IPv6 traceback methods have a 100% traceback rate. However, they achieve this at the expense of a 100% marking rate.

On the other hand, the proposed DFM6 approach achieves a 100% traceback rate by marking only approximately 14% percent of the packets. It should be noted here that the marking rate of DFM6 could be different based on the data set. However it would always be less than the other DPM approaches because the marking rate of all other DPM approaches is always 100%, regardless of the traffic pattern, as they mark every packet, whereas DFM6 only marks one packet per flow.

As described is Section 2.2, the DPM2006 [33], DPM2011 [96], DPM2012 [85] and DPM2014 [69] approaches mark each packet with an additional 24 octets, 24 octets, 8 octets and 40 octets of marking data, respectively. DFM6 marks the packets with an extra 32 octets, but only one packet per flow needs to be marked. Therefore as expected, the bandwidth overhead for DFM6 is much lower than for the other four DPM approaches (3.92%), and it can be even lower for DDoS attack traffic as the number of packets per flow in DDoS traffic is usually more than for normal traffic.

Finally, the number of required marked packets for completing the traceback process at the destination for all approaches, except for DPM2012, is one. It is because all of these approaches store the complete marking data in each marked packet. Therefore, at the destination, only one marked packet is enough to complete the traceback process. However at the source side, DFM6 marks only one packet per flow, while the other approaches mark every packet. In term of the number of required packets, DPM2012 has the worst performance, as it divides the marking data into 16 parts and marks each packet with one of the 16 fragments. Therefore, all of the 16 fragments should arrive at the destination to extract the marking data.

It should be noted that if any of the marked packets get lost and does not arrive to the destination, the specific flow that the lost marked packet belongs to cannot be traced back. However, the victim can traceback to the attacker using the other flows

Table 7.1: The Evaluation Results of the Proposed DFM6 Approach, as well as Four Previous Deterministic Approaches on IPv6 Traceback.

Method	**Traceback Rate**	**Marking Rate**	**Bandwidth Overhead**	**Number of Required Packets**
DFM6	100%	13.7% (One Packet/flow)	3.92%	1
DPM2006	100%	100% (Every Packet)	18.57%	1
DPM2011	100%	100% (Every Packet)	18.57%	1
DPM2012	100%	100% (Every Packet)	6.18%	16
DPM2014	100%	100% (Every Packet)	30.95%	1

from the same attacker.

The results show that the proposed DFM6 approach outperforms the other deterministic marking approaches, as it has the same “traceback rate”, but the lowest bandwidth overhead, marking rate and number of required packets.

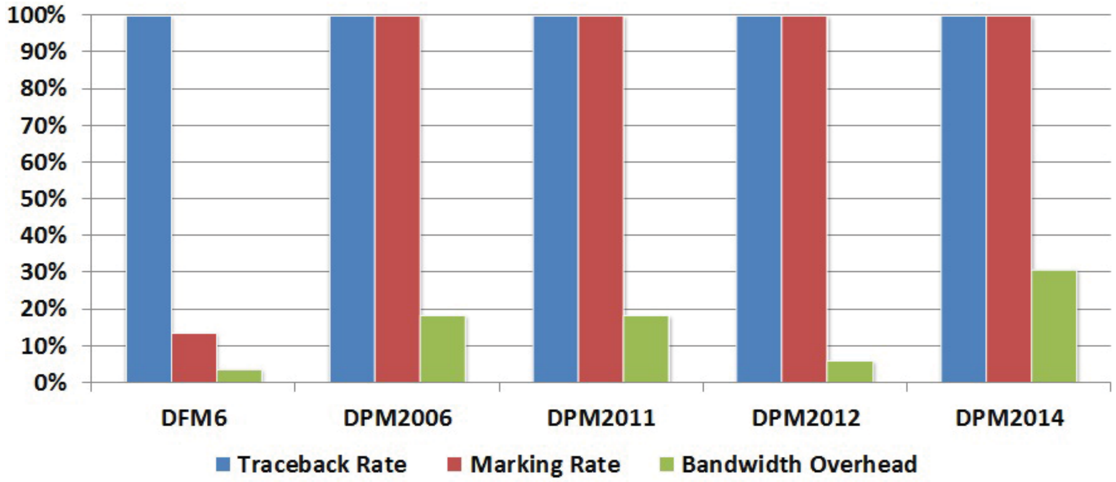


Figure 7.3: Comparison among DFM6 and Four other Deterministic Marking Approaches in Terms of Traceback Rate, Marking Rate and Bandwidth Overhead

7.2.2 Memory Usage

The DFMD6 module at the destination does not require any extra memory space. However, the space required for running the DFME6 module on a router, $DFME6_{mem}$, is equal to sum of the required spaces of the Marking-Table and the NI-ID-Table:

$$DFME6_{mem} = MT + NT \tag{7.1}$$

where MT is the size of the Marking-Table and NT is the size of the NI-ID-Table.

1. **Marking-Table:** as described above, the DFME6 module maintains the Marking-Table to keep track of the packets and their Flow-ID, as well as the marking data. Each row in this table stores a Flow-ID and its flow marking data. The DFME6 module does not keep the record of a flow when the flow is over. The end of a flow is detected by a four-way connection termination handshake or a pre-defined time-out duration for TCP flows and a pre-defined time-out duration for UDP and ICMP flows. Thus, the number of rows in the Marking-Table varies and depends on the number of concurrent flows, which can be calculated by:

   $$MT = N_{CF} \times (F_{ID} + MD) \tag{7.2}$$

   where N_{CF} represents the number of concurrent flows in the traffic, F_{ID} represents the size of Flow-ID and MD represents the size of the marking data.

   Flow-ID is also defined as the three tuple of the source IPv6, the destination IPv6 and the Flow Label. Therefore:

   $$F_{ID} = S_{IP} + D_{IP} + FL \tag{7.3}$$

   where S_{IP}, D_{IP} and FL are the size of the source IPv6 address, the destination IPv6 address and the Flow Label, respectively.

   As described above, the marking data consists of the egress interface IPv6 address of the marking router, the source MAC address and the NI-ID. Therefore:

$$MD = E_{IP} + S_{MAC} + NI_{ID} \tag{7.4}$$

where E_{IP}, S_{MAC} and NI_{IP} are the size of the egress interface IPv6 address of the marking router, the size of the source MAC address and the size of the NI-ID, respectively.

Since the sizes of the IPv6 the MAC address, the Flow-Label and the NI-ID are 16 octets, 6 octets, 2.5 octets and 2 octets, respectively, then Equation 7.2 can be summarized to:

$$MT = 58.5N_{CF} \tag{7.5}$$

2. **NI-ID-Table:** a marking router keeps an NI-ID-Table and assigns an NI-ID to each interface from 0 to a maximum of 65535. Each NI-ID-Table entry consists of an NI-ID, the MAC address of the network interface and the Vlan ID. Therefore NT is calculated by:

$$NT = N_i \times (NI_{ID} + I_{MAC} + V_{ID}) \tag{7.6}$$

where N_i is the number of physical and virtual interfaces on the marking router, I_{MAC} is the size of the router interface MAC address and V_{ID} is the size of the Vlan ID.

Given that the NI_{ID}, the I_{MAC} and the V_{ID} are 2 octets, 6 octets and 2 octets, respectively, Equation 7.6 can be summarized to:

$$NT = 10N_i \tag{7.7}$$

Therefore the space required for running the DFME6 module on an edge router, $DFME6_{mem}$, equation 7.8, is sum of equations 7.5 and 7.7:

$$DFME6_{mem} = 58.5N_{CF} + 10N_i \tag{7.8}$$

Since on the testbed network the maximum number of concurrent flows was 893, and the marking router had the interfaces (no vlan interface), therefore the maximum required space for running the DFME6 module was about $51KB$: (58.5×893)+(10×2) bytes.

7.3 Summary

This chapter has introduced DFM6, a novel Deterministic Flow Marking for IPv6 networks, which is able to traceback up to the attacker node behind a NAT or a proxy server. DFM6 performs this by selecting and marking only the first packet of each flow. This leads us to have both advantages of high traceback accuracy as well as low processing and marking overhead. DFM6 consists of two modules, the DFM6 encoding module (DFME6) and the DFM6 decoding module (DFMD). The DFME6 module runs at the egress interface of the marking router and marks the outgoing traffic. The PFMD module runs at the destination network and tries to infer the source of traffic by extracting the marking data from the marked packets. DFM6 was compared with four deterministic IPv6 traceback approaches. The results based on the CAIDA IPv6 data set show that DFM6 has the same traceback rate as the other four IPv6 traceack approaches (100%), but it outperforms the other approaches as it has the lowest bandwidth overhead (3.92%) and marking rate (13.7%). Finally, it requires only one packet per flow to complete the traceback process.

Chapter 8

Autonomous System-based Flow Marking: ASFM

Previous work on IP-Traceback generally requires deployment over all routers (Probabilistic Packet Marking (PPM) approaches) to rebuild the complete path taken by the attacker traffic, or all ISP edge routers (Deterministic Packet Marking (DPM) or Deterministic Flow Marking (DFM) approaches) to discover the source network of the attacker traffic. Naturally, these requirements limit the potential deployment of IP-Traceback solutions on the Internet.

By contrast, this chapter proposes an Autonomous System-based Flow Marking Scheme for IP-Traceback, ASFM, that could be partially deployed on large scale networks such as the Internet. The proposed ASFM traceback method operates on the Boarder Routers of Autonomous Systems, called ASBRs, which are members of an AS-level overlay network for IP-Traceback. Information exchanged between the ASBRs in the overlay network is carried by the update-message community attribute of the Border Gateway Protocol (BGP). Update-message community attribute enables information to be passed across ASs that are not necessarily involved in the overlay network. Deploying a traceback system on all routers is not required to enable an efficient IP-Traceback. Rather, it is sufficient to identify some key points on the path where attacker packets are being forwarded. This enables efficient countermeasures to be taken in a distributed way to block the ongoing attack (e.g., at the closest traceback-collaborative ASs with respect to the sources of a DDoS attack, or at the AS that forwards more traffic). Evaluations using real life data sets, demonstrate that this system can be partially deployed over the network and provides very promising performance. Moreover, findings indicate that having a relatively low number of ASs using the ASFM is sufficient to provide an efficient IP-Traceback at the AS level.

Previous chapters evaluated the accuracy of deploying a flow-based traceback system. This chapter builds upon those results and proposes an AS-level IP-Traceback system, ASFM.

8.1 Motivation and Background

This section describes some characteristics of the proposed IP-Traceback system, ASFM, as well as the implementation of an overlay network to support IP-Traceback.

8.1.1 AS-Level IP-Traceback

An Autonomous System (AS) is a group of IP networks operated by one or more network operator(s), which has a single and clearly defined external routing policy. An Autonomous System Number (ASN) is a globally unique number in the Internet to identify an AS. AS numbers are 32-bit integers, assigned and managed by the Internet Assigned Numbers Authority (IANA). Autonomous System Border Routers (ASBRs) are connected to more than one AS to exchange routing information with routers in other ASs. The traffic to/from an AS is controlled by its ASBRs.

Inter-domain traceback [72, 59, 71, 41, 32] has some advantages over router-level traceback. Using the AS in traceback will not reveal the topology, which is advantageous for network operators. If each router in the AS participates in the marking scheme, then one can easily infer the network architecture by observing the markings. Additionally, the number of AS hops is significantly less than the number of routers hops. As of August 2015, there are 51,459 ASs [2], compared to millions of routers on the Internet. As shown in Figure 8.3, in more than 99.5% of ASs, a packet passes through less than seven ASs before reaching its destination. On the other hand, the corresponding number of routers passed by a packet from source to destination is more than twenty. Therefore, it is much easier to traceback through ASs than routers. Furthermore, routing on the AS is carried by BGP and the distance between ASs can be known in advance. Knowing the distance in advance can be used to optimize the marking probability and reduce the required number of packets needed to reconstruct the path.

8.1.2 The Role of BGP in ASFM

The BGP routers periodically use update messages to exchange routing information between each other [77]. An update message has a field called path attribute which is actually a collection of attributes that are associated with a given route which may influence the route selection process. One of these attributes, called the "community attribute", is used to group destinations that share common characteristics. The community attribute is highly flexible and is indeed used for many different purposes, such as multihome routing, traffic engineering, support of virtual private networks (VPNs), and mobile honeypot systems. In this research, the use of this attribute is proposed in order for ASs to learn and discover which other ASs also support this solution in a partial deployment scenario where some ASs do not have it deployed.

An important characteristic of a community attribute is that it is an optional and transitive attribute of BGP. This means that if the BGP running in a border router does not recognize an attribute present in the update message, a verification of whether the transitive flag is set or not is made. In cases when the flag is set, the attribute is forwarded in a new update message by the AS border router to its peers. This feature enables the information about the IP-Traceback community to be forwarded by those ASs that are not participating in the scheme, and therefore information eventually reaches the ASs that implement the proposed IP-Traceback solution. After a sequence of update messages is forwarded, the overlay network for IP-Traceback is established / updated. At this time, each IP-Traceback AS has an overlay table that contains a list of all the other IP-Traceback ASs. The update of the overlay table occurs in a way similar to the updating of the BGP routing table because the overlay information is carried on the BGP messages.

8.1.3 Overlay Network for IP-Traceback

An overlay network is a virtual network that is built on top of the physical network. Nodes in the overlay network can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying physical network [10]. The overlay network enables IP-Traceback among participating AS routers (they are not required to be adjacent routers at the routing or AS-level). The IP-Traceback is performed hop by hop in the AS-level overlay network. This feature eliminates the requirement adopted by several previous approaches that the traceback system should be deployed in all the routers of the monitored network.

In a manner similar to previous efforts (Castelucio et al. [41]), this approach

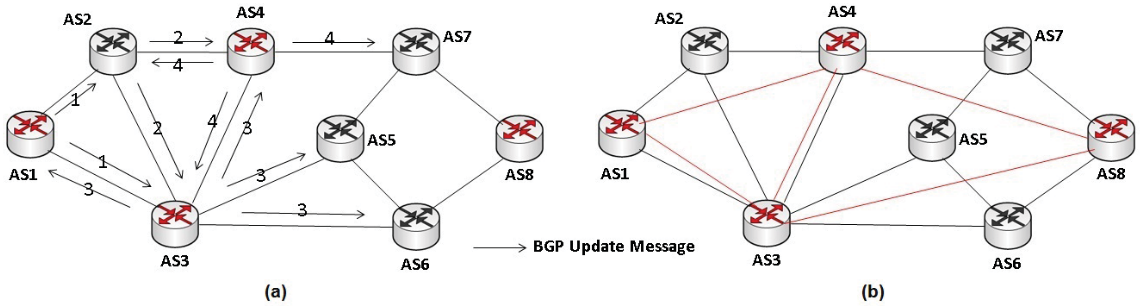


Figure 8.1: Building the AS-level Overlay Network for IP-Traceback. a) BGP Update Messages with the IP-Traceback Community; b) The Resulting AS-level Overlay Network for IP-Traceback.

proposes to deploy an overlay network for IP-Traceback and use BGP as a vehicle to distribute the deployment information. Figure 8.1.a illustrates the construction of the overlay network (the red ASs have the traceback system running). At the beginning, the overlay tables are empty. When AS1 sends an update message to its peers (step 1), AS3 receives the message and updates its table by registering AS1 as its neighbour in the overlay network. On the other hand, since AS2 does not have the system deployed, AS2 just generates a new update message and sends the information previously received about the IP-Traceback community in a transparent way to its neighbors AS4 and AS3 (step 2). This information is set as transitive in the update message received from AS1. AS3 receives the update message coming from AS2 and simply ignores it since AS3 has already received the information about AS1. When it is AS4's turn, AS4 inserts AS1 as its neighbour in the overlay table. When AS3 and AS4 create a new update message, they insert their information and send the message to their neighbors (steps 3 and 4, respectively). After receiving the update messages from each other, AS3 and AS4 insert each other as neighbors in their overlay tables. Similar procedures are repeated by all the ASs in the network until all the participating ASs are reached and know about each other, thus forming the overlay network. The resulting overlay network is illustrated in Figure 8.1.b where the red lines represent the connections of the overlay network.

I emphasize that the exchanged information about the IP-Traceback community incurs no significant additional overhead to the network because such information is carried inside update messages that are native and exchanged periodically by the BGP routers.

As the proposed AS-level traceback system depends on some collaboration among ASs, it is worth mentioning at least two motivations for the ASs to deploy the proposed system: 1) if an AS collaborates, it can perform filtering of the attacker traffic, thus reducing the consumption of network resources in its own domain because these resources are being used by attackers; 2) an AS with the traceback system installed can initiate its own traceback request when an attack is detected within its domain.

8.2 Autonomous System-based IP-Traceback

The actual scheme's algorithm as well as the marking fields in the packets' header are described in this section.

8.2.1 Overloading the IP Header

ASFM uses three fields of the IP header to store the marking information, similarly to previous work by Gao et al. [58] and Yang [94], Figure 4.5. They are the Identification field, Flags and the Fragment offset. Consequently, ASFM uses a total of 32 bits of the IP header space for storing the marking information.

8.2.2 Algorithm

Instead of marking the traffic with the IP address of a router, this approach marks the traffic with the AS number, ASN. Moreover, unlike the packet-based IP-Traceback approaches, ASFM is a flow-based approach, meaning that it marks the first two packets of each flow which results in lower marking rates. The definition of a flow for ASFM is the same as defined previously in Section 4.2.1 and accepted as a unidirectional sequence of packets between two networks that have five-tuple information: source IP address, destination IP address, L4 protocol type (TCP/UDP), source port and destination port numbers over a pre-defined duration. An ICMP flow is defined as a unidirectional sequence of ICMP packets between two networks that have six-tuple information as follows: source IP address, destination IP address, L4 protocol type (ICMP), ICMP type, ICMP code and ICMP ID in common.

This scheme reserves five fields in the packet header, totalling 32 bits, as described below.

- 16-bit ASN frag field: it is used to store one ASN fragment of the participating ASBRs on the path.

- 1-bit Flag field: the value is either 0 or 1. It indicates which packets of a flow are marked.

- 11-bit Hash field: this field is used for authentication and represents one hash value fragment of the ASN and a 5-tuple of flow identifiers for TCP/UDP traffic or a 6-tuple of ICMP flow identifiers (See Section 8.3).

- 3-bit Distance field: it represents the number of participating ASBRs from the marker ASBR to the last participating ASBRs in the path (including itself). 3 bits for the distance field can represent up to a distance of 8 hops in terms of ASs traversed which is sufficient for almost all Internet paths, as shown in Figure 8.3. Once the value of this field is written by one ASBR, the following ASBRs will not change it. However, if the following participating ASBRs find that the mark information in the marking space is incorrect, they will initialize the marking space, including changing the value in this field (the details can be found in Section 8.3).

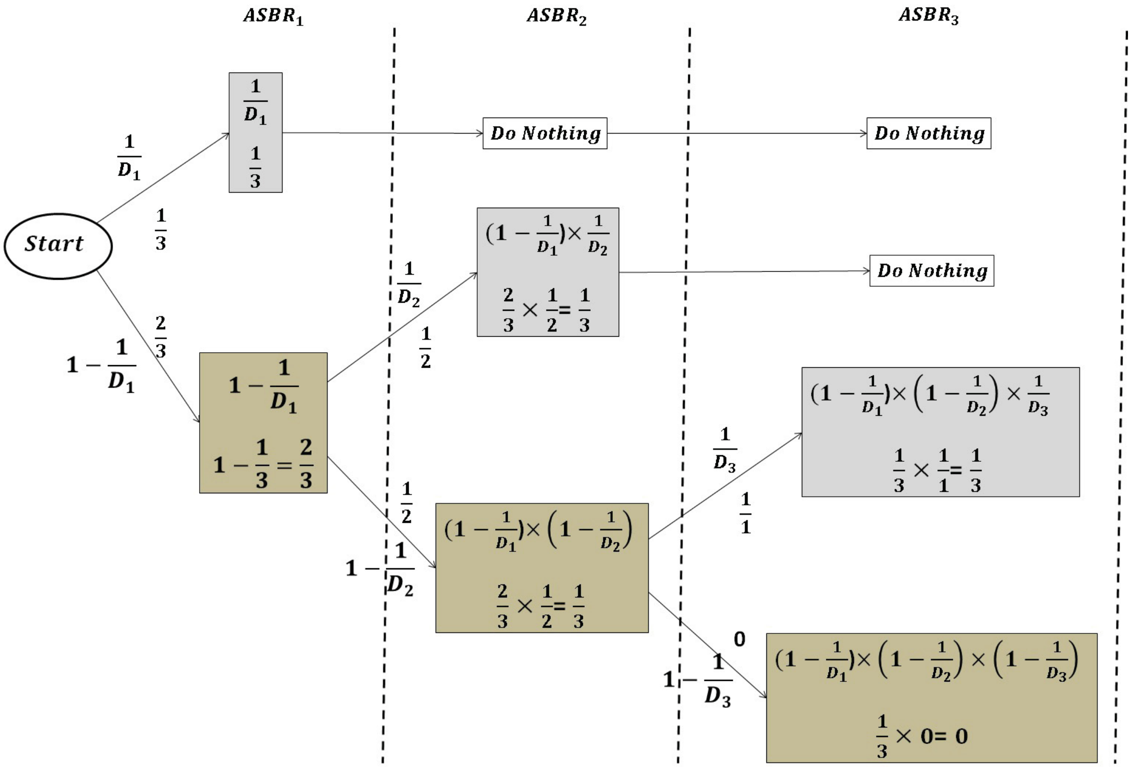
- 1-bit Frag# field: the value is either 0 or 1. It indicates which fragment of the ASN is filled in the packets.

A participating ASBR marks an unmarked flow only if the flow is forwarded to a router belonging to another AS. Thus, a flow might get marked only when it exits an AS.

Restricted by limited marking space in the packet header, upon receiving an unmarked flow that is being forwarded to another AS, each participating ASBR needs to split the corresponding ASN into two fragments and places one of the ASN fragments into a packet with probability P, set the Flag field, set the Frag# field, and set the distance field, which is the number of participating ASBRs from itself to the last participating ASBR in the path. The Hash field will be described in Section 8.3. If the ASBR chooses not to mark the flow, it just forwards the flow.

This approach is designed in such a way that eventually all flows in the traffic are marked. Since ASFM is a distributed approach over the Internet, the marking overhead is distributed over all the participating ASBRs along the path from source to destination. Assuming that there are N participating ASs between the source and the destination, the objective is to choose an appropriate marking probability in such a way that finally each participating ASBR marks $1/N$ of all flows in the traffic. Obviously, if the number of participating ASs increases, then the marking overhead on each ASBR decreases.

Knowing the distance for a packet's journey from source to destination in advance is one of the advantages of ASFM, as it is derived in the ASBR from the overlay table. The unknown distance for the packet's journey from source to destination was one of the main limitations of PPM and other techniques. Knowing the distance in advance can be used to optimize the marking probability and reduce the required number of packets for reconstructing the path.



$ASBR_1$
$ASBR_2$
$ASBR_3$
Start
$\frac{1}{D_1}$
$\frac{1}{3}$
$1-\frac{1}{D_1}$
$\frac{2}{3}$
$\frac{1}{D_1}$ $\frac{1}{3}$
Do Nothing
Do Nothing
$1-\frac{1}{D_1}$ $1-\frac{1}{3}=\frac{2}{3}$
$\frac{1}{D_2}$
$\frac{1}{2}$
$(1-\frac{1}{D_1})\times\frac{1}{D_2}$ $\frac{2}{3}\times\frac{1}{2}=\frac{1}{3}$
Do Nothing
$1-\frac{1}{D_2}$
$\frac{1}{2}$
$(1-\frac{1}{D_1})\times\left(1-\frac{1}{D_2}\right)$ $\frac{2}{3}\times\frac{1}{2}=\frac{1}{3}$
$\frac{1}{D_3}$
$\frac{1}{1}$
$(1-\frac{1}{D_1})\times\left(1-\frac{1}{D_2}\right)\times\frac{1}{D_3}$ $\frac{1}{3}\times\frac{1}{1}=\frac{1}{3}$
$1-\frac{1}{D_3}$
0
$(1-\frac{1}{D_1})\times\left(1-\frac{1}{D_2}\right)\times\left(1-\frac{1}{D_3}\right)$ $\frac{1}{3}\times 0=0$

Figure 8.2: The Marking Process in a Network Path with Three Collaborative ASBRs: ASBR1, ASBR2, and ASBR3

In this approach, the probability of marking the unmarked flows at $ASBR_i$ is:

$$P_i = \frac{1}{D_i} \tag{8.1}$$

where D_i is the number of participating ASBRs from $ASBR_i$ to the destination. Therefore, the probability of not making at $ASBR_i$ is:

$$1 - P_i = 1 - \frac{1}{D_i} \tag{8.2}$$

Once an ASBR marks a flow, the following participating ASBRs do not mark the same flow. The detailed marking procedure at each participating ASBR is depicted in Figure 8.2, in which the attack traverses the participating $ASBR_1$, $ASBR_2$ and $ASBR_3$. Each ASBR makes the decision whether to mark the current flow or not independently. At each ASBR, the bottom box shows the probability of not marking the flows from the first ASBR until $ASBR_i$, P_{U_i}, which is calculated by:

$$P_{U_i} = \prod_{j=1}^{i}(1 - \frac{1}{D_i}) \tag{8.3}$$

and the probability of marking the flows at $ASBR_i$, P_{M_i}, is presented in the upper box which is calculated by:

$$P_{M_i} = P_{U_{i-1}} \times \frac{1}{D_i} \quad \text{where} \quad P_{U_0} = 1 \tag{8.4}$$

Thus, the marking probability at each ASBR, P_{M_i}, is equal to $1/N$ where N is the total number of participating ASBRs in a path. It shows that the marking procedure is distributed equally over all participating ASBRs in the path. Consequently:

$$\sum_{j=1}^{i} P_{M_i} = 1 \tag{8.5}$$

and:

$$\prod_{j=1}^{i}(1 - \frac{1}{D_i}) = 0 \tag{8.6}$$

This means that finally all flows in the traffic are marked and eventually there is no unmarked flow left in the traffic.

For clarification, the numeric marking results of having three participating ASBRs in a path are shown in Figure 8.2. As can be seen, the marking results in all of the

ASBRs are equal which is $1/N = 1/3$. At $ASBR_1$, 1/3 of the flows are marked, so 2/3 of all flows are not marked. At $ASBR_2$, 1/3 of the flows are marked again, so 1/3 of the total flows are left unmarked. Finally at $ASBR_3$, the last 1/3 of the flows are marked, so there are no more unmarked flows at this point.

After marking the traffic, the victim will see three different marks, each from one participating ASBR in the attack path. The victim can locate the distance of each participating ASBR from itself by looking at the packet distance field. After enough packets have been sent, the victim will have received at least one marking from every participating ASBR. The victim can reconstruct the ordered path with the help of the distance field. One important contribution of this work is that ASFM can identify some key points in the path where attacker packets are being forwarded. For example, the closest traceback-collaborative AS with respect to the sources of a DDoS attack, or the AS that forwards more traffic. After identifying the key AS, the victim can decide how to proceed, such as contacting the owner of that AS (service provider) etc.

8.3 Authentication

This section presents a simple but effective authentication mechanism to prevent compromised routers from forging the markings and thus misleading the victims. For this purpose, it is assumed that it is hard to compromise Autonomous System Border Routers (ASBRs). This assumption is valid since once an ASBR is compromised much worse attacks than (D)DoS attacks can be possible [64].

The authentication scheme is based on a one-way hash function. A one-way hash function, H, maps an input of any length to a fixed-length bit string. Each participating AS is assigned a secret hash key. This key is known to all ASBRs that belong to the AS and to the participating ASBRs that are connected directly to the AS via the overlay network. For additional security, the one way hash function can be implemented as a one way hash chain method to produce many one-time keys from the initial secret key. This approach requires that a secure inter-AS routing infrastructure be presented, so that ASs deploying this solution can exchange their secret keys reliably. It is assumed that eventually there will be an adoption of a secured BGP routing infrastructure to combat other serious security issues on the

Internet, and to facilitate a multitude of security services. Different methods exist for securing BGP [88]. Hence, it can be assumed that whichever method is implemented eventually, the key exchange between ASs in the overlay network will be reliable.

Denote the secret key of an AS_i as K_i. $H(M, K_i)$ is used to denote the hash value of message M with key K_i. The 22-bit hash value generated is as follows:

$$H(ASN_i + RP, \quad K_i) \tag{8.7}$$

ASN is the 32-bit Autonomous System Number of the AS to which the marking ASBR belongs and RP is the Redundancy Predicate. PR should be flow-dependent to prevent a replay attack, otherwise, a compromised router can forge other routers' markings simply by copying the marking of one flow to another. One method of computing a Redundancy Predicate is to set RP to 5-tuple flow identifiers in TCP/UDP traffic or 6-tuple flow identifiers in ICMP traffic. The 22-bit hash result is fragmented into two parts and each marked packet in the flow carries one 11-bit hash value fragment in its Hash field.

If the flow is entering the AS, then the ASBR uses the symmetric secret key of the AS to which the flow is forwarded. However, should the flow be forwarded to another AS, then the ASBR uses the symmetric secret key of its AS. Once a participating ASBR receives a marked flow, it first authenticates the flow by computing the hash value of the received packet. If the verification is fulfilled, then it recalculates the hash value with its own secret key and rewrites the Hash field with the new value and forwards the flow. Otherwise, should the verification fail, the ASBR unsets the Flag field. This means that the flow is ready to be marked by either this ASBR or by the following ASBRs. The marking procedure is shown in Algorithm 3.

One of the advantages of the ASFM authentication algorithm is that the authentication process takes place in the intermediate ASBRs, not the victim. Therefore, the victim does not need to authenticate the marked flow.

8.4 Experimental Results

To evaluate the proposed scheme in real settings, a set of experiments was conducted using the same six data sets already used for evaluating the DFM, PFM, UFM and DPM approaches, including the MAWI (Measurement and Analysis on the WIDE

Algorithm 3 Marking Procedure at $ASBR_i$

```
$K_i$ is the secret key of $AS_i$;
$K_{i+1}$ is the secret key of the next participating $AS$;
$ASN_i$ is the $ASN$ of $AS_i$;
$ASN_m$ is the extracted ASN from the marked flow;
for each received flow $F$ in $ASBR_i$ do
    if Flag field is set {if flow is already marked} then
        if $H(ASN_m + PR, K_i)$ = hash value in Hash fields {if authentication passes} then
            set the Hash fields to $H(ASN_m + PR, K_{i+1})$;
        else
            unset Flag field;
        end if
    else
        get a random $P$ from $[0, 1)$;
        if $P < 1/D_i$ then
            mark $F$ with $ASN_i$;
            set Flag field;
            set Hash fields to $H(ASN_i + PR, K_{i+1})$;
        end if
    end if
end for
forward $F$;
```

Internet) traffic archive December 2012 [8], the CAIDA DDoS attack 2007 [17], the CAIDA anonymized Internet traces October 2012, December 2012 [14], March 2014 and July 2014 [15] data sets. The descriptions of these data sets are presented in Section 4.4. The data sets employed in this work are real-life tcpdump files and are available publicly. Table 4.4 presents statistical information on all of the aforementioned network traffic traces.

In addition, the CAIDA skitter AS links data set [18] was employed to have a view of the Internet topology. This data set reflects packets that have actually traversed a forward path to a destination. Using this data set, the distance can be calculated between any given two ASs on the Internet using Dijkstra's algorithm [46]. The results, Figure 8.3, show that 99% of the time each AS on the Internet could be reached by a 1 to 7 hop distance. These results confirm that using three bits in the packet header for the distance field is enough.

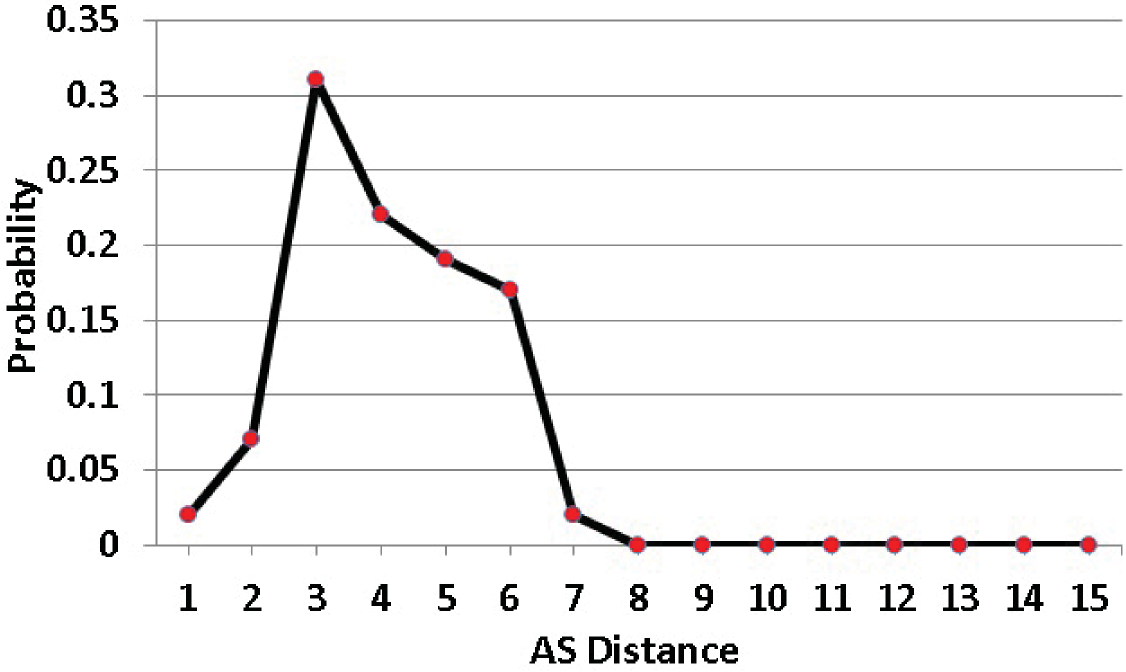


Figure 8.3: The Distance Probability between Two ASs, Using the CAIDA Skitter Data Set

8.4.1 Marking Rate

The performance of ASFM can be evaluated by the number of packets which must be marked. Table 8.1 shows the total marking rate of ASFM for the six evaluation data sets.

As with DFM when $K = 2$, ASFM marks the first two packets of each flow. Therefore the marking rate of DFM for $K = 2$ is equal to the marking rate of ASFM. This can be seen by comparing the results of Tables 4.5 and 8.1. However, since the marking task is distributed equally over all participating ASBRs on a single path, the marking rate on each individual ASBR is based on the number of participating ASBRs, as shown in Figure 8.4.

These results demonstrate that it is not necessary to deploy the proposed ASFM IP-Traceback approach on all ASs on the Internet to have an efficient marking system. Instead, it seems that choosing certain key locations on the Internet (e.g., the ASs which forward more traffic) enables efficient countermeasures to be taken in a distributed way to block the ongoing attack. During these experiments it was observed as well that when there are three participating ASBRs on a single path, the average marking rate for each participating ASBR is about 3%, which is not a high load.

Table 8.1: The Marking Rate (MR) and the Traceback Rate (TR) of the ASFM on Six Evaluation Data Sets

	CAIDA	CAIDA	CAIDA	MAWI	CAIDA	CAIDA
	Oct. 2012	Dec. 2012	DDoS 2007	Dec. 2012	Mar. 2014	Jul. 2014
MR	11.32%	6.24%	0.58%	8.6%	6.91%	9.11%
TR	91.48%	95.75%	99.84%	97.23%	94.57%	92.7%

8.4.2 Traceback Rate

The traceback rate is the ratio of the number of successfully traced back packets, to all packets. The traceback rate of ASFM is independent from the number of traceback-collaborative ASs in a single path. That is because increasing the number of participating ASs results in a lower marking rate for each individual ASBR, but this does not affect the traceback rate. Table 8.1 shows the traceback rate of ASFM on the five evaluation real-life traffic data sets. Since ASFM marks the first two packets of each flow, having some flows with only one packet decreases the traceback rate, as ASFM cannot traceback these kinds of flows. That is the reason why the traceback rate of ASFM is less than 100%. As a result, it can also be concluded that if losing the marked packets results in arriving less than two marked packets to the destination for a specific flow, that specific flow cannot be traced back. However, the victim can traceback to the attacker using the other marked flows from the same attacker.

8.4.3 False Positives (FP)

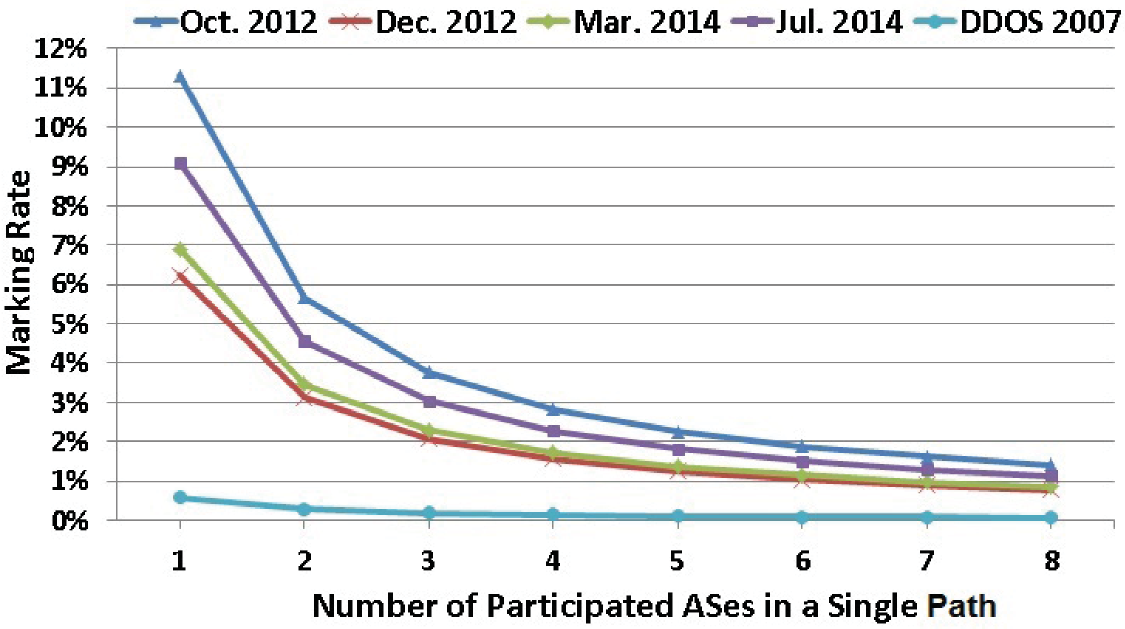


Figure 8.4: The Marking Rate of ASFM with Different numbers of Participating ASBRs in a Single Path on the Six Data Sets

The next performance metric for evaluating IP-Traceback is the number of false positives that the authentication algorithm produces. In the ASFM method, the hash value is added to the marking information to prevent compromised routers from forging the ASBR markings. A false positive occurs when two ASBRs share a common subset of hash values. The evaluation calculates the hash bit length and the probability of collision based on the number of simultaneous attackers. The probability of collision is calculated by the following expression:

$$1 - e^{\frac{-Z(Z-1)}{2X}} \tag{8.8}$$

where X is the number of possible hash values and Z is the number of attackers. Figure 8.5 illustrates the probability of collision for $X = 2^{22}$, when using 22-bit hash values. These results show that the probability of collision increases as the number of attackers increeases.

8.5 Summary

This chapter proposed an AS-level IP-Traceback system, ASFM, which takes advantage of some of the characteristics of BGP to build an AS-level overlay network for IP-Traceback. To establish and maintain this AS-level overlay network, the community attribute of BGP was employed to distribute the information about the ASs which have deployed the traceback system. This feature allows the proposed system to be deployed partially and incrementally in the network regardless of the topology and eliminating the requirement of being deployed in all network routers or all edge routers, which was a usual requirement in previous work. Also, a hash-based authentication method for ASFM was proposed to prevent compromised routers from forging the marking.

The experimental evaluations on real-life traffic data sets suggest that if a relatively small number of ASs take part in the AS-level traceback overlay network, the AS-level route(s) taken by the attacker packets can be identified with a high traceback rate and a very low marking rate. This enables countermeasures to be taken in a distributed way, that is the closest traceback-collaborative ASs with respect to the sources of a DDoS attack, or the ASes that forward more traffic.

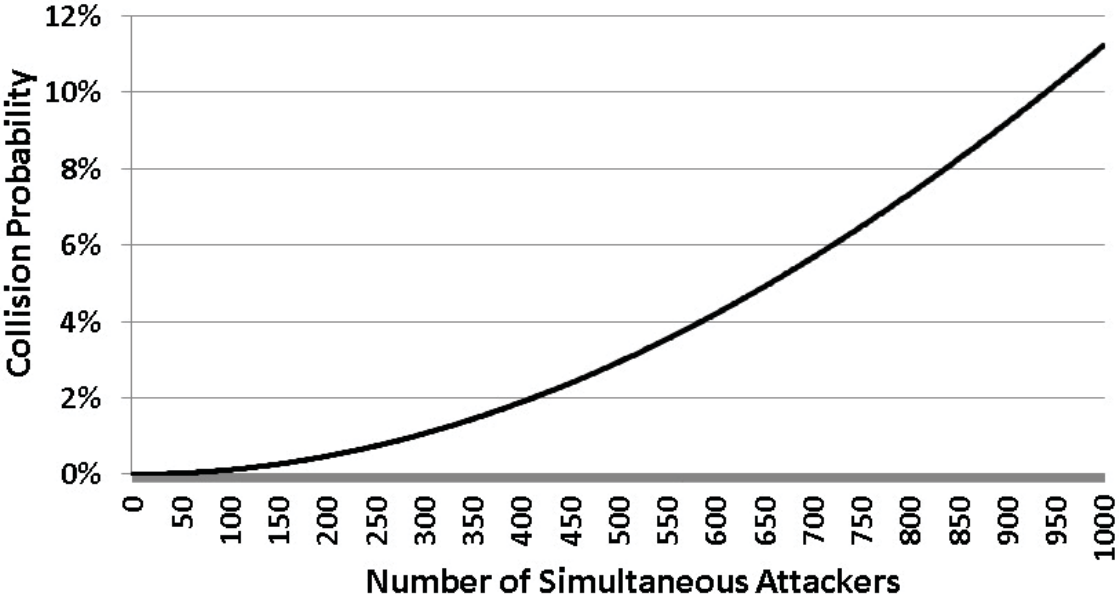


Figure 8.5: The Relationship between the Number of Simultaneous Attackers and the Collision Probability

Chapter 9

Traceback-based Defence against DDoS Flooding Attacks: TDFA

It was first observed by Mahajan et al. [74] that when a DDoS attack occurs, most of the traffic is dropped by upstream routers even before reaching the victim. In this case, when the attack is heavy, it is impossible for the defence system to react to the attack at the victim-end. To mitigate the attack, this chapter proposes a system model for setting up a second defence line at the upstream routers to react to the attack, Figure 9.1. This system can use any of the flow-based IP-Traceback approaches described in the previous chapters. The proposed system improves the throughput of the legitimate traffic during the attack by filtering the attack traffic at the source-end. This approach can be viewed as a distributed algorithm which consists of the components listed below.

- DDoS Detection component: this component is used at the victim-end edge network for recognizing anomalous changes on the network traffic. There are several algorithms and tools for detecting DDoS attacks that can be used for this component such as Snort [13]. The intention here is not to propose a new DDoS detection algorithm.

- IP-Traceback component: the IP-Traceback component employs any of the flow-based IP-Traceback approaches described in the previous chapters for identifying the source of a DDoS attack. As described above, the proposed IP-Traceback mechanisms consist of a lightweight flow marking module running near the source network and a mark decoding module running on the victim-end to infer the source of traffic based on the information extracted from the marked packets.

- Traffic Control component: this component consists of two modules: The traffic Adjustment (TA) and Packet Filtering (PF) modules. The TA module runs on

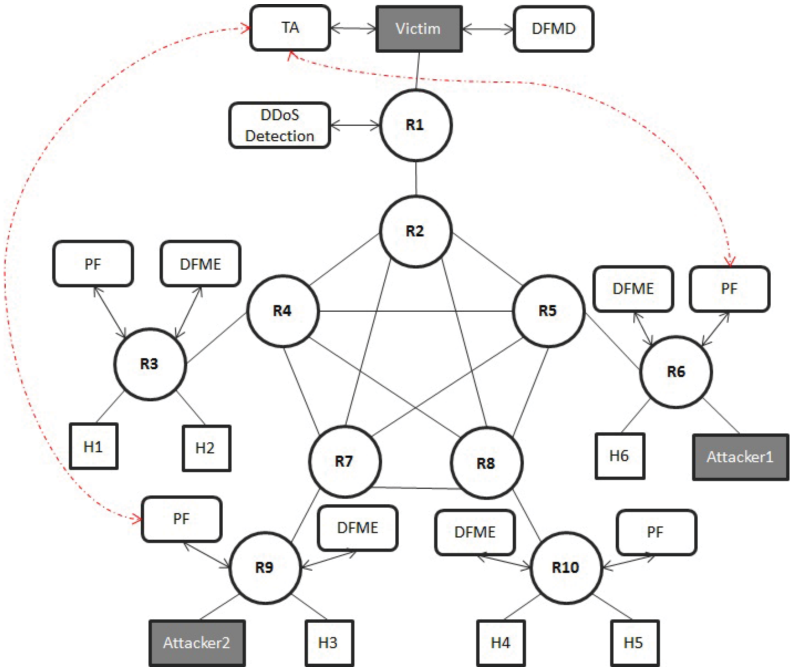


Figure 9.1: The Locations of TDFA Modules on a Sample Network. The Hosts $H1$, $H2$, $H3$, $H4$ and $H5$ are Legitimate Hosts. The TA Module at the Victim end is in Communication with the PF Module on the Edge Routers of the Attacking Networks.

the victim or the border gateway device (e.g. firewall) of the victim network. After finding the source of the attack, using the IP-Traceback component, the TA module sends a request message, which carries attack traffic information, to the defence system near the source network. On the other hand, the PF module runs on every participating router and filters packets that are directed to the victim based on the instructions issued from the TA module.

After analyzing the network traffic on the victim or the border devices (Router 1 in Figure 9.1), the DDoS detection component reports to the IP-Traceback module (eg. DFM Decoding module, $DFMD$) about the ongoing DDoS attack. By collecting the appropriate number of marked packets, the IP-Traceback component infers the IP address of the marking routers (e.g. Routers 6 and 9 in Figure 9.1). Then, the TA module at the victim-end sends a request message to the PF module at the marking router (e.g. Routers 6 and 9). Based on this information, the PF modules at the source-end sets a forwarding rate limit on the packets that are directed to the victim. What follows is a description of the aforementioned TA and PF modules in greater detail.

9.1 Traffic Adjustment (TA) and Packet Filtering (PF) Modules

To control the attack traffic, a rate limit algorithm is proposed which relies on some of the concepts of TCP flow and congestion control algorithms to allocate the bandwidth to the entire incoming traffic from the routers that forward the attack traffic. It is not fair to punish all these routers by setting the same rate limit on them. Therefore the traffic histories of individual routers should be considered by the TA module before sending a rate limit request to a PF module. The proposed defence algorithm can be divided into the phases described below.

- **Connection Establishment**: connections must be established properly in a 3-way handshake process. The TA module at the victim end opens a full duplex session with the PF module at the attacker edge router end for subsequent communications.

- **Congestion Avoidance**: the PF module performs the congestive avoidance phase once it gets a message indicating that the destination of the outgoing packets is under a DDoS flooding attack. In this case, the forwarding rate of the packets directed to the victim is decreased exponentially in such a manner that the forwarding rate is halved at each round trip (i.e. after X round trips, the forwarding rate will be decreased by $2X$ times). This operation continues until the PF module gets a message from the victim indicating that the receiving packet rate is tolerable.

- **Slow Start**: the PF module performs this phase once it is informed that the victim is able to tolerate the current packet forwarding rate. In this case, the packet forwarding rate to the victim is increased linearly. After each increase in the rate, the PF module sends an update message to the TA module indicating that the rate has been changed, and then waits for the confirmation of the new rate from the TA module. In this way, the PF module determines whether the victim is able to tolerate the new rate or not. If the new rate is not confirmed, then the algorithm gets back to the congestion avoidance phase; otherwise, it waits for a given period of time and then increases the forwarding rate again. This process continues until the packet drop rate, calculated by the PF module, is equal to zero. The packet drop rate is calculated using Equation 9.1:

$$DropRate = \frac{Sum(Droped)}{Sum(Droped) + Sum(Sent)} \tag{9.1}$$

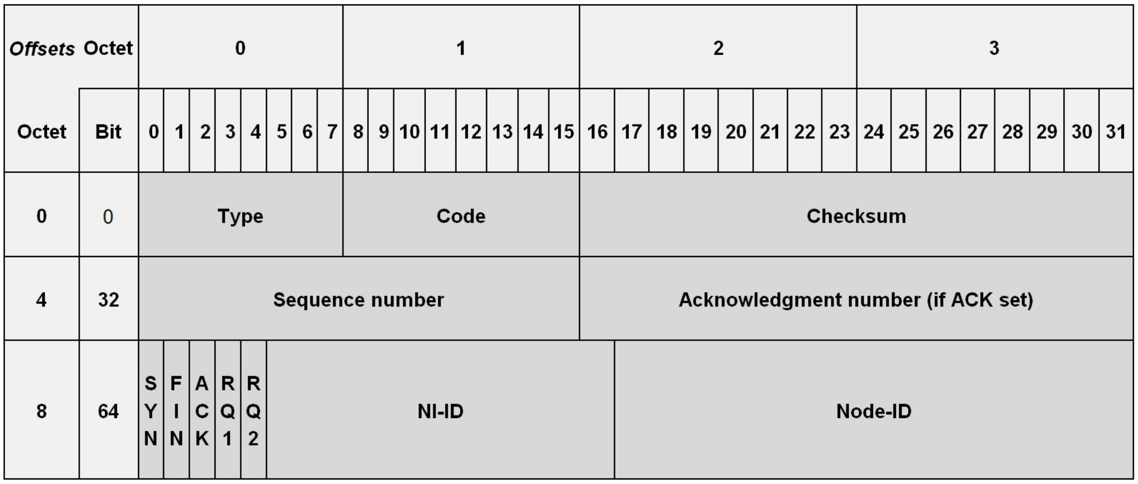
A zero drop rate means that either the flooding attack is finished, or the current rate of packet forwarding can be tolerated by the victim without any impact on the throughput of the legitimate traffic from the other paths.

- **Connection Termination**: If the drop rate is zero, then the PF module removes any rate limit on the forwarding packets destined to the victim and sends an update message to the TA module. If the TA module confirms the new rate limit, the algorithm enters the connection termination phase to close the established virtual circuit. This process can be done by a 3-way FIN/ACK exchange between the PF and the TA modules, which is started by the PF module. If the TA module does not confirm it, the algorithm gets back to the congestion avoidance phase.

Note that once the PF module changes the packet forwarding rate (it may happen in any of the congestion avoidance, slow start, and connection termination phases), it sends an update message to the TA module and waits for a response. If the TA module does not confirm the new rate, then the PF module performs the congestion avoidance phase again. Not confirming the new rate limit is determined by the PF module in any of the three ways listed below.

1. Getting an $ACK + REQ2$ message from the TA module: if the TA module gets an update message from the PF module while the victim is still under the flooding attack, it sends an $ACK + REQ2$ message to the PF.

2. Duplicated acknowledgment or request messages: it means that the victim is under a heavy flooding attack, so the update message has not reached the TA module due to the traffic congestion.



Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Type								Code								Checksum															
4	32	Sequence number																Acknowledgment number (if ACK set)															
8	64	SYN	FIN	ACK	RQ1	RQ2	NI-ID												Node-ID														

Figure 9.2: The Proposed ICMP Message, Designed for Communication between the TA and PF Modules

3. Lack of acknowledgment before time out: it indicates that because of heavy traffic, the bandwidth is full at the victim-end and the TA messages cannot reach the PC module.

9.2 The Proposed System Design

To transmit the control messages between the TA and PF modules, the proposed solution uses the Internet Control Message Protocol (ICMP). ICMP messages are used by network devices, like routers, to send error messages indicating, for example, that a requested service is not available or that a host or a router could not be reached. ICMP can also be used to relay query messages. Figure 9.2 presents the proposed ICMP message which is very similar to the ICMP Timestamp (Type 13) and the ICMP address Mask Request (Type 17) messages [6]. For all the ICMP packets, the first 4 bytes of the header will be consistent. The first byte is for the ICMP type. The second byte is for the ICMP code, and the third and fourth bytes are a checksum of the entire ICMP message. The first byte can be used to address 255 different ICMP Type values, but so far only 28 of them are used and the rest are free to be employed for other purposes. Types 1 and 2 are both free, so type 1 was employed for sending messages from the TA module to the PF module and type 2 for sending messages from the PF module to the TA module. The next 37 bits consists of a

16-bit sequence number, a 16-bit acknowledgment number and five control flag bits including SYN, FIN, ACK, $REQ1$ and $REQ2$. If the SYN flag is set, then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledgement number in the corresponding ACK are equal to this sequence number plus 1. If the SYN flag is not set, then this is the accumulated sequence number of the first data byte of this segment for the current session. If the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end acknowledges the other end's initial sequence number itself. The next two fields are the Interface-ID and Node-ID of the attacker which are identified by the IP-Traceback component. Using these two fields, the TA module sends the attacker identity to the PF module.

	SYN	**FIN**	**ACK**	**REQ1**	**REQ2**
SYN	1	0	0	0	0
SYN+ACK	1	0	1	0	0
FIN	0	1	0	0	0
FIN+ACK	0	1	1	0	0
ACK	0	0	1	0	0
REQ1: Node Filtering	0	0	0	1	0
REQ2: Interface Filtering	0	0	0	0	1
ACK+REQ2	0	0	1	0	1
Update	0	0	0	1	1

Table 9.1: Messages Exchanged between the TA and the PF Modules and their Flags in the ICMP Header

The proposed defence approach is shown in Figure 9.3. All of the messages exchanged between the TA and the PF modules and their flags in the ICMP header are shown in Table 9.1. To establish a connection between the TA module at the victim end, and the PF module at the attacker end edge router, a three-way handshake should be set. Establishing a connection is initiated by the TA module at the victim-end by sending a SYN message to the PF module and the attacker-end edge router (Line 1 in Figure 9.3). The TA module sets the segment's sequence number to a random value A. In response, the PF module replies with a $SYN-ACK$ message. The acknowledgment number is then set to the received sequence number plus one, $A + 1$, and the sequence number that the server chooses for the packet is another random number, B. Finally, the TA module sends an ACK back to the PF module. The sequence number is set to the value of the received acknowledgement, i.e. $A+1$, and the acknowledgement number is set to $B+1$. At this point, both the TA and the PF modules have received an acknowledgment of the connection. In the cases where the IP-Traceback component is able to find the exact node (Line 3 - Figure 9.3), the TA module sends a $REQ1$ message to ask to the PF module to block the entire traffic corresponding to the attacker. This can be done by providing the Node-ID and the NI-ID of the attacker node in the $REQ1$ message. Then, the PF module blocks all packets coming from that specific node and sends an ACK message to the victim. As described above, in some cases the IP-Traceback component can find the ingress interface of the edge router on the attacker side, but not the attacker node behind that edge router. In these cases the proposed algorithm performs the congestion avoidance phase to mitigate the effect of the DDoS attack. Lines 4-8 (Figure 9.3) demonstrate the congestion avoidance phase for decreasing the attack forwarding rate exponentially. Once the PF module receives a $REQ2$ message (Line 4 - Figure 9.3), it halves the forwarding rate (Line 5 - Figure 9.3) and sends an update message to the victim. Then, the TA module decides whether the current rate is still above the load limit (Line 6 - Figure 9.3), or not (Line 8 - Figure 9.3). Lines 13-17 (Figure 9.3) present the connection termination phase. This occurs when either the flooding attack has ended or the victim is able to tolerate the current packet forwarding rate. The connection termination is initiated by the PF module (Line 17 - Figure 9.3), by sending a FIN message to the TA module. In return, the TA module replies with a FIN+ACK message and then the PF module replies with an ACK message.

9.3 Experimental Results

Real life data sets are essential for the evaluation of any defence system. Very few data sets representing real DDoS attacks are publicly available, mostly because of legal and privacy issues. This study is validated using two real world Internet traffic traces, both of which are available in the public domain.

For a real-world DDoS attack, the “CAIDA DDoS Attack 2007” Data set was used [17]. This data set contains approximately one hour of anonymized traffic traces of DDoS attacks captured on August 4, 2007. This data set represents DDoS attacks in

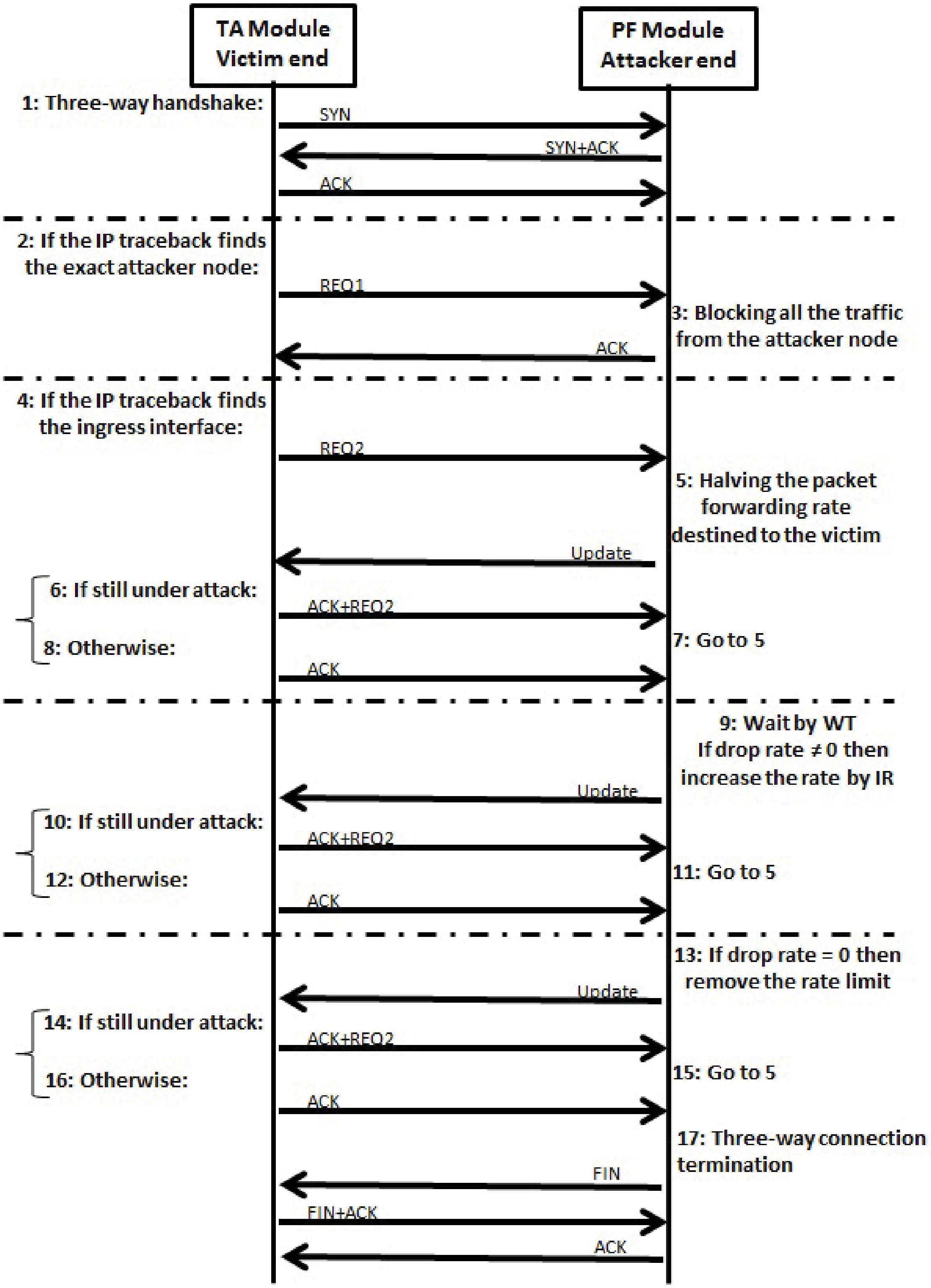

TA Module
Victim end
PF Module
Attacker end
1: Three-way handshake:
SYN
SYN+ACK
ACK
2: If the IP traceback finds the exact attacker node:
REQ1
3: Blocking all the traffic from the attacker node
ACK
4: If the IP traceback finds the ingress interface:
REQ2
5: Halving the packet forwarding rate destined to the victim
Update
6: If still under attack:
ACK+REQ2
7: Go to 5
8: Otherwise:
ACK
9: Wait by WT
If drop rate ≠ 0 then increase the rate by IR
Update
10: If still under attack:
ACK+REQ2
11: Go to 5
12: Otherwise:
ACK
13: If drop rate = 0 then remove the rate limit
Update
14: If still under attack:
ACK+REQ2
15: Go to 5
16: Otherwise:
ACK
17: Three-way connection termination
FIN
FIN+ACK
ACK

Figure 9.3: The Proposed Defence Approach in Operation.

which the attackers attempt to block access to the targetted victim by consuming the victim's computing resources and by consuming all of the bandwidth of the network connecting the victim to the Internet.

On the other hand, "The CAIDA anonymized Internet traces October 2012" Data set [14] is the data set representing the normal traffic used in this work. This data set contains approximately one hour of anonymized passive traffic traces from CAIDA's Equinix-Sanjose monitor on high-speed Internet links.

The characteristics of the CAIDA attack and normal data sets in terms of traffic volume are presented in Figure 9.4 and Figure 9.5, respectively. Figure 9.4 shows the CAIDA DDoS attack in which after the initial 26 minutes, the traffic rate abruptly jump and attains the maximum. This maximum is held for the remainder of the attack period. In DDoS attacks, the compromised machines (bots) are infected by malwares which are programed to direct traffic to the victim at their maximum capacity. In order to accelerate the damage, the bot master triggers the bots simultaneously. On getting the commands from the bot master, bots start to forward the traffic to the victim. Therefore, the victim experiences a sudden burst in the incoming traffic over a relativity short period of time. This causes the victim to exceed its maximum tolerable limits and thereby forces the victim to slow down its normal operations, or even shut down in some cases. In contrast to this, the traffic rate of normal traffic increases gradually over time, Figure 9.5. Hence, it should be possible to use this difference in the rates of incoming traffic to identify the DDoS attack. The DDoS detection component is invoked at regular time intervals to analyze the rate of incoming packets. In this case, such a sudden change in the average incoming traffic is used as evidence of a DDoS attack by the DDoS detection component (in fact human experts and other systems also use this type of information). As stated earlier, the traffic histories of individual traffic sources (differentiating the individual traffic sources is performed by the IP-Traceback component) are considered before sending a rate limit request to the PF module. For example, a change from an average of 40,000 incoming packets over the first 25 minutes versus an average of approximately 1,500,000 packets over the first 27 minutes is taken as a sign of a DDoS attack.

Indeed, a good defence system for such DDoS attacks should be able to mitigate

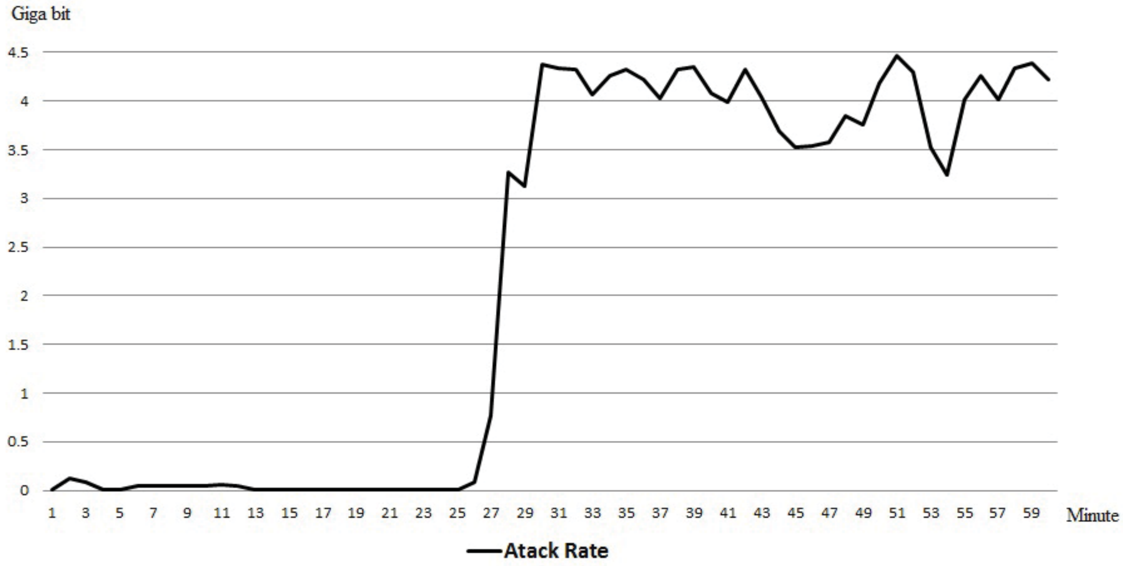


Figure 9.4: The DDoS Traffic Rate (Gb/m) from the CAIDA DDoS 2007 Data Set.

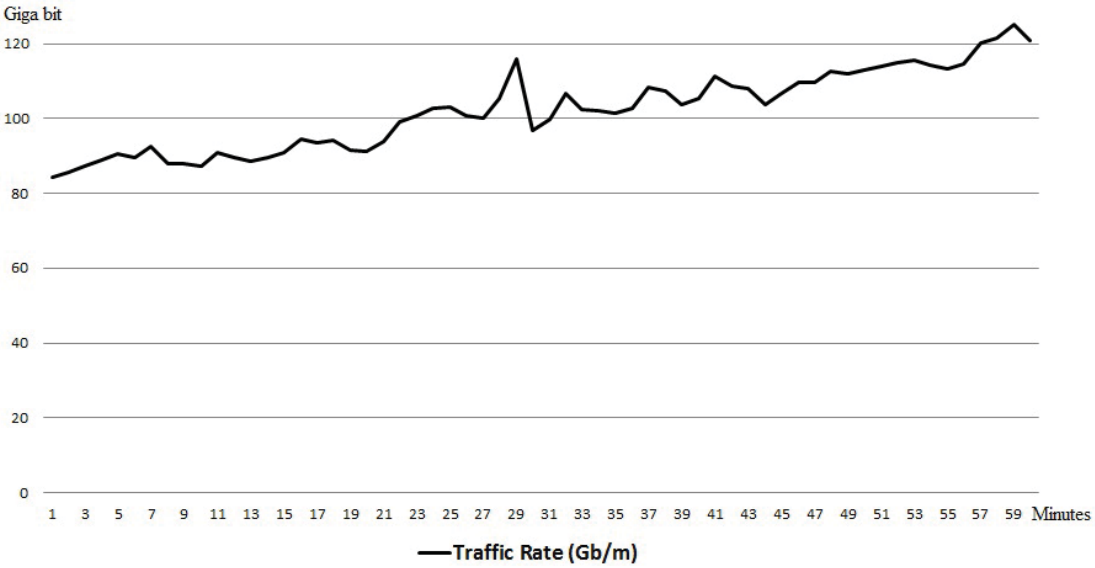


Figure 9.5: The Normal Traffic Rate (Gb/m) from the CAIDA 2012 Data Set.

the impact of the attack and it should maximize the high survival rate of legitimate traffic. To achieve this, the three parameters below are considered.

- **Rate Limit,** RL: RL is the maximum rate of the incoming packets from the attacker that the victim can tolerate. This parameter should be defined by the TA module before it establishes a connection to the PF module.

- **Increasing Rate,** IR: in the slow start phase, the PF module should gradually increase the forwarding rate by IR. IR can be calculated as a fraction of the last forwarding rate before the PF module performs the congestion avoidance phase.

- **Waiting Time,** WT: this parameter indicates the time intervals (in seconds) between each update in the forwarding rate during the slow start phase.

The desirable RL, IR and WT parameters should result in a higher incoming rate for the legitimate packets and a lower incoming rate for the attack packets. Moreover, it should enable a lower communication and bandwidth overhead as well as enabling a lower duration for the slow start phase which incurs less delay in the rate limit removal after the attack is over.

To evaluate the proposed system a test network was set up in the research lab. Two separated LANs, one for the DDoS attack data set and the other for the attack-free data set were set up with both connected to an edge router. For replaying the DDoS data set, advantage was taken of the tcpreplay and tcprewrite open source applications [1] to forward the entire traffic to a computer, in this case the victim node. In addition, two programs were developed, one as the TA module located at the victim end and the other as the PF module located at the edge router.

9.3.1 Attack Rate Reduction

The RL parameter determines the attack rate reduction. The IR and the WT parameters do not have any impact on this. Choosing the value of RL is a trade-off between the receiving rate of legitimate and DDoS packets from the same sub-network. While a lower RL is more effective in reducing the attack, which is desirable, at the same

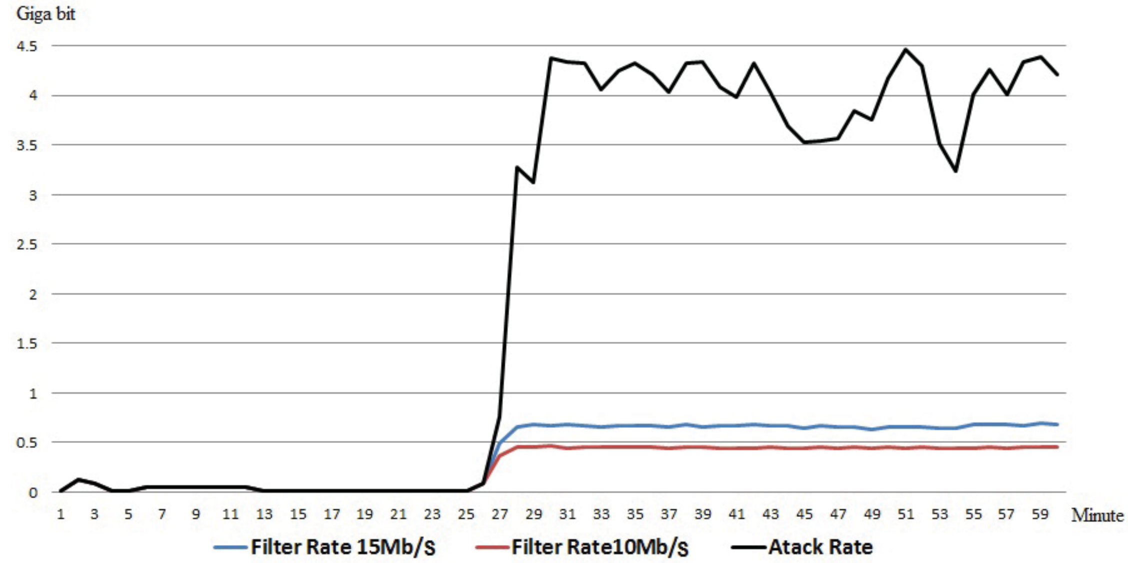


Figure 9.6: The DDoS Traffic Rate (Gb/m) with and without Using the TDFA Defence System.

time it reduces the incoming rate of the legitimate traffic, which is undesirable. Conversely, a higher RL maximizes the rate of legitimate traffic, but increases the impact of a DDoS attack. Selecting the RL depends greatly on different network conditions and traffic patterns such as the victim's bandwidth, the CPU and memory capacity of the victim, the total incoming traffic from the entire individual traffic sources, and the traffic history of the attacker network. This system was evaluated twice, setting the RL parameter to 10 Mb/s and 15 Mb/s, so that whenever the incoming traffic rate exceeds these limits, the TA module begins communication with the PF module to mitigate the impact of the attack. Obviously, choosing higher RL values maximizes the rate of the legitimate traffic, but also increase the impact of a DDoS attack.

Figure 9.6 presents the incoming DDoS traffic rate to the victim, with and without using the proposed TDFA defence system. It shows that the cooperation between the TA and PF modules is started around the 26^{th} minute to control the incoming rate of DDoS packets. The evaluation results show that the proposed system reduces the attack by 88% and 83% by setting $RL = 10$ Mb/s (red line) and $RL = 15$ Mb/s (blue line), respectively.

9.3.2 Communication Overhead

The WT parameter has a decisive role in the amount of communication overhead. As the WT increases (increasing the time interval between each forwarding rate update), the communication between the TA and the PF modules decreases, which is desirable. The duration of the slow start phase can be calculated by Equation 9.2:

$$SlowStartPhaseDuration(second) = \frac{WT}{2IR} \tag{9.2}$$

As can be seen in Equation 9.2, the WT and the IR parameters respectively have direct and inverse relations with the duration of the slow start phase. However, the RL parameter does not have any relation with the duration of the slow start phase. A higher WT and a lower IR incur more delay in the rate limit removal after the attack ends, which is undesirable. On the other hand, the IR value should be kept small, so that the TA and the PF modules can adjust the traffic rate around the RL value more accurately. Therefore, selecting an appropriate value for the WT parameter is a trade-off between the communication overhead and the delay in the rate limit removal after the attack is over.

Since the DDoS traffic rate in the CAIDA datatset reaches the maximum in 50 seconds, the slow start phase duration should not exceed this period. Consequently, in a worst case scenario, 50 seconds after an attack is completed, the PF module removes the rate limit. Note that if the system can tolerate more communication overhead, it is better to reduce the slow start phase duration as much as possible. The evaluation was repeated by setting the WT to 1, 2 and 3 seconds and the IR to 1/20 to keep the slow start phase duration less than 50 seconds (Equation 9.2). Table 9.2 shows the results of these evaluations. As can be seen, increasing or decreasing the WT parameter does not have high impact on the attack rate reduction. However, choosing the lower WT causes more communication and therefore more bandwidth overhead between the TF and the PF modules. On the other hand, a higher WT increases the slow start phase duration and therefore, incurs a delay in the rate limit removal after the attack completion. Selecting an appropriate value of WT is determined by a priority for a lower communication overhead (higher WT value), or a lower latency in the rate limit removal after the attack completion (Lower WT value). Consequently, all the results presented in Table 9.2 are equally good. In short, different values of WT can be set based on the different priorities of the organization that manages the network.

WT	Number of controlling Packets	Volume of controlling Packets (KB)	DDoS Attack Reduction %	Slow Start Phase Duration (second)
1	4023	236	88.023	10
2	2013	118	88.039	20
3	1343	79	88.037	30

Table 9.2: Evaluation Results of the Proposed Approach for Different Values of WT at $RL = 10$ and $IR = 0.5$ Mb/s

Figure 9.7 presents the incoming DDoS traffic rate with and without filtering between the 24th and 30th minutes (the time period that the sudden burst happens) at 1-second intervals. As shown in this figure, once the incoming attack traffic exceeds the RL (10 Mb/sec) value (this is at 26.31 minutes), the PF module performs the congestion avoidance phase to halve the forwarding rate immediately. Then the forwarding rate increases gradually at each WT period (in this case, 2 seconds) by IR (0.5 Mb/s) in the slow start phase. So it takes 20 seconds to reach the RL again (slow start phase duration). Then the congestion avoidance phase is entered. This is the reason why the forwarding rate decreases and increases alternately. This process is performed over and over until the end of the attack which is determined when the PF senses that the incoming rate is less than the forwarding rate (the drop rate is zero).

9.3.3 Processing Overhead

In the proposed system, the filtering decision is made by the TA module at the victim end, so the PF module does not incur any overhead on the edge routers for making such filtering decisions. In addition, filtering is an embedded and a key feature of almost all Internet edge routers, so there is no overhead related to installing such a feature on an edge router. The overhead for the PF modules on the edge routers is as low as computing the packet drop rate and communicating with the TA module by sending the ACK and the update messages. Therefore, a very low processing overhead (in these experiments, mere milliseconds) is incurred at the edge routers.

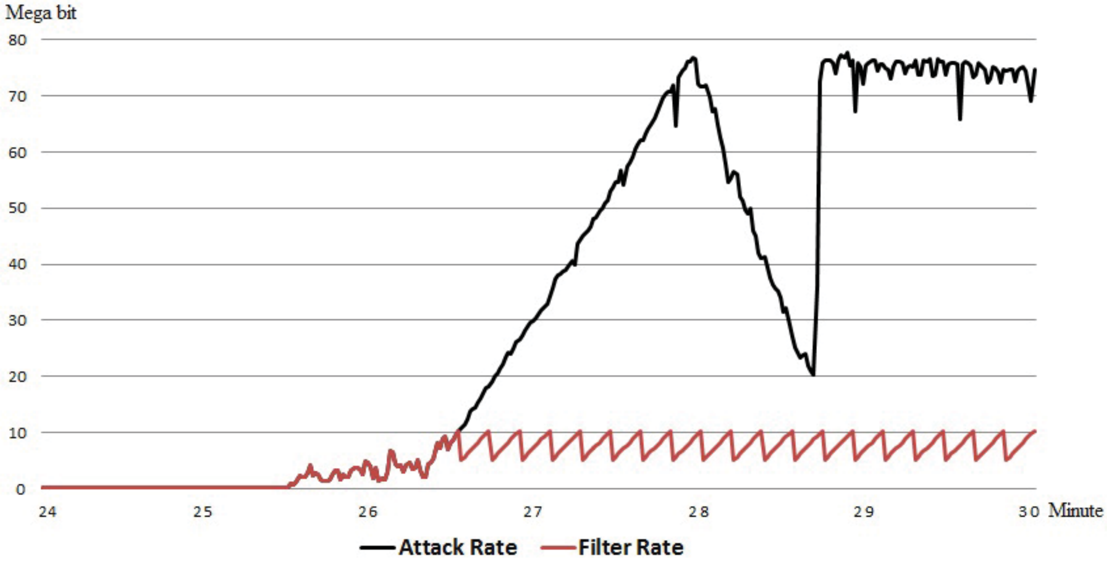


Figure 9.7: Six Minutes (24-30) of the DDoS Traffic Rate (Mb/s) with and without Attack Filtering by TDFA

9.4 Summary

One of the major defence challenges against DDoS attacks is how to control the attack traffic in order to preserve the quality of service of the legitimate traffic. To address this issue, TDFA, a traceback-based defence system has been proposed against flooding attacks which make use of spoofed addresses. The proposed framework uses the strategy of detecting the attack as well as finding the source of the attack at the victim end, and responding to the attack at the source end. These tasks are carried out by the traffic adjustment module (TF), the IP-Traceback decoding module at the victim end, and the packet filtering module (PF) at the edge router of the source end, respectively. As well, a recovery process is triggered when the traffic returns to normal. This work has implemented and evaluated the proposed system, TDFA, against real-world DDoS and normal traffic traces. The results show that TDFA drops the attack packets efficiently at the source end of the attack, while imposing a low overhead on the routers participating in the defence. Moreover, the locations of the defence modules are fully in line with the locations of the IP-Traceback encoding modules, making the scheme more deployable.

Chapter 10

Conclusion

This research has resulted in a modular security framework consisting of three main components: Detection, Traceback and Traffic Control. These three components can work independently as standalone systems, as well as collectively, bound by the proposed framework. The proposed framework aims to facilitate the replacement or addition of security modules without affecting the operation of the system as a whole.

For the Traceback component are proposed five distinct novel flow-based IP-Traceback approaches, namely Deterministic Flow Marking (DFM), Probabilistic Flow Marking (PFM), Unique Flow Marking (UFM), Deterministic Flow Marking for IPv6 Traceback (DFM6) and Autonomous System-based Flow Marking (ASFM). This component enables the identification of the origin of the traffic traversing through the Internet on a per flow basis, regardless of source IP address spoofing. The five IP-Traceback approaches are designed for different network environments with variant network requirements. They all embed a fingerprint in the packets, but each one of them has specific features and performances which make them suitable for various situations. Chapter 4 introduced the first IP-Traceback technique, called Deterministic Flow Marking, DFM, which is a novel real-time three-level traceback method.

Chapter 5 introduced the Probabilistic Flow Marking, PFM, approach. The motivation for proposing PFM is to decrease the marking rate of the DFM approach while keeping the traceback rate as high as possible. This has been performed by selecting the packets randomly and marking them based on the flows to which they belong. If the probability of selecting and marking the packets, P, is set to 1, then PFM works like DFM in terms of the traceback and the marking rates. However, results show that if P is reduced, PFM can decrease the number of selected and marked packets significantly without considerable change in the traceback rate.

Chapter 6 presented another IP-Traceback approach called Unique Flow Marking, UFM. The main concept behind UFM is to find and mark only some of the packets of a flow among all those that have the same Flow-ID. Once the source of some packets in a flow can be found, the source of any other packets in that flow, as well as the source of any other packets in any other flows with the same Flow-ID can also be found. UFM utilizes this fact to decrease the marking rate and to increase the traceback rate (compared with other approaches) simultaneously. Furthermore, the optional authentication scheme for DFM, PFM and UFM provides an efficient authentication of router markings so that even a compromised router cannot forge or tamper markings from other uncompromised routers.

Chapter 7 introduced DFM6, a novel method of Deterministic Flow Marking for IPv6 networks, which is able to traceback up to the attacker node behind a NAT or a proxy server. DFM6 performs by selecting and marking only the first packet of each flow. This results in the two advantages of a high traceback accuracy as well as a0 low processing and marking overhead.

Chapter 8 proposed an AS-level IP-Traceback system, ASFM, which takes advantage of some of the characteristics of BGP to build an AS-level overlay network for IP-Traceback. To establish and maintain this AS-level overlay network, the community attribute BGP was employed to distribute the information about the ASs that have deployed the traceback system. This feature allows the proposed system to be deployed partially and incrementally in the network regardless of the topology and eliminates the requirement of being deployed in all network routers or all edge routers, which was a usual requirement in previous work. Also, a hash-based authentication method was proposed for ASFM to prevent compromised routers from forging the marking.

For the traffic control component, TDFA, a traceback-based defence system against flooding attacks was proposed in Chapter 9. The proposed Traffic Control component defends against DDoS attacks by coordinating between the defence systems at the source and the victim ends. This necessitates communication between the victim and the filtering routers so that the filtering can be located as far away as possible from the victim. Once a DDoS attack is detected by the Detection component, the Traceback component finds the source of the attack. Then the Traffic Control component sends traffic control messages to the edge router of the attack network. When the edge router of the attack network receives the control messages, it will be triggered to adjust the packet forwarding rate to the victim. TDFA filters attack traffic at the source end to eliminate consumption of the computing resources and the bandwidth of the victim. In return, this improves the performance of the system for legitimate services and users.

The proposed systems have been implemented and evaluated using real-world normal and attack traffic traces. The results show that the proposed security framework traces back the attack efficiently and drops the malicious packets near the source of traffic while imposing a low overhead on the routers participating in the defence.

This thesis has integrated basic DFM with the IPv6 network and proposed the DFM6 approach. Also, ASFM integrated the DFM approach with the BGP protocol to have the IP-Traceback at the autonomous system level. For future research directions, it would be interesting to explore how to integrate PFM and UFM in ASFM or IPv6 situations. Even though this research has demonstrated satisfactory performance for PFM and UFM, it is likely such approaches have potential for further investigation. Moreover, testing ASFM and DFM6 under other network conditions and attacks such as application based DDoS will provide further insight into the generalization of the proposed framework.

Bibliography

[1] A. Turner, Tcpreplay Suite. http://tcpreplay.synfin.net/.

[2] CIDR Report. http://www.cidr-report.org/as2.0/. August 31, 2015.

[3] Cyber-attack. https://en.wikipedia.org/wiki/Cyber-attack.

[4] Diffie-Hellman Key Exchange. https://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange.

[5] Google IPv6 Statistics. http://www.google.ca/intl/en/ipv6/statistics.html.

[6] Internet Control Message Protocol. http://en.wikipedia.org/wiki/Internet_Control_Message_Protocol/.

[7] IPv6 Node Requirements, Cisco Enterprise IPv6 Solution. http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white_paper_c11-678658.html.

[8] MAWI (Measurement and Analysis on the WIDE Internet) traffic archive. http://mawi.wide.ad.jp/mawi/.

[9] Null Hypothesis. https://en.wikipedia.org/wiki/Null_hypothesis/.

[10] Overlay Network. https://en.wikipedia.org/wiki/Overlay_network. August 31, 2015.

[11] Regional Internet Registeries. https://www.arin.net/knowledge/rirs.html. August 31, 2015.

[12] Riverbed Technology, Winpcap, The Industry-Standard Windows Packet Capture Library. http://www.winpcap.org/.

[13] Snort Network Intrusion Prevention and Detection System. https://www.snort.org/.

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

[15] The CAIDA Anonymized Internet Traces 2014 Dataset. https://www.caida.org/data/passive/passive_2014_dataset.xml.

[16] The CAIDA Anonymized Internet Traces on World IPv6 Day and World IPv6 Launch Day Dataset. http://www.caida.org/data/passive/passive_ipv6day_and_ipv6launch_dataset.xml.

[17] The CAIDA DDoS Attack 2007 Dataset. http://www.caida.org/data/passive/ddos-20070804_dataset.xml.

[18] The CAIDA Skitter AS Links Dataset. http://www.caida.org/data/active/skitter_aslinks_dataset.xml. August 31, 2015.

[19] Internet Protocol, Version 6 (IPv6) Specification, RFC 2460. http://tools.ietf.org/html/rfc2460, Dec. 1998.

[20] IPv6 Node Requirements, RFC6434. https://tools.ietf.org/html/rfc6434, Dec. 2011.

[21] IPv6 Path MTU Interactions With Link Adaptation, Internet-Draft. https://tools.ietf.org/html/draft-templin-6man-linkadapt-02, Feb. 2015.

[22] Network Ingress Filtering, RFC2827. https://www.ietf.org/rfc/rfc2827.txt, May 2000.

[23] IPv6 Flow Label Specification, RFC6437. http://tools.ietf.org/html/rfc6437, Nov. 2011.

[24] Use of the IPv6 Flow Label as a Transport-Layer Nonce to Defend Against Off-Path Spoofing Attacks, Internet-Draft, Accessed September 14, 2015. http://tools.ietf.org/html/draft-blake-ipv6-flow-label-nonce-02, Oct. 2009.

[25] Textual Conventions for IPv6 Flow Label, RFC3595. http://tools.ietf.org/html/rfc3595, Sep. 2003.

[26] A. Perrig A. Yaar and D. Song. Pi: a Path Identification Mechanism to Defend against DDoS Attacks. *Proc. Symposium on Security and Privacy*, pages 93–107, May 2003.

[27] A. Perrig A. Yaar and D. Song. SIFF: a Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. *IEEE Symposium on Security and Privacy, Berkeley*, pages 130–143, May 2004.

[28] A. Perrig A. Yaar and D. Song. FIT: Fast Internet Traceback. *IEEE International Conference on Computer Communications (INFOCOM)*, 2:1395–1406, March 2005.

[29] A. Perrig A. Yaar and D. Song. StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense. *IEEE Journal on Selected Areas in Commiunication*, 24(10):1853–1863, October 2006.

[30] V. Aghaei-Foroushani and A.N. Zincir-Heywood. IP Traceback through (Authenticated) Deterministic Flow Marking: an Empirical Evaluation. *EURASIP Journal on Information Security*, 5:., 2013.

[31] V. Aghaei-Foroushani and N. Zincir-Heywood. On Evaluating IP Traceback Schemes: A Practical Perspective. *IEEE International Workshop on Cyber Crime (IWCC 2013)*, pages 127–134, May 2013.

[32] M. Alenezi and M.J. Reed. Efficient AS DoS Traceback. *International Conference on Computer Applications Technology (ICCAT2013)*, pages 1–5, Jan. 2013.

[33] Syed Obaid Amin and Choong Seon Hong. On IPv6 Traceback. *The 8th International Conference on Advanced Communication Technology (ICACT)*, pages 2139–2143, Feb. 2006.

[34] Syed Obaid Amin, Muhammad Shoaib Siddiqui, and Choong Seon Hong. A Novel IPv6 Traceback Architecture Using COPS Protocol. *annals of telecommunications - annales des tlcommunications*, Apr. 2008.

[35] H. Beitollahi and G. Deconinck. Analyzing well-known Countermeasures against Distributed Denial of Service Attacks. *Journal on Computer Communication*, 35(11):1312–1332, June 2012.

[36] A. Belenky and N. Ansari. IP Traceback with Deterministic Packet Marking. *IEEE Communications Letters*, 7(4):162–164, April 2003.

[37] A. Belenky and N. Ansari. On IP Traceback. *IEEE Communications Magazine*, 41(7):142–153, July 2003.

[38] A. Belenky and N. Ansari. On Deterministic Packet Marking. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(10):2677–2700, July 2007.

[39] S.M. Bellovin. ICMP Traceback Messages. *IETF Draft*, March 2000.

[40] H. Burch and B. Cheswick. Tracing Anonymous Packets to their Approximate Source. *USENIX LISA Conference*, pages 319–327, 2000.

[41] A. Castelucio and A. Ziviani. An AS-Level Overlay Network for IP Traceback. *IEEE Network*, 23:36–41, 2009.

[42] M. Franklin D. Dean and A. Stubblefield. An Algebraic Approach to IP Traceback. *Transactions on Information and System Security (TISSEC)*, 5(2):119–137, May 2002.

[43] N. Ansari D. Wei. Implementing IP Traceback in the Internet: an ISP Perspective. *Proceedings of 3rd Annual IEEE Workshop on Information Assurance*, pages 326–332, 2002.

[44] S. Su D. Yan, Y. Wang and F. Yang. A Precise and Practical IP Traceback Technique Based on Packet Marking and Logging. *Journal of Information Science and Engineering*, 28(3):453–469, May 2012.

[45] Xuan-Hien Dang, Emil Albright, and Abdullah A. Abonamah. Performance Analysis of Probabilistic Packet Marking in IPv6. *IEEE International Conference on Local Computer Networks (LCN'04)*, pages 3193–3202, Nov. 2007.

[46] E. W. Dijkstra. A Note on two Problems in Connexion with Graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[47] A.C. Snoeren et al. Single-Packet IP Traceback. *IEEE/ACM Transactions on Networking*, 10(6):721–734, December 2002.

[48] J. Li et al. Large-Scale IP Traceback in Highspeed Internet: practical Techniques and Theoretical Foundation. *IEEE/ACM Transactions on Networking*, 16(6):1253–1266, December 2008.

[49] S. Matsuda et al. Design and Implementation of Unauthorized Access Tracing System. *Symposium on Applications and the Internet (SAINT)*, pages 74–81, January/February 2002.

[50] S. Savage et al. Network Support for IP Traceback. *IEEE/ACM Transactions on Networking*, 9(3):226–237, June 2001.

[51] D. Evans and D. Larochelle. Improving Security using Extensible Lightweight Static Analysis. *IEEE Software*, 19(1):42–51, February 2002.

[52] Brian Everitt. *The Cambridge Dictionary of Statistics*. Cambridge University Press, Cambridge, UK New York, 1998.

[53] M.S. Fallah and N. Kahani. TDPF: a Traceback-based Distributed Packet Filter to Mitigate Spoofed DDoS Attacks. *Security and communication networks, Wiley Online Library, DOI: 10.1002/sec.725*, February 2013.

[54] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing. *RFC 2827, Internet Engineering Task Force (IETF)*, 2000.

[55] A. V. Vasilakos G. Yao, J. Bi. Passive IP Traceback: Disclosing the Locations of IP Spoofers from Path Backscatter. *IEEE Transactions on Information Forensics and Security*, 10(3):471–484, March 2015.

[56] Z. Zhou G. Yao, J. Bi. Passive IP Traceback: Capturing the Origin of Anonymous Traffic Through Network Telescopes. *ACM SIGCOMM Computer Communication Review*, 40(4):413–414, August 2010.

[57] Z. Gao and N. Ansari. Tracing Cyber Attacks from the Practical Perspective. *IEEE Communications Magazine*, 43(5):123–131, May 2005.

[58] Z. Gao and N. Ansari. A Practical and Robust Inter-Domain Marking Scheme for IP Traceback. *The International Journal of Computer and Telecommunications Networking*, 51(3):732–750, February 2007.

[59] C. Gong, T. Le, T. Korkmaz, and K. Sarac. Single Packet IP Traceback in AS-level Partial Deployment Scenario. *International Journal of Security and Networks*, 2:95–108, Mar. 2007.

[60] M.T. Goodrich. Efficient Packet Marking for large-Scale IP Traceback. *ACM Conference on Computer and Communications Security (CCS)*, pages 117–126, November 2002.

[61] S.T. Chanson H. Lam, C. Li and D. Yeung. A Coordinated Cetection and Cesponse Scheme for Distributed Denial-of-Service Attacks. *IEEE International Conf. on Communications*, 5:2165–2170, June 2006.

[62] T. Hongcheng and B. Jun. An Incrementally Deployable Flow-Based Scheme for IP Traceback. *IEEE Communications Letters*, 16(7):1140–1143, July 2012.

[63] Y. Rayudu K. Deepthi1, A. Swapna. A Novel Passive IP Approach for Path File Sharing Through BackScatter in Disclosing the Locations. *International Journal of Computer Science*, October 2015.

[64] S. Kent, C. Lynn, , and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas in Communications*, 18(2):582–592, Apr. 2001.

[65] RH Kim, JH Jang, and H. Youm. An Efficient IP Traceback Mechanism for the NGN based on IPv6 Protocol. *(JWIS)*, Aug. 2009.

[66] Sarvamangala D. R Manoj C Jagatap 1. Tracking Real Source of Attack in IP Networks and Protection Against Threats. *International Journal of Advanced Scientific Research and Publications*, 2(3):41–43, 2016.

[67] N. C. Fernandes M.D.D. Moreira, R. P. Laufer and O. C. M. B. Duarte. A Stateless Traceback Technique for Identifying the Origin of Attacks from a Single Packet. *IEEE International Conference on Communications (ICC)*, pages 1–6, June 2011.

[68] M. Talekar S. Tapkir D. Khade N. Humbir P. Deshpande, V. Patil. Forensic Spoofer Location Detection Using Passive IP Traceback Techniques. *International Research Journal of Advanced Engineering and Science*, 1(2):40–43, 2016.

[69] Ashwani Parashar and Ramaswamy Radhakrishnan. Improved Deterministic Packet Marking Algorithm for IPv6 Traceback. *International Conference on Electronics and Communication Systems (ICECS)*, pages 1–4, Feb. 2014.

[70] K. Park and H. Lee. On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack. *IEEE International Conference on Computer Communications (INFOCOM)*, 1:338–347, April 2001.

[71] V. Paruchuri, A. Durresi, and L. Barolli. FAST: Fast Autonomous System Traceback. *21st International Conference on Advanced Information Networking and Applications (AINA2007)*, pages 498–505, May 2007.

[72] V. Paruchuri, A. Durresi, R. Kannan, and S.S. Iyengar. Authenticated Autonomous System Traceback. *18th International Conference on Advanced Information Networking and Applications (AINA2004)*, 1:406–413, 2004.

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

[74] S. Floyd J. Ioannidis V. Paxson R. Mahajan, S. Bellovin and S. Shenker. Controlling High Bandwidth Aggregates in the Network. *Technical report, ACIRI and AT and T Labs Research*, February 2001.

[75] H. Mohammadi N. Yazdani R. Shokri, A. Varshovi and B. Sadeghian. DDPM: Dynamic Deterministic Packet Marking for IP Traceback. *IEEE International Conference on Networks (ICON)*, 2:1–6, September 2006.

[76] S. K. Rayanchu and G. Barua. Tracing Attackers with Deterministic Edge Router Marking (DERM). *International Conference on Distributed Computing and Internet Technology (ICDCIT)*, pages 400–409, December 2004.

[77] Y. Rekhter and T. Li. A Border Gateway Protocol 4, RFC 1771, 1995.

[78] D. Massey S.F. Wu, L. Zhang and A. Mankin. On Design and Evaluation of Intention-Driven ICMP Traceback. *International Conference on Computer Communications and Networks*, pages 159–165, October 2001.

[79] D.X. Song and A. Perrig. Advanced and Authenticated Marking Schemes for IP Traceback. *IEEE International Conference on Computer Communications (INFOCOM)*, 2:878–886, April 2001.

[80] Q. Song and S. Chen. Perimeter-based Defense against High Bandwidth DDoS Attacks. *IEEE Transactions on Parallel and Distributed Systems*, 16(9):526–537, June 2005.

[81] R. Stone. CenterTrack: an IP Overlay Network for Tracking DoS Floods. *9th Conf. USENIX Security Symposium, Denver*, (199-212), 2000.

[82] W. Timothy Strayer, Christine E. Jones, Fabrice Tchakountio, and Regina Rosales Hain. SPIE-IPv6: Single IPv6 Packet Traceback. *IEEE International Conference on Local Computer Networks (LCN'04)*, pages 118–125, Nov. 2004.

[83] M. Sung and J. Xu. IP Traceback-based Intelligent Packet Filtering: a Novel Technique for Defending against Internet DDoS Attacks. *IEEE Transactions on Parallel and Distributed Systems*, 14(9):861–872, September 2003.

[84] I. A. A. Guru T. Subbulakshmi and S. M. Shalinie. Attack Source Identification at Router Level in Real Time Using Marking Algorithm Deployed in Programmable Routers. *International Conference On Recent Trends In Information Technology*, pages 79–84, 2011.

[85] Animesh Tripathy, Jayanti Dansana, and Debi Prasad Mishra. A Secure Packet Marking Scheme for IP Traceback in IPv6. *the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 656–659, Aug. 2012.

[86] P.N. Klein T.W. Doeppner and A. Koyfman. Using Router Stamping to Identify the Source of IP Packets. *ACM Conference on Computer and Communications Security (CCS)*, pages 184–189, 2000.

[87] J. WU X. LI W. LIU, H. DUAN. Improved Marking Model ERPPM Tracing Back to DDoS Attacker. *Third International Conference on Information Technology and Applications (ICITA'05)*, 2:759–762, July 2005.

[88] T. Wan, P.C.V. Oorschot, and E. Kranakis. A Selective Introduction to Border Gateway Protocol (BGP) Security Issues. *NATO Advanced Studies Institute on Network Security and Intrusion Detection*, 2007.

[89] B. Wang and H. Schulzrinne. A Denial-of-Service-Resistant IP Traceback Approach. *Ninth International Symposium on Computers and Communications (ISCC)*, 1:351–356, June 2004.

[90] P. Dhar A. El Saddik Y. Chen, S. Das and A. Nayak. Detecting and Preventing IP-Spoofed Distributed DoS Attacks. *International Journal of Network Security*, 7(1):69–80, July 2008.

[91] H. Chen Y. Tseng and W. Hsieh. Probabilistic Packet Marking with Non-Preemptive Compensation. *IEEE Communications Letters*, 8(6):359–361, June 2004.

[92] W. Zhou Y. Xiang and M. Guo. Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks. *IEEE Transactions on Parallel and Distributed Systems*, 20(4):567–580, April 2009.

[93] M. Zulkernine Y. You and A. Haque. A Distributed Defense Framework for Flooding-based DDoS Attacks. *3rd Int. Conf. Availability, Reliability and Security, Barcelona*, pages 245–252, March 2008.

[94] M. Yang. RIHT: A Novel Hybrid IP Traceback Scheme. *IEEE Transactions on Information Forensics and Security*, 7(2):789–797, April 2012.

[95] Xinyu Yang and Ting Ma. A Link Signature based DDoS Attacker Tracing Algorithm Under IPv6. *Future Generation Communication and Networking (FGCN)*, pages 50–55, Dec. 2007.

[96] You ye Sun, Cui Zhang, Shao qing Meng, and Kai ning Lu. Modified Deterministic Packet Marking for DDoS Attack Traceback in IPv6 Network. *11th IEEE International Conference on Computer and Information Technology*, pages 245–248, Aug. 2011.

[97] X. Yuan Z. Duan and J. Chandrashekar. Controlling IP Spoofing through Interdomain Packet Filters. *IEEE Transactions on Dependable and Secure Computing*, 5(1):22–36, March 2008.

[98] C. Kim Y. Ke-xin Z. Jian-Qi, F. Feng and L. Yan-Heng. A DoS Detection Method based on Composition Self-similarity. *KSII Transactions on Internet and Information Systems*, 6(5):1463–1478, May 2012.

Appendix A

Publications

Conference Papers:

1. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, "Autonomous System-based Flow Marking Scheme for IP-Traceback", IEEE/IFIP Network Operations and Management Symposium (NOMS 2016), Istanbul, Turkey, April 2016.

2. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, "Deterministic flow marking for IPv6 traceback (DFM6)", 11th International Conference on Network and Service Management (CNSM), pp. 270-273, Barcelona, Spain, November 2015.

3. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, "Probabilistic flow marking for IP-Traceback (PFM)", 7th International Workshop on Reliable Networks Design and Modeling (RNDM), pp. 229-236, Munich, Germany, October 2015.

4. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, "Investigating unique flow marking for tracing back DDoS attacks", IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 762-765, Ottawa, Canada, May 2015.

5. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, "A Proxy Identifier Based on Patterns in Traffic Flows", IEEE 16th International Symposium on High Assurance Systems Engineering (HASE), pp. 118-125, Daytona Beach, Florida, USA, January 2015.

6. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, "TDFA: Traceback-based Defense against DDoS Flooding Attacks", 28th IEEE International Conference on Advanced Information Networking and Applications (AINA2014), Victoria, Canada, May 2014.

7. Yasemin Gokcen, Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, “Can we identify NAT behavior by analyzing Traffic Flows?”,IEEE International Workshop on Cyber Crime with IEEE S&P (IWCC 2014), San Jose, May 2014.

8. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, “On evaluating IP-Traceback schemes: a practical perspective”, IEEE International Workshop on Cyber Crime with IEEE S&P (IWCC 2013), pp. 127134, San Francisco, May 2013.

9. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, “Investigating Application Behavior in Network Traffic Traces”, 2013 IEEE Symposium Series on Computational Intelligence (SSCI 2013), pp. 72-79, Singapore, April 2013.

10. Vahid Aghaei-Foroushani and A. Nur Zincir-Heywood, “Deterministic and authenticated flow marking for IP-Traceback”, The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 397-404, Barcelona, March 2013.

Journal Paper:

1. V. Aghaei-Foroushani and A.N. Zincir-Heywood, “IP-Traceback through (authenticated) deterministic flow marking: an empirical evaluation”, EURASIP Journal on Information Security, 2013.