COMPUTERS & SECURITY XXX (2013) 1–20

Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks

Dongwon Seo [a], Heejo Lee [a,*], Adrian Perrig [b]

[a] Department of Computer Science and Engineering, Korea University, Seoul 136-713, Republic of Korea
[b] Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 15 March 2013
Received in revised form
30 July 2013
Accepted 2 September 2013

Keywords:
DDoS attack defense
Filter-based defense
Filter scheduling
Filter propagation
Adaptive packet marking

ABSTRACT

Distributed denial-of-service (DDoS) attacks are considered to be among the most crucial security challenges in current networks because they significantly disrupt the availability of a service by consuming extreme amount of resource and/or by creating link congestions. One type of countermeasure against DDoS attacks is a filter-based approach where filter-based intermediate routers within the network coordinate with each other to filter undesired flows. The key to success for this approach is effective filter propagation and management techniques. However, existing filter-based approaches do not consider effective filter propagation and management. In this paper, we define three necessary properties for a viable DDoS solution: how to practically propagate filters, how to place filters to effective filter routers, and how to manage filters to maximize the efficacy of the defense. We propose a novel mechanism, called Adaptive Probabilistic Filter Scheduling (APFS), that effectively defends against DDoS attacks and also satisfies the three necessary properties. In APFS, a filter router adaptively calculates its own marking probability based on three factors: 1) hop count from a sender, 2) the filter router's resource availability, and 3) the filter router's link degree. That is, a filter router that is closer to attackers, has more available resources, or has more connections to neighbors inserts its marking with a higher probability. These three factors lead a victim to receive more markings from more effective filter routers, and thus, filters are quickly distributed to effective filter routers. Moreover, each filter router manages multiple filters using a filter scheduling policy that allows it to selectively keep the most effective filters depending on attack situations. Experimental results show that APFS has a faster filter propagation and a higher attack blocking ratio than existing approaches that use fixed marking probability. In addition, APFS has a 44% higher defense effectiveness than existing filter-based approaches that do not use a filter scheduling policy.

1. Introduction

Distributed denial-of-service (DDoS) attacks are still one of the most significant threats in computer networks even though this type of attack has been around for over a decade. The first well-documented attack was launched in 1999 using at least 227 hosts to flood a single computer in University of Minnesota (Kessler, 2000). Recently, DDoS attacks have been used for

* Corresponding author.
E-mail addresses: aerosmiz@korea.ac.kr (D. Seo), heejo@korea.ac.kr (H. Lee), adrian.perrig@inf.ethz.ch (A. Perrig).

0167-4048/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cose.2013.09.002

other motives, such as racketeering and political motivations. A typical example of a racketeering or extortion attack is the one that was launched against the Itembay web site, a site where game items are traded, which was paralyzed by DDoS attacks in September 2007 (Digital Chosun, 2007). A politically motivated set of DDoS attacks, so called 7.7 DDoS attacks, occurred on July 7, 2009, in South Korea and the U.S. (Korea Times, 2009). The attacks targeted major portal sites in South Korea and several U.S. government agencies. Another similar DDoS attack, launched October 26, 2011, targeted national election watchdog web sites (Korea Herald, 2012) in South Korea. In the report by Arbor Networks in 2013 (Arbor Networks, 2013), DDoS attacks remain the most concerned operational threat for network operators. From these trends, it can be seen that DDoS attacks are still prevalent and have many reasons to continue in the current Internet.

Many approaches have been proposed to defend against DDoS attacks, and they can be categorized into three groups according to deployment location. First, source-end defense approaches provide the most effective benefits because they block malicious traffic before it can spread (Kang et al., 2005; Mirkovic et al., 2002). However, their deployments are problematic because the attackers are generally widely spreaded. Second, victim-end defense approaches, such as Intrusion Detection/Prevention System (IDS/IPS) (Paxson, 1999; Roesch, 1999) or flow-based detection (Braga et al., 2010), protect the victim's side networks from DDoS attacks. However, they only cover the victim's server or a small network area, and cannot counter link resource attacks (e.g., link congestion). Lastly, intermediate network defense approaches utilize intermediate routers within the network, and this is the most effective location to defend against both victim's resource and network link resource attacks (Argyraki and Cheriton, 2005, 2009; Mahajan et al., 2002; Yaar et al., 2004; Yau et al., 2005). In these approaches, filters that determine whether specific packets should be dropped or forwarded are installed on intermediate filter routers to block undesired flows. Typically a path between an attacker and a victim traverses through many of these routers, and they coordinate amongst each other to defend against DDoS attacks. If the coordination is not done properly that is filters are not exchanged and distributed properly, this in-network approach will not be effective. As a result, filter propagation and filter management are major challenges in filter-based approaches.

Active Internet Traffic Filtering (AITF) (Argyraki and Cheriton, 2005, 2009), considered as one of the most complete works on filter-based DDoS defenses (Liu et al., 2008), addresses these challenges. AITF utilizes two techniques: a record route to propagate filters and rate limiting to handle the number of filters. However, the record route for filter propagation has to use IP options, which is mostly disabled in current Internet routers. Although rate limiting can reduce the number of filters, it cannot distinguish which filter should or should not be installed. Furthermore, it propagates filters on a hop-by-hop basis, which can delay filter propagation to the attacker in scenarios where there are many filter routers on the path between the attacker and the victim.

Consequently, a viable solution is needed to address the following challenges: how to practically propagate filters along attack paths (path identification), how to place filters to effective filter routers (filter propagation), and how to manage many filters under limited router resources (filter management). We define resolving these challenges as a ***filter scheduling problem***. Our preliminary work (Seo et al., 2011) attempted to address the problem and showed the effectiveness of filter management; however, the effectiveness of its defense can be further enhanced to fully address the problem by considering effective filter location at proper filter routers.

In this paper, we propose a novel filter-based approach, called Adaptive Probabilistic Filter Scheduling (APFS), that resolves the filter scheduling problem through adaptive packet marking and a filter scheduling policy. In APFS, each filter router probabilistically inserts its own marking into unused IP header fields. In APFS, each filter router probabilistically inserts its own marking into unused IP header fields based on its own adaptive probability. This marking probability varies from router to router with respect to its filtering effectiveness. The filtering effectiveness is determined by three factors that can directly affect attack blocking performance: 1) how close the filter router is to the attacker (HOP), 2) how many filters the filter router can accept (RES), and 3) how many links the filter router has (DEG). These three factors lead a victim to receive more markings from more effective filter routers, and thus, APFS shows faster filter propagation than the hop-by-hop basis.

In addition, because filter routers can suffer from a flood of filter requests, we utilize a filter scheduling policy for filter selection. The filter scheduling policy decides which filter should be installed and which filter should be evicted from a filter router. Each filter router computes filter scores (priorities) reflecting how frequently its filters are used and how recently they have been used; therefore, the filter router keeps actively used filters, and it evicts useless filters. To the best of our knowledge, no existing work addresses the filter scheduling problem. We attempt to solve it using adaptive probabilistic packet marking and a filter scheduling policy.

In practice, no solution can be deployed to all the networks in a short time; we designed APFS such that it provides benefits for early adopters and supports incremental deployment. Each filter router solely computes all the necessary operations, marking probability and filter scheduling calculations, without any knowledge from other nodes and networks. That is, APFS does not necessarily require inter-AS cooperation. Thus, even if APFS are deployed to a single intra-AS network with low deployment rate such as 10%, APFS shows 40% attack blocking ratio. Surely, as the deployment increases, APFS shows higher attack blocking ratio.

To verify our proposed solution, we created Internet-like topologies to simulate APFS and compared it against other existing filter-based approaches. APFS had faster filter propagation and a higher attack blocking ratio than existing approaches that use fixed marking probability. In addition, APFS had a 44% higher defense effectiveness than existing filter-based approaches, such as AITF, that does not use a filter scheduling policy. The main contributions of our work are fourfold:

1. We define the filter scheduling problem, and design APFS, which effectively blocks attack traffic. APFS is able to defend against both victim resource attacks and link

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

resource attacks. It reduces attack traffic at the victim by as much as 78%, even when attackers attempt to saturate the link with attack traffic.
2. APFS increases the filtering performance than existing filter-based DDoS defenses in terms of attack blocking ratio and filter propagation speed. APFS propagates filters to the most effective locations among filter routers. Each filter router determines its filtering effectiveness based on three factors (HOP, RES, and DEG) and varies its marking probability. This results in victims receiving selective markings and filters being quickly sent to effective filter routers.
3. APFS utilizes a filter scheduling policy to manage multiple filters on filter routers that have limited resources. A filter router may receive multiplicity of filters from victims, and APFS determines the best-k filters to maximize the blocking of undesired flows.
4. APFS is designed to be easily adopted to legacy networks. Since a single filter router can solely process all the necessary operations without cooperation from other nodes and networks, it provides benefits for early adopters and supports incremental deployment.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss how existing DDoS defenses operate and look at the challenges facing them. Section 3 presents our problem definition and assumptions. Section 4 explains the design principle of APFS and its four phases of operation. Section 5 examines the effectiveness of APFS in terms of its parameters. Section 6 presents our experimental results. In Section 7, we analyze the architecture of APFS and discuss potential threats and countermeasures. Finally, Section 8 presents our conclusions.

2. Background

In this section, we introduce related work that gives a basic insight into the mechanism underlying the design of APFS. We then briefly explain how filter-based DDoS defense approaches operate and look at the several challenges.

2.1. DDoS defense at intermediate routers

DDoS defense at intermediate routers consists of two necessary operations: path identification and traffic control as shown in Fig. 1. For path identification, packet marking and message generation are widely used for path identification, and capability-based and filter-based approaches have been proposed for traffic control at intermediate routers.

Packet marking has an advantage that it can be easily embedded in existing protocols such as IP. The filter-based approach is designed to block malicious traffic, while the capability-based approach is designed to guarantee legitimate traffic. Even though the capability-based approach enables several legitimate channels to survive DDoS attacks, many end hosts still suffer from link congestion. Consequently, the filter-based approach was proposed as a means of blocking malicious traffic by filter propagation among intermediate routers.

We briefly introduce the three mechanisms, packet marking, capability-based DDoS defense, and filter-based DDoS defense, that utilize the intermediate network and motivated the design of our works.

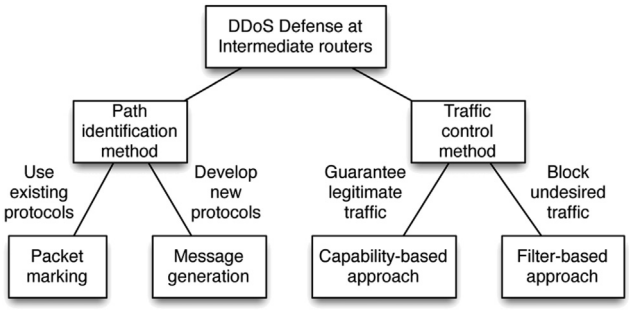


Fig. 1 – Path identification and traffic control are essential operations in DDoS defense using intermediate routers. For path identification, packet marking has an advantage that it easily embedded on exsisting protocols such as IP. For traffic control, the filter-based approach is designed to block malicious traffic. These two methods gives basic insights of APFS.

2.1.1. Packet marking

One of the basic mechanisms in APFS is packet marking. Ever since Savage et al. (Savage et al., 2000) proposed the packet marking for IP traceback, much related research has been conducted (Aljifri et al., 2003; Bai et al., 2004; Belenky and Ansari, 2007; Sung and Xu, 2003). Their common purpose is to search for methods to identify the path of a specific packet. Leveraging on this information, Path identification (Pi) (Yaar et al., 2003) defends against DDoS attacks. Pi verifies packets using path information and modifies TTL values. BASE (Lee et al., 2007) was proposed as an anti-spoofing mechanism. In BASE, each router marks Message Authentication Code (MAC) using hash chaining into IP packets, and packets that are marked with incorrect values are discarded.

2.1.2. Capability-based DDoS defense

The capability-based approach restricts the bandwidth of each sender. In this approach, each receiver grants the traffic that it desires to receive, before a sender establishes a connection. Yaar et al. (2004) presented SIFF, in which an end-host selectively stops individual flows. It divides network traffic into two classes, privileged and unprivileged, to protect privileged traffic from DDoS attacks. TVA, proposed by Yang et al. (2008), attempts to achieve a more complete design and protection against possible attacks, such as flooding of the setup channel, and exhausting of the router state. The capability-based approach focuses on guaranteeing bandwidth for legitimate users, while the filter-based approach focuses on blocking malicious traffic. Although these two intermediate router-based DDoS defense schemes can be combined to reduce malicious traffic and to increase legitimate traffic, our focus in this paper is on the filter-based approach.

2.1.3. Filter-based DDoS defense

Much research has been conducted on the filter-based approach to DDoS attacks. iTrace (Bellovin et al., 2001) utilizes ICMP messages to forward filter router information

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

to victims. However, it generates additional packets to traceback attack paths and incurs message authentication issues. Mahajan et al. (2002) proposed Pushback, a scheme that combines a rate limiting and filter propagation. However, the scheme does not know where the pushback routers are deployed (the necessity for topological knowledge), so partial deployment is not possible. StopIt (Liu et al., 2008) also adopts the undesired flow filtering idea at routers, and installs dedicated servers that forward the filters to the attacking areas. However, each network has to set up a StopIt server that knows who deploys StopIt in other domains. Moreover, the StopIt server can be the target of single point attacks. Router-based distributed packet filtering (DPF) (Park and Lee, 2001) also utilizes routing information to determine whether a packet follows an expected path. However, DPF focuses on source spoofing attacks, rather than DDoS attacks.

2.2. *Filter-based DDoS defense: operation and challenges*

The filter-based approach defends against DDoS attacks by blocking malicious traffic using filters on network routers, called filter routers (FR). Of the many filter-based approaches (Mahajan et al., 2002; Liu et al., 2008; Bellovin et al., 2001; Savage et al., 2000), Active Internet Traffic Filtering (AITF) (Argyraki and Cheriton, 2005, 2009) is regarded as one of the most recent and complete filter-based approaches. We will therefore explain the operation and the challenges when ATIF is used for DDoS mitigation.

2.2.1. *Operation of the filter-based approach*

The main purpose of the filter-based DDoS approach is to propagate filters along attack paths. It uses a record route (RR) scheme to identify the attack path and to generate a filter. Fig. 2 illustrates the way in which undesired flows are blocked using filter propagation. First, a victim sends a filter request to the victim's gateway (V_{gw}). Next, V_{gw} temporarily blocks the undesired flow and finds a border router located close to the attack source (A_{gw}). V_{gw} then sends the filter to A_{gw}, termed a counter-connection. Finally, A_{gw} demands that the attacker stop sending attack traffic. If the attacker continues sending, A_{gw} filters all traffic from the attacker.

2.2.2. *Challenges of the filter-based approach*

From the perspective of existing weaknesses of packet marking and filter-based approaches, we claim that a viable filter-based approach should address the several challenges such as how to practically identify attack paths, how to manage a multiplicity of filters, how to significantly speed up filter propagation, how to support incremental deployment. These problems motivate the work we present in this paper.

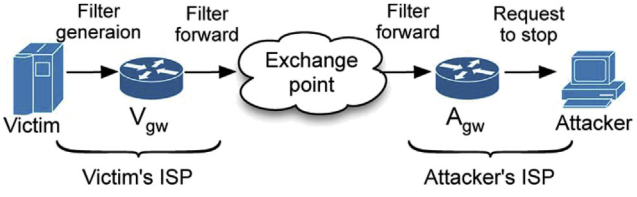


Fig. 2 – Filter propagation and attack blocking: the victim forwards filters using a record route scheme and requests an attacker to stop sending packets.

Incompatibility with legacy routers. Existing filter-based approaches such as AITF employs an RR scheme that is a variant of the traditional IP RR technique. As the RR information is placed at the beginning of the IP payload as an IP option, the size of the payload increases as the IP packet passes through several RR-enabled routers and can cause unexpected packet fragmentation. This can lead to high processing overhead, or even packet dropping in the worst case.

Delay due to hop-by-hop propagation. In practice, multiple FRs can be deployed between V_{gw} and A_{gw}, and filters are propagated on a hop-by-hop basis from the closest filter router to V_{gw} to the closest filter router to A_{gw}. This means that the speed of filter propagation may decrease as more filter routers are deployed in the network, and consequently decrease its defensive performance.

Vulnerability against filter flooding. Most filter-based approaches do not take filter flooding condition into consideration. In AITF, even though both V_{gw} and A_{gw} limit filter receiving and sending rates, attackers can still easily evade this technique by sending useless bogus filters using spoofed source addresses.

Small changes to networks. Since the filter-based approach needs to be installed to existing networks, network administrators such as Internet Service Provider (ISP) operators are not willing to install the approach if it requires many changes to existing networks. Therefore, a viable approach should require small changes to networks.

3. Problem definition

This section addresses the problems of existing filter-based approaches and outlines the goal we would like to achieve. In addition, we describe metrics to be used to measure the effectiveness of our proposed approach and assumptions made in our work.

3.1. *Requirements*

DDoS attacks are launched in concert of many compromised hosts (zombies). As a result, DDoS mitigation approaches face several important challenges. As shown in the AITF challenges (viz., Section 2.2.2), a filter-based approach needs to satisfy those challenges and accomplish the following requirements to be a viable solution. In particular, a viable solution is required to be easily deployable to existing environments. In that sense, we address the following requirements that make a filter-based approach viable.

R1: Practical path identification. A victim should be able to identify the path that the attack traffic traversed. To facilitate path identification, intermediate routers are required to give additional information to enable reconstruction of the flow's path for the victim. iTrace (Bellovin et al., 2001) and AITF utilize ICMP messages and a record route scheme, respectively,

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

to deliver the filter router's information. However, because iTrace sends ICMP traceback packets with low probability (1/20,000), it cannot effectively detect attack paths when attackers are widely distributed and send relatively small number of attack packets. If iTrace applied high probability to resolve the problem, it would incur bandwidth inefficiency and high overhead on routers due to the generation of multiple traceback packets. AITF's method (record route) is impractical, as discussed in Section 2.2.2. Therefore, the filter-based approach is needed to provide a practical method for path identification.

R2: Fast filter propagation. Multiple FRs can install filters on the attack path. In this case, the filter should be quickly installed at the filter router, which can effectively block malicious traffic. Therefore, a filter-based approach should leverage filter propagation to maximize filtering performance.

R3: Efficient filter management. A filter router has limited resources for filters, and its CPU and memory cannot process numerous filters efficiently. Therefore, a filter-based approach should incorporate an efficient filter management method for its limited resources.

R4: Incrementally deployablility. A new approach cannot be fully deployed to the whole network at once. Even if a new approach is partially deployed to a network such as a single intra-AS network, it should take effect so that early adopters can obtain benefits, and the benefit should be increased as the number of adopters grows.

3.2. Goal

We define ***the filter scheduling problem*** that a successful filter-based approach has to solve as follows:

- **Path identification**: How to practically identify attack paths,
- **Filter propagation**: How to propagate filters to the optimal locations (filter routers),
- **Filter management**: How to manage numerous filters such that the most effective ones are retained.

Our goal is to devise a filter-based approach that addresses the filter scheduling problem with supporting incremental deployment.

3.3. Metrics

We estimate filtering performance using five metrics: attack traffic ratio, effectiveness, first filter arrival time, and hop-by-hop block ratio. The first three metrics estimate false alarms in the approach, while the last two estimate the amount of attack and legitimate traffic at the victim's network.

Attack traffic ratio. This metric indicates how many attack packets reach the victim. It shows attack blocking performance by attack path identification (viz., R1 in Section 3.1) from the victim's perspective.

Effectiveness (ΦThis metric indicates how well the approach blocks attack traffic while allowing legitimate traffic to flow. It shows the performance of filter management (viz., R3 in Section 3.1). The best filter management method shows 1, whereas the worst one shows 0. We estimate Φas follows:

$$\Phi = 1 - \frac{(F_n + F_p)}{2}, \text{for } 0 \le F_n, F_p \le 1, \quad (1)$$

where F_n denotes false negative, which represents the probability of the FRs blocking legitimate packets, and F_p denotes false positive, which represents the probability of the FRs not blocking attack packets. Note that Φis measured from the filter router's perspective, whereas attack traffic ratio is measured from the victim's perspective.

First filter arrival time. This metric indicates the time at which the filter router receives the first filter. The lower the value of this metric, the better the system performs because a low value means that filter routers receive filters from either the victims or other FRs quickly. Therefore, this metric shows the speed of filter propagation (viz., R2 in Section 3.1).

Hop-by-hop block ratio. This metric indicates the location at which the attack packets were blocked. We check the number of attack packets blocked on a hop-by-hop basis. Blocking the attack packets closer to the attackers is the better. This metric is also related to filter propagation (viz., R2 in Section 3.1).

3.4. Assumptions

We set several assumptions from the perspective of attackers, routers, and victims.

3.4.1. Attacker's characteristics

Packet spoofing. Attackers may spoof IP source addresses in an attempt to make traceback difficult and to ruin the defense architecture. In addition, attackers may generate forged filters to neutralize the defense architecture. They can change any values in the filters, resulting in FRs blocking legitimate hosts.

Filter flooding. Attackers may generate large numbers of filters to degrade the performance of FRs. They can flood the table with useless filters, since each filter router has limitations on the size of its filter table.

No global attackers. Attackers cannot be global attackers. They cannot monitor every packet on every path. They only take partial information through many different paths and reassemble the information.

3.4.2. Router's characteristics

Limited resource. Routers have limited memory in which to store filters. Attackers may abuse this feature by filling the filter list with useless filters.

Non-compromised routers. Attackers cannot compromise any routers or set up bogus routers. Even though it is possible to establish a bogus router, we assume that a vigilant network administrator can easily recognize who is not cooperating and is pretending to be good.

3.4.3. Victim's characteristic

Undesired flow decision. Victims monitor traffic patterns and can identify attack traffic. This is not a tough assumption, because servers already adopt such traffic monitoring and analyzing systems. Besides, to paralyze those servers, attack flows have distinctively higher rates (pps or bps) than legitimate traffic.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

4. Adaptive Probabilistic Filter Scheduling

In this section, we propose a novel filter-based DDoS defense approach, called Adaptive Probabilistic Filter Scheduling (APFS) that efficiently blocks attack traffic by propagating filters to optimal FRs in the network. First, we explain the underlying design principle of APFS: how it propagates filters, how it effectively situates filters, and how it manages multiple filters. We then explain the operation of APFS in terms of its four phases: adaptive probabilistic packet marking, filter invocation, filter scheduling and propagation, and filter revocation.

4.1. *APFS design principle*

Defense against DDoS attacks becomes more effective as the malicious traffic are blocked at the ingress points to the network for attackers. In this sense, the filter-based approach becomes more effective when filters are installed on FRs closer to the attackers. Therefore, as mentioned in Section 3.1, a viable approach should incorporate a practical method to propagate filters to attacking source areas, an effective location to install the filters, and a management method to decide effective filters among the plenty of received filters.

How to propagate filters. Filter propagation can be performed after path identification. For path identification, intermediate routers either adopt a new protocol or use an existing protocol as shown in Fig. 1. Surely, using an existing protocol is a simpler and easier solution to implement and maintain compatibility, and packet marking is suitable for traffic path identification using an existing protocol such as IP (Yaar et al., 2003; Lee et al., 2007). In this technique, network routers insert unique information into unused IP header fields so that destination hosts are able to reconstruct the traffic path by gathering the inserted information.

Packet marking can be divided into two methods: probabilistic packet marking (PPM) and deterministic packet marking (DPM). In PPM, routers insert the unique information with a certain probability. For example, a router with 50% marking probability inserts the information every two packets. In DPM, routers insert the unique information into every packet. Since PPM results in lower router system overhead than DPM and it can forward the markings of multiple filter routers to destination hosts, we decided to use PPM to propagate filters. Another advantage of PPM is that it does not modify the IP payload, only the unused IP header fields. Therefore, it does not have any side effects on legacy routers such as the discarding or fragmenting of packets. ***APFS utilizes Probabilistic Packet Marking (PPM) to propagate filters.*** Thanks to PPM, a victim can reconstruct the gathered markings, identify which FRs forwarded the attack traffic, and send filtering requests to the corresponding FRs.

How to effectively place filters. PPM can employ either of two different marking methods to propagate filters: fixed PPM and adaptive PPM. Assume l FRs exist between an attacker and a victim. Let p_i be the marking probability of the ith FR. Then, in the fixed PPM,

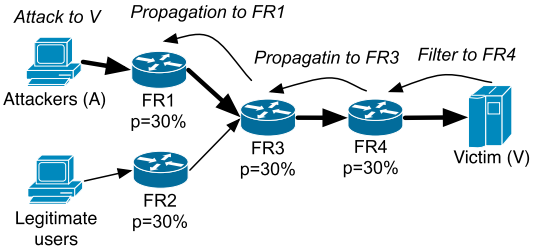
$$p_1 = p_2 \ldots = p_i \ldots = p_l (1 \le i \le l).$$

That is, FR inserts a marking with fixed probability, and thus, all the FRs apply the same marking probability. The fixed PPM cannot achieve effective propagation because the filters are passed on a hop-by-hop basis. Even though the destination host (victim) receives markings from different FRs, the victim still has greater chances of receiving the victim side FR's marking. This causes hop-by-hop filter propagation as shown in Fig. 3(a). Since the fixed PPM propagates filters to the closest upstream filter routers (FR4 in the figure), the filters need to pass through many FRs before reaching the attacker side FR (FR1 in the figure).

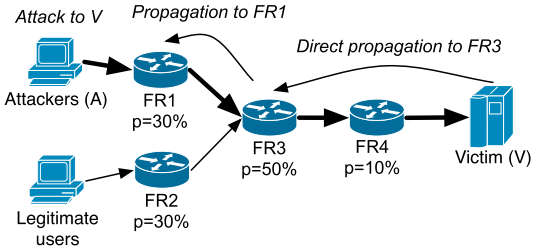
In the adaptive PPM, on the other hand, FR inserts a marking with adaptive probability, and thus, each FR applies different marking probabilities, as follows:

$$^{\exists} p_i \neq p_j (1 \le i, j \le l).$$

The victim can selectively receive markings from FRs that have high marking probabilities. In Fig. 3(b), we assume that FR1 has a relatively higher marking probability than the others. That is, FR1 is an effective filtering point. Accordingly, the victim has greater chances of receiving FR1's markings. Consequently, adaptive PPM allows the filter to be quickly forwarded to the most effective filter router. ***In APFS, we adopt the adaptive PPM, so each FR has its own marking probability.***



(a) Hop-by-hop filter propagation: All the FRs use the fixed marking probability (p=30%). This leads the hop-by-hop propagation; a filter slowly reaches attacker side FR (FR1).



(b) Direct filter propagation: The most effective FR has the highest marking probability (FR3: p=50%) while the least effective FR has the lowest marking probability (FR4: p=10%). This leads the direct propagation; a filter quickly reaches attacker side FR (FR1).

Fig. 3 – Filter propagation difference: (a) Hop-by-hop propagation forwards the filter to the neighbor FR, whereas (b) direct propagation quickly forwards the filter to the effective FR.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

That is, an FR that is capable of effectively blocking malicious traffic increases its marking probability so that the destination host gathers and reconstructs the FR's information with high probability. In contrast, an FR that is "not" capable of effectively blocking malicious traffic decreases its marking probability. We describe how each FR determines its own marking probability in Section 4.3.1.

How to manage filters. An FR receives many filters from many victims and other FRs. This situation can occurr as a result of DDoS attacks that are launched using botnets, such as the 7.7 DDoS attacks (Korea Times, 2009), or by filter flooding attacks in which attackers intentionally generate bogus filters to degrade filtering performance. Since a router has limited resources for installing filters, it is essential that an FR be able to distinguish between useful and useless filters. ***APFS adopts a filter scheduling policy that calculates the score of each filter and sets filter priority.*** Therefore, APFS selectively installs useful filters into FRs and maximizes filtering performance with limited resources. Furthermore, APFS is resistant to filter flooding attacks because it differentiates useless filters.

How to support incremental deployment. The performance of the filter-based approach depends on the deployment rate in the network by nature, meaning cooperation between filter routers. However, because there is no guarantee that other networks (ISPs) will adopt the approach and cooperate, the filter-based approach should solely provide sufficient benefits with low deployment rate, meaning no cooperation with other networks. Hence, we design that the necessary operations of APFS, such as computing marking probability and filter scheduling, are conducted by a single filter router itself. ***A filter router in APFS can solely decide its own parameters that maximize the defense effectiveness.*** Besides, if many filter routers are deployed throughout networks, filter routers will increase benefits by propagating filters.

4.2. *Overall APFS architecture*

In this section, we describe the overall architecture of APFS, then explain its four phases of operation. Fig. 4 illustrates the operation of APFS. The following numbered steps correspond to the numbers in Fig. 4.

1. Attacker (A) generates attack traffic directed at victim (V). In the mean time, a legitimate user sends packets to the victim.
2. The filter routers (FR1 and FR2) insert markings (m1 and m2) with their own adaptive probabilities, p_1 and p_2, where p_n denotes the marking probability of FRn. Since FR1 and FR2 are closer to the attacker than FR3, p_1 and p_2 are set higher than p_3. S1, S2, and S3 contain the information needed to reconstruct FR's IP address.
3. The filter router (FR3) marks its IP address with adaptive probability p_3 overwriting the probability that may have been marked there before. Steps 2 and 3 make up the first phase, and they are described in Section 4.3.1.
4. V collects the markings of the attack traffic: m1, m2, and m3. It then reconstructs and verifies the markings.
5. V sends filter requests to the filter routers. Note that V possibly reconstructs filters for either FR1 or FR2 first because p_1 and p_2 are higher than p_3. In this example, however, we assume that V firstly reconstructs the filter for FR3 in order to describe the filter propagation steps, which are steps 6 and 7. Steps 4 and 5 make up the second phase, and they are described in Section 4.3.2.
6. FR3 installs the filter and selects the most effective filters, termed the best-k filters, to block undesired (attack) traffic. FR3 then collects the markings corresponding to the best-k filters.
7. FR3 performs the same operation as V: i.e., collects, reconstructs and verifies the markings from the upstream filter routers (e.g., FR1). If FR3 succeeds in reconstructing the filter, FR3 sends the filter to FR1. Steps 6 and 7 make up the third phase, and they are described in Section 4.3.3.
8. After FR1 receives the filter from FR3, FR1 blocks the undesired traffic.

These eight steps describe the overall concept of APFS: how to use the marking value for the victim and how to propagate the filters among FRs. The remainder of this section describes the procedures that take place in the four phases in detail.

4.3. *Operation of APFS*

APFS consists of four phases: 1) adaptive probabilistic packet marking, 2) filter invocation, 3) filter scheduling and propagation, and 4) filter revocation. Fig. 5 briefly illustrates the flow of the four phases. In Phase 1, FRs probabilistically mark their own IP addresses into the packet headers. The FRs vary their marking probability in accordance with the filtering effectiveness calculated by three factors: HOP, RES, and DEG (explained in Section 4.3.1). Next, in Phase 2, victims collect and reconstruct the marking values to send filter requests. In Phase 3, the filter routers that receive the filter requests determine the best-k filters using a filter scheduling policy, and forward the filters to upstream routers. Finally, in Phase 4,

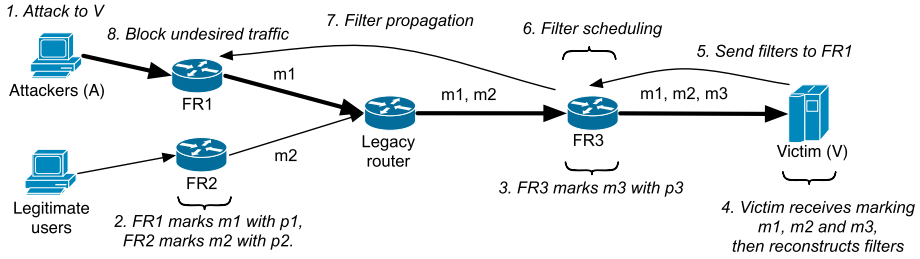


Fig. 4 – Overall APFS architecture.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

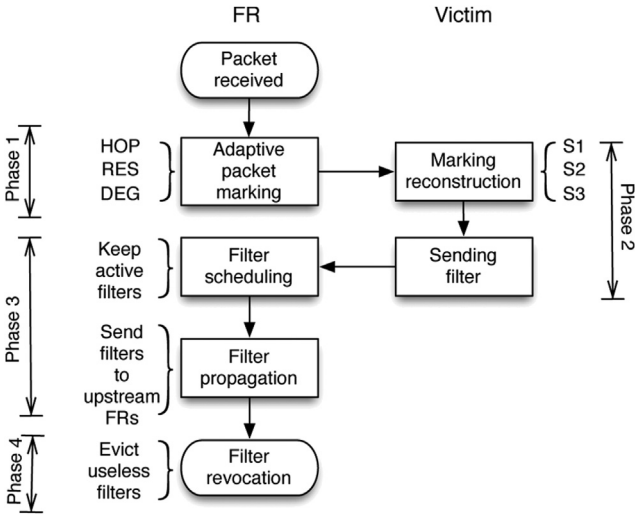


Fig. 5 – APFS consists of four phases: adaptive probabilistic packet marking, filter invocation, filter scheduling and propagation, and filter revocation.

when the attack stops, the filters' scores corresponding to the attacks decrease and the corresponding filters are eventually evicted from the filter routers. We now explain each phase in detail.

4.3.1. *Phase 1: adaptive probabilistic packet marking by FRs*
In order to send a filter request to an FR, a victim (V) has to know that FR's IP address; therefore, APFS utilizes PPM to identify which router is in charge of which flow. Each FR marks its unique information into the IP headers of outgoing packets with adaptive probability.

Determining adaptive probability for packet marking. As mentioned in Section 4.1, the principle of adaptive marking is that the FR that is expected to effectively block malicious traffic sets a high marking probability. Thus, each FR has to determine its own marking probability using the three factors (i.e., HOP, RES, and DEG).

• Hop count from attacker (HOP): HOP is used to indicate how far an FR is from the attacker in terms of hop count. This factor is necessary because blocking attack traffic at FRs near the attackers is more effective than blocking at the victim side FRs. An FR can measure HOP based on the TTL value of an IP packet because it is decreased by one each time passes through a router. However, the initial TTL value in the packet varies depending on the operating system (OS) of the packet sender. Table 1 shows the initial TTL value by OS (Davids, 2011). In addition, many researches (Jin et al., 2003; Wang et al., 2007; KrishnaKumar et al., 2010) has shown that the maximum hop count between a source and a destination in the Internet is approximately 30 hops. Thus, an FR can calculate how many hops (routers) a packet has passed through by referring to Table 1 and the maximum hop count. For example, if the TTL value of a packet is 45, we can infer that the initial TTL value was 64 and its hop count from the source is 19. In a normal scenario, the initial TTL value cannot be 255 because that would means that the hop count is 210 which exceeds the maximum hop count. We define the following formula to compute HOP:

$$HOP = \frac{h_{\max} - h}{h_{\max}}, \tag{2}$$

where h is calculated by subtracting the current TTL value from the initial TTL value, and h_{max} denotes the aforementioned maximum hop count. Hence, if the hop count is 19, HOP becomes $32 - 19/32 \cong 0.4$. The closer the FR is to the packet sender, the higher HOP is, and vice versa.

• Resource Availability (RES): RES is used to indicate the availability of an FR's filter list. The size of a FR's filter list is constrained, and so, if the filter list is full, it cannot install newly arriving filters. In this case, the FR stops inserting its marking in order to force V not to reconstruct the filter for the FR. We define a formula to compute RES as follows:

$$RES = \frac{q_{\max} - q}{q_{\max}}, \tag{3}$$

where q denotes the number of filters installed in FR, and $q_{\max}$ denotes the maximum number of filters that FR can install. For example, if we assume that $q_{\max}$ is 100 and q is 30, then RES would be $100 - 30/100 = 0.7$. The higher the number of filters installed is, the lower RES gets, and vice versa.

• Link degree (DEG): DEG is used to indicate how topologically important an FR is to the forwarding of traffic. In the Internet, some routers, so called hub routers or core routers, are placed in important places and play key roles in forwarding traffic. If a filter is installed into such hub routers, the malicious traffic can be effectively blocked. Several methods can be used to measure a router's "hubness." The simplest one is to considering the node degree by counting the total number of incoming and outgoing links:

$$DEG = \frac{k}{k_{avg}}, \tag{4}$$

where k denotes the number of link connections in the FR, and k_{avg} denotes the average number of link connections per FR in the network.

Another method is to calculate the "betweenness centrality". This factor indicates the node importance based on the shortest path. Given a source s and a destination d, the number of different shortest paths from s to d is $G(s,d)$. The number of shortest paths that contain a node m is $G(m;s,d)$. The proportion of shortest paths, from s to d, which contain

Table 1 – TTL by OS: The initial TTL values depends on OS. FR can infer the hop count of a packet using this information.

Operating system	Initial TTL
FreeBSD 5	64
MacOSX	64
Android	64
Windows 98, XP, and 7	128
Linux (2.4 kernel)	255
OpenBSD	255

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

node m is $P(m;s,d) = G(m;s,d)/G(s,d)$ (Zhou and Mondragón, 2011). Thus, we can calculate DEG using the betweenness centrality of m as follows:

$$DEG = \sum_{s} \sum_{s \neq d} P(m; s, d). \quad (5)$$

However, the drawback of calculating betweenness centrality is that we need to identify in advance how many shortest paths a node is in charge of. Therefore, if a network administrator can obtain the topological knowledge of shortest paths, then betweenness centrality is suitable as DEG. Otherwise, simply counting node degree is suitable due to its simple calculation.

Taking HOP, RES and DEG in consideration, each FR sets its adaptive marking probability (p_a) as follows:

$$p_a = p_d + (w_{\text{hop}} \cdot HOP) + (w_{\text{res}} \cdot RES) + (w_{\text{deg}} \cdot DEG), \quad (6)$$

where p_d denotes the default marking probability, and w_{hop}, w_{res}, and w_{deg} denote the weights of HOP, RES and DEG, respectively. p_d can be varied from 0.05% to 50% depending on the purpose of the mechanism. Savage et al. (2000) set 0.05% as the default marking probability because the purpose was full path reconstruction for IP traceback. However, for protecting against DDoS attacks, obtaining a filter router's information is more significant than full path reconstruction. Hence, our preliminary work (Seo et al., 2011) set it in the range of 30%–50% as the default marking probability. In our experiments, we show the difference of defense effectiveness depending on p_d (viz., Section 6.5).

Each w_x can also be varied in accordance with the network environment. For example, if the node degree of most FRs in the network is similar, then w_{res} can be low because DEG would not make much difference when calculating p_a.

To summarize, p_a is higher if FR is closer to attackers, has more resources for filters, and takes a topologically more important role.

Marking fragmentation and checksum. Another challenge to inserting marking is the questions of how to obtain the marking space in the IP header. According to Dean et al. (Dean et al., 2002), an IPv4 packet has 25 unused bits, which APFS utilizes for marking, and Fig. 6 illustrates how APFS marks using 25 unused bits. Technically, we need a 32 bit space to mark an FR's address. We therefore divide the FR's address into two parts: the first 16 bits (S1) and the last 16 bits (S2). V subsequently reconstructs the two parts to reconstruct the FR's address. However, ***incorrect reconstructions*** can occur. In Fig. 4, for example, V may incorrectly reconstruct the FR's IP address by combining S1 from FR1 with S2 from FR2. To prevent this possibility, an FR provides a checksum (CHK), which is a hash value computed based on the FR's address and Message Authentication Code (MAC).

Each FR generates two MACs with the FR's secret key computed over the destination IP address, and CHK can be changed depending on the MACs. Consequently, an FR serves three types of markings: S1 for the first 16 bits of the FR IP address, S2 for the last 16 bits of the FR IP address, and S3 for CHK. After collecting S1, S2, and S3, V extracts CHK from S3 and verifies the reconstruction if CHK = CHK′, where CHK′ is

$$H([FR(addr)]_{0-15} || [FR(addr)]_{16-31} || MAC1 || MAC2).$$

$<H(\cdot)$ denotes a cryptographic hash function that outputs 23 bits.

APFS needs the MACs for the following reasons. Assume that an attacker (A) knows the hash function for CHK, such that A can send the correct S3 to V. Additionally, assume that FR is not widely deployed and marks packets with a low probability. In such a scenario, A's packets can easily reach V without being marked by any FRs, and A can cause incorrect reconstructions using spoofed markings (explained in Section 7.2). To prevent this from happening, APFS adds the MACs computed over each FR's secret key and the destination IP address. As the MACs are dependent on the destination, A cannot find the MACs by sending packets to itself through the FR. However, it is only 12 bits (6bits + 6 bits). If the attacker sends various MACs using brute force, then V can suffer from incorrect reconstruction. Thus, APFS uses frequency analysis.

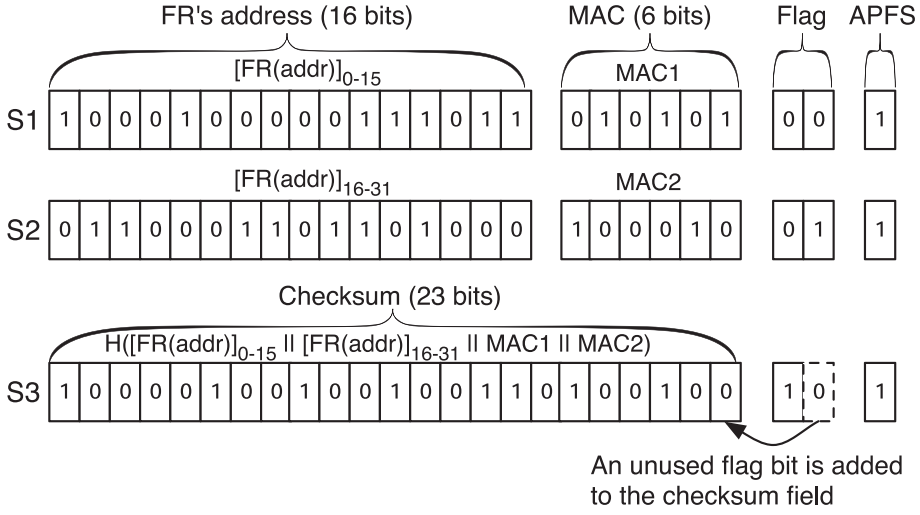


Fig. 6 – APFS utilizes 25 bits and creates three types of markings, S1, S2 and S3.

We discuss the details of this frequency analysis technique in Phase 2. The flag field (2 bits) indicates the type of the marking. As there are only three types (S1, S2, and S3), we can use the second bit of S3's flag as an additional checksum bit. Lastly, the APFS field (1 bit) checks if the packet is marked by FR.

4.3.2. *Phase 2: filter invocation*

After V collects S1, S2, and S3 related to an undesired flow, it carries out reconstruction, verification, and transmission of the filter request, Req{A, V, CHK}, to the corresponding FR. In Fig. 4, we assume that V collects FR3's markings first, and then sends a filter to FR3.

If there is a single FR between A and V, V can easily reconstruct the markings. Conversely, where there is multiple FRs between A and V, as in Fig. 4, we have to take the possibility of reconstruction failure into consideration.

Reconstruction failure. V may fail to reconstruct the markings due to an incorrect pairing of S1 and S2 (e.g., reconstructing using S1 from FR1 and S2 from FR2). Furthermore, as mentioned in Phase 1, there is the possibility that A floods the network with fake markings to cause incorrect reconstructions. We use ***frequency analysis***, which counts the frequency of S1, S2, and S3, to resolve the problem. The number of S1, S2, and S3 fragments should be large for a correct S1, S2, and S3, since APFS uses probabilistic packet marking. Thus, V sorts collected S1, S2, and S3 based on frequency, and attempts reconstruction by combining those S1, S2, and S3 fragments that have similar frequency. The number of S1, S2 and S3 from the same FR should be similar, no matter how many FRs exist between A and V, and irrespective of the FR's marking probability p. Even though the MACs are changed periodically, V can sense new MACs because new MACs eventually overwhelm old ones. If the reconstruction with the highest frequent S1, S2, and S3 fails, V resets the count of S1, S2, and S3 to 0. This is to avoid the selection of the same S1, S2, and S3 at the next reconstruction attempt. This frequency analysis technique is also used for filter propagation in Phase 3.

4.3.3. *Phase 3: filter scheduling and propagation by FRs*

After an FR receives filters from V, the FR attempts to propagate the filters to its upstream routers after verifying filter reconstruction using the checksum CHK, as shown in Fig. 7. In practice, FR has limited resources to store filters, while FR may receive many filters from many victims. Therefore, FR has to maintain the optimal filters that most effectively block attack traffic.

This issue bears similarities to the page replacement problem in OS caches, in which the best pages need to remain in the cache. The most popular policies are Least Frequently Used (LFU), Least Recently Used (LRU), and Adaptive Replacement Cache (ARC). According to Megiddo and Modha (2003), ARC performs better than the others. However, these policies are for a benign system, such as an OS. In APFS, therefore, we modified ARC to fit network environments that carry many malicious users, and designed a filter scheduling policy, which considers both frequency and recency.

Filter scheduling for filter management. The scheduling policy (ARC) scores each filter according to frequency and recency. Each FR maintains two lists of filters: a ghost list and a filter list. The ghost list stores suspicious filters. When FR receives a filter, it is initially stored in the ghost list. If the score of the ghost filter (the filter in the ghost list) exceeds a pre-defined threshold, termed the promotion threshold, then the ghost filter is promoted to the filter list, if the filter list is not full. If the filter list is full, the ghost filter's score has to exceed the lowest score in the filter list to be promoted. An FR blocks packets based on the filter list, and periodically computes scores for all filters in both the ghost and filter lists. A score (S) can be computed as follows: Let $S_n(I)$ be a moving average for filter I, where n is the current time. Let $P_n(I)$ be a score for I (initially $P_0(I) = 0$). t is a time window in which to compute $S_n(I)$. For example, if $t = 10$, FR computes $S_n(I)$ using the last ten $S_n(I)$. Thus, the calculation of $S_n(I)$ is given by:

$$S_n(I) = S_{n-1}(I) - \frac{P_{n-t}(I)}{n} + \frac{P_n(I)}{n} - \gamma, \tag{7}$$

where γ denotes the penalty score to decrease the filter score, so the filter should be evicted, if it becomes useless (e.g., no attacks). Note that the eviction is for filter revocation, which is Phase 4. $P_n(I)$ is computed using Eq. (8):

$$P_n(I) = F \cdot m + R \cdot (t_c - t_p), \tag{8}$$

where F denotes the weight of the frequency, and R denotes the weight of the recency. Moreover, m denotes how many times the filter is used, while t_c and t_p respectively denote the current packet arrival time and the previous packet arrival time related to the filter.

An FR does not install a received filter immediately. Initially, it stores the filter into the ghost list. If S_n(*new filter*) is higher than the lowest S_n(*old filter*) of a filter list, then the new filter is promoted to the filter list. The FR can keep the best-k filters via score comparison.

Direct filter propagation. Each FR performs the same procedures as V for the propagation to effective upstream FRs: collecting and reconstructing the markings. The effective upstream FR is a filter router that has a relatively higher p_a than other upstream FRs. Since an FR receives the markings of effective upstream FRs with high probability, filters are quickly distributed to effective FRs based on direct filter propagation. Moreover, each FR also uses frequency analysis to deal with the reconstruction failures.

To summarize, FR conducts two procedures in Phase 3: filter scheduling to keep the best-k filters that maximize DDoS defense, and direct filter propagation to forward the filter to effective upstream FRs.

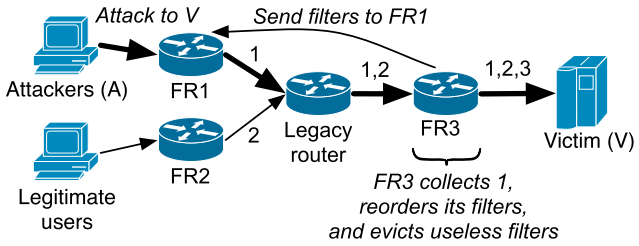


Fig. 7 – In Phase 3, FR3 collects markings from FR1 and propagates filters to FR1. In addition, FR3 reorders filters according to their scores. In Phase 4, FR3 evicts useless filters (low scored filters) from its filter table.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

4.3.4. *Phase 4: filter revocation*

The final phase is filter revocation. An FR can remove a filter by either of two means: explicit revocation or implicit revocation. In explicit revocation, V sends revocation messages to filter routers. However, APFS needs more complex procedures, such as a secure channel using key establishment to authenticate the revocation.

For our work, therefore, APFS only considers implicit revocation, which is conducted in accordance with a filter scheduling policy. When FR periodically performs filter scheduling, $S_n(I)$ is reduced by the penalty score (γ), as in Eq. (7). If the filter score is less than the promotion threshold, the filter is moved to the ghost list, and eventually removed.

5. Analysis of the effectiveness of APFS

As mentioned in Section 4.1, there can be two types of probabilistic filter scheduling depending on the dynamics of the marking probability. First, an FR can use a fixed probability for packet marking, termed Fixed Probabilistic Filter Scheduling (FPFS). Our preliminary work (Seo et al., 2011) for the filter-based DDoS defense was based on FPFS. Second, FR can use adaptive probability for packet marking, termed Adaptive Probabilistic Filter Scheduling (APFS).

The dynamics of marking probability affects filter reconstruction for the victim. Since FRs utilize frequency analysis (viz., Section 4.3.1) to reconstruct filters, the greater number of markings is more likely to be selected for filter reconstruction. Therefore, it is essential that the victim receives markings from effective FRs. If the victim receives many markings from effective FRs (e.g., attacker side FRs), the victim will reconstruct the markings and send filters to the effective FRs. Consequently, filtering effectiveness increases. On the other hand, if the victim receives many markings from ineffective FRs (e.g., victim side FRs), filters are distributed to the ineffective FRs first, and consequently the defense will not be effective.

In this section, we define δ, which measures the probability that an FR's marking will reach a victim. A FR with high δ can deliver its marking to the victim with high probability, and thus, the victim will send a filer to the FR with high probability. By calculating δ, we can find that which FR will likely receive filters from the victim, and the probability that effective FRs receives filters. Therefore, we compare the effectiveness of both FPFS and APFS in terms of δ.

5.1. *Effectiveness of FPFS*

Let p_d be the default fixed marking probability in FPFS, and let there exist l FRs between a certain FR and the victim. Then, the probability (δ) that the victim receives the markings of the certain FR is given by:

$$\delta = p_d \cdot (1 - p_d)^l. \tag{9}$$

Fig. 8(a) shows that most of the markings that the victim receives come from the victim side FRs. This is because the markings of the victim side FRs overwrite the markings of the attacker side FRs. In FPFS, this phenomenon, termed marking overwriting, is inevitable since all the FRs employ fixed marking probability, p_d. Accordingly, the marking overwriting causes hop-by-hop filter propagation and degrades the effectiveness of the defense.

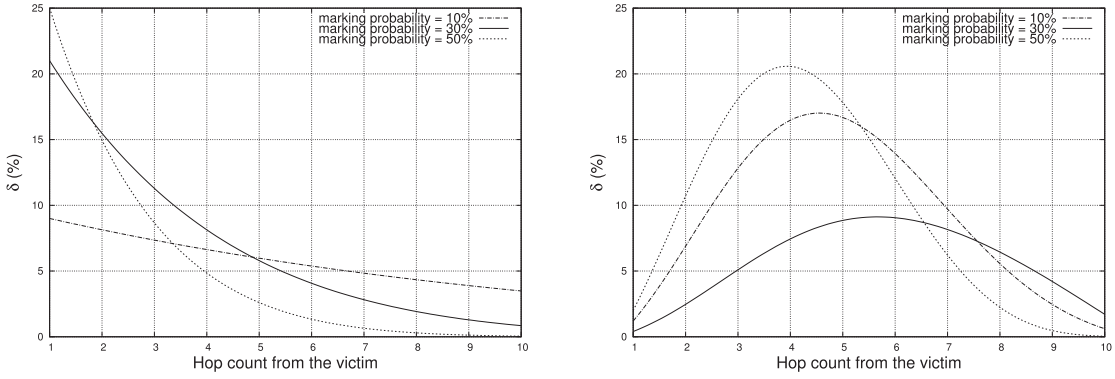
5.2. *Effectiveness of APFS*

We also define δ for APFS by modifying Eq. (9). Let p_a be the adaptive marking probability of a certain FR, i, and let there exist l FRs between i and the victim. Then, the probability (δ) that the victim receives the markings of the certain FR, i, is given by:

$$\delta = p_a(\mathrm{i}) \cdot \prod_{j=1}^{l} 1 - p_a(j), \tag{10}$$

where $p_a(j)$ is an adaptive marking probability (p_a) of the jth FR. The p_a of each FR is calculated by Eq. (6).

Fig. 8(b) shows that most of markings that the victim receives come from the core side FRs. Since p_a considers both HOP (hop count from a sender) and DEG (FR's link degree), core side FRs (hub FRs) likely have higher p_a than others. Note that Eq. (6) does not consider RES (FR's resource) because we assume that all the FRs have same amount of initial resources. Practically, in complex networks such as the Internet,



(a) In FPFS, the markings of the victim side FRs likely reach the victim.

(b) In APFS, the markings of the attacker side FRs likely reach the victim.

Fig. 8 – APFS delivers more markings of the attacker side FRs than FPFS. In APFS, therefore, filters can be quickly distributed to the attacker side, and the defense effectiveness improves.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

installing filters in hub routers is much more effective than installing them in edge routers, because the number of edge routers is significantly greater than that of hub routers, such that propagating filters to all the edge routers can result in deployment issues. This is why we consider DEG in APFS.

As a result, APFS can quickly propagate filters to attacker side FRs which improves the effectiveness of its defense. In Section 6.4, we show the experimental comparisons between FPFS and APFS in terms of attack traffic ratio, first filter arrival time, and hop-by-hop block ratio.

6. Experimental results

We evaluate APFS using three different topology data: Routeview (AS level topology) (Meyer, 2006), CAIDA skitter map (router level topology) (CAIDA Skitter, 2007), and Rocketfuel (intra-AS topology) (Spring et al., 2002). We utilize two network simulators: NS-2 (NS-2, 2008) with Routeview and Portcullis simulator (Parno et al., 2007) with CAIDA skitter map and Rocketfuel. NS-2 with Routeview is suitable to experiment filter flooding attacks with various network option changes making bogus filters pass through different ASes.

However, because NS-2 is not scalable, it cannot be used to conduct large scale DDoS simulations. Therefore, for large scale simulations, we use the Portcullis simulator, which can simulate thousands of network nodes.

We conduct six simulations: filter scheduling, filter flooding, changes in marking probability, changes in deployment rate, first filter arrival time, and attack block by hop. The first two simulations do not require a large scale network, but simply require adjustments to the attacker's strategy. Therefore, we use NS-2 for the first two simulations, which provides various options for network settings. The last four simulations require large scale networks to observe the effectiveness of APFS, as we need to vary APFS parameters. As a result, we use the Portcullis simulator (Parno et al., 2007) for the last four simulations.

6.1. Experimental setup for filter scheduling and filter flooding

We use a Routeview dataset from April 2009 to construct an Internet-like AS level topology in NS-2. We extract a subgraph by breadth first search, which consists of 1000 nodes: 200 attackers, 200 legitimate hosts and 600 ASes acting as APFS-enabled network. Fig. 9(a) shows the topology of the Routeview dataset. The legitimate user group sends packets with a lower rate than the attackers do. We divide the attackers into two groups, Group A (100 attackers) and Group B (100 attackers). Three scenarios (Scn #1 to SCN #3) consisting fo different attack strategies are used. Group A and B change the attack strategies using different rates (50 and 100 kbps), whereas the legitimate user group sends packets with the lowest rate (25 kbps).

- Scn #1: Group A sends packets for 0.5 s at 100 kbps and stops sending. Next, Group B sends packets for 0.5 s at 50 kbps. The procedure are then repeated.
- Scn #2: Group A sends packets at 100 kbps. At the same time, Group B sends packets at 50 kbps.
- Scn #3: Group A and B send packets at the same rate, 100 kbps.

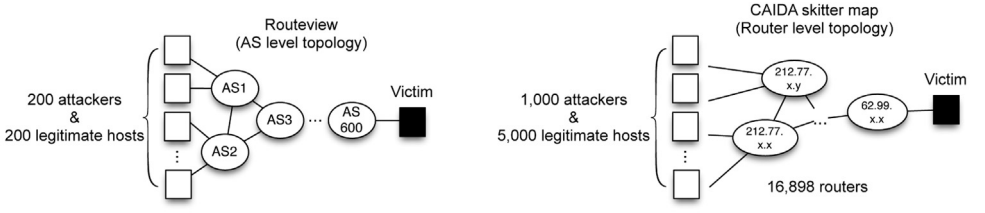
The experimental assumptions are as follows.

- The delay between links is 20 ms, and the queue size of a router is sufficiently large, so that a router does not discard any packet due to queue size.
- Default marking probability is 10% ($p_d = 0.1$).
- The sizes for the filter and ghost lists are both 100.
- γ in Eq. (7) is 1; thus, the score of each filter decreases by 1 per 20 ms.
- The promotion threshold for promoting from the ghost list to the filter list is 10.
- The timeout threshold for evicting a filter from the ghost list is 100 (=2 s).

6.2. Determining the filter scheduling policy

First, we determine which filter scheduling policy performs the best. As mentioned in Section 4.3.3, there are three candidates: Least Frequently Used (LFU), Least Recently Used (LRU), and Adaptive Replacement Cache (ARC) modified to fit to APFS. The following concepts apply to each of the policies in turn.

- LFU: Filter frequency determines the score; how many times the filter is used.
- LRU: Filter recency determines the score; how recently the filter is used.
- ARC: Both frequency and recency determine the score; how many times and how recently the filter is used.



(a) Routeview topology consists of 1,000 nodes: 200 attackers, 200 legitimate hosts and 600 ASes acting as APFS-enabled networks.

(b) CAIDA skitter map consists of 1,000 attackers, 5,000 legitimate hosts and 16,898 routers.

Fig. 9 – AFPS is experimented with two different topologies: (a) Routeview (AS level topology) and (b) CAIDA skitter map (router level topology).

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

The three policies differ in terms of the method used to compute $P_n(I)$ in Eq. (8). We set R to 0 for LFU, and set F to 0 for LRU. For ARC, we can vary both weights, F and R. We fix F as 1, and vary R to find the balance between F and R in Eq. (8). ARC shows the highest attack blocking ratio for $F{:}R = 1{:}50$ based on the experiments with three scenarios. This means that 1 score of recency is equivalent to 0.02 score of frequency.

In Fig. 10, F_n (False negative) denotes the probability that V receives attack packets from attackers (effectiveness), and the number of replacements describes how frequently the policy replaces filters (overhead). It can be seen that ARC > LRU > LFU in terms of the effectiveness and LRU > ARC > LFU in terms of the overhead. Therefore, we conclude that ARC achieves a higher performance with lower overhead than the other policies.

6.3. *Effectiveness under filter flooding attacks*

Under the filter flooding attack, we estimate the effectiveness of APFS compared to the existing scheme that does not use the filter scheduling policy. The three attack scenarios give results that are similar to those of the filter scheduling experiments. In Fig. 11, the attackers begin the DDoS attack in 1 s, and generate many useless filters in 2 s to cause the filter flooding attack. Φ is the filtering effectiveness from Eq. (1).

The result shows that APFS is unaffected by filter flooding, since it employs a filter scheduling policy (ARC). After the filter router selects the best-k filters, Φ increases. In AITF, it adopts the filters faster and seems to block successfully initially. However, after filter flooding occurs, AITF unconditionally drops the traffic irrespective of whether it is malicious or not. Consequently, APFS's Φ (0.881) outperforms AITF's one (0.485) because APFS attempts to maintain the best-k filters according to attack situations.

Consequently, attackers can degrade the defense effectiveness using the filter flooding attack, but APFS overcomes this weakness thanks to the filter scheduling policy.

6.4. *Experimental setup for the large scale simulations*

The Portcullis simulator has a lower number of optional functions than the NS-2 simulator, but conversely provides large scale network simulations. We utilize a topology map from the CAIDA skitter map, which stores a router-level topology (routing paths with real IP addresses of intermediate routers) 9(b). The topology consists of 1000 legitimate hosts, 5000 attackers and 16,898 routers. One round denotes the time that a packet takes to move one hop in the Portcullis simulator.

The experimental assumptions are as follows:

- The link delay is 20 ms; thus, 1 round is 20 ms.
- Attackers send 10 packets per round, while legitimate hosts send 1 packet per round.
- Attackers insert random spoofed markings with 10% marking probability. If attackers generate too many marked packets, the victim can recognize packet mark spoofing.
- The router queue size is 100 packets; thus, the queue handles 100 packets per round.
- The filter scheduling parameters, such as filter size, F, R, γ, promotion threshold and timeout threshold are the same as the NS-2 simulation parameters.

Furthermore, to show the effectiveness of adaptive marking probability, we conduct all the experiments using two different marking probability methods, APFS and FPFS. APFS uses adaptive marking probability while FPFS uses fixed marking probability.

6.5. *Changes in marking probability*

The default marking probability (p_d) is an important factor, because it determines how many marked packets the victim receives. In Section 5, our analysis shows that APFS and FPFS have different effectiveness in accordance with their marking probability. In this experiment, we find the optimal marking probability based on our simulation topology.

We vary the marking probability from 10% to 50%, and measure the attack traffic volume received at the victim. Fig. 12 compares the results obtained for APFS and FPFS. With FPFS (Fig. 12(b)), which uses fixed marking probability, the victim receives the lowest attack traffic when p_d = 30%. p_d = 10% delivers too small a number of markings to the victim for filter reconstruction, while p_d = 50% delivers too many

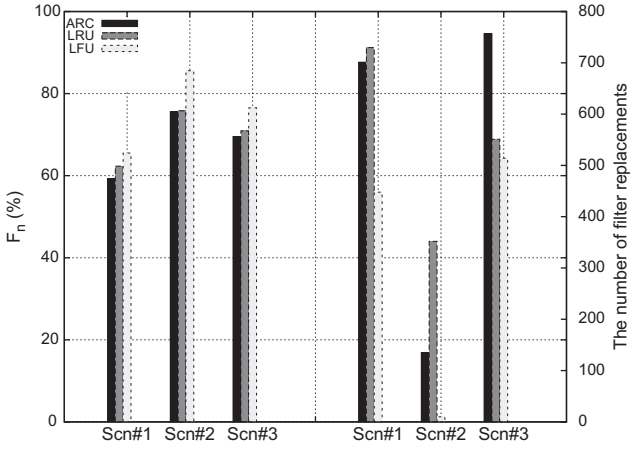


Fig. 10 – ARC shows the lowest F_n (left), and replaces the filters relatively fewer times than LRU (right) does.

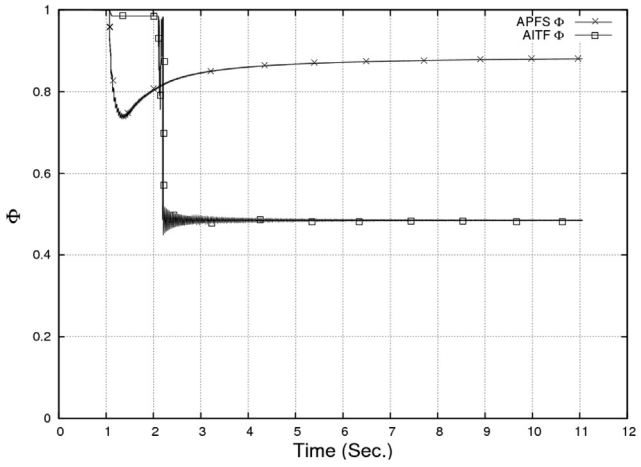


Fig. 11 – Filter flooding attack: APFS shows ≈44% higher effectiveness than AITF.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

markings to the victim, so that reconstruction failures frequently occur. However, with APFS (Fig. 12(a)), the FRs dynamically change their probability depending on filtering effectiveness, and thus, higher probability gives better performance. As a result, APFS with $p_d = 50\%$ gives the best performance. Even when FRs are deployed in a small portion of the network, i.e., 10%, APFS can block 80% of attack traffic.

The lines in the figure show the threshold-style because of link congestion. Even though when the FRs install many filters, link congestion can still occur. If sufficient filters are installed in a certain area to mitigate the link congestion, then the legitimate traffic from the area can reach the victim.

6.6. Changes in deployment rate

To evaluate the benefits by incremental deployment (R4), we conduct the experiments involving changes in deployment rate (d). From Fig. 13, it can be seen that APFS blocks attack traffic more effectively than FPFS. FPFS performs better when the deployment rate is 30% than when it is 50%. This occurs because with FPFS too many FRs, such as $d = 50\%$, can cause more confusion for the victim. The victim receives many different markings from various FRs, and thus cannot create filters efficiently. More importantly, FRs using FPFS keep sending their markings even when they are already full of filters. This forces the victim to generate useless filters that cannot be installed. For this reason, in Fig. 13(b), the attack traffic ratio in FPFS does not decrease after tens of seconds.

In contrast, APFS can effectively utilize the amount of FRs thanks to adaptive marking probability. First, even for a high deployment rate such as 50%, the FRs still change their marking probabilities so that the victim does not suffer from marking flooding. Furthermore, FRs that are full of filters do not insert their markings any more, and thus, the victim can effectively reconstruct filters for FRs that have resources for filter installation. In Fig. 13(a), especially when $d = 30\%$ and $d = 50\%$, the attack traffic ratio in APFS continuously decreases because APFS effectively distributes filters among the available FRs. Consequently, given 50% of the deployment rate, APFS can block 80% of attack traffic, and the attack blocking ratio increases as the number of FRs in the network increases.

In addition, we also investigate APFS for varying numbers of attackers. We vary the number of attackers from 100 to 5,000, while keeping the number of legitimate users fixed at 1000. Fig. 14 shows that the victim only receives 5–8% of the attack traffic when there are 100–500 attackers. Even when the number of attackers increases tenfold (1000–5000), the attack traffic at the victim increases only fivefold-sixfold (i.e., 30–35%). This indicates that the performance of APFS remains stable even when the number of attackers increases.

6.7. First filter arrival time

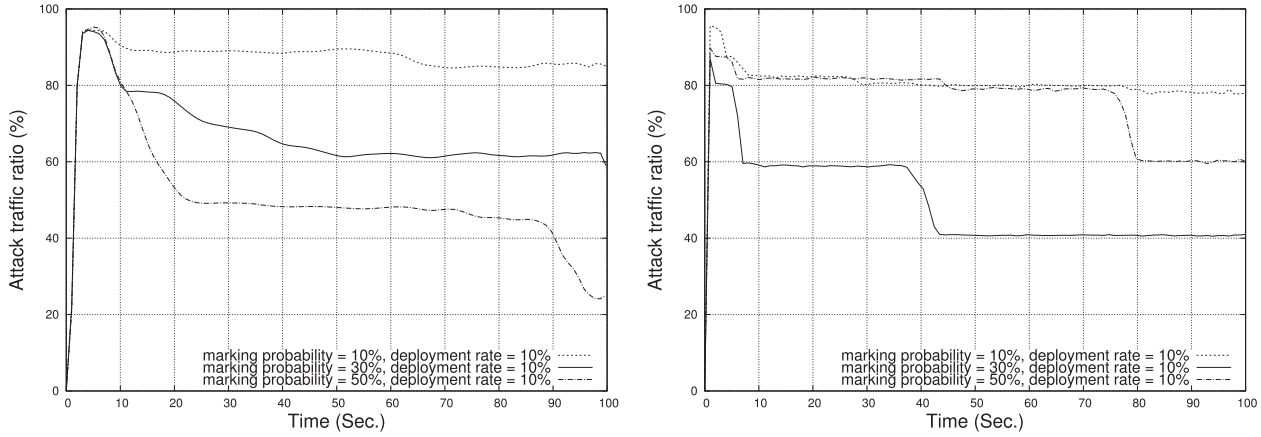
We monitor the time at which FR receives the first filter, termed first filter arrival time, to ascertain how adaptive marking probability affects the filter propagation speed throughout the network.

In Fig. 15, the x-axis signifies the hop distance from the victim (the higher the value, the closer to the attacker). Considering hop distance, we logically define each FR as belonging to one of the three sides in the network: victim side (1–6 hop distance), core side (7–12 hop distance), and attacker side (13–18 hop distance).

The results show that APFS receives the filter regardless of hop distance. In particular, the filter arrival time at the core side FR is as short as the time at the victim side FR. The time gap between FPFS and APFS tends to widen as hop distance increases.

Fig. 16 shows how many attack packets are blocked relative to hop distance (hop-by-hop block ratio). At the core and attacker sides, APFS blocks much more attack traffic than FPFS. We summarize the result in Table 2.

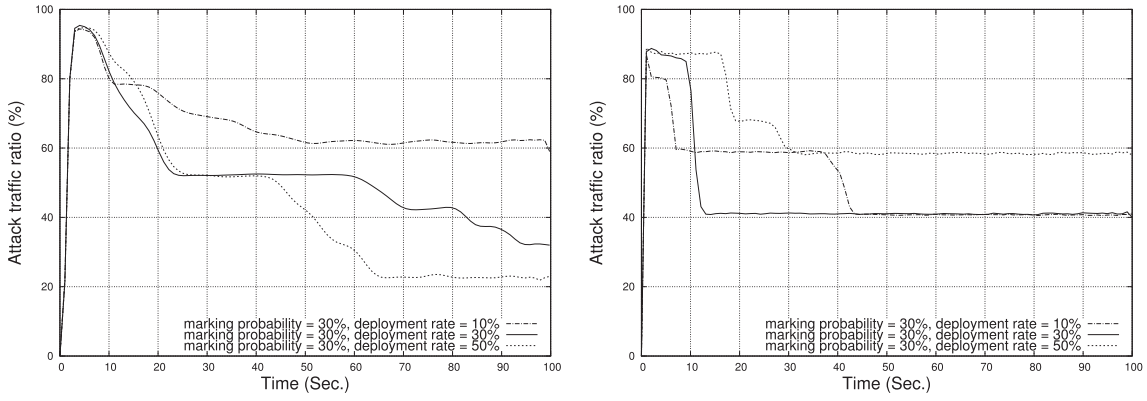
APFS blocks 91% of attack traffic at the core and attacker sides, while FPFS blocks 73% at the same sides. Since APFS propagates filters to effective FRs according to HOP, RES, and



(a) APFS attack traffic ratio. (b) FPFS attack traffic ratio.

Fig. 12 – Attack traffic ratio by changes of marking probability: (a) APFS shows the best block ratio with 50% marking probability. (b) FPFS shows the best block ratio with 30% marking probability.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002



(a) APFS attack traffic ratio. (b) FPFS attack traffic ratio.

Fig. 13 – Attack traffic ratio by changes of deployment rate: (a) APFS shows the best block ratio with 50% deployment rate. (b) FPFS shows the best block ratio with 30% deployment rate.

DEG, the filters spread to further and faster in the network, and therefore, APFS blocks more attack traffic near to attackers.

6.8. *Filter reconstruction effectiveness*

We also measure the reconstruction effectiveness: how many reconstruction failures and successes occur, how frequently incorrect reconstruction happens, and how many filters are sent by the victim and the FRs. Tables 3 and 4 show the statistics based on the experiments with regard to changes in p_d and d. In the tables, APFS shows a lower failure ratio and higher success ratio for filter reconstruction on average than FPFS.

Incorrect reconstructions occur because attackers generate spoofed markings. However, the probability is very low (FPFS: 0.008% and APFS: 0.05%), and the filters created by incorrect reconstructions do not match any FRs' addresses in the network. Consequently, the incorrect reconstruction does not cause negative effects.

6.9. *Experimental results for the intra-AS level simulations*

Since the current Internet is provided by many ISPs and there is no guarantee that the ISPs cooperate with each other, APFS should finely perform within a single ISP network. In this experiment, we simulate with the Rocketfuel AT&T data (Spring et al., 2002) shown in Fig. 17(a), which provides the intra-AS level topology. The topology has 115 cities (nodes) across U.S.A. We select Abingdon, VA as a potential victim because the city has risk of link congestion in that it has a single link to a neighbor. Then, we establish 114 distinct paths from other nodes to the victim, assuming that 50% of the paths generate undesired flows. All the other experimental environment are equal to the one in Section 6.4.

In experiments with varying the deployment rate (10%–50%), as shown in Fig. 17(b), the result shows that APFS provides sufficient attack blocking ratio (60%) with low deployment rate (10%), and the performance increases as the deployment rate goes high. Finally, it shows complete

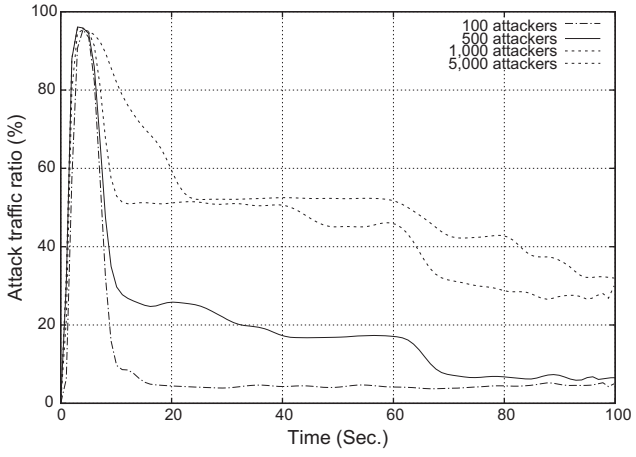


Fig. 14 – APFS shows stable attack blocking performance even though the number of attackers dramatically increases.

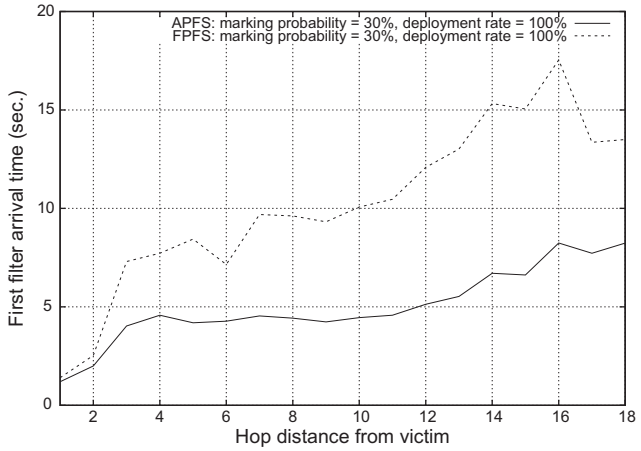


Fig. 15 – FRs in APFS receives filters faster than the ones in FPFS. At the attacker side FRs, the time gap increases.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

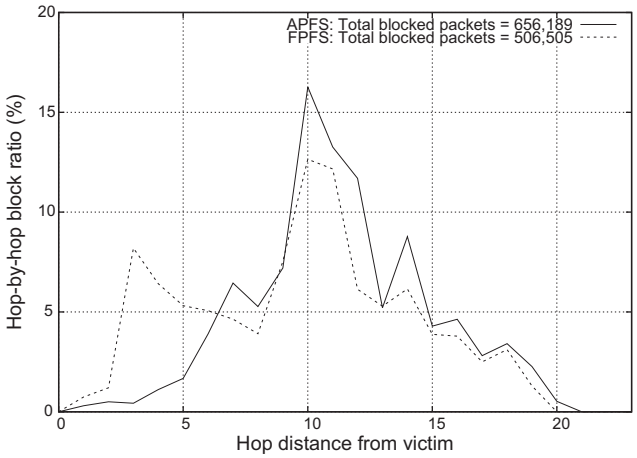


Fig. 16 – APFS blocks more attack traffic near attackers than FPFS. Since core side FRs send markings with high probability, attack traffic is more likely blocked at core side FRs.

protection with 50% deployment rate. It means APFS provides benefits for early adopters and supports incremental deployment even if it is deployed to intra-AS networks.

7. Security analysis

In this section, we look into strategic attacks that may try to exploit the architecture of APFS and discuss how APFS is resistant to such attacks.

7.1. Filter flooding attack

There are two types of filter flooding: filter flash crowds and filter flooding attacks. Filter flash crowds occur when legitimate users send many legitimate requests to a specific filter router, like flash crowds. Conversely, in filter flooding attacks, an attacker (A) intentionally sends numerous filters to fill specific FRs with useless filters. This can neutralize filter-based DDoS defense schemes if they do not utilize any filter scheduling policy.

Many filter-based approaches have not yet addressed this issue, however, AITF (viz., Section 2) uses rate limiting for filter generation to prevent it. In AITF, each host and gateway have rate limitations for filter sending and receiving. However, A can easily break down this technique by sending useless filters using source address spoofing. In APFS, FR maintains the best-k filters and evicts useless filters implicitly. That is, the ghost list can become filled with useless filters for a while, but the ghost list will not promote the useless filters to the filter list unless A generates a lot of traffic to promote the useless filters.

Table 2 – The number of attack packets blocked at victim, core and attacker sides.

	APFS	FPFS
	# of attack packets blocked (%)	# of attack packets blocked (%)
Victim side	51,979 (7.92%)	136,435 (26.93%)
Core side	394,700 (60.15%)	238,250 (47.03%)
Attacker side	206,070 (31.40%)	131,820 (26.02%)
Total	656,189	506,505

7.2. Packet mark spoofing

An attacker (A) can intentionally insert forged marking values that lead to hash collision when the victim (V) reconstructs the marking. APFS uses frequency analysis to resolve this problem, as mentioned in Section 4.3.2. In this technique, V and the FRs reconstruct with high frequency S1, S2, and S3. V receives various markings from A, and the number of genuine markings generated by the FRs becomes higher than the number of A's markings because A has to conduct brute force attacks to cause hash collision, especially to break second pre-image collision resistance.[1]

7.3. Source address spoofing

APFS can block attack traffic even when there is source address spoofing. For example, in Fig. 4, A launches a source address spoofed DDoS attack imposing the legitimate user (L)'s address on V. Simultaneously, L also sends packets to V. Then, V can reconstruct three filters: Req{L, V, CHK(FR1)},[2] Req{L, V, CHK(FR2), and Req{L, V, CHK(FR3)}. Initially, FR3 can block the traffic coming from both A and L; however, soon, FR3 propagates the filter to FR1 and FR2. FR2 does not block genuine L's traffic because L's traffic volume is insufficiently large to promote the filter to the filter list; whereas, FR1 blocks spoofed L's traffic because A generates attack traffic at a high rate.

7.4. TTL spoofing

Since APFS varies marking probability depending on the TTL value, A can intentionally modify the TTL value in the attack packet. For example, A may examine the hop distance to V in advance, and gain knowledge of the minimum TTL value to reach V. A modifies its TTL value to the minimum value so that intermediate FRs decreases their marking probabilities according to Eq. (7). However, the fact that the TTL value is decreased by 1 after passing through a router is unchangeable. Thus, the marking probability of FR located near A is still higher than the one near V.

7.5. APFS enhancement

Network monitoring. In order to effectively detect DDoS attacks, V needs to regularly monitor surrounding networks to identify who may be an authentic FR in order to quickly

[1] It is also known as weak collision resistance: Given x, an attacker cannot find $x' \neq x$ such that $H(x) = H(x')$.

[2] Note that Req{L, V, CHK(FR1)} denotes the filter request for FR1 to block the flow from L to V.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

Table 3 – FPFS shows 20.03% reconstruction success rate in the experiment (p_d is the default marking probability and d is the deployment rate).

Simulation	Reconstruction failure	Reconstruction success	Incorrect reconstruction	Filters sent by the victim	Filters sent by filter routers
$p_d = 10\%$, $d = 10\%$	876,778	97,723	16	351	97,356
$p_d = 30\%$, $d = 10\%$	401,043	15,557	66	555	14,936
$p_d = 50\%$, $d = 10\%$	381,217	123,774	37	647	123,090
$p_d = 30\%$, $d = 30\%$	575,523	231,280	134	911	230,253
$p_d = 30\%$, $d = 50\%$	120,818	394,130	86	704	393,340
Average (%)	618,387 (79.94%)	155,084 (20.05%)	59 (0.008%)	580 (0.37%)	154,446 (99.55%)

reconstruct filters. A can forge many packets using brute force attacks to incur incorrect reconstructions with high probability. In such a scenario, the ratio of APFS marked packets would increase significantly, and a network administrator or an ISP operator would easily identify the situation as being abnormal. In addition, we can easily detect the path of the forged packet which shows temporal high marking probability.

IP header authentication. Even though APFS utilizes CHK and frequency analysis to detect spoofing attacks in IP headers, the victim can additionally apply IP header authentication solution. For example, a traffic normalization technique using IP header fields is applicable to the victim side network as an anomaly detection method. The traffic normalizer (Handley et al., 2001) filters out abnormal traffic according to predefined rules. For instance, it filters out all the packets that do not meet to IP ID sequence rules. Haris et al., 2011 proposed a TCP flooding-specified anomaly detection technique. It checks the header size of a TCP packet and identifies the validity of TCP flag combinations according to the TCP handshaking process. Since these techniques are suitable for detecting packet flooding such as probing and port scanning, it helps to detect packet mark spoofing by brute force in APFS. Nevertheless, because anomaly detection for every packets can be a burden for filter routers that have limited resources, it can be installed to the victim side network as a part of IDS/IPS.

7.6. *Supporting IPv6 and Mobile IP*

When APFS is adopted to different IPs, the challenge is to deliver filter router's IP address to a victim; how to mark a filter router's IP address into IP packets. Here, we consider how APFS can be adopted to IPv6 and Mobile IP.

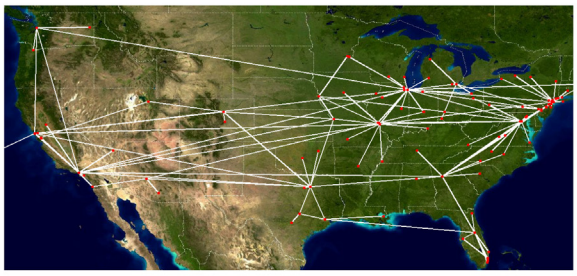
APFS in IPv6. Several researches for probabilistic packet marking in IPv6 (Dang and Albright, 2005; Dang et al., 2007) have been published, and they utilized unused fields in IPv6 which is the "flow label" like the identification field in IPv4. Similarly, APFS can run on IPv6 since packet marking in IPv6 is also available. IPv6 specification describes extension headers that include hop-by-hop options, destination options, authentication information, etc. However, similar to IPv4 option fields, IPv6 extension headers are not mandatory fields and many legacy routers do not utilize them. Therefore, we have to insert packet markings in IPv6 basic headers instead of extension headers.

APFS can deliver a filter router's IP address using the marking in the flow label. Since the IP address in IPv6 is 128 bits and the flow label has only 20 bits, markings for a filter router are divided into seven fragments. Compared to three fragments in IPv4, seven fragments in IPv6 mean that the victim will take more time to receive markings for constructing a filter. To see the time overhead due to the fragments in IPv6, we used the linear packet marking model (Saurabh and Sairam, 2012), Eq. (11), that calculates expected

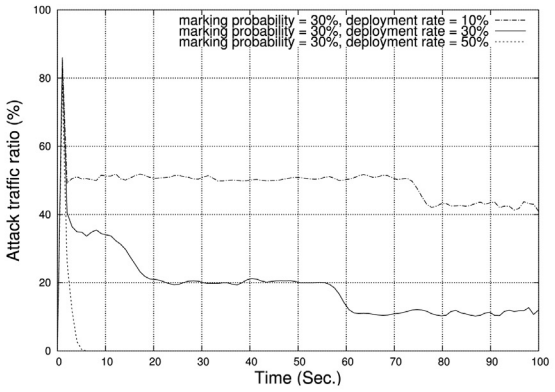
Table 4 – APFS shows lower failure ratio and higher success ratio than FPFS (p_d is the default marking probability and d is the deployment rate).

Simulation	Reconstruction failure	Reconstruction success	Incorrect reconstruction	Filters sent by the victim	Filters sent by filter routers
$p_d = 10\%$, $d = 10\%$	273,368	81,370	620	3680	77,690
$p_d = 30\%$, $d = 10\%$	205,317	90,091	748	8330	81,761
$p_d = 50\%$, $d = 10\%$	262,786	90,355	238	2541	87,813
$p_d = 30\%$, $d = 30\%$	757,384	245,947	102	1768	244,179
$p_d = 30\%$, $d = 50\%$	1,443,113	392,385	391	2210	390,175
Average (%)	588,393 (76.57%)	180,030 (23.42%)	419 (0.05%)	3706 (0.48%)	176,324 (97.71%)

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002



(a) Rocketfuel topology: An intra-AS network of AT&T in U.S.A



(b) APFS attack blocking ratio on Rocketfuel with varying deployment rate

Fig. 17 – In experiments with the intra-AS network, APFS shows sufficient attack blocking ratio (60%) with low deployment rate (10%), and the performance increases as the deployment rate goes high (complete protection with 50% deployment rate).

number of packets ($E(X)$) given marking probability (p), number of fragments (k), and number of filter routers between an attacker and a victim (d).

$$E(X) \leq \frac{k \cdot ln(kd)}{p(1-p)^{d-1}} \tag{11}$$

As the result, APFS in IPv4 needs 500 packets to construct a filter while APFS in IPv6 needs 1500 packets. That is, assuming our experimental environments (DDoS attacks using 5000 attackers with 10 pps), the victim in IPv4 receives 500 packets every 0.01 s while one in IPv6 receives 1500 packets every 0.03 s. This time overhead (0.02 s delay for reconstructing a filter) is negligible.

APFS in Mobile IP. In Mobile IP, only when packets are destined to mobile node such as smartphones, Mobile IP is used by encapsulating IPv4 or IPv6 packets and delivering to the mobile node using Mobile IP tunneling through Home Agent (HA). In APFS, we consider a non-mobile server as a victim because DDoS attacks attempting to cause link congestion target non-mobile servers. Therefore, IPv4 or IPv6 packets are used even though attackers are using mobile nodes. However, mobile attackers can frequently move from one's home network to another foreign network, which causes many changes of attacker's source IP address. This situation is similar to the source address spoofing attack but it is based on legitimately spoofed addresses binded to Foreign Agent (FA). In this case, APFS can defend against this strategic DDoS attack since the intermediate filter routers that are placed between mobile attackers and the victim will be activated. Nevertheless, APFS can cooperate with notification techniques to HA in Mobile IP traceback researches (Lee et al., 2003; Jang et al., 2011) so that HA easily identifies the mobile attacker's home address.

8. Conclusion

In this paper, we presented our Adaptive Probabilistic Filter Scheduling (APFS) architecture, which utilizes Probabilistic Packet Marking (PPM) and a filter scheduling policy to defeat DDoS attacks. APFS propagates filters to optimal filter routers using three factors: 1) hop count from a sender, 2) the filter routers resource availability, and 3) the filter routers link degree. Our filter scheduling policy (modified ARC), which considers the filter weights between frequency and recency, maintains the best-k filters that maximize the effectiveness. We evaluate APFS and compare it with the conventional filter-based approach. APFS shows 44% more effective than AITF, since APFS is resistant to filter flooding attacks. Furthermore, thanks to its incorporation of adaptive packet marking based on the three factors, APFS can quickly propagate filters to upstream filter routers so that malicious traffic are blocked close to attack sources.

Acknowledgments

This research was supported by the R&BD Support Center of Seoul Development Institute and the South Korean government (WR080951). The preliminary version of this paper was presented in the 36th IEEE Local Computer Networks (LCN 2011) (Seo et al., 2011).

REFERENCES

Aljifri H, Smets M, Pons AP. IP Traceback using header compression. Comput Secur 2003;22(2):136–51.
Arbor Networks. Worldwide infrastructure security report; 2013.
Argyraki KJ, Cheriton DR. Active internet traffic filtering: real-time response to denial-of-service attacks. In: USENIX annual technical conference, general track 2005. p. 135–48.
Argyraki KJ, Cheriton DR. Scalable network-layer defense against internet bandwidth-flooding attacks. IEEE/ACM Trans Netw 2009;17(4):1284–97.
Bai C, Feng G, Wang G. Algebraic geometric code based IP traceback. In: Performance, computing, and communications, 2004 IEEE international conference on 2004. p. 49–56.

Belenky A, Ansari N. On deterministic packet marking. Comput Netw 2007;51(10):2677–700.
Bellovin S, Leech M, Taylor T. The ICMP traceback message. In: IETF internet draft 2001.
Braga R, Mota E, Passito A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: The 35th IEEE conference on local computer networks (LCN) 2010.
CAIDA Skitter. http://www.caida.org/tools/measurement/skitter/; 2007.
Dang X-H, Albright E. An implementation of IP traceback in IPv6 using probabilistic packet marking. In: International conference on internet computing 2005. p. 416–21.
Dang X-H, Albright E, Abonamah AA. Performance analysis of probabilistic packet marking in IPv6. Comput Commun 2007;30(16):3193–202.
Davids N. Initial TTL values. http://noahdavids.org/self_published/TTL_values.html; 2011.
Dean D, Franklin MK, Stubblefield A. An algebraic approach to IP traceback. ACM Trans Inf Syst Secur 2002;5(2):119–37.
Digital Chosun. Hackers threaten cyber money sites. http://english.chosun.com/w21data/html/news/200710/200710100013.html; 2007.
Handley M, Paxson V, Kreibich C. Network intrusion detection: evasion, traffic normalization, and end-to-end protocol semantics. In: USENIX security symposium 2001.
Haris S, Waleed GM, Ahmad R, Ghani M. Anomaly detection of IP header threats. Int J Comput Sci Security, (IJCSS) 2011;4(6):497–504.
Jang J-h, Yeo D-G, Lee D-h, Youm H-Y. An IP traceback mechanism against mobile attacker for IPv6 and PMIPv6. In: Proceedings of the 11th international conference on information security applications 2011. p. 150–9.
Jin C, Wang H, Shin K. Hop-count filtering: an effective defense against spoofed DDoS traffic. In: Proc. of the 10th ACM conference on computer and communication security 2003. p. 30–41.
Kang J, Zhang Z, bin Ju J. Protect e-commerce against DDoS attacks with improved d-WARD detection system. In: IEEE international conference on e-technology, e-commerce, and e-services (EEE) 2005. p. 100–5.
Kessler GC. Defenses against distributed denial of service attacks. http://www.garykessler.net/library/ddos.html; 2000.
Korea Herald. Election watchdogs web site attacked before elections. http://nwww.koreaherald.com/view.php?ud=20120411000153; 2012.
Korea Times. Hacking attacks. http://www.koreatimes.co.kr/www/news/opinon/2009/07/137_48133.html; 2009.
KrishnaKumar B, Kumar PK, Sukanesh R. Hop count based packet processing approach to counter DDoS attacks. In: Proceedings of the 2010 international conference on recent trends in information, telecommunication and computing (ITC) 2010. p. 271–3.
Lee HCJ, Ma M, Thing VLL, Xu Y. On the issues of IP traceback for IPv6 and mobile IPv6. In: In proceedings of the 8th IEEE international symposium on computers and communication (ISCC) 2003.
Lee H, Kwon M, Hasker G, Perrig A. BASE: an incrementally deployable mechanism for viable IP spoofing prevention. In: ASIACCS 2007. p. 20–31.
Liu X, Yang X, Lu Y. To filter or to authorize: network-layer DoS defense against multimillion-node botnets. In: SIGCOMM 2008. p. 195–206.
Mahajan R, Bellovin SM, Floyd S, Ioannidis J, Paxson V, Shenker S. Controlling high bandwidth aggregates in the network. Comput Commun Rev 2002;32(3):62–73.
Megiddo N, Modha DS. ARC: a self-tuning, low overhead replacement cache. In: FAST 2003.
Meyer D. University of Oregon route views archive project. http://archive.routeviews.org; 2006.
Mirkovic J, Prier G, Reiher PL. Attacking DDoS at the source. In: ICNP 2002. p. 312–21.
NS-2. The network simulator – Ns-2. http://www.isi.edu/nsnam/ns; 2008.
Park K, Lee H. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. In: SIGCOMM 2001. p. 15–26.
Parno B, Wendlandt D, Shi E, Perrig A, Maggs BM, Hu Y-C. Portcullis: protecting connection setup from denial-of-capability attacks. In: SIGCOMM 2007. p. 289–300.
Paxson V. Bro: a system for detecting network intruders in real-time. Comput Netw 1999;31:23–4. 2435–2463.
Roesch M. Snort: lightweight intrusion detection for networks. In: LISA 1999. p. 229–38.
Saurabh S, Sairam AS. Linear and remainder packet marking for fast IP traceback. In: Fourth international conference on communication systems and networks (COMSNETS) 2012. p. 1–8.
Savage S, Wetherall D, Karlin A, Anderson T. Practical network support for IP traceback. In: SIGCOMM 2000. p. 295–396.
Seo D, Lee H, Perrig A. PFS: probabilistic filter scheduling against distributed denial-of-service attacks. In: The 36th IEEE conference on local computer networks (LCN) 2011. p. 9–17 [Best paper award].
Spring N, Mahajan R, Wetherall D. Measuring ISP topologies with rocketfuel. SIGCOMM Comput Commun Rev 2002;32(4). ISSN: 0146-4833:133–45.
Sung M, Xu J. IP traceback-based intelligent packet filtering: a novel technique for defending against internet DDoS attacks. IEEE Trans Parallel Distrib Syst 2003;14(9):861–72.
Wang H, Jin C, Shin KG. Defense against spoofed IP traffic using hop-count filtering. IEEE/ACM Trans Netw 2007;15(1):40–53.
Yaar A, Perrig A, Song DX. Pi: a path identification mechanism to defend against DDoS attack. In: IEEE symposium on security and privacy 2003. p. 93.
Yaar A, Perrig A, Song DX. SIFF: a stateless internet flow filter to mitigate DDoS flooding attacks. In: IEEE symposium on security and privacy 2004.
Yang X, Wetherall D, Anderson TE. TVA: a DoS-limiting network architecture. IEEE/ACM Trans Netw 2008;16(6):1267–80.
Yau DKY, Lui JCS, Liang F, Yam Y. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. IEEE/ACM Trans Netw 2005;13(1):29–42.
Zhou S, Mondragón RJ. Accurately modeling the internet topology. CoRR cs.NI/0402011.

Dongwon Seo received the B.S. and M.S. degree in computer science and en- gineering from Korea University in Seoul, Korea, in 2007 and 2009, respectively. He was an intern at Cylab in Carnegie Mellon University, 2010. He is currently working toward doctorate degree in computer and communication engineering at Korea University in Seoul, Korea. His research interests include network and system security such as DDoS defense and software attestation.

Heejo Lee is a Professor at the Division of Computer and Communication Engineering, Korea University, Seoul, Korea. Before joining Korea University, he was at AhnLab,Inc. as a CTO from 2001 to 2003. From 2000 to 2001, he was a postdoctorate at CERIAS, Purdue University. Dr.Lee received his B.S., M.S., Ph.D. degrees in Computer Science and Engineering from POSTECH, Pohang, Korea. Dr.Lee serves as an editor of the Journal of Communications and Networks. He has been an advisory member of Korea Internet Security Agency and Korea Supreme Prosecutor's Office.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002

Adrian Perrig is a Full Professor at the Department of Computer Science at ETH Zurich. Formerly a Professor in Electrical and Computer Engineering, Engineering and Public Policy, and Computer Science at Carnegie Mellon Uni- versity, Dr. Perrig earned his Ph.D. degree in Computer Science from Carnegie Mellon University, and spent three years during his Ph.D. degree at University of California at Berkeley. He received his B.Sc. degree in Computer Engineering from the Swiss Federal Institute of Technology in Lausanne (EPFL). Adrian's research interests revolve around building secure systems and include Internet security, security for sensor networks and mobile applications.

Please cite this article in press as: Seo D, et al., APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks, Computers & Security (2013), http://dx.doi.org/10.1016/j.cose.2013.09.002