Traceback of DDoS Attacks Using Entropy Variations

Shui Yu, *Member*, *IEEE*, Wanlei Zhou, *Senior Member*, *IEEE*,
Robin Doss, *Member*, *IEEE*, and Weijia Jia, *Senior Member*, *IEEE*

Abstract—Distributed Denial-of-Service (DDoS) attacks are a critical threat to the Internet. However, the memoryless feature of the Internet routing mechanisms makes it extremely hard to trace back to the source of these attacks. As a result, there is no effective and efficient method to deal with this issue so far. In this paper, we propose a novel traceback method for DDoS attacks that is based on entropy variations between normal and DDoS attack traffic, which is fundamentally different from commonly used packet marking techniques. In comparison to the existing DDoS traceback methods, the proposed strategy possesses a number of advantages—it is memory nonintensive, efficiently scalable, robust against packet pollution, and independent of attack traffic patterns. The results of extensive experimental and simulation studies are presented to demonstrate the effectiveness and efficiency of the proposed method. Our experiments show that accurate traceback is possible within 20 seconds (approximately) in a large-scale attack network with thousands of zombies.

Index Terms—DDoS, IP traceback, entropy variation, flow.

1 Introduction

It is an extraordinary challenge to traceback the source of Distributed Denial-of-Service (DDoS) attacks in the Internet. In DDoS attacks, attackers generate a huge amount of requests to victims through compromised computers (zombies), with the aim of denying normal service or degrading of the quality of services. It has been a major threat to the Internet since year 2000, and a recent survey [1] on the largest 70 Internet operators in the world demonstrated that DDoS attacks are increasing dramatically, and individual attacks are more strong and sophisticated. Furthermore, the survey also found that the peak of 40 gigabit DDoS attacks nearly doubled in 2008 compared with the previous year. The key reason behind this phenomena is that the network security community does not have effective and efficient traceback methods to locate attackers as it is easy for attackers to disguise themselves by taking advantages of the vulnerabilities of the World Wide Web, such as the dynamic, stateless, and anonymous nature of the Internet [2], [3]. IP traceback means the capability of identifying the actual source of any packet sent across the Internet. Because of the vulnerability of the original design of the Internet, we may not be able to find the actual hackers at present. In fact, IP traceback schemes are considered successful if they can identify the zombies from which the DDoS attack packets entered the Internet. Research on DDoS detection [4], [5], [6], [7], [8], [9], mitigation [10], [11], [12], and filtering [13], [14], [15], [16], [17], [18] has been conducted pervasively. However, the efforts on IP traceback are limited.

A number of IP traceback approaches have been suggested to identify attackers [19], [20], and there are two major methods for IP traceback, the probabilistic packet marking (PPM) [21], [22], [23], [24] and the deterministic packet marking (DPM) [25], [26], [27], [28]. Both of these strategies require routers to inject marks into individual packets. Moreover, the PPM strategy can only operate in a local range of the Internet (ISP network), where the defender has the authority to manage. However, this kind of ISP networks is generally quite small, and we cannot traceback to the attack sources located out of the ISP network. The DPM strategy requires all the Internet routers to be updated for packet marking. However, with only 25 spare bits available in as IP packet, the scalability of DPM is a huge problem [22]. Moreover, the DPM mechanism poses an extraordinary challenge on storage for packet logging for routers [29]. Therefore, it is infeasible in practice at present. Further, both PPM and DPM are vulnerable to hacking [30], which is referred to as packet pollution.

IP traceback methods should be independent of packet pollution and various attack patterns. In our previous work [31], [32] on DDoS attack detection, we compared the packet number distributions of packet flows, which are out of the control of attackers once the attack is launched, and we found that the similarity of attack flows is much higher than the similarity among legitimate flows, e.g., flash crowds. Entropy rate, the entropy growth rate as the length of a stochastic sequence increases [33], was employed to find the similarity between two flows on the entropy growth pattern [31], and relative entropy, an abstract distance between two

- *S. Yu, W. Zhou, and R. Doss are with the School of Information Technology, Deakin University, Burwood, VIC 3125, Australia. E-mail: {syu, wanlei, rchell}@deakin.edu.au.*
- *W. Jia is with the Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong. E-mail: itjia@cityu.edu.hk.*

Manuscript received 27 Oct. 2008; revised 20 July 2009; accepted 21 Oct. 2009; published online 30 Apr. 2010.
Recommended for acceptance by M. Singhal.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-10-0433.
Digital Object Identifier no. 10.1109/TPDS.2010.97.

probabilistic mass distributions [33], was taken to measure the instant difference between two flows [32].

In this paper, we propose a novel mechanism for IP traceback using information theoretical parameters, and there is no packet marking in the proposed strategy; we, therefore, can avoid the inherited shortcomings of the packet marking mechanisms. We categorize packets that are passing through a router into *flows*, which are defined by the upstream router where a packet came from, and the destination address of the packet. During nonattack periods, routers are required to observe and record entropy variations of local flows. In this paper, we use *flow entropy variation* or *entropy variation* interchangeably. Once a DDoS attack has been identified, the victim initiates the following pushback process to identify the locations of zombies: the victim first identifies which of its upstream routers are in the attack tree based on the flow entropy variations it has accumulated, and then submits requests to the related immediate upstream routers. The upstream routers identify where the attack flows came from based on their local entropy variations that they have monitored. Once the immediate upstream routers have identified the attack flows, they will forward the requests to their immediate upstream routers, respectively, to identify the attacker sources further; this procedure is repeated in a parallel and distributed fashion until it reaches the attack source(s) or the discrimination limit between attack flows and legitimate flows is satisfied.

Our analysis, experiments, and simulations demonstrate that the proposed traceback mechanism is effective and efficient compared with the existing methods [23], [34]. In particular, it possesses the following advantages:

- The proposed strategy is fundamentally different from the existing PPM or DPM traceback mechanisms, and it outperforms the available PPM and DPM methods. Because of this essential change, the proposed strategy overcomes the inherited drawbacks of packet marking methods, such as limited scalability, huge demands on storage space, and vulnerability to packet pollutions.
- The implementation of the proposed method brings no modifications on current routing software. Both PPM and DPM require update on the existing routing software, which is extremely hard to achieve on the Internet. On the other hand, our proposed method can work independently as an additional module on routers for monitoring and recording flow information, and communicating with its upstream and downstream routers when the pushback procedure is carried out.
- The proposed method will be effective for future packet flooding DDoS attacks because it is independent of traffic patterns. Some previous works [23] depend heavily on traffic patterns to conduct their traceback. For example, they expected that traffic patterns obey Poisson distribution or Normal distribution. However, traffic patterns have no impact on the proposed scheme; therefore, we can deal with any complicated attack patterns, even legitimate traffic pattern mimicking attacks.
- The proposed method can archive real-time traceback to attackers. Once the short-term flow information is in place at routers, and the victim notices that it is under attack, it will start the traceback procedure. The workload of traceback is distributed, and the overall traceback time mainly depends on the network delays between the victim and the attackers.

The rest of the paper is organized as follows: Section 2 describes the background of DDoS attacks and the related work which has been done so far on IP traceback. Our entropy variation-based IP traceback model is proposed in Section 3. Detailed analysis of the proposed scheme is conducted in Section 4. The related algorithms are designed in Section 5. Section 6 focuses on the performance analysis for every aspect of the proposed mechanism with Section 7 summarizing the paper and discussing future work.

2 Background and Related Work

2.1 Background of DDoS Attacks

DDoS attacks are targeted at exhausting the victim's resources, such as network bandwidth, computing power, and operating system data structures. To launch a DDoS attack, the attacker(s) first establishes a network of computers that will be used to generate the huge volume of traffic needed to deny services to legitimate users of the victim. To create this attack network, attackers discover vulnerable hosts on the network. Vulnerable hosts are those that are either running no antivirus or out-of-date antivirus software, or those that have not been properly patched. These are exploited by the attackers who use the vulnerability to gain access to these hosts. The next step for the attacker is to install new programs (known as *attack tools*) on the compromised hosts of the attack network. The hosts running these attack tools are known as *zombies*, and they can be used to carry out any attack under the control of the attacker. Numerous zombies together form an *army* or *botnet* [3], [35].

There are two categories of DDoS attacks, typical DDoS attacks and Distributed Reflection Denial-of-Service (DRDoS) attacks. In a typical DDoS attack, the master computer orders the zombies to run the attack tools to send huge volume of packets to the victim, to exhaust the victim's resources. Unlike the typical DDoS attacks, the army of a DRDoS attack consists of master zombies, slave zombies, and reflectors. The difference in this type of attack is that slave zombies are led by master zombies to send a stream of packets with the victim's IP address as the source IP address to other uninfected machines (known as *reflectors*), exhorting these machines to connect with the victim. Then the reflectors send the victim a great volume of traffic, as a reply to its exhortation for the opening of a new connection, because they believe that the victim was the host that asked for it.

Understanding the features of DDoS attack is critical for effective attack traceback. However, we have limited real data sets about DDoS attacks. The current knowledge of DDoS attack can be classified as follows: inference based on partial information [34], real network emulation [35] or simulations [36], and real attack and defence between two cooperate research groups [37].

2.2 Related Work of IP Traceback

It is obvious that hunting down the attackers (zombies), and further to the hackers, is essential in solving the DDoS attack challenge. The summary of the existing DDoS traceback methods can be found in [38] and [39]. In general, the traceback strategies are based on packet marking.

Packet marking methods include the PPM and the DPM. The PPM mechanism tries to mark packets with the router's IP address information by probability on the local router, and the victim can reconstruct the paths that the attack packets went through. The PPM method is vulnerable to attackers, as pointed out in [30], as attackers can send spoofed marking information to the victim to mislead the victim. The accuracy of PPM is another problem because the marked messages by the routers who are closer to the leaves (which means far away from the victim) could be overwritten by the downstream routers on the attack tree [21]. At the same time, most of the PPM algorithms suffer from the storage space problem to store large amount of marked packets for reconstructing the attack tree [22], [24]. Moreover, PPM requires all the Internet routers to be involved in marking.

Based on the PPM mechanism, Law et al. tried to traceback the attackers using traffic rates of packets, which were targeted on the victim [23]. The model bears a very strong assumption: the traffic pattern has to obey the Poisson distribution, which is not always true in the Internet. Moreover, it inherits the disadvantages of the PPM mechanism: large amount of marked packets are expected to reconstruct the attack diagram, centralized processing on the victim, and it is easy be fooled by attackers using packet pollution.

The deterministic packet marking mechanism tries to mark the spare space of a packet with the packet's initial router's information, e.g., IP address. Therefore, the receiver can identify the source location of the packets once it has sufficient information of the marks. The major problem of DPM is that it involves modifications of the current routing software, and it may require very large amount of marks for packet reconstruction. Moreover, similar to PPM, the DPM mechanism cannot avoid pollution from attackers.

Savage et al. [24] first introduced the probability-based packet marking method, node appending, which appends each node's address to the end of the packet as it travels from the attack source to the victim. Obviously, it is infeasible when the path is long or there is insufficient unused space in the original packet. The authors proposed the node sampling algorithm, which records the router address to the packet with probability, p, on the routers of the attack path. Then, the probability of a packet marked by a router d that hops away from the victim is $p(1-p)^{d-1}$. Based on the number of marked packets, we can reconstruct the attack path. However, it requires large number of packets to improve the accuracy of the attack path reconstruction. Therefore, an edge sampling algorithm was proposed to mark the start router address and end router address of an attack link and the distance between the two ends. The edge sampling algorithm fixed the problems of the node sampling algorithm to some extent.

Based on the PPM mechanism, in [23], the traffic that targeted the victim was measured to construct the attack diagram, and then identified where the attackers were located. They focused on the traffic flows, which end at the victim, and therefore, there was a tree which was rooted at the victim. For a router on the attack tree, the outgoing flow included two parts: the locally generated flows and the transit flows from the upstream router(s) of the attack tree. If X_1 and X_2 are two flows on the attack tree, and X_1 is the upstream flow of X_2, then $\Pr ob[X_1 > x] \geq \Pr ob[X_2 > x], \forall x$. The victim will collect all the marked packets from the routers and reconstruct the attack tree based on the traffic rates of the different routers. This traceback method heavily depends on the queuing model, and it requires the traffic flows to obey specific patterns, e.g., the Poisson distribution.

In [22], the randomize-and-link approach to implement IP traceback based on the probabilistic packet marking mechanism was proposed. The algorithm targets two aspects: to reconstruct the marks from the marker efficiently and to make the PPM more secure against hackers' pollution. The idea is to have every router X to fragment its unique message M_x (e.g., IP address) into several pieces, $M_0, M_1, \ldots M_l$. At the same time, the router calculates the checksum $C = C(M_x)$, named as *cord*. The router assembles the mark as b_i, and injects b_i randomly into the unused IPv4 packet header (say, N bits, which is 25 bits in the paper: 16 bits of fragmentation ID, 1 bit of the fragmentation index, and 8 bits of service type, all of them are used rarely in a common IPv4 packet). b_i includes three parts: an index of the pieces ($\log_2 l$ bits), a large checksum cord $C = C(M_x)(N - \log_2 l - |M_i|$bits), and a piece of $M_i, i = 0, 1, \ldots, l$ ($|M_i|$ bits). The cord is quite large, for example, 14 out of 25 bits, therefore, we can treat the cord as a random number, which is hard for hackers to predict. The victim can reconstruct the message efficiently by checking the cord and the index sequence.

Yaar et al. [40] studied the marking technique to improve the PPM mechanism. They broke the 16-bits marking space into three parts: 1 bit for distance, 2 bits for fragmentation index, and a hash fragmentation of 13 bits. By this modification, the proposed FIT algorithm can traceback the attack paths with high probability after receiving only tens of packets. The FIT algorithm also performed well even in the presence of legacy routers and it is a scalable algorithm for thousands of attack sources.

Snoeren et al. proposed a method by logging packets or digests of packets at routers [41], [42]. The packets are digested using bloom filter at all the routers. Based on these logged information, the victim can traceback the leaves on an attack tree. The methods can even traceback a single packet. However, it also places a significant strain on the storage capability of intermediate routers.

In [21], two hybrid schemes that combine the packet marking and packet logging method to traceback the attack sources are proposed—Distributed Link-List Traceback (DLLT) and the Probabilistic Pipelined Packet Marking (PPPM). The first one preserves the marking information at intermediate routers in a specific way so that it can be collected using a link-list-based approach. The second algorithm targets propagating the IP addresses of the routers that were involved in marking certain packets by loading them into packets going to the same destination,

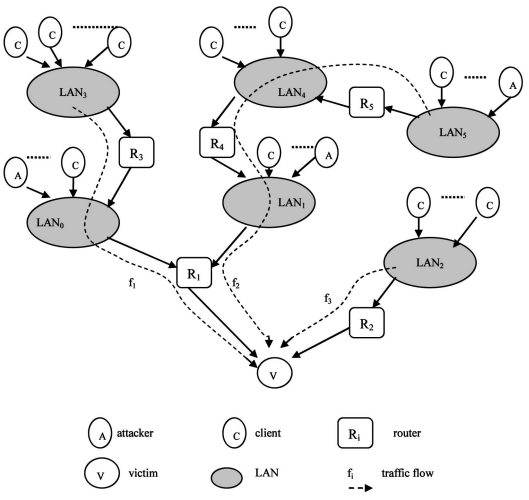


Fig. 1. A sample network with DDoS attacks.

therefore, preserving these addresses while avoiding the need for long-term storage at the intermediate routers.

Different from PPM, Dean et al. [26] proposed a deterministic packet marking strategy for IP traceback. Every ingress router writes its own IP address into the outgoing IP packet header, and there is no more marking for the packet. They used an algebraic approach, originally developed for coding theory and learning theory, for encoding traceback information. Their idea is that for any polynomial $f(x)$ of degree d in the prime field $GF(p)$, $f(x)$ can be recovered given $f(x)$ evaluated at $d+1$ unique points.

Belenky and Ansari [25] noticed that the PPM mechanism can only solve large flooding attacks, and it is not applicable for attacks consisted of a small number of packets. Moreover, PPM is vulnerable if hackers inject marked packets into the network. Therefore, the paper proposed a deterministic packet marking method for IP traceback. The basic idea is that at the initial router for an information source, the router embeds its IP address into the packet by chopping the router's IP into two segments with 17 bits each (16 bits for half of the IP address and 1 bit works as index). As a result, the victim can trace which router the packets came from.

Jin and Yang [27] improved the ID coding of the deterministic packet marking scheme using redundant decomposition of the initial router IP address. For an IP address, they divided them into three redundant segments, 0-13 bits, 9-22 bits, and 18-31 bits, and then five different hash functions are applied on the three segments to create five results. The resulting eight segments are recorded in the outgoing packets randomly. The victim can reassemble the source router IP using the packets it has received.

3 System Modeling for IP Traceback on Entropy Variations

3.1 A Sample Network with DDoS Attacks

In order to clearly describe our traceback mechanism, we use Fig. 1 as a sample network with DDoS attacks to demonstrate our traceback strategy.

In a DDoS attack scenario, as shown in Fig. 1, the flows with destination as the victim include legitimate flows, such as f_3, and a combination of attack flows and legitimate flows, such as f_1 and f_2. Compared with nonattack cases, the volumes of some flows increase significantly in a very short time period in DDoS attack cases. Observers at routers R_1, R_4, R_5, and V will notice the dramatic changes; however, the routers who are not in the attack paths, such as R_2 and R_3, will not be able to sense the variations. Therefore, once the victim realizes an ongoing attack, it can pushback to the LANs, which caused the changes based on the information of flow entropy variations, and therefore, we can identify the locations of attackers.

The traceback can be done in a parallel and distributed fashion in our proposed scheme. In Fig. 1, based on its knowledge of entropy variations, the victim knows that attackers are somewhere behind router R_1, and no attackers are behind router R_2. Then the traceback request is delivered to router R_1. Similar to the victim, router R_1 knows that there are two groups of attackers, one group is behind the link to LAN_0 and another group is behind the link to LAN_1. Then the traceback requests are further delivered to the edge routers of LAN_0 and LAN_1, respectively. Based on entropy variation information of router R_3, the edge router of LAN_0 can infer that the attackers are located in the local area network, LAN_0. Similarly, the edge router of LAN_1 finds that there are attackers in LAN_1; furthermore, there are attackers behind router R_4. The traceback request is then further passed to the upstream routers, until we locate the attackers in LAN_5.

3.2 System Modeling

In this paper, we categorize the packets that are passing through a router into *flows*. A flow is defined by a pair—the upstream router where the packet came from, and the destination address of the packet. Entropy is an information-theoretic concept, which is a measure of randomness. We employ *entropy variation* in this paper to measure changes of randomness of flows at a router for a given time interval. We notice that entropy variation is only one of the possible metrics. Chen and Hwang used a statistical feature, change-point of flows, to identify the abnormality of DDoS attacks [6]; however, attackers could cheat this feature by increasing attack strength slowly. We can also employ other statistic metrics to measure the randomness, such as standard variation or high-order moments of flows. We choose entropy variation rather than others in this paper because of the low computing workload for entropy variations.

First, let us have a close investigation on the flows of a router, as shown in Fig. 2. Generally, a router knows its local topology , e.g., its upstream routers, the local area network attached to the router, and the downstream routers.

We name the router that we are investigating now as a *local router.* In the rest of the paper, we use I as the set of positive integers, and R as the set of real numbers. We denote a *flow* on a local router by $<u_i, d_j, t>, i, j \in I, t \in R$, where u_i is an upstream router of a local router R_i, d_j is the destination address of a group of packets that are passing through the local router R_i, and t is the current time stamp. For example, the local router R_i in Fig. 2 has two different incoming flows—the ones from the upstream routers R_j and R_k, respectively. We name this kind of flows as *transit* flows. Another type of incoming flows of the local router R_i is

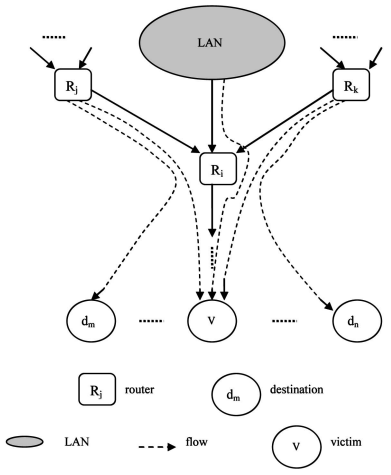


Fig. 2. Traffic flows at a router on an attack path.

generated at the local area network; we call these *local flows*, and we use L to represent the local flows. We name all the incoming flows as *input flows*, and all the flows leaving router R_i are named as *output flows*. We denote $u_i, i \in I$ as the immediate upstream routers of the local router R_i, and set U as the set of incoming flows of router R_i. Therefore, $U = \{u_i, i \in I\} + \{L\}$. We use a set $D = \{d_i, i \in I\}$ to represent the destinations of the packets that are passing through the local router R_i. If v is the victim router, then $v \in D$. Therefore, a flow at a local router can be defined as follows:

$$f_{ij}(u_i, d_j) = \{<u_i, d_j, t> | u_i \in U, d_j \in D, i, j \in I\}. \quad (1)$$

We denote $|f_{ij}(u_i, d_j, t)|$ as the count number of packets of the flow f_{ij} at time t. For a given time interval ΔT, we define the variation of the number of packets for a given flow as follows:

$$N_{ij}(u_i, d_j, t + \Delta T) = |f_{ij}(u_i, d_j, t + \Delta T)| - |f_{ij}(u_i, d_j, t)|. \quad (2)$$

If we set $|f_{ij}(u_i, d_j, t)| = 0$, then $N_{ij}(u_i, d_j, t + \Delta T)$ is the number of packets of flow f_{ij}, which went through the local router during the time interval ΔT. In order to make the presentation tidy, we use $N_{ij}(u_i, d_j)$ to represent $N_{ij}(u_i, d_j, t + \Delta T)$ in the rest of this paper.

Based on the large number theorem, we have the probability of each flow at a local router as follows:

$$p_{ij}(u_i, d_j) = \frac{N_{ij}(u_i, d_j)}{\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} N_{ij}(u_i, d_j)}, \quad (3)$$

where $p_{ij}(u_i, d_j)$ gives the probability of the flow f_{ij} over all the flows on the local router, and $\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} p_{ij}(u_i, d_j) = 1$.

Let F be the random variable of the number of flows during the time interval ΔT on a local router, therefore, we define the entropy [33] of flows for the local router as follows:

$$H(F) = -\sum_{i,j} p_{ij}(u_i, d_j) \log p_{ij}(u_i, d_j). \quad (4)$$

In order to differentiate from the original definition of entropy, we call *H(F)* as *entropy variation* in this paper, which measures the variations of randomness of flows on a given local router.

4 Traceback Model Analysis

In this section, we first compare the proposed model with the existing proposals in order to show the advantages of the proposed mechanism. We then analyze the proposed entropy variation-based traceback model in detail. The features of a stand-alone router are analyzed first, followed by the investigation on the properties of the whole attack tree of a DDoS attack.

4.1 Comparison of Traceback Models

In order to show the advantages of the proposed mechanism, we compare our proposed method with the representatives of DPM [42] and PPM [22] algorithms. The settings and network environment for the proposed algorithm are the same as that of DPM [42] and PPM [22], respectively, in the comparisons. We take [42] as the representative for DPM mechanism because it is a typical research instance for that category. It chooses one source (attacker) and one destination randomly from a tier-one ISP made up of roughly 70 backbone routers with links ranging from T1 to OC-3. The routers between the source and the destination perform packet digests using a bloom filter, and the average packet size is 400 bytes as indicated in the paper. Routers process more than 20 Mpkts/sec (roughly 2 OC-192 links, or 8 OC-48s), and there are around 1,000 flows at a router. We use [22] as an instance for the PPM strategy, which is treated as the most scalable PPM algorithm, and we calculate the related storage space and traceback time as the parameters provided by the paper, such as sampling probability $p = 0.05$. Both of these two schemes are used as the benchmark in [43] as well. The comparisons are listed in Table 1, and it shows clearly that the proposed mechanism outperforms the other two mechanisms in terms of scalability (the size of attack network that we can handle), storage (the storage space that we need on routers or victims to conduct IP traceback), traceback time (the overall time we need from the start time until the end of tracing process), and the operation workload (the operations on possible routers or victims).

There are some improvements for DPM by distributing logging information among routers [29] and PPM by reducing the probability of sampling [43]. However, there are no fundamental changes, and the improvements are limited compared to our proposed methodology.

4.2 Analysis of Entropy-Variation-Based Traceback Model

We present our assumptions below in order to make our analysis simple and clear. We assume the following:

1. There is no extraordinary change of network traffic in a very short time interval (e.g., at the level of seconds) for non-DDoS attack cases. It is true that the network traffic for a router may dynamically change a lot from peak to off-peak service times. However, this kind of change lasts for a relatively long time interval, e.g., at least at the level of minutes. If we break down these changes into seconds, the change of traffic is quite smooth in our context.
2. The number of attack packets is at least an order of magnitude higher than that of normal flows. During a DDoS flooding attack, the number of attack packets

TABLE 1
The Comparison of the Entropy Variation Mechanisms against DPM and PPM
(with the Same Settings and Network Environment, Respectively)

	DPM	PPM	Entropy Variation
Scalability	**High** 2^{17}-2^{25} computers for single packet marking, more or multiple packet marking	**Low** 100 routers range of attack tree	**Very High** Unlimited under condition that every zombie generates obvious traffic
Storage	**Very High** 3.3G-44G/minute at each involved router	**High** Around 900M at the victim for one attack	**Very low** Around 240k/minute at involved routers
Traceback Time	**Low** Network delay	**Medium** Network delay plus calculation time	**Low** Network delay
Operation Workload	**Very High** Digesting packets with probability P (about 1M packets/second)	**Very High** Marking packets with probability P (about 1M packets/second)	**Very Low** Counting packet numbers for each flow

increases dramatically, and the attack packets are generated by thousands of zombies or bots [3], [34]. Consequently, the number of attack packets is much higher than that of legitimate flows. Therefore, this assumption is reasonable. Of course, for the nonflooding attacks, this may not hold, and in this paper, we focus on the majority of the attack tools—flooding attacks. Furthermore, this is the lower bound that we can discriminate attack flows from the legitimate flows (see the experiments in Section 6).

3. Only one DDoS attack is ongoing at a given time. It could be true that a number of attacks are ongoing concurrently in the Internet, the attack paths may overlap as well, but we only consider the one attack scenario to make it simple and clear.
4. The number of flows for a given router is stable at both the attack cases and nonattack cases.

For a local router, suppose that the number of flows is N, and the probability distribution is $P\{p_1, p_2, \ldots, p_N\}$. We can simplify the expression of entropy of (4) as follows:

$$H(F) = H(p_1, p_2, \ldots, p_N) = -\sum_{i=1}^{N} p_i \log p_i. \tag{5}$$

Based on the characteristics of the entropy function [33], we obtain the upper bound and lower bound of *H(F)* as follows:

$$0 \le H(F) \le \log N. \tag{6}$$

We reach the lower bound when $p_i = 1, 1 \le i \le N$, $p_k = 0, k = 1, 2, \ldots, N$, and $k \ne i$; we have the upper bound when $p_1 = p_2 = \cdots = P_N$. Based on our definition of the random variable of flows, we have the following special cases to reach the lower bound and the upper bound, respectively: when there is only one flow alive during the sampling time interval, and there are no packets going through the local router for the other flows, $H(F) = 0$; when the number of packets for each flow is the same among all the flows at a local router, then we have $H(F) = \log N$.

We divide our timeline into two segments for the following investigation: before DDoS attack and under DDoS attack. The local router's entropy variation is, therefore, denoted by $H^-(F)$ and $H^+(F)$, respectively. Let δ be a reasonable threshold, and C be the mean of $H^-(F)$, and the standard variation of $H^-(F)$ be σ. We know that $H^-(F)$ is quite stable for a long time period. We justify our threshold δ to make the following equation holds with high probability:

$$|H^-(F) - C| \le \delta. \tag{7}$$

In order to make the mean C and standard variation δ adaptive to the network traffic variations, let

$$C[t] = \sum_{i=1}^{n} \alpha_i \cdot C[t-i], \ \sum_{i=1}^{n} \alpha_i = 1, \tag{8}$$

$$\sigma[t] = \sum_{i=1}^{n} \beta_i \cdot \sigma[t-i], \ \sum_{i=1}^{n} \beta_i = 1, \tag{9}$$

where C[t] represents the current mean, C[t − i] is the mean of the ith sample instance in the near past, and $\alpha_i, i = 1, 2, \ldots, n$ are the weights for the n past samples, respectively. In order to reflect the nearest changes, let $\alpha_i > \alpha_j \ for \ i < j, i, j \in I$. The values of $\alpha_i (i = 1, 2, \ldots, n)$ are fixed and could be decided by the experiments of nonattack cases. The same for $\sigma[t]$, $\sigma[t-i]$ and $\beta_i, i = 1, 2, \ldots, n$, respectively. The evolutions will be suspended when a DDoS attack is ongoing.

If an attack flow is going through a local router, then the following equation holds with high probability:

$$|H^+(F) - C| > \delta. \tag{10}$$

Moreover, we know that the reason behind this is that the packet numbers of flows $<u_i, v>, u_i \in U$ increase significantly. In order to find the immediate sources of the attack flows from the upstream routers, we sort the flows $<u_i, v>, u_i \in U$ in terms of number of packets of a given attack flow, $N_{iv}(u_i, v)$. We calculate the entropy variation

reiteratively by taking the suspicious flows out starting with the flow which has the greatest packet number, until the difference between the entropy of the remaining flows and the mean is less than or equal to the threshold, δ. In other words, the process stops when the following equation holds:

$$|H^+(F\backslash \max(\{<u_i, v>\}) - C| \leq \delta, \tag{11}$$

where $F\backslash \max(\{<u_i, v>\}$ means taking the maximum element from set $\{<u_i, v>\}$ from set F. Then the subset $\{u_i\} \subseteq U$, which includes the upstream routers that we have taken out before (11) holds, is the set of suspicious immediate sources of the DDoS attack. Then the traceback requests are further forwarded to the elements of set $\{u_i\}$, respectively. The traceback processing terminates under the following conditions:

$$\begin{cases} L = \max\{<u_i, v>\}, \\ |H^+(F\backslash L) - C| \leq \delta. \end{cases} \tag{12}$$

Then L, the flows of the local area network, is the attack source of that branch on the attack tree.

The threshold δ is important for us to make the decision, and it also introduces possible false positive and false negative. Suppose that δ is the true value of the threshold, and we have $0 \leq \delta_1 \leq \delta \leq \delta_2 < \infty$. We denote the probability density function as a continuous function $f(x)$, then the probability of false positive and false negative can be expressed as follows:

$$P_{FP}(x, \delta, \delta_1) = \int_{-\delta}^{-\delta_1} f(x)dx + \int_{\delta_1}^{\delta} f(x)dx, \tag{13}$$

$$P_{FN}(x, \delta, \delta_2) = \int_{-\delta_2}^{-\delta} f(x)dx + \int_{\delta}^{\delta_2} f(x)dx, \tag{14}$$

where $f(x)$ could be Gaussian distribution in practice. Equation (13) gives the probability of false positive when the chosen threshold δ_1 is less than the true value δ; while (14) gives the probability of false negative when the chosen threshold δ_2 is greater than the true value δ.

For a nonattack case, the number of packets of each flow is stable as we assumed; therefore, the entropy variation $H(F)$ is stable with minor fluctuations, namely (7) holds with high probability. Our traceback procedure starts when a DDoS attack alarm has been raised. We now investigate the features of the entropy variation when a DDoS attack is ongoing.

Lemma 1. *Compared with the nonattack scenario, the upper bound of entropy variation drops when DDoS attack flows are passing through a local router.*

Proof. Based on (6), we know that entropy variation reaches the maximum, *logN*, when the distribution is even, namely $p_1 = p_2 = \cdots = P_N$, and it reaches the minimum, 0, when the distribution is extremely uneven, say, $p_i = 1, 1 \leq i \leq N$, $p_k = 0, k = 1, 2, \ldots, N$, and $k \neq i$. We also know that entropy is a monotonic function [33]. Therefore, it is clear that when a DDoS attack occurs, the distribution moves toward the extreme uneven point; as a result, the upper bound of the entropy variation drops. □

Theorem 1. *Compared with the nonattack situation, the entropy variation of a local router drops dramatically when attack flows are passing through the local router, in other words, $H^-(F) \gg H^+(F)$.*

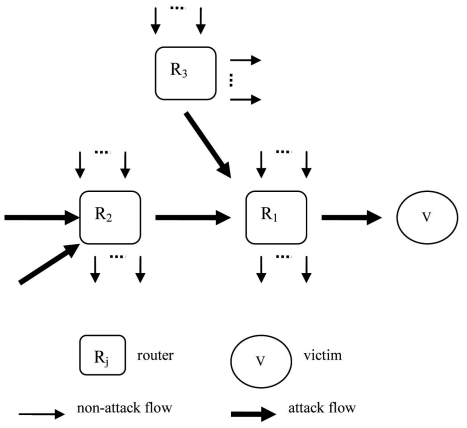


Fig. 3. A sample attack tree near the victim.

Proof. Let $f(x) = x \log x, x \geq 0$. We know that *f(x)* is a monotonically increasing convex function. Therefore, $-f(x) = -x \log x, x \geq 0$, is a monotonically decreasing concave function. Let X be the random variable for the flow distributions. Applying Jensen's inequality [33] to $f(X)$, we have $Ef(X) \geq f(EX)$, and further $-Ef(X) \leq -f(EX)$ holds. □

Let $P(X) = \{p_1, p_2, \ldots, p_N\}$ be the distribution of flows at a local router, $P(X^0) = \{p_1^0, p_2^0, \ldots, p_N^0\}$ be the distribution for the nonattack case, and $P(X^1) = \{p_1^1, p_2^1, \ldots, p_N^1\}$ be the distribution when attack flows are passing through the local router. As mentioned at the beginning of this section, we suppose that the variation of the nonattack flows remain at the same level. Let $p_i, 1 \leq i \leq N$ represent the instance of the attack flow, then $p_i^0 \ll p_i^1$. We further have $EX^0 \ll EX^1$ and $-f(EX^0) \gg -f(EX^1)$. Therefore, $-Ef(X^0) \gg -Ef(X^1)$, specifically $-\sum_{i=1}^{N} p_i^0 \log p_i^0 \gg -\sum_{i=1}^{N} p_i^1 \log p_i^1$. Hence, the result $H(X^0) \gg H(X^1)$, in other words, $H^-(F) \gg H^+(F)$.

So far, we have analyzed the characteristics of a local router with and without DDoS attacks. It is also important to have a deep understanding of the changing patterns of entropy variation among routers on an attack tree. We use the network in Fig. 3 as a sample for this study.

Lemma 2. *For a local router on an attack path, the entropy variation of the output flows is not greater than the summation of the entropy variation of the incoming flows.*

Proof. For a router in Fig. 3, we assume that there are n incoming flows, $f_1, f_2, \ldots, f_n$ (n upstream routers pump packets to one router). For any incoming flow f_i, it has n_i subflows, $f_i^1, f_i^2, \ldots, f_i^{n_i}$ (flows share one upstream router with different destinations), with the probability distribution of $p_i^1, p_i^2, \ldots, p_i^{n_i}$, respectively. The entropy variation of flow f_i is given by the following equation:

$$H(f_i) = H\big(p_i^1, p_i^2, \ldots, p_i^{n_i}\big). \tag{15}$$

□

Without loss of generality, let f_i^1 be the attack flow. Then the output flow entropy variation is $H(\sum_{i=1}^{n} f_i)$. Based on

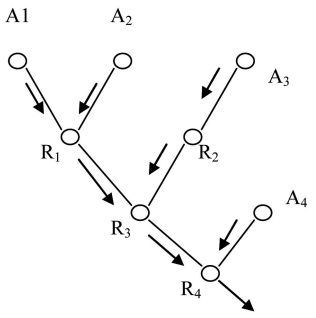


Fig. 4. A sample traceback branch in an attack tree.

Jensen's inequality, $Ef \geq f(EX)$ when f is convex, and the fact that *H(p)* is convex implies that

$$\sum_{i=1}^{n} H(f_i) \geq H\left(\sum_{i=1}^{n} f_i\right). \tag{16}$$

Theorem 2 (Entropy variation convergence of attack flows). *The entropy variation drops when a local router is closer to the victim, and vice versa.*

This theorem can be easily concluded from lemma 2 in an attack tree.

With the progress of the traceback procedure, we have more and more information about the ongoing attack. We can, therefore, estimate the number of attackers, which are going to be traced and the distances between attackers and the current local router. We use Fig. 4 as a sample attack branch for this investigation.

In Fig. 4, the traceback request has been submitted to router R_4, and from the diagram, we know that four attackers, A_1, A_2, A_3, and A_4, are to be traced, and the length of the most far away zombies is three hops away. Unfortunately, this knowledge is not available to our traceback algorithm. However, we can estimate the number of zombies that are located behind router R_4, and the maximum of the length to the most far away zombie(s) with the knowledge of the attack that we have collected so far on the way to router R_4.

Theorem 3 (Estimation on traceback distance and number of zombies). *Based on the partial information of the attack that the traceback algorithm has accumulated, we can estimate the number of zombies to be traced and the maximum length to the most far away zombie(s).*

Suppose that the zombies are distributed evenly, the attack tree is a *d-branch* tree, and we have knowledge of k attack packet rates, $a_1, a_2, \ldots, a_k$ when we reach the current router, say, router R_4 in Fig. 4. Moreover, we also know the output attack packet rate of the current local router, N. We use n to denote the number of zombies to be traced.

We use $p_i, 1 \leq i \leq k$ to represent the distribution of $t_i, 1 \leq i \leq k$, then the mean of $a_1, a_2, \ldots, a_k$ is close enough to the mean of all zombies in the whole attack tree if k is sufficient enough, in this case, $\overline{a} = \sum_{i=1}^{k} p_i \cdot a_i$. Therefore, in terms of statistics, we can infer the number of attackers to be traced in the branch as (17)

$$n \approx \frac{N}{\overline{a}} = \frac{N}{\sum_{i=1}^{k} p_i \cdot a_i}. \tag{17}$$

For a *d-branch* tree ($d = 2$ in Fig. 4), we assume that the distance from the current node (the local router) to the n leaves (zombies) is given as $l_1, l_2, \ldots, l_n$, respectively. From the Kraft inequality [33], the following equation holds:

$$\sum_{i=1}^{n} d^{-l_i} \leq 1. \tag{18}$$

Let $l_{\max} = \max(l_1, l_2, \ldots, l_n)$. Based on (17) and inequality (18), we obtain

$$l_{\max} \geq \frac{\log n}{\log d} \approx \frac{\log N - \log \sum_{i=1}^{k} p_i \cdot a_i}{\log d}. \tag{19}$$

From (17) and inequality (19), we can estimate the number of zombies to be traced and the maximum length to reach the most far away zombie(s), respectively.

Theorem 4 (Termination condition for traceback). *There are no attackers at the upstream routers if a local router's entropy variation is reasonable, namely $|H^+(F) - C| \leq \delta$ holds with high probability.*

We know that $|H^-(F) - C| \leq \delta$ holds with high probability for a local router before a DDoS attack occurs. When a DDoS attack is ongoing, however, there are no DDoS attack packets passing through the local router. Then $H^-(F) \approx H^+(F)$, and therefore, $|H^+(F) - C| \leq \delta$ holds with high probability.

Theorem 4 is quite important for the traceback algorithm to make a decision on when to terminate the pushback procedure. Of course, if the attack flows are similar to the legitimate flows in terms of packet rate, say, within 10 times range, the entropy variation cannot discriminate them. However, we are close enough to the zombies in this case.

Further, the total time for traceback, the period from starting traceback procedure to zombies are identified, is a critical parameter for any traceback algorithms. There are two reasons why this parameter is important. Moore et al. [34] indicated that the average attack duration is around 5-10 minutes for typical DDoS attacks, therefore, an effective traceback procedure has to be completed within this time limitation, say, 5 minutes; another reason is that we would be able to reduce the damage caused by DDoS attack if we could identify the zombies earlier, and therefore, block them earlier.

Suppose that the attack network is a d-branch tree, the height of the tree is n, and there are totally N zombies. Based on [34], the maximum hops between two end points at the Internet are 31. In [23], the experiments were conducted with the maximum hops as 23. However, we take 31 hops as the maximum hops between two end points on the Internet. Suppose that the zombies are distributed evenly in the attack tree. Then,

$$\sum_{i=0}^{n} d^i \leq N, \quad 1 \leq n \leq 31. \tag{20}$$

Let the normal delay between two routers in nonattack scenario be t, which is at a millisecond level usually, and the

The local flow monitoring algorithm

1. initialize the local threshold parameter, C,δ, and sampling interval ΔT;
2. identify flows, $f_1, f_2, ..., f_n$, and set count number of each flow to zero, $x_1 = x_2 = ... = x_n = 0$;
3. when ΔT is over, calculate the probability distribution and the entropy variation as follows.
   $p_i = x_i \cdot \left(\sum_{i=1}^{n} x \right)^{-1}$, $H(F) = -\sum_{i=1}^{n} p_i \log p_i$;
4. save $x_1, x_2, ..., x_n$ and $H(F)$;
5. if there is no dramatic change of the entropy variation $H(F)$, namely, $|H(F) - C| \le \delta$, progress the mean $C[t] = \sum_{i=1}^{n} \alpha_i \cdot C[t-i]$, $\sum_{i=1}^{n} \alpha_i = 1$, and the standard variation $\delta[t] = \sum_{i=1}^{n} \beta_i \cdot \delta[t-i]$, $\sum_{i=1}^{n} \beta_i = 1$
6. go to step 2.

Fig. 5. The algorithm for local flow traffic monitoring.

The IP traceback algorithm

1. initialize a set $A = \phi$, and obtain the local parameter of C and δ;
2. Let $U = \{u_i\}, i \in I$ be a set of the upstream routers, $D = \{d_i\}, i \in I$ be a set of the destinations of the packets, and v be the victim.
3. define attack flows, $f_i = <u_j, v>, i = 1,2,...,n, u_j \in U$, and sort the attack flows in the descent order, and we have $f_1', f_2', ..., f_n'$,
4. for i=1 to n
   {
   calculate $H(F \setminus f_i')$
   if $(|H(F) - C| > \delta)$ then append the responding upstream router of f_i' to set A
   else break;
   end if;
   end for;
5. submit traceback requests to the routers in set A respectively, and deliver the confirmed zombies information, set A, to the victim.

Fig. 6. The IP traceback algorithm on a router.

delay on a link be proportional to the number of packets passing through the link. Then, the traceback time can be calculated as follows:

$$T = t \times \sum_{i=0}^{n} (31 - n) \times d^i, \quad 1 \le n \le 30. \tag{21}$$

Combining (20) and (21), we can calculate the total traceback time to the most far away zombies using the proposed traceback method. The numerical results will be shown in Section 6.

5 Algorithms for the IP Traceback Model

In this section, we design the related algorithms according to our previous modeling and analysis. There are two algorithms in the proposed traceback suite, the local flow monitoring algorithm and the IP traceback algorithm.

The local flow monitoring algorithm is running at the nonattack period, accumulating information from normal network flows, and progressing the mean and the standard variation of flows. The progressing suspends when a DDoS attack is ongoing. The local flow monitoring algorithm is shown as Fig. 5.

Once a DDoS attack has been confirmed by any of the existing DDoS detection algorithms, then the victim starts the IP traceback algorithm, which is shown as Fig. 6.

The IP traceback algorithm is installed at routers. It is initiated by the victim, and at the upstream routers, it is triggered by the IP traceback requests from the victim or the downstream routers which are on the attack path.

The proposed algorithms are independent from the current routing software, they can work as independent modules at routers. As a result, we do not need to change the current routing software.

6 Performance Evaluations

In this section, we evaluate the effectiveness and efficiency of the proposed entropy variation based on IP traceback mechanism. Our first task is to show that the flow entropy variation is stable for nonattack cases, and find out the fluctuations for normal situations; the second task is to demonstrate the relationship between the drop of flow entropy variation and the increase of attack strength, so that we can identify the threshold for identifying attack sources; we further simulate the whole attack tree for traceback, and evaluate the total traceback time.

As mentioned in Section 2, the network security community lacks suitable data sets of real large-scale DDoS attacks, and it is even harder to find suitable data sets for our algorithms. Consequently, in order to evaluate our scheme, we have carefully conducted extensive simulations and real case observations. The simulation settings are arranged according to Fig. 1. We set the attack tree as a binary tree or three-branch tree, respectively, and zombies are distributed in the attack tree uniformly. We note that, our entropy variation traceback mechanism is independent from the topology of attack network and it is also independent from the network topology of victims. We use the essential DDoS attack parameters as presented in [34] in our simulations, such as, 5-10 minutes attack duration, 10,000 packets per second of attack flows. The performance evaluation included two parts—the first one focussed on the entropy variation monitoring at a local router; and the second part was to demonstrate the effectiveness of DDoS attacker traceback and the overall traceback time.

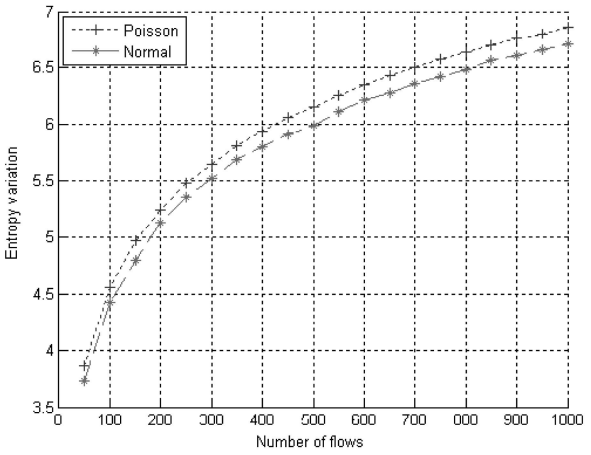


Fig. 7. The entropy variation against number of flows for Poisson and normal distributions.

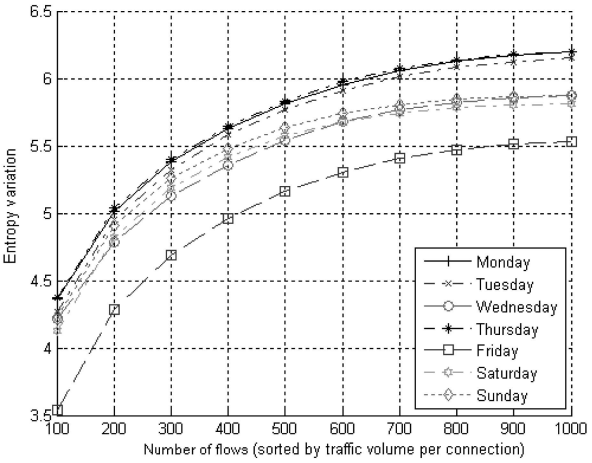


Fig. 8. Entropy variation against number of flows (sorted by traffic volumn per connection).

First, we observe the stability of entropy variations at a local router during nonattack periods. We examine two kinds of flows, the Poisson distribution flows and the Normal distribution flows. The Poisson distribution is treated as the pattern of Internet traffic by most researchers, and the combination of Normal distributions with different parameters can be used to approximate most distributions. For confidence level in the simulations, we take $\alpha = 0.05$, so that the confidence level is $1 - \alpha = 95\%$, and the corresponding confidence interval is $[-0.196, +0.196]$. We first investigated the impact on the entropy variation against the number of flows, and the results are shown in Fig. 7.

Fig. 7 indicates that the entropy variation increases smoothly against the increase of the number of flows which are passing through the local router. It also shows the similarity of the entropy variation patterns for the two distributions, and the difference of the variation entropies is quite limited.

In order to confirm this in reality, we conducted a real case study by collecting network flow information from a gateway server of our campus [44] over one week, when there is no DDoS attack. We examine the real data set to observe the patterns of flow entropy variation against number of flows for each day in that week. There were thousands of flows per day from the collected data set. We sorted the flows in different ways: by traffic volume per connection and by the number of connections for a given time interval, and considered the top 1,000 flows as input for the experiments. The results are listed in Figs. 8 and 9, respectively.

In Fig. 8, we increase the number of flows (the flows are sorted by traffic volume per connection) and check the variation of flow entropy variation on the gateway server. In Fig. 9, we sort the flows by the number of connections for different clients. We notice that the entropy variation is steady and smooth as we found in the simulation.

We therefore can conclude that the entropy variation is stable; moreover, it is independent from specific distribution patterns as shown in Fig. 7. This is one of the foundations for our proposed algorithms.

Furthermore, it is important to know the stability of the entropy variation against the fluctuations of flows in nonattack cases. We conducted two simulations for this purpose: we make the mean of flows as 100 packets per time unit, and the standard variations (std) of the flows are 25 and 50, respectively. We observe the changes of the entropy variation against the number of flows. The results are shown in Fig. 10. It indicates that the standard variation of the entropy variation is quite stable (the fluctuation is around 1-3 percent), even when the fluctuations of the flows are quite big, $\pm 25\%$ and $\pm 50\%$, respectively. Moreover, the standard variation of entropy variation decreases when the number of flows increases. For example, for a router with around 500 flows, the variation is about 0.015. As previously presented by (13) and (14), if we set our discrimination threshold δ to be less than 0.015, then we will create false positive; while when we set δ to be greater than 0.015, we will create false negative.

Based on Figs. 7 and 10, we can further conclude that the entropy variation is stable against huge flow fluctuations and number of flows in nonattack cases. Therefore, we can use it as a benchmark to discriminate DDoS attack flows.

We investigate the changes of entropy variation when a DDoS attack is ongoing. We use the term attack strength to present the packet rate of attacks. We fix the number of

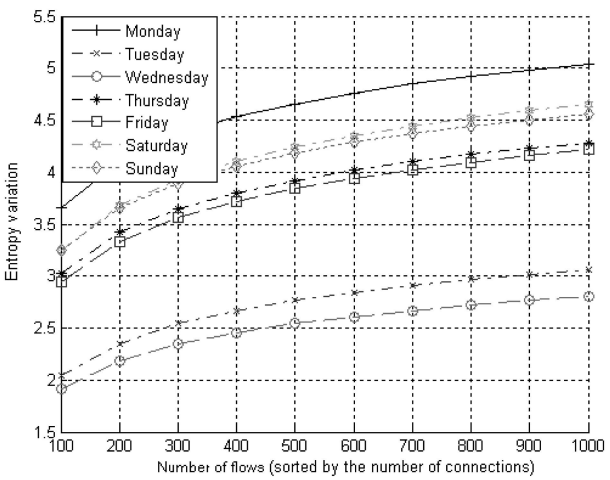


Fig. 9. Entropy variation against number of flows (sorted by the number of connections).

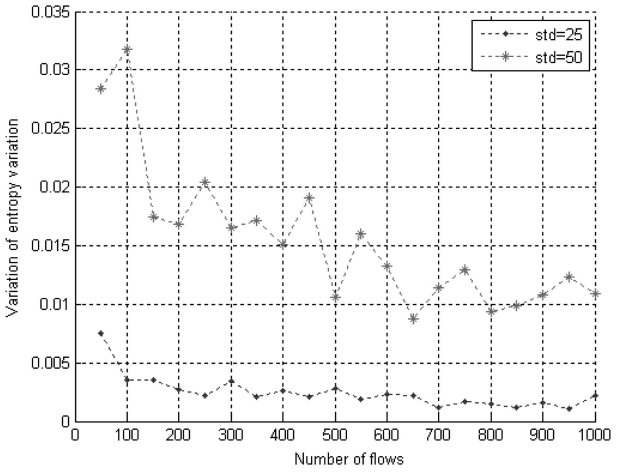


Fig. 10. The standard variation of entropy variation against number of flows with different standard variations.

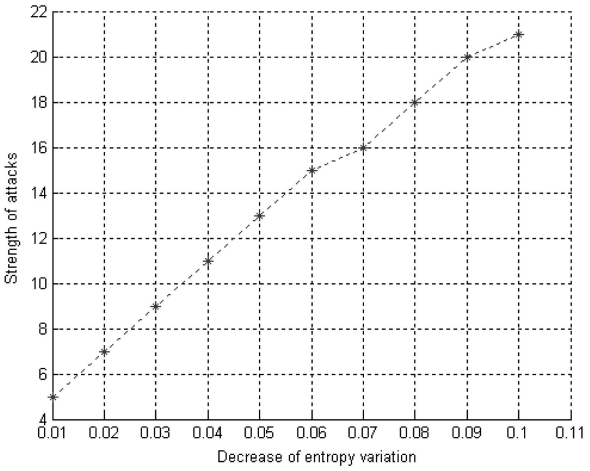


Fig. 12. The relationship between the decrease of the entropy variation and attack strength.

flows for a local router as 1,000, and among them, there is one attack flow. We keep the packet rates of the nonattack flows at the same level, and increase the packet rate of the attack flow, from 1 to 600 times of the nonattack flow packet rate. The results are shown in Fig. 11.

Fig. 11 indicates clearly that the entropy variation drops almost linearly with the increase of attack strength.

Furthermore, in order to have a direct presentation about the relationship between the decrease of entropy variation and the increase of attack strength, we transformed the results of Fig. 11 into Fig. 12.

In Fig. 10, we have learned that the standard variation of entropy variation of nonattack flows is about 0.015, and Fig. 12 indicates that the decrease of entropy variation is 0.02 when the attack strength is seven times of the normal flow, in other words, we can only discriminate DDoS attack flows when its attack strength is about seven times of the normal flow; and we cannot further our traceback procedure once the attack strength is not strong, say, less than seven times of the legitimate flows.

The attach strength is the critical element for our traceback mechanism, and our strategy is effective once the attack strength is obvious from legitimate flows, for example, at least seven times stronger. Therefore, the proposed traceback method can deal with the majority of DDoS attacks, e.g., packet flooding attacks. We have to point out that our method cannot traceback to zombies whose attack strength is less than seven times the legitimate flows, as this may cause false negative. Another accuracy issue for our method is the false positive, for example, flash crowds will create false positive if we start the traceback procedure at the victim site.

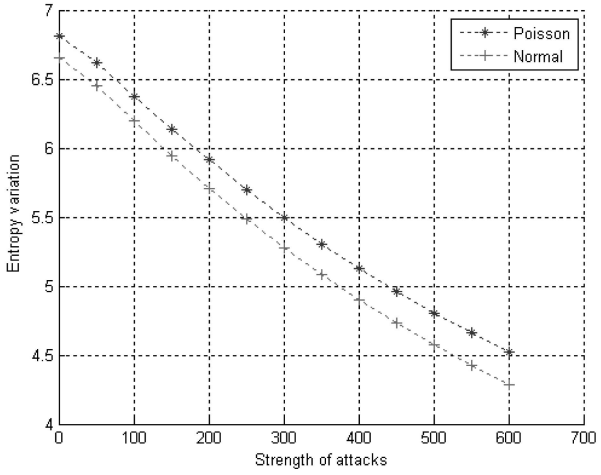


Fig. 11. The changes of entropy variation against strength of attacks.

We now consider the entire attack tree and investigate the convergence of entropy variation when a DDoS attack is ongoing. Assume that there are 1,024 zombies in the following simulations, and they are distributed uniformly in terms of hops from the victim. In these simulations, we ignore the hops with no zombies, and the most far away zombies are 10 hops away from the victim, namely, each hop has around 100 zombies. We examine the convergence of the entropy variation with different attack tree structures, e.g., two-branch tree and three-branch tree. For each simulation, we examine three cases, 100 flows, 500 flows, and 1,000 flows, respectively. The results for the binary tree and three-branch tree cases are shown in Figs. 13 and 14, respectively.

From Figs. 13 and 14, we find that the entropy variation converges when the attack flows are aggregated to the victim, namely, the entropy variation of a router decreases when the router is getting closer to the victim. For example, the entropy variation at 10 hops away from the victim is much higher than that of router which is 2 hops away from the victim. In general, the variation entropy decreases when the attack flows get closer to the victim. This confirms the conclusion of Theorem 2 of Section 4. Moreover, the entropy variation converges faster in the three-branch attack tree compared with that of the binary attack tree because the attack strength is higher and concentrated in the first case. These two simulations also demonstrate that the change of entropy variation is related to the attack strength: higher attack packet rates result in greater drop of entropy variation, in other words, it is easier for our proposed IP traceback strategy to complete its tasks.

Based on these two convergence simulations, we can conclude that if a node in the attack tree possesses more

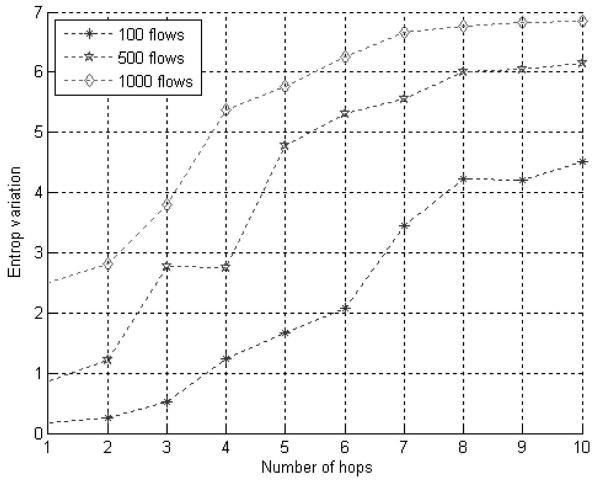


Fig. 13. The convergence of the entropy variation on a binary attack tree.

child nodes, then the entropy variation converges faster. As a result, it is easier for us to conduct the traceback procedure.

In order to estimate the overall traceback time, we assume the same number of zombies ($N = 1{,}024$) and aforementioned parameters. The zombies are evenly distributed in 10 groups in terms of hops away from the victim. Each of the groups can be anywhere from the victim: from 1 hop away to 31 hops away. In the worst case, the zombies are located evenly at the far end on the attack tree, in other words, the 10 groups of zombies are located from 21 to 30 hops away from the victim. We simulated each case for the binary attack tree and the three-branch attack tree, respectively, and the results are shown in Fig. 15.

Fig. 15 shows that the total traceback time is about 25 seconds in the worst case (the most far away zombies are 30 hops away from the victim), and it is less than 20 seconds if the most far away zombies are 23 hops away from the victim. In [23], it was reported that their traceback time is about 20 seconds for a single attack source with maximum 23 hops away from the victim. Based on [23], if the number of hops

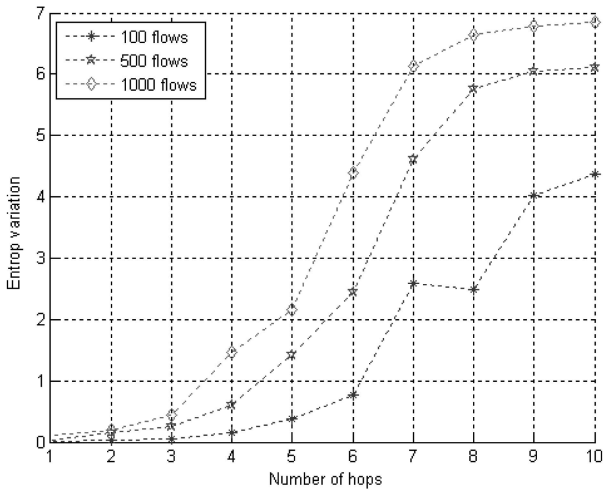


Fig. 14. The convergence of the entropy variation on a three-branch attack tree.

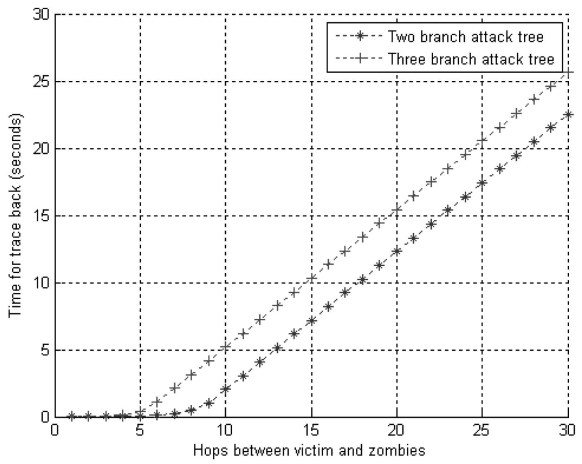


Fig. 15. The traceback time for DDoS attacks.

between two Internet ends is 15, then the general traceback time is around 10 seconds for the binary attack tree, and less than 7 seconds for three-branch attack tree for our traceback method. This simulation demonstrates that our method is better than the previous traceback method in terms of overall traceback time. Moreover, it is shown in [34] that the average duration for DDoS attacks is 5-10 minutes, so that our method can traceback to the most far away zombie effectively before it disappears from the attacking scene.

7 Summary and Future Work

In this paper, we proposed an effective and efficient IP traceback scheme against DDoS attacks based on entropy variations. It is a fundamentally different traceback mechanism from the currently adopted packet marking strategies. Many of the available work on IP traceback depend on packet marking, either probabilistic packet marking or deterministic packet marking. Because of the vulnerability of the Internet, the packet marking mechanism suffers a number of serious drawbacks: lack of scalability; vulnerability to packet pollution from hackers and extraordinary challenge on storage space at victims or intermediate routers. On the other hand, the proposed method needs no marking on packets, and therefore, avoids the inherent shortcomings of packet marking mechanisms. It employs the features that are out of the control of hackers to conduct IP traceback. We observe and store short-term information of flow entropy variations at routers. Once a DDoS attack has been identified by the victim via detection algorithms, the victim then initiates the pushback tracing procedure. The traceback algorithm first identifies its upstream routers where the attack flows came from, and then submits the traceback requests to the related upstream routers. This procedure continues until the most far away zombies are identified or when it reaches the discrimination limitation of DDoS attack flows. Extensive experiments and simulations have been conducted, and the results demonstrate that the proposed mechanism works very well in terms of effectiveness and efficiency. Compared with previous works, the proposed strategy can traceback fast in larger scale attack networks. It can traceback to the most far away zombies

within 25 seconds in the worst case under the condition of thousands of zombies. Moreover, the proposed model can work as an independent software module with current routing software. This makes it a feasible and easy to be implemented solution for the current Internet.

Future work could be carried out in the following promising directions:

1. The metric for DDoS attack flows could be further explored. The proposed method deals with the packet flooding type of attacks perfectly. However, for the attacks with small number attack packet rates, e.g., if the attack strength is less than seven times of the strength of nonattack flows, then the current metric cannot discriminate it. Therefore, a metric of finer granularity is required to deal with such situations.
2. Location estimation of attackers with partial information. When the attack strength is less than seven times of the normal flow packet rate, the proposed method cannot succeed at the moment. However, we can detect the attack with the information that we have accumulated so far using traditional methods, e.g., the hidden Markov chain model, or recently developed tools, e.g., the network tomography. We have a strong interest to explore this for the whole attack diagram.
3. Differentiation of the DDoS attacks and flash crowds. In this paper, we did not consider this issue—the proposed method may treat flash crowd as a DDoS attack, and therefore, resulting in false positive alarms. We have a high interest to explore this issue.

ACKNOWLEDGMENTS

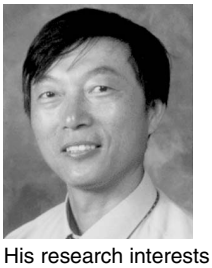
The authors would like to express their thanks to the anonymous reviews for their insightful comments and suggestions. The work was supported in part by grants from The Australia Research Council (Project numbers DP0773264 and DP1095498), Research Grants Council of the Hong Kong SAR, China, General Research Fund (GRF) (No. CityU 114609), and CityU Applied R & D Funding (ARD-(Ctr) No. 9681001).

REFERENCES

[1] "IP Flow-Based Technology," ArborNetworks, http://www.arbornetworks.com, 2010.
[2] C. Patrikakis, M. Masikos, and O. Zouraraki, "Distributed Denial of Service Attacks," *The Internet Protocol J.*, vol. 7, no. 4, pp. 13-35, 2004.
[3] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys*, vol. 39, no. 1, p. 3, 2007.
[4] Y. Kim et al., "PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks," *IEEE Trans. Dependable and Secure Computing*, vol. 3, no. 2, pp. 141-155, Apr.-June 2006.
[5] H. Wang, C. Jin, and K.G. Shin, "Defense against Spoofed IP Traffic Using Hop-Count Filtering," *IEEE/ACM Trans. Networking*, vol. 15, no. 1, pp. 40-53, Feb. 2007.
[6] Y. Chen and K. Hwang, "Collaborative Detection and Filtering of Shrew DDoS Attacks Using Spectral Analysis," *J. Parallel and Distributed Computing*, vol. 66, pp. 1137-1151, 2006.
[7] K. Lu et al., "Robust and Efficient Detection of DDoS Attacks for Large-Scale Internet," *Computer Networks*, vol. 51, no. 9, pp. 5036-5056, 2007.
[8] R.R. Kompella, S. Singh, and G. Varghese, "On Scalable Attack Detection in the Network," *IEEE/ACM Trans. Networking*, vol. 15, no. 1, pp. 14-25, Feb. 2007.
[9] P.E. Ayres et al., "ALPi: A DDoS Defense System for High-Speed Networks," *IEEE J. Selected Areas Comm.*, vol. 24, no. 10, pp. 1864-1876, Oct. 2006.
[10] R. Chen, J. Park, and R. Marchany, "A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 5, pp. 577-588, May 2007.
[11] A. Yaar, A. Perrig, and D. Song, "StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense," *IEEE J. Selected Areas Comm.*, vol. 24, no. 10, pp. 1853-1863, Oct. 2006.
[12] A. Bremler-Bar and H. Levy, "Spoofing Prevention Method," *Proc. IEEE INFOCOM*, pp. 536- 547, 2005.
[13] J. Xu and W. Lee, "Sustaining Availability of Web Services under Distributed Denial of Services Attacks," *IEEE Trans. Computers*, vol. 52, no. 2, pp. 195-208, Feb. 2003.
[14] W. Feng, E. Kaiser, and A. Luu, "Design and Implementation of Network Puzzles," *Proc. IEEE INFOCOM*, pp. 2372-2382, 2005.
[15] X. Yang, D. Wetherall, and T. Anderson, "A DoS-Limiting Network Architecture," *Proc. ACM SIGCOMM*, pp. 241-252, 2005.
[16] Z. Duan, X. Yuan, and J. Chandrashekar, "Controlling IP Spoofing through Interdomain Packet Filters," *IEEE Trans. Dependable and Secure Computing*, vol. 5, no. 1, pp. 22-36, Jan.-Mar. 2007.
[17] F. Soldo, A. Markopoulou, and K. Argyraki, "Optimal Filtering of Source Address Prefixes: Models and Algorithms," *Proc. IEEE INFOCOM*, 2009.
[18] A. El-Atawy et al., "Adaptive Early Packet Filtering for Protecting Firewalls against DoS Attacks," *Proc. IEEE INFOCOM*, 2009.
[19] T. Baba and S. Matsuda, "Tracing Network Attacks to Their Sources," *IEEE Internet Computing*, vol. 6, no. 2, pp. 20-26, Mar. 2002.
[20] A. Belenky and N. Ansari, "On IP Traceback," *IEEE Comm. Magazine*, pp. 142-153, July 2003.
[21] B. Al-Duwairi and M. Govindarasu, "Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback," *IEEE Trans. Parallel and Distributed Systems*, vol. 17, no. 5, pp. 403-418, May 2006.
[22] M.T. Goodrich, "Probabilistic Packet Marking for Large-Scale IP Traceback," *IEEE/ACM Trans. Networking*, vol. 16, no. 1, pp. 15-24, Feb. 2008.
[23] T.K.T. Law, J.C.S. Lui, and D.K.Y. Yau, "You Can Run, But You Can't Hide: An Effective Statistical Methodology to Traceback DDoS Attackers," *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 9, pp. 799-813, Sept. 2005.
[24] S. Savage, "Network Support for IP Traceback," *IEEE/ACM Trans. Networking*, vol. 9, no. 3, pp. 226-237, June 2001.
[25] A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," *IEEE Comm. Letters*, vol. 7, no. 4, pp. 162-164, Apr. 2003.
[26] D. Dean, M. Franlin, and A. Stubblefield, "An Algebraic Approach to IP Traceback," *ACM Trans. Information and System Security*, vol. 5, no. 2, pp. 119-137, May 2006.
[27] G. Jin and J. Yang, "Deterministic Packet Marking Based on Redundant Decomposition for IP Traceback," *IEEE Comm. Letters*, vol. 10, no. 3, pp. 204-206, Mar. 2006.
[28] Y. Xiang, W. Zhou, and M. Guo, "Flexible Deterministic Packet Marking: An IP Traceback System to Find the Real Source of Attacks," *IEEE Trans. Parallel and Distributed Systems*, vol. 20, no. 4, pp. 567-580, Apr. 2009.
[29] C. Gong and K. Sarac, "A More Practical Approach for Single-Packet IP Traceback Using Packet Logging and Marking," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 10, pp. 1310-1324, Oct. 2008.
[30] K. Park and H. Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack," *Proc. IEEE INFOCOM*, 2001.
[31] S. Yu and W. Zhou, "Entropy-Based Collaborative Detection of DDoS Attacks on Community Networks," *Proc. Sixth Ann. IEEE Int'l Conf. Pervasive Computing and Comm.*, pp. 566-571, 2008.

[32] S. Yu, W. Zhou, and R. Doss, "Information Theory Based Detection against Network Behavior Mimicking DDoS Attacks," *IEEE Comm. Letters,* vol. 12, no. 4, pp. 318-321, Apr. 2008.
[33] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* Wiley-Interscience, 2007.
[34] D. Moore et al., "Inferring Internet Denial-of-Service Activity," *ACM Trans. Computer Systems,* vol. 24, no. 2, pp. 115-139, May 2006.
[35] "Lincoln Laboratory Scenario (DDoS) 1.0," MIT, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/2000/LLS_DDOS_1.0.html, 2010.
[36] J. Mirkovic et al., "Accurately Measuring Denial of Service in Simulation and Testbed Experiments," *IEEE Trans. Dependable and Secure Computing,* vol. 6, no. 2, pp. 81-95, Apr.-June 2009.
[37] J. Mirkovic et al., "Testing a Collabotative DDoS Defense in a Red/Blue Team Exercise," *IEEE Trans. Computers,* vol. 57, no. 8, pp. 1098-1112, Aug. 2008.
[38] H. Aljifri, "IP Traceback: A New Denial-of-Service Deterrent?" *IEEE Security & Privacy,* vol. 1, no. 3, pp. 24-31, May/June 2003.
[39] Z. Gao and N. Ansari, "Tracing Cyber Attacks from the Practical Perspective," *IEEE Comm. Letters,* vol. 43, no. 5, pp. 123-131, May 2005.
[40] A. Yaar, A. Perrig, and D. Song, "FIT: Fast Internet Traceback," *Proc. IEEE INFOCOM,* pp. 1395-1406, 2005.
[41] A.C. Snoeren et al., "Hash-Based IP Traceback," *Proc. ACM SIGCOMM,* 2001.
[42] A.C. Snoeren et al., "Single-Packet IP Traceback," *IEEE/ACM Trans. Networking,* vol. 10, no. 6, pp. 721-734, Dec. 2002.
[43] M. Sung et al., "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Information-Theoretic Foundation," *IEEE/ACM Trans. Networking,* vol. 16, no. 6, pp. 1253-1266, Dec. 2008.
[44] http://www.deakin.edu.au/noc, 2010.

Shui Yu (M'05) received the BEng and MEng degrees from the University of Electronic Science and Technology of China in 1993 and 1999, respectively, and the PhD degree from Deakin University, Victoria, Australia, in 2004. He is currently a lecturer in the School of Information Technology, Deakin University, Melbourne, Australia. His research interests include network security, network theory, and information theory. He is a member of the IEEE.



Wanlei Zhou received the BEng and MEng degrees from Harbin Institute of Technology, China, in 1982 and 1984, respectively; the PhD degree from the Australian National University, Canberra, in 1991; and the DSc degree from Deakin University, Victoria, Australia, in 2002. He is currently the chair professor of information technology and the head of School of Information Technology, Faculty of Science and Technology, Deakin University, Melbourne, Australia. His research interests include distributed and parallel systems, network security, mobile computing, bioinformatics, 8and e-learning. He has published more than 200 papers in refereed international journals and refereed international conferences proceedings. Since 1997, he has been involved in more than 50 international conferences as general chair, steering committee chair, PC chair, session chair, publication chair, and PC member. He is a senior member of the IEEE.

Robin Doss received the bachelor's degree in electronics and communication engineering from the University of Madras, India, in 1999, and the master's degree in engineering and the PhD degree from the Royal Melbourne Institute of Technology (RMIT), Australia, in 2000 and 2004, respectively. He has held professional appointments with Ericsson Australia, RMIT University, and IBM Research, Switzerland. He joined Deakin University, Melbourne, Australia, in 2003, and currently, is a lecturer in computing. His current research interests include data management and routing in mobile ad hoc and wireless sensor networks, network security, and next generation networks. He is a member of the IEEE.

Weijia Jia (SM'08) received the BSc and MSc degrees from the Central South University, Changsha, China, and the MASc and PhD degrees from the Polytechnic Faculty of Mons, Belgium, all in computer science. He is currently a professor of computer science, Department of Computer Science, City University of Hong Kong (CityU), Kowloon. In 2006 and 2008, he was awarded HK $22 millions from the Innovation and Technology Fund of the Hong Kong SAR Government for two projects with the intentions of designing and implementing cyber cross-platform secure communications to integrate the Internet with 3G, WiFi, WiMAX, ad hoc, sensor, and PSTN networks for real-time multimedia communications such as VoIP. In these fields, he has authored more than 300 publications in international journals, books/chapters, and refereed international conference proceedings. He has coauthored (with W. Zhou) the book *Distributed Network Systems* (Springer, 2006), which contains extensive research materials and implementation examples. His research interests include wireless communication and networks, distributed systems, and multicast and anycast QoS routing protocols for the Internet. He has served as the editor and guest editor of several international journals and has been a program committee (PC) chair, a PC member, and a keynote speaker for various prestigious international conferences. He was a recipient of the Best Paper Award at a prestigious IEEE conference. He has been listed in the *Marquis Who's Who (VIP) in the World* (2000-2008). He is a member of the ACM and a senior member of the IEEE.