Ad Hoc Networks 8 (2010) 193–213

Contents lists available at ScienceDirect

Ad Hoc Networks

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

CATCH: A protocol framework for cross-layer attacker traceback in mobile multi-hop networks

Yongjin Kim [a,*], Ahmed Helmy [b]

[a] *Qualcomm, 5775 Morehouse Drive, San Diego, CA, USA*
[b] *Computer and Information Science and Engineering (CISE) Department, University of Florida, Gainesville, FL, USA*

ARTICLE INFO

Article history:
Received 30 July 2008
Received in revised form 29 March 2009
Accepted 13 July 2009
Available online 17 July 2009

Keywords:
Attacker traceback
Mobile multi-hop networks
Denial of Service (DoS) attack
Distributed DoS (DDoS) attack

ABSTRACT

Flooding-type Denial-of-Service (*DoS*) and Distributed DoS (*DDoS*) attacks can cause serious problems in mobile multi-hop networks due to its limited network/host resources. *Attacker traceback* is a promising solution to take a proper countermeasure near attack origins, for forensics and to discourage attackers from launching the attacks. However, attacker traceback in mobile multi-hop networks is a challenging problem. Existing IP traceback schemes developed for the fixed networks cannot be directly applied to mobile multi-hop networks due to the peculiar characteristics of the mobile multi-hop networks (e.g., dynamic/autonomous network topology, limited network/host resources such as memory, bandwidth and battery life). We introduce a protocol framework for attacker traceback, *CATCH*, geared towards mobile multi-hop networks utilizing MAC and network cross-layer approach. We also perform systematic risk analysis on mobile multi-hop networks. Based on the risk analysis, we extend *CATCH* for a mobile attacker traceback scheme. We show that *CATCH* successfully tracks down attacker under diverse mobile multi-hop network environment with low communication, computation, and memory overhead. We provide comprehensive evaluation of our proposed protocols through extensive simulations.

1. Introduction

Mobile multi-hop networks include Mobile Ad-hoc NETworks (MANET), wireless mesh networks, and wireless sensor networks, among others. Various types of mobile multi-hop networks have been under active research recently due to its numerous promising applications and practical deployment is near. However, in general, security issues are not properly addressed in the design of such networks. DoS/DDoS attack can cause serious problems in mobile multi-hop networks since (1) it is easy to perform attack using existing tools, and (2) in general, mobile multi-hop networks are severely limited in network resources (e.g., bandwidth) and host resources (e.g., battery, memory, etc).

Different types of DoS/DDoS attacks can be broadly classified into software exploits and flooding-type attacks. In software exploits (e.g., Course of silence attack [8]), attacker sends a few packets or even single packet to exercise specific software bugs within the target's OS or application disabling or harming the victim. On the other hand, in flooding-type attack [7], one or more attackers send incessant streams of packets aimed at overwhelming link bandwidth or computing resources at the victim. We focus on flooding-type DoS/DDoS attack since it cannot be fixed with software debugging. In flooding-type DoS/DDoS attack, an attacker transmits a large number of packets towards a victim with spoofed source address. For instance, in SYN Flood [18], at least 200–500 pps (packet per second) of SYN packets are transmitted to a single victim. DNS amplification attack [16] also attacks victim using a large

* Corresponding author. Tel.: +1 858 740 4505.
E-mail addresses: yongjink@qualcomm.com, v2yjkim@gmail.com (Y. Kim), helmy@cise.ufl.edu (A. Helmy).

1570-8705/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2009.07.002

amount of packets with spoofed DNS server address. In general, we can say that the following are some characteristics of flooding-type DoS/DDoS attacks: (I) Traffic volume is abnormally increased during attack period. (II) Attackers routinely disguise their location using incorrect/spoofed addresses. (III) Such attacks may persist for tens of minutes and in some case for several days.

We define our goal of attacker traceback as the ability to identify the machines that directly generate attack traffic and the network path this traffic subsequently follows [4], or at least identify their neighborhood (e.g., location, neighboring nodes) if not identity. There are several attacker traceback schemes proposed for the Internet such as packet marking, logging, and ICMP traceback [4]. Such traceback schemes were developed for the fixed networks and are not directly applicable to mobile multi-hop networks due to the following particular characteristics of mobile multi-hop networks. (1) In mobile multi-hop networks, there is no fixed infrastructure, gateways or firewalls. Each node works as an autonomous terminal, acting as both host and a router. (2) Each node can move in and out of the network, frequently changing network topology. (3) In general, network bandwidth and battery power are severely limited. (4) It may be difficult to physically secure a mobile node that could be captured, compromised to later rejoin the networks as a Byzantine node.

To perform efficient DoS/DDoS attacker traceback under such a harsh environment in mobile multi-hop networks, we propose an efficient protocol framework, called *CATCH*. The building blocks of our framework consist of (I) abnormality detection, (II) abnormality characterization, (III) abnormality searching, (V) countermeasures. We validate and evaluate our framework through extensive simulation-based analysis and comparison. We utilize cross-layer traffic monitoring and filtering to dramatically improve the success and accuracy of our proposed protocol.

We also systematically analyze mobility-induced risks. One of the most serious obstacles in attacker traceback under mobile multi-hop networks is dynamic topology. Existing attacker traceback schemes cannot be directly used under the presence of node mobility. Node mobility can be classified into two classes; intentional/malicious attacker's mobility and legitimate mobility of intermediate/victim nodes. Intentional/malicious attacker's mobility can cause numerous problems and illusions in traceback. To identify various risks caused by attacker's mobility, we propose *multi-dimensional set-based risk analysis* method. In addition, we analyze how various innocent mobility of intermediate and victim can affect traceback performance.

In sum, we make the following contributions in this paper:

- We provide a complete set of attacker traceback protocol framework for mobile multi-hop networks. In each component of the framework (i.e., abnormality detection, characterization, and searching), we compare various possible sets of schemes and identify the most optimal scheme among those sets.
- We use cross-layer (i.e., network and MAC layer) information to increase traceback efficiency and decrease the associated overhead. We also effectively utilize overhearing capability of the wireless MAC layer, to drastically increase robustness against node compromise and mobility and to reduce false positive and negative rates.
- We propose traceback-assisted countermeasure, which provides an effective defense strategy.
- We propose systematic risk analysis methodology based on multi-dimensional set-based approach. The risk analysis methodology provides an effective way to analyze how attackers can exploit mobility and associated risks. We also propose mobile attacker traceback scheme, systematically analyze how legitimate mobility can affect the traceback performance, and use various mobility models to evaluate the traceback performance.

The rest of the paper is organized as follows. In Section 2, we discuss design requirements for robust attacker traceback in mobile multi-hop networks and provide comparison of existing schemes. In Section 3, we provide an overview of our traceback protocol framework. We describe abnormality detection, characterization, searching, and the overall traceback algorithm in Sections 4–7, respectively. In Section 8, we provide the traceback-assisted countermeasure scheme. In Section 9, we perform systematic risk analysis of sophisticated mobile attacks. Then, we describe how legitimate mobility can affect traceback performance in Section 10 and provide mobile attacker traceback scheme in Section 11. Finally, we conclude our paper in Section 12.

2. Design requirements

To analyze and identify design requirements for traceback protocol in mobile multi-hop networks, we classify the main building blocks of the attacker traceback protocol as follows: (I) information searching and gathering, (II) information storage, and (III) information analysis. Information searching and gathering are the processes to put together or seek clues on the attack traffic. Information storage is the process to store the gathered clue in some storage for analysis. Information analysis is the process to reconstruct the attack path based on the clue obtained through information storing process or real-time data provided by information searching and gathering processes. Based on the classified building blocks, we identify the design requirements (Table 1) for our traceback protocol in mobile multi-hop networks.

2.1. Information gathering

For robust and efficient information searching and gathering in mobile multi-hop networks, we need to satisfy the following protocol requirements: First, the traceback scheme should be robust against route instability due to node mobility and topology change. Second, it may be difficult to physically secure nodes that could be captured, compromised and later rejoin the networks as Byzantine node. Hence, we need robustness against node compromise. Third, in general, mobile multi-hop networks are severely limited in networks resource (i.e., bandwidth). In

Table 1
Design requirements for attacker traceback in mobile multi-hop networks.

Protocol building block	Design requirement
Information gathering	• Robustness against topology change and mobility • Robustness against node collusion • Low communication overhead and energy consumption
Information storage	• Low storage consumption
Information analysis	• Low computational overhead • Low delay

addition, energy conservation is one of major concern. Hence, we need to reduce communication overhead and energy consumption.

Existing attacker traceback schemes (e.g., packet marking [13], iTrace [4]) rely on hop-by-hop traceback. Hence, when one intermediate node moves out or powered down, the traceback process fails. In addition, when several nodes are compromised, the traceback process stops at the compromised nodes. Consequently, a majority-based scheme is required in mobile multi-hop networks, which is robust against multiple node compromises and failures. Huang and Lee [11] provides dynamic topology reconstruction mechanism using TTL and neighbor node information to traceback attacker. However, it still suffers form information storage/analysis problem as will be mentioned in the following.

2.2. Information storage

Clue information, obtained through information searching and gathering processes, needs to be stored for traceback. Information can be stored at the end-host or inside the network. However, in general, nodes in mobile multi-hop network have limited storage space. Hence, it is important to reduce storage requirement.

iTrace [5] or FIT [17] is end-host storage scheme. ICMP or marked packets are stored at the end-host and used for path reconstruction. The logging scheme in [11,13] is a network-storage scheme, where clue information is stored in inside networks. An obvious drawback of these schemes is that large amount of data needs to be stored at either the end-host or inside the network since per-packet information is required. Sy and Bao [15] tries to solve the storage problem by gradual refreshing of memory. Sung et al. [14] also tries to solve storage problem by storing small percentage of packets and using sophisticated scheme to reconstruct the attack path. However, those schemes still suffer form information gathering problem under dynamic topology change. Al-Duwair and Goyindarasu [1] proposes hybrid scheme between packet marking and logging to reduce amount of data to be stored. However, it does not resolve information gathering/analysis issue.

On the other hand, controlled flooding [6] does not require information storage. However, it consumes network bandwidth, which is highly undesirable in resource constrained mobile multi-hop networks.

2.3. Information analysis

Information analysis in existing schemes (e.g., iTrace [5], FIT [17], logging [13]) incurs high processing overhead and delay since it takes per-packet analysis approach. For instance, in iTrace, end host first searches the database, which stores packet information. Then, based on the packet information, end-host should reconstruct the attack path.

3. Overview of the *CATCH* protocol framework

Our *CATCH* traceback protocol consists of the following four components: (1) abnormality detection, (2) abnormality characterization, (3) abnormality searching, and (4) countermeasures.

Abnormality is monitored by all nodes in the network. Each node monitors network and MAC layer activity (e.g., number of packet, busy time in MAC layer). Once abnormality is detected, the information is captured and logged. We introduce several classes of detection technique to accurately detect attack with low overhead. We classify abnormality into two classes: coarse-grained abnormality and fine-grained abnormality. Basically, with coarse-grained abnormality, we trace-back attackers using only packet counters, without using payload-level details. Coarse-grained abnormality monitoring is further divided into the following two classes: Coarse-grained Network Layer Monitoring (C-NLM) and Coarse-grained MAC Layer Monitoring (C-MLM). On the other hand, with fine-grained traceback, we trace-back attackers by analyzing payload-level details. Fine-grained abnormality monitoring is further divided into the following three classes: Fine-grained Network Layer Monitoring (F-NLM), Fine-grained MAC Layer Monitoring (F-MLM) and Fine-grained Cross-layer Monitoring (F-CM) that includes both network layer and MAC layer monitoring. There exists a clear tradeoff between the two above mechanisms. In coarse-grained abnormality-based traceback, computational/storage overhead is minimized by sacrificing payload level analysis for traceback. It is an effective way of traceback in many cases when attack traffic shows obvious abnormality and background traffic is low or moderate, as we shall show. In fine-grained abnormality-based traceback, payload-level information is considered and analyzed to trace back attackers. It requires more computation/storage overhead because we need to store and analyze more detailed information, but it can more accurately trace back attackers. Fine-grained abnormality-based traceback becomes essential in the following cases: (1) In DDoS attacks only reduced abnormality is observed near the edges of the attack route. Hence, we need to differentiate between attack traffic and background traffic accurately. (2) Under high background traffic, abnormality becomes less obvious. We show that we can increase traceback accuracy even under low abnormality and high background traffic by using fine-grained abnormality monitoring by filtering much of the noise traffic (i.e., background traffic). The fine-grained cross-layer information can further filter out background traffic. With the various classes of abnormalities defined above, we perform the following traceback process.

Once abnormality is detected, the abnormality needs to be characterized for traceback. Characterized abnormality at the victim is called the 'attack signature' and abnormality characterized at an intermediate node is called 'candidate attack signature'.

We need to find nodes that observe candidate attack signature which is sufficiently similar to attack signature. By progressively finding nodes that observe similar attack signature from nodes near victim to attack origin, we can find attack route. To provide energy efficiency and robustness against false reports, we use majority voting. Majority voting is performed by multiple nodes that observe or overhear abnormality in a certain region. We also utilize the small world model [9,10] to increase the search efficiency.

After we identify the attack origin(s), we carry out countermeasures (Section 9) to ameliorate the intensity of (or stop) the attack. Existing countermeasures (e.g., packet filtering, rate limiting) suffer from several drawbacks. A packet filtering scheme drops a large volume of legitimate packets. While in a rate limiting scheme it is hard to find the optimal limiting rate. We take a hybrid approach of packet filtering and rate limiting. We use matching level – that we call 'confidence index' – to find a reasonable limiting rate.

We also analyze how mobility affects traceback performance (Sections 10 and 11) and propose a scheme to track down mobile attackers (Section 12). We first systematically analyze how mobility can be exploited by attacker(s). Legitimate mobility of intermediate/victim can also bring about negative impact on traceback performance and is also analyzed. To track down mobile attackers effectively, we utilize spatio-temporal relation of attack signature. In addition, we systematically evaluate how various parameters of the mobility pattern can affect the traceback performance.

4. Abnormality detection

The first component of our framework is the abnormality or attack detection by the victim node and intermediate nodes. Based on this scheme the victim node may trigger a traceback search, and intermediate nodes log information to be used during the search as needed.

4.1. Definition of abnormality

Once flooding-type DoS/DDoS attack is launched, a large volume of traffic is generated towards a victim. The flooding-type attack causes protocol layer (i.e., network layer and MAC layer) abnormality. MAC layer abnormality is observed by neighboring nodes around the attack route by the overhearing capability of MAC layer activity. In this paper, we use 802.11 MAC mechanism, which is widely used as MAC layer for wireless devices. However, note that our scheme can be generally applied to other MAC protocols. We analyze how the flooding-type attack traffic causes abnormality in network and MAC layer as follows:

4.1.1. Increased packets at network layer

Flooding-type DoS/DDoS attack causes abnormally increased packets both in relay nodes of attack traffic and the victim. The increase can be statistically detected and identified as abnormality.

4.1.2. Increased collisions at MAC layer

Increased collision can be inferred by several symptoms. (I) Increased retry count due to lack of ACK or CTS: frame or fragment has a single retry counter associated with it. Frames that are shorter than the RTS threshold have short retry count. Frames that are longer than the threshold are considered long frames and have long retry count. Frame retry counts begin at 0 and are incremented when a frame transmission fails. (II) Large contention window (CW). After each unsuccessful transmission, CW is doubled up to a maximum value $CW_{max} = 2^m * CW_{min}$, where m is the number of attempt. (III) Long lifetime: when the first fragment is transmitted, the lifetime counter is started. When the lifetime limit is reached, the frame is discarded and no attempt is made to transmit any remaining fragments.

4.1.3. Increased busy time at MAC layer

A node monitors the channel to check whether it is idle or not. If it is busy for a certain time interval, it cannot go into backoff stage and should defer. Frequent busy time and consequent transition from backoff state to defer stage are considered as symptom of abnormal traffic.

4.1.4. Increased frames at MAC layer

As attack packets are increased, the number of corresponding data frames and ACK frames are increased. In addition, to access channel, the number of RTS and CTS frames are also increased.

We perform simulations to investigate the abnormal behavior of network and MAC layer under DoS attack, and to address the following question: *what is the best information to use as indicator of the attack signature?* We use *ns-2* for simulation with 50 nodes. The network size is 670 m $\times$ 670 m and DSDV is used for underlying routing protocol. Average distance between attacker and victim is four hops. In Figs. 1 and 2, we varied the number of nodes that generate background traffic from 1 to 25 and measured increased rate. Increased rate is defined as the ratio between abnormal behavior and normal behavior. For instance, when collision count under attack is η and collision count under normal background traffic is η', the increased rate is calculated as η/η'. In the simulation, attack traffic is generated as ten times of normal traffic size. As shown in the Fig. 2, frame count, packet counts and busy time show high increased rate when background traffic is low, and the increased rate decrease as background traffic increases. It is because as background traffic increases the attack traffic does not show drastic abnormality. On the other hand, increased rate in collision is low even when background traffic is low. It is because collision rarely occurs when there exists only attack traffic. The increased rate of collision gradually goes up as background traffic increases and decreases after certain point.

Fig. 3 show relative variance of abnormality information. Relative variance is defined as (*variance of abnormality information*)/(*mean of abnormality information*). Collision rate shows the highest variance. It is because collision

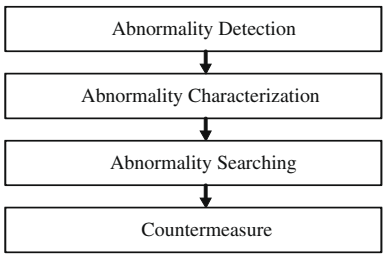


Fig. 1. Attacker traceback protocol framework.

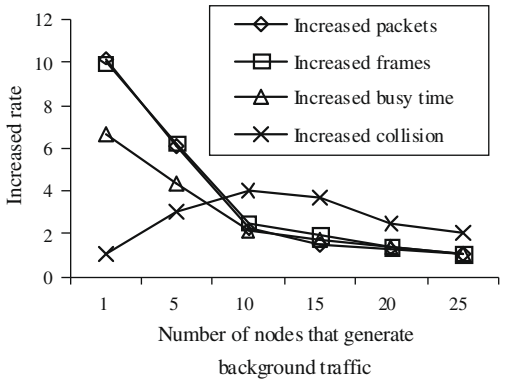


Fig. 2. Network and MAC layer abnormality.

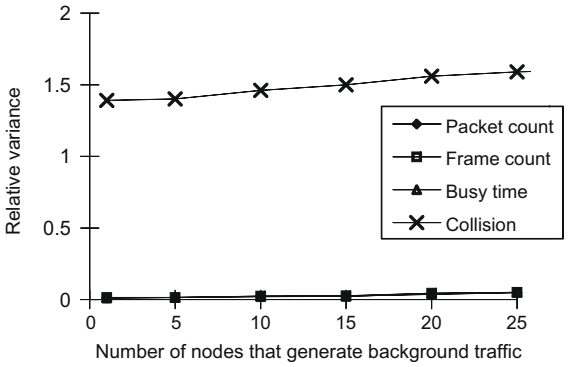


Fig. 3. Variance of network/MAC layer abnormality.

occurrence varies depending on temporal and spatial traffic distribution, which is not desirable as our abnormality information.

We use a two-way contingency table to evaluate the dependency of protocol layer activity on attack traffic. In two-way contingency table, data is classified according to the directions (row and column) of classification, based on two qualitative variables. In our test, the row is the normal/abnormal and the column is attack region/non-attack region. Attack region is the area around the attack path where nodes can observe attack traffic activity. In the contents of table, the number of nodes that observes corresponding protocol layer abnormality is used. Here, we define abnormality as traffic with increased rate greater

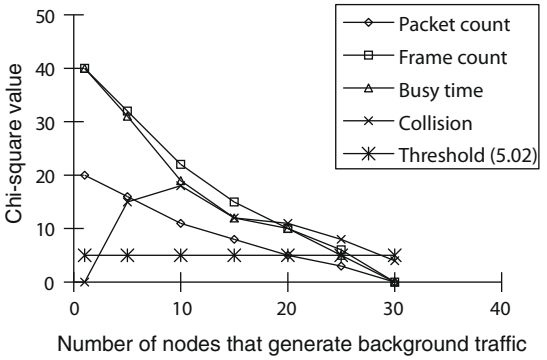


Fig. 4. Relation between attack traffic and protocol layer activity.

than 2. More formal definition of abnormality follows in the next section. Then, we set the following null and alternative hypotheses to test the dependency of the two classifications.

H_0: The two classifications are independent.
H_a: The two classifications are dependent.

$$\text{Test statistic}: \chi^2 = \sum_{j-1}^{c}\sum_{i-1}^{r}\frac{[n_{ij}-\hat{E}(n_{ij})]^2}{\hat{E}(n_{ij})}, \quad (1)$$

where $\hat{E}(n_{ij}) = \frac{n_i n_j}{n}$, n_i is total for row i and n_j is total for column j. The rejection region where we can conclude that two classifications are dependent is as follow.

$$\chi^2 > \chi^2_\alpha \quad (2)$$

where χ^2 is chi-square probability distribution with $(r-1)(c-1)$ degree of freedom and α is the probability of a type I error (a type I error is made if H_0 is rejected when H_0 is true). Intuitively, χ^2 is high when more the percentage of nodes observe abnormality.

In Fig. 4, we show χ^2 value of each abnormality component. The threshold χ^2_α is 5.02 with 97.5% confidence interval. We can infer that if the χ^2 value is above the threshold, there exists dependency (reject the null hypothesis H_0). As shown in the Fig. 4, We can constantly observe high χ^2 value ($\chi^2 > \chi^2_\alpha$) with frame count and busy time information, which means that attack traffic have clear impact (dependency) on the overhearing nodes around the attack route. On the other hand, packet count shows low dependency. It is because packet count is based on network layer information that cannot use overhearing capability. Consequently, we can say that frame count and busy time are better candidates as abnormality information to be robust against node collusion.

Based on all the observation above, *we can conclude that the frame count information is the best candidate as attack signature*. We will use the frame count as our main abnormality indicator. In addition, we will utilize network layer information (i.e., packet-header information) to complement MAC layer information.

Each node monitors protocol layer activity. Once abnormality is detected, the node logs the abnormality information as candidate attack signature. Later, during the search

phase the candidate attack signature is compared with the attack signature which is characterized by a victim. To detect abnormality, we need to define a threshold. If the observed value exceeds the threshold, it is defined as abnormality. Threshold can be set either as fixed value or adaptive value. For fixed threshold, we use the Fractional Deviation from the Mean (FDM), and we use the Pivot method for the adaptive threshold. Both methods will be considered and compared in our study.

4.2. FDM-based detection

Let A_S be the number of frames in a given time slot and A_R be the average number of frames in the long-term reference model. Then the distance of the Fractional Deviation from the Mean (FDM) statistic is given as follows:

$$Dist = \frac{A_S - A_R}{A_R}. \tag{3}$$

The distance, *Dist*, is defined as the abnormality level. If the abnormality level is over the threshold (e.g., 0.5), it is considered suspicious and the (candidate) attack signature is characterized and logged. The obvious advantage of FDM is its simplicity in defining the threshold. However, it does not consider the variance of background traffic to detect abnormality.

4.3. Pivot-based detection

To consider the background traffic variance, we use the pivotal method. We calculate the normal interval with the confidence interval of $100(1-\alpha)\%$ as follows:

$$\bar{x}_n \pm z_{\alpha/2}\left(\frac{\sigma}{\sqrt{n}}\right) \approx \bar{x}_n \pm z_{\alpha/2}\left(\frac{s_n}{\sqrt{n}}\right), \tag{4}$$

where $\bar{x}_n$ is the sample mean of x_i and σ is the standard deviation of x_i. Since the value of σ is unknown, the sample standard deviation s_n is used.

When the new value of x_i is outside the normal range, we define it as abnormality. The abnormal values are excluded from calculating normal range. The advantage of using the Pivotal method is accurate abnormality detection. The computational complexity of Pivot-based method is $O(1)$ in each detection.

In both FDM-based and Pivot-based detection, we can either use simple average or leverage Exponentially Weighted Moving Average (EWMA) to calculate normal profile (i.e., A_R. or $\bar{x}_n$). In EWMA, normal profile at time $n+1$ (i.e., $\bar{x}_{n+1}$) is calculated as follows:

$$\bar{x}_{n+1} = \beta * \bar{x}_n + (1-\beta) * x_{n+1}, \tag{5}$$

where x_{n+1} is abnormality observed at time $n+1$ and $\bar{x}_n$ is normal abnormality profile up to time n. Based on the value of β, we can put more weight on short-term observation or long-term observation. In performance analysis section, we compare performance difference based on selection of β. In addition, to accommodate spatial traffic variation in mobile network, we recommend obtaining new normal profile whenever movement is detected (e.g., using GPS or routing table change). In addition, to accommodate temporal traffic variance, we recommend updating normal profile every hour.

4.4. Coarse-grained vs. fine-grained detection

In coarse-grained detection, abnormality is detected by the aggregate traffic level using total counts, without keeping track of individual flow statistics. The advantage of coarse-grained detection is its computational efficiency. However, the problem of aggregate (or coarse-grained) traffic-based abnormality detection is that it is hard to detect small abnormalities accurately under the presence of large/bursty background traffic, which will prove necessary, especially for DDoS attacks. To address this problem, we define fine-grained abnormality detection, which uses minimal fine-grained cross-layer information (i.e., destination address, previous-hop MAC address).

Fine-grained cross-layer information provides the following advantages: First, we can reduce noise traffic using minimal network-layer (i.e., destination address) information. We call this Fine-grained Network layer Monitoring (F-NLM) scheme compared to Coarse-grained NLM (C-NLM) where only aggregate network layer information is used. In F-NLM, the attack signature is captured based on the traffic destined to each destination (i.e., we make an abnormality table indexed by destination address). We can rely on the destination address since attackers do not spoof destination address to achieve their goal. As shown in Fig. 5, a monitoring node (inside the dotted circle) can remove noise traffic that is destined to non-victim nodes. We call the noise traffic as forward noise.

Secondly, by using minimal information from MAC layer (i.e., previous hop MAC address), we can also drastically reduce noise traffic. As shown in Fig. 5, a monitoring node (inside dotted circle) can remove noise traffic that is not coming from the same previous-hop MAC address as attack traffic. We call the noise traffic as backward noise. We also call it Fine-grained MAC Layer Monitoring (F-MLM) scheme compared to Coarse-grained MLM (C-MLM) where only aggregate MAC layer information is used.

By using fine-grained information of both network and MAC layer (i.e., destination address, previous hop MAC

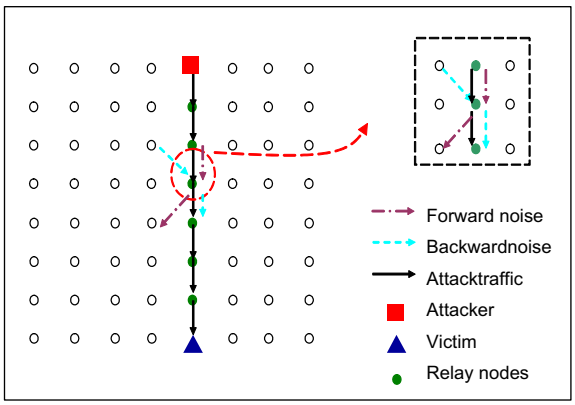


Fig. 5. Illustration of forward/backward noise reduction using cross-layer monitoring.

address), we can drastically reduce noise. We call this Fine-grained Cross-layer Monitoring (F-CM) scheme.

4.5. Performance analysis

In this section, we evaluate the performance of the proposed abnormality detection schemes. We use the following simulation environment. This simulation setting is used throughout this paper, unless otherwise noted. We have performed simulations using *ns-2* and *C* code. Transmission range of each node is set at 150 m. We repeated each simulation 10 times using various random topologies and calculated the average value. We set *NoC* (Number of Contacts) = 6, *R* (vicinity radius) = 3, *r* (contact distance) = 3, *d* (search depth) = 5 for contact selection (refer to Section 7). DSDV is used as underlying routing protocol. Network size is 2750 m × 2750 m. The number of nodes is 1500 and the nodes are static (except in the mobility simulation section). Average node degree is ∼14. Attack traffic is generated from random positions (in some cases clustered or spread as noted).

The performance of abnormality detection depends on the following several factors: (1) Underlying background traffic. (2) Normal profile calculation method. (3) Threshold value to detect abnormality. We varied each of the above factors and investigated its impact on the performance.

We say that the background traffic is "stable" if the variance of the background traffic is between 0% and 10% of the average background traffic. The attack percentage represents the ratio of average attack traffic to the average normal traffic. *FDM* and *MFDM* use fixed normal profile calculation and fixed threshold of 0.5. *MPivot* varies the threshold based on the standard deviation of the normal traffic. Fig. 6 shows the detection success rate under stable background traffic, with the use of the *F-MLM* as abnormality monitoring method. The Pivot and MPivot schemes show 100% success rate consistently. On the other hand, FDM, and MFDM show low detection success rate when attack percentage is low. It is because of the use of fixed thresholds. More specifically, when the attack traffic volume is small, it fails to capture the abnormality because the abnormality goes below the fixed threshold. There is not much performance difference between using average

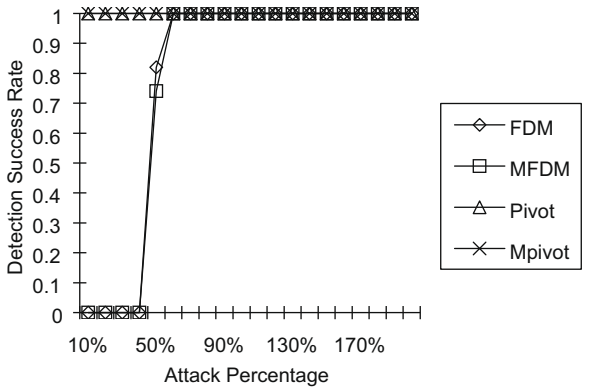


Fig. 6. Detection success rate under stable background traffic with F-MLM.

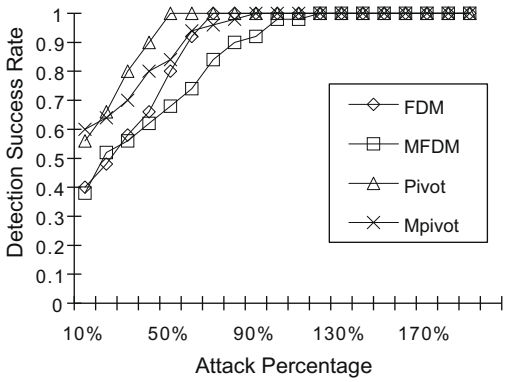


Fig. 7. Detection success rate under fluctuating background traffic.

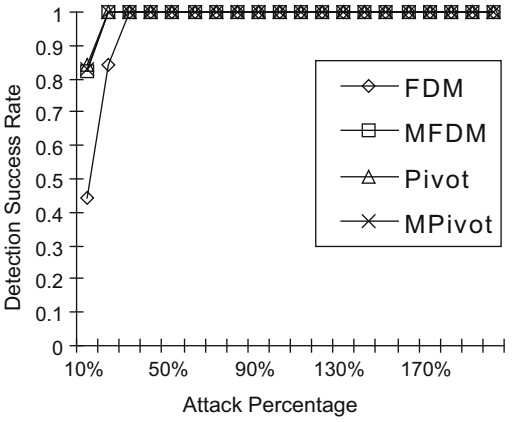


Fig. 8. Improvement in detection success rate with F-CM.

and moving average (i.e., EWMA) when the background traffic is stable.

Fig. 7 shows detection success rate under highly fluctuating background traffic. In fluctuating background traffic, the variance of background traffic is between 0% and 100% of average background traffic. Pivot and MPivot show better performance than FDM and MFDM. The result is similar to Fig. 6. However, EWMA ($\beta = 0.8$) causes approximately 10% lower performance, which is counterintuitive. It means that putting more weight on the recent data reduces detection success rate. After careful investigation, we found that it is because recent short-term bursty traffic leads to high average normal profile, which leads to detection failure with small abnormality.

Fig. 8 shows the detection success rate with F-CM under fluctuating background traffic. As expected, the detection success rate is largely improved by using F-CM since it filters out a lot of noise traffic. More detailed evaluation of F-CM is given in the next section where the results consistently show the superiority of the cross-layer monitoring with fine-grained information.

5. Abnormality characterization

Once abnormality is detected, the abnormality needs to be characterized. We characterize the abnormality as time

Table 2
Abnormality table using cross-layer information.

Destination_addr	Sourse_MAC_addr	Abnormality
1	2	$\xi(1,2)$
1	3	$\xi(1,3)$
.	.	.
.	.	.
.	.	.
.	.	.

series data. That is, attack signature is defined by the sequence of *number of frames* in n. time slots, $(a_1, a_2, \ldots, a_n)$, where $a_i (1 \leqslant i \leqslant n)$ is the number of frames at time slot i. Sampling window, D, is expressed as:

$$D = n \cdot d, \tag{6}$$

where d is the time slot length.

For fine-grained characterization, destination address, and previous hop MAC address are used (Table 2). There is obvious tradeoff between coarse-grained and fine-grained characterization. When coarse-grained characterization is used, space complexity for abnormality logging becomes $O(1)$. However, abnormality matching and subsequent traceback result becomes less accurate. On the other hand, when fine-grained characterization is used, space complexity becomes $(ON * M)$, where N is the number of destination network addresses and M is the number of previous hop source MAC addresses. N can grow to the number of nodes in the network, and M is the average node degree (number of direct neighbors). However, traceback accuracy is improved since we can reduce noise traffic, as we shall show later in this section.

6. Abnormality searching

Once abnormality is characterized, abnormality matching is done between candidate attack signature and attack signature. If the two signatures are closely matching, we can infer the attack route. Following, we examine and compare two techniques for signature matching: 1. traffic pattern matching and 2. KS-fitness test.

6.1. Traffic pattern matching

Traffic Pattern Matching (TPM) is defined in [12]. It uses the correlation coefficient between two signatures at node A and B. When a signature observed at node A is given as $(a_1, a_2, \ldots, a_n)$, and a signature observed at node B is given as $(b_1, b_2, \ldots, b_n)$, the correlation coefficient is obtained as follows:

$$r(A,B) = \frac{1}{nS_AS_B}\sum_{i=1}^{n}(a_i - \overline{A})(b_i - \overline{B}), \tag{7}$$

where

$$S_A = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i - \overline{A})^2}, \tag{8}$$

$$S_B = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(b_i - \overline{B})^2} \tag{9}$$

and $\overline{A}$ and $\overline{B}$ are the averages of $(a_1, a_2, \ldots, a_n)$, and $(b_1, b_2, \ldots, b_n)$, respectively. In case the correlation coefficient $r(A,B)$ is high (greater than 0.7), the signature at node A is said to match the signature at node B. Computational complexity of TPM is $O(n)$, where n is the number of observation time slots.

6.2. KS-fitness test

We use the Kolmogorov–Smirnov (KS) statistic D_n to test the hypothesis that the two attack signature, $F_n(x)$, and $F_0(x)$ are matching. $F_0(x)$ corresponds to the attack signature, which is characterized by a victim and $F_n(x)$ is the candidate attack signature, which is characterized by the intermediate nodes.

$$D_n = \sup_x[|F_n(x) - F_0(x)|], \tag{10}$$

$$\begin{aligned} &H_0 : F_n(x) = F_0(x), \\ &H_a : F_n(x) \neq F_0(x). \end{aligned} \tag{11}$$

We accept H_0 if the distribution function $F_n(x)$ is sufficiently close to $F_0(x)$, i.e., if the value of D_n is sufficiently small. The hypothesis H_0 is rejected if the observed value of D_n is greater than the selected critical value that depends on the desired significance level and sample size. When the H_0 is accepted (sufficiently similar), we can infer that the abnormality is matching, meaning that the attack traffic has traversed the region, where candidate attack signature is observed. Computational complexity of KS-fitness test is *O(nlogn)* and it is not considered significant since we can achieve good matching performance with reasonable small n (observation time slots). We will verify this in the analysis section. In addition, traceback is initiated on-demand only when a device detects attack. Hence, the short-term matching process should not affect scalability.

In case of DDoS attack, there is a subtle problem in using KS-fitness test or TPM. DDoS attack is performed from multiple nodes. Partial attack traffic is merged at the victim or intermediate nodes. Consequently, combination of partial candidate attack signature from multiple incoming interfaces should be compared with the attack signature to find the distributed attack routes. There can be S number of combinations from K candidate $(L \leqslant K)$ partial attack signatures as follows:

$$S = \sum_{i=1}^{K} {}_K C_i. \tag{12}$$

In our scheme, the combination that shows the highest matching level is selected as the branch paths of distributed DDoS attack traffic.

Similar to abnormality detection and characterization, we use coarse-grained and fine-grained matching. By reducing the noise with fine-grained information, we can increase the matching accuracy. The noise can be reduced with MAC address (previous-hop MAC address) and network address (destination) information.

For efficient and robust attacker searching, we use the small world model. Helmy [9] found that path length in wireless networks is drastically reduced by adding a few random links (resembling a small world). These random links need not be totally random, but in fact may be

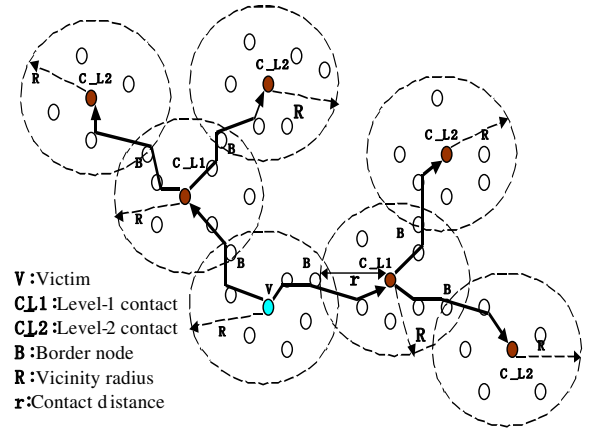


Fig. 9. Small world construction with multi-level contacts. Victim, *v*, selects level-1 contacts. Level-1 contacts select its level-2 contacts, and so on.

confined to small fraction of the network diameter, thus reducing the overhead of creating such network. The random links can be established using contacts [10]. We extend the contact architecture to build a small world in wireless networks, to increase attacker searching efficiency, and to increase robustness against node compromise. Contact nodes are a set of nodes outside the vicinity, which are used as short-cut (random links) to build small world. Following, we describe small world construction scheme.

Each node in the networks keeps track of a number of nodes in its 'vicinity' within R hops away. This defines the vicinity of a node. The vicinity information is obtained through underlying routing protocol. Each node chooses its vicinity independently, and hence no major re-configuration is needed when a node moves or fails. There is no notion of cluster head, and no elections that require consensus among nodes.

On-demand, a victim node selects a set of contacts outside its vicinity. The main purpose of contact nodes is to act as a short cut. Hence, it is important for contacts to have vicinity that does not overlap significantly with that of the victim node, V, or the other contacts of V. The vicinity overlap occurs between the contact's vicinity and the victim's vicinity. To reduce this overlap, victim node attempts to push the request as far out from the victim's vicinity as possible. Let the borders of victim V be B (Fig. 9). V sends a query to the number of contacts (NoC) through its border nodes (denoted by B in Fig. 9). B constructs a topology view up to R hops away using its own vicinity information, and chooses a border in its vicinity that has maximum distance to V. V also selects NoC borders with maximum separation to reduce route overlap. This is done using vicinity information [10].

The above contact selection scheme provides a mechanism to select NoC contacts that have distances up to $R + r$ hops away from V. We call these contacts level-1 contacts. To select farther contacts (contact of contact), this process is further repeated as needed at the level-1 contacts, level-2 contacts and so on, up to a number of levels called maxDepth, D. Detailed evaluation of this general architecture is given in [10].

Our contact selection and search policy have the following important distinctions from [10]: (1) Contacts are randomly selected every time it launches search to prevent the divulgence of contact information to attackers. That is, if contact nodes for a victim are fixed, an attacker may attempt to compromise the fixed contact nodes to disable traceback. To reduce this risk, we select the contacts randomly. (2) The contacts in our protocol perform in-network processing to check whether attack traffic is traversed through vicinity nodes or not. (3) We perform directional search in which the searching process is directed towards only the attacker(s) to reduce communication overhead. Directional search becomes possible through query suppression, in which contacts that do not observe matching abnormality in their vicinity suppress further queries. (4) Our contact selection is independent of any specific routing protocols.

The attack traffic may not have traversed the contacts themselves but may have traversed nodes in their vicinity. To find region through which attack traffic has traversed, we define attack signature energy. Attack signature energy incorporates abnormality matching level, geographic closeness, and *majority* voting factor, as we shall show in our evaluation section. Use of attack signature energy provides robustness against node compromise, accurate attack route selection, and robustness against bursty background traffic. Basically, we choose the region that shows high attack signature energy as attack path. Attack signature energy is divided into three classes: individual signature energy, local signature energy, and global signature energy.

6.2.1. Individual attack signature energy

Each contact gathers individual attack signature energy from nodes in its vicinity. The individual attack signature energy is defined as follows.

$$E_i(\Delta t) = \frac{1}{D_i(\Delta t)}, \tag{13}$$

where $D_i(\Delta t)$ is the inverse of abnormality matching level between candidate attack signature and attack signature. Hence, $D_n(\Delta t)$ becomes small when there is high abnormality matching between attack signature and candidate attack signature during timeframe Δt. Individual signature energy is affected by noise traffic (i.e., background traffic).

The individual attack signature energy concept can be directly applied to DoS attacker traceback. However, in case of DDoS attack, the individual attack signature energy can not be directly applied since multiple partial attack signatures are merged at either victim or intermediate nodes. Hence, the protocol first identifies branch attack signature and applies the individual attack signature energy concept. As was mentioned in Section 6.2, branch attack signature is identified by finding a combination of candidate attack signatures which shows the highest matching level with the attack signature or branch attack signature. For instance, when there are multiple candidate attack signatures, $(\psi_1, \psi_2, \psi_3, \ldots, \psi_N)$ in multiple contact regions, we select k out of N candidate attack signature which shows the highest matching level with χ. Then those

k candidate attack signature become a branch attack signature. Then, traceback process proceeds like DoS attacker traceback case with each branch attack signature.

6.2.2. Local attack signature energy

Given individual attack signature energy, a contact calculates the local attack signature energy as follows:

$$LE(\Delta t) = \frac{E^u_{1/2}(\Delta t)}{\mu_{1/2}}, \tag{14}$$

where $E^u_i(\Delta t)$ is the median of the individual attack signature energy observed by the vicinity nodes of contact u. $\mu_{1/2}$ is the median distance (in hops) between contact and the nodes that observe similar abnormality. The reason that we use median value instead of average is to prevent negative impact of false report from malicious or compromised node on $LE(t\Delta t)$. In addition, $LE(t\Delta t)$ should satisfy the following condition.

$$\alpha = \frac{n}{N} > \delta, \tag{15}$$

where α is the voting factor, N is the total number of vicinity nodes of the contact, and n. is the number of nodes that observe abnormality. When α is extremely low, we can infer that there is high chance of false reporting. The regions around the victim and relay nodes of the attack traffic should show high $LE(t\Delta t)$. The local signature energy is affected by the percentage of nodes observing the signature energy, median distance from contact, and median individual signature energy in the contact region. Intuitively, we can infer the attacker origin or attack traffic route where high local attack signature energy is observed.

If we use only network layer information, α becomes low. It is because only intermediate nodes that relay attack traffic observe abnormality. Consequently, it has weakness against node compromise and mobility. On the other hand, if we use MAC layer information or cross layer information, α becomes high since neighbor of relay nodes can also overhear the abnormality, which drastically increases robustness against node compromise and mobility.

Attacker may try to maliciously use the searching process to launch DoS attack which we are aiming to protect. However, since we are taking directional searching, query suppression, and majority voting, attacker's malicious searching query will be confined in local area. To further protect searching query/response messages, we can consider authentication of the messages. However, the authentication process is out of the scope of this paper.

6.2.3. Global attack signature energy

To systematically analyze how mobility affects traceback performance, we define the global attack signature energy. Global attack signature Energy (GE) is defined as follows:

$$GE(\Delta t) = \sum_{i=1}^{n} E_i(\Delta t), \tag{16}$$

where n is the total number of nodes that observe high abnormality matching around the attack route(s). This metric provides useful information to analyze mobility effects on traceback performance, but is not used in our protocol per se because it is centralized.

6.3. Performance analysis

We evaluate the previous two schemes for signature matching to assess their abnormality detection ability. Abnormality characterization has the following two parameters: (1) unit monitoring window: this is the atomic time window during which abnormality is monitored, and (2) total monitoring time window. This is the aggregation of unit monitoring windows. We use Signature Timeframe (ST) size to denote total number of unit monitoring windows.

Fig. 10 shows the impact of time asynchronization between attack signature and candidate attack signature on matching test with TPM. Asynchronization occurs since attack signatures are monitored from geographically spread locations on the attack route. N represents the degree of time asynchronization. For instance with $N = 0.5\%$, 50% of attack signatures are asynchronized. As time asynchronization degree becomes larger, the matching level becomes lower in TPM (in general, it is considered "closely matching" if correlation coefficient (i.e., matching level) is at or above 0.7).

Unlike TPM-based matching test, KS-fitness test does not show negative impact by time asynchronization as shown in Fig. 11. That is, the distance in KS-fitness test is always below threshold. It is because KS-fitness test uses statistical abnormality distribution instead of time-series abnormality data.

Fig. 12 shows correlation between unit monitoring window and time asynchronization. We used $N = 0.5$ to represent the worst time asynchronization case. L represents the size of unit monitoring window. The negative impact of time asynchronization is reduced when the unit monitoring window is larger. It is because short-term abnormality outlier can be smoothed out with larger unit monitoring window. However, the obvious disadvantage of larger unit monitoring window is its delay in abnormality characterization.

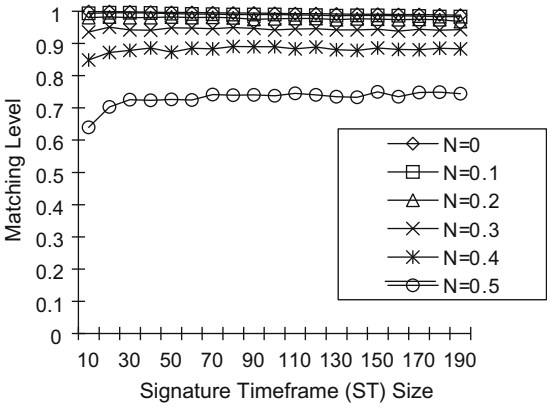


Fig. 10. Impact of time asynchronization on abnormality matching test (with TPM).

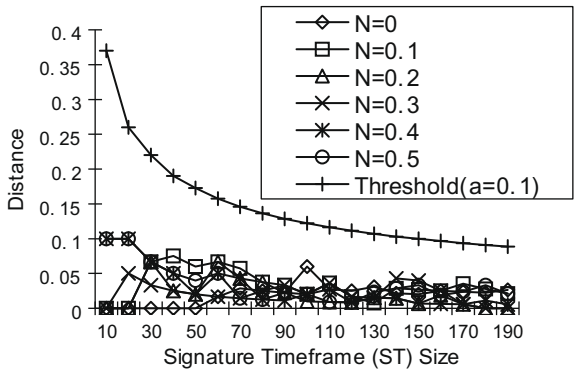


Fig. 11. Impact of time asynchronization on abnormality matching test (with K–S test).

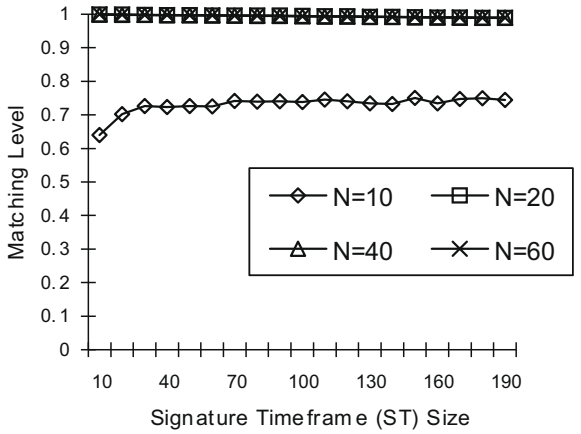


Fig. 12. Impact of unit monitoring window on matching test (with TPM).

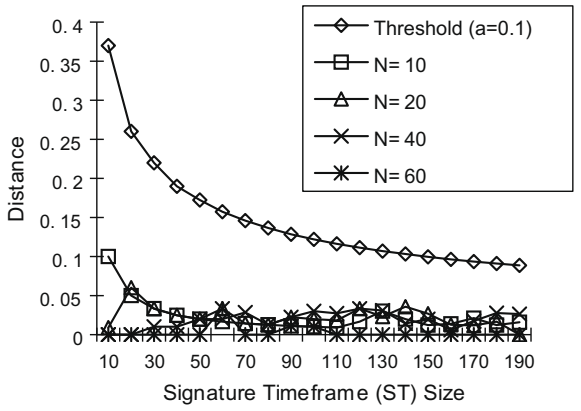


Fig. 13. Impact of unit monitoring window on matching test (with K–S test).

Under the same situation as Fig. 12, KS fitness test in Fig. 13 constantly shows good performance (i.e., below threshold of 0.1), regardless of unit monitoring window size. It is because K–S test uses statistical abnormality distribution instead of time-series data. Hence, delay in abnormality characterization can be avoided.

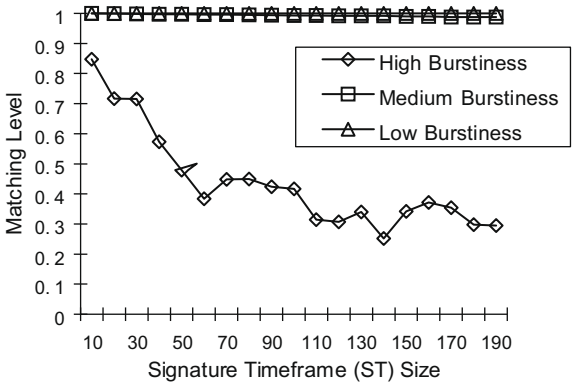


Fig. 14. Impact of background traffic on abnormality matching test (with TPM).

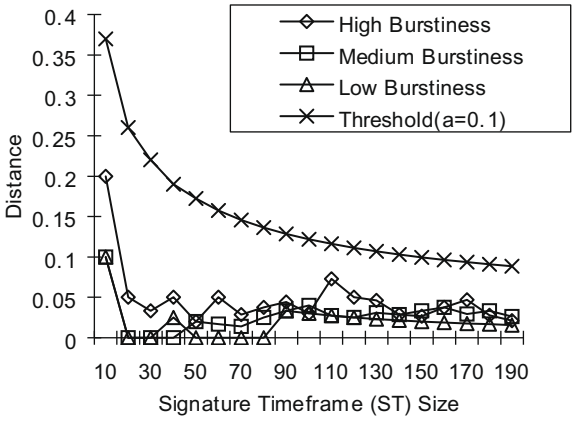


Fig. 15. Impact of background traffic on abnormality matching test (with K–S test).

Fig. 14 shows the impact of various background traffic on the TPM level. Unlike our initial expectation, larger ST size showed lower performance. More specifically, when high bursty traffic exists under large ST size, traffic matching level drastically goes down (0.32 with ST size of 190). It is because there is more chance that the short-term burstiness is included in the attack signature as ST size is increased.

Fig. 15 shows the impact of various background traffic on KS fitness test. We observed high matching level (i.e., below threshold of 0.1) regardless of ST size. It is because abnormality distribution statistics is not affected by small amount of deviation from the reference profile.

Fig. 16 shows the false positive in abnormality matching test with KS test. We compare the distance between (1) attack signature and candidate attack signature with MLM, (2) attack signature and normal traffic with MLM, (3) attack signature and normal traffic with F-MLM, (4) attack signature and normal traffic with F-NLM, and (5) attack signature and normal traffic with F-CM. Originally, we expected that KS-fitness test would show high false

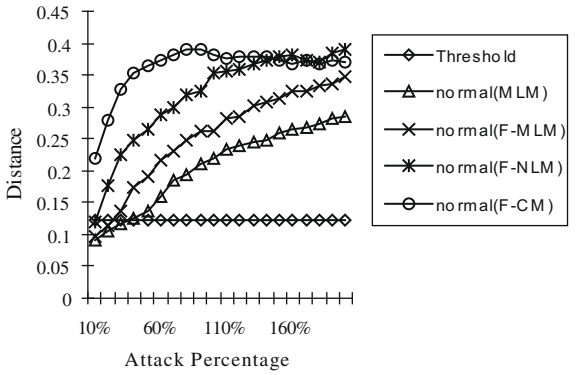


Fig. 16. False positive by normal bursy traffic.

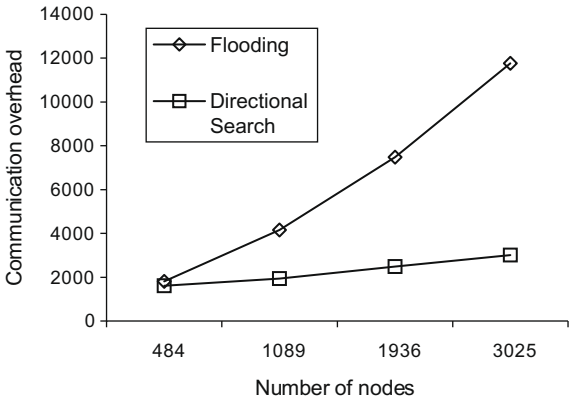


Fig. 17. Comparison of communication overhead in DoS attack.

positive rate since it shows low false negative rate. However, KS-fitness shows low false positive rate, i.e., most of bursty normal traffic is far above threshold.

Based on the analysis, we can conclude that *KS fitness test far outperforms TPM for abnormality matching test*. There is one exceptional case where KS fitness test can show low performance. It when both signatures (i.e., attack signature, and candidate attack signature) show the same statistical characteristics (i.e., same average, variance, etc.) even if its time-series characteristics is different. Under such scenario, we cannot differentiate between attack traffic and normal bursty traffic, which may lead to false positive. However, this is extremely rare case as we can infer from Fig. 16.

We estimates communication overhead (the number of transmitted/received packets) to trace back an attacker in Fig. 17. A victim is located at the center of network and an attacker is located at random positions (17 hops away) on the edge of the network. In flooding, query message with attack signature is flooded to the entire network. Consequently, communication overhead drastically increases as network size increases. On the other hand, our scheme shows very low communication overhead (22% with network size of 3025 nodes) compared to flooding since it deploys directional search and query suppression to reduce communication overhead. Note that the energy saving becomes more significant as the network size increases.

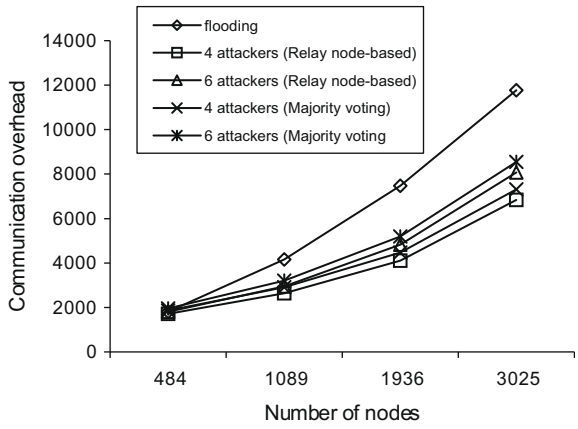


Fig. 18. Comparison of communication overhead in DDoS attack (DS: Directional Search).

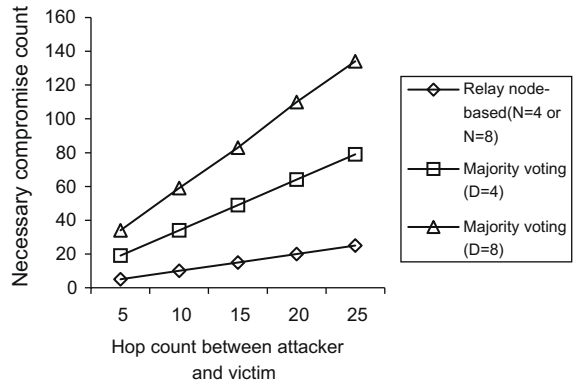


Fig. 19. Robustness against node compromise.

Similar to the DoS case, our scheme incurs low communication overhead in DDoS traceback. In the simulations, a victim is located at the center of the network and attackers are located at random positions at the edges of the network (average of 10 hops away). As the number of attackers increase, communication overhead to search distributed attackers also increases. However, compared with flooding-type query, our scheme incurs lower communication overhead (52% reduction with 4 attackers and 3025 nodes) than flooding as shown in Fig. 18. Note that the number of attacker does not affect flooding-type query overhead since it is flooded to the entire network anyway. The improvement becomes significant as the network size increases.

Fig. 19 shows the number of nodes around the attack route(s) that observes attack signature. To avoid traceback, attacker needs to compromise the observers. General traceback schemes (e.g., PPM, iTrace, logging, etc) rely only on the intermediate nodes that relay the attack traffic for traceback. On the other hand, our scheme (e.g., MLM, F-CM) tracks down attackers by utilizing the information from overhearing nodes around the attack route(s). When node density (D: average number of nodes within transmission range) increases, more nodes (approximately

700% increase with $D = 8$ and 5 hop distance) are able to overhear the attack signature, which drastically increases the robustness against node compromise.

7. Overall attacker traceback

We compare traceback success rate for DoS and DDoS attacker traceback. Traceback success is defined as the event of identifying all the attack origins that generated the attack traffic. We use the Kolmogorov–Smirnov (K–S) fitness test for abnormality matching.

We set up a simulation network size of 2750 m × 2750 m with transmission range of each node set to 150 m and about 20 hops network diameter. We repeat each simulation 10 times in random topology with 1000–3000 static nodes and calculate the average value. We also set *NoC* (Number of Contacts) = 6, *R* (vicinity radius) = 3, *r* (contact distance) = 3, *d* (search depth) = 5 for contact selection. DSDV is used as underlying routing protocol. DoS attacker is 17 hops away from a victim, and DDoS attackers are 10 hops away from a victim.

Fig. 20 shows success rate for DoS attacker traceback with C-MLM and F-MLM. F-MLM shows higher success rate (Average 20% higher than C-MLM). The improvement becomes significant as background traffic is increased. It is because F-MLM uses fine-grained information (i.e., previous-hop MAC address). Fig. 21. shows further improvement (100% success rate) when we use F-CM since we use network layer information to filter out more background noise traffic. Figs. 22 and 23 show success rate for DDoS attacker traceback with 25% of background traffic. C-MLM shows low performance since branch attack traffic has low abnormality. F-MLM shows high success rate when average number of one-hop neighbor is large. However, when average number of one-hop neighbor is small (<4), traceback success goes down in even F-MLM. It is because more noise (i.e., background traffic) is carried over the link where the attack traffic passes.

In Fig. 23, the success rate drastically increases (Average 51%) by using fine-grained network-layer information (i.e., destination address) in addition to fine-grained MAC-layer information. It is because network layer information can filter out most of the noise traffic.

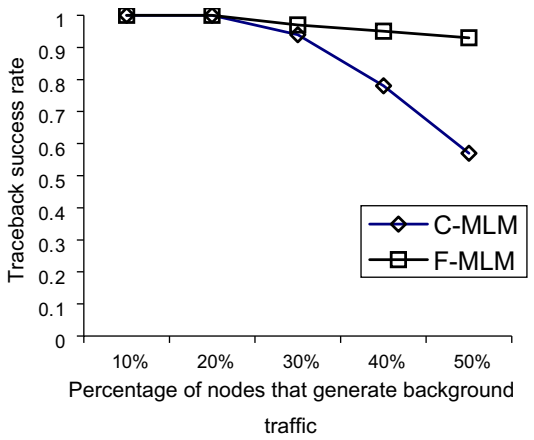


Fig. 20. DoS attacker traceback success rate comparison between C-MLM and F-MLM.

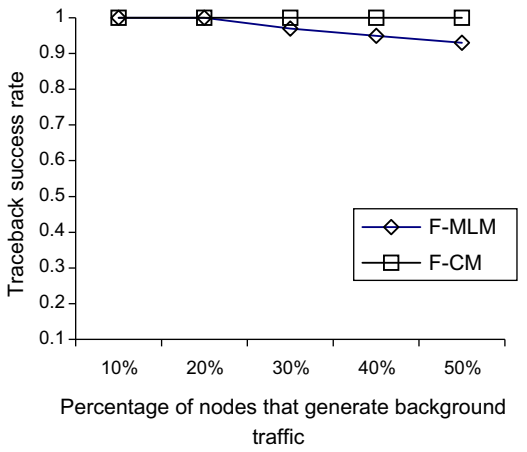


Fig. 21. DoS attacker traceback success rate comparison between F-MLM and F-CM.

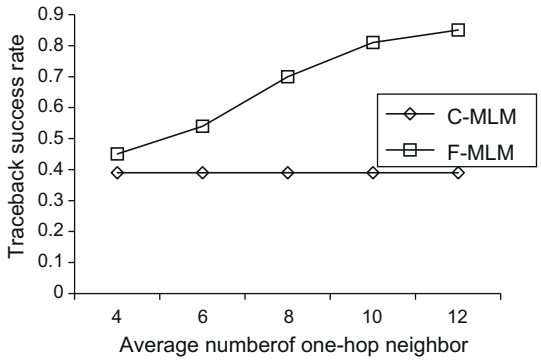


Fig. 22. DDoS Attacker traceback success rate comparison between C-MLM and F-MLM.

8. Traceback-aided countermeasures

Existing countermeasures against DoS/DDoS attack can be broadly classified into two classes: packet filtering, and rate limiting. Current techniques against DoS/DDoS attack have the following drawbacks: (1) The countermeasure is taken at the nodes where the attack is detected. For instance, it is taken at the ingress points of the victim network. However, it is inefficient since the attack traffic exhausts valuable network/host resources of intermediate nodes. (2) Packet filtering is challenging since it is hard to distinguish between bad and good traffic. Legitimate traffic can experience sudden QoS degradation due to packet filtering. (3) In rate limiting, it is hard to know how much the applied rate limit should be to strike a balance between dropping attack traffic and saving legitimate traffic.

We propose a countermeasure which effectively makes use of traceback information. Basically, our countermeasure finds the closest node where the attack occurred

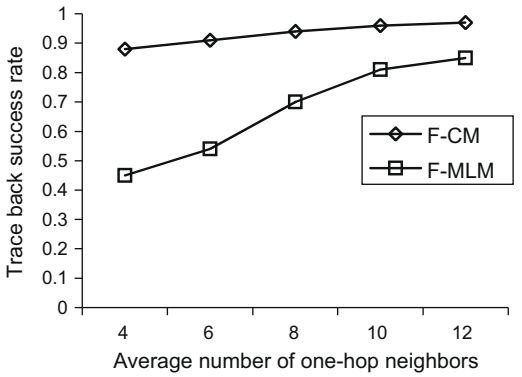


Fig. 23. DDoS attacker traceback success rate comparison between F-MLM and F-CM.

and takes countermeasure based on abnormality matching level. We also use cross-layer information (i.e., destination address, previous MAC address), to increase countermeasure efficiency. More specifically, using cross-layer information, we reduce negative impact on legitimate traffic and increase packet-dropping efficiency on attack traffic. Basically, our scheme can be considered as a hybrid scheme between packet filtering and rate limiting. That is, when abnormality matching level is high, we apply packet filtering. On the other hand, when abnormality matching is medium/low level, we apply rate limiting. To determine a reasonable rate limiting level under medium/low matching level, we use a Confidence Index (*CI*). CI is the inverse of normalized matching level as follows:

$$CI = \frac{1}{norm(D_n)}. \tag{17}$$

The rate limiting level (P is determined through the following equation:

$$P = MaxP \cdot \frac{CI - MinCIThresh}{MaxCIThresh - MinCIThresh} \tag{18}$$

MaxP, *MinCIThresh*, and *MinCIThresh* are parameters selected based on the design policy (see Fig. 24). For instance, high *MaxP*(>0.7) shall be selected to sharply increase the rate limiting effect between *MinCIThresh* and *MaxCIThresh*. When *CI* is very high it reduces to packet filtering since it implies that there is no background traffic. On the other hand, when *CI* is medium/low, it becomes rate limiting based on *CI* level to reduce negative impact on the legitimate traffic. The advantage of using *CI*-based rate limiting over fixed rate limiting is multifold: (1) When CI is low, only small amount of traffic (both attack and legitimate packet) are dropped. Even if we cannot drop much attack traffic in such case, it does not decrease countermeasure efficiency since there exist only small amounts of attack traffic (perhaps part of DDoS attack). On the other hand, when CI is high, many packets are dropped. Even if more legitimate packets are also dropped, its negative impact is not significant since, there exists only small amount of legitimate traffic. We shall investigate our insight further through simulations later in this section.

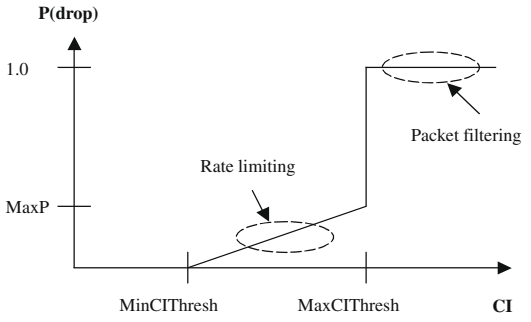


Fig. 24. Hybrid countermeasure.

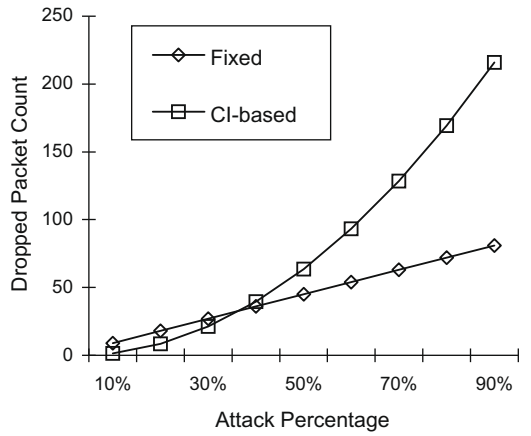


Fig. 25. Dropping efficiency of attack traffic.

To further alleviate QoS degradation of legitimate traffic under the countermeasure, we use cross-layer information. That is, traffic is classified into several classes based on the fine-grained information. When one class (e.g., previous MAC address X, and destination network address Y) of traffic is identified as highly matching candidate attack traffic, we apply rate limiting to the class based on the CI value.

To measure countermeasure efficiency, we define the *LPP* as follows.

$$LPP = (Survived\_legitimate\_traffic) * (Dropped\_attack\_traffic). \tag{19}$$

A high *LPP* value indicates that a large amount of attack packets are blocked and small amount of legitimate packets are dropped. On the other hand, a low *LPP* value indicates that a large amount of legitimate packets are dropped and small amount of attack packets are blocked.

We evaluate the performance of our traceback-assisted countermeasure scheme. We measure the dropped packet count (Fig. 25), survived legitimate packet count (Fig. 26), and LPP (Eq. (19)). We compare our scheme with fixed-rate (i.e.,0.5) packet filtering countermeasure scheme, in which packets are dropped with a given rate under DoS/DDoS attack. As shown in Fig. 25, our scheme shows drastic increase in the attack packet dropping as attack percentage increases (400% higher than fixed scheme). It is because abnormality matching level (i.e., confidence index, CI)

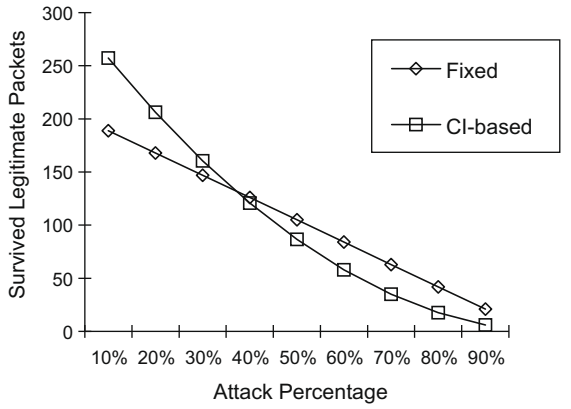


Fig. 26. Comparison of legitimate traffic survival rate.

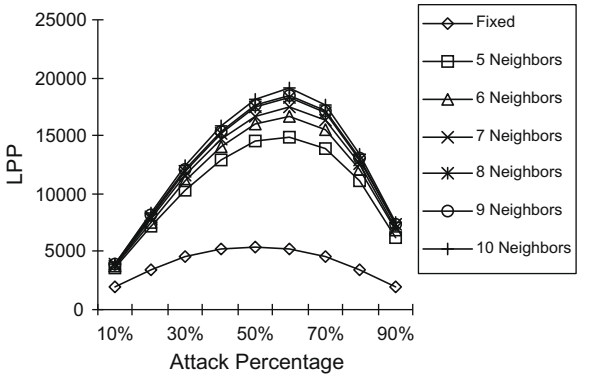


Fig. 27. LPP improvement.

increases as attack percentage increases. Consequently, more attack packets can be dropped. Fig. 26 shows the number of survived legitimate packets. When attack percentage is low, more legitimate packets survive because matching level is low. On the other hand, only a small number of legitimate packets survive when attack percentage is high, since more packets are dropped due to high abnormality matching level. However, the negative impact is not significant (1.4% less survival rate than fixed scheme at 90% attack percentage) because there is only small amount of legitimate traffic when abnormality matching level is high. *Overall, we can say that the positive impact far exceeds the negative impact with our scheme.*

In Fig. 27, we measured LPP with fine-grained information (i.e., F-CM) with varying number of one-hop neighbors. LPP shows drastic increase with CI-based countermeasure scheme. More specifically, LPP increases approximately 110% with CI-based countermeasure compared to the scheme without using CI information (i.e., fixed scheme). It is because we can drop attack traffic more aggressively when there is more attack traffic.

9. Systematic RISK analysis of mobile attacks

Mobility poses several challenges for attacker traceback. First, the attacker(s) can maliciously exploit mobility to avoid traceback, and increase attack efficiency. Second, mobility of intermediate and victim nodes can degrade traceback performance even without malicious intention. To systematically analyze how mobility can affect traceback performance, we take a *multi-dimensional approach*. In this approach, mobile multi-hop network domains are classified into multiple dimensions: (1) temporal transition domain, (2) spatial transition domain, (3) address domain, (4) area domain, and (5) the node coordination domain. These domains were selected after a careful process. We show that the combination of each domain attributes can identify a class of attack scenarios with a unique risk in mobile multi-hop networks.

9.1. Mobile multi-hop network domains

9.1.1. Temporal transition domain (T-domain)

The T domain defines the temporal relation among attack traffic observed at different location. T domain consists of three attributes: temporal continuity (T_c), temporal discontinuity (T_d), and temporal randomness (T_r) as described in Fig. 28. *T0*, *T1*, and *T2* are the time slots during which the attack is observed. In Fig. 28a, attack signatures $\xi 1, \xi 2$, and $\xi 3$ are observed at $T0, T1$, and $T2$ time slots continuously (Temporal continuity). On the other hand, temporal discontinuity is observed in Fig. 29b. The attack signatures ($\xi 1$, and $\xi 2$) are observed at discontinuous time slots at $T0$ and $T2$.

9.1.2. Spatial transition domain (S-domain)

S-domain defines spatial relation among attack occurrence. *S*-domain has three attributes: (1) spatial continuity (S_c) (2) spatial discontinuity (S_d) (3) spatial randomness (S_r). For instance, in mobile DoS attack, the attack signature is observed in a spatially continuous manner (Fig. 29a). In general, spatial discontinuity is observed in DDoS attack as in Fig. 29b. Note that DDoS attack can also show spatial continuity. In that case, we can distinguish between DDoS attack and mobile DoS attack using temporal relation (i.e., *T*-domain).

9.1.3. Address domain (AD-domain)

An attacker can perform malicious attack by spoofing, sometimes using multiple addresses. Some possible attack scenarios include (1) single random address (AD_{sr}) (2) multiple random addresses (AD_{mr}) (3) targeted single address

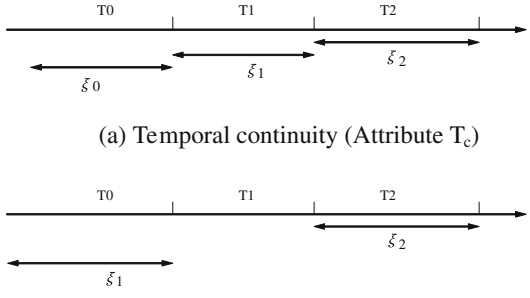


Fig. 28. Three attributes of the temporal *T*-domain.

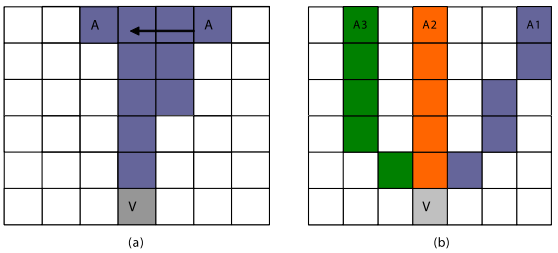


Fig. 29. (a) Spatial continuity of mobile DoS attack. Attacker, A, is moving from right to left attacking victim V. (b) Spatial discontinuity of DDoS attack. Attackers, $A1$, $A2$, and $A3$ are launching attacks towards victim V. Each cell logically corresponds to a contact vicinity.

(AD_{st}), and (4) targeted multiple addresses (AD_{mt}). Unlike fixed networks, the address is not fixed in mobile multi-hop networks. An attacker can easily use false, changing addresses.

9.1.4. Area domain (A-domain)

In mobile multi-hop networks, an attacker can choose its location freely, where an attacker can choose: (1) single random area (A_{sr}) (2) multiple random area (A_{mr}) (3) targeted single area (A_{st}) (4) targeted multiple areas (A_{mt}).

9.1.5. Node coordination domain (N-domain)

Coordination of multiple nodes can lead to serious confusion to a victim. Coordination maybe: (1) temporal coordination of compromised nodes (N_t), or (2) spatial coordination of compromised nodes (N_s).

9.2. Mobile attack identification

Using various combinations of domain assignments we can identify numerous attack scenarios and assess their risks. We define the combinational set as the *set* of *minimal necessary entities* in mobile network domains/attributes to perform certain attack. We identify some of the attack scenarios in this section. Note that these are five example attacks to show the usefulness of the combinational-set based approach and not exhaustive set of all attack scenarios.

9.2.1. Mobility misuse (MM) attack

MM is the simplest form of attack exploiting mobility. In the MM attack, attacker sends attack traffic continuously to a victim. To avoid traceback, the attacker constantly changes its location. The MM attack has the following domain setting:

$$MM\ Domain\ Setting = \{T_c, S_c, A_{mr}\}.$$

As a result of the MM attack, a victim can be confused between DDoS attack and mobile attack. Without considering attacker's mobility, existing traceback schemes will infer that attack traffic is coming from distributed locations, which leads to false positive in distributed nodes.

9.2.2. Mobility and address misuse (MAM) attack

In MAM attack, an attacker sends attack traffic continuously to a victim. To avoid traceback, attacker changes not only its location but also it address.

$$MAM\ Domain\ Setting = \{T_c, S_c, AD_{mr}, A_{mr}\}.$$

Similar to MM attack, a victim will be confused between DDoS attack and mobile attack. In addition, since the attacker changes its address, some preventive techniques such as firewalls or filtering become useless.

9.2.3. False mobility generation (FMG) attack

In FMG attack, the attackers intentionally generate false mobility. That is, the attack is performed from multiple nodes with spatial and temporal continuity.

$$FMG\ Domain\ Setting = \{T_d, S_c, N_s, N_t\}$$

A traceback mechanism capable of detecting mobile attack (e.g., MM attack) may be misled by FMG attack. That is, even if the attack is launched from distributed nodes, a victim might conclude that attacker is moving and performing MM attack.

9.2.4. Distributed blinking (DB) attack

The drawback of FMG attack is that the continuity of the attack is detectable. To overcome this, DB attack can be performed by an attacker. In DB attack, the attacker compromises multiple nodes and performs the attack from distributed random nodes at random times. Attack is launched with spatial/temporal transition randomness.

$$DB\ Domain\ Setting = \{T_r, S_r, N_s, N_t\}.$$

From victim's point of view, attack traffic comes from random location for short period of time. However, bulk attack traffic comes continuously to a victim since multiple nodes are compromised.

9.2.5. Disabling targeted area (DTA) attack

In DTA attack, the attacker is aware of the countermeasure after traceback process. The attacker intentionally generates attack traffic near targeted area where attacker wants to disable or harm networking through expected countermeasure (e.g., packet filtering, or rate limiting). Attacker launches attack at targeted area.

$$DTA\ Domain\ Setting = \{T_c, A_{st}\}.$$

Once the traceback mechanism identifies the attack origin(s), countermeasures are taken near those origin(s). However, since the attacker may intentionally choose the attack origin, legitimate traffic may also be dropped by the countermeasure.

10. Impact of legitimate mobility on traceback

Legitimate mobility of nodes can affect traceback performance. The negative impact of legitimate node mobility occurs due to the following factors:

- Reduction of witness nodesIntermediate nodes that observe abnormality (i.e., witness nodes) can move away from the attack route(s). The problem can become worse when we rely only on intermediate nodes that relayed the attack traffic. Once the relay nodes move away from the attack route, the traceback cannot proceed after that point. By using MAC or cross-layer

monitoring, we can reduce the negative impact by intermediate node mobility since we can use multiple nodes around the attack route that stay and overhear the abnormality.
- Abnormality mismatchingFor traceback, we need to find intermediate nodes that observe similar attack signature (i.e., high signature matching level). However, during attack period, new nodes can move into the attack route, which can reduce signature matching level due to insufficient abnormality monitoring.

To systematically analyze how mobility affects traceback performance, we use the Global signature Energy (GE). GE is defined in Eq. (16), and provides useful information in analyzing how mobility affects traceback performance. We further define Relative attack signature Energy (RE) as follows:

$$RE(\Delta t) = \frac{GE^{dynamic}(\Delta t)}{GE^{static}(\Delta t)}. \quad (20)$$

$GE^{static}(\Delta t)$ represents $GE(\Delta t)$ without mobility, and $GE^{dynamic}(\Delta t)$ represents $GE(\Delta t)$ with mobility during given time duration, Δt. $RE(\Delta t)$ is affected by the mobility model. When $RE(\Delta t)$ is low, attacker traceback becomes difficult since attack signature energy around attack route is reduced due to high node mobility.

In the following sections, we define mobility metrics to systematically analyze how mobility affects traceback performance. Some of metrics are borrowed from our group's earlier work [2].

10.1. Mobility metrics

10.1.1. Directional correlation (DC)

DC is defined as follows:

$$DC(i,j,t) = \frac{\vec{v}_i(t) \cdot \vec{v}_j(t)}{|\vec{v}_i| * |\vec{v}_j|}, \quad (21)$$

where $\vec{v}_i(t)$ and $\vec{v}_j(t)$ are the velocity vectors of nodes i and j at time t. High DC implies, two nodes i and j are moving in the same direction. One the contrary, low DC implies two nodes i and j are moving in opposite directions.

10.1.2. Speed correlation (SC)

SC is defined as follows:

$$SC(i,j,t) = \frac{\min(|\vec{v}_i(t)|, |\vec{v}_j(t)|)}{\max(|\vec{v}_i(t)|, |\vec{v}_j(t)|)}. \quad (22)$$

High SC implies that two nodes i and j are moving with similar speed. One the contrary, low DC implies two nodes i and j are moving at different speed.

10.1.3. Geographic restriction (GR)

Geographic restriction represents the degree of freedom of node movement on a map. More specifically, the degree of freedom represents the number of directions a node can go.

10.1.4. Reference restriction (RR)

Reference restriction represents the degree of freedom of reference point nodes. When all the nodes are going to the same reference point, high RR is observed.

10.2. Mobility dependence

By using the mobility metrics defined above, we can further define mobility dependence among nodes or among groups of nodes as follows:

10.2.1. Mobility dependence between attacker and victim

We define mobility dependence between attacker and victim as follows:

$$MD(a,v,t) = DC(a,v,t) * SC(a,v,t). \quad (23)$$

When attacker (a) and victim (v) have high directional correlation and speed correlation, mobility dependence becomes high.

10.2.2. Mobility dependence among intermediate nodes

Mobility dependence between intermediate nodes is defined as follows:

$$MD(i,i',t) = DC(i,i',t) * SC(i,i',t). \quad (24)$$

When intermediate nodes (i and i') move in a similar direction with similar speed, the mobility dependence becomes high. Mean mobility dependence among nodes N during time duration T is also defined as follows:

$$\overline{MD}(i) = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{t'=1}^{T} MD(i,i',t)}{P}. \quad (25)$$

10.2.3. Mobility dependence among attacker, intermediate, and victim

$$MD(a,v,i,t) = MD(a,v,t) * \overline{MD}(i). \quad (26)$$

When attacker, victim and intermediate nodes are moving in a similar direction with similar speed, the dependence becomes high.

We note that mobility can affect traceback performance. Different mobility models have different characteristics (i.e., high/low DC, SC, GR, RR). To analyze the impact of mobility on traceback performance, we perform numerous simulations with the group and freeway mobility models [2] that encompass various mobility characteristics of DC, SC, GR and RR. To focus on the mobility related effects, we do not generate background traffic in these simulations.

10.2.4. Reference point group mobility (RPGM) model

The RPGM model is defined in [2,3]. Each group has a logical center (group leader) that determines the group's motion behavior. Initially, each member of the group is uniformly distributed in the neighborhood of the group leader. Subsequently, at each instant, every node has a speed and direction (angle) that is derived by randomly deviating from that of the group leader. RPGM model can be used in military battlefield communications where the commander and soldiers form a logical group.

Fig. 30 shows $RE(\Delta t)$ (Eq. (20)) of RPGM model with single group (angle deviation of 20). It shows high $RE(\Delta t)$ (0.99 with 10 ST), which implies that high attack signature energy is observed on the attack route even under high mobility. Consequently, negative impact of mobility is negligible in RPGM with single group. It is because RPGM

model with single group has high mobility dependency (i.e., high *MD(a,v,i,t)*) among attacker, victim and intermediate nodes. As Signature Timeframe (ST) increases, $RE(\Delta t)$ is slightly decreased. It is because there is a higher chance that some nodes move out/in from attack route during a longer timeframe. In addition, it shows lower $RE(\Delta t)$ when speed is high. It is because a few intermediate nodes can move out from overhearing range deviating from the reference points (group leader).

Fig. 31 shows RPGM model with single and multiple (i.e., 4) groups with 10 ST. RPGM with 4 group shows lower $RE(\Delta t)$ (Avg. 32% reduction) than single group case. It is because DC and SC among groups are low and RR is loose among groups. In RPGM model, there is no geographic restriction (GR). RPGM model with 4 groups also shows lower relative energy rate when speed is high due to the same reason as single group case.

10.2.5. Freeway model

The freeway model is introduced in [2]. Fig. 32 shows $RE(\Delta t)$ in freeway model when attacker and victim are on the same lane. Relative energy rate shows medium value (≈Average 0.5) and consequently traceback performance is relatively good in freeway model when attacker and

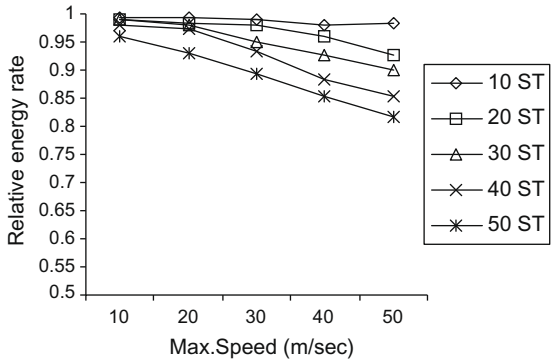


Fig. 30. Relative attack signature energy rate with various Signature Timeframe (ST) size for the RPGM (single group) mobility model.

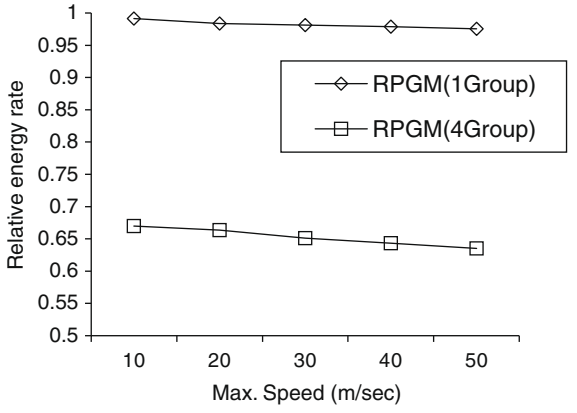


Fig. 31. Relative attack signature energy rate with 1 group RPGM model and 4 group RPGM model.

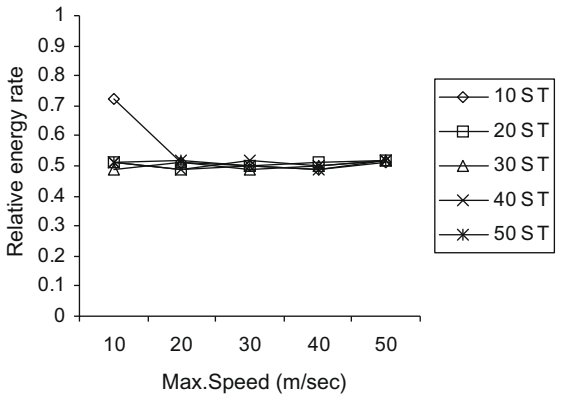


Fig. 32. Relative attack signature energy rate with the freeway model (attacker and victim on the same lane).

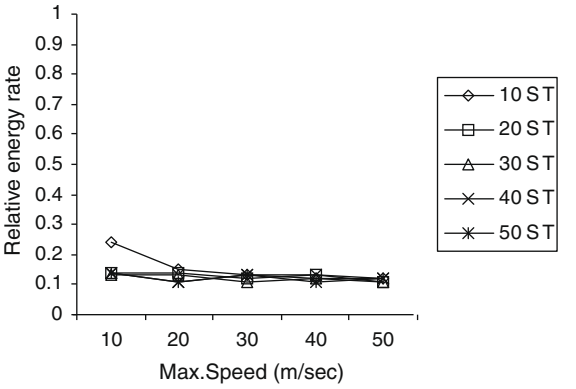


Fig. 33. Relative attack signature energy rate with freeway model (attacker and victim on the opposite lane).

victim exist on the same lane. It is because of high mobility dependency (MD) among attacker, intermediate and victim on the same lane. However, the relative energy rate is not as high as RPGM model since high DC and SC are observed only in the intermediate nodes on the same lane. On the other hand, traceback performance is drastically degraded (≈Average 0.18 of $RE(\Delta t)$ when attacker and victim are on the opposite direction (i.e., low MD) as shown in Fig. 33. High GR in freeway model leads to constant $RE(\Delta t)$ across diverse ST size.

11. Mobile attacker traceback

In this section, we propose a mobile attacker traceback scheme. Our scheme consists of: (1) information gathering, and (2) information fusion processes.

11.1. Information gathering

Information gathering in mobile attacker traceback has the following features: (1) In addition to traceback information used in static attacker traceback, age information is gathered by contact nodes. More specifically, spatio-temporal attack signatures $(\xi, t_S, t_L S)$ are gathered. ξ is

candidate attack signature, S is the relative position of attacker (e.g., 2 hops away from level-1 contact i), t_S is the start time of (or time since) abnormality and t_L is the last (or most recent) time when abnormality is observed. This spatio-temporal attack signature is effectively used to classify attack type (e.g., DDoS attack, mobile attack, etc). (2) All the attack signature information (i.e., spatio-temporal attack signature) from every level of contact needs to be returned to the victim for network-wide analysis.

11.2. Information fusion

Information fusion is the process to correlate and analyze the spatio-temporal signature information obtained through the information gathering process.

To quantitatively represent spatial relation among candidate attack signatures, we define Spatial Relation Factor (SRF) as follows:

$$SRF = \frac{\alpha \cdot P}{\sum_{\eta_{C\_1}=1}^{N_{C\_1}} \sum_{\eta_{C\_2}=1}^{N_{C\_2}} D_S(\eta_{C\_1}, \eta_{C\_2}, \xi_{C\_1}, \xi_{C\_2})}, \quad (27)$$

where

$$\alpha = \frac{n_S}{N_{C\_1} + N_{C\_2}}. \quad (28)$$

N_{c-1} is the total number of vicinity nodes of contact *c_1* and N_{c-2} is the total number of vicinity nodes of contact *c_2*. n_S is the number of nodes that observe attack signature, ξ_{c-1} and ξ_{c-2} in the vicinity of contacts *c_1* and *c_2I*, respectively. η_{c-1} is a vicinity node of contact *c_1*and η_{c-2} is a vicinity node of contact *c_2*.$D_S(\eta_{c-1}, \eta_{c-2}, \xi_{c-1}, \xi_{c-2})$ is the hop count between node η_{c-1} and η_{c-2} that observe the attack signature ξ_{c-1}, and ξ_{c-2}, respectively. The hop count information can be obtained using underlying routing table or through explicit query. $D_S(\eta_{c-1}, \eta_{c-2}, ,\xi_{c-1}, \xi_{c-2}) = 0$ if node η_{c-1} and η_{c-2} do not observe any candidate attack signature. P is the total number of pairs of (η_{c-1}, η_{c-2}), where $D_S(\eta_{c-1}, \eta_{c-2}, \xi_{c-1}, \xi_{c-2}) > 0$. α is majority voting factor. For a high value of α, we can infer that attack is occurring near the central region of *c_1*'s vicinity and *c_2*'s vicinity. It is because more vicinity nodes can overhear the abnormality when attack traffic passes through the central region of the contact's vicinity. When α is small, we can infer that the attack traffic is not passing through the central region of the contact's vicinity or the candidate attack signature report is not reliable (false reporting). When attacker moves from vicinity of *c_1* to vicinity of *c_2*, we can observe small $D_S(\eta_{c-1}, \eta_{c-2}, \xi_{c-1}, \xi_{c-2})$ and high SRF. When *c_1* and *c_2* is not adjacent contacts and η_{c-1} and η_{c-2} are far away, large $D_S(\eta_{c-1}, \eta_{c-2}, \xi)$ is obtained, which leads to low SRF.

We also quantitatively formulate temporal relation of candidate attack signatures as Temporal Relation Factor (TRF).

$$TRF = \frac{\alpha \cdot P}{\sum_{\eta_{C\_1}=1}^{N_{C\_1}} \sum_{\eta_{C\_2}=1}^{N_{C\_2}} D_T(t_L(\eta_{C\_1}), t_S(\eta_{C\_2}), \xi_{C\_1}, \xi_{C\_2})}, \quad (29)$$

where $D_T(t_L(\eta_{c-1}), t_S(\eta_{c-2}), \xi_{c-1}, \xi_{c-2})$ is the time difference between the start time (i.e., $t_S(\eta_{c-2})$) when attack signatures ξ_{c-2} is observed by node η_{c-2} and the last (or most recent) time (i.e., $t_L(\eta_{c-1})$) when the attack signature $\xi_{c-1}$is observed by η_{c-1} where $t_S(\eta_{c-2}) \geqslant t_S(\eta_{c-1})$. Under mobile attack, temporal continuity is observed and TRF becomes large since $D_T(t_L(\eta_{c-1}), t_S(\eta_{c-2}), \xi_{c-1}, \xi_{c-2})$ becomes small.

We use SRF and TRF metrics to infer attack type as follows: (I) When high SRF and high TRF is observed, we can infer that mobile attack has occurred. (II) When high SRF and low TRF are observed, we can infer that attack traffic has been intermittently generated from geographically clustered attackers. (III) When high SRF and negative TRF are observed, we can infer that DDoS attack has occurred from clustered attackers (clustered DDoS attack) (IV) When low SRF and high TRF are observed, we can infer that attack has occurred from geographically spread attackers with temporal continuity. (V) When low SRF and low TRF are observed, we can infer that attack traffic has been generated from geographically spread attackers. (VI) When low SRF and negative TRF are observed, we can infer that DDoS attack from geographically spread attackers has been generated (spread DDoS attack). These results have been validated via extensive simulations.

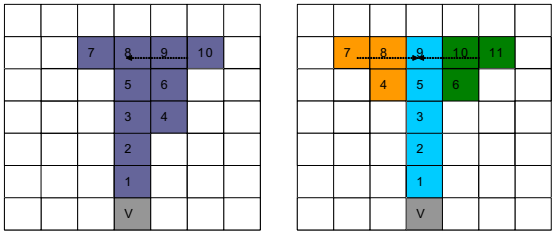
11.3. Examples for mobile attacker traceback

11.3.1. Mobile DoS attacker traceback

Fig. 34a shows the example of mobile DoS attacker traceback using the TRF and SRF metrics. In the figure, attacker moved from region $10 \rightarrow 9 \rightarrow 8 \rightarrow 7$. Attack paths from each attack origin are as follows: $10 \rightarrow 6 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow$ v, $(9 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow$ v), $(8 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow$ v), $(7 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow$ v). A victim will find the first level temporal/spatial relation in regions 3 and 4. In region 3 and region 4, high TRF and high SRF are observed. At this point, we can infer that mobile attack is occurring. Similarly, in region 5 and 6, high TRF and high SRF are observed. Lastly, in region 7, 8, 9, 10, high TRF and SRF are observed, which leads us to conclude that attacker is moving and currently located in the region 7. Vertical or diagonal movement of attacker can be detected similarly.

11.3.2. Mobile DDoS attacker traceback

Basically, mobile DDoS attacker can be detected and tracked down using the same mechanism mentioned above with separate threads for each branch attack route. A difficult problem in mobile DDoS attack occurs when multiple attackers are crossing each other as in Fig. 34b. The crossing mobile DDoS attack can be detected by using TRF and SRF metrics plus attack signature surge detection. For instance, in Fig. 34b, the first attacker is moving $7 \rightarrow 8 \rightarrow 9$ and the second attacker is moving $11 \rightarrow 10 \rightarrow 9$. Attack traffic is merged on the path $9 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 1$. Region 4, and 5 observe high SRF and high TRF. In addition, region 5, and 6 also observe high SRF and high TRF. The relations enable us to infer mobile attack has occurred in (4,5) regions and (5, 6) regions. In addition, region 5 observes attack traffic surge, which allows us to infer the crossing of mobile attack traffic. Similarly, region 7, 8 and 9 observe high SRF and high TRF. In addition, region 11, 10, and 9 observe high SRF and high TRF. Region 9 will also observe attack signature surge. Consequently, relative location of an



(a) MobileDoS attack **(b) CrossingmobileDDoS attack**

Fig. 34. Illustration of mobile attacks.

Procedure at victim *v*

STEP 1: Detect flooding-type DoS/DDoS attack.
STEP 2: Send attack signature query to level-*1* contacts.
STEP 3: If there are multiple signature reports from contacts, calculate SRF and TRF.
STEP 4: If SRF > SRF_thresh and TRF > TRF_thresh between contact c_{1a} and c_{1b}, infer that attacker is moving from region c_{1a} to region c_{1a}.
STEP 5: Check signature surging. If the surging exists, infer that multiple attackers are crossing.
STEP 6: Wait for matching report from higher level contact and perform SRF and TRF calculation.
STEP 7: If there is no more reports after receiving level-*N* contact report, infer the current relative position of attack at level-*i* contact, with the largest age.

Procedure at intermediate contact c_i

STEP 1: Receive attack signature query from contact c_{i-1} or victim.
STEP 2: Gather abnormality information from vicinity nodes.
STEP 3: If abnormality exists, report the attack signature to the victim.
STEP 4: If abnormality is observed and level depth is not reached yet then select contact (c_{i+1}) and send next-level query c_{i+1}. Otherwise suppress query.

Fig. 35. Algorithm for mobile DDoS attack trace-back.

attacker can be inferred from gathered information at contact region 9. The overall algorithm to detect and trace mobile DDoS attack is outlined in Fig. 35.

11.4. Performance analysis for mobile attacks

To evaluate and show the effectiveness of our mobile attacker traceback scheme, we compare the SRF and TRF values in DDoS attacks and mobile DoS attacks. DDoS attacks are performed from six randomly selected nodes. In mobile DoS attacks, the attacker and 5% of intermediate nodes move with random waypoint mobility model ($V_{max} = 2$ m/s, pause time = 2.5 s). Average SRF and TRF values are calculated where mobility is detected. We exclude the regions where α (Eq. (28)) is small (<0.1) since it implies that the nodes that report the attack signature moved out from original attack path. As shown in Table 3, SRF is high (>0.1) in both in mobile attack and clustered DDoS the attack since attack is observed in a close region. SRF shows low value (<0.1) when DDoS attacks are performed from geographically spread locations. TRF can differentiate between mobile attacks and clustered DDoS attacks since DDoS attacks are launched at around the same time regardless of the observation region. Consequently, we can effectively differentiate between DDoS attacks and mobile attacks using combination SRF and TRF metrics.

Table 3
Attack classification using SRF and TRF metrics.

	SRF (m^{-1})	TRF (s^{-1})
Mobile DoS	*0.17*	*28.1*
Clustered DDoS	0.18	5.12×10^{-3}
Spread DDoS	0.032	5.38×10^{-3}

12. Conclusions

In this paper, we propose the *CATCH* framework with a comprehensive set of attacker traceback protocols for mobile multi-hop networks. We use cross-layer (i.e., network and MAC layer) information to increase traceback efficiency and decrease associated overhead. We also effectively utilize overhearing capability of MAC layer, which drastically increases robustness against node compromise and mobility. In addition, it reduces false positive and negative rates. We also proposed traceback-assisted countermeasure, which provides an effective defense strategy utilizing cross-layer information.

We propose a systematic attack risk analysis using a multi-dimensional approach. This risk analysis provides insight to how mobility can be exploited for wireless attacks. We also propose a mobile attacker traceback scheme. We further analyze how mobility, legitimate or otherwise, can affect the traceback performance.

Through extensive simulation-based performance analysis, we showed that our proposed scheme satisfy all the design requirements (Table 1) for attacker traceback in mobile multi-hop networks.

References

[1] B. Al-Duwair, M. Goyindarasu, Novel hybrid schemes employing packet marking and logging for IP traceback, IEEE Transactions on Parallel and Distributed Systems 17 (5) (2006) 403–418. May.
[2] F. Bai, N. Sadagopan, A. Helmy, The IMPORTANT framework for analyzing the impact of mobility on performance of routing for adhoc networks, Ad Hoc Networks Journal 1 (4) (2003).
[3] F. Bai, N. Sadagopan, B. Krishnamachari, A. Helmy, Modeling path duration distributions in MANETs and their impact on routing performance, IEEE Journal on Selected Areas of Communications (JSAC) 22 (7) (2004) 1357–1373. September.
[4] A. Belenky, Nirwan Ansari, On IP traceback, IEEE Communication Magazine (2003). July.
[5] S.M. Bellovin, M. Leech, T. Taylor, ICMP Traceback Messages, October 2001. <draft-ietf-itrace-01.txt>.
[6] H. Burch et al., Tracing anonymous packets to their approximate source, in: Proceedings of 2000 USENIX LISA Conference, December 2000, pp. 319–327.

[7] R.K.C. Chang, Defending against flooding-based distributed denial-of-service attacks: a tutorial, IEEE Communication Magazine (2002). October.
[8] T. Engel, Course of Silence Attack, Chaos Communication Congress, Berlin, 2008. <http://berlin.ccc.de/~tobias/cos/s60-curse-of-silence-advisory.txt>.
[9] A. Helmy, Small world in wireless networks, IEEE Communication Letters (2001).
[10] A. Helmy et al., A contact-based architecture for resource discovery in ad hoc networks, ACM Baltzer MONET Journal (2004).
[11] Yi-an Huang, Wenke Lee, Hotspot-based traceback for mobile ad hoc networks, in: Proceedings of the 4th ACM Workshop on Wireless Security, WiSe '05.
[12] G. Mansfield et al., Towards trapping wily intruders in the large, Computer Networks 34 (2000) 650–670.
[13] A.C. Snoeren et al., Hash-based IP traceback, ACM SIGCOMM (2001).
[14] Minho Sung, Jun Xu, Jun Li, Li Li, Large-scale IP traceback in high-speed internet: practical techniques and information-theoretic foundation, IEEE/ACM Transactions on Networking 26 (6) (2008) 1253–1266. December.
[15] Denh Sy, Lichun Bao, CAPTRA: coordinated packet traceback, in: Proceedings of the 5th International Conference on Information Processing in Sensor Networks, IEEE/ACM IPSN 06.
[16] R. Vaughn, G. Evron, DNS Amplification Attack, 2006. <http://www.isotf.org/news/DNS-Amplification-Attack.pdf>.
[17] A. Yaar, A. Perrig, D. Song, FIT: fast internet traceback, in: Proceedings of IEEE INFOCOM, Miami, USA, March 2005.
[18] CERT Advisory CA-96.21, TCP SYN Flooding and IP Spoofing Attacks, September 24, 1996.

Yongjin Kim received his B.S. degree from Electronics Engineering Department at Yonsei University in South Korean, M.S. degree from Information Science Department at Tohoku university in Japan, and Ph.D. degree from Computer Engineering Department at University of Southern California, USA in 2006. His research interest lies in attacker traceback, risk analysis of various mobile wireless system and software security. He has been also working on QoS provisioning in wireless and wired networks. He is current working at Qualcomm on various security issues for wireless system.

Ahmed Helmy received his Ph.D. in Computer Science (1999), M.S. in Electrical Engineering (EE) (1995) from the University of Southern California (USC), M.S. Eng. Math. (1994) and B.S. in EE (1992) with highest honors from Cairo University, Egypt. He is an Associate Professor and the founder and director of the wireless networking lab at the Computer and Information Science and Engineering (CISE) Department, University of Florida, Gainesville.

From 1999 to 2006, he was an Assistant Professor of Electrical Engineering (EE) at USC. He was also the founder and director of the wireless networking laboratory at USC. He was a key researcher in the network simulator (NS-2) and the protocol independent multicast (PIM-SM) projects in the Information Sciences Institute (ISI), USC. He is leading (or has led) the STRESS, MARS, ACQUIRE and Aware NSF-funded projects. His research interests lie in the areas of network protocol design and analysis for mobile ad hoc and sensor networks, mobility modeling, multicast protocols, IP micro-mobility, and network simulation. In 2002, he received the National Science Foundation (NSF) CAREER Award. In 2000, he received the USC Zumberge Research Award, and in 2002, he received the best paper award from the IEEE/IFIP International Conference on Management of Multimedia and Mobile Networks and Services (MMNS). In 2003, he was the EE nominee for the USC Engineering Jr. Faculty Research Award and a nominee for the Sloan Fellowship. In 2004 and 2005, he got the best merit ranking in the EE-USC faculty. In 2007, he was a winner in the ACM MobiCom SRC research competition, and a finalist in 2008. His projects have been funded by NSF, DARPA, NASA, Cisco, Intel, Nortel, P&W and Silicon Graphics.

He is an Area editor of the Adhoc Networks Journal - Elsevier since 2004. He served as co-chair for the IFIP/IEEE MMNS 2006 and IEEE INFOCOM Global Internet (GI) Workshop 2008, local chair for IEEE ICNP 2008, poster and area chair for ICNP 2009, vice-chair for IEEE ICPADS 2006, and IEEE HiPC 2007. He has been the ACM SIGMOBILE workshop coordination chair (for ACM MobiCom, MobiHoc, MobiSys, and SenSys) since 2006. He served on the program committees for numerous IEEE and ACM conferences in the areas of computer and wireless networks. URL: http://www.cise.ufl.edu/~helmy.