A Hotspot-based Protocol for Attack Traceback in Mobile Ad Hoc Networks

Hungyuan Hsu
The Pennsylvania State University
University Park, PA 16802
rodin_hsu@hotmail.com

Sencun Zhu
The Pennsylvania State University
University Park, PA 16802
szhu@cse.psu.edu

Ali Hurson
Missouri University of Science and Technology
Rolla, MO 65409-0350
hurson@mst.edu

ABSTRACT

Based on the principle of *divide and conquer*, in this paper we propose an efficient traceback protocol for mobile ad hoc networks, The protocol is capable of detecting a hotspot where the attacker resides. It works by dividing the forwarding path of every packet into multiple interweaving fragments and each reachable fragment is individually reconstructed during a traceback process. Through simulations in theoretical mobility models as well as real mobility traces, we show that each traceback of our scheme can attribute to a very small hotspot and the attacker can be accurately identified after a number of traceback operations.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Security, Algorithm, Design

Keywords

Mobile Ad-hoc Networks, Traceback, Path Fragmentation, Packet Marking, Packet Logging

1. INTRODUCTION

IP traceback has been extensively studied in the literature, however, within the scope of mobile ad hoc networks (MANETs), very little research has been attempted [6, 1, 3]. In MANETs, the network nodes forward packets in a self-configuration and self-maintenance purpose without any infrastructure support. While both the scale of a MANET and its data traffic rate are much smaller than its high-speed Internet counterpart, nevertheless, online (or real-time) traceback in MANETs imposes some unique challenges that are uncommon in the Internet. *Node mobility is one key factor.* In a MANET, the network topology is constantly changing because the wireless link is dynamically built up when two mobile nodes move into each other's radio transmission range. The dynamic network topology fundamentally changes the paradigm for attack source traceback. The existing IP traceback schemes do not work in MANETs because almost all of the IP traceback schemes assume a static network topology. *Trust is another key factor.* Unlike the Internet where routers are often trusted, the mobile nodes in MANETs normally should not be trusted. Consequently, a traceback protocol for MANETs itself suffers from malicious attacks.

In this paper, we propose a novel traceback scheme for MANETs, where a traceback is triggered by either a single malicious packet (e.g., a worm) or multiple attack packets (e.g., a DoS attack). Our scheme is capable of detecting a hotspot where a malicious attacker resides. Based on the principle of *divide and conquer*, our scheme works by dividing a forwarding path into multiple smaller interweaving fragments. During a traceback process, those reachable fragments are reconstructed and fragmentation information is gathered. Through simulations with mobility model and real data traces, we show that each traceback that employs our scheme can attribute to a much smaller hotspot where the attack source resides than a conventional logging scheme. In addition, the attack source can be precisely pinpointed with a number of traceback rounds.

2. PRELIMINARIES

2.1 Network Model and Security Assumptions

In a mobile ad hoc network (MANET), nodes form a network on-the-fly and forward packets for one another. Furthermore, they can establish trust through either a PKI, a Trusted Third Party (TTP), or predistributed shared keys. During data forwarding, every packet is authenticated in a hop-by-hop fashion [7]; that is, the link between two neighboring nodes is assumed to be authenticated and a malicious node cannot impersonate any good node.

2.2 Attack Model

We assume the adversary may compromise one or multiple nodes and take full control of the compromised node(s). Since the links are authenticated, an attack source cannot impersonate any normal (benign) node to its downstream node. To hide itself, it will not put its address into the packet source field; instead, it will act as if it was a data

ASIACCS'10 April 13–16, 2010, Beijing, China.

forwarder for the packets while spoofing valid source ids. The attack source may change its location over time to hide itself.

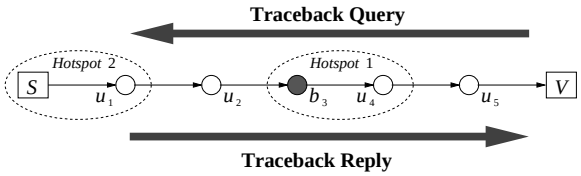


Figure 1: An attack path $\mathcal{A}_M$ of packet M where node S and node b_3 are the compromised nodes. S injects bogus packets, and b_3 conspires to neutralize the traceback attempt.

Without loss of generality, consider an attack path $\mathcal{A}_M$ of packet M in Figure 1, where the source node S and the intermediate node b_3 are compromised and are both at the disposal of the adversary. b_3 may alter packet markings (when a PPM scheme is in place) or drop traceback queries. If a traceback protocol can only trace to u_4/b_3 (e.g., b_3 drops the traceback query or traceback reply), then S becomes invisible to the downstream nodes and the victim. If b_3 does not interfere the traceback process, the protocol may reach u_1/S. In either case, b_3 or S will probably deny its attacking behavior and consequently a dispute may arise between b_3 and u_4 or between S and u_1. Indeed, without digitally signing every packet, nonrepudiation is not possible; thus an attacker node can always accuse a good node.

2.3 Design Goals

We set forth the following design goals. First, we aim to locate the *hotspot* where the compromised node resides. Because a traceback protocol alone cannot precisely identify the attack node, in [1] hotspot-based traceback was first introduced. Given a potential hotspot, it relies on other online or offline analysis/detection measures (e.g., neighbor watching [4]) or human intelligence to identify the attacker [1]. We will also leverage such measures to identify the nodes in a hotspot, given the authenticated links among nodes. Hotspot analysis, however, is generally expensive, so hotspot size should be as small as possible (the minimum is 2).

Second, at least one of the malicious nodes (S or b_3 in the example) should be included. From the attacker's perspective, the exposure of any one of its controlled nodes may have the same impact on the potential of its future attacks. Third, our defense should minimize the number of packets required for a successful traceback. This will not only allow us to detect low rate attacks and catch the attacker as early as possible, but also reduce the bandwidth overhead for launching traceback.

3. ONLINE TRACEBACK IN MANETS

Our design is based on the principle of *divide-and-conquer*. To reduce the number of packets needed to reconstruct the entire path, we propose the ideas of *path-fragmentation* and *fragment interweaving*. With path-fragmentation, each packet during forwarding (probabilistically) divides its entire path into multiple fragments. The fragmentation information is stored in a few intermediate nodes. The fragments formed by multiple packets may be overlapped, building *virtual* interweaved links among en-route nodes. Thus, a broken link due to mobility could be bypassed and farther links may be reconstructed, approaching to the real attack source. Later on, in an online traceback phase, the fragmentation information is gathered, each constructed link is verified, and a hotspot is located.

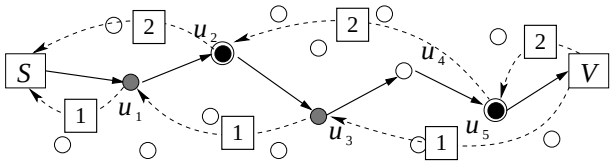


Figure 2: An example of packet marking where u_1 and u_3 are markers in regard to M_1, and u_2 and u_5 are markers in regard to M_2.

For the illustration of the basic idea, let us consider Figure 2. A data packet M_1 is delivered through the path $\mathcal{A} = (S, u_1, \ldots, u_5, V)$, where u_1 and u_3 are the markers in regard to M_1. Whenever an intermediate node decides to mark a packet, it must first log the existing mark in the packet before inscribing its own mark. Consequently, u_1 records the mark from S, u_3 records the mark from u_1, and V records the mark from u_3. As a result, this path is divided into three fragments by u_1 and u_3 and a *reverse virtual link* is created pointing from one marker to its previous one.

Assume another packet, M_2, is sent through the same path $\mathcal{A}$. The markers in regard to M_2 may be different, say u_2 and u_5, due to the probabilistic nature of the marking algorithm. Similarly, this path is divided into three fragments by u_2 and u_5, forming different virtual links. As shown in Figure 2, the fragments of M_1 and M_2 interweave with one another. Two sets of reverse virtual paths are built along $\mathcal{A}$ after M_1 and M_2 are sent through the path $\mathcal{A}$.

The reverse virtual links help to localize the hotspot. In Figure 2, suppose S is the attack source that has forged packet markings for M_1 and M_2. u_1 and u_2 recorded the false packet markings from S, respectively. If later we can trace back to u_1 or u_2, and know the markings stored in u_1 or u_2 are false and these packet markings did not travel more than two or three hops with a high probability (we will show how to achieve these goals shortly), then we may conclude that u_2, u_1 and S are within the hotspot with a high confidence. Compared to a logging scheme, our scheme does not have to physically trace back to u_1 along the forwarding path; the virtual link from u_3 can be used to identify u_1 and its neighbors as a hotspot.

To realize the above idea, we have constructed four building blocks (BBs). During packet forwarding, every node employs a building block *BB-I*, a verifiable, distance-based packet marking scheme to process every packet; meanwhile, it logs the abstract information of the packet into its traceback table. During a traceback process, *BB-II*, a multicast-based forwarding scheme, is employed to forward the traceback query. When a node finds that it was in the attack path, it reports to the victim its virtual links. Then the victim node calls upon the *BB-III*, a path-reconstruction algorithm, to locate the hotspot. Finally, based on the information of the hotspots resulted from multiple tracebacks, building block *BB-IV* is applied to evaluate the reputation

of every node and identify the most suspicious nodes. Due to space limit, next we will only introduce the building block BB-I in more detail, and refer the readers to our full version paper [2] for the other blocks.

Verifiable, Distance-based Packet Marking: Existing marking schemes let every node decide whether or not to mark a packet. Since this is a random process, it is hard for the other nodes to determine whether a node marks a packet according to a probability function or it (as a compromised node) selectively marks the packet to disrupt the potential traceback (e.g., to cover the real attack source). As such, we consider it critical that only a *dynamically* selected set of nodes are allowed to mark in a packet, preventing a malicious node from arbitrarily marking packets. The qualification of a node to mark a packet should also be verifiable. On the other hand, during path fragmentation, it is necessary to control the size of a fragment (i.e., the distance that a packet mark traverses) to increase traceability within each fragment. Based on these observations, below we design a verifiable, distance-based packet marking scheme.

A mark in our scheme has three fields: *marker id*, *distance*, and *authentication code*. Let $H : \mathcal{K} \times D \longrightarrow \mathcal{R}$ be a keyed hash function, denoted as $H_K(\cdot)$. For each packet M destined at node V, an intermediate node u_i calculates a probability r^* as:

$$r^* = \frac{H_{K_{u_iV}}(M|u_i|V|d^*)}{|\mathcal{R}|} \quad (1)$$

where K_{u_iV} is the pairwise key shared between u_i and V. d^* is the distance between u_i and the prior marker recorded in packet M, and $|\mathcal{R}|$ is the cardinality of the range of $H_K(\cdot)$. u_i then compares r^* with a distance-based marking probability $p(d^*)$, which will be discussed shortly. If $r^* \geq p(d^*)$, u_i is illegal to mark M, so it simply increments the distance field d^*. Otherwise, u_i replaces the existing marker id with its own id and resets d^* to 0. Also, $H_{K_{u_iV}}(M|u_i|V|d^*)$ is written into the authentication code field.

This marking scheme not only authenticates the packet mark but also allows V to check if the claimed marker u_i is legitimate to mark M based on V's pairwise key shared with u_i and d^*. The security benefits are two folds: First, without knowing the key K_{u_iV}, an adversary cannot forge an authenticated mark. Second, even if the adversary compromises u_i and gets the key, it still cannot arbitrarily select packets to mark. Note that although an intermediate compromised node may slightly change the packet to make itself a legitimate marker, by doing this it becomes the source of the new packet, subject to the detection of our traceback protocol.

Whether marking a packet or not, an intermediate node logs into its *traceback table* a hextuple containing such information as the packet digest, the prior marker, the distance to the prior marker, an authentication code, the forwarder (i.e., the immediate upstream node), and the destination.

Determining the Marking Probability The marking probability p in an ordinary PPM scheme is a fixed system parameter. From the security point of view, we prefer a packet mark not to travel too far. Nevertheless, from the traceability point of view, we prefer the packet marks to be delivered closer to the destination to increase traceability under node mobility.

To fulfill these two seemingly contradicting demands, we introduce the technique of adjusted probabilistic packet marking. In [5], it is proposed to vary the marking probability from hop to hop according to the position of the marker in the path. As a result, the destination node can get the information of upstream nodes with fewer packets. In our scenario, we desire the majority of packet markings be overwritten by downstream nodes within two or three hops with a high probability so that when a spoofed mark is found in an intermediate node u_i, a hotspot resides within three hops around u_i with a high probability. Moreover, as we will see later, the resulted *virtual links* also facilitate our online traceback to bypass broken links. On the other hand, we still allow the long-distance traversal of packet markings at a low probability so that we may benefit more from marking when many attack packets appear.

Specifically, an example of our marking policy is as follows:

$$p(d^*) = \begin{cases} p_0 + (1-p_0)(1-e^{k\cdot(2-d^*)}), & 1 < d^* < 4 \\ \frac{1}{D^*-d^*+1}, & 4 \leq d^* < D^* \\ 1, & d^* \geq D^* \end{cases} \quad (2)$$

Again, here d^* denotes the distance from the prior marker. In Figure 2, for u_3, $d^* = 2$ regarding M_1. D^* refers to the upper bound of d^*; when seeing D^* in the distance field, the downstream node is required to overwrite the marking. p_0 is the marking probability when the node is two hops ($d^* = 2$) away from the prior marker ($p(2) = p_0$). k is a tunable parameter. For example, when we choose a marking policy where $p_0 = 0.35$, $k = 0.34$, and $D^* = 6$, 70% packet markings are expected to traverse 2 to 3 hops, and the rest of 30% packet markings traverse the distance uniformly distributed among 4 to 6 hops. Hence, in this setting the average traverse distance is 3.25 hops.

4. PERFORMANCE EVALUATION

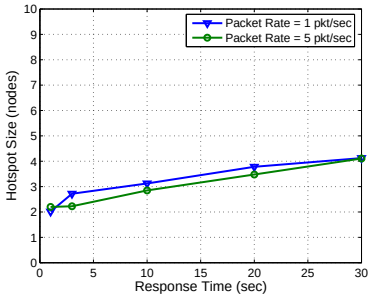
In [2], we provide detailed security analysis on our protocol, covering three protocol phases (i.e.,packet marking and logging, traceback queries forwarding, and traceback report). Here we only briefly report our performance evaluation results.

We use simulations to evaluate the performance of our online traceback scheme. We use the *hotspot size* as the metric to indicate the effectiveness of our traceback protocol; for example, if we can trace to the immediate neighbor of the attack source, then the hotspot size is two. The smaller the hotspot size, the more accurate the traceback result.

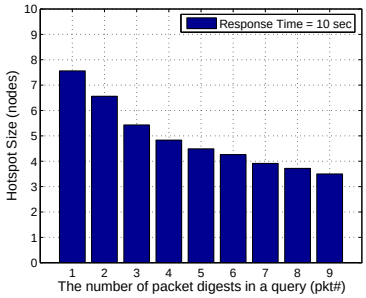
The Impact of Response Time and Packet Rate Figure 3(a) shows that a larger response time leads to a larger hotspot size, which indicates it is more difficult for the victim to trace to the attack source. In addition, we observe that the higher attack packet rate results in a smaller hotspot size because more packets traversing through the attack path leaves more virtual links for traceback. In all the cases, the hotspot size is between $2 \sim 4$.

The Benefit of Using Virtual Links To see how virtual links help traceback, we also let the victim node try 9 traceback queries with the number of packet digests in each of them ranging from 1 to 9. Then the victim node reconstructs the attack path separately for each of the queries.

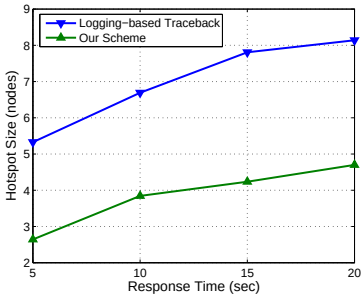
Figure 3(b) shows that with more packet digests embedded in a traceback query (or with more single-packet traceback attempts), the hotspot size decreases. Specifically, if only tracing a single packet, the hotspot size is 7.7, whereas after tracing 9 packets, the hotspot size is reduced to 3.5.



(a) Hotspot size vs. IDS response time



(b) The benefit of virtual links



(c) Comparison with logging-based traceback

Figure 3: Hotspot size as a function of different system parameters

This achievement is attributed to the interweaving virtual links, because more information toward the attack source can be gathered by the victim with more virtual links.

Comparison to the logging scheme Finally we compare our scheme to the logging scheme. It is fair for such comparison because in both schemes the authenticity of the reported links can be verified. We do not directly compare our scheme with a PPM scheme because of the weaker security (links cannot be verified) of PPM under insider attacks.

Figure 3(c) shows a comparison with the logging scheme. Here we can see on average our scheme outperforms a logging scheme at 2-3 hops and the larger the delay, the bigger the difference. Note that more packets in a logging scheme help little because their paths break at the same point.

5. RELATED WORK

At present, only a few traceback schemes have been proposed for MANETs. Thing and Lee [6] conducted simulations to investigate the feasibility of detecting the attack path based on existing IP Traceback techniques. Kim and Helmy [3] proposed a DoS attacker traceback scheme. The major drawback of this scheme is the prohibitive communication cost. Our scheme is on-demand and there is no need to maintain topology information. Huang and Lee [1] developed a hotspot-based traceback protocol for MANETs. In their protocol, every intermediate node records the neighbor list and the time-to-live (TTL) value of each forwarding packet. In the traceback request phase, an investigator broadcasts a query to collect the reports from all of the nodes that have previously forwarded the packet. Based on the network topology reconstructed, a hotspot detection algorithm is run to identify single or multiple approximate locations (hotspots). Although their protocol could result in smaller hotspots than our protocol because of its broadcast nature, it incurs much higher communication overhead than ours because of its network-wide flooding. We will study how these two schemes may compensate each other in our future work.

6. CONCLUSIONS AND FUTURE WORK

Traceback in MANETs is a challenging research issue because of node mobility and the lack of trust among mobile nodes. In this work we presented a new attack source traceback scheme. Our simulation study showed that our scheme could catch the attack source in a small hotspot and it outperforms the logging scheme in general. Our future work will continue to improve the effectiveness of tracing attack sources in mobile environments. We will choose and evaluate other marking probability functions, and study the case of multiple attack sources. We also consider designing a traceback framework for MANETs that will work in a large spectrum of network settings by, for example, integrating the advantages of various schemes.

7. ACKNOWLEDGMENTS

We thank the reviewers for the valuable comments. This work of Zhu was supported in part by NSF CAREER 0643906 and the work of Hurson was supported in part by NSF under contract IIS-0324835.

8. REFERENCES

[1] Y. an Huang and W. Lee. Hotspot-based traceback for mobile ad hoc networks. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, pages 43–54, New York, NY, USA, 2005. ACM Press.

[2] H. Hsu, S. Zhu, and A. Hurson. A hotspot-based protocol for attack traceback in mobile ad hoc networks. http://www.cse.psu.edu/ szhu/papers/traceback.pdf.

[3] Y. Kim and A. Helmy. SWAT: Small world-based attacker traceback in ad-hoc networks. In *MobiQuitous*, 2005.

[4] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, 2000.

[5] T. Peng, C. Leckie, and K. Ramamohanarao. Adjusted probabilistic packet marking for IP traceback. In *IFIP NETWORKING*. Springer, 2002.

[6] V. Thing and H. Lee. Ip traceback for wireless ad-hoc networks. In *Proceedings of Vehicular Technology Conference (VTC2004-Fall)*, 2004.

[7] S. Zhu, S. Xu, S. Setia, and S. Jajodia. LHAP: a lightweight hop-by-hop authentication protocol for ad-hoc networks. *International Workshop on Mobile and Wireless Network (MWN)*, 2003.