AK-PPM: An Authenticated Packet Attribution Scheme for Mobile Ad Hoc Networks

Zhi Xu[2], Hungyuan Hsu[3], Xin Chen[2], Sencun Zhu[1,2], and Ali R. Hurson[4]

[1] College of Information Sciences and Technology, The Pennsylvania State University,
[2] Department of Computer Science and Engineering, The Pennsylvania State University
{zux103,xvc5038,szhu}@cse.psu.edu,
[3] Samsung Telecommunications America,USA,
hyhsu@sta.samsung.com,
[4] Department of Computer Science, Missouri University of Science and Technology
Hurson@mst.edu

Abstract. Packet traceback in mobile ad hoc networks (MANETs) is a technique for identifying the source and intermediaries of a packet forwarding path. While many IP traceback techniques have been introduced for packet attribution in the Internet, they are not directly applicable in MANETs due to unique challenges of MANET environments.

In this work, we make the first effort to quantitatively analyze the impacts of node mobility, attack packet rate, and path length on the traceability of two types of well-known IP traceback schemes: probabilistic packet marking (PPM) and hash-based logging. We then present the design of an authenticated K-sized Probabilistic Packet Marking (AK-PPM) scheme, which not only improves the effectiveness of source traceback in the MANET environment, but also provides authentication for forwarding paths. We prove that AK-PPM can achieve *asymptotically one-hop precise*, and present the performance measurement of AK-PPM in MANETs with both analytical models and simulations.

Keywords: Traceback; MANET, Probabilistic Packet Marking; Packet Source Identification; Path Reconstruction;

1 Introduction

Packet attribution includes identifying the source node of packets as well as the forwarding path from the source to the destination during the communication [1, 2]. Both source and path information can help the defender to identify the attack source and locate its geographic location in many mobile ad hoc networks (MANETs) applications, such as defending Denial-of-Service (DoS) attacks [3] and false data injection attacks [4]. In business applications, packet attribution can be used in a positive way to provide the trustworthiness (or credibility) of data received by a destination node (e.g., data sink node). Data credibility is not just about who reports the data, but also the path the data comes from [5].

Many IP traceback protocols have been proposed for the Internet [6–8]. Among these, two types of IP traceback schemes dominate the literature: probabilistic packet marking (PPM) [8, 9] and hash-based logging [7, 10]. However, these IP traceback techniques are not directly applicable in MANETs due to several unique challenges

in MANETs. First, packet forwarding paths in MANETs are easy to change due to node mobility [11], which causes the difficulty in reconstructing the attack path from a victim back to the attack source. Second, unlike the fixed routers in the Internet which are often assumed trusted, forwarding nodes in MANETs cannot be assumed as trusted, and compromised nodes may collude to confuse the traceback techniques. Moreover, because both the scale of a MANET and its data traffic rate are much smaller than that of the high-speed Internet, traceback in MANETs must be more efficient. So far, very little research has been done on traceback in MANETs [12–14].

In this paper, we make the first effort to quantitatively analyze the impact of node mobility on the performance of two representative traceback schemes (i.e. PPM and logging schemes). We formulate the impact of network parameters (e.g., the length of an attack path, the victim response time, and the mobility) on the *traceability* of these schemes in MANETs. Our analytical results show that (i) the traceability of both schemes decreases as node mobility increases; (ii) a PPM scheme is vulnerable to low-rate attacks, while a logging scheme performs poorly when a victim has a relatively high intrusion response time.

Further, we propose a new authenticated K-sized Probabilistic Packet Marking (AK-PPM) Scheme, which considers the efficiency and security requirements of traceback in the MANET environment. Our AK-PPM scheme stores multiple (up to K) marks within a single packet; thus, with the same number of packets received by a victim, more information about the forwarding path can be collected. Also, the AK-PPM scheme includes chained authentication mechanisms to protect the integrity of the mark sequence within a packet from being manipulated by colluding nodes in a forwarding path. We prove that is always *asymptotically one-hop precise*; that is, given enough attack packets, it can always trace to either an attack node, or the one-hop neighborhood of an attack node. We use analytical models and simulations to measure the performance of AK-PPM in MANETs of different settings.

2 Preliminaries

2.1 Network Model and Security Assumptions

In a MANET, nodes form a network on-the-fly and forward packets for one another. Nodes can establish trust through either a PKI, a Trusted Third Party (TTP), or pre-distributed shared keys. Further, any two nodes in the network, as long as knowing each other's id, can efficiently establish a pairwise key based on one of the existing schemes [15–17]. The key used in the message authentication code (MAC) generation at an intermediate node is its pairwise key share with the victim node. Therefore, malicious nodes cannot impersonate any benign node to the victim node. The link between two neighboring nodes is authenticated. During data forwarding, every packet is authenticated in a hop-by-hop fashion [18] with the pairwise key shared between neighboring nodes; thus, a malicious node cannot impersonate any good node and invalid packets are dropped right away. Such settings exist in many military MANET applications.

Without loss of generality, consider a forwarding path $\mathcal{A}_M$ of packet M in Figure 1. For a node u_i located on a forwarding path between the source S and the destination V, a node u_j is called its *upstream* node if u_j is closer to S than u_i is. Similarly, u_j is a *downstream* node of u_i if u_j is closer to V. The distance of u_i from V on a path is the

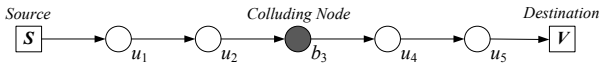


Fig. 1. An attack path $\mathcal{A}_M$ of packet M where node S and node b_3 are the compromised nodes. S injects bogus packets, and b_3 conspires to neutralize the traceback attempt.

number of hops(i.e. nodes) between u_i and V on the path. For example, the distance of b_3 to V is 2 in the Figure 1.

In the case of identifying the attack source node in a DoS attack, it is reasonable to assume the victim (i.e., destination) node has installed an appropriate intrusion detection system (IDS) (e.g., Snort [19]) which can detect the malicious intrusion in the first place. This is a realistic assumption for intrusion detection based on attack packets received by destination node. A path may become broken for various reasons. In this study, we focus on the factor of node mobility. The *Link Duration* between two neighbor nodes on the path is defined the length of time interval during which two nodes are within each other's transmission range.

2.2 Attack Model

We assume the adversary may compromise one or multiple nodes and take full control of the compromised node(s). For example, in Figure 1, 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 the packet's marks (if existing in the packet) or drop traceback queries. We present more details about specific attacks when introducing the proposed schemes in later Section 6.1.

Since a compromised node possesses a valid security credential, the injected packets will not be detected by its downstream nodes. However, because the links are authenticated, an attack source cannot impersonate any normal (benign) node. To hide itself, it will not put its address into the packet source field, and act as if it was a data forwarder for the packets while spoofing valid source ids. The attack source may change its location over time to hide itself.

2.3 Traceback Objectives

Ideally, a traceback procedure can identify the source node S and reconstruct the attack path $\mathcal{A}_M$. However, this goal is difficult to achieve due to two reasons.

Firstly, the source node S may never reveal its true identity so as to hide itself from traceback (a.k.a., the *first-hop* problem [13]). Thus, the best a traceback scheme can achieve is to identify the immediate downstream neighbor (e.g., u_1) of the attack source and reconstruct the path from the victim up to it. Once u_1 is identified, it relies on other online or offline analysis/detection measures (e.g., neighbor watching [20] or human intelligence) to identify the source node.

Secondly, compromised nodes on the attack path (e.g., b_3) may collude in order to confuse the traceback process. In an extreme case, b_3 may manipulate all attack packets going through it or even sacrifice itself to protect S. In this study, we assume that, from the attacker's perspective, the exposure of any one of its controlled nodes may have the same impact on the potential of future attacks. We call it a *success* when the immediate downstream neighbor of an attack source (e.g., S) or colluding node (e.g., b_3) on the attack path is identified.

3 Traceability Analysis of Existing Schemes for MANETs

Intuitively, an IP traceback scheme is not well suited for MANETs. However, no concrete analysis to quantify such intuition has been reported in the literature. A quantified analysis will clearly show how the current IP traceback protocols are susceptible in MANETs and it will serve as a metric for evaluating any new proposed traceback scheme for MANETs. As such, we will first make a traceability analysis of existing IP traceback schemes before presenting new schemes.

The common IP traceback schemes in the literature can be roughly categorized into *marking-based* schemes and *logging-based* schemes. Therefore, our discussion below will be focused on these two approaches. We define *traceability* $\mathcal{T}$ as the success rate of traceback in MANETs. $\mathcal{T}$ measures the probability that a traceback can be successfully performed before the attack forwarding path changes. We call it a success when the immediate downstream neighbor of an attack node (e.g., S or b_3) is identified.

3.1 Marking-Based Schemes

In a marking-based scheme, e.g., probabilistic packet marking (PPM) [9, 21], intermediate nodes (probabilistically) mark the packets being forwarded with partial path information, which later on allows a receiver to reconstruct the forwarding path given a modest number of the marked packets.

Scheme Description Take the edge sampling based PPM algorithm [9] as an example. An IP traceback mark consists of a distance field and a *start-end* pair. Every intermediate node decides to either inscribe a packet (with a preset probability p), or not to mark (with probability $1 - p$). As nodes are allowed to overwrite the existing mark in the received packet, nodes closer to V will have more chance to leave their marks in packets. Relying on the relation between the distance to V and the distribution of received marks, a traceback can reconstruct the attack path with order from u_1 to V. Here we assume that each packet carries only one mark at a time,and the nodes between u_1 and V are trustworthy.

Traceability Analysis for MANETs For an attack path $\mathcal{A}_M$ of length d, the victim can trace to the attack source only if it receives at least one packet marks from the immediate neighbor u_1 of the attack source before the path breaks up. Hence, the traceability of PPM for MANETs ,$\mathcal{T}_{ppm}$, is determined by two factors: *packet rate* γ and *path duration* PD, given a marking probability p. Let X_{u_1} denotes the number of packets that the victim has to receive before receiving the marking from u_1. The expected number of packets $E(X_{u_1})$ is:

$$E(X_{u_1}) = 1/(p(1-p)^{d-1}) \tag{1}$$

If the attacker sends the packets at a constant packet rate γ, then the expected time $T_{mark_{u_1}}$ by which the victim receives the marking from u_1 would be

$$T_{mark_{u_1}} = E(X_{u_1})/\gamma. \tag{2}$$

Sadagopan et al. [22] proposed a theoretical model to approximate the path duration based on the analysis of statistical extensive simulation results. According to [22], path duration can be approximated by an exponential distribution when the network nodes

move in moderate to high velocities. The exponential random variable has the following cumulative distribution function (CDF)

$$F_{PD}(t,d) = \begin{cases} 1 - e^{\frac{-\lambda_0 dv}{R}t}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{3}$$

where R denotes the radio transmission range, d denotes path length, v denotes the maximum velocity of a mobile node, and λ_0 is the proportionality constant.

Given the set of packets received by V, the victim V can launch a path reconstruction procedure [21] to reconstruct the order of marks(i.e., hops) on the path. Suppose that the reconstruction procedure is always performed correctly. The farthest hop u_1 will always be identified if V have received at least one mark from u_1. Thus, we define the traceability to identify u_1 as the probability that the path duration PD is greater than $T_{mark_{u_1}}$. The u_1-traceability for PPM is

$$\Im_{ppm_{u_1}} = 1 - F_{PD}(T_{mark_{u_1}}, d) \tag{4}$$

According to Equation 2 and 3, we may derive the traceability $\Im_{ppm,u_1}$ as

$$\Im_{ppm,u_1}(d,\gamma) = \exp\{\frac{-\lambda_0 v}{R \cdot p(1-p)^{d-1}} \cdot \frac{d}{\gamma}\} \tag{5}$$

The problem of marking every intermediate node in the path can be formalized as a *Coupon Collector's Problem with sample size as one* [21, 23]. Briefly, the number of trials required to select one of each of d coupons (i.e. hops in our case) can be estimated as $d(H(d) + O(1))$, where $H(d) = 1/1 + 1/2 + ... + 1/d$.

Note that, in works such as [21], $H(d)$ was replaced by $\ln(d)$. However, we know that $H(d) \to \ln(d)$ if and only if $d \to \infty$. As the number of hops d in MANETs is small, the replacement may cause errors. For example, when d is less than $e \approx 2.7148$, the value of $\frac{\ln(d)}{p(1-p)^{d-1}}$ will be less than $\frac{1}{p(1-p)^{d-1}}$. Thus, we use $H(d)$ in our study instead of $\ln(d)$. Because the mark of one node may be overwritten by downstream nodes in a PPM scheme, the upper bound of the number of packets required to collect marks of all hops can be estimated as

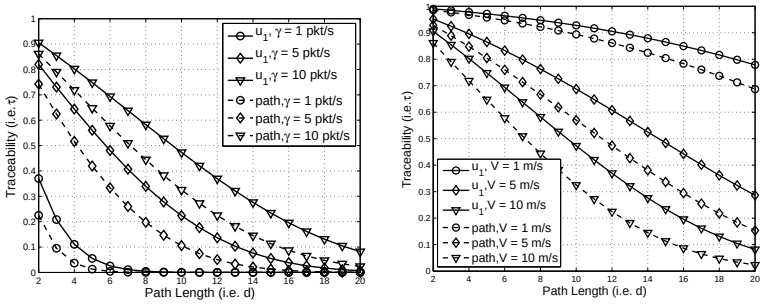
$$E(X_{path(d)}) < H(d)/(p(1-p)^{d-1}) \tag{6}$$

We derive the lower bound of path traceability $\Im_{ppm,path(d)}$, i.e., the probability of reconstructing an entire path of length d within the path duration PD.

$$\Im_{ppm,path(d)}(d,\gamma) > \exp\{\frac{-\lambda_0 v}{R \cdot p(1-p)^{d-1}} \cdot \frac{d \cdot H(d)}{\gamma}\} \tag{7}$$

Traceability Evaluation To demonstrates the limitation of PPM in MANETs, we show the impacts of packet rate and mobility on the traceability $\Im_{ppm,u_1}$ and $\Im_{ppm,path}$ in Figure 2. The parameters applied in experiments are: $\lambda_0 = 0.59$, R = 250m, v = 10 m/s (in Figure 2(a)), $p = \frac{1}{20}$, $\gamma = 10$ pkt/sec (in Figure 2(b)).

From Figure 2(a) and (b), one can conclude that the traceability of a PPM scheme decreases when d increases. Also, the efficiency of traceability drops when the packet rate decreases or the mobility of network increases. For example, an attacker may control the packet rate at 1 pkt/s and launch the attack at six or seven hops away from the victim to avoid traceback with PPM schemes. Since d and γ affect $\Im_{ppm}$ reversely and



(a) $\mathcal{T}_{ppm,u_1}$ vs. d and γ (b) $\mathcal{T}_{ppm,u_1}$ vs. d and velocity

Fig. 2. Traceability $\mathcal{T}_{ppm,u_1}$

v, R, and p are environment variables, we consider the traceability as a function of d and γ, i.e., $\mathcal{T}_{ppm,u_1} = f(d,\gamma)$. Unfortunately, both the path length d and the packet rate γ are controlled by the adversary. The victim can basically do nothing to improve the traceability in the PPM scheme. Therefore, the application of PPM traceback in MANETs, i.e. $\min_{d,\gamma} \mathcal{T}_{ppm,u_1}(d,\gamma)$, overwhelmingly favors the attacker.

3.2 Logging-Based Schemes

In a logging-based scheme, intermediate nodes record the message digest of a forwarded packet; thus, every packet leaves a trail on the path from its source to its destination. To reduce the storage overhead for keeping the message digests, a typical space-efficient data structure called *Bloom filter* can be used. A practical architecture, *Source Path Isolation Engine* (SPIE) (also known as the *hash-based traceback*) was proposed in [7], in which the SPIE-deployed routers audit the traffic and digest the invariant portion of each packet for later queries. When the victim identifies an attack, it launches a traceback query to a traceback agent that is authorized to poll each of the intermediate nodes. Each polled node identifies the false packet trail by looking it up in its own Bloom filter and reports the result to the agent. The traceback agent rebuilds the attack graph with the help of the information about the network topology.

Scheme Description To conduct on-line traceback for MANETs,the victim quickly responds to an attack by initiating a traceback process. The traceback query is expected to rapidly propagate along the reverse attack path, hoping that it can reach the neighbor of the attack source before the attack path breaks up. After processing the packet M, V finds that M is an attack packet(e.g., containing malicious code), so it sends out a traceback request along the reverse path of $\mathcal{A}_M$.

Traceability Analysis for MANETs Analogous to the PPM based scheme, the path duration PD also controls traceability here. Thus, we consider the traceback behavior from two points of view: *path duration* and *traceback latency*. In logging-based schemes, a traceback is successful when the traceback latency is smaller than path duration PD. The traceback latency consists of the end-to-end propagation delays plus the victim IDS response time t_{resp}. Unlike in the PPM based schemes, path reconstruction

and u_1 identification here are performed altogether in one round of traceback query. Therefore, we define the *traceability* $\mathcal{T}_{log}$ with regard to a detected attack packet x as $\mathcal{T}_{log} = \Pr\{PD > t_{SV} + t_{VS} + t_{resp}\}$. As a traceback process will only be initiated after the victim node receives the attack packet x successfully, we can rewrite the *traceability* formula as

$$\mathcal{T}_{log} = \Pr\{PD > t_{SV} + t_{VS} + t_{resp} | PD > t_{SV}\}. \tag{8}$$

As described previously, PD can be modeled as an exponential random variable that is *memoryless*. Thus, Equation 8 can be reduced to

$$\mathcal{T}_{log} = \Pr\{PD > t_{VS} + t_{resp}\} \tag{9}$$

Assuming the end-to-end delay is uniformly distributed and is proportional to the path length d (i.e., $t_{SV} \propto d$ and $t_{VS} \propto d$), the traceability $\mathcal{T}_{log}$ will then be a function of d and t_{resp}. Based on Equation 3, we have

$$\mathcal{T}_{log}(d, t_{resp}) = 1 - F_{PD}(t_{VS} + t_{resp}) = \exp\{\frac{-\lambda_0 dv}{R}(k \cdot d + t_{resp})\} \tag{10}$$

where k is the average delay in an intermediate node. Clearly, $\mathcal{T}_{log}$ decreases as the node mobility (i.e., v) increases. Similar to the way we treat v and $\mathcal{T}_{ppm}$, because v and t_{resp} affects $\mathcal{T}_{log}$ in the same trend and v is an environment variable, we study the impact of t_{resp} on $\mathcal{T}_{log}$ instead.

Traceability Evaluation Compared with a marking-based scheme, the logging-based online traceback scheme is more fair. The victim node controls the response time t_{resp} and the attacker only controls the length of the attack path d. According to Equation 10, we notice that $\mathcal{T}_{log} \searrow 0$ as $d \nearrow \infty$. Thus,$\mathcal{T}_{log}$ unfairly favors the adversary when the path length d is unbounded. But, the longer the attack path is, the more likely the attack packet will be lost due to the changes in topology.

3.3 Review of Factors

Basing on our analytic results on PPM and logging-based schemes, several factors should be considered when designing traceback schemes for MANETs.

Attack Packet Rate In MANETs, the attack packet rate required to launch DoS or DDoS attacks may be much lower [24]. Firstly, the targeted mobile node is more resource restricted, requiring lower attack packet rate to exhaust its resources (e.g., bandwidth consumption attack). Secondly, to protect the attack source from being exposed or from exhausting resources, the attacker may reduce the attack packet rate. Therefore, traceback schemes in MANETs have to adapt to the cases with low attack packet rates. PPM schemes may find it difficult to receive markings from u_1 and all other nodes on the path. In contrast, logging-based schemes are more resistent to low-rate attacks.

Communication Overhead Due to the limited bandwidth in MANETs, a traceback scheme should introduce little traffic bandwidth overhead so as to prevent the downgrade of network services. PPM schemes generate little communication overhead, especially when marks are stored in the packet header. Logging-based schemes, however, generate much more communication overhead by sending traceback queries and receiving responses from the network.

Resource Costs One assumption made in most existing traceback schemes is that all involved nodes are willing to cooperate during the traceback, by marking or logging the packets. In MANETs, the willingness of nodes may be affected by the cost for cooperation in the traceback, such as the cost of bandwidth, computing power, storage, and battery power. Thus, traceback schemes should avoid excessive workload on mobile nodes in order to make the scheme feasible to deploy in MANETs. In this case, PPM schemes usually demand comparatively lower resource consumption on nodes than logging-based schemes do.

Speed of Traceback Improving the speed of traceback will help reduce the damage of DoS attacks to the network. Moreover, in MANETs, the speed of traceback is also important to identify the attack source and locate its geographic location. For example, the location information can be used to physically isolate or remove the attack source after traceback . The longer time it takes by the traceback the less chance the attacker can be found. Unfortunately, both PPM schemes and logging-based schemes have drawbacks in terms of speed. PPM schemes have to wait passively until enough marked attack packets have been received. On the other hand, logging-based schemes actively send traceback queries to the network but have to wait until receiving enough responses. Moreover, when the response time of the IDS in the victim is large, the attack path could become broken when a traceback is launched.

In general, PPM schemes seem to be more adaptable than logging-based schemes because the resource consumption of nodes and network is a critical issue in MANETs. However, improvements are needed on existing PPM schemes.

4 Improving Traceback Efficiency With Multiple Marks

Most existing marking based schemes store marks in the packet header [21]. Due to the fixed size of IP header space, the amount of routing information that can be carried in an IP packet is limited. For example,the single bit based PPM schemes in [25] and [26] require a huge number of packets for a successful path reconstruction. Differently, in MANETs, the packet format is relatively flexible, making it feasible for MANET designers to allocate more space in the packet header for multiple marks. Therefore, we may store multiple marks within a single packet to improve the efficiency of traceback. In untrusted MANETs, enabling multiple marks in a packet faces two challenges.

Challenge I: How to determine the number of allowed marks in the packet? The number of marks allowed per packet is a key factor to the traceback efficiency. The more marks allowed, the more information a packet can carry about the forwarding path. Thus, if the number of marks is too small (e.g., single mark in [9]), both source identification and path reconstruction will require a huge amount of packets. To another extreme, if the number of marks is unrestricted (e.g., nested marking approach in [27]), the amount of payload information in a packet will be affected.

Challenge II: How to protect the integrity of marks and their ordered sequence in a packet under the colluding attacks? A malicious intermediate node on the forwarding path may attempt to alter the marks carried by packets, in order to hide the source node and itself. Security mechanisms are needed in the mark design to protect the integrity of mark sequence and allow the victim node to detect the manipulation if existing.

We will address these two challenges in the following sections one by one.

5 K-Sized Probabilistic Packet Marking Scheme

In this section, we present a base scheme of K-sized Probabilistic Packet Marking (K-PPM) scheme, which improves the efficiency of packet traceback by allowing multiple marks in a packet. The proposed K-PPM scheme can be applied in trusted MANETs where the intermediate nodes are trusted. In the next section, we present an extended scheme that provides protection in untrusted MANETs.

5.1 Scheme Design

K-PPM scheme consists of two phases. In the first phase, every node in the MANET inscribes packets it is forwarding with a predefined probability p. Whenever source attribution is needed (e.g., DoS attack detected), a path reconstruction algorithm will be executed at the destination node based on the marks carried in received packets.

Marking Scheme In the proposed K-PPM scheme, every packet contains a K-sized queue (i.e.,Q), which is managed by the First-In First-Out (FIFO) replacement algorithm. Each mark in Q consists of two node IDs. One is the ID of the current node which is forwarding this packet (i.e.,rcv), and the other is the ID of the node from whom the current node receives this packet (i.e., sdr).

Initially, when a packet leaves the source S, the queue Q is empty. When the packet arrives at an intermediate node u_i on the path, the node places a mark in the packet with a preset probability p. If the node decides to mark, it will generate a new mark containing its node ID i as rcv, and append this mark to the end of Q. If the queue is full already when arriving, the first (i.e., oldest or leftmost) mark in the queue will be discarded so that a new mark can be appended if u_i decides to mark. On the other hand, if u_i decides not to mark the packet, it will pass the packet to its downstream neighbor with no modification.

Path Reconstruction With the collected marks, we first build a directed graph $G = (V, E)$: V consists of the set of nodes whose IDs appeared in received marks (either as a sdr or a rcv) at the destination node; and E consists of edges created by two rules:
Rule One: Assign a directed edge $j \rightarrow i$ if there exists a received mark within which the sdr is node j and the rcv is node i;
Rule Two: Assign a directed edge $j \rightarrow i$ if node i is the sdr of a received mark and node j is the rcv of its left adjacent mark in Q. Because nodes appearing in the left adjacent mark must be upstream nodes in the forwarding path.

In G, let I denote the set of nodes whose in-degree is 0, and let O denote the set of sink nodes whose out-degree is 0. Suppose that all packets traveled through the same forwarding path. With a sufficient amount of received packets, the sizes of I and O will be narrowed down to 1. The only node left in I will be output as the *identified source node* and the longest path from the node in I to the node in O will be output as the *identified forwarding path*.

With insufficient number of received packets, the accuracy of source identification and path reconstruction may be affected. First, the longest path may not traverse all nodes in G. In this case, we utilize the distribution of received marks to infer the order of nodes. Because the marking process is probabilistic with permission of dequeue,

nodes closer to the destination node will have more chance to keep their marks in the packets. Due to the space limit, please refer to [9] for the detailed procedure. Second, the size of I may not be one. In this case, I represents a hotspot where any one of them has equal chance to be the real source node. Thus, extra investigation, e.g., neighbor monitoring [20] and physical security [28], can be conducted on nodes in I.

5.2 Improvement for u_1 Identification

We analyze the traceability of the proposed K-PPM scheme in the same setting as in Figure 1. Let $E_k(X_{u_1})$ denote the expected number of packets that the victim has to receive before receiving the mark from u_1. In the K-PPM scheme, the mark from u_1 will remain in the packet as long as the packet is later marked by no more than $K-1$ downstream nodes. If more than $K-1$ nodes mark the packet after u_1, the mark from u_1 will be discarded from the queue according to the FIFO policy. Thus, the expected number of packets, $E_k(X_{u_1})$, can be computed as:

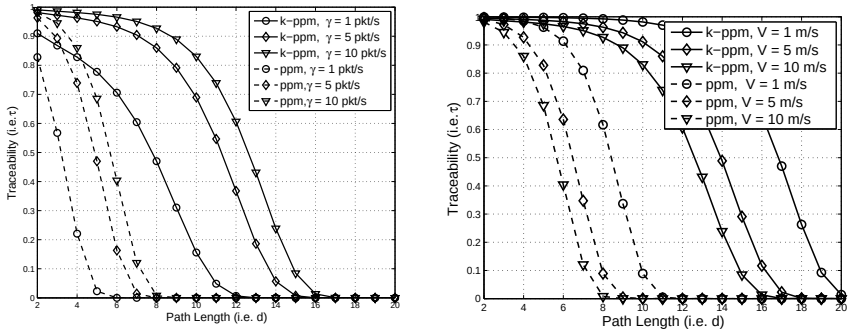
$$E_k(X_{u_1}) = \frac{1}{p\sum_{i=0}^{k-1}(\binom{i}{d-1}p^i(1-p)^{d-1-i})}. \tag{11}$$

In Equ. 11, the value of $0 \leq i \leq k-1$ represents the number of marks within the packet, other than that from u_1. We claim that the K-PPM always requires less expected packets to receive a mark from u_1 (i.e., $E_k(X_{u_1}) \leq E(X_{u_1})$). The proof is as follows: When $k=1$, we see that $E_k(X_{u_1}) = \frac{1}{p\sum_{i=0}^{k-1}(\binom{i}{d-1}p^i(1-p)^{d-1-i})} = \frac{1}{p(1-p)^{d-1-i}} = E(X_{u_1})$. Thus, we show that, when $k=1$, $E_k(X_{u_1})$ and $E_(X_{u_1})$ are equivalent. When $k>1$, $\sum_{i=0}^{k-1}(\binom{i}{d-1}p^i(1-p)^{d-1-i})$ is always greater than $\sum_{i=0}^{0}(\binom{i}{d-1}p^i(1-p)^{d-1-i})$, thus greater than $p(1-p)^{d-1-i}$. ■

Accordingly, we derive the traceability $\Im_{k-ppm,u_1}$ with respect to node u_1 as

$$\Im_{k-ppm,u_1}(d,\gamma) = \exp\{\frac{-\lambda_0 v}{R\cdot p\sum_{i=0}^{k-1}(\binom{i}{d-1}p^i(1-p)^{d-1-i})}\cdot\frac{d}{\gamma}\} \tag{12}$$

In Figure 3(a) and 3(b), we compare the K-PPM scheme with PPM scheme in terms of u_1 traceability. The default setting in these two figures are $\lambda_0 = 0.59$, R = 250m ,$K=4$, $p=0.5$,v = 10 m/s(in Figure 3(a)), $\gamma = 10$ pkt/sec(in Figure 3(b)). Clearly, the K-PPM scheme can greatly improve the traceability in all mobility and attack packet rate settings.



(a) $\Im_{k-ppm,u_1}$ vs. d and γ (b) $\Im_{k-ppm,u_1}$ vs. d and V

Fig. 3. Traceability $\Im_{k-ppm,u_1}$

5.3 Improvement for Path Reconstruction

We formulate the path reconstruction problem in K-PPM scheme as a *Coupon Collector's Problem with random sample size* [29, 30]. In our case, the sample size is a random integer value between 0 and the size of queue, i.e., K. Following the approximation equation given in [30], we present the approximation of expected number of packets required to reconstruct a path with d intermediate nodes as:

$$E_k(X_{Path_d}) \approx \frac{\sum_{i=0}^{d-1} \frac{1}{d-i}}{\sum_{i=0}^{d-1} \frac{1}{d-i} Pr\{L > i\}} + \frac{\sum_{r=1}^{d-1} \frac{1}{d-r} Pr\{L > r\} \sum_{j=1}^{r} 1/(d-j+1)}{[\sum_{i=0}^{d-1} 1/(d-i) Pr\{L > i\}]^2} \quad (13)$$

In the above Equation 13, L represents the number of marks carried by a packet and $Pr\{L > i\}$ represents the probability that L is greater than a value i. In our case, L is restricted by K and d, the length of path. This approximation is accurate when K is small with respect to d according to [29]. Thus, the value of $Pr\{L > i\}$ for $0 \leq i \leq d$ can be computed by:

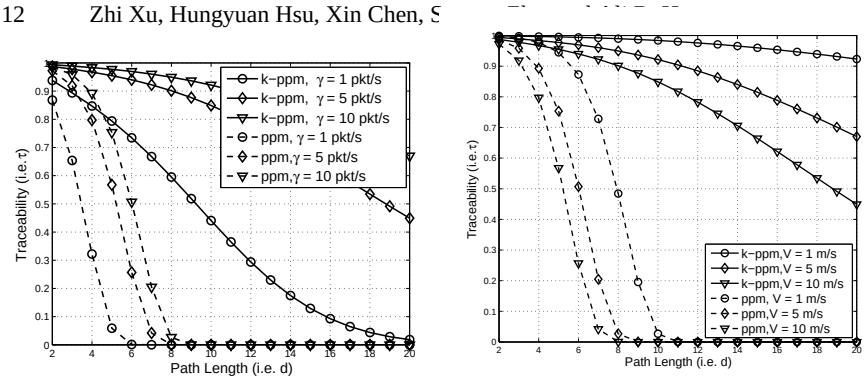
- when $d \leq K$,
  - if $i > d$, $Pr\{L > i\}$ = 0;
  - if $i \leq d$, $Pr\{L > i\}$ = $1 - \sum_{j=0}^{i} Pr\{L = j\}$;
  - $Pr\{L = i\} = \binom{i}{d} p^i (1-p)^{d-i}$ for $0 \leq i \leq d$;
- when $d > K$
  - if $i > K$, $Pr\{L > i\}$ = 0;
  - if $i \leq K$, $Pr\{L > i\}$ = $1 - \sum_{j=0}^{i} Pr\{L = j\}$;
  - $Pr\{L = i\} = \binom{i}{d} p^i (1-p)^{d-i}$ for $0 \leq i < K$; $Pr\{L = K\} = 1 - \sum_{j=0}^{K-1} Pr\{L = j\}$;

In Figure 4(a) and 4(b), we present an example of comparison with $p = 0.5$, $k = 4$, $\lambda_0 = 0.59$, R = 250m , v = 10 m/s(in Figure 4(a)), $\gamma = 10$ pkt/sec(in Figure 4(b)). Through the comparison, we show that, the proposed K-PPM is more resistent to the increase of mobility and the decrease of packet rate, thus it is more suitable to be applied in MANETs than PPM scheme.

Moreover, by examining the Equations (12) and (13), the traceability of a K-PPM scheme decreases when d increases. Also, the efficiency of traceability drops when the packet rate decreases or the mobility of network increases. Luckily, the destination node can increase K to improve the traceability in the K-PPM scheme. Therefore, the K-PPM scheme is a more fair scheme compared with a PPM scheme, in which both the u_1 identification and path reconstruction can be described as: $\min_{d,\gamma} \max_K \mathcal{T}_{k-ppm}(d, \gamma, K)$.

5.4 Parameter Selections

Of all parameters, K and p are defined by the network administrator. K is decided according to the tradeoff between traceback efficiency and overhead. The impact of p on traceability varies, depending on the path length. If p is too small, not enough marks can be collected at the victim. If it is too big, marks from nodes closer to the attack source are likely to be dequeued. In both cases, the traceability will be low. To maximize the usage of allocated space while avoiding too few markings, it is recommended that p be set according to equation $K = p * d_{est}$, where d_{est} denotes the estimated path length in a specific MANET. Detailed evaluations are presented in Section 7.



(a) $\mathcal{T}_{k-ppm,path}$ vs. d and γ (b) $\mathcal{T}_{k-ppm,path}$ vs. d and velocity

Fig. 4. Traceability $\mathcal{T}_{k-ppm,path}$

6 AK-PPM Scheme

In this section, we present an extended scheme, named AK-PPM (authenticated K-sized Probabilistic Packet Marking), to perform the traceback with the same efficiency as the proposed base scheme in untrusted MANET environments.

We first introduce the possible attacks toward the K-PPM scheme, then present the detailed design of AK-PPM scheme, and finally analyze its security.

6.1 Possible Attacks

The integrity of mark sequence in packets is crucial to the correctness of source attribution. Therefore, in AK-PPM, we focus on colluding attacks that could be launched by a malicious intermediate node on the forwarding path at the integrity of mark sequence in a packet. The purpose of these attacks is to conceal the real source node of a forwarding path. A colluding node succeeds if it can launch the attack without being noticed by the verifier at V.

Specifically, we consider the following possible attacks: (1) No-Mark Attacks: the colluding node may remove all marks in Q; (2) Mark Altering Attacks: the colluding node may modify the marks; (3) Mark Re-ordering Attacks: the colluding node may re-order the existing marks in Q; (4) Mark Insertion Attacks: the colluding node may insert at least one faked mark in Q; (5) Mark Deletion Attacks: the colluding node may arbitrary drop marks in Q; (6) Prefix Removal Attack: the colluding node may dequeue marks from the head of Q when Q is not full, dequeue one or many marks without adding its mark, or dequeue multiple marks while only adding its own mark. (7) Suffix Removal Attack: the colluding node may remove one or many marks from the tail of Q. (8) Jamming Attack: the colluding node(s) may jam Q by faked or legitimate marks (when the attack has control of multiple nodes in the network).

6.2 Scheme Objectives

The objective of AK-PPM scheme is to detect those attacks and identify either the real source node or the colluding node. Note that simply computing a single MAC over the entire packet (e.g., as in the nested marking scheme [27]) does not solve the attacks in

our case, because of the dequeue operation in the K-PPM scheme. Briefly, if a mark is dequeued, all MACs including this dequeued mark will not be verifiable by the receiver.

Therefore, the AK-PPM scheme extends the base scheme by introducing a new chained authentication mechanism to protect the integrity of the mark sequence within a packet from being manipulated by colluding nodes in a forwarding path. Through security analysis in the end of this section, we prove that is always *asymptotically one-hop precise*; that is, given enough attack packets, it can always trace to either an attack (i.e., source or colluding) node, or the one-hop neighborhood of an attack node.

6.3 Revised Mark Design

In AK-PPM, an intermediate node will still mark the packet with the probability p. However, the mark format will be different. In Figure 5, we explain the details of AK-PPM with $K = 2$. Initially, S sends out a message M towards the destination V with a random variable $F = F_0$. Let H_k be a MAC function. If an intermediate node u_i, whose previous hop is u_j, decides to inscribe the packet, it will update the value of $F = F \oplus H_{k_i}(M|u_i)$, here k_i is the pairwise key shared between u_i and V. Note that a node can easily compute its pairwise key shared with another node given each other's id [15–17]. The MAC of $M|u_i$ serves as its "footprint" and we will see in the verification phase how it helps detect the marks in Q from being removed starting from the end. It also inserts its own mark $mark_i$ in the queue Q. In its mark, u_i includes its id and its previous hop u_j, the position I_i ($1 \leq I_i \leq K$, starting from the leftmost) of its mark in Q after adding its mark, a MAC $H_{k_i}(mark_j)$ over the previous mark $mark_j$ in Q (or a MAC of M if no previous mark exists), and a second MAC $H_{k_i}(M|F|u_j|u_i|I_i|H_{k_i}(mark_j))$. The new $mark_i$ will be appended at the end of Q. Note that here the first MAC in the mark provides an authenticated link to the previous mark; as a result, all the marks in a packet are protected in a chained structure. Removing any existing mark in the middle of Q will cause the chain to be broken and hence detected. If Q is already full, its first mark will be dequeued. If the node u_i decides not to inscribe the packet, it will simply forward the received packet to the downstream neighbor node without any change.

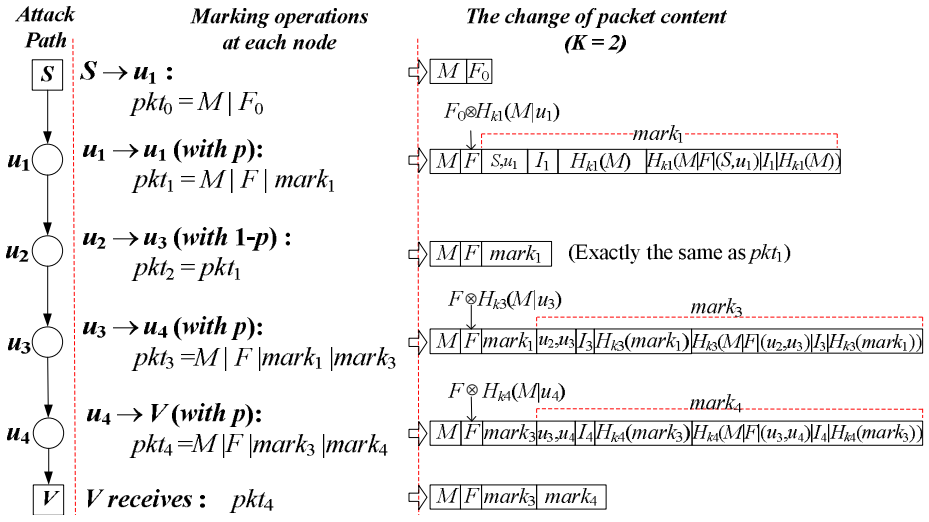


Fig. 5. An example of marking procedure

6.4 Mark Verification

As forwarding nodes in MANETs cannot be assumed as trusted, we have to verify the integrity of marks carried in received packets before using them in traceback. For each received packet, the destination V verifies the marks in Q in a backward order starting from the end of Q. Given a mark $mark_i$ whose marker is u_i and the previous hop is u_j (shown in the beginning of the mark), V first identifies its pairwise key k_i shared with u_i. With k_i, it computes the value $MAC_x = H_{k_i}(M|F|u_j|u_i|I_i|MAC_i^1)$ where MAC_i^1 is the first MAC in $mark_i$, and then compare MAC_x with the second MAC in $mark_i$. If the two MAC values are the same, we know the mark $mark_i$ has not been tampered. Thus, we can use its first MAC to verify whether the link between the current mark $mark_i$ and the left adjacent mark $mark_j$ is authenticated. Specifically, we compute the value of $H_{k_i}(mark_j)$ and compare it with the first MAC in $mark_i$. If the results match, it means $mark_j$ was the last mark in Q when u_i was adding its own mark $mark_i$ at the time of packet forwarding.

If the mark $mark_i$ itself is not tampered, we continue to check if there exist illegal enqueue/dequeue operations in Q using the position variable I_i. As mentioned in the marking phase, according to our policy a node is allowed to dequeue the first mark of Q if and only if Q is already full when it tries to insert its own mark at the end of Q. Thus, if Q in the received packet is not full, the position of each mark, contained in each mark, should be exactly the same as its current position in Q. If Q is full, the position value in the last mark must be K. As a result, a valid position sequence $\{I_i\}$ will be either K or fewer consecutive items from the list $\{1, 2, ..., K, K\}$ depending on whether Q is full or not. If an inconsistency is detected, there must be some violation of the enqueue/dequeue policy (e.g., dequeue marks when Q is not full) in the stored marks. For example, when $K = 3$, it could be $\{1, 2\}$ (when the queue is not full), $\{1, 2, 3\}$ (when the queue is full but no dequeue happened), $\{2, 3, 3\}$ or $\{3, 3, 3\}$ (when dequeue happened). On the other hand, $\{2, 3\}$ is not legitimate because it indicates a malicious node in the path removed the first mark without adding its own one.

At last, we verify the value of F, which is used to prevent a malicious intermediate node from removing marks from the end of Q without being detected. During the marking process, every node u_i inscribing the packet has updated F with its "footprint", i.e. $H_{k_i}(M|u_i)$. Since F is part of input to the second MAC in the last mark, if the mark is not tampered, F will be authenticated. We will then compute $F = F \oplus H_{k_i}(M|u_i)$, which should be the F used in the calculation of the second MAC in the left adjacent mark. Iteratively, we can derive the value of F used in the construction of each mark in Q in the reverse order. This is possible because $\oplus$ is symmetric. If an intermediate malicious node removed the last mark in Q, it will also need to update F correctly. Otherwise none of the marks can be verified. This will require the malicious node to know the secret key shared between the previous marker and the destination node to compute the "footprint" correctly.

6.5 Security Analysis

Mark Integrity Assurance We prove that the compromise of integrity of Q will always be detected in the proposed *AK-PPM* scheme.
Claim 1: Dropping marks from the end of Q will be detected. **Proof:** Every node inscribing the packet updates F with its "footprint". If an intermediate malicious node

removes marks from the end of Q, the inconsistency on value F will be detected in mark verification.
Claim 2: Dropping marks from the beginning of Q will always be detected.
Proof: The position variable I_i in a mark shows the status of Q when this mark is enqueued. As mentioned in mark verification, inconsistency can be detected on I_i if illegal dequeue operations were done by an intermediate malicious node.
Claim 3: Any enqueue/dequeue operation that violates the enqueue/dequeue policy will be detected.
Proof: The same as the proof of the previous claim.
Claim 4: Removing or inserting marks in the middle of Q will be detected.
Proof: The first MAC in every mark within Q provides an authenticated link to its previous mark. Consider a mark $mark_i$ and its previous mark $mark_j$ in Q. If an intermediate malicious node deletes $mark_j$ or adds a new mark after $mark_j$, it will be detected by $mark_i$ in mark verification.

Traceback Capacity Analysis We prove that *AK-PPM is always asymptotically one-hop precise no matter whether the attack path is trusted or untrusted*.
Definition 1 (Trusted Attack Path): an attack path is trusted if all intermediate nodes in this attack path between attack source S and destination V are legitimate nodes.
Definition 2 (Untrusted Attack Path): an attack path is untrusted if there exist at least one colluding node in the attack path.
Definition 3 (One-hop precise): A traceback scheme is *one-hop precise* if it can always trace to either an attack node (i.e., the attack source or a colluding node in the path), or the one-hop neighborhood of an attack node.
Definition 4 (Asymptotically One-hop precise): A traceback scheme is *asymptotically one-hop precise* if it can always achieve one-hop precision when enough attack packets are received at destination node V.
Theorem 1 *The AK-PPM scheme is asymptotically one-hop precise if the attack path is trusted*.
Proof: First of all, in AK-PPM, the destination node V is able to receive marks from every node in the trusted path when enough attack packets are received, because every intermediate node in the attack path will inscribe the packet with the probability p. Further, the enqueue/dequeue operation allows the nodes closer to V to inscribe the packet event when Q is full.

Secondly, we prove that AK-PPM is asymptotically consecutively traceable. Consider two consecutive legitimate forwarding nodes u_j and u_i. As we have just proved, V is able to receive marks from both u_j and u_i. When $K \geq 2$, V will be able to receive packets which carry marks from both u_j and u_i. As both nodes are neighbors in the path, their marks must be neighboring with the same order in Q. With the chained structure, an authenticated link from u_j to u_i can be verified. Therefore, if we can trace to u_i, we can trace to u_j as well.

In [27], the author has proven that a scheme can achieve one-hop precision if and only if it is consecutively traceable. Therefore, the proposed AK-PPM scheme is asymptotically one-hop precise.
Theorem 2 *AK-PPM is asymptotically one-hop precise if there exists only one colluding node in the attack path.*
Proof: Let b denote the colluding node. For packets passing through b, it will have two choices: either tamper the existing marks in Q in order to hide S from traceback, or not

tamper the marks. If b decides to tamper the marks, according to Theorem 1, we can trace to b because the sub-path between b and V is a trusted path. If b decides not to tamper the marks, S will be traced. In either case, an attack node is identified and we achieve our goal.

Theorem 3 *AK-PPM is asymptotically one-hop precise if there exist more than one colluding node in the attack path*

Proof: If all colluding nodes do not tamper the marks within packets, S will be traced. Otherwise, let b denote the colluding node closest to V, which tampers the marks within packets it forwarded. As we have shown in Theorem 2, we will be able to trace to b. Again, in either case, an attack node is identified and we achieve our goal.

Corollary 1 *AK-PPM is always asymptotically one-hop precise.*

Proof: With Theorem 1, 2, and 3, we come to the Corollary 1 that, AK-PPM is always asymptotically one-hop precise.

6.6 Parameter Selections

The proposed AK-PPM requires $K * Size(Mark) + Size(F)$ of extra space in every packet. Each mark consists of two node IDs, a position index up to K, and two MAC values, so its size is $2|ID| + \lfloor \log_2(K) \rfloor + 2|H()|$. The MAC value can be generated by a secure one-way hash function. The detailed settings are decided by the network administrator. Suppose that $K = 3$, $|ID| = 8$ bits, $|H()| = 16$ bits, $|F| = 16$ bits. The per-packet overhead of AK-PPM is $(16 + 3 * (16 + 2 + 32)) = 166$ bits = 21 Bytes. Suppose that we do not change the IP header and store all marks within the packet payload. For a packet of size 512 bytes, the overhead rate is 4%. To further lower the marking overhead, we may make a tradeoff between security and performance. For example, we may reduce the size of the MACs contained in the marks to one byte. In the previous example, it will give us the per packet overhead of 15 bytes. Because the number of attack packets is not big in an MANET, a one-byte MAC could be sufficient to filter out forged packets.

6.7 Anonymous ID

In the AK-PPM scheme, an intermediate node will include its node ID in generated marks. If the included node ID is in plaintext, the colluding node on the path may selectively drop only packets which contain marks generated by nodes close to S (i.e., *selective dropping attack*). So that the later traceback will end at an upstream node of the colluding node that is far from S. To avoid other nodes knowing who have inscribed the packet, an intermediate node may include an anonymous ID instead of its real ID in plaintext. Depending on the initiator of traceback, the design of anonymous ID can be different. If the traceback will be launched by the network administrator, the intermediate node may use a pseudo ID that is verifiable by the network administrator only. If the traceback will be launched by V and the intermediate node shares a paired key with V, we may use the paired key to encrypt the node ID and use the encrypted ID in the mark. Further, we may add randomness in the anonymous ID (e.g., by applying *Counter-Mode Encryption*) to prevent attackers from mapping the anonymous IDs with their real ID/nodes by observing the packet traffic.

7 Simulation and Evaluation

We use simulations to show (1) how to select proper values for parameters K and p, and (2) the improvement of traceability by the proposed AK-PPM scheme.

7.1 Simulation Settings

Our simulations are based on the GlomoSim 2.03 simulator. Major simulation parameters are listed below: physical link bandwidth (2 Mbps), transmission range ($R = 250m$), number of nodes (100), territory ($3000m \times 3000m$), MAC layer protocol (IEEE 802.11). The AK-PPM scheme is implemented in the network layer. In the network layer, we apply the DSR routing protocol.

In the simulation, we randomly chose two nodes as the attack source and the destination node, respectively. The attack node sends packets to the destination node in a constant rate (i.e., γ). γ is chosen from 1, 5, and 10 pkt/sec to simulate different attack intensities. An intermediate node follows the marking policy proposed in the AK-PPM scheme with probability p chosen in the range of 0.05 and 1.0. In addition, K is chosen from 1 to 5. We evaluate the performance of the AK-PPM protocol while changing these three parameters. By default, $\gamma = 10$pkts/s, $p = 0.5$, and $K = 4$. For each parameter setting, we ran the simulation for at least 20 times. Each simulation run lasts two hours in simulation time. Every node moves in the random waypoint mobility model at a speed uniformly distributed between 0 and $10m/s$.

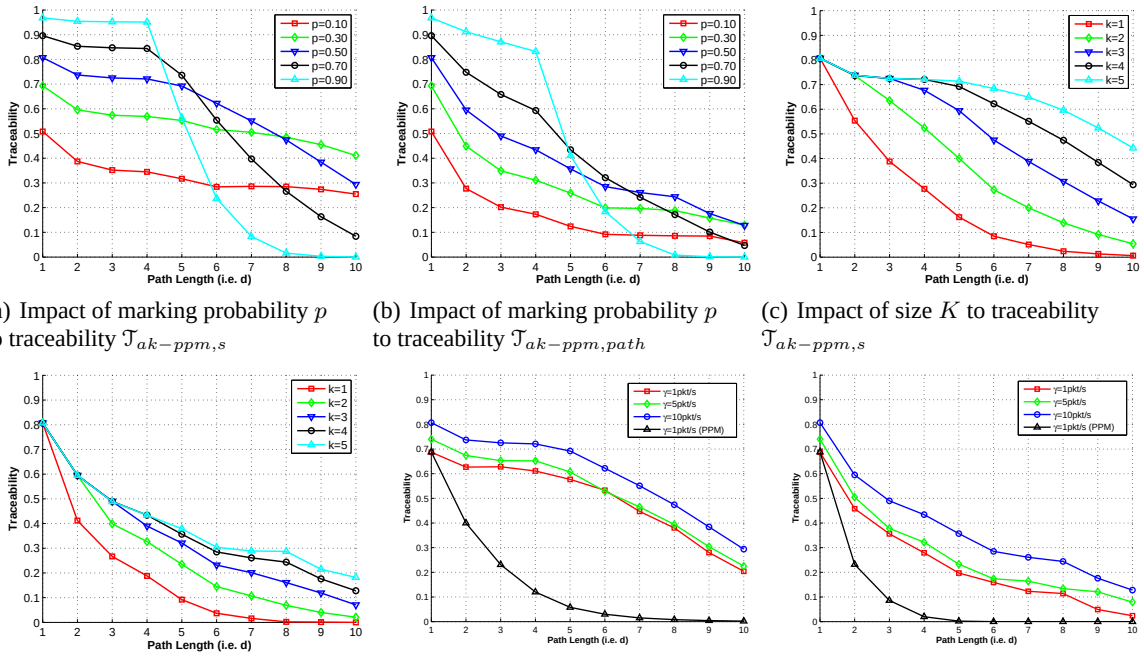
7.2 Simulation Results

Figure 6 shows the simulation results for traceability of both source and path.

Parameter p: As shown in Figure 6(a) and (b), increasing p does not necessarily improve the traceability. The curves can be divided into two parts. When $d \leq K = 4$, because no mark overwriting happens, a larger p will certainly improve the traceability. However, when $d > K$, the traceability will also be affected by possible mark dequeuing operations. In this case, when p is large, the probability that marks of u_1 and other nodes far from V are overwritten will be high. For example, we see a swift downgrade of traceability when $p = 0.9$ in both figures.

Parameter K: the size of queue K represents the space in a packet reserved for source attribution. In Figure 6(c) and (d), p is set as 0.5 and we show the impact of K to the traceability in the proposed AK-PPM scheme. When $K = 1$, the AK-PPM scheme is the same as the PPM scheme. From these two figures, we can see that increasing K will obviously improve traceability, especially for source identification. However, the improvement is not significant when the d is small. In summary, to maximize the usage of allocated space while avoiding insufficient marking, it is recommended that p and K be set according to equation $K = p * d_{est}$, where d_{est} denotes the estimated path length in a specific MANET.

Low Packet Rate γ: the packet rate γ will be small in cases, such as low rate DoS attacks [24], In Figure 6(e) and (f), we show the impact of the attack packet rate on traceability when $K = 4$ and $p = 0.5$. For comparison, we also show the simulation result of PPM scheme on the same traces. As we can see, compared to the PPM scheme, the downgrade of traceability for AK-PPM is less significant, meaning that the AK-PPM scheme is more resistent to low-rate attacks.



(a) Impact of marking probability p to traceability $\mathcal{T}_{ak-ppm,s}$

(b) Impact of marking probability p to traceability $\mathcal{T}_{ak-ppm,path}$

(c) Impact of size K to traceability $\mathcal{T}_{ak-ppm,s}$

(d) Impact of size K to traceability $\mathcal{T}_{ak-ppm,path}$

(e) Impact of packet rate γ to traceability $\mathcal{T}_{ak-ppm,s}$

(f) Impact of packet rate γ to traceability $\mathcal{T}_{ak-ppm,path}$

Fig. 6. Simulation Results.

8 Related Work

About traceback schemes in MANETs, Thing and Lee [12] conducted simulation studies to investigate the feasibility of applying SPIE, PPM, and ITrace protocols in MANETs. However, no quantitative study about node mobility is presented and the factor of mobility is considered by simulations. Zarai et al. [31] and Kim et al. [32] proposed cluster-based traceback schemes only for MANETs with trusted nodes. Kim and Helmy [14] proposed a traceback scheme in MANETs, named $SWAT$. Their scheme utilizes the small world model and sends the attack traffic signature to neighbor nodes of a victim which observe similar attack traffic. The major drawback of SWAT is the large amount of communication cost during the traceback. Huang and Lee [13] first introduced the concept of hotspot-based traceback and proposed traceback schemes to reconstruct the attack path in MANETs. Again, the communication overhead in the network in the major drawback. Recently, Hsu et al. [33] proposed a hotspot-based traceback protocol for MANETs, which divides a forwarding path dynamically into multiple smaller interweaving fragments. Unlike previous traceback schemes, our proposed *AK-PPM* scheme is much light-weighted and is secure in untrusted MANETs. It places little computation workload to the intermediate nodes. Once an attack is identified, no additional query is required to reconstruct the attack path.

In addition, [34] and [27] adopt multiple marks but neither takes mobility into consideration. In [34], the authors proposed a router stamping scheme for wired networks, and discussed the tradeoff between the number of marks allowed in a packet and the

performance of identifying the u_1 node on the path. However, all intermediate nodes on the attack path are assumed trusted and no security mechanism was presented to protect marks. Also, path reconstruction was not discussed in this work. In [27], the authors proposed a Probabilistic Nested Marking approach for traceback in sensor networks, which protects the integrity of multiple marks stored in packets using cryptographic techniques. However, the number of marks in the proposed scheme is unrestricted, making it difficult to packet format design. When the path is long, the large amount of marks will take too much space in the packet. Also, there exists a flaw in the proposed nested marking scheme which could allow a colluding node in the path to remove the marks in the end without being detected. We propose a new authenticated K-sized Probabilistic Packet Marking (AK-PPM) scheme, which improves the efficiency of traceback in the untrusted MANET environment.

9 Conclusion

In this paper, we made the first effort to quantitatively analyze the impacts of node mobility, attack packet rate, and path length on the traceability of well-known IP traceback schemes. We then presented an K-PPM and an AK-PPM scheme for source attribution in trusted and untrusted MANET environments, respectively. The proposed schemes can improve the efficiency of source identification and forwarding path reconstruction.

10 Acknowledgments

This work was supported in part by the NS-CTA grant from the Army Research Laboratory (ARL), the U.S. NSF CAREER 0643906, and the U.S. NSF Grant No. IIS-0324835. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARL or NSF.

References

1. Snoeren, A.C., Kohno, T., Savage, S., Vahdat, A., Voelker, G.M.: Collaborative research: Nets-find: Privacy-preserving attribution and provenence. Technical report, University of California, San Diego and University of Washington (2010)
2. Hunker, J., Hutchinson, B., Margulies, J.: Role and challenges for sufficient cyber-attack attribution. Technical report, Instituite for Information Infrastructure Protection (2008)
3. Mirkovic, J., Reiher, P.: A taxonomy of ddos attack and ddos defense mechanisms. SIGCOMM Comput. Commun. Rev. **34** (2004) 39–53
4. Ye, F., Luo, H., Lu, S., Zhang, L.: Statistical en-route filtering of injected false data in sensor networks. In: Proc. of Infocom. (2004)
5. Wang, X., Govindan, K., Mohapatra, P.: Provenance-based information trustworthiness evaluation in multi-hop networks. In: Proc. of GLOBECOM'10. (2010)
6. Dean, D., Franklin, M., Stubblefield, A.: An Algebraic Approach to IP Traceback. ACM Trans. on Information and System Security **5** (2002) 119–137
7. Snoeren, A., Partridge, C., Sanchez, L., Jones, C., Tchakountio, F., Kent, S., Strayer, W.: Hash-Based IP traceback. In: Proc. of the ACM SIGCOMM. (2001) 3–14
8. Song, D.X., Perrig, A.: Advanced and authenticated marking schemes for IP traceback. In: IEEE Infocom '01. (2001) 878 – 886

9. Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Network support for IP traceback. ACM Trans. on Networking **9**(3) (June 2001) 226–237
10. Sung, M., Xu, J., Li, J., Li, L.: Large-scale ip traceback in high-speed internet: practical techniques and information-theoretic foundation. IEEE/ACM Trans. Netw. **16** (2008) 1253–1266
11. Jeong, J., Guo, S., Gu, Y., He, T., Du, D.: TBD: Trajectory-Based Data Forwarding for Light-Traffic Vehicular Networks. In: ICDCS'09. (2009) 743 – 757
12. Thing, V., Lee, H.: Ip traceback for wireless ad-hoc networks. In: Proc. of Vehicular Technology Conference (VTC2004-Fall). (2004)
13. an Huang, Y., Lee, W.: Hotspot-based traceback for mobile ad hoc networks. In: Proc. of WiSec '05. 43–54
14. Kim, Y., Helmy, A.: SWAT: Small world-based attacker traceback in ad-hoc networks. In: Proc. of MobiQuitous'05. (2005) 85–96
15. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: Proc. of the ACM conference on Computer and communications security'03. (2003) 52–61
16. Du, W., Deng, J., Han, Y., Varshney, P.: A pairwise key pre-distribution scheme for wireless sensor networks. In: Proc. of CCS'03. (2003) 42–51
17. Blundo, C., Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Proc. of CRYPTO'92. (1993) 471–486
18. Zhu, S., Xu, S., Setia, S., Jajodia, S.: LHAP: a lightweight network access control protocol for ad hoc networks. J. of Ad Hoc Networks **4** (2006) 567–585
19. Sourcefire, Inc.: Snort (http://www.snort.org/)
20. Marti, S., Giuli, T., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. Proc. of MobiCom'00 (2000) 255–265
21. Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical network support for ip traceback. SIGCOMM Comput. Commun. Rev. **30** (2000) 295–306
22. Sadagopan, N., Bai, F., Krishnamachari, B., Helmy, A.: Paths: analysis of path duration statistics and their impact on reactive manet routing protocols. In: Proc. of MobiHoc '03. (2003) 245–256
23. Feller, W.: An Introduction to Probability Theory and Applications. Volume 1. John Wiley & Sons Publishers, (New York NY): , 3rd ed., ; Vol.2, 2nd ed., 1971 (1968)
24. Kuzmanovic, A., Knightly, E.W.: Low-rate tcp-targeted denial of service attacks. In: Proc. of SIGCOMM 03. (2003) 7586
25. Adler, M.: Tradeoffs in probabilistic packet marking for IP traceback. In: Proc. of STOC'02. (2002) 407–418
26. Goodrich, M.: Efficient packet marking for large-scale IP traceback. Proc. of the 9th ACM CCS conference (2002) 117–126
27. Ye, F., Yang, H., Liu, Z.: Catching "moles" in sensor networks. In: Proc. of ICDCS '07. (2007) 69–
28. Stajano, F., Anderson, R.: The resurrecting duckling: security issues for ubiquitous computing. Computer (2002) 22 – 26
29. John E Kobza, S.H.J., Vaughan, D.E.: A survey of the coupon collectors problem with random sample sizes. Methodology and Comp. in Applied Probability **9** (2007) 1387–5841
30. Sellke, T.M.: How many iid samples does it take to see all the balls in a box? The Annals of Applied Probability **5** (1995) 294–309
31. Zarai, F., Rekhis, S., Boudriga, N., Zidane, K.: Sdppm: An ip traceback scheme for manet. In: Proc. of ICECS'05. (2005) 1–4
32. Kim, I.Y., Kim, K.C.: A resource-efficient ip traceback technique for mobile ad-hoc networks based on time-tagged bloom filter. In: Proc. of ICCIT'08. (2008) 549–554
33. Hungyuan Hsu, Sencun Zhu, A.H.: A hotspot-based protocol for attack traceback in mobile ad hoc networks. In: Proc. of ASIACCS'10. (2010) 333–336
34. Thomas W. Doeppner, Philip N. Klein, A.K.: Using router stamping to identify the source of ip packets. In: Proc. of CCS'00. (2000) 184–189