

# Privacy Preserving Communication in MANETs

Heesook Choi, Patrick McDaniel, and Thomas F. La Porta

Networking and Security Research Center

Department of Computer Science and Engineering The Pennsylvania State University

E-mai:{hchoi,mcdaniel,tlp}@cse.psu.edu

**Abstract**—Mobile ad hoc networks often support sensitive applications. These applications may require that user’s identity, location, and correspondents be kept secret. This is a challenge in a MANET because of the cooperative nature of the network. In this paper, we propose a privacy preserving communication system (PPCS) for MANETs. It consists of three components: random node pseudonyms, dynamic flow pseudonyms, and resilient packet forwarding. Random node and flow pseudonyms conceal the linkability of identity and location, the real identity of a source and destination, and correspondents. The resilient traffic forwarding provides strong resistance against a range of traffic analysis and eavesdropping attacks. We present a thorough analysis of the security of PPCS against passive internal attackers, provide a qualitative discussion on its strength against external attackers, and characterize its performance tradeoffs.

## I. INTRODUCTION

Mobile nodes in MANETS cooperate to forward traffic on behalf of each other. In performing this function, most nodes must disclose information about themselves such as their identifiers, neighbors and destination of their traffic. Many systems also require nodes to disclose their location to support routing. In short, nodes must advertise a profile of their identity and location to participate in existing MANETs.

The goal of a privacy preserving protocol is to prevent exposure of precisely this kinds of information. Hence, supporting privacy in MANETs is by enormously challenging. Ideally, a node should be able to keep its identity, its location and its correspondents private, i.e., remain *anonymous*. Any solution providing anonymity must face a myriad of issues: it must overcome the broadcast nature of wireless environments (which enables eavesdropping) and operate under often tight resource constraints. Past “wired world” privacy solutions do not map well to MANETs because of the processing requirements they place on the nodes. Simple solutions like simple packet encryption are also largely ineffective because of ease of traffic analysis in broadcast media.

In this paper we present an anonymous communication system for MANETS that prevents linkage between identity and location and maintains anonymity of a source and destination. We call our solution a privacy preserving communication system (PPCS). It has three components: *dynamic random node pseudonyms* (RNI), *dynamic flow pseudonyms*, and *resilient packet forwarding*.

RNI prevents linkage between node location and identity. A node uses a random pseudonym to route packets and communicate with neighbors, and updates its random pseudonym after a random interval. Neighboring nodes cannot differentiate

an updated random pseudonym from a new node entering a neighborhood due to mobility.

We use dynamic flow pseudonyms to conceal the source and destination of a flow, and to prevent long term correlation of traffic within the same flow. This approach provides for low cost flow identifier reassignment by exploiting associations between the source and destination and the use of cryptographic primitives.

To hinder traffic analysis of a single path between a source and destination, we use a resilient packet forwarding scheme. The scheme consists of multi-path random forwarding (*MPRF*), efficient hop-by-hop wrapping (*Hint*), and random TTLs (*RTTL*). MPRF establishes multiple paths and distributes traffic over these paths obscuring the end-to-end path and relationship between packets. *Hints* provide an efficient transformation of packets on each link such that incoming packets and outgoing packets from a node cannot be directly correlated. With the *RTTL*, an intermediate node en route may not determine its position on the path from a source to a destination.

Through analysis and simulation, we show that PPCS provides anonymity with a small tradeoff of performance.

This paper is organized as follows: Section II-A reviews previous work on anonymity in Internet and MANETs. Section III examines passive attacks by internal compromised nodes and eavesdroppers. Section IV proposes an anonymous communication system (PPCS). Section V inspects the effectiveness of an adversary in PPCS. In Section VI, we evaluate the performance impact of PPCS. In Section VII, we discuss the trade-offs of PPCS.

## II. BACKGROUND

### A. Related Work

A great deal of previous research has focused on providing confidentiality, integrity, and authenticity of data in MANETs, but anonymity remains an open problem. In this section, we review the existing anonymous communication systems designed for the Internet and MANETs. Pfitzman and Hansen [21] define general terminologies of anonymity. In their article, anonymity is defined as “state of being not identifiable within a set of subjects, the anonymity set.” Source anonymity is defined that an adversary cannot discover who is generating packets. Destination anonymity is similarly defined. Unlinkability of the source and destination is defined that an adversary cannot discover the relationship between the source and destination, i.e., who is communicating with whom.

In the Internet, Chaum's [4] pioneering anonymity solution introduces a mix or a series of mixes (mix network) into a network for hiding communicating endpoints [8], [11]. A source selects the route (set of mixes) and encrypts data packets with the public key of each mix in reverse order (from last mix to the first mix). Each mix peels off one layer by decrypting the received packet with its private key and forwarding it to the next hop. The last mix processes the packet in the same way and transmits it to the final destination.

Onion routing [23] is built on a mix-net approach. An onion consists of next hop information and an onion for the next hop. For example, if there are  $k$  onion routers ( $N_1, N_2, \dots, N_k$ ) on the path, the source wraps the packet as follows: Packet =  $E_{p1}(N_2, E_{p2}(N_3, E_{p3}(\dots E_{pk}(D, C), \dots)))$ . Each intermediate onion router decrypts the received message with its private key to get the next hop and onion for the next hop. The last onion peels off its layer and transmits the encrypted data to the destination. Tor [6] extended onion routing with features that provide forward secrecy.

Mix-nets are not applicable to MANETs. This is because the resource demands of the underlying public key operations are too expensive for mobile nodes with energy and computation limitations. Moreover, with high mobility, it is not easy to maintain the full path from the source.

In Crowds [24], groups of users (called *crowds*) cooperate to ensure client anonymity in web systems, e.g., web-browsing. *Jundos* run by each client decide randomly if they should relay the packet to another jundo or transmit it to the web server directly. All users in the group share their symmetric keys to encrypt the relayed packet. Hordes [18] is based on Crowds and proposes to use multicast routing to provide initiator anonymity. Brent [27] proposes receiver anonymity based on incomparable public keys and multicast. In MANETs, however, the maintenance cost of multicast is known to be high.

Most solutions proposed for the Internet use a proxy function (Mix, Jundo, and Onion Router) to provide anonymity. In MANETs, Jian et al. [9], [10] propose a dynamic mix method that accommodates dynamic topology changes that are characteristic of MANETs. Boukerche et al. [3] propose to make only trustworthy nodes participate in routing to protect anonymity. Blaze et al. propose WAR [2], in which anonymous routing is combined with a key distribution protocol and an onion routing structure. However, in MANETs, it is not feasible to form a set of proxy functions since mobile nodes all play an equal role. In civilian applications of MANETs, in particular, mobile nodes may not cooperate to play the larger role of a proxy.

J. Kong and X. Hong [12] apply MIX-Net to MANETs by using symmetric key cryptography to provide anonymity. This approach uses cryptographic trapdoor within a broadcast message to hide the identifiers of local intermediate nodes and the destination. However, in a situation in which adversaries are located on each link, they may simply monitor the transmission to determine who is broadcasting and how many packets are being broadcast. After that, adversaries may combine the collected information to detect the source, destination, and source-destination pair.

Recently, Zhang et al. proposed MASK [29] in which nodes use a random node identifier (*pseudonym*). The random identifier is generated in a Trusted Authority (TA) and loaded before nodes are deployed. Intermediate nodes use virtual multipaths to relay packets to next hops randomly. The destination unicasts back only one reply toward a source. Hence, traffic flows may have the same initial or terminating hops on a path. In MASK, the destination identifier is transmitted in the clear during the route discovery, which violates anonymity.

## B. Network Model

We assume that wireless interface between nodes is bidirectional, i.e., if node  $i$  hears the transmission of node  $j$ , then node  $j$  is also able to hear node  $i$ .

In MANETs, a mobile node has unique layer 2 MAC address and layer 3 node identifier which are assigned during the network deployment. Especially, the layer 3 node identifier is used for end-to-end communication. We assume that these layer 2 and 3 identifiers are 32 bits and 48 bits respectively, similar to the real network environment.

We assume that there exists a key management service to establish pair-wise keys between nodes, and the source and destination establish symmetric keys prior to communications. Such services are well studied in the general security community [25], [19], [15], and their design and anonymity is explicitly outside the scope of this work. The source and destination know each other's real identifier. Non-compromised nodes in the network do not disclose any information to compromised nodes.

We use the following notation throughout:

- $S$ : Identifier of a source
- $D$ : Identifier of a destination
- $M$ : End-to-end message
- $n$ : Average number of neighboring nodes in transmission range
- $P_{Si}$ : Source  $S$ 's  $i$ -th flow pseudonym
- $P_{Di}$ : Destination  $D$ 's  $i$ -th flow pseudonym
- $K_{ij}$ : Symmetric key established between node  $i$  and  $j$
- $E_{K_{SD}}(\cdot)$ : Encryption with a key  $K_{SD}$
- $D_{K_{SD}}(\cdot)$ : Decryption with a key  $K_{SD}$

## III. THREAT MODEL

In this section, we examine attacks on anonymity in MANETs. We adopt Diaz et al.'s [1] classification of adversaries based on the following characteristics: Internal-External, Passive-Active, and Local-Global. An internal adversary is a compromised node in the network. We define an internal adversary as a node that is compromised and on the routing path. An external adversary is a compromised node not on the path, or an external node not directly participating in the MANET, i.e., only eavesdrops traffic between MANET nodes, but does not perform an active attack.

An active adversary may alter traffic (inject, drop, or modify packets) while a passive adversary only eavesdrops on the communication and collects private information. *This paper only considers passive attacks.* A local adversary can see and launch attacks in a limited range. A global adversary covers all

the path or the network. A set of colluding local adversaries may form a global adversary by sharing information.

Traffic analysis is often used to subvert anonymity [1], [26], [22]. In this attack, adversaries monitor packet transmission to infer important information such as a source, destination, and source-destination pair. We consider the following traffic analysis attacks in this work:

- 1) **Packet Tracing Attack:** A packet may be traced from source to destination by eavesdropping the transmission of the same packet as it traverses the network. Note that the adversary need not be able to recover the packet content to infer valuable information, i.e., the source and destination of the flow.
- 2) **Packet Counting Attack:** Eavesdropping nodes collaborate to discover a path by overhearing and “counting” packets. Figure 1 shows an example. Node  $S$  generates 150 packets in a fixed interval. Node 6 overhears that node 1 transmits 150 packets. Node 7 also overhears that node 3 and 4 transmits 150 packets in the same interval. Eavesdropping nodes 6 and 7 knows the path from  $S$  to  $D$  ( $S$ -1-3-4- $D$ ). Thus, it is not difficult for a global passive eavesdropper to discover a flow by simply tallying packets.

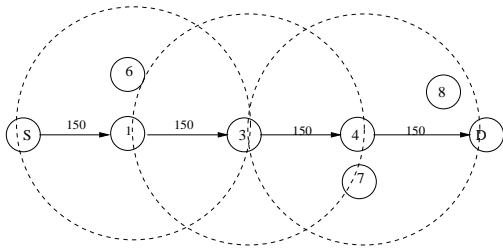


Fig. 1. Eavesdropping by neighboring nodes (node 6, 7, and 8)

- 3) **Timing Attack:** An adversary may analyze the correlation between packets passing through nodes to discover a flow, called timing attack [16]. Suppose that there are only two eavesdroppers (6 and 8) in Figure 1, and node  $S$  is sending packets to node  $D$ . Node 6 eavesdrops packet transmission, and analyzes the correlation between packets, using a technique such as Time Series. Node 8 also eavesdrops packet transmission between 4 and  $D$  to do the same. Nodes 6 and 8 compare information they collect and find out the source-destination pair.

Another attack exploits the packet time-to-live (TTL) to identify the destination. A The TTL is a value set by a source to limit the number of hops a packet takes in the network. Every intermediate node decreases the TTL by 1 before it forwards the packet. A node discards a packet if it is not the destination and the TTL of the packet becomes zero. Obviously, close to the destination, an adversary can loosely identify the potential destinations by looking at the TTL.

To discover the source-destination pair, adversaries may also try to discover path information (nodes en route). Since some of control information is open to intermediate nodes on the path, an internal adversary has access to the control

information in the clear. When an internal adversary receives a packet, the compromised node forwards it as other honest nodes. Figure 2 illustrates an example. Node 3 is a compromised node, and 3, 5, and 4 are on the path. Node 3 transmits a modified packet  $M1$  to 5, and attempt to discover which node is the next hop beyond node 5. In the existing MANET routing protocols, data packets are transmitted without any modification. Node 3 may easily trace the next hop of the packet by overhearing the transmission by node 5. To preserve anonymity, any information that may contain flow information should not be sent in the clear.

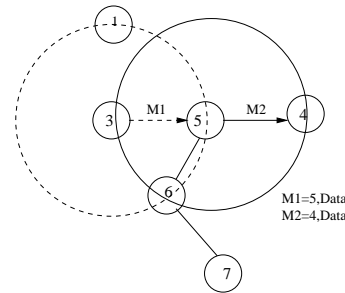


Fig. 2. Local View of Transmission in MANET

Mobile nodes may obtain their location information from global positioning system (GPS) or other similar techniques. If a node knows the identifiers of neighboring nodes, it also may estimate the location of neighbors, using transmission power. In MANETs, there are several location based services such as location-based routing, performance optimization, and clustering. An adversary may also use location information to launch various attacks, by tracing an object’s location. Therefore, dissociation of location and identity is an important issue.

## IV. PRIVACY PRESERVING COMMUNICATION

### A. Overview

Our goals are to preserve the anonymity of a node and its correspondents and to provide unlinkability of a node to its location.

To achieve unlinkability of the location and a node we propose an efficient and simple identification scheme, called Random Node Identification (RNI). A mobile node has two addresses: a layer 2 address (MAC address) and a layer 3 address (node identifier). In RNI, every node uses two random pseudonyms: one each for MAC address and node identifier. These random pseudonyms are used for discovering neighbors and routing. With the RNI, even if a source neighbors a destination, it may not discover that a destination is within transmission range.

In traditional MANET routing protocols, source and destination addresses are also used for identifying a flow. We propose to use flow pseudonyms between communicating endpoints to hide the source and destination. Initially, a flow pseudonym is generated with the symmetric key and identifier of a source and destination so that other nodes may not discover the original identity information. The flow pseudonym appears as a random value to intermediate nodes. However, if a flow

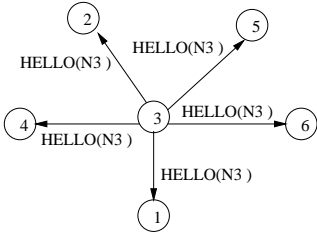


Fig. 3. Neighbor Relationship Setup with Random Node Identifiers.

persists for an extended time, it may become susceptible to traffic analysis attacks. Thus, after a random amount of time the flow pseudonym is changed.

The random node and flow pseudonym methods described above provide anonymity of the source and destination identity, location privacy, and afford some protection against simple traffic analysis attacks. An adversary, however, may still attempt to discover the end-to-end communication by using more sophisticated traffic analysis attacks discussed in Section III. If traffic traverses a single path, it is easy for an adversary to learn the routing path. To combat this threat we propose a resilient traffic forwarding scheme which consists of multi-path random forwarding (MPRF), Hint, and random TTL (RTTL). MPRF allows intermediate nodes to randomly select a next hop from a list of candidates. This way, every packet may take a different path to obscure the traffic flow from an adversary. With Hint each packet is efficiently transformed on each hop to prevent eavesdroppers from correlating packets entering and exiting a node. The location of a node on a path may be revealed by the Time-To-Live (TTL) field within data packets which expresses a hop count. We propose to use a random TTL (RTTL) to mask the position of a node on a forwarding path. In the following subsections, we present each of these schemes in detail.

### B. Random Node Identification (RNI)

We devise a scheme that dissociates an identifier from location information. This requires node identity and location to be unlinkable and untraceable. Even in local communication, any information, if possible, should be not released.

To achieve the unlinkability of location and identity, we propose an efficient and simple identification scheme called Random Node Identification (RNI). Every node in the network generates a random node pseudonym and advertises the node pseudonym via a message such as a HELLO message in AODV [20].

Neighboring nodes know each other through their random pseudonyms. The pseudonym is used for routing and communicating with neighboring nodes. Intermediate nodes use pseudonyms of neighbor nodes to maintain next hop information for a flow.

Figure 3 illustrates the protocol for nodes to setup neighbor relationships using their pseudonyms. For example, node 3 advertise itself as  $N3$ , instead of its real identifier 3.

Each node changes its pseudonym after a random interval, to prevent an adversary from learning the location of a node. After changing its pseudonym, a node starts advertising itself

with the new pseudonym. The MAC address pseudonym is also changed to prevent other nodes from matching a fixed MAC address to a new pseudonym. The protocol to change a pseudonym is the same as for an update due to mobility. Therefore, nodes cannot distinguish an updated pseudonym from a new neighbor. Since the source and destination do not know each other's pseudonym, the communication between a source and destination does not disclose the location of either party to the other.

Due to the randomness and independence of the new and old pseudonyms, an adversary cannot trace the changes of node pseudonyms. One risk with this approach is that identifier collisions, in which two nodes choose the same pseudonym, might occur. However, the probability that two nodes generate the same MAC address pseudonym (48bits) and same node identifier pseudonym (32bits) is very low,  $(\frac{1}{2^{48}} \frac{1}{2^{32}})^2$ .

### C. Dynamic Flow Pseudonyms

Traditional MANET routing protocols require each control and data packet to contain the source and destination addresses to find a route and identify a flow. With this general approach, an adversary close to the source or destination, or an adversary on the communication path between the two, will be able to link the correspondents, and perhaps learn their location.

To define a flow without releasing identity, we propose a dynamic and random flow pseudonym based on a forward chaining. A random flow pseudonym replaces the source and destination address in the packets. A flow pseudonym is defined for the forward and reverse path. It is constructed from the shared symmetric key between the source and destination, and the nodes identifiers.

An initial reverse flow pseudonym,  $P_{S0}$ , is generated by using the symmetric key and real identifier  $S$ ,  $P_{S0} = f_{K_{SD}}(S)$ . The forward flow pseudonym is  $P_{D0} = f_{K_{SD}}(D)$ .

Since the source knows the real identifier ( $D$ ) of a destination, it generates a flow pseudonym for the forward direction,  $P_{D0} = f_{K_{SD}}(D)$ . A source broadcasts a RREQ packet which contains these flow pseudonyms. The RREQ contains  $\langle RREQ, P_{S0}, P_{D0}, E_{K_{SD}}(.) \rangle$ . Intermediate nodes receive a RREQ packet and check if they are the destination by attempting to open the "trapdoor [12]." If they are not the destination, they add a routing table entry for the backward flow which is identified by the flow pseudonym  $P_{S0}$  in the RREQ, before forwarding. A destination receives a RREQ and determines that it is the destination by checking the received  $P_{D0}$ . Either a source or a destination may decide to change the flow pseudonym at anytime.

Note that the trapdoor check only occurs when processing the RREQ message; once the flow has been routed, the check is not required for forwarding subsequent packets. Because each node must perform the check, however, it is desirable for the trapdoor check to be efficient. To enable this efficiency we construct pseudonyms using a forward chaining as follows:

$$\begin{aligned}
 P_{S0} &= f_{K_{SD}}(S) \\
 P_{S1} &= f_{K_{SD}}(P_{S0}) \\
 &\dots \\
 P_{Sn} &= f_{K_{SD}}(P_{Sn-1})
 \end{aligned} \tag{1}$$

The new flow pseudonym is generated from the symmetric key and the previous flow pseudonym.  $f$  is a cryptographic keyed one-way function (HMAC) which takes  $k$  bits input and outputs  $k$  bits. A node that decides to change the flow pseudonym initiates a route discovery with the new flow pseudonym. Since other nodes and adversaries do not have the symmetric key  $K_{SD}$ , they cannot generate a new flow pseudonym, nor do they know the source and destination.

The use of forward chaining enables an efficient trapdoor checking mechanisms to be employed. Nodes construct a tree of flow pseudonyms in the initialization phase as they are deployed. The tree is a general sorted binary tree. At a node  $i$ , it is composed of  $P_{i0} = f_{K_{ij}}(i)$  for all possible nodes  $j$ . Once a path is established between the source and destination, a node updates the corresponding tree value with the next flow pseudonym. Since a next flow pseudonym is generated with the current flow pseudonym and symmetric key based on a forward chaining, a node can maintain a tree of next flow pseudonyms. A node on the tree consists of a next flow pseudonym, peer identifier, and symmetric key, i.e.  $\langle P_{ik}, ID_j, K_{ij} \rangle$ , where  $i$  is the local node and the peer node is  $j$ .

For a RREQ packet having a flow pseudonym  $P_{D0}$ , an intermediate node searches its tree. If it finds an entry, it is a destination.

#### D. Resilient Packet Forwarding

There are many feasible traffic analysis attacks that may be attempted by eavesdropping nodes (i.e., external attackers). To combat these attacks we propose a resilient traffic forwarding scheme which is composed of multi-path random forwarding (MPRF), Hint, and random TTL (RTTL).

**Multi-Path Random Forwarding (MPRF):** In a relatively stable network (mild traffic load and low mobility), a path between a source and destination may be used for an extended period of time. This type of path, in particular, is susceptible to a traffic analysis attack. To thwart attacks on a single path MPRF establishes multiple paths between the source and destination. For each packet, an intermediate node en route randomly selects a next hop from its local list of possible next hop nodes, and forwards the packet to the selected node. Thus, a path that a packet takes is decided dynamically at each intermediate node.

Multi-path routing protocols have been proposed for improving reliability and providing quality of service in ad hoc networks [14], [17], [28]. These multi-path routing protocols establish link/node disjoint paths to distribute traffic to avoid congestion. Node/link disjoint paths are vulnerable to traffic analysis attacks. Collaborating eavesdroppers may easily obtain exact packet counts and reconstruct the end-to-end paths. To resolve these vulnerabilities and establish a sufficient number of multiple paths, we relax the node/link jointness condition present in most multi-path routing protocols. By allowing non-disjoint paths, MPRF diffuses traffic in an irregular manner making traffic analysis more difficult, i.e., requiring a larger number of colluders.

**Hint:** Although a packet is encrypted by a source, if the encrypted packet is transmitted without any modification on each link, it is vulnerable to traffic analysis attacks. Several approaches have been proposed to transform packets on a hop-by-hop basis to combat these attacks.

ANODR [12] uses a route pseudonym to wrap a data packet on each link using encryption techniques. Packets are broadcast to all neighbors. Neighbors that receive a packet attempt to decrypt it. Since each neighboring node has a list of route pseudonyms, it may try to decrypt the received packet with all possible route pseudonyms. If there does exist a match, it will process the packet according to the mapping table. Otherwise, the packet will be discarded. The processing required in every node may be high which is a concern in a MANET. In MASK [29], each intermediate node encrypts a packet using a symmetric key shared with its next hop. The transmitted packet is of format  $\langle \text{next-LinkID}, \text{MASK payload} \rangle$ . For a fixed amount of time, the linkID is used for relaying packets between the two neighboring nodes thus allowing an eavesdropper a chance of reconstructing a path.

To make the hop-by-hop transformation more efficient and anonymous, we propose an HMAC [13] based scheme, called *Hint*. An intermediate node randomly selects a next hop node according to MPRF. It encrypts a packet using a shared key with the selected node and computes an HMAC over the encrypted packet. This HMAC result is called the **Hint**. Then it broadcasts the packet which consists of the Hint and encrypted packet. The following shows how each intermediate node transforms the packet and generates Hint.

$$\begin{aligned}
 N_S: & \\
 & C = E_{K_{SD}}(M) \\
 & MC = \langle P_{S0}, P_{D0}, TTL, C \rangle \\
 & EL_S = E_{K_{N_S N_1}}(MC) \\
 & \text{Hint} = \text{HMAC}(K_{N_S N_1}, EL_S) \\
 & \text{Broadcast} \langle \text{Hint}, EL_S \rangle \\
 N_1: & \\
 & MC = D_{K_{N_S N_1}}(EL_S) \\
 & EL_1 = E_{K_{N_1 N_2}}(MC) \\
 & \text{Hint} = \text{HMAC}(K_{N_1 N_2}, EL_1) \\
 & \text{Broadcast} \langle \text{Hint}, EL_1 \rangle \\
 N_2: & \\
 & MC = D_{K_{N_1 N_2}}(EL_1) \\
 & EL_2 = E_{K_{N_2 N_D}}(MC) \\
 & \text{Hint} = \text{HMAC}(K_{N_2 N_D}, EL_2) \\
 & \text{Broadcast} \langle \text{Hint}, EL_2 \rangle \\
 N_D: & \\
 & MC = D_{K_{N_2 N_D}}(EL_2) \\
 & M = D_{K_{SD}}(MC)
 \end{aligned}$$

Neighboring nodes check if a received packet is for a flow which they serve by simply computing the HMAC for the received packet. If the check results in success, it decrypts the received packet with the corresponding key and forwards it according to MPRF. The HMAC calculation takes a few micro seconds as shown in [5]. Only the corresponding local receiver decrypts the packet. If  $D(\cdot)$  denotes the overhead for packet decryption, and  $n$  is the average number of neighbors in transmission range, Hint reduces the computation at a node from  $\frac{1}{2}n^2 D(\cdot)$  to  $\frac{1}{2}n^2 \text{HMAC}(\cdot)$  when compared to schemes that encrypt and broadcast a packet.

Due to the transformation on each link combined with broadcast transmission, eavesdroppers are not able to learn the relationship between incoming and outgoing packets of a node. Although a compromised node en route may see several control fields like TTL in clear text, it cannot discover which node will be the next hop of its neighboring next hop. For each traffic flow, since there is no relation between flows, an adversary may have difficulty in discovering the flow. Furthermore, when a destination receives a packet, it broadcasts a random packet as a response, hiding its role from neighboring nodes. This random packet is not distinguishable from a transformed packet by Hints. Neighboring nodes may discard the packet since it does not match.

During route discovery, Hints are used to transform a RREP in the same way. Otherwise, an adversary may discover a route through tracing RREP messages. A destination or an intermediate node having a routing entry replies to a RREQ packet. It encrypts a RREP packet with a key shared with a next hop toward a source, and computes the Hint for the encrypted RREP. It broadcasts the RREP.

**Random Time-To-Live (RTTL):** The TTL field is used for discarding packets which have not found a destination and circulated through the network. In MANETs, the TTL is set to the length of a path by a source node. Each node on the path decreases the value by 1. Thus, the TTL value reveals the position of a node on a path from a source or a destination. The receiver anonymity set may be reduced to a set of nodes neighboring a compromised node from a set of all possible receivers.

To prevent compromised nodes from learning their position on a path, we propose a Random Time-To-Live (RTTL). A source node generates a random value and sets the TTL field with the sum of this random value and path length, RTTL. The RTTL should be less than the maximum hop count (Network diameter). The source includes the initial random value in the encrypted data packet. Intermediate nodes decrease TTL field value of a packet by 1 as they do in the normal packet forwarding. This TTL field does not release the position of a node due to the random value. A destination decrypts the received packet and checks if the received RTTL is valid.

## V. SECURITY ANALYSIS

In Section III, we presented a classification of attackers. In this section, we initially characterize the anonymity provided by PPCS against attacks by internal compromised nodes analytically. We then argue informally about the anonymity provided by our system against eavesdropping attacks.

### A. Internal Attackers

In this subsection we examine the effectiveness of PPCS against collaborating adversarial nodes. Internal compromised nodes learn information by eavesdropping and by being on the forwarding path of a flow. As discussed in the next subsection, limited information is gained by eavesdropping because of the packet transformations and the local broadcasts.

Source and destination information is not directly disclosed to any other nodes on the path. Intermediate nodes on the path, however, can see the flow pseudonym and TTL field of a packet. Intermediate nodes also have previous and next hop nodes of a packet on the routing path. Using this information, the compromised nodes on a path collude to make an educated guess as to the source and destination of a flow.

To characterize the probability that a set of internal compromised nodes collaborate on discovering anonymity we first derive a general equation which can be applied to each case of anonymity (source/destination and communicating pair). Note that because protocol behavior is unique to its design and the environment in which it is run, we ignore their affect on anonymity. For example, knowing the timing characteristics of a RPC call may provide some additional information upon which to infer correlations between the sources and destinations.

The following notation is used in the remainder of our analysis.

- $N$ : Total number of nodes
- $C$ : Number of compromised nodes in the network
- $L$ : Average path length
- $ND$ : Number of uncompromised nodes disclosed by intermediate compromised nodes en route
- $NP$ : Number of intermediate nodes on multiple paths established between the source and destination
- $NC$ :  $(N - C) - ND$
- $p$ : probability that a node is compromised
- $P_{f,s}=P_{l,r}$ : probability that the first/last hop node guesses a source/destination correctly, respectively
- $P_{i,s}=P_{i,r}$ : probability that an intermediate node guesses a source or a destination correctly
- $P_{i+f,l}=P_{i+l,l}$ : probability that the first/last hop node and intermediate nodes together guess linkability of the source correctly and destination
- $P_{f+l,l}$ : probability that the first and last hop nodes together guess linkability of the source and destination correctly
- $P_{i+l,l}$ : probability that intermediate nodes together guess linkability of the source and destination correctly

Let  $P(A = s)$  and  $P(A = r)$  denote the probability that an adversary discovers a source or a destination. Because the values are the same, we discuss the probability  $P(A = s)$  below. Let  $P(A = (s, r))$  denote the probability that an adversary discovers the source and destination pair.

1) *Generalization:* We assume that the probability of a compromised node being able to exploit a vulnerability is dependent on its position on a path. In particular, the first and last hop nodes on a path may have a higher probability of finding a source or destination, respectively, than an intermediate node on the path depending on the characteristics of the security solution.

To this end we derive the probability of four cases of node compromise as Table V-A.1.

We determine the probabilities of  $P(CH)$ ,  $P(HC)$ ,  $P(CC)$ , and  $P(HH)$  for a path that has  $k$  compromised nodes

TABLE I  
CLASSIFICATION OF NODE COMPROMISE

<i>CH</i>	the first hop of a source is compromised and zero or more other compromised nodes are on the path, but not the last hop.
<i>HC</i>	the last hop is compromised and zero or more other compromised nodes are on the path, but not the first hop node.
<i>CC</i>	the first and last hop nodes are compromised, as well as zero or more compromised nodes on the path.
<i>HH</i>	the first and last hop nodes are not compromised nodes, but one or more compromised nodes are on the path

in each case.

$$\begin{aligned}
P(CH) &= (1-p)^{L-k} p^k \binom{L-2}{k-1} \\
P(HC) &= (1-p)^{L-k} p^k \binom{L-2}{k-1} \\
P(CC) &= (1-p)^{L-k} p^k \binom{L-2}{k-2} \\
P(HH) &= (1-p)^{L-k} p^k \binom{L-2}{k}
\end{aligned}$$

Let  $P_{CH}|P_{HC}|P_{CC}|P_{HH}$  denote the probability that an adversary discovers target anonymity in each case.

$$\begin{aligned}
P_{CH} &= P(A|CH)P(CH) \\
P_{HC} &= P(A|HC)P(HC) \\
P_{CC} &= P(A|CC)P(CC) \\
P_{HH} &= P(A|HH)P(HH)
\end{aligned}$$

In these equations,  $P(A|X)$  is the probability that anonymity is discovered given that the compromise scenario  $X$  has occurred.

The probability that an adversary discovers target anonymity is defined

$$P(A) = P_{CH} + P_{CC} + P_{HC} + P_{HH} \quad (2)$$

This is a measure of the effectiveness of compromised nodes. In disjoint multi-paths environments, the probability that an adversary discovers anonymity is defined as follows:

$$P_m(A) = 1 - (1 - P(A))^R \quad (3)$$

where  $R$  is the number of disjoint paths established between the source and destination.

2) **Source/Destination Anonymity:** Compromised internal nodes collaborate to determine a source using explicit information such as the flow pseudonym, TTL value, and next and previous hop nodes.

Let us suppose that there is more than one compromised node on a routing path. These nodes conspire to discover a source of traffic.  $P_{f,s}$  and  $P_{i,s}$  are the probabilities that the first hop and intermediate nodes guess a source, respectively.

The probability  $P(A = s)$  is

$$\begin{aligned}
P(A = s) &= P_{CH} + P_{HC} + P_{CC} + P_{HH} \\
&= P_{f,s} \sum_{k=1}^{L-1} (1-p)^{L-k} p^k \binom{L-2}{k-1} \\
&\quad + P_{f,s} \sum_{k=2}^{L-2} (1-p)^{L-k} p^k \binom{L-2}{k-2} \\
&\quad + P_{i,s} \sum_{k=1}^{L-1} (1-p)^{L-k} p^k \binom{L-2}{k-1} \\
&\quad + P_{i,s} \sum_{k=1}^{L-2} (1-p)^{L-k} p^k \binom{L-2}{k}
\end{aligned} \quad (4)$$

The first two terms correspond to the first two terms in equation 2. The last two terms correspond to the last two terms in equation 2. Note that we do not need to account for intermediate nodes compromised in the scenarios covered by the first two terms in equation 2 because of the manner in which compromised nodes will collaborate. That is, if two nodes on a path are compromised and collaborate, they can compare the TTL field of the packets they receive and determine who is closer to the source. This is the only node that can correctly guess the source if an optimal guessing policy is used as discussed directly below.

First, consider an optimal anonymity solution in which no information is leaked. In this case a compromised node does not know its previous or next hops, or its position on a path. It only knows of other compromised nodes. In this situation, the best an adversary can do is guess the source from the set of uncompromised nodes. The probability of guessing correctly is  $\frac{1}{(N-C)}$ .

Now consider a non-ideal anonymity solution in which an adversary can identify its position on the path, but not other nodes on the path except for its direct previous and next hops. If the node is the first hop (information learned by seeing the TTL in the reverse path), it knows its previous hop is the traffic source. If a node is not the first hop on a path, its best guess is a random choice of all nodes in the network not counting the nodes it knows to be compromised or the nodes that compromised nodes can rule out as the source, such as their next hop nodes or previous hop nodes if they are not first on the path. We call this set  $U$ , which has  $NC = N - C - ND$  members. Thus the probability of an intermediate node guessing correctly is  $\frac{1}{NC}$ .

Finally, consider the situation provided with RTTL is used within PPCS. In this case an adversary knows it is on the path, but cannot tell its position on the path. Therefore, a different guessing strategy will be used. The adversaries have two choices. First, they can make a random guess of all nodes in set  $U$ , in which case their chance of guess correct is  $\frac{1}{NC}$ . A better strategy is simply to guess its previous hop as being the source. Although the adversary does not know its place on the path, it has a  $\frac{1}{L}$  chance of being the first hop node and thus guessing correctly. Even if several nodes on the path are compromised and collaborate, the only information they can learn is which adversary is closest to the source, and guess the

previous hop to that node, i.e., they will all guess the same node. The only way that the random guess strategy will be better for an individual node is if  $NC \leq L$ , i.e., the average path length is greater than the number of uncompromised nodes in the network which is an unlikely scenario.

Based on the discussion above, we assume the following three strategies to guess the source node on a path: (1) In an ideal environment, adversaries make a random guess from the set of non-compromised nodes; (2) If an adversary is on a path, and it knows its position on the path, it will guess its previous hop as the source if it is the first hop node, otherwise it will make a random choice from the set  $U$ ; (3) If an adversary is on a path, and it does not know its position on the path, it will always guess its previous hop on the path as the source.

Based on these strategies, we can now evaluate  $P_{f,s}$  and  $P_{i,s}$  and determine the impact of PPCS. In cases in which an adversary knows its position on the path,  $P_{f,s} = 1$  and  $P_{i,s} = \frac{1}{NC}$ . In cases in which an adversary does not know its position on the path, such as if RTTL is used with PPCS,  $P_{f,s} = 1$  and  $P_{i,s} = 0$ . This is because all adversaries will always guess the previous hop of a first adversary (the same guess), so in cases in which the first hop node is an adversary, all guess will be correct, and in cases in which the first hop node is not an adversary, all guesses will be incorrect.

We now extend this analysis to consider the impact of MPRF on security. PPCS establishes multiple paths between the source and destination. With the assumption that each path of the  $R$  paths is disjoint, the probability an adversary discovers a source or destination is

$$P_m(A = s) = 1 - (1 - P(A = s))^R \quad (5)$$

In a disjoint multi-path environment, intermediate nodes have only one previous and next hop nodes. Since intermediate nodes do not know their position on a path, compromised nodes have the same probability  $P_{f,s} = 1$  and  $P_{i,s} = 0$  which is used to compute  $P_m(A = s)$ .

Figure 5 (a) shows the effectiveness of compromised nodes in a disjoint multiple-path environment. An adversary has a higher probability of guessing the source in a multiple disjoint path environment since more information may be open to more compromised nodes.

However, MPRF uses multiple non-disjoint paths. Thus every intermediate node may have multiple forward and backward hops for a flow. Furthermore, the first hop node on one path may be a non-first hop node on a different path of which it is a part. As Figure 6 (b) shows, compromised node  $A$  has two previous hops (node 1 and 2). These multiple incoming links increase the number of choices for guessing, and hence reduce the probability of an adversary guessing correctly.

In Figure 4, the addition of each dotted link increases the incoming degree of the corresponding nodes (1, 4, 7, and 8). From this, we can compute the average incoming degree of a node,  $\frac{NP+i}{NP}$ , where  $NP$  is the number of nodes on disjoint multipaths and  $i$  is the number of added directed links.

Hence, the probability that an intermediate node determines a node from candidate previous hop nodes is  $\frac{NP}{NP+i}$ .  $P_{f,s}$  becomes  $\frac{NP}{NP+i}$ .  $P_{i,s}$  is still 0 since intermediate nodes beyond the first hop will always guess wrong.

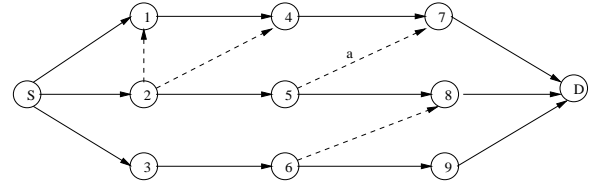


Fig. 4. Non-Disjoint Multi-Paths

Figure 5 (a) compares the probability that an adversary may guess a source in disjoint multi-path and non-disjoint multipath environments where 4 disjoint multipaths exist and the average path length is 5. This result demonstrates that MPRF in PPCS reduces the effectiveness of an adversary.

In summary, Table II shows the effect of using PPCS on the probability that intermediate and first hop nodes guess a source correctly.

TABLE II  
IMPACT OF PPCS ON PROBABILITY

Probability	Perfect Anonymity	No PPCS	PPCS (Previous Hop Policy)		
			Single Path	Disjoint Multipath	Non-Disjoint Multipath
$P_{f,s}$	$\frac{1}{N-C}$	$\frac{1}{NC}$	1	1	$\frac{NP}{NP+i}$
$P_{i,s}$	$\frac{1}{N-C}$	$\frac{1}{NC}$	0	0	0

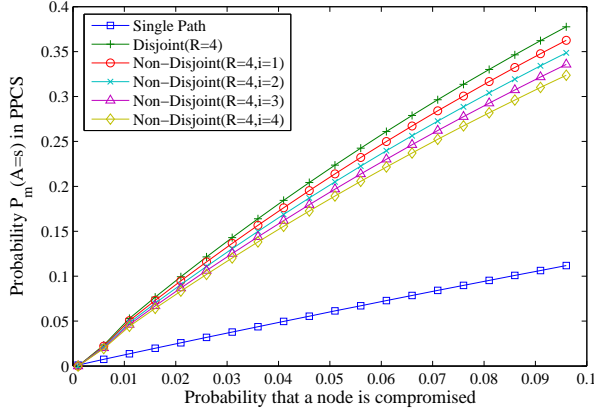
For destination anonymity, the analysis and equations are similar.

3) **Source and Destination Unlinkability:** If the path between a source and destination is known, the source and destination pair is also discovered. The probability that an adversary discovers the source and destination pair in a single path environment is

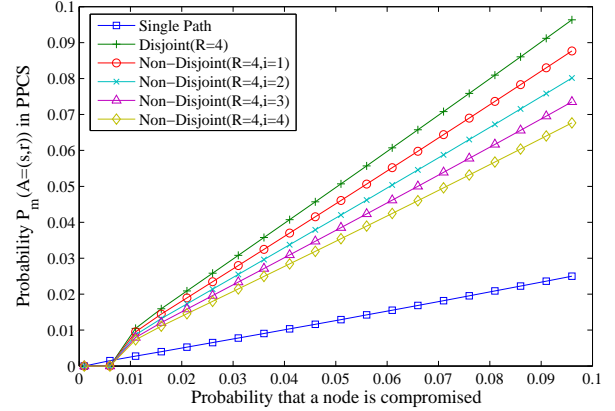
$$\begin{aligned}
P(A = (s, r)) &= P(A = (s, r)|CH)P(CH) \\
&+ P(A = (s, r)|HC)P(HC) \\
&+ P(A = (s, r)|CC)P(CC) \\
&+ P(A = (s, r)|HH)P(HH) \\
&= P_{i+f,l} \sum_{k=1}^{L-1} (1-p)^{L-k} p^k \binom{L-2}{k-1} \\
&+ P_{i+l,l} \sum_{k=1}^{L-1} (1-p)^{L-k} p^k \binom{L-2}{k-1} \\
&+ P_{f+l,l} \sum_{k=2}^{L-2} (1-p)^{L-k} p^k \binom{L-2}{k-2} \\
&+ P_{i+i,l} \sum_{k=1}^{L-2} (1-p)^{L-k} p^k \binom{L-2}{k}
\end{aligned} \quad (6)$$

$P_{*,l}$  denotes the probability that nodes en route guess the source and destination pair.

As discussed in the previous section, if an adversary knows its position on a path, the probability that the first/last hop node determines a source or a destination is 1. The probability that other intermediate nodes guess a source/destination becomes  $\frac{1}{NC}$ , since intermediate nodes know that their previous/next



(a) Source Anonymity



(b) Source and Destination Linkability

Fig. 5. Probability of an adversary

hop is not the source/destination and may guess one node of a set of possible sources/destinations. Therefore, if intermediate nodes know their position,  $P_{f+i,l}$  and  $P_{i+l,l}$  are  $\frac{1}{NC}$ ,  $P_{i+i,l}$  is  $(\frac{1}{NC})^2$ , and  $P_{f+l,l}$  is 1.

If the adversary does not know its position on a path because of RTTL, the same guessing strategy as previously discussed is used. Thus,  $P_{f+l,l}$  is 1, and  $P_{i+f,l}|P_{i+l,l}|P_{i+i,l}$  become 0.

By extending the above single path case to a disjoint multi-path, the probability of discovering the source and destination pair is

$$P_m(A = (s, r)) = 1 - (1 - P(A = (s, r)))^R \quad (7)$$

In disjoint multi-path environments, intermediate nodes have the same probability as the single path to guess the source and destination pair. Figure 5 (b) shows the probability that an adversary discovers the communicating pair in a disjoint multi-path environment.

In a non-disjoint multi-path environment, we can apply the same reasoning as for the source anonymity case to determine that  $P_{f+l,l}$  is  $(\frac{NP}{NP+i})^2$ , and  $P_{i+f,l}|P_{i+l,l}|P_{i+i,l}$  become 0.

As Figure 5 (b) shows, an adversary has a lower probability to discover the communicating pair in non-disjoint multi-path environments than disjoint multi-path environments. This verifies that MPRF of PPCS reduces the effectiveness of internal compromised nodes, while providing defense against eavesdropping attacks.

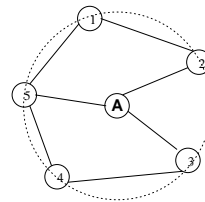
### B. Eavesdropping

Since nodes in MANETs share a common broadcast channel, they overhear all communication in transmission range. Hence, an adversary may learn information by collecting and analyzing overheard data without revealing its existence. A set of local eavesdroppers form a global eavesdropper to cover a path. They may have a dedicated communication channel to exchange information.

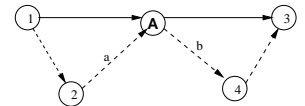
In PPCS, an intermediate node en route uses a Hint to prevent correlation between forwarded packets. The intermediate node broadcasts the transformed packet locally. The eavesdroppers may not learn which node is the local sender

and receiver of a packet, due to the local broadcasting of packets. This, combined with the fact that the packet has been transformed, limits eavesdroppers from obtaining information about the relationship between the incoming and outgoing packet of a node.

For example, in Figure 6, let us suppose that node 5 relays traffic to node 1,  $A$ , and 4 where  $A$  denotes a compromised intermediate node. Five nodes are in transmission range of node  $A$ . Node  $A$  may receive a portion of the traffic that is relayed by node 5. In PPCS, all five neighboring nodes broadcast packets for relaying, after transforming using Hint. Hence the compromised node  $A$  may not learn how much traffic node 5 relays to which next hop unless all neighboring nodes are compromised and collaborate.



(a) Established Paths in Transmission Range

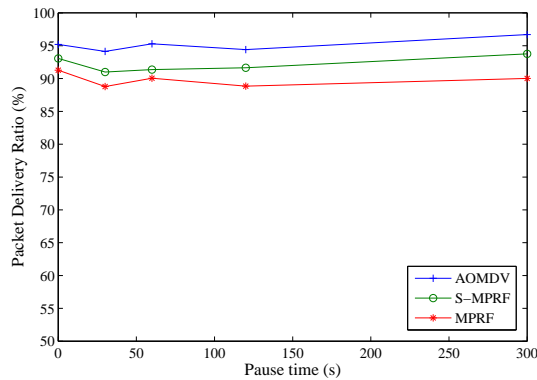


(b) Multiple Forward and Backward Next Hops

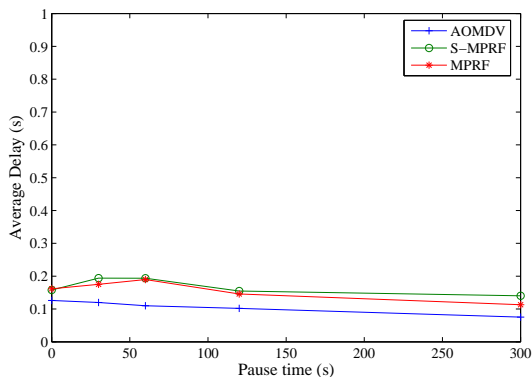
Fig. 6. Local Broadcasting and Eavesdropping

MPRF in PPCS spreads traffic over multiple paths, preventing eavesdroppers from learning the source, destination, or communicating pair by counting broadcast packets. Eavesdroppers located in different areas see different amounts of broadcast traffic with varying delay. Thus, a global eavesdropper is unable to discover significant information about node identity or flows.

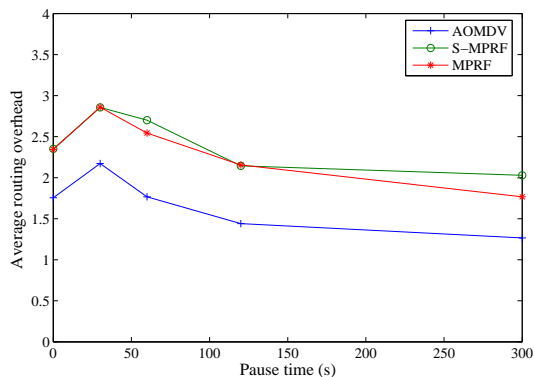
Understanding eavesdropping requires a model of traffic that encompasses the amount of information an adjacent eavesdropping node can observe, and distribution of information sent through that victim and intermediate nodes, and the frequency and structure of the underlying traffic. We are currently developing an analytical model for this exceedingly complex environment. For brevity, we defer the initial model's structure and results to future work.



(a) Packet Delivery Ratio



(b) Packet Delay



(c) Routing Protocol Overhead

Fig. 7. Performance with different pause times

## VI. PERFORMANCE EVALUATION

In this section, we evaluate the effect of PPCS on the performance of routing and data transmission. We performed our simulation in ns2 [7]. Specifically, we evaluate the effect of MPRF in which multiple paths are established and each packet on a flow may take a different path.

As a baseline multi-path routing protocol we use ad hoc on-demand multipath distance vector routing (AOMDV) [17]. To implement MPRF, we modified AOMDV to relax the node/link disjointness requirement and to randomly choose a next hop node at each intermediate node. Finally, to determine the impact of randomly changing the node pseudonym during the life of a flow, we modified MPRF to create a version that uses stable node pseudonym, called S-MPRF. Table VI summarizes the simulation environment.

TABLE III  
SIMULATION PARAMETERS

Simulation Time	900 seconds
Number of nodes	50
Area	900X900
Speed	Maximum 20 m/sec
Mobility model	Random Waypoint Model
Routing Protocol	Ad hoc On-demand Multipath Distance Vector Routing (AOMDV)
Packet size	512 bytes
Traffic pattern	10 CBR/UDP connections (4 packets/s)

We measured packet delivery ratio (PDR), end-to-end

packet delay, and routing overhead with different pause times under a random waypoint mobility model.

MPRF increasingly degrades the packet delivery ratio as mobility increases. Since each packet takes a different path, packets are more vulnerable to link failure or network congestion. Figure 7 (a) shows that the packet delivery ratio is decreased 3% and 5% in S-MPRF and MPRF, respectively. This result shows that the impact of changing node pseudonyms is small. The fact that multiple paths are susceptible to breaking for each flow, increases the routing overhead to overcome these failures. As shown in Figure 7 (c), there is a 42% increase in routing overhead in MPRF over AOMDV.

In traditional routing protocols, packets are transmitted on the shortest path. With MPRF packets are randomly distributed to across multiple paths. Because some paths will be longer than the shortest path, the end-to-end packet delay will increase. Figure 7 (b) shows a 51% increase in packet delivery delay in MPRF and S-MPRF.

We discuss the tradeoffs between the security and performance in the next section.

## VII. DISCUSSION

In this section we discuss the trade-offs of MPRF. According to the analysis in Section V-A, as the number of paths increases, the probability of an internal adversary compromising anonymity increases. While using non-disjoint paths is better

than using disjoint paths, both are less secure against internal attackers.

Although a single path solution is more secure against internal compromised nodes, it is less secure against anonymity attacks performed by eavesdroppers. To combat these attacks, it is better to establish more paths to distribute traffic. As an extreme example, if a packet is broadcast all over the network (the number of multiple paths is infinite), eavesdroppers may not discover a flow at all.

Based on a security perspective alone, the choice of using MPRF should be based on a risk analysis of the network. If an attacker is more likely to be external, MPRF should be used. If the attacker is more likely to be internal, it should not.

If MPRF is to be used, the packet forwarding performance of the network will decrease as discussed in VI. Disjoint multi-path forwarding provides a better packet delivery ratio (3-5%) than the non-disjoint multi-path forwarding used in MPRF. In non-disjoint multi-path environments, an intermediate node may receive packets of a flow from multiple neighbors which may cause more collisions on the wireless interface. However, given that the difference in performance is small, using MPRF is advisable as it does improve security as shown in Figure 5.

## VIII. CONCLUSION

MANETs may be used in support of many applications in which some level of privacy is required. While data privacy may be provided by encrypting end-to-end data, it is much more challenging to provide anonymity of node identity, location and correspondents. MANETs, by their nature, require nodes to cooperate for a network to work properly and efficiently. Most traditional methods of establishing routes and forwarding packets in MANETs require the disclosure of potentially sensitive information, such as an identity, location or desired destination.

In this paper we presented PPCS, a comprehensive system for providing anonymity in a MANET. The solution is efficient, so it is appropriate for a MANET environment. The solution is comprised of several components. The use of node identifier and flow pseudonyms provides a level of node anonymity and unlinkability between a source and destination. The use of multipath random forwarding combined with transforming packets on each link and using broadcast mechanisms to forward packets raises the level of difficulty in performing traffic analysis attacks. Obscuring the hop counts provided by many MANET protocols in the form of a TTL field reduces the ability of an adversary to determine its position on a path and use this information to derive a source or destination.

We provided a detailed security analysis of PPCS for passive internal attackers, i.e., against compromised nodes on the forwarding path between a source and destination. The analysis showed that PPCS is effective at reducing the effectiveness of adversaries. We also provided a discussion of the tradeoffs between performance and the security solution.

Finally, we provided a qualitative treatment of the security of PPCS against passive eavesdroppers. The detailed analysis of security against these types of attacks is very challenging and is left for future work. Likewise, an analysis of security against active attackers is also the subject of future work.

## REFERENCES

- [1] A. Back, U. Moller, and A. Stiglic. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. *Proceedings of Information Hiding Workshop (IH 2001)*, 2001.
- [2] M. Blaze, J. Ioannidis, and A. D. Keromytis. WAR: Wireless Anonymous Routing. *Security Protocols Workshop*, 2003.
- [3] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. A novel solution for achieving anonymity in wireless ad hoc networks. *Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks(PE-WASUN )*, 2004.
- [4] D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 1981.
- [5] H. Choi, W. Enck, P. McDaniel, and T. F. L. Porta. Secure Reporting of Traffic Forwarding Activity in Mobile Ad Hoc Networks. *Proceedings of The Second Annual International Conference on Mobile and Ubiquitous Systems*, 2005.
- [6] R. Dingleline, N. Mathewson, and P. Mathewson. Tor: The Second-Generation Onion Router. *Proceedings of the 13th USENIX Security Symposium*, 2004.
- [7] <http://www.isi.edu>. The Network Simulator - ns-2, 2000.
- [8] A. Jerichow, J. Muller, A. Pfizmann, B. Pfizmann, and M. Waidner. Real-Time Mixes: A Bandwidth-Efficient Anonymity Protocol. *IEEE Journal on Selected Areas in Communications*, 1998.
- [9] S. Jiang, N. Vaidya, and W. Zhao. A Dynamic Mix Method for Wireless Ad Hoc Networks. *MILCOM*, 2001.
- [10] S. Jiang, N. Vaidya, and W. Zhao. A Mix Route Algorithm For Mix-net in Wireless Mobile Ad Hoc c Network. *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2004.
- [11] D. Kesdogan, J. Egner, and R. Buschkes. Stop-And-Go-MIXes Providing Probabilistic Anonymity in an Open System. *Proceedings of the Second International Workshop on Information Hiding*, 1998.
- [12] J. Kong and X. Hong. ANODR:ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. *In ACM MOBIHOC*, 2003.
- [13] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. *IETF RFC 2104* (<http://www.ietf.org/rfc/rfc2104.txt>), 1997.
- [14] S.-J. Lee and M. Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. *IEEE International Conference on Communications*, 2001.
- [15] T. Leighton and S. Micali. Secret-key Agreement without Public-Key Cryptography. *In Proceedings of Crypto 93*, pages 456–479, August 1994.
- [16] B. N. Levine, M. K. R. C. Wang, and M. Wright. On timing attacks in low-latency mix-based systems. *In Proceedings of the 8th International Conference on Financial Cryptography*, 2004.
- [17] M. K. Marina and S. R. Das. AOMDV: Ad hoc On-demand Multipath Distance Vector Routing Protocol. *IEEE ICNP*, 2001.
- [18] B. Neil and C. Shields. Hordes: A Protocol for Anonymous Communication Over the Internet. *ACM Journal of Computer Security*, 2002.
- [19] B. C. Neuman and T. Ts'o. Kerberos: An Authentication Service for Computer Networks. *IEEE Communications*, 32(9):33–38, Sept. 1994.
- [20] C. Perkins and E. Royer. Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF RFC 3561* (<http://www.ietf.org/rfc/rfc3561.txt>), 1999.
- [21] A. Pfizmann and M. Hansen. Anonymity, unlinkability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology version v0.23. [dud.inf.tu-dresden.de/literatur/](http://dud.inf.tu-dresden.de/literatur/).
- [22] J.-F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems. *Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, 2000.
- [23] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. *Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection*, 1998.
- [24] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
- [25] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). *Internet Engineering Task Force*, June 2000. RFC 2865.
- [26] A. Serjantov and P. Sewell. Passive attack analysis for connection-based anonymity systems. *In European Symposium on Research in Computer Security*, 2003.
- [27] B. R. Waters, E. W. Felten, and A. Sahai. Receiver anonymity via incomparable public keys. *ACM conference on Computer and communications security*, 2003.

- [28] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A Framework for Reliable Routing in Mobile Ad Hoc Networks. *IEEE INFOCOM*, 2003.
- [29] Y. Zhang, W. Liu, and W. Lou. Anonymous Communications in Mobile Ad Hoc Networks. *IEEE INFOCOM*, 2005.