A Mix Route Algorithm For Mix-net in Wireless Mobile Ad Hoc Networks

Shu Jiang
Dept. of Computer Science
Texas A&M University
email: jiangs@cs.tamu.edu

Nitin H. Vaidya
Dept. of Electrical and Computer Eng., and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
email: nhv@uiuc.edu

Wei Zhao
Dept. of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
email: zhao@cs.tamu.edu

Abstract

Providing anonymous connection service in mobile ad hoc networks is a challenging task. In addition to security concern, performance concern must be addressed properly as well. Chaum's Mix method [4] can effectively thwart an adversary's attempt of tracing packet routes and hide source and/or destination of packets. However, applying the Mix method in ad hoc networks may cause significant performance degradation due to its non-adaptive Mix route selection algorithm. We propose a dynamic Mix route algorithm to find topology-dependent Mix routes for anonymous connections. Its effectiveness in improving network performance is validated by simulation results. We also address the potential degradation of anonymity due to dynamic Mix route.

1. Introduction

A wireless mobile ad hoc network is formed by a group of mobile hosts (or nodes) that communicate through radio transmissions, without support of fixed routing infrastructure. Due to its ease of deployment, it has a large amount of applications in military (such as battlefield) as well as in civilian (such as emergency, conference) environments. However, wireless medium introduces great opportunities for eavesdropping of wireless data communications. Anyone with the appropriate wireless receiver can eavesdrop and this kind of eavesdropping is virtually undetectable. So communication privacy is one of the issues that a network designer must address with higher priority.

By definition, privacy means the protection of data from unauthorized parties. Federrath et al. [5] discuss the communication privacy requirements in mobile networks in terms of content, location, identity privacy. Content privacy, i.e. protection of the contents of a message, can be provided by encryption schemes (such as AES, DES and RSA). In the wireless ad hoc network we are considering, node address itself does not contain location information, but may disclose identity of mobile users. An adversary may learn user communication patterns such as who communicates with whom, when, how long, etc. from traffic information. To thwart traffic analysis, it is desirable that user communications remain anonymous. How to provide anonymity support in wireless ad hoc network is the topic of this paper.

Achieving anonymity is a different problem than achieving data confidentiality. While data can be protected by cryptographic means, the recipient node address (and maybe the sender node address) of a packet can not be simply encrypted because they are needed by the network to route the packet. Most existing anonymizing schemes are originated from Chaum's Mix-net concept [4]. The idea is that traffic sent from sender to destination should pass one or more Mixes. A Mix relays data from different end-to-end connections, and its task is to reorder and re-encrypt the data such that incoming and outgoing data cannot be related. This should thwart attempts of an outside eavesdropper to follow an end-to-end connection. A Mix-net can protect against colluding Mixes if not all Mixes involved in relaying an end-to-end connection collude with the adversary. This is an important property because, in a hostile environment (e.g., battlefield), the probability that roaming nodes be captured cannot be neglected. Generally, the more

0-7803-8815-1/04/$20.00©2004IEEE

Mixes are involved in relaying an end-to-end connection, the lower the probability that the connection be compromised. However, relaying data traffic through too many Mixes would inevitably increase the average data latency and decrease the average data delivery ratio. So the number and sequence of Mixes in the path of an end-to-end connection must be appropriately determined in order to reach a balance between the two contradictory goals. This is the so-called *Mix routing* problem.

Mix routing has not received sufficient attention in the design of existing Mix-based anonymizing systems. The reason behind this is that most existing systems are designed for operating over the Internet. One class of anonymizing systems is represented by Onion Routing [15], where the Mix set is small, all Mixes are administered by a central authority, and the Mix-net topology remains stable during run time [2, 3]. Another class of anonymizing systems that emerged recently is of peer-to-peer type [16, 6], where all participating nodes are potential originators of traffic as well as potential relays. Since a peer-to-peer anonymizing network has a very large node base, an adversary cannot observe the *entire* network. Mix route can be constructed as follows. The source node of an end-to-end connection chooses the first Mix from its neighbor set, which then chooses the second Mix similarly, and so on. The Mix route length can be controlled by the source node [6], or by the last Mix based on a *probability of forwarding* [16]. The biggest challenge of Mix routing in wireless ad hoc network is dynamic change of topology, which makes a static or random Mix route inefficient. We make contributions in improving Mix-net performance by proposing dynamic Mix route which adapts to topology change.

The rest of the paper is organized as follows. In section 2, we describe the basic functions of a Mix-net and potential adversary. In section 3, we present a dynamic Mix route algorithm designed for wireless ad hoc network, and conduct a qualitative cost and security analysis. Performance evaluation of the algorithm is conducted by means of simulations using ns-2 [1]. The results are presented in section 4. Section 5 discusses related work. Finally, section 6 concludes this paper.

2. Basic Model

We consider a wireless ad hoc network in which a subset of mobile nodes are Mixes. Mixes cooperate to provide anonymous connection service to any source/destination pairs, regardless of node type. In other words, anonymous connections can be established between two non-Mix nodes, one non-Mix node and one Mix, and two Mixes. For the ease of presentation, we assume that the source and destination of an anonymous connection are both non-Mix nodes.

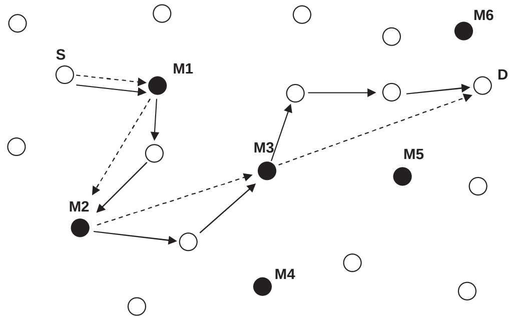

Figure 1. A Mix-net example in a wireless ad hoc network

The original Mix-net is based on public key cryptosystem. Assuming that each Mix i generates a pair of keys K_i and K_i^{-1}, the public key K_i is made known to all users and the private key K_i^{-1} is never divulged. Chaum described a way of delivering message without disclosing sender/recipient relationship, as follows [4]. First, the sender S decides a *Mix route*, which is a sequence of Mixes. Second, S "seals" a message M for delivery by successively encrypting M with public keys of the Mixes in the route. Say the Mix route is $M_1 \rightarrow M_2 \rightarrow \ldots \rightarrow M_k$, and the encryption of a message X with key K is denoted $K(X)$. The sealed message M would be of the form

$$K_1(R_1, K_2(R_2, \ldots, K_k(R_k, M)\ldots))$$

where R_i is a random string attached to message before each encryption. Only the holder of the private key K_i^{-1} can interpret a message encrypted with the public key K_i. So the sealed message will be sent to M_1, who can remove one layer of encryption, throw away R_1, and send the remaider of the message to the next Mix M_2. Each Mix in the route follows the same procedure, and the last Mix M_k will finally deliver M to its recipient node. M can be encrypted with the recipient's key or plain text. Note that each Mix knows only the previous and next Mix, except that the first and last Mix know the sender and recipient of the message respectively. Hence, unless all Mixes are compromised, an adversary cannot determine both sender and recipient of the message.

The purpose of a Mix is to hide the correspondences between the messages in its input and those in its output. How well a Mix achieves this goal depends on a number of factors, such as the adversary's ability, the Mix flushing algorithm, the Mix input size (i.e., traffic load), etc. We assume the same adversary and attack model as in [10], i.e., a powerful adversary with unbounded eavesdropping capability but bounded computing and node intrusion capability. This means that (i) the adversary can eavesdrop transmissions on all wireless links but cannot break public key or symmetric key crypto-systems to discover the contents of the messages without acquiring the corresponding keys; (ii) the adversary may capture and compromise Mixes but cannot successfully compromise more than K members during a time window T. In addition, intrusion detection is not perfect. So a compromised Mix exhibiting no malicious behavior will stay in the network and participate in relaying traffic. This means that the untraceability of an end-to-end connection can never be guaranteed.

An example of ad hoc Mix-net is given in Figure 1, where Mixes are indicated by dark nodes. In this setting, each packet from node S to node D pass through three Mixes, M_1, M_2 and M_3. Hence the mix route created for the source-destination pair (S, D) is $M_1 \rightarrow M_2 \rightarrow M_3$, as shown by the dashed-line. The solid line draws the physical route that data packets actually take. Clearly, a mix route is a logical route, not a physical route, and the Mix-net is an overlay network.

When one deploys Mix-net in wireless ad hoc network, there are more issues than mix routing to address. We list the most prominent issues below. However, in this paper, we are focused on the mix routing problem and leave other problems for future work.

1. The public key cryptographic operation incurs considerable processing burden on mobile nodes under power constraints and with limited computing capabilities. So the Mix-net based on public key cryptosystem is not efficient in wireless ad hoc network. Fortunately, Mix-net variants based on symmetric key or hybrid crypto-systems have been proposed by many researchers and can be adapted to wireless networks [14, 10, 12, 7, 11]. For example, using a hybrid encryption method, the sender of a message may generate symmetric keys for sealing the message and use public keys of the Mixes in distributing those keys. Since the key length is much smaller than the message size, a great saving on processing overhead is expected.

2. A mutual authentication mechanism is needed for the non-Mix nodes to establish trust relationship with the Mix nodes that advertise the anonymity service.

3. A charging and accounting mechanism may be needed when the Mix nodes do not provide the service for free.

3. Proposed Mix Route Algorithm

In this section, we present a dynamic Mix route algorithm. The purpose of the algorithm is to find Mix route for an end-to-end connection. We set several design goals for the algorithm. First, connection anonymity should not be violated during the Mix route discovery process. Second, the algorithm should find a short Mix route based on the current network topology. As the network topology changes, the algorithm should update the Mix route. Third, the algorithm should have low and bounded overhead.

We briefly describe the algorithm first, followed by a detailed discussion. The proposed algorithm consists of two independent processes: Mix advertisement (using MADV messages), and Mix route discovery and update (using DREG, RREQ, and RUPD messages). We should emphasize that the "Mix route discovery" process runs on top of any underlying routing protocol. In essence, the Mix route discovery process finds routes consisting of "virtual links" between Mix nodes – a virtual link in the Mix-net is a path in the physical network.

- The purpose of Mix advertisements is for the Mix nodes to announce their presence to non-Mix nodes. Each non-Mix node tries to pick the nearest Mix node as its "dominator" Mix node – the dominators serve a function in anonymous routing as seen below.

- Due to node mobility, each non-Mix node may dynamically change the Mix node chosen as its dominator. To make each Mix node aware of its dominator relationship with non-Mix nodes, the non-Mix nodes use DREG messages to register at their dominator Mix nodes.

- In our approach, when a node S needs to find an anonymous route (through one or more Mix nodes), it sends a RREQ message to the destination D via a custom Mix route formed by a set of randomly chosen Mixes or by S's dominator Mix. The custom Mix route may not be right choice from performance perspective, therefore, the rest of the Mix route discovery process attempts to find a better Mix route for the connection. For instance, if S chooses a Mix M_3 randomly, then the Mix route for the RREQ will be $S \rightarrow M_3 \rightarrow D$. The RREQ packet is routed from S to M_3 using the underlying routing protocols (e.g., DSR [8]), and from M_3 to D similarly. When D receives the RREQ, the destination node realizes that it is an endpoint for an active connection. Therefore, it registers with its dominator Mix by sending a DREG message.

- Any Mix node that has a non-empty list of registered non-Mix nodes periodically transmits a RUPD message as elaborated later. The purpose of RUPD transmissions is to allow a source node to discover a Mix route regarding a particular destination node (A RUPD message contains a list of all destination nodes currently registered at the Mix node who creates the message).

We elaborate on the above algorithm in the rest of this section.

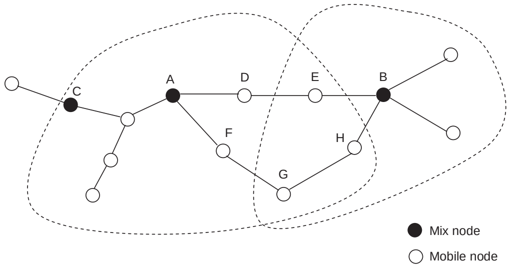

Figure 2. Flooded area of Mix advertisements

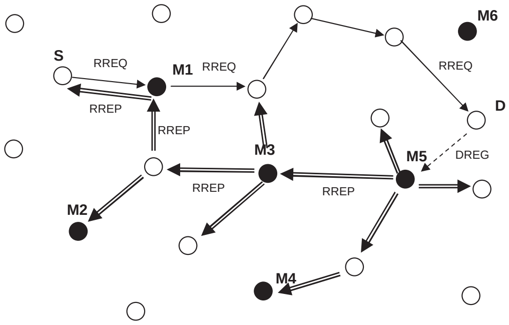

Figure 3. Mix Route Discovery Process

3.1. Mix Advertisement

We introduce a low-cost Mix advertising algorithm for non-Mix nodes to find the closest Mix to each of them as dominator:

1. Every Mix periodically broadcasts Mix Advertisement (MADV) messages to announce its presence to non-Mix nodes in the neighborhood. The time interval between two consecutive advertisements is ADVERTISE_INTERVAL. MADV from Mix M has message format:

   $$< MADV, M \rightarrow ALL, seqnum, radius >$$

   where (i) *seqnum* together with M's address uniquely identify a MADV message. (ii) *radius* value indicates how far the message has propagated. When the message is created, it is set as zero.

2. A non-Mix node learns Mixes in its neighborhood from received MADV messages and maintains the closest Mix information, which is also the node's dominator Mix. As time elapses, the node's neighborhood may change. Therefore, a non-Mix node's dominator Mix is not constant. It is also possible that a non-Mix node loses connectivity with its current dominator Mix. So if a non-Mix node does not receive MADV packet from the current dominator Mix for a time interval of length 2 $*$ ADVERTISE_INTERVAL, it switchs to a new dominator Mix. A non-Mix node only retransmits MADV messages from its dominator Mix. Every time when a MADV message is retransmitted, the *radius* value in it is incremented by 1.

3. A Mix node discards MADV messages it received.

The described algorithm is unlike the conventional, network-wide flooding algorithm. Each MADV message has a limited flooded area. Typically, it only arrives at nodes that are closer to it than to any other Mixes. We use an example to illustrate this idea. In Figure 2, the border of two Mixes' flooded area is shown by dashed-line. A is the closest Mix to D and hence, D's dominator. So D will retransmit A's MADV messages. But E does not retransmit A's MADV's it received from D because Mix B is closer to it than A is. The validity of this algorithm can be shown by considering a non-Mix node that receives two MADV messages, one from the closest Mix M, another from a farther Mix X. The *radius* values in the two messages must satisfy $radius(M) < radius(X)$. Suppose that the node retransmits both messages. A neighboring node that receives the two messages will find that the above relationship still holds because the *radius* values in both messages are increased by 1, respectively. In other words, based on these two messages, X can never be closer to any downstream nodes than M is. So it is unnecessary to forward the MADV messages from X.

3.2. Mix Route Discovery and Update

The Mix route discovery process might be best described by example. Figure 3 shows a Mix-net of 6 Mixes (marked as dark). Node S wishes to find a mix route for an anonymous connection destined to node D. The Mix route discovery process can be divided into three phases:

1. *RREQ phase*: S assembles a RREQ message and sends it to D via a custom Mix route. As we mentioned, a custom Mix route can be a random route consisting of randomly chosen Mixes, or be the dominator Mix of S as in this example. The RREQ message is a unicast message. So S can encrypt the content of the message with D's public key to prevent tracing of the message by an attacker. The RREQ packet may be lost during transmission. So a timeout-based retransmission mechanism must be activated by S.

2. *DREG phase*: When D receives a RREQ message, it knows that it is destination of a new end-to-end connection. If D did not yet register at its dominator Mix

(M_5 in this example), it does so by sending a Destination Registration (DREG) message to the Mix. Let M be D's dominator Mix. The DREG message would have format

$$< DREG, D \rightarrow M, seqnum >$$

D must send DREG messages periodically to maintain its association with the Mix. There are several reasons for this design. First, DREG message may be lost during transmission and never reaches the Mix. Second, as network topology changes, D may switch to different dominator Mix. In this case, D simply sends DREG messages to the new dominator Mix and increases the *seqnum* in it. The old dominator Mix may learn this change from one of two events. One is expiration of D's registration record because there is no new DREG message arriving from D. Let DREG_INTERVAL be the time interval between two consecutive DREG messages. The expiration time of a destination node's registration at Mix is set as $2 *$DREG_INTERVAL in the algorithm. Another is receiving RUPD message from D's new dominator Mix (explained below).

3. *RUPD phase*: Every Mix maintains a list of registered destination nodes. If the list is not empty, it periodically broadcasts RUPD messages. The time interval between two consecutive broadcasts is RUPD_INTERVAL. RUPD message from a Mix M is of the format

$$< RUPD, M \rightarrow ALL, seqnum, l, path >$$

where (i) *seqnum* together with M's address uniquely identify a RUPD message. (ii) *l* is the list of destination nodes currently registered at M. Each entry of the list includes node address and the latest DREG *seqnum*. (iii) *path* records a Mix route that the packet has traversed during flooding. Initially, *path* contains M, the initiator of the message.

The flooding of a RUPD message proceeds as follows. The initiator Mix broadcasts the message locally. If a node X that receives the RUPD message has pending data packets in its queue addressed to destination node(s) in l, then it copies the Mix route in *path* and uses the reverse Mix route in delivering those data packets[1]. If X is a Mix, then it checks whether any destination node in l carries a higher DREG *seqnum* and updates its own list accordingly. When the above processing is completed, X retransmits the RUPD message, and if X is a Mix, it appends its ID to the $path$ before retransmitting. It is possible that X receives the same RUPD message for multiple times. To ensure that a RUPD message is retransmitted only once, X keeps a record of each RUPD message it retransmitted. However, from the multiple RUPD messages that arrive via different paths, X may obtain multiple distinct Mix routes to the same destination node. In Figure 3, the retransmissions of RUPD message are indicated by double arrows. It is shown that S will find a Mix route $M_1 \rightarrow M_3 \rightarrow M_5$ for its connection to D.

From the above description, we know that the RUPD message is flooded along the shortest path tree rooted at the initiator Mix. For the same destination node, different source nodes receive different Mix routes and the minimum length of each Mix route is 1. By making small change to the algorithm, we may obtain Mix routes with minimal length 2. The idea is that each Mix caches the Mix routes it received and broadcasts them along with MADV messages. The source node of a connection will use the Mix Route received from its dominator Mix, which contains at least two Mixes. If a larger minimal length of mix route is desired, then we need to develop new Mix Route Discovery algorithm.

The update of Mix route for an anonymous connection is realized by periodical RUPD broadcasts. If a node is not destination of any active connection, it should stop sending DREG messages to its dominator Mix. We assume that an in-band protocol exists for the source node to inform the destination node of connection termination.

3.3. Security Analysis

In this section, we analyze the security aspect of the proposed algorithm. Raymond [13] presents a good survey of known attacks against Mix-net. So we focus on new attacks that employ vulnerabilities in the dynamic mix route algorithm to reveal source and destination of an anonymous connection.

During the mix route discovery process, an attacker may employ the correlation between RREQ message and DREG message to reach its goal. For instance, if the attacker observes that node S sends a RREQ message and node D sends a DREG message shortly later, then it is very likely that S and D are two end-points of a new anonymous connection. We have mentioned that RREQ messages can be encrypted with destination node's key so that only the destination node can interpret the contents of the messages. However, message encryption is not effective when there is no sufficient cover traffic. In this case, S can send multiple dummy messages, say to itself, before it sends real RREQ message.

[1] If the RUPD packet is received from an unidirectional link, it should be discarded because there is no reverse link.

Packet Type	Packet Count	Asymptotic Upper Bound
MADV	$\frac{T}{ADVERTISE\_INTERVAL} * n$	$O(n)$
RREQ	c	$O(c)$
DREG	$c * \frac{T}{DREG\_INTERVAL}$	$O(c)$
RUPD	$m * \frac{T}{RUPD\_INTERVAL} * n$	$O(mn)$

Table 1. Analysis of Control Packet Load

During the mix route update process, a long-lived connection is subject to intersection attack due to change of Mix route. In a high-mobility network, it is very likely that the source node of a connection receives multiple updates of Mix route during the connection lifetime. Assuming that the source node uses the shortest Mix route all the time, the attacker can perform attack as follows. The attacker finds the shortest Mix routes and the first Mixes in the route to each suspected destination. If the shortest Mix route to a destination node changes and the new Mix route has different "first Mix" than the old Mix Route had, the attacker can observe whether the source node "shifts" data traffic from the connection to the old "first Mix" onto the connection to the new "first Mix". The attacker has better chance to succeed when the source node has only a few connections. To prevent this attack, a perfect, but very costly, solution is that the source node maintains constant traffic loads to each Mix by use of dummy traffic. A less costly solution is that the source node splits the data traffic between multiple Mix routes, which are learned from RUPD messages. This should complicate traffic analysis and reduce dummy traffic load as well. However, either solution decreases network performance.

3.4. Cost Analysis

In this section, we analyze the control overhead of the proposed algorithm. We count the total number of control packets generated during a time window T. For broadcast control packet, retransmissions of the packet are counted individually.

Let n be the total number of network nodes, and m be the number of Mixes. For analysis purpose, we assume that c end-to-end connections (each with different destination node) are set up during the time window T. In the algorithm, MADV packets are generated and flooded by each Mix at an interval of ADVERTISE_INTERVAL. During each advertisement cycle, every non-Mix node retransmits MADV packet (from the dominator Mix) only once. So the total number of transmissions of MADV packets by all nodes is n. RREQ packets are generated by the source node of each end-to-end connection. Assuming that all RREQ packets are successfully delivered, the total number of RREQ packets during the time window T must be c. The destination node of each end-to-end connection generates DREG packets periodically at an interval of DREG_INTERVAL. So the total number of DREG packets during the time window T must be $c * \frac{T}{DREG\_INTERVAL}$. In the worst case, all Mixes need to generate RUPD packets. Each RUPD packet is flooded to all nodes and each node retransmits each RUPD packet only once. Hence, the total number of transmissions of each RUPD packet amounts to n.

The above analysis is summarized in Table 1. It is shown that majority of the control overhead is due to the periodical flooding of RUPD packets. In the worst case, the overall control packet load is $O(mn + c)$. But on the average, the number of Mixes that broadcast RUPD packets is expected to be less than m.

4. Performance Evaluation

We evaluate the performance of the proposed Mix route algorithm in three aspects. First, we investigate the network performance of Mix-net in a wireless ad hoc network. Two metrics are used: (i) *Packet delivery ratio* - the ratio between the number of data packets received and those originated by the sources. (ii) *Average end-to-end data packet latency* - the time from when the source generates the data packet to when the destination receives it. This includes: latency for determining Mix route, network routing latency, cryptographic processing delays, queueing delay at the interface queue, retransmission delay at the MAC, propagation and transfer times. In addition to the proposed algorithm, we also implement a static Mix route algorithm and a random Mix route algorithm for comparison. Both algorithms construct topology-independent Mix route by selecting Mixes randomly. In a static Mix route algorithm, all end-to-end connections use the same Mix route, whereas in a random Mix route algorithm, Mix route is determined on a per-connection basis. Second, we measure the average length of dynamic Mix route. When each Mix has the same independent probability of being compromised, the probability that all Mixes in a Mix route be compromised decreases exponentially with the number of Mixes. Third, we evaluate the control overhead of the proposed algorithm. The metric we use is *normalized control packet load*, which is defined as the number of control packets transmitted per data packet delivered.

4.1. Simulation Model

We use ns-2 [1] simulation package to simulate a wireless ad hoc network, and implement the proposed Mix route algorithm. At the physical layer, we simulate Lucent's WaveLAN card with a nominal bit rate of 1 Mbits/sec and a nominal transmission range of 250 meters. At the MAC layer, we use the distributed coordination function (DCF) of IEEE 802.11. At the network layer, the DSR routing protocol is used in routing of data packets. In our experiments, we simulate the stop-and-go Mix [9] where each Mix adds a random delay (uniformly distributed between 0 and 100 milliseconds) to each received packet before sending it out [9]. The processing overhead of packet encryption/decryption is modeled based on Kong and Hong's measurement in [10], which however is negligible compared to end-to-end packet latency. At the beginning of each simulation run, a given number of randomly chosen nodes are designated as Mixes. The parameter values in our implementation of the dynamic Mix route algorithm are listed in Table 2.

The network field is 1000m x 1000m with 50 nodes initially uniformly distributed. *Random Way-point* mobility model [8] is used to generate node movement scenario. According to this model, a node travels to a random chosen location in a certain speed and stays for a while before going to another random location. In our simulation, the maximum node speed varies from 0 to 20 m/sec, and the pause time is fixed to 30 seconds. Constant Bit Rate (CBR) sessions are used to generate data traffic. For each session, data packets of 512 bytes are generated in a rate of 4 packets per second. The source-destination pairs are chosen randomly from all the nodes (including Mixes). During 300 minutes simulation time, totally 25 sessions are scheduled with start times uniformly distributed between 0 and 180 seconds, and each session lasts for approximately 75 seconds.

4.2. Simulation Results

In figure 4 and 5, we show the network performance of an ad hoc Mix-net with different Mix route algorithms. The Y-axis represents packet delivery ratio or average data packet latency in seconds. The X-axis represents the maximum node speed in figure 4, or the number of Mixes in figure 5. These figures illustrate that the proposed dynamic Mix route algorithm performs better than the static Mix route and the random Mix route algorithm by consistently achieving higher packet delivery ratio and lower packet latency. These results are not surprising because dynamic Mix route is adaptive to network topology change and ensures that data packets are routed along a short route in the physical network. As shown in figure 4, the network performance suffers as node mobility increases, due to frequent change of Mix route and large packet losses. It is interesting to note that, with static Mix route or random Mix route, the average packet latency is lower in a high-mobility network than in a static network. The reason is that, when network topology changes, the physical route for the same Mix route is not constant and there is a good chance that some "instances" are short. Figure 5 shows that the number of Mixes in a network has slight effect on network performance. This is decided by the DSR-like Mix route discovery process in our algorithm, and should not be accepted as a general rule. In general, more Mixes in a network means shorter Mix route and better network performance.

In figure 6 and 7, we plot the average length of dynamic Mix route as functions of increasing mobility and number of Mixes, respectively. The static Mix route and random Mix route each contain one Mix in our experiments. So both have constant length 1. As shown in figure 7, there is a linear correlation between the dynamic Mix route length and the number of Mixes in the network. This is because Mixes are randomly selected from the node set, and hence, uniformly distributed over the network area. Figure 6 shows that network topology change does not degrade the quality of dynamic Mix route as long as the number of Mixes in the network remains the same. It suggests that high degree of anonymity can be achieved if there is sufficient number of Mixes in the network.

The overhead analysis of the proposed algorithm is presented in figure 8, in which the Y-axis represents the ratio between the number of control packets transmitted and the number of data packets that are delivered, and the X-axis represents the maximum node speed. The figure illustrates that the normalized control overhead is slightly higher in dynamic network than in static network. When the traffic load is low (with 5 connections), the control overhead of the algorithm is pretty high (up to 5 control packets for delivering one data packet). The reason is that, in our algorithm, Mix advertisement packets are generated with no regard to data traffic load and set a lower bound for control traffic load. As the number of connections increases, the number of delivered packets increases as well, which drops the normalized control overhead.

5. Related Work

Basagni et al. [17] proposed to encrypt routing messages with a network-wide symmetric key. This scheme effectively stops eavesdroppers, but fails when a single node is compromised and discloses the key. The authors argue to protect the key using tamper resistance facilities which introduce physical cost and offer indefinite physical warranty.

ANODR [10] is a recently proposed on-demand anony-

ADVERTISE_INTERVAL	3 secs
DREG_INTERVAL	3 secs
RUPD_INTERVAL	10 secs

Table 2. Parameter Values in Simulations

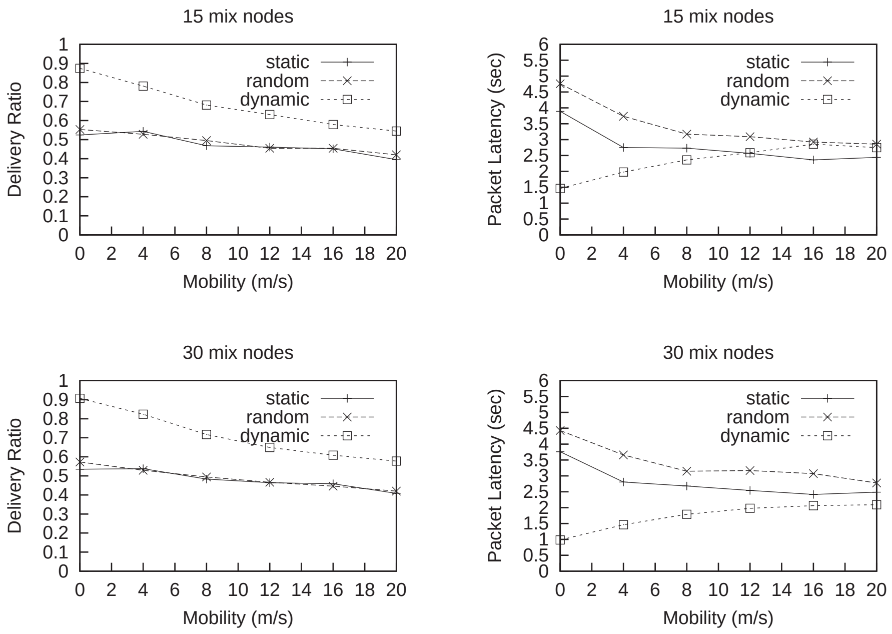

Figure 4. Network Performance vs. Mobility

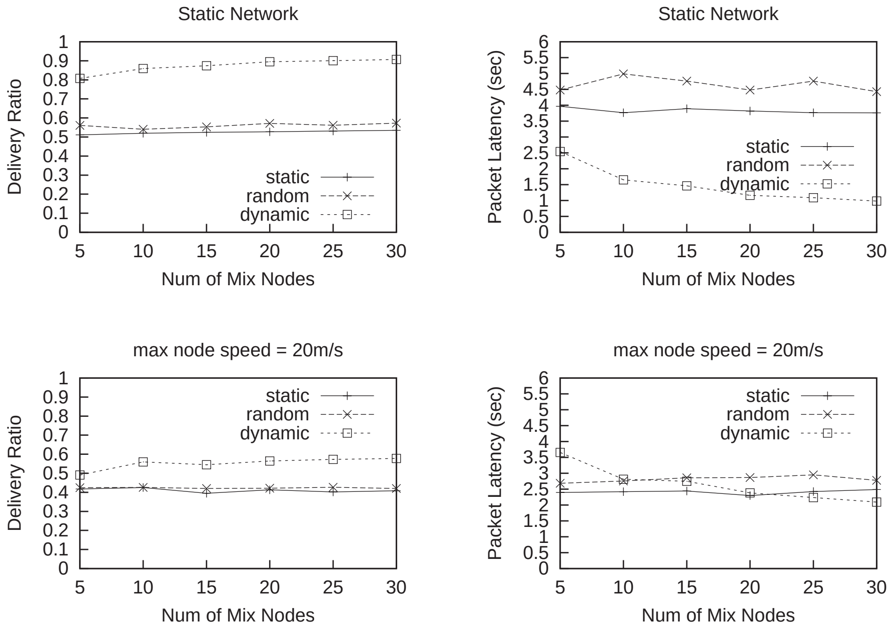

Figure 5. Network Performance vs. Number of Mixes

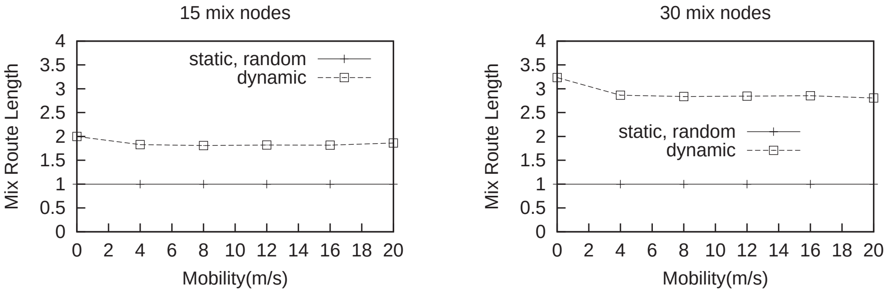

Figure 6. Mix Route Length vs. Mobility

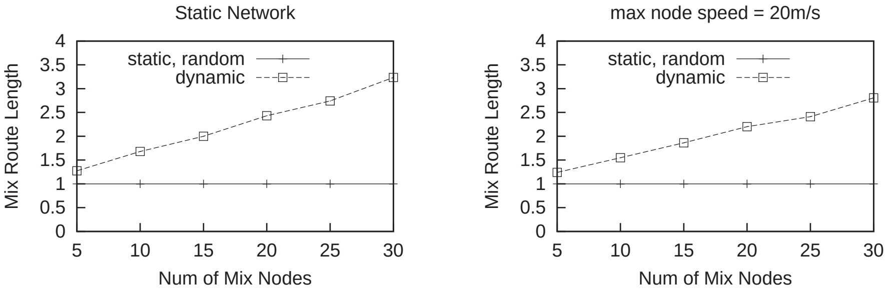

Figure 7. Mix Route Length vs. Number of Mixes

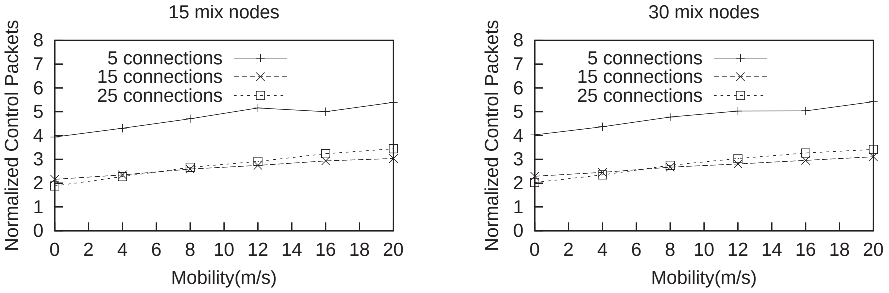

Figure 8. Normalized Control Packet Load vs. Mobility

mous routing protocol which incorporates anonymous connection setup into the route discovery process. The design of the protocol is based on a novel "broadcast with trapdoor information" technique. However, broadcast transmission in wireless ad hoc network is not really anonymous. On the physical layer, it is usually possible to locate a sending device by recording signal delays and performing triangulation. This means that an unbounded eavesdropper can establish links between route pseudonyms and physical nodes. According to the protocol, when a node detects a broken link, it broadcasts RERR packets to notify upstream nodes. The route pseudonyms in RERR packets disclose the upstream nodes based on above analysis.

6. Conclusions

In this paper, we describe new efforts in providing anonymous communication service in wireless ad hoc network based on Mix-net scheme. We propose an efficient algorithm for determining Mix route for an end-to-end connection. The design of the algorithm is based on two flooding processes: Mix advertisement, and Mix route discovery and update. Much efforts have been made to reduce transmission overhead in the two processes. Simulation results demonstrate significant performance gain achieved by dynamic Mix route in contrast with topology-independent Mix route in the conventional Mix-net. Moreover, the average length of the dynamic Mix route is decided by the number of Mixes in the network, and does not change as node mobility increases.

References

[1] U. Berkeley, LBL, USC/ISI, and Xerox-PARC. ns notes and documentation, 2003. http://www-mash.cs.berkeley.edu/ns.

[2] O. Berthold, H. Federrath, and S. Köpsell. Web mixes: A system for anonymous and unobservable internet access. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 115–129. Springer-Verlag, July 2000.

[3] C.Gulcu and G.Tsudik. Mixing email with babel. In *IEEE Symposium on Network and Distributed System Security (NDSS'96)*, San Diego, CA, USA, Feb. 1996.

[4] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, Feb. 1981.

[5] H. Federrath, A. Jerichow, D. Kesdogan, and A. Pfitzman. Security in public mobile communication networks. In *Proceedings of IFIP/TC6 Personal Wireless Communications*, pages 105–116, Prague, 1995.

[6] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *ACM Conference on Computer and Communications Security (CCS'02)*, Washington, DC, USA, Nov. 2002.

[7] M. Jakobsson and A. Juels. An optimally robust hybrid mix network. In *ACM Symposium on Principles of Distributed Computing (PODC'01)*, Newport, Rhode Island, USA, Aug. 2001.

[8] D. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwere Academic Publishers, 1996.

[9] D. Kesdogan, J. Egner, and R. Büschkes. Stop-and-go mixes: Providing probabilistic anonymity in an open system. In *Information Hiding Workshop (IH'98)*, volume 1525 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[10] J. Kong and X. Hong. Anodr: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03)*, Annapolis, MD, USA, June 2003.

[11] M. Ohkubo and M. Abe. A length-invariant hybrid mix. In *ASIACRYPT'00*, volume 1976 of *Lecture Notes in Computer Science*. Springer-Verlag, 2000.

[12] A. Pfitzmann, B. Pfitzmann, and M. Waidner. Isdn-mixes: Untraceable communication with very small bandwidth overhead, feb 1991.

[13] J.-F. Raymond. Traffic analysis: Protocols, attacks, design issues and open problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 10–29. Springer-Verlag, July 2000.

[14] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, Dec. 1997.

[15] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998.

[16] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[17] D. B. S. Basagni, K. Herrin and E. Rosti. Secure pebblenets. In *MobiHoc '01*, pages 156–163, 2001.

[18] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: A core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8):1454–1465, Aug. 1999.