ANONYMOUS HANDSHAKES IN MOBILE AD HOC NETWORKS

†Yanchao Zhang, †Wei Liu
†Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611
{yczhang@,liuw@,fang@ece.}ufl.edu

‡Wenjing Lou , †Yuguang Fang
‡Department of Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA 01609
Email: wjlou@ece.wpi.edu

ABSTRACT

The broadcast nature of radio transmissions renders communications in mobile ad hoc networks more vulnerable to malicious traffic analysis than their wired counterparts. As a result, adversaries can easily locate and trace mobile nodes based on their invariant identifiers so as to launch pinpoint attacks. To tackle this problem, this paper presents a novel anonymous on-demand routing protocol, called MASK, which nicely fulfills the routing and packet forwarding tasks without disclosing the identities of participating nodes under a rather strong adversarial model. MASK provides the desirable sender and receiver anonymity, as well as the relationship anonymity of the sender and receiver. It is also resistant to a wide range of adversarial attacks.

I. INTRODUCTION

In hostile environments, in addition to the well-known security objectives such as confidentiality, data integrity, authentication, and non-repudiation, *anonymity* is another desirable property in the sense that the identity privacy of mobile nodes should be well protected from adversaries. The leakage of such information is often devastating under many circumstances. As an example, adversaries may be capable of locating and chasing some VIP nodes based on their invariant exposed identifiers so as to launch pinpoint attacks on them and paralyze the whole communication system. The shared wireless medium of mobile ad hoc networks (MANETs) introduces opportunities for passive eavesdropping on data communications, which further deteriorates the leakage of nodal identity information[1]. This situation necessitates the development of anonymous communication protocols for preserving the anonymity (identity privacy) of mobile nodes in MANETs.

Communications in MANETs consist of one-hop communications between neighboring nodes and multi-hop communications between multi-hop-away nodes. To guarantee the security of one-hop communications, secure authentication between neighboring nodes is indispensable so that one node can reject accepting or forwarding messages from unauthenticated neighbors. Otherwise, adversaries can inject arbitrary phony messages into the network to deplete the network resources as well as interrupting the proper network functions. However, the conventionally prevalent authentication techniques based on public-key certificates may inevitably disclose nodal identity or group information contained in public-key certificates and hence is not appropriate for achieving anonymous one-hop communications.

Multi-hop communications rely on routing protocols to find end-to-end paths between sources and destinations. But common routing protocols for MANETs are lack of the anonymity property in that they may disclose the node identities through routing or data packets. For example, DSR [1] explicitly embeds nodal identity information in packet headers. Although AODV [2] is less dangerous than DSR in that routing information is stored in routing tables instead of packet headers, multiple collaborative adversaries en route can still ascertain the identities of some or all the participating nodes of one on-going communication by combining and analyzing their eavesdropped routing information. Some of such attacks can be found in [3].

We propose the notion of *anonymous handshakes* to handle the above problems, by which we intend to achieve the following: 1) neighboring nodes can anonymously authenticate and communicate with each other without disclosing their identities to each other; 2) Multi-hop-away nodes can anonymously communicate with each other without divulging the real identities of sources, destinations, and all the intermediate nodes; and 3) no one can link a give node identity to a particular mobile node in the network.

In particular, based on a new cryptographic concept called *pairing*, we first present an anonymous neighborhood authentication protocol which allows neighboring nodes to authenticate each other without revealing their identities. By utilizing the secret pairwise link identifiers and keys established between neighbors during the neighborhood

[1]In MANETs, the identity of one node can be its invariant identifier or network-layer address or MAC (Medium Access Control) address that can uniquely identify the node.

0-7803-8847-X/04/$20.00 ©2004 IEEE

authentication process, we then develop a novel anonymous on-demand routing protocol, termed MASK, to fulfill the routing and packet forwarding tasks without disclosing the identities of all the participating nodes. MASK provides the desirable sender and receiver anonymity, as well as the relationship anonymity of the sender and receiver[2]. It is also designed to be resistant to a wide range of adversarial attacks.

The rest of this paper is structured as follows. Section II describes the cryptographic tools and the adversarial model used in this paper. Section III details the MASK design and outlines a number of malicious attacks that MASK is able to withstand. Section IV reviews the related work and this paper is concluded in Section V.

II. PRELIMINARIES

A. Pairing concept

Pairing has recently found a number of interesting applications in cryptography, e.g., [4]–[6], and it forms the cryptographic foundation of our scheme. The basic concept of pairing is outlined as follows.

Let $\mathbb{G}_1, \mathbb{G}_2$ be two groups of the same prime order q. We view $\mathbb{G}_1$ as an additive group and $\mathbb{G}_2$ as a multiplicative group throughout the paper. A pairing is a computable *bilinear map* $f : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties:

1. *Bilinearity*: $\forall\ P, Q, R, S \in \mathbb{G}_1$, we have
$$f(P+Q, R+S) = f(P,R)f(P,S)f(Q,R)f(Q,S).^{3} \quad (1)$$
2. *Non-degeneracy*: If $f(P,Q) = 1$ for all $Q \in \mathbb{G}_1$, then P must be the identity element in $\mathbb{G}_1$.
3. *Computability*: There is an efficient algorithm to compute $f(P,Q)$ for all $P, Q \in \mathbb{G}_1$.

Modified Weil [4] and Tate [5] pairings on supersingular elliptic curves are examples of such bilinear maps, for which the *Bilinear Diffie-Hellman Problem* (BDHP) is believed to be hard, i.e., given $< P, xP, yP, zP >$ for random $x, y, z \in \mathbb{Z}_q^{*}$[4] and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $f(P,P)^{xyz} \in \mathbb{G}_2$ with non-negligible probability.

B. Adversarial model

Adversaries in ad hoc networks can be classified into two categories, namely, *active* adversaries and *passive* adversaries. The former always try to launch more "visible" attacks such as radio jamming or other denial-of-service attacks on the target network without worrying about being caught, and may appear abnormal under many circumstances. Intrusion detection systems or other non-cryptographic methods like frequency hopping, though beyond the scope of this paper, can act as countermeasures against such active adversaries. In contrast, passive adversaries may just perform passive eavesdropping, or inject a small amount of less noticeable packets infrequently to achieve better traffic analysis. However, once locating certain critical nodes through overheard routing information, passive adversaries can mount pinpoint attacks on the victim objects. Therefore, passive adversaries are more dangerous than active adversaries because they are much more "invisible" and difficult to detect. Our purpose in this paper is to provide countermeasures against such passive adversaries.

We assume that passive adversaries can communicate with each other through private and fast communication methods, either wireless or wired. They can collaborate with each other to monitor every radio transmission on every communication link. In addition, they might compromise any node in the target network to become an *internal* adversary. However, we assume that passive adversaries cannot compromise unlimited number of nodes. They do not have unbounded computational capabilities to easily invert and read encrypted messages, and break the above *BDHP*'s hardness assumption either. It is believed that there is no workable cryptographic solutions without this assumption.

III. MASK SYSTEM DESIGN

A. Motivation

Common routing protocols for MANETs, such as AODV and DSR, usually assume nodes have invariant identifiers or network-layer addresses throughout the network lifetime, based on which routing paths are established and data packets are forwarded. As a result, by utilizing the routing information eavesdropped from routing and data packets, adversaries can easily ascertain the identities of the source, the destination, and intermediate nodes involved in one ongoing communication. They are also able to locate one particular node and/or chase its movement. In addition, adversaries can achieve the same purpose through the unique MAC addresses of mobile nodes leaked in MAC frames "flying in the air". For reasons of brevity, we equate node identifiers with their network-layer addresses in the rest of this paper.

One seemingly possible solution is to let mobile nodes dynamically change their identifiers and MAC addresses. However, this may not work in practice because one node

[2]For a given packet, a sender can be its original source or local transmitter, and a receiver can be its final destination or local recipient.

[3]In particular, $\forall\ P, Q \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{Z}_q^{*},\ f(aP, bQ) = f(aP,Q)^b = f(P,bQ)^a = f(P,Q)^{ab}$ etc.

[4]$\mathbb{Z}_q^{*}$ is the *multiplicative group* of integers modulo q. In particular, if q is a prime, $\mathbb{Z}_q^{*} = \{a \mid 1 \leq a \leq q-1\}$.

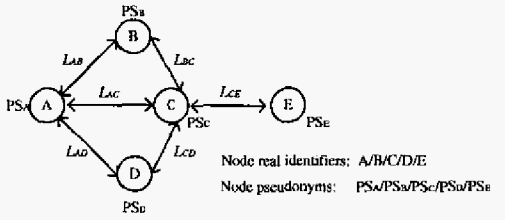

Fig. 1. Anonymous local packet exchange based on link identifiers

has to inform all potential communication partners about its address changes in time, which is rather awkward and may result in the sharp degradation in the routing efficiency.

Based on the aforementioned pairing technique, Balfanz *et al.* [6] proposed the notion of secret handshakes in the sense that nodes can achieve mutual authentication without disclosing their real identities. Motivated by their work, we propose to use node *pseudonyms* instead of their real identifiers in the routing process. More specifically, we require neighboring nodes to establish pairwise unique *link identifiers* based on their exchanged pseudonyms and then route data packets with those link identifiers.

Fig. 1 shows one example of anonymous local packet exchange based on link identifiers, where nodes A/B/C/D/E first exchange their pseudonyms with neighboring nodes and establish pairwise link identifiers as $\{L_{xy}\}$. Those link identifiers are unique in the sense that they are mutually different and any link identifier is only known to the pair of nodes who established it. After that, suppose node *A* sends a MAC frame with a predefined universal address such as all 1's as the source address and L_{AC} as the destination address. Due to the broadcast nature of wireless channels, all its neighboring nodes including *B/C/D* will hear that frame, however, only node *C* will accept it because of its unique sharing of L_{AC} with node *A*. Note that, in this scenario, the real identifiers and MAC addresses of node *A* and node *C* are concealed in the sense that other nodes cannot determine that node *A* and node *C* are the source and destination of one observed transmission. In the later sections, we extend this anonymous local packet exchange to multihop scenarios and design a novel anonymous on-demand routing protocol, called MASK. MASK allows nodes to establish pairwise link identifiers with dynamically-changing pseudonyms and then accomplishes anonymous packet routing and packet forwarding tasks in an efficient manner.

B. System model

We examine an ad hoc network consisting of ξ non-adversary nodes that belong to or have trustable relationship with the same party Ψ ($|\Psi| = \xi$). Non-adversary nodes have common interests and are ready to relay packets for others. Each node has one unique non-zero identifier ID_i ($1 \leq i \leq \xi$). For reasons of brevity, we do not differentiate between ID_i and the i^{th} node in the remainder of this paper.

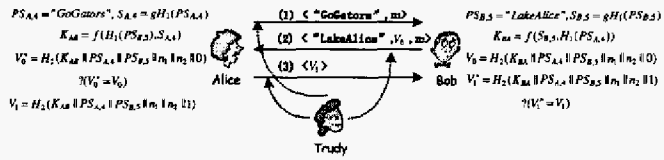

Fig. 2. Anonymous neighborhood authentication

During the bootstrapping phase, a trusted authority (TA) who does not enter the network first determines two q-order cyclic groups $\mathbb{G}_1$ and $\mathbb{G}_2$ as defined in Section II-A, one bilinear map f, and a system master key $g \in \mathbb{Z}_q^*$. It then chooses two collision-resistant cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ mapping arbitrary strings to points in $\mathbb{G}_1$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^\beta$ mapping arbitrary strings to fixed-length strings of β bits, e.g., SHA-1 [7]. In the end, each non-adversary node has the knowledge of the system parameters as $<\mathbb{G}_1, \mathbb{G}_2, f, H_1, H_2>$, but is blind to the system master key g.

Moreover, the TA furnishes each node ID_i with a sufficiently large set $\mathcal{PS}_i$ of collision-resistant pseudonyms and a corresponding *secret point* set as $\mathcal{S}_i = gH_1(\mathcal{PS}_i) = \{S_{i,j}\} = \{gH_1(PS_{i,j}) \in \mathbb{G}_1\}$ ($1 \leq j \leq |\mathcal{PS}_i|$). Given one pseudonym and secret point pair $< PS_{i,j}, S_{i,j} >$, adversaries cannot deduce the system master key g with non-negligible probability due to the hardness in number theory. Besides, there is no one but the TA (not entering the network) can link a given pseudonym to a particular node or identity, or deduce the corresponding secret point.

C. Anonymous neighborhood authentication

We utilize Fig.2 to illustrate the anonymous neighborhood authentication process between two nodes Alice (ID_A) and Bob (ID_B). In the rest of the paper, unless otherwise stated, we will assume that there is a pre-defined universal address such as all 1's, which is used by any node as the source and destination addresses of outgoing MAC broadcast frames to avoid the situation that adversaries can chase one node based on its unique MAC address.

When moving to a new place and intending to achieve mutual authentication with neighboring nodes, Alice pulls out one unused pseudonym, say $PS_{A,4}$ ="GoGators", from her pseudonym set $\mathcal{PS}_A$ and then locally broadcasts it with one random nonce n_1. The reason for using one unused pseudonym is to prevent adversaries from tracing one node based on its invariable pseudonym. Upon seeing such an authentication request and if agreeing to conduct an handshake with node "GoGators", Bob needs to utilize the pseudonym he is currently using (refer to as *active pseudonym* in the rest of the paper), say $PS_{B,5}$="LakeAlice", to calculate a master session key as $K_{BA} = f(S_{B,5}, H_1(PS_{A,4}))$, where $S_{B,5} = gH_1(PS_{B,5})$ is the secret point corresponding to

"LakeAlice". Then Bob broadcasts a reply consisting of $PS_{B,5}$, one random nonce n_2, and an authenticator V_0 computed as

$$V_0 = H_2(K_{BA} \parallel PS_{A,4} \parallel PS_{B,5} \parallel n_1 \parallel n_2 \parallel 0). \quad (2)$$

After receiving Bob's reply, Alice can also calculate a master session key as $K_{AB} = f(H_1(PS_{B,5}), S_{A,4})$, where $S_{A,4} = gH_1(PS_{A,4})$ is the secret point corresponding to "GoGators". According to Eq. 1, if and only if Alice and Bob belong to the same party, they can have

$$K_{BA} = K_{AB} = f(H_1(PS_{B,5}), H_1(PS_{A,4}))^g \in \mathbb{G}_2. \quad (3)$$

Therefore, Alice can easily authenticate Bob by a simple calculation for validating V_0. In order for Bob to ascertain her party membership as well, Alice needs to return her own authenticator V_1 computed as

$$V_1 = H_2(K_{AB} \parallel PS_{A,4} \parallel PS_{B,5} \parallel n_1 \parallel n_2 \parallel 1). \quad (4)$$

Accordingly, Bob can ensure that Alice belongs to the same party after verifying V_1. In the similar manner, other neighboring nodes of Alice can achieve mutual authentication with her.

After a successful handshake, both Alice and Bob can calculate Γ pairs of shared session key (*SKey*) and link identifier (*LinkID*) as

$$\begin{cases} K_{AB}^{\gamma} = H_2(K_{AB}||PS_{A,4}||PS_{B,5}||n_1||n_2||2*\gamma) \\ L_{AB}^{\gamma} = H_2(K_{AB}||PS_{A,4}||PS_{B,5}||n_1||n_2||2*\gamma+1) \end{cases}, \quad (5)$$

where K_{AB}^{γ} and L_{AB}^{γ} ($1 \le \gamma \le \Gamma$) indicate the γ^{th} *SKey* and *LinkID*, respectively, and Γ is a design parameter. Such <*SKey*, *LinkID*> pairs are unique in the sense that collision-resistant hash functions H_1 and H_2, and the bilinear map f ensure no identical pairs would be generated by different pairs of nodes or by the same pair of nodes with different nonces. Moreover, there is even no apparent relationship among the <*SKey*, *LinkID*> pairs generated by the same pair of neighboring nodes with the same pair of nonces.

If the above neighborhood authentication succeeds, Alice knows all her neighbors and will be able to create a *neighbor table* in which each entry contains the pseudonym of a neighbor, the pairwise shared <*SKey*, *LinkID*> pairs, and the index γ of the <*SKey*, *LinkID*> pair that is currently in use. The *LinkID* will be used to identify the packet transmitted between Alice and Bob and the *Skey* can be used to cryptographically protect the content of the packet. Later, when Bob broadcasts a packet identified by L_{AB}^{γ}, Alice knows that the packet is destined for her and can use K_{AB}^{γ} to decrypt the packet if needed, and vice versa. In addition, Alice and Bob should have a simple agreement so they can synchronize the use of the <*SKey*, *LinkID*> pairs. These pairs will be used in the future routing process in an increasing sequence. It means that if the index of the currently-used *LinkID* is γ, the index of the *LinkID* for next packet exchange should be no less than γ. The purpose is to prevent message replay attacks with previously exposed *LinkID*s. Whenever these Γ pairs are used up, Alice and Bob are required to automatically increase both n_1 and n_2 by one and generate new Γ pairs. Hence, the synchronization of <*SKey*, *LinkID*> pairs is implicitly guaranteed.

In the above authentication process, Alice knows that there is a trustable party member in her neighborhood to communicate with, but has no knowledge of the real identifier except one of the public pseudonyms of Bob. So does Bob. If the authentication fails, they reveal nothing but the pseudonyms to each other. Moreover, since only the TA can link a given pseudonym to a particular node, the eavesdropper Trudy learns nothing more than some random strings from the above information exchange. For example, Trudy is blind to the party membership of Alice or Bob, or the specific identifiers of Alice (ID_A), Bob (ID_B), or the party Ψ itself. Trudy cannot calculate the shared <*SKey*, *LinkID*> pairs either due to the hardness of the aforementioned *BDHP*. Therefore, we simultaneously accomplish two seemingly contradictory objectives, namely, authentication and anonymity.

D. MASK: An anonymous routing protocol

Resting on the established link identifiers and session keys, we can implement an efficient anonymous route discovery process, which is illustrated with the exemplary network in Fig. 3.

Anonymous route requests

A communication source S initiates the route discovery for the destination D by locally broadcasting an anonymous route request (ARREQ) packet of the format <*ARREQ*, *ARREQ_id*, *dest_id*, *destSeq*, PS_s>, where *ARREQ_id*[5] is a globally unique value that uniquely identifies an ARREQ, *destSeq* is set to be the last known sequence number for the destination or to be an unknown flag if needed, and PS_S is the active pseudonym of S. Here we ignore the index of PS_S in $\mathcal{PS}_S$ for simplicity.

For an intermediate node not satisfying the ARREQ, it needs to insert an entry into an internal data structure called *reverse route table* where this ARREQ is from and rebroadcasts the ARREQ after changing the embedded pseudonym to its own. This process continues until all the nodes in the network has rebroadcasted the ARREQ once. Different from the traditional on-demand routing protocols, in MASK every node needs to rebroadcast the ARREQ

[5]*ARREQ_id* could be generated by applying a collision-resistant hash function like SHA-1 [7] on the concatenation of node's pseudonym, sequence number, and timestamp.

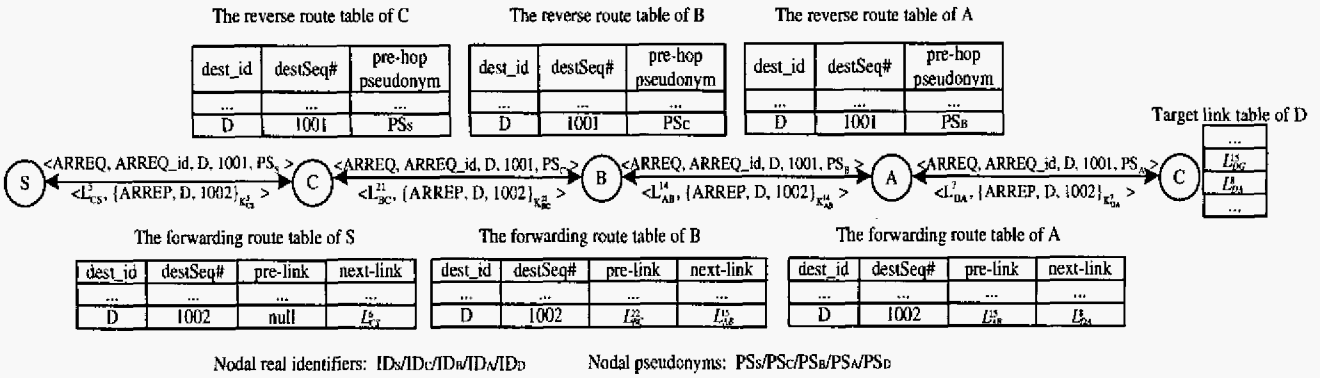

Fig. 3. Anonymous route discovery with a route reply generated by the destination D

once, including the destination node D and any intermediate node who has a valid routing entry to D and generates a reply back to the source. The purpose is to effectively hide the whereabout of the destination node - even though the adversaries know that there is such a node, they will have difficulty to match the *dest_id* to any of the nodes in the network.

Anonymous route replies

An anonymous route reply (ARREP) could be generated and sent back to the source at the destination or at an intermediate node who has a valid path to the destination. Again we use the example in Fig. 3 to illustrate the route replies from the destination.

When an ARREQ arrives at the destination D, D can generate an anonymous route reply (ARREP) which will be unicasted back to the source following the reverse path established before. With the anonymous neighborhood authentication, neighboring nodes have established a set of pairwise shared secret <*Skey, LinkID*> pairs. In our design, the ARREP packet is of format $< LinkID, \{ARREP, dest\_id, destSeq\}_{SKey}>$, where *LinkID* is the next to be used, say L_{DA}^{γ} $(1 \leq \gamma \leq \Gamma)$, shared between D and the pre-hop-pseudonym node A, $\{M\}_{SKey}$ denotes the ciphertext of message M encrypted with corresponding *SKey*, i.e., K_{DA}^{γ} in this case. Therefore, the content of ARREP packet is well protected. The packet is identified by the *LinkID* which only the intended receiver (pre-hop-pseudonym node) will be able to interpret by looking it up in its *neighbor table*. While for a passive eavesdropper, the *LinkID* only appears as some meaningless random number, and he/she has no idea what a particular packet is about and to whom the packet is sent. Moreover, D is required to add $L_{DA}^{\gamma+1}$ to his/her *target link table*. Later on, when seeing a packet identified by $L_{DA}^{\gamma+1}$, D knows that he/she is the end-to-end destination of that packet. It is worth pointing out that the source and destination addresses of the ARREP MAC frame are set to the embedded *LinkID* as well in order to implement anonymous MAC frame exchange.

An intermediate node can also generate a route reply if he/she has one forward route entry for the *dest_id* with *destSeq* equal to or larger than that contained in the received ARREQ. The node needs to prepare an ARREP packet to be sent to its pre-hop-pseudonym node in its reverse route table. Different from the destination, the intermediate node does not need to modify his/her target link table.

For a node that is on the reverse path, say node A, when it receives an ARREP $< L_{DA}^{\gamma}, \{ARREP, dest\_id, destSeq\}_{K_{DA}^{\gamma}}>$ from its next-hop D, node A will discard it if the embedded *destSeq* is smaller than that in its reverse route table. Otherwise, node A will form and transmit a new ARREP $< L_{AB}^{j}, \{ARREP, dest\_id, destSeq\}_{K_{AB}^{j}}>$, where $<K_{AB}^{j}, L_{AB}^{j}>$ is the next to be used <*SKey, LinkID*> pair shared between A and the pre-hop-pseudonym node stored in its reverse route table, which is B in the example. A also needs to update its forwarding route table. If A does not have an entry for *dest_id*, a new entry will be created. Or if the entry for *dest_id* has a smaller *destSeq* than that in the ARREP, the old entry will be replaced with the new information, i.e., *dest_id*, *destSeq*, *pre-link-list*, and *next-link-list* will be set to *dest_id*, *destSeq* in the ARREP, L_{AB}^{j+1}, and $L_{DA}^{\gamma+1}$ respectively, where L_{AB}^{j+1} and $L_{DA}^{\gamma+1}$ denote the next to be used *LinkID*s shared between node A and B and node A and D. If A already has an entry for the *dest_id*, and the new *destSeq* in the ARREP is equal to the old one, A updates the route entry by appending $L_{DA}^{\gamma+1}$ and L_{AB}^{j+1} to the *next-link-list* and the *pre-link-list* field of its forwarding route entry, respectively. Therefore, MASK may simultaneously maintain several next-hop and pre-hop *LinkID*s for one *dest_id* (called *virtual multipath functionality* in this paper) in the forwarding route table. This operation is different from that of AODV [2] in which a node suppresses routing replies with the same destination sequence number. The above process continues until the ARREP reaches the source node S.

Anonymous data forwarding

The data forwarding in MASK is more like a virtual circuit switching process. By looking up in the forwarding route table, the source S picks one *next-LinkID* randomly from the *next-link-list* field in the entry for the destination. A packet is then formed and sent out to the next-hop neighbor who shares the chosen *next-LinkID*. A packet is of format <*next-LinkID, MASK payload*>, where the MASK payload carries other protocol data and application data. Depending on different applications, the MASK payload part can be encrypted and/or integrity-protected using cryptographic methods. Or it can be encrypted by the corresponding *Skey* shared between the two neighboring nodes. As those of ARREP MAC frames, the source and destination addresses of data MAC frames are set to the embedded *LinkID*s as well.

When seeing such a packet, the first intermediate node sharing the embedded *next-LinkID* needs to change the *next-LinkID* field of the packet to one value randomly selected from its *next-link-list* of the forwarding route entry of which the embedded *next-LinkID* matches one of its values in the *pre-link-list*. It then re-unicasts the packet to the chosen next hop. Continuing this process, a packet can finally reach the destination D who will terminate the forwarding as it finds the *next-LinkID* in its target link table.

E. Discussion and more enhancements

Till now, we have presented the basic operations of MASK. In this subsection, we describe some enhancements to the basic operations and discuss more attacks that MASK can withstand.

Message coding attack

The *Message coding attack* happens when adversaries can easily link and trace some packets that do not change their contents or lengths during transmission. Two countermeasures are designed in MASK to cope with this kind of attack. First, random padding on every forwarded packet is used by intermediate nodes to prevent from the attack resulting from the fixed packet length. Intermediate nodes can randomly adjust the length and content of the random padding. Second, the per-hop link encryption method through established pairwise *SKey*s can be used in MASK as well. The purpose here is to make the same packet appear quite different across links.

Flow recognition and message replay attacks

The *Flow recognition attack* occurs when adversaries can recognize packets that belong to a same ongoing communication flow. Notice that in our MASK, a same packet bears completely different and uncorrelated *LinkID*s when transmitted across different hops. Therefore, it is not possible to trace a packet by its *LinkID*. However, if the packets belonging to a single flow always use the same *LinkID* at a same hop, it may reveal some useful information to the adversaries too. Fortunately, the random multipath forwarding of MASK can partially mitigate this attack. In fact, an intermediate node works as a multiplexer which takes inputs from multiple pre-links and mixes them together and sends them out to multiple next-links. In addition, we request that two neighboring nodes automatically change their currently-used shared *LinkID* either on a per-packet basis or periodically. By doing this, MASK leaves the adversaries a dynamic changing set of *LinkID*s for the same flow and at each hop. Moreover, dynamically changing *LinkID*s effectively thwart the *message replay attack* in which the adversaries try to replay an old message repeatedly in order to see the repeated pattern of packet forwarding.

Timing analysis attack

Suppose adversaries can divide the monitored area into small cells. They might ascertain that one source or destination exists in one cell by observing that no packets come into or out of that cell during a certain time interval, while some packets come out of or into that cell. In addition, in IEEE 802.11-type ad hoc networks, adversaries might guess that two consecutive radio transmissions belong to the same communication flow. These attacks belongs to the category of the *timing analysis attack*.

In MASK, packets transmitted in the air are only identified by anonymous *LinkID*s. When network traffic load is high and every node is busy in transmitting and receiving, all the transmissions will be mixed together which leads to very difficult timing analysis. However, when the traffic load is light, several precautions need to be taken against the alleged timing analysis attack. First, when one destination receives a packet destined for it, it can forge a packet with a fake *LinkID* and forward it further, by doing so it tries to fool the adversaries into belief that one observed radio transmission does not end at the destination. The destination can also use genuine *LinkID*s to ask its trustful neighbors to help further enlarge the suspicious area of adversaries. Second, a packet needs to wait a random amount of time to be forwarded so that an earlier arriving packet may be forwarded after a later comer. Last, even without involved in any communications, nodes can send dummy packets with fake *LinkID*s at random intervals to increase the difficulty of adversaries in determining the originating and terminating areas of observed radio transmissions. The purpose here is to introduce more randomness of the radio transmissions so that the real traffic pattern can be concealed.

Node compromise attack

Adversaries might depend on one single compromised node(s) to launch several types of attacks. First, the compromised node can freely perform anonymous neighborhood

authentication with others, based on which to beguile normal nodes into disclosing their real identifiers. Assume that normal nodes do not reveal their real identifiers to others except some critical nodes such as captains or generals even when the anonymous authentication succeeds. What the compromised nodes can get is just some unmeaning pseudonyms. One may wonder that if the compromised node is a critical node, he/she can learn the real identifiers of certain neighboring nodes. However, it does not help much because the beguiled nodes may move to another place and switch to other pseudonyms. As we mentioned before, adversaries cannot link one given pseudonym to a particular node(identifier). Second, if the compromised node lies on the forwarding path from the source to the destination, he/she may only know that a packet is transmitted to the destination. But if the above countermeasures against timing analysis attack are applied, he/she does know where and which node the destination is, even when the destination is his/her neighbor.

We notice that there is an extreme case that a packet source or destination is all surrounded by compromised nodes. Under this rare circumstance, the above countermeasures against timing analysis attack do not take effect and adversaries can ascertain the location of the source or the location and identifier of the destination, depending which one of the source and destination is in trap. Currently, we have no better way to deal with this worst case. Fortunately, node mobility can help mitigate this attack in that the source or destination may quickly move out of the area full of compromised nodes. To further chase victim nodes, compromised nodes have to actively move as a group, which makes them run a high risk of exposing themselves.

IV. Related work

To the best of our knowledge, there are only two publications that are closely related to our work. Kong and Hong [3] first demonstrated that existing ad hoc routing protocols are subject to so-called passive attacks in the sense that the locations and movement patterns of nodes can be traced, and proactive and reactive ad hoc routes across multiple nodes can be visualized by collaborative efforts of adversaries. To deal with such passive attacks, they proposed an anonymous on-demand routing protocol named ANODR [8], which provides the source and destination anonymity at the cost of complicated and computationally expensive cryptographic operations. ANODR requires each source-destination pair to share pairwise secret information. In addition, as the authors mentioned, another limitation of ANODR is its sensitivity to node mobility, which may result in the sharp degradation of routing efficiency in face of node mobility.

V. Conclusion

In this paper, we proposed the notion of anonymous handshakes to enable anonymous one-hop and multi-hop communications in mobile ad hoc networks, i.e., hiding the identities of participating nodes involved in one ongoing communication. We first developed an anonymous neighborhood authentication protocol which provide secure yet anonymous mutual authentication between neighboring nodes without need of their real identifers. We then presented a preliminary version of an anonymous on-demand routing protocol, called MASK, which can nicely fulfill the routing and packet forwarding tasks without disclosing the identities of participating nodes.

As the future research, we will first evaluate the performance of MASK through simulations and more practical field studies. We will then plan to combine MASK with other secure routing schemes to provide an anonymous yet secure routing protocol.

VI. Acknowledgement

This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and under grant N000140210554.

References

[1] D.B. Johnsom, D.A. Maltz, and Y. Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). <draft-ietf-manet-dsr-09.txt>, Apr. 2003.

[2] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July 2003.

[3] J. Kong, X. Hong, and M. Gerla. A New Set of Passive Routing Attacks in Mobile Ad Hoc Networks. In *IEEE Milcom*, Oct. 2003.

[4] D. Boneh and M. Franklin. Identify-based encryption from the Weil pairing. In *Proc. CRYPTO 01*, pages 213-219, Springer-Verlag, 2001.

[5] P. S. L. M. Barreto, H. Y. Kim, B. Bynn, and M. Scott. Efficient algorithms for pairing-base cryptosystems. In *Proc. CRYPTO 02*, Springer Verlag, August 2002.

[6] D. Balfanz, G. Durfee, and N. Shankar et al. Secure Handshakes from Pairing-Based Key Agreements. In *IEEE Symposium on Security & Privacy*, Oakland, CA, May 2003.

[7] A.J. Menezes, P.C. van Oorschot, S.A. Vanston: Handbook of Applied Cryptography, CRC Press, ISBN 0-8493-8523-7, 1996.

[8] J. Kong and X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. In *ACM MobiHoc*, Annapolis, Maryland, June 2003.