Anonymous Communications in Mobile Ad Hoc Networks

Yanchao Zhang*, Wei Liu* and Wenjing Lou†
*Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611
Email: {yczhang@, liuw@}ufl.edu
†Department of Electrical and Computer Engineering
Worcester Polytechnic Institute, Worcester, MA 01609
Email: wjlou@ece.wpi.edu

***Abstract*— Due to the broadcast nature of radio transmissions, communications in mobile ad hoc networks (MANETs) are more susceptible to malicious traffic analysis. In this paper we propose a novel anonymous on-demand routing protocol, termed MASK, to enable anonymous communications thereby thwarting possible traffic analysis attacks. Based on a new cryptographic concept called pairing, we first propose an anonymous neighborhood authentication protocol which allows neighboring nodes to authenticate each other without revealing their identities. Then utilizing the secret pairwise link identifiers and keys established between neighbors during the neighborhood authentication process, MASK fulfills the routing and packet forwarding tasks nicely 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. Moreover, MASK preserves the routing efficiency in contrast to previous proposals. Detailed anonymity analysis and simulation studies are carried out to validate and justify the effectiveness of MASK.**

I. Introduction

Mobile ad hoc networks (MANETs) are finding ever-increasing applications in both military and civilian systems due to their self-configuration and self-maintenance capabilities. Many of these applications are security sensitive, such as military battlefield operations, homeland security scenarios, law enforcement, and rescue missions. As a result, security in MANETs has drawn intensive attention recently [1].

Traffic analysis is one of the most subtle and unsolved security attacks against MANETs. By definition, traffic analysis is a security attack where an adversary observes network traffic in order to infer sensitive information about the applications and/or the underlying system [2]. Adversaries aim to learn the identities of communicating parties and acquire information such as network traffic patterns[1] and/or traffic pattern changes. The leakage of such information is often devastating in security-sensitive scenarios. For example, an unexpected change of the traffic pattern in a military network may indicate a forthcoming action, a chain of commands, or a state change of network alertness [3]. It may also reveal the locations of command centers or mobile VIP nodes, which will enable the adversaries to launch the pinpoint attacks on them. In contrast to *active attacks* which usually involve the launch of denial-of-service or other more "visible" and aggressive attacks on the target network, traffic analysis is a kind of *passive attack* which is "invisible" and difficult to detect. It is, therefore, important to design countermeasures against such malicious traffic analysis.

The shared wireless medium of MANETs introduces opportunities for passive eavesdropping on data communications. Adversaries can easily overhear all the messages "flying in the air" without physically compromising a node. Several methods have been investigated to withstand eavesdropping and further the traffic analysis. One attempt is to prevent the wireless signals from being intercepted or even detected by developing some LPI/LPD (low probability of interception/low probability of detection) communication techniques. Examples of such techniques include the spread-spectrum modulation, effective power control, and directional antennas [4]. However, it is impossible to completely avoid signal detection in the open wireless environments. The second one relies on the use of traffic padding, i.e., introducing dummy packets into the network [5] to camouflage the real traffic pattern. However, this approach adds significant extra load to the network and consumes the scarce network resources. A third method is to perform end-to-end encryption and/or link encryption on data traffic. However, it only prevents adversaries from accessing traffic contents. Adversaries can still carry out traffic analysis based on the bare network-layer and/or MAC addresses, both of which are unprotected and unencrypted in common ad hoc routing protocols such as AODV [6] and DSR [7] and the de facto MAC protocol IEEE 802.11.

In this paper, we shift our attention to a new paradigm, i.e., designing anonymous communication protocols for MANETs. The essence of anonymous communications is to hide sender and/or receiver's identities from outside observers. As a result, adversaries cannot correlate eavesdropped traffic information to actual network traffic patterns so that traffic analysis attack can be efficiently defeated.

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

[1] A network traffic pattern can be represented by a set of (source, destination, average rate) 3-tuples with each describing one flow [5]. A flow can be either a end-to-end flow between any pair of nodes in the network or a local link flow between two neighboring nodes.

0-7803-8968-9/05/$20.00 (c)2005 IEEE

The contribution of this paper is the design of a novel anonymous on-demand routing protocol for MANETs, called MASK, which nicely fulfills the routing task without disclosing the real identities of participating nodes. The basic idea of MASK is (1) the anonymous neighborhood authentication based on the *dynamically changing pseudonyms* of nodes instead of their real identifiers or network-layer addresses and/or MAC addresses; and (2) the anonymous route discovery and data forwarding based on the pairwise shared *link identifiers (LinkIDs)* between neighbors which are established during the neighborhood authentication. More specifically, MASK is designed to meet the following objectives:

- *Sender-, receiver-, and relationship anonymity*. Sender or receiver anonymity means the concealment of who sending or receiving a particular packet, and relationship anonymity indicates the concealment of who talking to whom. 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. With MASK in place, although adversaries might see a packet flying in the air, they will not be able to determine the packet is from whom to whom (point-to-point transmission) in terms of node identifiers, neither can they determine the two end systems of a conversation (end-to-end communication).
- *Untraceability and unlocatability.* Adversaries cannot trace a particular packet back to its source or trace it forward to its destination.
- *Anonymous yet secure neighborhood authentication*. With MASK, any pair of neighboring nodes can achieve mutual authentication without disclosing their real identifiers.
- *Low cryptographic overhead and high routing efficiency*. The computational overhead introduced by cryptographic operations should be low, which is achieved by utilizing a new cryptographic concept called *pairing* [8], efficient hash functions, and symmetric-key algorithms. In addition, MASK should achieve comparable routing efficiency to classic ad hoc routing protocols such as AODV [6].
- *Resistance to a wide range of adversarial attacks*. MASK can withstand a wide range of attacks, including message coding attack, flow recognition attack, timing analysis attack, etc.

The rest of this paper is structured as follows. Section II describes the cryptographic tools, the adversarial model, and the network model used in this paper. Section III details the MASK design and analyzes its anonymity property. Section IV evaluates the computational overhead and routing efficiency of MASK through simulation studies. Section V reviews the related work. Finally, Section VI presents some concluding remarks.

II. Preliminaries and Models

A. Pairing Concept

Pairing has recently found a number of interesting applications in cryptography, e.g., [8]–[10], 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. 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
$$f(P+Q, R+S) = f(P,R)f(P,S)f(Q,R)f(Q,S).^{2} \quad (1)$$
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 [8] and Tate [9] 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 $x, y, z \in \mathbb{Z}_q^{*3}$ 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. We refer readers to [8], [9] for further details on pairing. An exemplary implementation of pairing can be found in Section IV-A.

B. Adversarial Model

We observe that there might be two kinds of adversaries in ad hoc networks, 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 may 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

[2] 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.

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

break the above *BDHP*'s hardness assumption either. It is believed that there is no workable cryptographic solutions without this assumption.

There is a rich literature on secure routing algorithms for MANETs, e.g., [11]–[13], aiming to secure route discovery and maintenance processes. Though important, the secure routing problem is orthogonal to the anonymous routing problem we focus on in this paper. For the lack of space, we leave the discussion on their interactions in another separate paper and assume that adversaries do not aggressively falsify or forge routing messages.

C. Network Model

We assume that each node has limited transmission and reception capabilities. Nodes within the transmission range of each other are called neighboring nodes. Non-neighboring nodes communicate with each other via multi-hop and unreliable wireless links. In addition, we assume that wireless links are symmetric in the sense that if node *X* can hear another node *Y*'s transmission, *Y* can also hear *X*'s transmission. Furthermore, we assume that each node can run its medium access control (MAC) interface in the "promiscuous" mode to receive all the MAC frames broadcasted in its neighborhood. For example, Lucent Technologies' WaveLAN interfaces have such a capability. Moreover, each node is capable of manipulating the source addresses of its outgoing MAC frames. This assumption is prerequisite for preventing traffic analysis, otherwise adversaries can easily identify and trace a node based on its unique MAC address.

III. MASK System Design

A. System Model

We consider an ad hoc network consisting of ξ non-adversary nodes that belong to or have trustable relationship with the same party Ψ ($|\Psi| = \xi$[4]). In this paper we do not consider node selfishness [14] and assume that 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. We assume that nodes may freely roam in the network, but do not continuously move so rapidly as to make the flooding of every individual data packet the only possible routing protocol.

During the bootstrapping phase, a trusted authority (TA), e.g., the system administrator or network planner (not entering 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 $g \in \mathbb{Z}_q^*$. He/she 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 β-bit fixed-length output, e.g., SHA-1 [15]. 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.

[4] $|\Psi|$ is the cardinality of Ψ and can be dynamically changing with node addition/reduction.

In MASK, nodes use pseudonyms instead of their real identifiers in the routing process. If one node uses one pseudonym all the time, it won't help to defend against traffic analysis because the pseudonym will be analyzed the same way as the real identifier. Therefore, each node should use dynamically changing pseudonyms. For this purpose, the TA furnishes each node ID_i with a sufficiently large set $\mathcal{PS}_i$ of collision-resistant pseudonyms[5] and a corresponding *secret point set* as $\mathcal{S}_i = gH_1(\mathcal{PS}_i) = \{S_{i,j}\} = \{gH_1(PS_{i,j}) \in \mathbb{G}_1\}$ ($1 \leq j \leq |\mathcal{PS}_i|$). Since the discrete logarithm problem (DLP)[6] is believed to be hard in $\mathbb{G}_1$ [8], 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. In addition, there is no one but the TA can link a given pseudonym to a particular node or identity, or deduce the corresponding secret point with non-negligible probability.

B. Anonymous Neighborhood Authentication

By definition, anonymous neighborhood authentication means that two neighboring nodes can ensure that they belong to the same party or have trustable relationship with each other without revealing their either real identifiers or party membership information. Notice that the secrecy of nodal party membership is equally important as nodal identifiers. For example, if one node is a CIA agent, it would be dangerous to release this information to strangers. There are three conventional authentication approaches in large-scale MANETs. The first one is to use a network-wide key shared by all the nodes [17], but this approach is vulnerable to single node compromise. The second one is to let each node share pairwise keys with all the other nodes in the network, but it suffers from the lack of scalability because it may need $\frac{\xi(\xi-1)}{2}$ keys to bootstrap a network with ξ nodes. The third one relies on the use of public-key certificates, based on which any two nodes can achieve mutual authentication through challenge-response by using public-key decryption or digital signatures [18]. However, authentication based on public-key certificates may inevitably disclose either nodal identity or party membership information or both that is implied or embedded in public-key certificates. For example, to correctly verify the other one's certificate, one node has to know the authentic public key of the CA that generates the certificate to be verified. This would cause the disclosure of a node's party membership, i.e., from which CA it obtains the certificate. Therefore, certificate-based authentication is not appropriate for achieving anonymous neighborhood authentication either. In the following, we illustrate how to utilize the aforementioned pairing concept to implement anonymous neighborhood authentication and accordingly establish pairwise shared link keys and link identifiers. Our scheme is a simple adaptation of Balfanz *et al.*'s scheme [10] to the mobile setting.

[5] One possible way to implement those node identifiers is to use statistically unique cryptographically verifiable (SUCV) addresses [16].

[6] The DLP in the additive group $\mathbb{G}_1$ is as follows: given two group elements P and Q, find an integer $n \in \mathbb{Z}_q^*$ such that $Q = nP$ whenever such an integer exists.

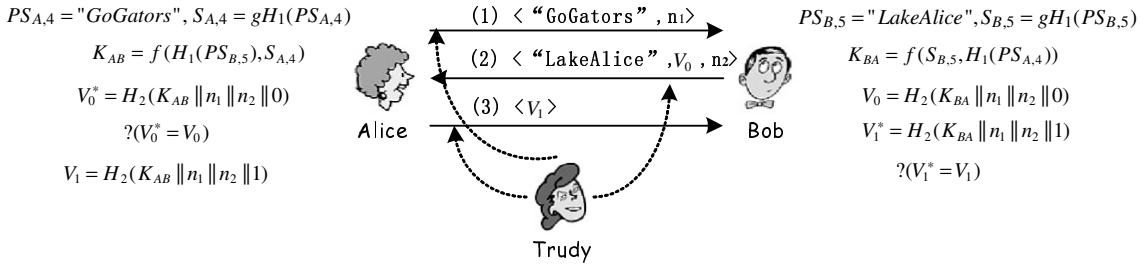


Fig. 1. Anonymous neighborhood authentication.

Fig. 1 shows the authentication process between two nodes Alice and Bob. For the ease of explanation, we assume that Alice's identifier is ID_A and Bob's identifier is ID_B instead of integer-indexed identifiers like ID_i. In the rest of the paper, unless otherwise stated, we will assume that there is a predefined universal address such as all 1's, which is used by any node as the source and destination addresses of outgoing MAC broadcast frames such as broadcast frames for routing requests described later.

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. Upon seeing such an authentication request and if agreeing to conduct a handshake with node "GoGators", Bob needs to utilize the pseudonym he is currently using (refer to as *active pseudonym* hereafter), 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

$$V_0 = H_2(K_{BA} \parallel n_1 \parallel n_2 \parallel 0). \tag{2}$$

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

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

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

$$V_1 = H_2(K_{AB} \parallel n_1 \parallel n_2 \parallel 1). \tag{4}$$

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. Notice that if all the neighboring nodes simultaneously send replies to the same request broadcast from node A, a possible collision may occur. In this paper, we assume the reliable transmission of such authentication requests/replies. It can be achieved for instance through MAC-layer retransmission or by using a random jitter delay for which each node has to wait before responding to an authentication request.

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

$$\begin{cases} K_{AB}^{\gamma} = H_2(K_{AB} \parallel n_1 \parallel n_2 \parallel 2 * \gamma) \\ L_{AB}^{\gamma} = H_2(K_{AB} \parallel n_1 \parallel n_2 \parallel 2 * \gamma + 1) \ , \end{cases} \tag{5}$$

where K_{AB}^{γ} and L_{AB}^{γ} ($1 \leq \gamma \leq \Gamma$) 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.

Through the same procedure, Alice knows all her authentic neighbors and will be able to create a *neighbor table* in which each entry contains the pseudonym of one neighbor, the pairwise shared <*SKey*, *LinkID*> pairs, and the index γ of the <*SKey*, *LinkID*> pair that is currently in use. The *LinkID*s will be used to identify the packets transmitted between Alice and Bob and the *Skey*s can be used to encrypt, integrity-protect, or authenticate the content of the packets if needed. 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 addition, since the generation of these pairs involve only efficient hash functions, it is an inexpensive operation, as will be shown in Section IV-A.

In the above authentication process, Alice knows that there

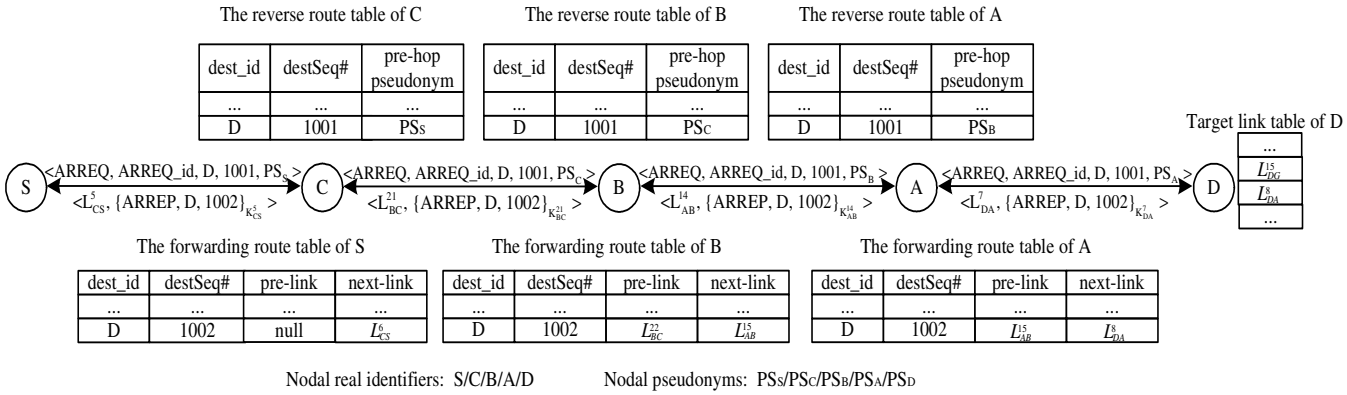


Fig. 2. Anonymous route discovery with a route reply generated by the destination *D*.

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, which may occur for instance when one of Alice and Bob is an adversary impersonating a legitimate node, 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.

C. Anonymous Route Discovery

With the anonymous neighborhood authentication, neighbors can authenticate each other and establish <*SKey*, *LinkID*> pairs which are pairwise shared secrets between them. In this subsection, we present an anonymous route discovery process which makes use of the <*SKey*, *LinkID*> pairs and is able to find routes between a source and a destination on demand anonymously. And we use the the exemplary network in Fig. 2 for illustration purpose.

Besides the neighbor table, each node also maintains the following data structures:

- *Forwarding route table*: A table consisting of entries of format <*dest_id*, *destSeq*, *pre-link-list*, *next-link-list*>, where *dest_id* is the real identifier of the destination node and *destSeq*[7] is the corresponding node sequence number. The *pre-link-list* is the set of pre-hop link identifiers (*pre-LinkID*) from which packets destined for *dest_id* may come, and *next-link-list* is the set of next-hop link identifiers (*next-LinkID*) to which packets destined for *dest_id* are supposed to be forwarded.
- *Reverse route table*: A table consisting of entries of format <*dest_id*, *destSeq*, *pre-hop-pseudonym*>, based on which route replies are relayed back to the source.
- *Target link table*: A table consisting of selected link identifiers shared with neighbors. The current node is the final destination (end-to-end) for the packets bearing the linkIDs which are in its target link table.

There will be an appropriate timer associated with each entry of the above tables. And an entry should be recycled when its timer expires.

Anonymous Route Requests

Similar to other on-demand routing protocols, our anonymous route discovery starts from broadcasting route request messages when a node has a packet to a certain destination but it does not know a path to that destination. The anonymous route request (ARREQ) packet has the format <*ARREQ*, *ARREQ_id*, *dest_id*, *destSeq*, PS_s>, where *ARREQ_id*[8] 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.

When an intermediate node, say node *C*, receives an ARREQ message for the first time, it inserts an entry into its reverse route table where this ARREQ comes from, and then rebroadcasts the ARREQ after changing the embedded *pseudonym* field to its own. ARREQs with previously seen *ARREQ_id*s [9] are simply discarded. 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 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.

[7]The maintenance of node sequence numbers strictly follows the steps defined in AODV [6].

[8]*ARREQ_id* could be generated by applying a collision-resistant hash function like SHA-1 [15] on the concatenation of node's pseudonym, sequence number, and timestamp.

[9]Note that ARREQ flooding is supposed to be finished in a limited period so that each node does not need to keep too many old *ARREQ_id*s.

It is worth noting that in the propagation of ARREQs, the real identifiers of the source and intermediate nodes are concealed, while the real identifier of the destination (*dest_id*) has to be exposed. In the traditional route discovery by flooding, the destination node does not need to rebroadcast the route request message. However, that design allows the adversary to identify the destination node easily by tracking the activities at each node - every node broadcasts the message once except the destination and/or some nodes knowing the paths to the destination. Therefore, in our design, every node, including the destination, needs to rebroadcast the ARREQ message once. This will 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. Note that the overhead introduced by this modification is minimal - in a route discovery protocol using flooding, every node need to broadcast once anyway except the destination and the nodes that already have a path to return to the source. So the extra overheard introduced is one or a few more transmissions by the destination and the intermediate nodes who can reply.

Anonymous Route Replies

An anonymous route reply (ARREP) packet can 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. 2 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 under the corresponding *SKey*, i.e., K_{DA}^{γ} in this case, with any efficient symmetric cipher such as RC6 [19]. 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 its *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 both 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 it 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 its 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 *B* 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 [6] 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*. An exemption in the route reply process is that, in MASK, since each node is required to rebroadcast the ARREQ message no matter it replies or not, the ARREPs coming back to an intermediate node who replied before may present inconsistent state information which may cause routing loops. Therefore, we require that the intermediate nodes who already replied ignore the route replies with the same *destSeq*.

Notice that in the route reply process, all the ARREP packets are encrypted and identified by the *LinkID*s which are only interpretable by the intended local receivers. A passive eavesdropper might see discrete transmissions everywhere but he/she will not be able to tell the content of a particular transmission, neither can he/she tell who is transmitting and who is receiving. For an internal adversary who happens to reside in the reverse route to the source, what he/she can learn is the identifier of the destination, but not which and where that destination is, even when the destination is his/her neighbor because of the anonymous neighborhood authentication.

D. 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

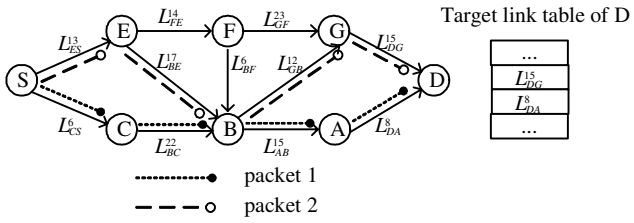


Fig. 3. Anonymous packet forwarding from S to D.

<*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 end-to-end encrypted and/or integrity-protected using cryptographic methods. Or it can be encrypted and authenticated by the corresponding *Skey* shared between two neighboring nodes. As those of ARREP MAC frames, the source and destination addresses of data MAC frames are both 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.

An example of anonymous packet forwarding is depicted in Fig. 3, in which a set of forwarding links, denoted by directional solid lines have been established, and each is labelled by its *LinkID*. As we can see, due to the random selection of *next-LinkID*s at each intermediate node, MASK has the nice *Traffic Rerouting* property [2] that packets of the same flow may travel through different paths to the destination. It makes it more difficult for adversaries to correlate the observed radio transmissions with each other and to acquire the actual network traffic patterns. And it also makes it more difficult for adversaries to trace one packet from the source to the destination. The drawback is that MASK does not always use the best path, e.g., the shortest-hop path, for packet forwarding. Another drawback is that MASK may introduce extra delay and/or delay jitter. However, for security-sensitive applications demanding anonymity, we argue that this tradeoff of routing efficiency for anonymity is acceptable.

When all the *next-LinkID*s for one destination become unavailable due to mobility or other reasons, one node needs to locally broadcast an anonymous route error (ARRER) packet of format <*ARRER*, *pre-link-list*> to inform its up-stream nodes. Any neighboring node who has the *LinkID* in the received *pre-link-list* should remove the *LinkID* from the *next-link-list* field of its corresponding forwarding route entry. If its *next-link-list* becomes empty as well, it should also locally broadcast a similar ARRER packet. When the source has no available *next-LinkID* for the destination, it should restart the anonymous routing discovery.

E. Discussion and More Enhancements

Up to now, we have described the basic operations of MASK with a focus on how to provide anonymity in neighborhood authentication, route discovery, and packet forwarding. In what follows, we describe some enhancements to the basic operations and discuss more attacks that MASK is able to defend against.

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 mentioned in Section III-D 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 [5] 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.

F. Anonymity Analysis

Here we analyze how well MASK meets the design objectives listed in Section I. We assume that there are two types of adversaries, namely, Type I – external eavesdroppers or internal adversaries (cf. Section II-B) not on any forwarding path, and Type II – internal adversaries residing on the forwarding paths. We use "conditional" anonymity to indicate the case that adversaries may know the sender and/or receiver identifiers of a particular packet but can not match the identifier to a particular node. We use "unconditional" anonymity to indicate the case that adversaries know neither of the sender and receiver identifiers.

First of all, anonymous neighborhood authentication guarantees that any two neighboring nodes can establish an anonymous yet secure link without revealing their identifiers. And both routing packets and data packets are locally exchanged between two neighboring nodes with the established *LinkID*s rather than their real identifiers. Except the two points constituting the link, the *LinkID*s do not provide any information to a passive observer. Therefore, MASK provides unconditional local transmitter and recipient anonymity, and also the relationship anonymity between the local transmitter and recipient against both types of adversaries.

During the route discovery and data forwarding phases, the real identifiers of the source and intermediate nodes are well concealed by their pseudonyms. As a result, MASK guarantees the unconditional anonymity of the source and intermediate nodes against both types of adversaries as well.

To implement an on-demand route discovery process guaranteeing the unconditional destination anonymity, the only known approach, to the authors's knowledge, is to utilize a so-called cryptographic "trapdoor" [23]. That is, the source sends a route request including a global trapdoor instead of the destination's identifier. A trapdoor can only be correctly opened by the desired destination and only the destination is allowed to generate a route reply after correctly opening and verifying the trapdoor. There are three major concerns with this approach. First, how to efficiently implement such kind of global trapdoors without contradicting the anonymity requirement is a rather challenging task in resource-constrained MANETs. Second, the route discovery process is computationally intensive because each intermediate node has to try to open the "trapdoor" to see if it is the desired destination, which often involves expensive certificate-based public-key operations. Last, since only the destination can generate a route reply, the well-known routing optimizations based on intermediate node routing caches such as those in AODV and DSR cannot be applied. Whenever a route is broken due to node mobility or other reasons, the source has to restart the expensive route discovery process.

In contrast, MASK provides conditional destination anonymity by utilizing the destination's identifier in ARREQs to achieve much better routing efficiency. During the propagation of ARREQs, both types of adversaries can know that an ARREQ is issued for the *dest_id* and hence MASK only provides conditional destination anonymity. Since following ARREPs are cryptographically protected, only Type II adversaries can link them with previously seen ARREQs. During the data forwarding phase, there is no destination identifier used and only Type II adversaries know the destination identifiers of forwarded packets identified by *LinkID*s. As a result, MASK provides unconditional anonymity against Type I adversaries and conditional anonymity against Type II adversaries in both phases. The resulting benefit from sacrificing unconditional destination anonymity is that MASK has the similar routing structure to that of AODV so that the well-known routing optimizations can be applied. We believe that such a trade-off of unconditional destination anonymity and poor routing efficiency for conditional destination anonymity and efficient routing performance is often necessary to accomplish both anonymous communication and efficient on-demand routing in MANETs, especially when packet sources such as generals desire more anonymity protection than destinations such as soldiers.

Furthermore, MASK ensures the *unlocatability* of nodes in that nodes do not reveal their real identifiers to other nodes, and they change their pseudonyms dynamically. Therefore, for a given node identifier, both types of adversaries cannot easily determine which node and where the corresponding node is. Moreover, since source identifiers are never disclosed in the routing discovery and packet forwarding phases, MASK ensures the relationship anonymity between the source and destination against both types of adversaries.

IV. Performance Evaluation

A. Cryptographic Implementation

The cryptographic operations in MASK mainly consist of two parts, i.e., the anonymous neighborhood authentication and the hop-by-hop link encryption/decryption of routing replies (ARREPs) and data packets.

The bilinear map f we used is the Tate pairing, with some of the modifications and performance improvements described

TABLE I

PROCESSING TIMINGS OF CRYPTOGRAPHIC OPERATIONS.

Item	Processing timings
Tate paring	8.5 ms
SHA-1	18.980 MB/S
Computation of <SKey,LinkID> pairs	2.4 ms (for 1000 pairs)
RC6	7.111 MB/S

in [9]. The elliptic curve E we used is $y^2 = x^3 + x$. The aforementioned group order q was a large 160-bit prime, based on which we generated another 512-bit prime $p = 12qr - 1$ (for some r large enough to make p be the correct size)[10]. Then $\mathbb{G}_1$ was a cyclic subgroup of the additive group of points of the elliptic curve E over the finite field $\mathbb{F}_p$, while G_2 was a cyclic subgroup of the multiplicative group associated to the finite field $\mathbb{F}^*_{p^2}$.

To implement the collision-resistant hash function H_1 in Section III-A, we simply inputted a given string into SHA-1 [15] to get the x-coordinate of a point, and then generated the corresponding y-coordinate according to the elliptic curve equation. In addition, we used SHA-1 to implement the other collision-resistant hash function H_2 as well. Moreover, we chose the highly efficient symmetric algorithm RC6 [19] as the hop-by-hop link encryption/decryption method applied to ARREPs and data packets.

We evaluated the computation costs of the critical cryptographic operations in MASK on a Pentium III 1 GHz processor under Windows 2000, in which SHA-1 and RC6 were evaluated using the Crypto++ Library 5.1 [20], and the Tate pairing was evaluated within the MIRACL Library [21]. For convenience only, we assume the length of node pseudonyms to be 8 bytes, the length of nonces used in MASK to be 4 bytes, the length of *LinkID*s or *SKey*s to be 20 bytes, and the length of Γ to be 2 bytes. In fact, the impact of larger values of those items on the computation results is negligible.

From Table I, we can see that the most time-consuming operation is the Tate pairing required by the neighborhood authentication. However, for a protocol demanding anonymous authentication in such dynamic environments as MANETs, the Tate pairing seems to be an indispensable operation (cf. Section III-B). In addition, since the pairing is a relatively new concept, we anticipate that the evaluation cost of the pairing will be much reduced with the rapid advance in the realm of number theory. For example, Barreto *et al.* [22] proposed an approach to evaluate the Tata pairing by up to 10 times faster than previous approaches. Currently, we are working on implementing their algorithm in low-end embedded devices.

The Tate pairing only needs to be performed once for one pair of neighboring nodes, and then the result can be fed into the fast SHA-1 hash function for the future computation of shared *LinkID*s and *SKeys*. Suppose one node is required to maintain $\Gamma = 1000$ <*SKey*, *LinkID*> pairs shared with one neighbor. The computation of such 1000 pairs only costs around 2.4 ms. Hence, when two neighboring node runs out of the established shared <*SKey*, *LinkID*> pairs, they can generate new Γ pairs instantly. Moreover, the hop-by-hop link encryption/decryption are not time-consuming and can be done in a very fast manner.

Therefore, although we introduce some cryptographic operations into MASK to provide the desirable anonymity property, the resulting computation overhead and end-to-end packet delay are affordable.

[10]According to [10], such bit-length configurations of p and q provide a comparable level of security to RSA cryptography with a key size of 1024 bits.

TABLE II

AVERAGE SPEEDS OF VARIOUS SPEED RANGES (UNIT: M/S).

Speed range	$\bar{V}$	Speed range	$\bar{V}$
[1,4]	2.16	[2,29]	10.10
[1,11]	4.17	[4,27]	12.04
[1,19]	6.11	[6,27]	13.96
[1,28]	8.10	[8,28]	15.96

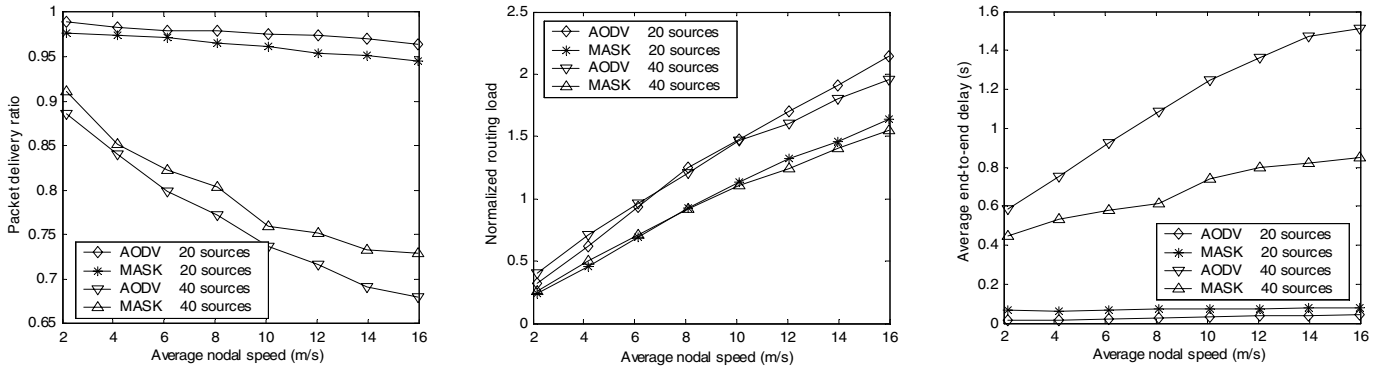
B. Communication Performance

1) Simulation setup: We conducted simulations to evaluate the performance of MASK in terms of routing efficiency. The simulation was conducted within GloMoSim V2.03. The physical-layer path loss model is the two-ray model. The MAC layer protocol used is the Distributed Coordination Function (DCF) of IEEE 802.11. The radio propagation range for each node is 250 meters and the channel capacity is 2 Mbps.

We simulated an ad hoc network with 50 node uniformly deployed in a 700 m $\times$ 700 m square field. To emulate node mobility patterns, we improved the random waypoint model in GloMoSim library according to the approaches presented in [25], which guarantee the convergence of the average nodal speed within the simulation time. In particular, initial speeds of nodes are chosen from the steady-state distribution, and subsequent speeds uniformly from the designated speed range. Table II shows various speed ranges and corresponding average speeds ($\bar{V}$) calculated according to [25]. Among those pairs of ($[V_{min}, V_{max}]$) that converge to the proximate average speed of the integer value, we chose the maximum range to increase the variation of nodal speed within the same simulation. In addition, the pause time is set to be zero in our simulations, meaning nodes are always moving.

CBR sessions are used to generate network data traffic and various number of sources are used to simulate different offered load. All the data packets are 512 bytes and are sent at a speed of 4 packets/second, unless otherwise stated. Each simulation is executed for 15 simulated minutes. Each data point represents an average of ten runs with identical traffic models, but different randomly generated mobility scenarios. For fairness, identical mobility and traffic scenarios are used across protocols when performance comparison is involved.

We compared the routing performance of MASK with AODV [6] with regard to three commonly used metrics:(1) *Packet delivery ratio* – the ratio of the data packets successfully delivered to the destination over those generated at



(a) PDR vs. $\bar{V}$. (b) Normalized routing load vs. $\bar{V}$. (c) Average packet delay vs. $\bar{V}$.

Fig. 4. The comparison between MASK and AODV.

the sources; (2) *Average end-to-end delay of data packets* – this includes all possible delays caused by buffering during route discovery, queuing delay at the interface, retransmission delays at the MAC, and propagation delay; (3) *Normalized routing load* – the total number of routing control packets "transmitted" for each delivered data packet. Each hop-wise transmission of a routing packet is counted as one transmission.

For the implementation of MASK, we introduced a fixed delay of 150 μs into each node to mimic the encryption/decryption processing of routing replies and data packets with RC6 for simplicity. The purpose is to withstand the aforementioned *message coding attack*. In addition, random delay method for data packets to be forwarded was also adopted in each node to thwart the *timing analysis attack*, where the random delay is uniformly distributed between [0, 50] ms. Moreover, we implemented the anonymous neighborhood authentication described in Section III-B. Furthermore, we set the maximum number of next-hop link identifiers for one destination to be three.

2) Simulation results: Fig. 4(a) compares the packet delivery ratio of MASK and AODV under different traffic load. We can see that MASK has the similar PDR to AODV under normal traffic load (i.e., 20 sources). The slight difference partly comes from the fact that routing request packets in MASK have a higher probability of colliding with and causing the dropping of data packets than those in AODV due to the simple network-wide flooding of ARREQs in contrast to the expanding-ring-search method of AODV [6]. Another reason is that data packets in MASK are not always routed along the shortest paths due to the random selection of next-hops at intermediate nodes, which increases the dropping chances of data packets forwarded along longer paths. However, MASK outperforms AODV under heavy traffic load (i.e., 40 sources), where packets are more subject to collisions due to the high level of network congestion. The observed advantages mainly result from the aforementioned *virtual multipath* effect in MASK, that is, MASK may simultaneously maintain several pre-hops and next-hops for one given destination. If one of the next-hops becomes unreachable due to mobility or collision or other reasons, a packet could still be forwarded through another available next-hop rather than being dropped as AODV does. Moreover, the random selection of next-hops at intermediate nodes also acts as a load balancing method for evenly distributing the traffic in the network.

For the same reason, MASK demonstrates comparable or lower routing overhead than AODV (see Fig. 4(b)) because MASK conducts the costly route discovery less frequently than AODV.

In terms of the average packet delay (Fig. 4(c)), MASK behaves worse than AODV under normal traffic load as a result of the per-hop random delay, the fixed encryption/decryption delay, and the delay incurred by the Tate pairing operations. Therefore, there is a tradeoff between the desired packet delay and the level of anonymity. However, under heavy traffic load, both the *virtual multipath* effect and the processing delay (including the above three) introduced into MASK can help mitigate the possible MAC layer collisions, which contributes to the shown advantage of MASK over AODV in Fig. 4(c).

In summary, our anonymous routing protocol MASK not only achieves the desirable anonymity without the sacrifice of routing efficiency, but also helps improve it under heavy traffic load. Furthermore, the overall routing performance does not suffer from the random delay and per-hop encryptions/decryptions introduced into MASK used to combat a wide range of attacks.

V. Related work

Anonymous communication protocols have been studied intensively in the wired networks. Chaum [26] defined a layered object that routes data through a chain of pre-deployed intermediate nodes called *mixes*. Following their work, Reed *et al*. proposed an interesting Onion routing protocol [27], in which data is wrapped in a series of encrypted layers to form a onion by a series of proxies communicating over encrypted channels. For the lack of space, readers are referred to [28] for the state of art of wired networks anonymity. However, those proposals in the Internet realm cannot be directly applied to

MANETs mainly because of the lack of required pre-deployed infrastructures such as the well-known *mixes* and public-key infrastructure, and the scarcity of resources for computationally expensive certificate-based public-key operations and the related complicated certificate management.

Jiang *et al.* proposed to prevent traffic analysis in ad hoc networks by using traffic padding, i.e., generating dummy traffic into the network [29]. This approach did not aim to hide the identifiers of communicating nodes and so cannot completely prevent traffic analysis. They also explored the use of *mixes* in ad hoc networks [30] by designing a mix discovery protocol that allows communicating nodes to choose mix nodes at run time. The second approach is not an anonymous routing protocol and also vulnerable to the compromise of mix nodes.

Recently, Kong and Hong [31] 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 presented an anonymous on-demand routing protocol, called ANODR [23], to conceal the real identifiers of packet sources, destinations, and intermediate nodes. The design of ANODR relies on the aforementioned "trapdoor" in Section III-F, as a result of which ANODR suffers from the computationally intensive route discovery process. In addition, as the authors mentioned, ANODR is very sensitive to node mobility.

VI. Conclusion

In this paper, we proposed a novel pairing-based anonymous on-demand routing protocol, called MASK, to thwart malicious traffic analysis by passive adversaries. Based on the pairing technique, we proposed an anonymous neighborhood authentication protocol which enables neighboring nodes in the network to authenticate each other without revealing their real identities. Moreover, it allows the neighboring nodes to establish pairwise secret $<SKey, LinkID>$ pairs which are further used by MASK to identify and cryptographically protect the packets transmitted between them without revealing the identities of the local transmitter and recipient. MASK provides strong sender and receiver anonymity, the relationship anonymity between senders and receivers[11], the unlocatability of mobile nodes, and the untraceability of packet flows under a rather strong adversarial model. Based on the comprehensive anonymity analysis, MASK is shown to be immune to a wide range of attacks. In addition, MASK is proved to have comparable routing performance with the classic AODV routing protocol. Therefore, our MASK can serve as a lightweight underlying routing protocol for MANETs where anonymity is desired.

As the future research, we will first extend MASK to a hierarchical anonymous routing framework, which considers also the multiple parties scenario. Since the routing information is not authenticated in the current design of MASK, we will then plan to combine MASK with other secure routing schemes to provide an anonymous yet secure routing protocol. Finally, we will incorporate some intrusion detection capabilities into our framework to defend against active adversaries.

[11]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.

References

[1] W. Lou, Y. Fang. A Survey on Wireless Security in Mobile Ad Hoc Networks: challenges and available solutions. Book chapter in Ad Hoc Wireless Networking, Kluwer, May 2003.

[2] Y. Guan, X. Fu, D. Xuan, P. Shenoy, R. Bettati, and Wei Zhao. NetCamo: Camouflaging Network Traffic for QoS-Guaranteed Mission Critical Applications. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(4), July 2001.

[3] DARPA. Research Challenges in High Confidence Networking. July 1998.

[4] O. Berg, T. Berg, S. Haavik, J. Hjelmstad, and R. Skaug. Spread Spectrum in Mobile Communication. IEEE, 1998.

[5] S. Jiang, N. Vaidya, and Wei Zhao, Prevent Traffic Analysis in Packet Radio Networks. In *Proc. DISCEX II*, June 2001.

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

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

[8] D. Boneh and M. Franklin. Identify-based Encryption from The Weil Pairing. In *Proc. CRYPTO 01*, Springer-Verlag, 2001.

[9] P. S. L. M. Barreto, H. Y. Kim, B. Bynn, and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems. In *Proc. CRYPTO 02*, Springer Verlag, August 2002.

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

[11] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Royer. A Secure Routing Protocol for Ad Hoc Networks. *IEEE ICNP*, Paris, France, Nov. 2002.

[12] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure On-Demand Routing Protocol for Ad hoc Networks. *ACM MobiCom*, Atlanta, Georgia, Sep. 2002.

[13] M. Zapata. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. IETF Internet Draft, draft-guerrero-manet-saodv-00.txt, August 2001 (Work in Progress).

[14] Y. Zhang, W. Lou, and Y. Fang. SIP: A Secure Incentive Protocol against Selfishness in Mobile Ad Hoc Networks. *IEEE WCNC*, March 2004.

[15] NIST. Digital Hash Standard. Federal Information Processing Standards PUBlication 180-1, April 1995.

[16] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Veriable (SUCV) Identitiers and Adresses. In *Proc. NDSS*, Feb. 2002.

[17] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi. Secure Pebblenets. *ACM MobiHoc*, 2001.

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

[19] R. Rivest, M. Robshaw, R. Sidney, and L. Yin. The RC6 Block Cipher. v1.1, Aug. 1998. Available at `http://www.rsasecurity.com/rsalabs/rc6/`.

[20] W. Dai. Crypto++ Library 5.1 , Available at http://www.eskimo.com/ weidai/.

[21] Shamus Software Ltd. MIRACL library. Available at `http://indigo.ie/~mscott/`.

[22] P. Barreto, B. Lynn, M. Scott. On the Selection of Pairing-Friendly Groups. Selected Areas in Cryptography – SAC'2003, Lecture Notes on Computer Science 3006, Springer-Verlag (2004), pp. 17–25.

[23] J. Kong, X. Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. ACM MobiHoc, June 2003.

[24] J. Lopez and R. Dahab. An Overview of Elliptic Curve Cryptography. Technical report, Institute of Computing, State University of Campinas, Brazil, May 2000.

[25] J. Yoon, M. Liu, and B. Nobles. Sound mobility models. *ACM MobiCom*, San Diego, CA, Sep. 2003.

[26] David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM, 24(2), 1981.
[27] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications, May 1998.
[28] Anonymity bibliography. Available at `http://freehaven.net/anonbib/`.
[29] S. Jiang, N. Vaidya, and Wei Zhao. Dynamic Mix Method in Wireless Ad Hoc Networks. In *Proc. IEEE Milcom*, Oct 2001.
[30] S. Jiang, N. Vaidya, and Wei Zhao. Energy Consumption of Traffic Padding Schemes in Wireless AdHoc Networks. *Journal of Parallel and Distributed Computing Practices*, June 2001.
[31] J. Kong, X. Hong, and M. Gerla. A New Set of Passive Routing Attacks in Mobile Ad Hoc Networks. *IEEE Milcom*, Oct. 2003.