MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks

Yanchao Zhang, *Student Member, IEEE*, Wei Liu, Wenjing Lou, *Member, IEEE*, and Yuguang Fang, *Senior Member, IEEE*

Abstract— **The shared wireless medium of mobile ad hoc networks facilitates passive, adversarial eavesdropping on data communications whereby adversaries can launch various devastating attacks on the target network. To thwart passive eavesdropping and the resulting attacks, we propose a novel anonymous on-demand routing protocol, termed MASK, which can accomplish both MAC-layer and network-layer communications without disclosing real IDs of the participating nodes under a rather strong adversary model. MASK offers the anonymity of senders, receivers, and sender-receiver relationships in addition to node unlocatability and untrackability and end-to-end flow untraceability. It is also resistant to a wide range of attacks. Moreover, MASK preserves the high routing efficiency as compared to previous proposals. Detailed simulation studies have shown that MASK is highly effective and efficient.**

Index Terms— **Mobile ad hoc networks, security, eavesdropping, anonymity, routing.**

I. Introduction

MOBILE ad hoc networks (MANETs) are finding ever-increasing applications in both military and civilian operations. In this paper, we are concerned with MANETs deployed in hostile environments, such as those facilitating large-scale theater-wide communications or relatively small-scale communications in MOUT (Military Operations on Urban Terrain).

The shared wireless medium of MANETs introduces abundant opportunities for passive eavesdropping on data communications. This means that, without physically compromising a node, adversaries can easily overhear all the MAC frames "flying in the air," each typically including <MAC addresses, network addresses, data>[1]. Although end-to-end and/or link encryption can be enforced to prevent adversarial access to data contents, for any observed frame, adversaries can still learn not only the network and MAC addresses of its local transmitter and receiver, but also the network addresses of its end-to-end source and destination. Such MAC and network address information is currently left bare without protection in the de facto MAC protocol IEEE 802.11 and existing MANET routing protocols such as AODV [1] and DSR [2].

The leakage of MAC and network addresses may result in a number of severe consequences. First of all, it would facilitate adversarial traffic analysis run to infer network traffic patterns and/or traffic pattern changes[2]. In a tactical military MANET, an abnormal change of the network traffic pattern may indicate a forthcoming action, a chain of commands, or a state change of network alertness [3]. Its disclosure to adversaries would thus lead to the failure of urgent military actions. In addition, adversaries are able to trace any packet backward to its original source or forward to its final destination. This is also undesirable because in many cases packet sources are critical nodes such as captains or majors, while packet destinations are nodes commanded to carry out certain military operations. Moreover, adversaries can locate individual nodes and track their movements. This is extremely dangerous in that adversaries can easily identify critical network nodes and then launch directed attacks on them. Most previous proposals such as [4], [5] aim to deal with *active attacks*, which usually involve the launch of denial-of-service (DoS) or other more "visible," aggressive attacks on the target network. By contrast, the aforementioned attacks belong to the category of *once-passive-then-active* attacks, or passive attacks for short, which are more subtle, "invisible," and difficult to detect before severe damage actually occurs. In this paper, we seek efficient solutions to such more dangerous passive attacks.

For ease of presentation, we use the notion "network ID" (or simply "ID") to indicate both the MAC and network addresses of a mobile node, which should be understandable from the context. We also define "anonymity" as the privacy preservation of network IDs of mobile nodes and their group membership information, e.g., belonging to nation *A* or *B*, or affiliated with battalion 1 or 2. Although less intuitive, the privacy of node affiliations is as important as that of node IDs in many security-sensitive environments. For example, suppose a coalition force of multiple nations is dispatched to carry out a common military mission. Soldiers of the same nation can form an exclusive MANET among themselves, and

Manuscript received December 24, 2003; revised June 2, 2004 and October 18, 2005; accepted October 27, 2005. The associate editor coordinating the review of this paper and approving it for publication was S. Sarkar. This work was supported in part by the U.S. Office of Naval Research under Young Investigator Award N000140210464 and the US National Science Foundation under grant ANI-0093241 (CAREER Award).

Y. Zhang and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (email: yczhang@ufl.edu, fang@ece.ufl.edu).

W. Liu is with Scalable Network Technologies, Los Angeles, CA 90045 USA (email:liuw@ufl.edu).

W. Lou is with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA (email: wjlou@ece.wpi.edu).

Digital Object Identifier 10.1109/TWC.2006.03705.

[1] We use the terms "packets" and "frames" interchangeably in this paper.

[2] A network traffic pattern consists of triplets <*sender_addr, receiver_addr, average rate*>, each describing one flow. A flow can be an end-to-end network flow; then the address fields are the network addresses of an end-to-end source and destination pair. It can also be a local link flow; then the address fields are the MAC addresses of a local transmitter and a receiver.

thus there would co-exist multiple MANETs in the battlefield. In this case, each node may want to avoid unnecessary exposure of both its ID and nationality because adversaries or terrorists may perform selective directed attacks according to not only IDs but also nationalities. As demonstrated in Section III-B.1, conventional cryptographic techniques such as Diffe-Hellman key exchange [6] cannot satisfy this anonymity requirement and thus fail to withstand passive attacks.

We observe that passive attacks are feasible for two reasons: (1) each node can be uniquely identified by its network ID, and (2) each node uses the invariant network ID in both MAC-layer and network-layer communications. Motivated by this observation, we propose to thwart passive attacks by designing anonymous communication protocols. The fundamental purpose is to realize both efficient MAC-layer and network-layer communications, while anonymizing all the involved nodes, therefore effectively defeating passive attacks.

The contribution of this paper is the design of a novel anonymous on-demand routing protocol, called MASK, which can simultaneously achieve anonymous MAC-layer and network-layer communications. The novelty of MASK lies in the use of dynamic pseudonyms rather than static MAC and network addresses. MASK offers both *sender* and *receiver anonymity* as well as *sender-receiver relationship anonymity*[3]. Specifically, although adversaries might observe a packet transmission, they cannot determine real network IDs of its sender and receiver, nor can they decide if (or when) any two nodes in the network are communicating. In addition, MASK ensures *node unlocatability* and *untrackability*, meaning that, although adversaries might know some real network IDs and/or group memberships, they are unable to decide whom and where the corresponding nodes are in the network. Moreover, MASK guarantees *end-to-end flow untraceability*, which means that adversaries cannot trace a packet forward to its final destination or backward to its original source, nor can they recognize packets belonging to a same ongoing communication flow. Furthermore, MASK is as efficient as classical routing protocols such as AODV [1], which is confirmed by detailed simulation results. It can also withstand a variety of attacks, e.g., message coding, flow recognition, and timing analysis.

II. Preliminaries

A. Pairing Concept

Pairing-based cryptography [7]–[9] is the cryptographic foundation of our work. Let $\mathbb{G}_1$ be an additive cyclic group of prime order q and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order. Assume that the discrete logarithm problem (DLP) is hard[4] in both $\mathbb{G}_1$ and $\mathbb{G}_2$. A pairing is a *bilinear map* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ if, for all $P, Q, R, S \in \mathbb{G}_1$, we have[5]

$$\hat{e}(P + Q, R + S) = \hat{e}(P, R)\hat{e}(P, S)\hat{e}(Q, R)\hat{e}(Q, S). \quad (1)$$

Modified Weil [7] and Tate [8] pairings are examples of such bilinear maps, for which the *Bilinear Diffie-Hellman Problem* (BDHP) is believed to be hard[6]. It is also worth mentioning that $\hat{e}$ is *symmetric*, i.e., $\hat{e}(P, Q) = \hat{e}(Q, P)$ for $\forall\ P, Q \in \mathbb{G}_1$, which follows immediately from the bilinearity and the fact that $\mathbb{G}_1$ is a cyclic group. We refer to [7], [8] for a more comprehensive description of how the pairing parameters should be chosen in practice for both efficiency and security.

B. Adversary Model

We assume that adversaries can collaborate to passively 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 postulate that passive adversaries cannot compromise an unlimited number of nodes. Nor can they have unbounded computational capabilities to easily invert and read encrypted messages and break the BDHP assumption. Otherwise, it is believed that there is no workable cryptographic solution.

III. MASK Design

In this section, we elaborate the design of MASK. We start with describing the network model and then discuss how to achieve single-hop MAC-layer communications. Subsequently, we present an on-demand routing protocol to realize anonymous network-layer communications. After that, some countermeasures against attacks and a security enhancement based on the secret-sharing technique [10] are introduced.

A. Network Model

We consider a general case that there co-exist multiple MANETs, each comprising nodes of the same group. For simplicity, we use a capital letter, such as A, B, or C, to indicate each MANET and the group it corresponds to. The concrete meanings of groups may vary across different application contexts. For example, each group or the related MANET may be related to a troop of a different nation, or a different company or battalion in the same brigade. Hereafter, we will utilize network A as an example to illustrate our MASK design. We denote by $A.i$ the ith node of A for $1 \leq i \leq N_A$, where N_A is the number of nodes in A. We assume that each $A.i$ has a unique non-zero network ID $ID_{A.i}$. As discussed before, both $ID_{A.i}$ and node $A.i$'s membership in A should be well protected from adversaries.

Prior to network deployment, a trusted authority (TA) who himself/herself does not enter the network first determines the pairing parameters $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e})$ along with a group-wise master key $g_A \in \mathbb{Z}_q^*$. The TA then chooses two collision-resistant cryptographic hash functions: H_1, mapping strings to non-zero elements in $\mathbb{G}_1$, and H_2, mapping arbitrary inputs to fixed-length outputs, e.g., SHA-1 [11]. Public system parameters $< q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1, H_2 >$ are preloaded to each $A.i$. By contrast, g_A should be well safeguarded from unauthorized access and never be disclosed to ordinary group members dispatched to execute dangerous military actions.

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

[4] It is computationally infeasible to extract the integer $x \in \mathbb{Z}_q^* = \{i | 1 \leq i \leq q-1\}$, given $P, Q \in \mathbb{G}_1$ (respectively, $P, Q \in \mathbb{G}_2$) such that $Q = xP$ (respectively, $Q = P^x$).

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

[6] It is believed that, given $< P, xP, yP, zP >$ for random $x, y, z \in \mathbb{Z}_q^*$ and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $\hat{e}(P, P)^{xyz} \in \mathbb{G}_2$ with non-negligible probability

In MASK, nodes substitute pseudonyms for real IDs in communications. If a node uses one pseudonym all the time, it will not help to defend against passive attacks we have in mind, because the pseudonym will be analyzed the same way as its real ID. Therefore, each node should use dynamic pseudonyms instead. For this purpose, the TA furnishes each $A.i$ with a sufficiently large set $\mathcal{PS}_{A.i} = \{PS^k_{A.i} | 1 \leq k \leq |\mathcal{PS}_{A.i}|\}$ of collision-resistant pseudonyms[7]. A pseudonym can be any type of string and collision-resistance means that all the pseudonyms are different from each other. In addition, each $A.i$ is armed with a corresponding *secret point set* as $\mathcal{SP}_{A.i} = \{SP^k_{A.i}\} = \{g_A H_1(PS^k_{A.i}) \in \mathbb{G}_1\}$ $(1 \leq k \leq |\mathcal{PS}_{A.i}|)$. Due to the difficulty of solving the DLP in $\mathbb{G}_1$ (cf. Section II-A), given any $< PS^k_{A.i}, SP^k_{A.i} >$ pair, it is impossible to deduce g_A with non-negligible probability.

B. Anonymous MAC-Layer Communications

In this subsection, we discuss how to achieve anonymous single-hop MAC-layer communications through an anonymous neighborhood authentication protocol.

1) Anonymous neighborhood authentication: As the name suggests, anonymous authentication allows two neighboring nodes of the same group to identify each other *secretly*, in the sense that each party reveals its group membership to the other only if the other party is also a group member. This notion bears similarity to the concept of *secret handshakes* introduced by Balfanz *et al.* [9]. As an example, node $A.i$ might want to authenticate itself to a neighboring node x, but only if x is also a member of group A. In addition, if x does not belong to A, the authentication protocol should not help x in determining either the real ID ($ID_{A.i}$) of $A.i$ or whether $A.i$ is a member of A or not. As mentioned in [9], realizing anonymous authentication (or secret handshakes) requires new cryptographic protocols since it cannot be easily accomplished through existing cryptographic tools. For example, authentication techniques based on public-key certificates, such as authenticated two-party Diffe-Hellman key exchange [6], may inevitably disclose either real IDs of mobile nodes or their group memberships or both, which are either implied or explicitly embedded in public-key certificates. For instance, for its certificate to be verified, a node has to tell the other party the authentic public key of the CA (Certificate Authority) that generates its certificate. Obviously, this would cause the exposure of that node's group membership, i.e., from which CA it obtains the certificate, no matter whether the other party belongs to the same group or not. In the following, we illustrate a pairing-based anonymous neighborhood authentication protocol, which is an extension of the secret handshake scheme introduced in [9] to MANETs.

Without loss of generality, below is shown the authentication process between nodes $A.1$ and $A.2$, where $\|$ denotes message concatenation.

$$\begin{array}{llll} A.1 & \rightarrow & A.2 & : PS^i_{A.1}, n_1 \\ A.2 & \rightarrow & A.1 & : PS^j_{A.2}, n_2, V_{2,1} = H_2(n_1 \parallel n_2 \parallel 0 \parallel K_{2,1}) \\ A.1 & \rightarrow & A.2 & : V_{1,2} = H_2(n_1 \parallel n_2 \parallel 1 \parallel K_{1,2}) \end{array}$$

$A.1$ starts the protocol by pulling out from $\mathcal{PS}_{A.1}$ an unused pseudonym $PS^i_{A.1}$ and locally broadcasts a MAC frame including $PS^i_{A.1}$ and a random nonce n_1. Upon seeing the request, $A.2$ also draws an unused pseudonym $PS^j_{A.2}$ from $\mathcal{PS}_{A.2}$ and then generates a master key as $K_{2,1} = \hat{e}(H_1(PS^i_{A.1}), SP^j_{A.2})$. After that, $A.2$ locally broadcasts a reply frame consisting of $PS^j_{A.2}$, a random nonce n_2, and a value $V_{2,1}$ shown above. Upon reception of the reply from $A.2$, node $A.1$ calculates a master key as $K_{1,2} = \hat{e}(H_1(PS^j_{A.2}), SP^i_{A.1})$ as well and checks $V_{2,1} \overset{?}{=} H_2(n_1 \parallel n_2 \parallel 0 \parallel K_{1,2})$. According to Eq. (1) and the symmetric property of $\hat{e}$, if and only if both nodes are affiliated with the same group A, could they have

$$\begin{aligned} K_{2,1} &= \hat{e}(H_1(PS^i_{A.1}), H_1(PS^j_{A.2}))^{g_A} \\ &= \hat{e}(H_1(PS^j_{A.2}), H_1(PS^i_{A.1}))^{g_A} \quad = \quad K_{1,2} \ . \end{aligned}$$

As a result, if the verification succeeds, $A.1$ knows that $A.2$ must be an authentic group peer. To authenticate itself to $A.2$, $A.1$ returns a value $V_{1,2}$ shown above. If $V_{1,2} = H_2(n_1 \parallel n_2 \parallel 1 \parallel K_{2,1})$, node $A.2$ can rest assured that $A.1$ belongs to the same group A as itself. Notice that the source and destination addresses of the three involved MAC frames should both be set to be a pre-defined universal address such as all 1's instead of their real network IDs (MAC addresses in this case).

After a successful three-way handshake, $A.1$ learns that there is a trustable group peer in its neighborhood, but has no knowledge of the real ID except one of the public pseudonyms of $A.2$. So does $A.2$. If the authentication fails, which may occur for instance when one of them is an adversarial impersonator, the legitimate one reveals nothing but a pseudonym to the impersonator. In addition, an adversarial eavesdropper learns nothing more than some seemingly random numbers from the protocol execution.

Since $A.1$ and $A.2$ have established a shared master key $K_{1,2} = K_{2,1}$, they can proceed to calculate Γ pairs of shared session key (*Skey*) and link identifier (*LinkID*) as

$$\left\{ \begin{array}{lll} k^{\gamma}_{1,2} & = & H_2(n_1 \parallel n_2 \parallel 2 * \gamma \parallel K_{1,2}) \\ L^{\gamma}_{1,2} & = & H_2(n_1 \parallel n_2 \parallel 2 * \gamma + 1 \parallel K_{1,2}) \ , \end{array} \right. \tag{2}$$

where Γ is a design parameter, and $k^{\gamma}_{1,2}$ and $L^{\gamma}_{1,2}$ $(1 \leq \gamma \leq \Gamma)$ indicate the γ^{th} Skey and LinkID, respectively. The collision-resistance of node pseudonyms, H_1 and H_2 ensures that such <Skey, LinkID> pairs are also collision-resistant, meaning that no identical pairs would be generated by different pairs of nodes or two same nodes with different pairs of nonces. In addition, each <Skey, LinkID> pair is only known to the two nodes which established it and there is even no apparent relationship among the <Skey, LinkID> pairs generated by two same nodes under the same pair of nonces. Such $< k^{\gamma}_{1,2}, L^{\gamma}_{1,2} >$ pairs are to be used in an increasing sequence for subsequent data communications between $A.1$ and $A.2$, as will be explained shortly. Whenever established Γ pairs are used up, $A.1$ and $A.2$ are required to automatically increase both n_1 and n_2 by one and generate new Γ pairs using the computationally efficient hash function H_2. Of course, $A.1$ and $A.2$ should have a simple agreement so as to synchronize the use of such pairs.

Similarly, each node can achieve anonymous mutual authentication and establish pairwise shared <Skey, LinkID> pairs

[7] If X is a set, $|X|$ means its cardinality.

with all its neighboring nodes. Notice that if multiple nodes simultaneously answer the same request, possible MAC-layer collisions may occur. In this paper, we assume the reliable transmissions of authentication requests/replies, which can be achieved for instance by using a random delay for which each node has to wait before answering an authentication request.

In our design, we leave the decision when and whether a node wants to initiate the anonymous neighborhood authentication to the node itself. Ideally, a node should keep track of its neighbors at all time and should perform the authentication whenever it moves to a new place or finds new neighbors. In this case, a neighbor discovery/maintanence mechanism such as the "Hello" messages used in AODV [1] will be necessary. Notice here that although the "Hello" messages are transmitted periodically, the authentication is done only once for each neighbor. A node may also choose not to do the authentication while it is on the constant and fast movement. Another option is that a node only initiates the authentication on-demand, e.g., when it receives a route discovery message from an unauthenticated neighbor. Authentication purely on-demand could reduce the overhead caused by running the neighborhood authentication protocol, while at the same time it would introduce extra delay on the route discovery process.

We would like to point out that anonymous neighborhood authentication would incur additional computational overhead in contrast to other on-demand routing protocols such as AODV and DSR, which do not provide either security or anonymity guarantees. However, mutual authentication between neighboring nodes is indispensable in MANETs, only by which one node can reject accepting messages from or forwarding messages for unauthenticated neighbors. Otherwise, adversaries can easily inject bogus messages into the network to deplete scarce network resources as well as interrupting proper network functionalities. In addition, any two neighboring nodes only need to perform authentication once and subsequent communications can be encrypted and authenticated using efficient symmetric-key algorithms based on established shared Skeys. It will be shown in Section IV that anonymous neighborhood authentication can be implemented efficiently without much degrading the routing efficiency.

2) Anonymous MAC frame exchange: Based on established shared <Skey, LinkID> pairs, two neighboring nodes can easily realize anonymous single-hop MAC-layer communications. In our design, we replace the transmitter and receiver MAC addresses in a conventional MAC frame with a single LinkID. In fact, we will see later that the same LinkID also eliminates the necessity of network addresses. In other words, a conventional MAC frame <MAC addresses, network addresses, data> changes to <LinkID, data> in our scheme.

For example, $A.1$ sends a MAC frame of format $< L^1_{1,2}, \{data\}_{k^1_{1,2}} >$, where $\{msg\}_K$ stands for a message msg encrypted under key K using any symmetric-key encryption algorithm such as RC6 [12]. That frame can be heard by all its neighboring nodes, among which only $A.2$ will accept the frame because of its unique sharing of $L^1_{1,2}$ with $A.1$. $A.2$ can decrypt the data with the corresponding Skey $k^1_{1,2}$. Similarly, $A.2$ can reply with a MAC frame $< L^2_{1,2}, \{data\}_{k^2_{1,2}} >$. If the MAC protocol in use is contention-based, such as the Distributed Coordination Function (DCF) of the IEEE 802.11, conventional RTS-CTS-DATA-ACK frame exchange is also easy to implement based on pairwise shared LinkIDs to alleviate notorious hidden and exposed terminal problems.

Since real IDs of mobile nodes are kept confidential in anonymous neighborhood authentication and subsequent local MAC frame exchange, we have successfully realized anonymous single-hop MAC-layer communications. In other words, local transmitter and receiver anonymity and their relationship anonymity have been achieved. Also notice that our anonymous neighborhood authentication protocol ensures both node unlocatability and untrackability at the same time.

C. Anonymous Network-Layer Communications

Network-layer communications, most likely multi-hop, rely on routing protocols to find end-to-end routing paths between any source-destination pair and relay packets in a hop-by-hop manner enroute from the source to the destination. To realize anonymous network-layer communications, we present here an anonymous on-demand routing protocol, called MASK, to establish a sequence of <Skey, LinkID> pairs between any source and destination pair. In our MASK, each node maintains the following data structures:

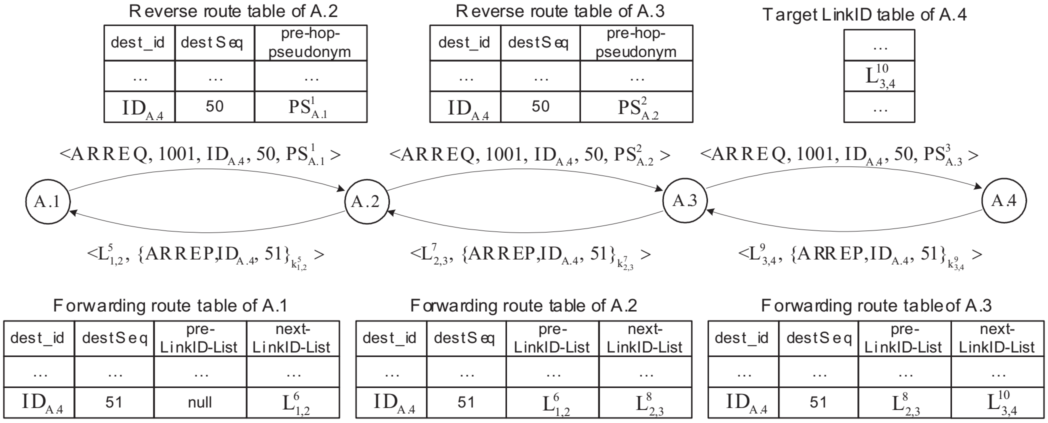
- *Forwarding route table*: A table consisting of entries of format *<dest_id*, *destSeq*, *pre-LinkID-list*, *next-LinkID-list>*, where *dest_id* is the real ID of the destination and *destSeq*[8] is the corresponding node sequence number. The *pre-LinkID-list* is the set of pre-hop LinkIDs from which packets destined for *dest_id* may come, and *next-LinkID-list* is the set of next-hop LinkIDs 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 LinkID table*: A table consisting of selected LinkIDs shared with neighbors. The current node is the final destination (end-to-end) for the packets bearing the LinkIDs in its target LinkID table.

An appropriate timer is associated with each entry of the above tables and an entry should be recycled when its timer expires.

1) Anonymous route discovery: Without loss of generality, we illustrate the anonymous route discovery process in MASK using the simple chain topology shown in Fig. 1, where nodes $A.1, A.2, A.3$, and $A.4$ are assumed to be using pseudonyms $PS^1_{A.1}, PS^2_{A.2}, PS^3_{A.3}$, and $PS^4_{A.4}$, respectively, in their current places. To ease the presentation, we further assume that each node has finished anonymous mutual authentication using the same pseudonym with all its neighboring nodes and has established shared <Skey, LinkID> pairs with them.

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. An anonymous route request (ARREQ) has the format *<ARREQ*, *ARREQ_id*, *dest_id*, *destSeq*, PS_{src}*>*, where *dest_id* is the real ID of

[8] The maintenance of node sequence numbers strictly follows the steps defined in AODV [1].



Reverse route table of A.2

dest_id	destSeq	pre-hop-pseudonym
…	…	…
$ID_{A.4}$	50	$PS^1_{A.1}$

Reverse route table of A.3

dest_id	destSeq	pre-hop-pseudonym
…	…	…
$ID_{A.4}$	50	$PS^2_{A.2}$

Target LinkID table of A.4

…
$L^{10}_{3,4}$
…

Forwarding route table of A.1

dest_id	destSeq	pre-LinkID-List	next-LinkID-List
…	…	…	…
$ID_{A.4}$	51	null	$L^6_{1,2}$

Forwarding route table of A.2

dest_id	destSeq	pre-LinkID-List	next-LinkID-List
…	…	…	…
$ID_{A.4}$	51	$L^6_{1,2}$	$L^8_{2,3}$

Forwarding route tableof A.3

dest_id	destSeq	pre-LinkID-List	next-LinkID-List
…	…	…	…
$ID_{A.4}$	51	$L^8_{2,3}$	$L^{10}_{3,4}$

Fig. 1. Anonymous route discovery with a route reply generated by the destination $A.4$.

the destination, [9]*ARREQ_id* 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_{src} is the active pseudonym of the source. To be consistent with the aforementioned MASK packet format, a predefined LinkID such as all 1's should be used to identify the ARREQ, which is not shown for brevity. In the shown example, the ARREQ takes the form of <*ARREQ*, 1001, $ID_{A.4}$, 50, $PS^1_{A.1}$ >. When an intermediate node, say node $A.2$, 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 replacing the embedded pseudonym $PS^1_{A.1}$ with its currently-used one, i.e., $PS^2_{A.2}$. ARREQs with previously seen *ARREQ_id*s are simply discarded[10]. This process continues until all the nodes in the network have rebroadcasted the ARREQ once.

It is worth noting that in the propagation of ARREQs, the real IDs of the source and all the intermediate nodes are concealed, while the real ID of the destination has to be exposed. In traditional on-demand routing protocols such as AODV [1], the destination itself and any intermediate node which has a valid routing entry to the destination do not need to rebroadcast the route request message. However, that design allows adversaries to identify the destination node easily by monitoring the activities at each node - every node broadcasts the routing request once except the destination and/or some nodes having the routes to the destination. Therefore, in our design, every node, including the destination and qualified intermediate nodes, needs to rebroadcast the ARREQ message once. This will effectively hide the whereabout of the destination - even though adversaries know that there is such a node, they will have difficulty to match the *dest_id* ($ID_{A.4}$ in this case) 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 needs to broadcast once anyway except the destination and qualified intermediate nodes. So the extra overheard introduced is only one or a few more transmissions by the destination and the intermediate nodes which can reply.

An anonymous route reply (ARREP) can be generated and sent back to the source at the destination or at any intermediate node which has a valid route to the destination. Fig. 1 demonstrates the case that a route reply is generated by the destination $A.4$ itself. Once receiving an ARREQ toward itself, $A.4$ can generate an ARREP to be unicasted back to the source following the reverse route established before. In our design, an ARREP packet is of format <LinkID, {*ARREP*, *dest_id*, *destSeq*}$_{Skey}$>, where LinkID is the next to be used shared between the destination and the pre-hop node from which the ARREQ comes, and the corresponding Skey is used to encrypt the packet content so that adversaries cannot recognize that this is an ARREP corresponding to the previously-observed ARREQ. In the shown example, an ARREP is in the form of $<$ $L^9_{3,4}$, {*ARREP*, $ID_{A.4}$, 51}$_{k^9_{3,4}}$ $>$. As noted before, only the intended receiver $A.3$ will be able to interpret $L^9_{3,4}$ and decrypt the packet content accordingly. While for a passive eavesdropper, $L^9_{3,4}$ only appears to be some meaningless random number, and it has no idea of what the packet is about and to whom the packet is sent. Moreover, $A.4$ adds $L^{10}_{3,4}$ to its target LinkID table. The reason of inserting $L^{10}_{3,4}$ instead of $L^9_{3,4}$ is to prevent adversaries from identifying the relationship between this ARREP packet and subsequent data packets. Later on, when seeing a packet identified by $L^{10}_{3,4}$, $A.4$ knows that it is the end-to-end destination of that packet. An intermediate node can also generate an ARREP 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 node as well. Different from the destination, the intermediate node need not modify its target LinkID table. This case is straightforward and not shown for lack of space.

For a node on the reverse path, say $A.3$, when receiving an ARREP $<$ $L^9_{3,4}$, {*ARREP*, $ID_{A.4}$, 51}$_{k^9_{3,4}}$ $>$ from its next-hop, $A.3$ will discard it if the embedded *destSeq*, 51 in this case, is smaller than that in its reverse route table. Otherwise, $A.3$ will decrypt the ARREP, form and transmit a new ARREP $<$ $L^7_{2,3}$, {*ARREP*, $ID_{A.4}$, 51}$_{k^7_{2,3}}$ $>$. Here $<k^7_{2,3}, L^7_{2,3}>$ is the next to be used <Skey, LinkID> pair shared between $A.3$

[9] *ARREQ_id* could be generated by applying a collision-resistant hash function like SHA-1 [11] on the concatenation of a node's pseudonym, sequence number, and a timestamp.

[10] 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.

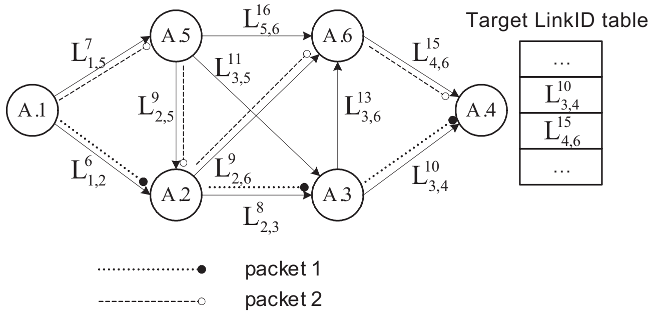


Fig. 2. Anonymous hop-by-hop packet forwarding from $A.1$ to $A.4$.

and the pre-hop node "$PS^2_{A.2}$" (in fact, node $A.2$) stored in its reverse route table. $A.3$ also needs to update its forwarding route table as follows. If it does not have an entry for $ID_{A.4}$, a new entry will be created. Or if the entry for $ID_{A.4}$ 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-LinkID-list*, and *next-LinkID-list* will be set to $ID_{A.4}$, *destSeq* in the ARREP, $L^8_{2,3}$, and $L^{10}_{3,4}$, respectively. If $A.3$ already has an entry for $ID_{A.4}$, and the new *destSeq* in the ARREP is equal to the old one, it updates the route entry by appending $L^{10}_{3,4}$ and $L^8_{2,3}$ to the *next-LinkID-list* and *pre-LinkID-list* fields of its forwarding route entry, respectively. Therefore, MASK may simultaneously maintain several next-hop and pre-hop LinkIDs for one *dest_id* (called *virtual multipath functionality* in this paper) in the forwarding route table. This operation is different from that of AODV [1] in which a node suppresses routing replies with the same destination sequence number. The reason for adopting this design will be stated in the subsequent subsection. Also notice that LinkIDs inserted into forwarding route tables are always next to the ones used to identify the ARREPs so that adversaries cannot correlate the ARREPs with subsequent data packets. The above process continues until the ARREP reaches the source $A.1$. An exemption in the route reply process is that, in MASK, since each node is required to rebroadcast the ARREQ message no matter whether it replies or not, the ARREPs coming back to an intermediate node which replied before may present inconsistent state information that may cause routing loops. Therefore, we require that the intermediate nodes which have 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 LinkIDs which are only interpretable by the intended local receivers. A passive eavesdropper might see discrete transmissions everywhere but it will not be able to tell the content of a particular transmission, nor can it tell who is transmitting and who is receiving. For an internal adversary who happens to reside in the reverse route to the source, due to the anonymous neighborhood authentication, what it can learn is the ID of the destination, but not which and where that destination is even when the destination is its neighbor.

2) Anonymous packet forwarding: The packet forwarding in MASK is more like a virtual circuit switching process. By looking up in the forwarding route table, the source picks a random LinkID from the *next-LinkID-list* field in the entry for the destination. A packet is then formed and sent to the next-hop neighbor that shares the chosen LinkID. As noted before, a packet is of format <LinkID, data>, where the data part carries other protocol and application data. Depending on different applications, the data part can be end-to-end encrypted and/or authenticated using cryptographic methods. Or it can be encrypted and authenticated by the Skey corresponding to the LinkID. When seeing such a packet, the first intermediate node sharing the embedded LinkID needs to change it to one randomly selected from its *next-LinkID-list* field of the forwarding route entry in which the embedded LinkID matches one of the values in the *pre-LinkID-list*. It then re-unicasts the packet to the chosen next hop. Following this process, a packet can finally reach the destination which will terminate the forwarding when finding the LinkID in its target LinkID table.

An example of anonymous packet forwarding is depicted in Fig. 2, in which a set of forwarding links (denoted by directional solid lines) have been established, each labelled by its respective LinkID. The incoming and outgoing links of a node constitute the *pre-LinkID-List* and *next-LinkID-List* fields of its forwarding route entry for the destination $A.4$, respectively. As we can see, due to the random selection of next-hop LinkIDs at each intermediate node, MASK has the nice *traffic mixing* property that packets of the same flow may travel through different paths to the destination. This makes it more difficult for adversaries to correlate observed radio transmissions to acquire actual network traffic patterns. It also increases the difficulty of adversaries in tracing a packet enroute from its original source to the final destination. The shortcoming is that, MASK does not always use the best path, e.g., the shortest-hop path, for packet forwarding, so it may introduce extra delay and/or delay jitter. However, for security-sensitive MANETs demanding anonymity protection, we argue that this tradeoff of routing efficiency for anonymity is acceptable. In addition, we will see in Section IV-B that such random packet forwarding can help improve the routing performance under heavy traffic load.

When all the next-hop nodes for one destination become unavailable due to mobility or other reasons, a node needs to locally broadcast an anonymous route error (ARRER) packet of format <*ARRER*, *pre-LinkID-list*> to inform its up-stream nodes, which is again identified by a predefined universal LinkID including all 1's. Any neighboring node which has one of the LinkIDs in the received *pre-LinkID-list* should remove it from the *next-LinkID-list* field of its corresponding forwarding route entry. If its own *next-LinkID-list* becomes empty as well, it should also broadcast a similar ARRER packet. When the source has no available next-hop LinkIDs for the destination, it should restart the anonymous routing discovery process.

D. Countermeasures against Attacks

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 security enhancements 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 Skeys 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 related to a same communication flow. Notice that, in MASK, a same packet bears completely different and uncorrelated LinkIDs 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, adversaries may obtain some useful information. Fortunately, the aforementioned random packet forwarding can partially mitigate this attack. In fact, an intermediate node works as a multiplexer which takes inputs from multiple pre-links, 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. In doing so, MASK leaves adversaries a dynamic set of LinkIDs for the same flow and at each hop. Moreover, dynamic LinkIDs at each hop effectively thwart the *message replay attack* in which adversaries replay an old packet repeatedly to recognize the packet forwarding pattern.

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 go into or come out of that cell while some packets come out of or go into that cell during a certain time interval. In addition, adversaries might guess that two consecutive radio transmissions belong to the same communication flow. These attacks belong to the category of the *timing analysis attack*.

In MASK, packets transmitted in the air are only identified by seemingly random LinkIDs. 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 adversaries into believing that one observed radio transmission does not end at the destination. The destination can also use genuine LinkIDs to ask its trustful neighbors to help further enlarge the suspicious area viewed by 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 arrival. Last, even without being involved in any communications, nodes can send dummy packets [13] with fake LinkIDs 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 as to conceal the real network traffic patterns, at the cost of increasing communication overhead.

E. Replenishing Pseudonym/Secret Point Pairs

In our MASK, each node is required to use dynamic pseudonym/secret point pairs. If the network has a rather long lifetime, however, a node may use up the preloaded pseudonym/secret point pairs sooner or later. If this occurs, a node can reuse old pairs, staring from the first one. This measure can prevent adversaries from continuously tracking the movement of individual nodes if there are sufficiently many preloaded pairs. Nevertheless, it may still offer useful attack clues to powerful adversaries - adversaries may roughly ascertain the movement of certain nodes by observing that a pre-recorded pseudonym reappears in certain network location.

To avoid the above situation and ensure strong anonymity protection, it is necessary to introduce the TA functionality into the network whereby mobile nodes can get replenishment of pseudonym/secret point pairs. Since using a single TA is vulnerable to single point of failure, we propose to employ Shamir' secret-sharing technique [10] to enable a more scalable, secure solution. To do this, the TA executes the following additional operations when bootstrapping network A:

1. Determine a (t-1)-degree ($1 \leq t \leq N_A$) polynomial, $h(x) = g_A + \sum_{i=1}^{t-1} a_i x^i$, with random coefficients a_i in $\mathbb{Z}_q^*$ and g_A being the group master key.
2. Select n ($t \leq n \leq N_A$) nodes from A, either without distinction or by considering node heterogeneity and choosing physically more secure or computationally more powerful ones. We call these nodes *shareholders*, denoted by $\mathcal{SH} = \{SH.k | 1 \leq k \leq n\}$.
3. Calculate n shares of g_A as $g_k = h(ID_{SH.k})$ and assign it to $SH.k$.
4. Choose an arbitrary generator $W \in \mathbb{G}_1$ and compute a set of share commitments as $\mathcal{SC} = \{W_k^{pub} = g_k W \in \mathbb{G}_1 | 1 \leq k \leq n\}$.

$\mathcal{SH}$, $\mathcal{SC}$ and W are appended to the public system parameters known to every node. An interesting fact is that, although each $SH.k$ does not have the full knowledge of g_A, any t of them can collectively construct g_A, while any less than t cannot. For example, based on the Lagrange interpolation, shareholders $SH.1, SH.2, ..., SH.t$ can determine g_A:

$$g_A = \sum\nolimits_{i=1}^{t} \lambda_i g_i, \text{ where } \lambda_i = \prod\nolimits_{j=1, j \neq i}^{t} \frac{ID_{SH.j}}{ID_{SH.j} - ID_{SH.i}} . \tag{3}$$

During network operation, when a node, say $A.1$, almost runs out of preloaded pseudonym/secret point pairs, it can get replenishment by sending a request including the list of desired new pseudonyms to each of t randomly-picked shareholders. Without loss of generality, assume that shareholders $SH.1, SH.2, ..., SH.t$ are selected by $A.1$. For each pseudonym $PS_{A.1}^x$ in the request, each chosen $SH.i$ generates a partial secret point $SP_{A.1}^{x,i} = g_i H_1(PS_{A.1}^x)$ sent back to $A.1$. To verify the authenticity of each $SP_{A.1}^{x,i}$, $A.1$ needs to check if $\hat{e}(SP_{A.1}^{x,i}, W) = \hat{e}(H_1(PS_{A.1}^x), W_i^{pub})$. Notice that,

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

due to Eq. (1), the two sides of the equation are equal to the same value $\hat{e}(H_1(PS^x_{A.1}), W)^{g_i}$ if $SP^{x,i}_{A.1}$ is authentic. As a result, if the verification fails, $A.1$ knows that there must be something wrong with $SH.i$. For example, the reply from $SH.i$ might have undergone transmission errors, or even $SH.i$ itself might have been physically or logically controlled by adversaries. $A.1$ can then request a new partial secret point from another unselected shareholder. Once obtaining t authentic partial secret points, $A.1$ utilizes Eq. (3) to calculate the complete secret point:

$$SP^x_{A.1} = \sum\nolimits_{i=1}^{t} \lambda_i SP^{x,i}_{A.1} = g_A H_1(PS^x_{A.1}) \tag{4}$$

Same as before, node $A.1$ cannot deduce g_i from $SP^{x,i}_{A.1}$, nor can it obtain g_A from $SP^x_{A.1}$, due to the difficulty in solving the DLP in $\mathbb{G}_1$. It is worth noting that all the requests and replies should be end-to-end encrypted and authenticated to prevent from adversarial access and modification. How to fulfill them is beyond the scope of this paper.

In terms of the choice of the secret-sharing parameters t, n, we have shown in [14] that, when $t = \lceil n/2 \rceil$, and n is equal to either $2\left\lceil \frac{N_A-2}{5} \right\rceil - 1$ or $2\left\lfloor \frac{N_A+3}{5} \right\rfloor - 1$, the maximum security can be obtained. Currently, we are investigating proactive approaches to further improve the security of the proposed scheme, e.g., by dynamically adjusting the shareholder set and the values of t, n to allow dynamic node join/leave without changing g_A while maintaining the highest level of security.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the routing performance of MASK through simulations.

A. Simulation Setup

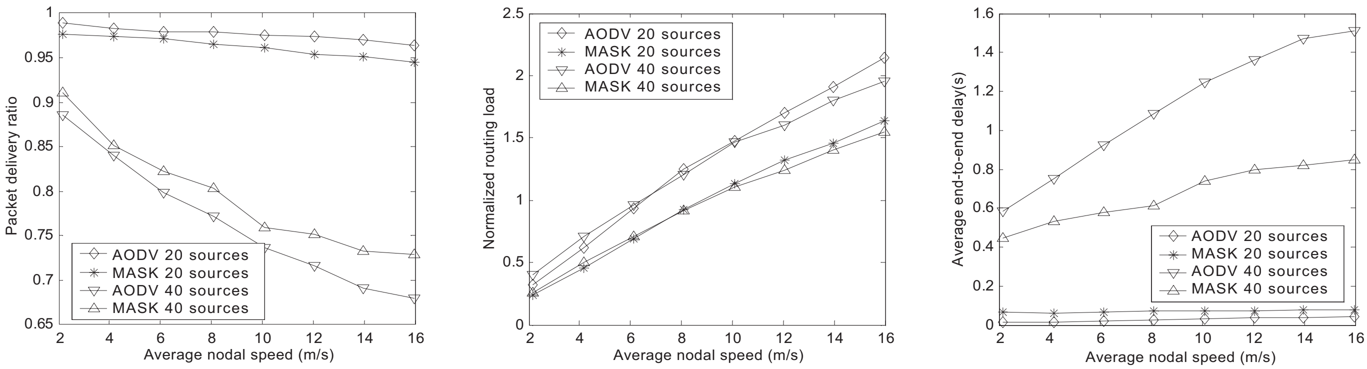
We implement MASK in GloMoSim [15], a popular network simulator for MANETs, and the pairing implementation is based on MIRACL library [16]. The bilinear map $\hat{e}$ we use is the Tate pairing, with some of the modifications and performance improvements described in [7], [8]. We use two security parameters, a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and a 512-bit prime $p = 12qr - 1$ (for some r large enough to make p the correct size). Such bit-length configurations of q, p can deliver a comparable level of security to 1024-bit RSA cryptography. The elliptic curve E we use is $y^2 = x^3 + x$ defined over the finite field $\mathbb{F}_p$ (denoted by $E(\mathbb{F}_p)$). Then $\mathbb{G}_1$ is a q-order subgroup of the additive group of points of $E(\mathbb{F}_p)$, while $\mathbb{G}_2$ is a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. In addition, we use SHA-1 [11] as the hash function H_2 and RC6 [12] as the encryption method used for ARREPs and data packets.

We evaluate the computational costs of critical cryptographic operations in MASK on a Pentium III 1 GHz processor under Windows 2000. For convenience only, we assume the lengths of node pseudonyms, random nonces, Γ, and LinkIDs (also Skeys) to be 8, 4, 2, and 20 bytes, respectively. In fact, the impact of larger lengths on the results is negligible. From Table I, we can see that the most time-consuming operation is the Tate pairing required by anonymous neighborhood authentication. Since the pairing is a relatively new concept, we anticipate that its evaluation cost will be much reduced with the rapid advance in cryptography. For example, Barreto *et al.* [17] recently announce an approach to evaluate the Tata pairing by up to 10 times faster than previous methods, the implementation of which is underway.

Also note that the Tate pairing only needs to be performed once for a pair of neighboring nodes, and then the result can be fed into the fast SHA-1 to compute shared <Skey, LinkID> pairs. Supposing a node maintains $\Gamma = 1000$ <Skey, LinkID> pairs with each neighbor, the computation of such 1000 pairs only costs around 2.4 ms. Hence, when two neighboring nodes run out of the established shared <Skey, LinkID> pairs, they can generate new Γ pairs instantly. Moreover, the hop-by-hop link encryption/decryption operations based RC6 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.

The physical-layer path loss model is the two-ray model. The radio propagation range for each node is 250 meters and the channel capacity is 2 Mb/s. The base MAC protocol used is the DCF of IEEE 802.11, with some modifications according to MASK operations. We simulate an ad hoc network with 50 nodes uniformly deployed in a 700×700 m^2 square field. To emulate node mobility, we modify the random waypoint model in GloMoSim library according to [18] in order to guarantee the convergence of 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. In addition, the pause time is set to be zero, meaning that 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. Each simulation is executed for 15 simulated minutes and each data point represents an average of ten runs with identical traffic models, but different randomly generated mobility scenarios.

In our implementation of MASK, we use a fixed delay of 150 μs into each node to mimic the encryption/decryption processing of ARREPs and data packets with RC6 for simplicity. The purpose is to withstand the aforementioned *message coding attack* (cf. III-D). In addition, the random delay method for data packets to be forwarded is also adopted in each node to thwart the *timing analysis attack* (cf. III-D), where the random delay is uniformly distributed between [0, 50] ms. Furthermore, we set the maximum number of next-hop LinkIDs maintained for one destination to be three. We compare the routing performance of MASK with classical AODV routing protocol [1] with regard to three commonly-used metrics:(1) *Packet delivery ratio* (PDR) – the ratio of



(a) PDR vs. $\bar{V}$.

(b) Normalized routing load vs. $\bar{V}$.

(c) Average packet delay vs. $\bar{V}$.

Fig. 3. The comparison between MASK and AODV.

data packets successfully delivered to the destination over those generated at the sources; (2) *Average end-to-end delay of data packets* – this includes all possible delay caused by buffering during route discovery, queuing delay at the interface, retransmission delay 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 control packet is counted as one transmission.

B. Simulation Results

Fig. 3(a) compares the PDRs 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 [1]. 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 probability 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 advantage mainly results from the aforementioned *virtual multipath* effect in MASK, that is, MASK may simultaneously maintain several next-hop LinkIDs for one given destination. If one of the next-hops becomes unreachable due to mobility or collisions 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 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. 3(b)) because MASK conducts the route discovery less frequently than AODV.

In terms of the average packet delay (Fig. 3(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. 3(c).

In summary, our MASK not only achieves the desirable anonymity without sacrificing the routing efficiency, but also helps improve it under heavy traffic load.

V. Related Work

Anonymous communication protocols have been studied extensively in the wired networks. Chaum [19] defines a layered object that routes data through a chain of pre-deployed intermediate nodes called *mixes*. Following their work, Reed *et al.* propose an interesting Onion routing protocol [20], in which data is wrapped in a series of encrypted layers to form an onion by a series of proxies communicating over encrypted channels. The state of the art of wired networks anonymity can be found in [21]. However, the proposals in the Internet realm cannot be directly applied to MANETs mainly because the prerequisite pre-deployed infrastructure such as the well-known mixes is often unavailable in infrastructureless MANETs.

In contrast, there is little work done to address the anonymity problem and related issues in the context of MANETs. Jiang *et al.* explore the use of mixes in MANETs [22] by designing a mix discovery protocol that allows communicating nodes to choose mix nodes at run time. As noted before, such mix nodes are either unavailable or unreliable in MANETs deployed in hostile environments. The same authors also propose to prevent traffic analysis by using traffic padding, i.e., generating dummy traffic into the network [13], but their work does not aim to enable anonymous communications. Most recently, Kong and Hong propose an anonymous on-demand routing protocol, called ANODR [23], to conceal network IDs of communicating nodes. Besides the computationally intensive route discovery process, ANODR is very sensitive to node mobility, which leads to a low routing efficiency, as the authors mentioned. By comparison, our MASK enables an AODV-like anonymous on-demand routing protocol with high routing efficiency. In addition, MASK addresses anonymous MAC-layer communications, which is left untouched in [23].

VI. Conclusion

In this paper, we propose MASK, a novel anonymous on-demand routing protocol, to enable both anonymous MAC-layer and network-layer communications so as to thwart adversarial, passive eavesdropping and the resulting attacks. By a careful design, MASK provides the anonymity of senders, receivers and sender-receiver relationships, as well as node unlocatability and untrackability and end-to-end flow untraceability. It is also resilient to a wide range of attacks. Detailed simulation studies demonstrate that MASK has comparably high routing efficiency to classical AODV routing protocol while achieving the nice anonymity property.

This paper focuses on dealing with passive attacks and thus there are several unaddressed issues in the current MASK design. First, anonymous neighborhood authentication in MASK relies on pairing operations, which currently have similar computational overhead to conventional public-key operations. Therefore, adversaries might launch active DoS attacks on target nodes by continuously sending a number of bogus authentication requests, which is a problem any authentication scheme has to face. Second, the routing information in the current design is only secured against external adversaries. Once becoming internal adversaries by compromising certain nodes, adversaries can send bogus routing messages that are difficult to verify by legitimate nodes. Third, although pairing-based cryptography is an active research topic nowadays, the implementation on low-end devices is still an open problem.

As the future research, we will first incorporate some intrusion detection capabilities into MASK to defend against not only passive attacks but also active DoS-type attacks such as those mounted on neighborhood authentication. In addition, we will plan to combine MASK with other secure routing protocols such as [4], [5] to ensure both routing anonymity and strong routing security. Finally, we will seek theoretical proofs to show the resilience of MASK to rigorous adversarial cryptanalysis.

Acknowledgment

The authors would like to thank the editor, Dr. Saswati Sarkar, and three anonymous reviewers for their helpful comments and suggestions which have greatly improved the quality of this paper.

References

[1] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, July 2003.

[2] D. Johnson and D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic Publishers, 1996, vol. 353, pp. 153–181.

[3] DARPA, “Research challenges in high confidence networking,” White paper, July 1998.

[4] Y.-C. Hu, A. Perrig, and D. B. Johnson, “Ariadne: A secure on-demand routing protocol for ad hoc networks,” in *ACM MobiCom*, Atlanta, GA, Sep. 2002.

[5] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Royer, “A secure routing protocol for ad hoc networks,” in *IEEE ICNP’02*, Paris, France, Nov. 2002.

[6] A. Menezes, P. van Oorschot, and S. Vanston, *Handbook of Applied Cryptography*. CRC Press, 1996.

[7] D. Boneh and M. Franklin, “Identify-based encryption from the weil pairing,” in *Proc. CRYPTO’01*, ser. LNCS, vol. 2139. Springer-Verlag, 2001, pp. 213–229.

[8] P. Barreto, H. Kim, B. Bynn, and M. Scott, “Efficient algorithms for pairing-based cryptosystems,” in *Proc. CRYPTO’02*, ser. LNCS, vol. 2442. Springer-Verlag, 2002, pp. 354–368.

[9] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon, and H.-C. Wong, “Secure handshakes from pairing-based key agreements,” in *IEEE Symposium on Security & Privacy*, Oakland, CA, May 2003.

[10] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[11] NIST, “Digital hash standard,” Federal Information Processing Standards PUBlication 180-1, April 1995.

[12] R. Rivest, M. Robshaw, R. Sidney, and L. Yin, “The rc6 block cipher,” v1.1, Aug. 1998. [Online]. Available: http://www.rsasecurity.com/rsalabs/rc6/.

[13] S. Jiang, N. Vaidya, and W. Zhao, *Real-time System Security*. Nova Science Publishers, Inc., 2003, ch. Energy Consumption of Traffic Padding Schemes in Wireless Ad Hoc Networks, pp. 21–42.

[14] Y. Zhang, W. Liu, W. Lou, Y. Fang, and Y. Kwon, “AC-PKI: Anonymous and certificateless public-key infrastructure for mobile ad hoc networks,” in *IEEE ICC’05*, Seoul, Korea, May 2005.

[15] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: A library for parallel simulation of large scale wireless networks,” in *Proc. 12 Workshop on Parallel and Distributed Simulations (PADS’98)*, Banff, Alberta, Canada, May 1998, pp. 154–161.

[16] Shamus Software Ltd., “Miracl library.” [Online]. Available: http://indigo.ie/~mscott/.

[17] P. Barreto, B. Lynn, and M. Scott, “On the selection of pairing-friendly groups,” in *Selected Areas in Cryptography – SAC’2003*, ser. LNCS, vol. 3006. Springer-Verlag, 2004, pp. 17–25.

[18] J. Yoon, M. Liu, and B. Nobles, “Sound mobility models,” in *ACM MobiCom*, San Diego, CA, Sept. 2003.

[19] D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Comm. ACM*, vol. 24, no. 2, 1981.

[20] M. Reed, P. Syverson, and D. Goldschlag, “Anonymous connections and onion routing,” *IEEE J. Select. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.

[21] Anonymity bibliography. [Online]. Available: http://freehaven.net/anonbib/

[22] S. Jiang, N. Vaidya, and W. Zhao, “Dynamic mix method in wireless ad hoc networks,” in *IEEE Milcom’01*, Washington, D.C., Oct. 2001.

[23] J. Kong and X. Hong, “ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks,” in *ACM MobiHoc’03*, Annapolis, MD, June 2003.