An Anonymous Routing Protocol with The Local-repair Mechanism for Mobile Ad Hoc Networks

Bo Zhu*, Sushil Jajodia*, Mohan S. Kankanhalli†, Feng Bao‡, Robert H. Deng§

*Center for Secure Information Systems
Volgenau School of Information Technology and Engineering
George Mason University, Fairfax, VA 22030-4444

†School of Computing
National University of Singapore
Singapore 117543

‡Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613

§School of Information Systems
Singapore Management University
Singapore 259756

***Abstract*—In this paper, we first define the requirements on anonymity and security properties of the routing protocol in mobile ad hoc networks, and then propose a new anonymous routing protocol with the local-repair mechanism. Detailed analysis shows that our protocol achieves both anonymity and security properties defined. A major challenge in designing anonymous routing protocols is to reduce computation and communication costs. To overcome this challenge, our protocol is design to require neither asymmetric nor symmetric encryption/decryption while updating the flooding route requests; more importantly, once a route is broken, instead of re-launching a new costly flooding route discovery process like previous work, our protocol provides a local-repair mechanism to fix broken parts of a route without compromising anonymity.**

I. Introduction

Anonymity is an important part of the overall solution for truly secure *Mobile Ad-hoc Networks* (**MANET**), especially in certain privacy-vital environments. For example, in a battle field, we not only want to ensure that adversaries cannot disclose the content of our communications or disable the communications, but also expect that the identities and location information of parties in communications are anonymous to adversaries. Otherwise, adversaries may deduce important information about the location or mobility model of communication parties, which can be used to locate the target of their physical attacks, e.g. the commander, at a later time. There have been several related works [12], [7], [21], [19] addressing the anonymity issue in terms of MANET.

Anonymity achieved in most previous works, including SDAR [7], MASK [21], and AO2P [19], is insufficient. In MASK [21], the real identity of the destination is open to all nodes in the network. In contrast, in SDAR [7], the identities of the source and destination are anonymous to other nodes, but the identities of nodes en route are open to the destination. Therefore, two cooperative adversaries can easily collect the identities of other nodes and their relative locations. In AO2P [19], the location of the destination and the distance between the source and the destination are disclosed during the route discovery process. In SDAR [7], although the exact location of the source is hidden, nodes en route have the knowledge about how far, i.e. the number of hops, they are from the source. In particular, when adversaries know that the source is just one hop away, they can locate the source node using a directed antenna.

A major challenge in designing anonymous routing protocols for MANET is to reduce the communication and computation costs. In previous works, once a route is broken, a new route discovery process is launched, and the new route request will be flooding the whole network. Obviously, the route maintenance process is very costly in dynamic environments like MANET. Optimizations like a local-repair mechanism are desirable.

In this paper, we first define the requirements on the anonymity and security properties of the routing protocol in MANET. Following that, we propose the Efficient Anonymity and Security-Enabled (EASE) routing protocol that can not only protect the privacy of nodes and routes, but also ensure other properties, such as security and efficiency. Detailed analysis in Section V shows that, EASE can achieve both anonymity and security properties defined. Moreover, to mitigate the communication and computation costs, EASE is designed to require neither asymmetric nor symmetric encryption/decryption while updating the flooding route requests before rebroadcasting, and provide a local repair mechanism to repair the broken part of the route without compromising the anonymity.

The rest of the paper is organized as follows. In Section II and Section III, we present the goals and the framework of our works, respectively. The details of our protocol are presented in Section IV. In Section V, we analyze the anonymity and security properties achieved in EASE. The related work is presented in Section VII. Finally, in Section VIII, we draw the conclusion.

II. Design Goals

We define the expected goals or properties that we want to achieve in EASE as follows:

Ensure Privacy 1) *Identity Anonymity*: (a) No one knows the real identities of the source and the destination, except themselves; (b) The source and the destination have no information about the real identities of intermediate nodes en route.

2) *Location Privacy*: (a) No one knows the exact location of the source or the destination, except themselves; (b) Other nodes, including both those nodes outside the route discovered and the intermediate nodes en route, have no information about their distance, i.e. the number of hops, from either the source or the destination. This requirement is optional, but it is desirable in keeping both identity and location anonymity of the source or the destination, especially when the distance is just one hop.

For a protocol satisfying (a), we say that such a protocol provides *Weak Location Privacy*; for a protocol satisfying both (a) and (b), we say that such a protocol provides *Strong Location Privacy*.

3) *Route Anonymity*: (a) Adversaries, either en route or outside the route, cannot trace a packet flow back to its source or destination; (b) For adversaries not in the route, they have no information on any part of the route; (c) It is difficult for adversaries to infer the transmission pattern and motion pattern of the source or the destination;

Ensure Security The protocol should ensure that the discovered route could function properly (namely, the protocol can find the route correctly and efficiently) under different attacks.

Ensure Efficiency The protocol should be efficient in the terms of both computation and communication costs.

III. The Framework of EASE

In this section, we first present the framework of the EASE protocol, including system model, adversary model, and network model. Afterwards, we give an outline of the EASE protocol before presenting the details in Section IV.

System Mode We assume that there are a large number of users in a mobile ad hoc network, a small part of which are adversaries. We assume that there is a shared secret between the source and the destination by employing some anonymous end-to-end key agreement, e.g. TESLA [15].

Adversary Model We assume that, adversaries have the same eavesdropping and computing capabilities as normal nodes and certain intrusion capability. Adversaries may try to compromise their neighbors only when some evidence shows that the source or destination is only one or two hops away or a node en route is one hop away. To find out a node en route, which is a few hops away, does not provide sufficient incentives for adversaries to compromise their neighbors. We assume that adversaries have no prior information about potential senders in the future. In other words, from the view of adversaries, each node in the network has the same possibility of being a sender launching a route discovery process.

We assume that adversaries may launch both passive and active attacks at the same time, and the information obtained from the former can be used to enhance the effectiveness of the latter. We consider attacks from both internal nodes (i.e. en route) and external nodes (i.e. out of the route).

Network Model We assume that, wireless links are symmetric. Namely, if node A is in transmission range of some node B, then B is in transmission range of A as well. Each node can change the source address of its outgoing MAC frames, so that adversaries cannot trace the node based on its unique MAC address.

A. Outline of The EASE Protocol

The whole protocol consists of the following procedures: *Route Request, Route Reply, Data Transmission, and Route Maintenance*.

At the beginning of the *Route Request* procedure, the source broadcasts the request to its neighbors, and the request is forwarded recursively and flooding the whole network. Before forwarding the packet, the intermediate node modifies the packet in two ways: (1) replace the one-time public key of the upstream node with the one of itself; (2) update a specific field of the packet denoted as U_i. As such, at the end of the *Route Request* procedure, each node has the one-time public key of its upstream node en route. Making use of the shared secret with the source, the destination can deduce the length of the route.

Similar to route request, route reply is forwarded recursively till reaching the source. However, since the reply is encrypted with one-time public key of the upstream node, it will only be forwarded along the route instead of flooding. Before forwarding the packet, the intermediate node modifies the packet in two ways: (1) generate a random secret, which will be used as the shared secret with the upstream node during the session, and encrypt it with the upstream node's one-time public key; (2) update a specific field of the packet denoted as V_i. At the end of the *Route Reply* procedure, each node en route has shared secrets with its upstream and downstream nodes, if exist. In addition, all the nodes en route hold the public key of the PKI pair generated by the destination for current session. Moreover, making use of the shared secret with the destination, the source can deduce additional information from the route reply, including the private key of the PKI pair and the shared secrets between each pair of consecutive nodes en route.

Using the shared secrets with the upstream and downstream node, each node en route can establish anonymous data transmission. The local repair mechanism is built on the fact that the source knows the shared secrets between each pair of consecutive nodes en route, although it has no knowledge about their identities.

IV. EASE Routing Protocol

The notions of the types of packets involved in the route discovery process are shown in Table I. In this paper, let $X(s)$ denote a symmetric encryption/decryption on s using X as the key and $\{s\}_X$ denote an asymmetric encryption/decryption on s using X as the key.

RREQ	Route Request Packet	RREP	Route Reply Packet
DATA	Data Packet	RERR	Route Error Packet
RRPR	Route Repair Packet	RUPD	Route Update Packet

TABLE I

NOTIONS OF TYPES OF PACKETS INVOLVED IN ROUTE DISCOVERY

We denote the source node, nodes en route, and the destination node as S, X_i ($i = 1, 2, \ldots, n$), and D, respectively. n denotes the number of nodes between the source and the destination.

A. Route Request

During the route request process, each node en route denoted as X_i ($i = 1, 2, \ldots, n$) receives a route request with the following format:

$$\left[\begin{array}{c} RREQ,\ seq,\ K_T(dest, K_s, U_{orig}), \\ K_s(seq, END),\ PK_{i-1},\ U_{i-1} \end{array}\right]$$

where

seq — the session number.

K_T — the secret shared between the source and destination[1].

dest — the identity of the destination D.

K_s — a session key of current session.

END — a sign showing that the destination has received the route request.

PK_{i-1}— the public key of the one-time key pair generated by the upstream node X_{i-1}. PK_0 is chosen by the source S.

U_{orig} — a random number chosen by the source S.

U_{i-1} — a number generated by X_{i-1}. U_0 is generated by the source S.

To avoid the collision, seq is set to be a random number with the sufficient length, e.g. 160 bits.

For U_i $(i = 1, 2, \cdots, n)$ in RREQ, X_i computes it according to Equation (1):

$$U_i = f(U_{i-1}, S_i) = (U_{i-1} \oplus S_i) \gg p_x, \tag{1}$$

where S_i is a random number chosen by X_i with size p_x. When $i = 0$, U_0 is calculated by the source S according to Equation 2:

$$U_0 = f(U_{orig}, S_0) = (U_{orig} \oplus S_0) \gg p_x, \tag{2}$$

where U_{orig} and S_0 are random numbers chosen by the source S with size p_s and p_x, respectively. Note that, in Equation (1) and (2), $\oplus$ means the operation that S_i or S_0 is XORed with the least p_x bits of U_{i-1} or U_{orig}. Thus, the computation denoted by Equation (1) and (2) includes two steps. The output of the first step is a number with size p_s. The least p_x bits of the output is the result that S_i or S_0 XORs with the least p_x bits of U_{i-1} or U_{orig}, while the higher bits are the same as the corresponding bits of U_{i-1} or U_{orig}. The next step is to rotate the result of the first step right for p_x bits.

Let H_{max} denote the maximum number of hops that S wishes the route to be. Then, we have:

$$p_s = (H_{max} + 1) \cdot p_x \tag{3}$$

For instance, given that the length of the random number chosen by X_i, i.e. S_i, is 16 bits, the source wants to discover a route between the destination and itself, and expects the length of the route is no more than 10 hops (i.e. $H_{max} = 10$). According to Equation (3), we know that $p_x = 176$, and thus generate a random number U_{orig} with 176 bits during the generation of the route request packet.

Once receiving the RREQ packet, each forwarding node X_i first checks whether seq has been recorded in its RREQ buffer table. If yes, it simply discards the packet. Otherwise, X_i tries to decrypt $K_T(dest, K_s, U_{orig})$ by using all the secrets that it shares with other nodes.

X_i succeeds only if it has a shared secret with the source and it is the intended destination of the route discovery process. If it fails, X_i first adds a new record into the RREQ buffer table. The format of a record in the RREQ buffer table of X_i is shown as follows:

$$[seq,\ PK_{i-1}, K_s(seq, END)]$$

Then X_i generates U_i as shown in Equation (1), and replaces PK_{i-1} and U_{i-1} with its one-time public key[2] (i.e. PK_i) and U_i, respectively. Finally, X_i broadcasts the modified packet locally.

If succeeds, it means that X_i is the destination of this route request, since only the destination can successfully decrypt the packet. Afterwards, D compares U_{orig} to U_n (i.e. the sixth element of the RREQ packet), and figure out the exact distance from the source, if the lengths of the route found is less or equal to the hop limit. Thereafter, depending on whether EASE provides the multiple-path functionality, D may send out a RREP packet for each route with less than H_{max} hops or only for the first such route, and at the same time adds a new record into its local route table. To counterattack sniffing, unlike AODV [14], the destination needs to forward RREQs received like intermediate nodes.

B. Route Reply

During the route reply process, each node en route denoted as X_i ($i = 1, 2, \ldots, n$) receives a route reply with the following format:

$$\left[\begin{array}{c} RREP,\ \{T_{i+1}\}_{PK_i}, \\ T_{i+1}(seq, K_s', TPK, V_{i+1}, K_T(H_{route}, V_{orig}, TSK)) \end{array}\right]$$

where

[1] There have been extensive research in key distribution in mobile ad hoc networks, which is beyond the scope of this paper.

[2] The overheads of generating one-time public/private key pairs can be mitigated with pre-computation.

T_{i+1} — a random number chosen by X_{i+1}, which is used as the shared secret between X_i and X_{i+1} after the routing discovery process.

K'_s — the proof that D has recovered K_s from the RREQ packet.

TPK, TSK — the key pair generated for the current session.

V_{i+1} — a number generated by X_{i+1}. V_{n+1} is generated by the destination D.

H_{route} — the number of hops that this RREP packet is forwarded before reaching the source.

V_{orig} — a random number chosen by D.

For V_i $(i = 1, 2, \cdots, n)$ in RREP, X_i computes it according to Equation (4):

$$V_i = g(V_{i+1},\ T_i) = (V_{i+1} \oplus T_i) \gg q_y \tag{4}$$

where T_{i+1} is a random number chosen by X_{i+1} with size q_y. When $i = n+1$, V_{n+1} is calculated by the destination D according to Equation (5):

$$V_{n+1} = g(V_{orig},\ T_{n+1}) = (V_{orig} \oplus T_{n+1}) \gg q_y \tag{5}$$

V_{orig} and T_{n+1} are random numbers chosen by D with size q_d and q_y, respectively.

For the sake of anonymity, q_d cannot be equal to $H_{route} \cdot q_y$. Otherwise, adversaries can easily obtain the information about the route length by sniffing the RREP packets. Therefore, q_d should be set a few q_x bits longer than $H_{route} \cdot q_y$. Namely, we have

$$q_d > H_{route} \cdot q_y$$

For example, if we assume that the H_{route} of a given RREP packet is 7 and q_y is set to be 128 bits, we may set q_d to be 1280 bits. Alternatively, we can set q_d according to Equation (6), in spite of what is the exact length of the route, although it might be communicationally inefficient when the route length is much shorter than H_{max}, e.g. when $H_{route} = 1$ and $H_{max} = 10$.

$$q_d = (H_{max} + 1) \cdot q_y \tag{6}$$

Once receiving the RREP packet, each forwarding node X_i first tries to decrypt $\{T_{i+1}\}_{PK_i}$, and recovers the last element of the packet. Since the last element is encrypted by T_{i+1}, only X_i can decrypt it. Then X_i extracts seq from the recovered information, and checks whether seq has been recorded in its RREQ buffer table. If no, it simply discards the packet without any furtherer checking. Otherwise, X_i extracts K'_s from the recovered information. Thereafter, X_i also needs to make sure that the RREP packet is from the destination. It can be verified by Equation (7), because only the destination D can recover K_s from the RREQ packet. If Equation (7) is not satisfied, X_i simply discards this RREP packet.

$$K'_s(seq, END) \overset{?}{=} K_s(seq, END), \tag{7}$$

After successfully verifying the validity of the RREP packet, X_i chooses a random number T_i with size q_y, and computes V_i according to Equation (4). Following that, X_i builds a new record in its route table. Then computes $\{T_i\}_{PK_{i-1}}$ and $T_i(seq, K'_s, TPK, V_i,\ K_T(H_{route}, V_{orig}, TSK))$, which are used to replace the last two elements of the RREP. Finally, the modified RREP packet is broadcasted locally.

Upon receiving the RREP packet, by comparing V_{i+1} with V_{orig}, the source S can extract the H_{route} and the shared secrets along the route (i.e. $T_1,\ T_2, \ldots,\ T_{H_{route}}$) from the packet, and then record them into the local route table. In our scheme, instead of only recording the shared secret with the first forwarding node, the source needs to record all the shared secrets from itself to the destination. In Section IV-D, we present how this additional effort benefits the route maintenance process. Since these secrets are generated randomly and are used only for this specific route from S to D, the source cannot deduce the identities of those forwarding nodes with the knowledge.

C. Data Transmission

To realize anonymous data transmission, we need to make sure that adversaries are not able to read or deduce information about the source and destination from data packets, and such information is only open to entities holding corresponding secrets. It is definitely not a good idea to encrypt the whole data packet using the shared secrets, although this solution is workable in theory; otherwise, each node has to try to decrypt the whole content of every packet received before deciding whether to accept it or not. Consequently, this method requires a huge amount of computation.

In EASE, we provide a solution by making use of the shared secrets between any two consecutive nodes (i.e. T_i). Our idea is to construct some small-size information (denoted as TAG) which is sent together with the data packet so that a forwarding node only needs to verify TAG instead of the whole packet. It is similar to the construction of route pseudonym in ANODR [12], but is more simple and efficient. The TAG is constructed as follows. Given that, node X_{i-1} and node X_i share a secret denoted as T_i. Let $H_K()$ be a keyed fast one-way function, which uses K as the key. The format of TAG on the packet between X_{i-1} and node X_i, denoted as TAG_i, is calculated as $H_{T_i}(N)$, where N is a non-decreasing number. N is initialized to 1 at both sides (i.e. the two consecutive nodes en route) upon the establishment of the shared secret.

We assume that, before sending out the data packet, the real message transmitted has been padded to a uniform length and encrypted with the secret shared between the source and the destination. The details about padding and encrypting the real message into the data portion of the data packet denoted as PL are beyond the scope of this paper.

During the data transmission, each node en route denoted as X_i $(i = 1, 2, \ldots, n)$ receives a data packet with the following format:

$$[DATA,\ TAG_i,\ T_i(PL)]$$

where

TAG_i — the current TAG that X_i shares with X_{i-1}.

T_i — the secret that X_i shares with X_{i-1}.

Once receiving the DATA packet, X_i verifies the validity of the TAG_i. If the packet passes the verification stage, X_i replaces TAG_i by TAG_{i+1} which is the current TAG that X_i shares with X_{i+1}. In addition, before broadcasting the packet, the content of the data packet should be shuffled, i.e. decrypt the last element of the DATA packet with T_i and then encrypt with T_{i+1}, so that the adversaries outside the route cannot match payload contents to trace data forwarding. If the DATA packet fails to pass the verification, it is discarded. Such a process is repeated until the DATA packet reaches the destination.

D. Route Maintenance

1) Motive of Providing The Local Repair Functionality: In MANET, we wish that the discovered anonymous route is robust and efficient against failures due to the following reasons: (1) node mobility; (2) join/leave operations of nodes; (3) nodes en route are hacked, and refuse to provide the data-forwarding function. A straightforward solution is to re-launch the route discovery process. All previous anonymous routing protocols follow this method. However, we argue that route requests are broadcasted to flood the whole network, and thus it is costly to launch a new route discovery process.

In EASE, we make use of the shared secrets along the route found, which are obtained during the previous route discovery process, to find a usable route with less computation and communication overheads, and at the same time, make sure that the repairing process does not impair the anonymous property of the route. The idea of designing the locally-repairing mechanism is based on two observations: (1) The two communication parties, namely the source and destination, can have some extra privileges over nodes en route, as long as such privileges do not compromise the anonymity of other nodes; (2) Knowing the secret shares along the route does not help the source deduce the identities of the forwarding nodes en route, because shared secrets used by any part of the route are totally randomly chosen and are used only for this specific route from S to D. Similarly, for nodes en route, the knowledge of the shared secrets along a middle part of the route does not compromise the anonymity of nodes en route.

2) The Local Repair Mechanism: We assume that, nodes can detect route failures when re-transmission count exceeds a predefined number. For example, a node en route X_i detects that the route to X_{i+1} is not available any more. Upon detection, it looks up the corresponding entry in its forwarding table, finds the current TAG information that it shares with the previous node, i.e. TAG_i, and the secret shared with the next node, i.e. T_{i+1}, and then broadcasts a route error packet. For a node en route between the source and X_i, denoted as X_j, it receives a RERR with the following format:

$$[RERR,\ TAG_{j+1},\ T_{j+1}(\{T_{i+1}\}_{TPK})]$$

where

TAG_{j+1} — the TAG that X_j shares with X_{j+1}.
T_{j+1} — the shared secret that X_j shares with X_{j+1}.
T_{i+1} — the shared secret that X_i shares with X_{i+1}.

Once receiving the RERR packet, X_j first verifies the validity of the TAG_{j+1}. If the packet passes the verification stage, X_j replaces TAG_{j+1} by TAG_j which is the current TAG that X_j shares with X_{j-1}. In addition, before broadcasting the RERR packet, the last element should be shuffled in a similar way as the DATA packet so that the adversaries outside the route cannot match payload contents to trace the route. If the RERR packet fails to pass the verification, it is discarded. Such a process is repeated until the RERR packet reaches the source.

After extracting T_{i+1} from the RERR packet, S compares it with the record in its local route table, and finds out the exact node, X_i here, reporting the route error. To discover a new route to the destination, S sends out a RRPR along the previous route. For a node en route between the source and X_i, denoted as X_j, it receives a RRPR with the following format:

$$[RRPR,\ TAG_j,\ T_j(T_{i+1}(T_{i+2}))]$$

where

TAG_j — the TAG that X_j shares with X_{j-1}.
T_j — the shared secret that X_j shares with X_{j-1}.
T_{i+2} — the shared secret that X_{i+1} shares with X_{i+2}.

Once receiving the RRPR packet, X_j first verifies the validity of the TAG_j. If the packet passes the verification stage, X_j replaces TAG_j by TAG_{j+1} which is the current TAG that X_j shares with X_{j+1}. In addition, before broadcasting the RRPR packet, the last element should be shuffled in a similar way as the DATA packet so that the adversaries outside the route cannot match payload contents to trace the route. If the RRPR packet fails to pass the verification, it is discarded. Such a process is repeated until the RRPR packet reaches X_i, i.e. the node reported the route failure.

When the RRPR packet reaches X_i, X_i decrypts it and extracts T_{i+2} which becomes a shared secret between X_i and X_{i+2}. Following that, X_i launches a route request from itself to X_{i+2}. It is similar to the previous route discovery process between S and D but with fewer hops. The format of the local repair RREQ packet is shown as follows:

$$\left[\begin{array}{c} RREQ,\ seq,\ T_{i+2}(Broadcast, K_s, U_{orig}), \\ K_s(seq, END),\ PK_{i-1},\ U_{i-1},\ Time_{LR} \end{array}\right]$$

where

Broadcast — a broadcasting address.
$Time_{LR}$ — the time that this local repair RREQ timeouts.

One major difference is that the identity of the destination in the RREQ packet is replaced with a broadcasting address. The other major difference is that in the local repair process, we import a new element denoted as $Time_{LR}$ into the RREQ packet. $Time_{LR}$ indicates when this local repair RREQ timeouts. The setting of $Time_{LR}$ is related to the random jitter before sending out a packet. A receiver simply ignores the local repair RREQ packet, if it is timeout.

Here, we assume that a new route via X'_{i+1} is found. Upon the completion of the route discovery between X_i and X_{i+2}, X_i sends out a route update packet. For a node en route between the source and X_i, denoted as X_j, it receives a RUPD with the following format:

$$[RUPD,\ TAG_{j+1},\ T_{j+1}(\{T'_{i+1}, T'_{i+2}\}_{TPK})]$$

where

T'_{i+1} — the shared secret that X_i shares with X'_{i+1}.
T'_{i+2} — the shared secret that X'_{i+1} shares with X_{i+2}.

Once receiving the RUPD packet, X_j verifies the validity of the TAG_{j+1}. If the packet passes the verification stage, X_j replaces TAG_{j+1} by TAG_j which is the current TAG that X_j shares with X_{j-1}. Again, the last element of the RUPD packet should be shuffled before broadcasting. Such a process is repeated until the RUPD packet reaches S. Finally, S updates the corresponding record in its route table with T'_{i+1} and T'_{i+2}.

V. Analysis on Anonymity and Security

Firstly, we need to make clear that the *Security* term discussed in this section does not include issues about security of the content of data packets being transmitted. It is easy to see that security of the content of data packets is orthogonal to anonymity and security of the route protocol.

A. Anonymity Analysis

Here, we want to check whether EASE has achieved anonymity-related goals defined in Section II, namely *Identity Anonymity*, *Location Privacy*, and *Route Anonymity*. In the context of anonymity analysis, we assume that all the nodes including nodes on the discovered route are potential adversaries and are interested in the privacy information about the two communication parties and discovered routes.

Generally, the methods of breaking anonymity can be divided into two categories: traffic-based analysis [17] and protocol-based analysis. The idea behind traffic-based analysis is to detect common information among sniffed packets, and assume that any two packets are transferred along the same route, if they have information in common. The "common information" could be either identical content in sniffed packets, or identical time consumed by handling sniffed packets. This kind of analysis might be executed independently from the contexts of the protocols being analyzed. A typical traffic-based analysis is time analysis, where the adversary can use temporal dependency between transmissions to trace a victim message's forwarding path. In contrast, in protocol-based analysis, adversaries try to deduce the information of the sender through investigating the semantic context. For example, they may obtain the identity-related information from the meaningful content of the packet sniffed, or find out certain pattern of variations based on the additional knowledge/understanding of the targeted protocols.

There have been extensive research in protecting the route protocol and data transmission from traffic-based analysis, and most of them can be easily integrated with solutions against protocol-based analysis. Unless otherwise specified, therefore, in this paper we mainly focus on the protocol-based analysis, and assume the existence of methods that prevent or mitigate traffic-based analysis (e.g. traffic mixing technique [16], [10], [3]) in all the anonymous routing and communication protocols for MANET.

1) Identity Anonymity: In EASE, there is no identity-related information involved except the destination's identity, namely $dest$, in the RREQ packet. Fortunately, $dest$ is encrypted by the shared secret between the source and the destination, and thus it is known only to the two communication parties.

2) Location Privacy: Theoretically, it is possible that adversaries discover the location of the sender, if the sender happens to be surrounded by adversaries (e.g. in a triangle) and they keep sniffing the traffic to and from the sender long before it launches the route discovery process. As a result, even if mixing techniques are employed, adversaries can distinguish the route request originated from the sender from that forwarded by it, given that the message delay is reasonable. However, in this paper we assume that the number of adversaries in the network is small, and they have the same eavesdropping and computing capabilities as normal nodes. Besides that, adversaries have no prior information about potential senders in the future. Therefore, we argue that the possibility of such cases is negligible.

The idea of current practical attacks on *Location Privacy* is to overhear the routing packets and make use of the semantic weaknesses of the protocol to obtain the exact location or deduce the relative location (i.e. the distance from the target) of the source or the destination. A typical attack frequently used by attackers is to observe the variation on the length of the packet while it is forwarded, and a few anonymous routing protocols proposed [7], [5], [4], [6] are vulnerable to these attacks.

EASE is robust against attacks aiming at location privacy. There is no explicit location information involved in the protocol. More importantly, for all kinds of the routing packets, the lengths of both the whole packet and each element inside are constant when the packet updated and transmitted along the route so that even internal nodes cannot deduce how far they are from the source or the destination. Moreover, we notice that, to deduce the distance, adversaries need to not only discover the pattern of variations but also find an anchor point, which is corresponding to the source or the destination, so that they can measure the times of variations from the anchor point, i.e. the distance from the source or the destination. In EASE, the only pattern of variations that can be detected is U_i in RREQ and V_i in RREP. Fortunately, since U_{orig} and V_{orig} are secrets shared by the source and destination only and are indistinguishable from random numbers for other nodes, adversaries fail to find the anchor point.

3) Route Anonymity: In EASE, hop-by-hop shuffle is employed to prevent adversaries from matching the content of packets. On one hand, we use TAG to ensure efficient data forwarding between consecutive nodes along the route established. For adversaries without the secrets generating the

TAGs, both the TAGs generated between a pair of consecutive nodes for different data packets and the TAGs generated between different pairs of consecutive nodes for the same data packet are deemed to be random numbers generated independently. As a result, adversaries fail to link data packets sniffed using the TAGs. On the other hand, the meaningful contents of all types of routing packets except RREQ[3] are encrypted with the shared secret between the consecutive nodes along the route. Thus, the content of the same packet evolves hop by hop, and all the evolutions are deemed to be random numbers generated independently.

Moreover, in EASE hop-by-hop shuffle prevents adversaries outside the route from detecting patterns of variations in the meaningful contents of the routing packets. For internal adversaries, although they can discover the patterns of variations, they cannot trace a packet flow back to the source or destination due to the failure of detecting the anchor point.

B. Security Analysis

1) Passive Attacks: The simplest attack on the route protocol is that adversaries or selfish nodes silently refuse to perform functions requested in the protocol. In normal routing protocols, the watchdog model [13] can be employed to detect such actions. However, in anonymous routing, the route reply is modified hop-by-hop and is supposed to be undistinguishable from other route replies. Therefore, by nature, we can not figure out which route a given sniffed route reply belongs to, since it is a trade-off between anonymity and security. The only usable solution is to discover and maintain multiple routes at the stage of route discovery.

2) DoS Attacks: According to the target of the attack, DoS attacks in the context of anonymous routing can be classified into two types: *Multiple-to-One* attacks and *One-to-Multiple* attacks. In the former attacks, multiple adversaries (or one adversary with strong power) may cooperate to exhaust the resource of a given target. The most critical step of such attacks is to identify the target, either its identity or its exact location. EASE is immune to this type of attacks, since both *Identity Anonymity* and *Location Privacy* are ensured in EASE. As to the latter attacks, one adversary can send fake route request or route reply packets which exhaust the computation resources of other nodes, since those nodes would perform the cryptographic computation as requested in the protocol. In EASE, such attacks are mitigated by (a) little computation, i.e., a XOR operation and a rotation, is involved in updating the RREQ packet before rebroadcasting; (b) employ the hop-by-hop authentication on the RREP packet.

3) Attacks on Route Maintenance: One possible attack is that adversaries send fake route error packets to fool the source to choose another route or even re-launch the route discovery process. It makes no sense when adversaries en route launch such an attack. Therefore, in the context of attacks on route maintenance, we only consider adversaries which are not in the route. In EASE, no adversary out of the route can construct fake route error packets, because it does not hold any secret with any node en route, which is necessary to generate the TAG in the route error packet.

4) Wormhole Attacks: In *Wormhole Attacks* [8], an attacker records packets received at one location in the network, tunnels them to another location, and retransmits them into the network. Hu, Perrig, and Johnson proposed an approach to detect wormhole attacks based on packet leashes [8]. The key intuition is that by authenticating either an extremely precise timestamp (i.e., *temporal leashes*) or location information combined with a loose timestamp (i.e., *geographical leashes*), a receiver can determine if the packet has traversed a distance that is unrealistic for the specific network technology used. Both of the solutions can be easily integrated into EASE without any conflict.

VI. Simulation Results & Efficiency Analysis

In this section, we analyze the efficiency of EASE and compare it with other generic anonymous routing protocols [12], [7], [5], [4], [6] from two aspects: computation costs and communication costs.

A. Computational Costs

In Table II[4], we show the benchmark of typical symmetric (AES) and asymmetric (RSA) decryptions on both low-end (i.e. iPAQ3670 with Intel StrongARM 206MHz CPU) and high-end (i.e. Pentium IV) devices.

TABLE II

Cryptographic Benchmark of AES and RSA in Decryption

	AES (128 bits)	RSA (1024 bits)
low-end device	29.2 Mbps	900 ms
high-end device	488.08 Mbps	4.77 ms

The computation costs required for an intermediate node to handle a RREQ packet consist of three parts: generating one-time public/private key pairs, checking whether it is the destination of this route request, updating the RREQ packet before rebroadcasting. The first part of costs can be mitigated through pre-computation. The second part is proportional to the number of nodes that have shared secrets with the intermediate node and may launch route discovery towards it, denoted as t. EASE and ANODR [12], [11] requires t symmetric decryptions, while SDAR [7], [5], [4], [6] needs one asymmetric decryption. The third part of cost is shown in Table III.

In Table III and Table IV we compare the cost of updating the RREQ packet before rebroadcasting and the cryptographic operations that intermediate nodes perform on a RREP packet, respectively. AO and SO in Table III and IV denote the numbers of asymmetric and symmetric operations executed, respectively. In all anonymous routing protocols, the number

[3] Part of meaningful content in the RREQ packet, including seq, PK_{i-1}, and U_{i-1}, are not encrypted. Since the route has not been set up at that moment, adversaries cannot take advantage from such information.

[4] The results are obtained from [12] and [1], respectively. Note that, the unit for the benchmark of AES on the Pentium IV desktop has been converted from MB to Mbps for easy comparison.

of route requests is much higher that of route replies, because the route request is forwarded to flood the whole network, while the route reply is only forwarded reversely along the route found. Therefore, overall, EASE and ANODR have better performance compared to SDAR. And EASE is slightly efficient than ANODR, since no cryptographic operation is involved while updating the RREQ packet.

	AO	SO	Others
SDAR	2	0	none
ANODR	0	1	none
EASE	0	0	one XOR and one rotation

TABLE III
OPERATIONS FOR HANDLING A ROUTE REQUEST

	AO	SO	Others
SDAR	0	1	none
ANODR	2	2	none
EASE	2	2	none

TABLE IV
CRYPTOGRAPHIC OPERATIONS FOR HANDLING A ROUTE REPLY

B. Communication Costs

The routing protocols and related simulation models are implemented using *Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator* (*JiST/SWANS*) [2]. We run the simulation in a 1000m X 1000m network, and the node transmission radius is set to be 225m. The Random Waypoint model [9] is applied to emulate node mobility pattern. According to the model, a node travels to a random chosen location in a certain speed and stays for a while before going to another random location. To avoid the problem indicated in Yoon et al.'s paper [20], the minimum speed is set to 1 m/s. In addition, the routing discovery process begins after a 300-second warm-up period to eliminate the initial drop in average node speed [20]. For each test, a pair of source and destination is randomly chosen, and 100 data messages have been sent from the source to the destination at the speed of one message per second.

In the simulation, we design and evaluate two simulation models: *SP-NLR* (only store the first route found, and do not employ the local repair mechanism) and *SP-LR* (only store the first route found, and employ the local repair mechanism provided). All previous anonymous routing protocols only support $SP-NLR$, and EASE can support both of them.

We evaluate anonymous routing protocols using the following metrics: (1) the number of route discovery packets (including RREQ, RREP, RERR, and RRPR, RUPD, if any) received (denoted as RDIS); (2) the number of route reply packets (denoted as RREP); (3) the success rate of transmitting data messages (denoted as SucRate). The first two metrics not only measure the communication cost but also help indicate the overall computation cost of determining the route and transferring data packets. RREP is of particular interests due to its expensive asymmetric operations. The last metric is to measure the overall network performance.

Given that, the speed of any node in the network is chosen randomly from 1 to V_{max}. Five different mobility settings, i.e. $V_{max} = 16-20\ m/s$, are simulated to analyze the efficiencies of the two models. Figure 1 and 2 show that, SP-LR is more efficient than SP-NLR. More specifically, RDIS of SP-LR is around 8% to 17% less than that of SP-NLR. Similarly, RREP of SP-LR is around 7% to 17% under that of SP-NLR. As to the success rate of transmission, Figure 3 shows that the SP-LR model provides a better network performance. SucRate of SP-NLR is 1.40% to 2.55% smaller under all the settings simulated.

According to the empirical results, as the only anonymous routing protocol supporting the local-repair mechanism, EASE provides better efficiency than previous work.

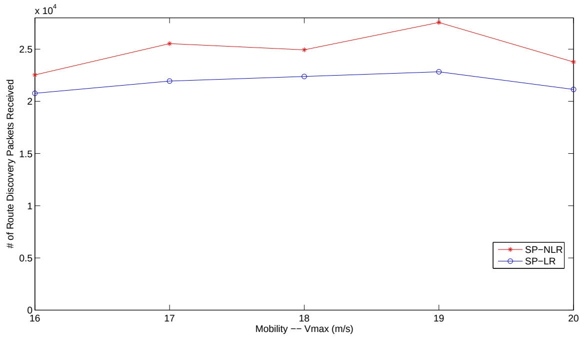

Fig. 1. # of Route Discovery Packets Received under Different Mobility Settings

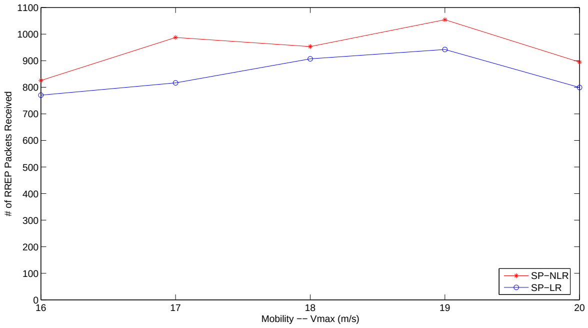

Fig. 2. # of RREP Packets Received under Different Mobility Settings

VII. RELATED WORK

In [12], [11], Kong and Hong designed the first anonymous routing protocol for MANET, i.e. the ANonymous On Demand Routing (ANODR) protocol. Similar to Hordes, ANODR [12], [11] also explores multicast/broadcast to improve recipient anonymity. However, ANODR is an on-demand protocol, and it extensively explores trapdoor information in broadcast. These features are not discussed in Hordes' multicast mechanisms. Compared to [7], [5], [4], [6], Kong and Hong gave a

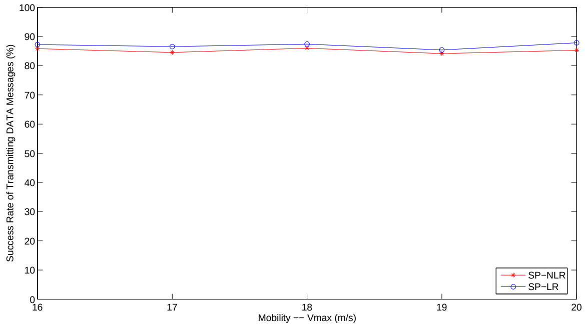

Fig. 3. Success Rate of Transmitting DATA Messages (%)

more comprehensive analysis on the anonymity and security properties achieved, and provided detailed simulation results on the efficiency of ANODR. In addition, ANODR is more efficient than [7], [5], [4], [6] at both the route discovery and the data transmission stages. An insider can deduce the hop count between itself and the source in the old version of ANODR presented in MOBIHOC'03 [12]. After June 2004, as the UCLA PHD thesis [11] has fixed the problem, the current ANODR is not vulnerable to the attack.

In [7], [5], [4], [6], El-Khatib et al. proposed a secure dynamic distributed routing algorithm for ad hoc wireless networks, which is based on the onion routing protocol [18]. SDAR can protect the exact location of the communication parties and the anonymous route found during the route discovery process from adversaries. However, the identities of nodes en route are open to the destination node. Therefore, two cooperative adversaries can easily collect identities of other nodes, and even know the relative locations of these nodes. It is certainly undesirable in the real world. Moreover, *Strong Location Privacy* is not provided in [7], [5], [4], [6].

In [22], Zhu et. al. proposed the Anonymous Secure Routing (ASR) protocol. It identifies a few weaknesses in the old version of ANODR presented in MOBIHOC'03 [12] and SDAR [7], [5], [4], [6]. Compared to ANODR and SDAR, ASR is more efficient, because it requires only a XOR and a shift operation, instead of the symmetric [12], [11] or asymmetric [7], [5], [4], [6] encryptions, while updating the content of the route request packets before rebroadcasting them.

In [21], Zhang et al. proposed an anonymous on-demanding routing protocol, termed MASK. In this protocol, each node is assigned a large set PS_i of collision resistant pseudonyms by an off-line TA beforehand. It cannot generate random pseudonyms by itself. Due to the limitation of storage on ad hoc network nodes, they cannot store a large number of pseudonyms, and thus these pseudonyms may be used out soon. Besides that, in MASK the real identifier of the destination node is disclosed to all the nodes en route during the route discovery process.

Wu and Bhargava proposed an anonymous routing protocol in [19], i.e. AO2P, in which real identities for the source nodes, the destination nodes, and the forwarding nodes in the end-to-end connections are kept private. However, in AO2P, the position of the destination is exposed for route discovery. In addition, because the distance from the source to the destination is included in the route request, the location of the source in fact is also partially disclosed. AO2P assumes the existence of a secure position service system, which requires a number of fixed servers, and thus is not suitable for purely mobile ad hoc networks.

VIII. Conclusion

Anonymity is a very important part of the overall solution for securing mobile ad-hoc networks. In this paper, we defined more strict requirements on the anonymity and security properties of the routing protocol in MANET, and proposed a new anonymous routing protocol, i.e. EASE, that can provide the anonymity properties defined and at the same time ensure the security of discovered routes against various passive and active attacks. We also gave a detailed analysis on how anonymity and security are achieved in EASE. Simulation results showed that, compared to previous work, EASE is more efficient in the sense of both computation and communication costs, and is suitable for highly dynamic environments like MANET.

References

[1] Crypto++ 5.2.1 benchmarks. http://www.eskimo.com/ weidai/benchmarks.html.

[2] Rimon Barr, Zygmunt J. Haas, and Robbert Van Renesse. JiST: Embedding simulation time into a virtual machine. In *EuroSim Congress on Modelling and Simulation*, September 2004.

[3] Oliver Berthold, Hannes Federrath, and Stefan Kospell. Web MIXes: A system for anonymous and unobservable internet access. In *Proceedings of Workshop on Design Issues in Anonymity and Unobservability (DIAU'00), Lecture Notes in Computer Science 2009*, pages 115–129, 2000.

[4] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. A novel solution for achieving anonymity in wireless ad hoc routing protocol. In *Proceedings of Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (ACM PE-WASUN'2004)*, November 2004.

[5] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. SDAR: A secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *The Fourth International IEEE Workshop on Wireless Local Networks (WLN 2004)*, pages 618–624, October 2004.

[6] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *Computer Communications Journal*, 28(10):1193–1203, June 2005.

[7] Khalil El-Khatib, Larry Korba, Ronggong Song, and George Yee. Secure dynamic distributed routing algorithm for ad hoc wireless networks. In *International Conference on Parallel Processing Workshops (ICPPW'03)*, pages 359–366, 2003.

[8] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 2003.

[9] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353, 1996.

[10] Dogan Kesdogan, Jan Egner, and Roland Bschkes. Stop-and-go-MIXes providing probabilistic anonymity in an open system. In *Second International Workshop on Information Hiding, Lecture Notes in Computer Science 1525*, pages 83–98, 1998.

[11] Jiejun Kong. *Anonymous and Untraceable Communications in MobileWireless Networks*. PhD thesis, University of California, Los Angeles, 2004.

[12] Jiejun Kong and Xiaoyan Hong. ANODR: ANonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03)*, pages 291–302, 2003.

[13] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 255–265, 2000.

[14] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *WMCSA'99*, 1999.

[15] A. Perrig, R. Canetti, D. Tygar, and D. Song. The TESLA broadcast authentication protocol. *RSA CryptoBytes*, 5(2):2–13, 2002.

[16] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-MIXes: Untraceable communication with very small bandwidth overhead. In *Proc. GI/ITG-Conference "Kommunikation in Verteilten Systemen" (Communication in Distributed Systems)*, pages 451–463, 1991.

[17] J.-F. Raymond. Traffic analysis: Protocols, attacks, design issues, and open problems. In *DIAU00, Lecture Notes in Computer Science 2009*, pages 10–29, 2000.

[18] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications, Special Issue on Copyright and Privacy Protection*, 16(4):482–494, 1998.

[19] Xiaoxin Wu and Bharat Bhargava. AO2P: Ad hoc on-demand position-based private routing protocol. *IEEE Transactions on Mobile Computing*, 4(4):335–348, 2005.

[20] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 1312–1321, 2003.

[21] Yanchao Zhang, Wei Liu, and Wenjing Lou. Anonymous communications in mobile ad hoc networks. In *IEEE INFOCOM 2005*, volume 3, pages 1940–1951, March 2005.

[22] Bo Zhu, Zhiguo Wan, Mohan S. Kankanhalli, Feng Bao, and Robert H. Deng. Anonymous secure routing in mobile ad-hoc networks. In *Proceedings of The 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004)*, pages 102–108, 2004.