A Hierarchical Anonymous Routing Scheme for Mobile Ad-Hoc Networks

Jun Liu*, Xiaoyan Hong*, Jiejun Kong†, Qunwei Zheng*, Ning Hu*, Phillip G. Bradford*
*Department of Computer Science
University of Alabama
Tuscaloosa, AL 35487
{jliu,hxy,qzheng,nhu,pgb}@cs.ua.edu
†Department of Computer Science
University of California
Los Angeles, CA 90095
jkong@cs.ucla.edu

***Abstract*— Privacy and anonymity are critical security issues to many large-scale MANET applications such as military communication networks. These applications are more likely deploying the networks heterogeneously and hierarchically due to administrative needs or routing efficiency. When the size of the network scales up, the routing overhead incurred by existing flat anonymous routing protocols increases fast as the required number of public key operations increases, thus resulting in deteriorated routing and data communication performance. In this paper, we introduce a novel hierarchical anonymous on-demand routing protocol tackling this limitation. In addition to guaranteeing routing and data delivering security, the scheme provides two levels of anonymity: intra-group and inter-group. By exploiting the hierarchical network structure, it effectively controls computational overhead while preserving anonymity, hence accommodates to larger-scale MANETs.**

I. Introduction

Instant communication support using mobile ad hoc networks (MANETs) in applications often demands that networks operate in a large scale. Examples of such applications include automated battlefield support, disaster relief, and vehicular networks, etc.. Such networks will be heterogeneous and hierarchically organized due to administrative needs or for routing efficiency. Many routing protocols have been proposed as scalable solutions for large-scale scenarios. These protocols use different mechanisms to achieve routing efficiency, including: clustering mechanisms (HSR [1] and Hi-TORA) [2]), geological information (GPSR [3]), dynamic addressing (DART [4]), grouped motion behavior (LANMAR [5]), proactive hierarchical routing (HOLSR [6]), or a hybrid of proactive and on-demand update strategies (ZRP [7], HARP [8] and SAFARI [9]).

Many of MANET applications also take network privacy and anonymity as a critical security requirement in order to protect the operation against the security vulnerability of wireless media. A number of anonymous MANET routing protocols have been proposed in recent years. These protocols include ANODR [10], ANONDSR [11], ASR [12], MASK [13] and SDAR [14]. They achieve anonymity goals such as identity anonymity and unlinkability in routing, as well as anonymous data delivery by using various security mechanisms. Typically these protocols use public key cryptography more or less in the route discovery phase. For resource-constrained mobile devices, the public-key operations could result in long route acquisition delay and degrade packet delivery ratio [15] [16]. When the size of the network scales up, the lengths of end-to-end paths grow accordingly on average. This will incur prohibitive computation and communication overhead along a long path. In addition, a long path tends to break more frequently in a mobile network, resulting in frequent maintenance and re-discovery processes. All these greatly deteriorate communication efficiency and network performance.

On the other hand, some networks like military communication networks feature hierarchical structures [6]. In civil applications the hierarchicalization of large-scale MANETs improves efficiency and scalability as well. In these cases, hierarchical anonymous routing would help both in adapting to the heterogeneous network constitution and to ensure the anonymous and hierarchical delivery of critical orders and reports.

Thus we are motivated to develop a novel Hierarchical ANonymous On-demand Routing protocol (HANOR). Our new protocol is based on a hierarchical MANET architecture with multi-hop clustering (called *group* in the paper). We intend to utilize the inherited group management with security features in order to tackle the limitations of flat schemes and achieve an efficient anonymous protocol suitable for hierarchical network architecture. The hierarchical structure allows us to separate anonymity protection for intra-group and inter-group communication. While the small scale intra-group anonymous routing uses flat anonymous protocol, the inter-group routing, instead, utilizes group key management to practically project groups into individual routing units. The HANOR allows the anonymous discovery of routes and sends data with dramatically reduced cryptographic computation overhead compared with pure flat routing.

The contribution of HANOR is three-fold: first, it's designed to take hierarchical MANET structures into account. For example, in a large-scale MANET consisting of groups formed due to application or administration requirements, each group has a subset of nodes, such as nomadic command posts in the battle field. These nodes are specifically in charge of communication with the outsides of the group. The HANOR operational premises satisfy such a scenario. During inter-group routing, HANOR practically considers groups as individual routing units, and achieves path anonymity at the upper level of groups. In the meantime, the protocol still achieves node anonymity and path anonymity at the lower node level through

exploiting group security premises. Second, HANOR greatly reduces computational overhead for routing. With inter-group routing, computational overhead for nodes interacting in the same group is minimized. As a result, the protocol is expected to require less computation overhead in route discovery. For low-end mobile communication devices, this effect directly translates into less route discovery latency. As in MANETs route discovery performance and route durability are largely affected by mobility, shorter route discovery latency results in a higher data delivery ratio. Third, while maintaining node and path anonymity, HANOR conducts group authentication during route discovery, effectively reinforcing the security on the group-level. To this extent, HANOR also bears the potential capability of group-level access control. To summarize, by exploiting the hierarchical network structure, HANOR effectively controls computational overhead while preserving anonymity and providing additional security, hence accommodates to larger-scale MANETs.

The rest of the paper is organized as follows. Section II presents a brief summary of the flat anonymous routing protocols, the measurements of cryptographic overhead and other related work to motivate our work. Section III describes the network model of this work. Section IV introduces the protocol in detail and Section V presents analysis on anonymity properties. Section VI shows our simulation results. Finally, Section VII concludes the paper.

II. Related Work and Motivation

A number of anonymous routing protocols have been proposed such as ANODR [10], ANONDSR [11], ASR [12], MASK [13] and SDAR [14]. They are all on-demand protocols but use different approaches for anonymous routing. ANODR and ASR use a boomerang type *onion*, a layered cryptographic structure on which appending and peeling off are performed by the same forwarding nodes. ANONDSR and SDAR use a *suggestion box* cryptographic structure, i.e, each node appends a cryptographic layer, and the destination peels off all the layers and reconstructs a new *onion* for return path. MASK and SDAR use periodic *hello* messages to establish pairwise trust relationship between neighbors. MASK then uses the trust and pseudonyms for route discovery.

Cryptographic tools are important in order to achieve security and privacy in data communications. In these protocols, public key cryptography is used at different stages in routing operations. Usually, public key cryptography uses more CPU time than symmetric key cryptography. For resource-constraint mobile devices, the computation time could be very long. Some measurements on Intel StrongARM 200MHz CPU based Pocket PC running Linux are presented in [17]. Based on the measurements, if a payload is of 512 Bytes, ECC uses 1209ms/637ms for encryption/decryption respectively, while AES uses 140μs for encryption/decryption. During the route discovery, ANODR and ASR perform asymmetric encryption/decryption primarily in RREP forwarding stage at each hop. ANONDSR and SDAR, instead, perform asymmetric encryption/decryption in RREQ flooding stage at each hop. In addition, ANONDSR and SDAR perform both public key and symmetric key operations at the destination nodes. Assume the path length of a discovered route is L, the computational overhead for discovering the route is $OH_{asymmetric} \cdot L$ for ANODR and ASR, and $2 \cdot OH_{asymmetric} \cdot L$ for SDAR and ANONDSR, where $OH_{asymmetric}$ is the computation latency of using public key cryptography. When message size is taking into consideration, the overhead will increase if a massage needs to be processed in several blocks. We then draw our attention to the usage of the public key cryptography when evaluating existing routing protocols and designing new protocols.

Apparently when the network scales up to a certain extent, the flat anonymous routing schemes will incur very long route acquisition latency. In a mobile network, such initial latency in data communication will result in low data delivery ratio, since a discovered path may have broken at the time data is transferred.

In [18], a location privacy framework for wireless networks with infrastructure is proposed which bears the flavor of a hierarchical scheme. For achieving unlinkable communication an anonymous bulletin board is introduced as a means of rendezvous. This approach requires the nodes in the network to check the bulletin board periodically to see if there are call-back requests from potential communication counterparts. In case of a multi-hop network attaching to a base station, an aforementioned anonymous routing is suggested. In the framework, the infrastructure is used as the upper level but no ad hoc routing is needed. This differs from HANOR in which a fully mobile ad-hoc network is targeted.

III. System Model

A. Network Scenarios

The hierarchical mobile ad-hoc network scenario we base this research on has two logical tiers. The lower tier is a network of multi-hop clusters and the high lever is a network of cluster headers (referred as groups and group leaders in the rest of the papers). Such network architecture can be pre-configured by network administrators or fully self-configured. When high-bandwidth backbone networks are possible, gateways in each group will interconnect group leaders. When no physical hierarchy exists, we assume a multi-hop clustering algorithm to form groups and elect leaders. Communication between two group leaders (a virtual link) needs to be relayed by other wireless nodes. Obviously, when groups can be pre-configured and/or physical support is feasible, we expect better performance. So at times, we will include such discussions.

In HANOR, we assume a distributed certificate authority(CA) infrastructure. The CA is responsible for assigning (and thus possessing) the public keys and private keys of all nodes before they join the network. For each group(elected or pre-configured), a pair of asymmetric keys, denoted as (PK_g, SK_g) are assigned. The group ID is derived from PK_g by the group leader, and distributed to the group members securely. The way the group ID is generated ensures that the group's public key is kept secret from group members. For data communication, we assume that each source-destination pair shares a global trapdoor, as been

widely used in existing anonymous routing protocols such as SDAR, ANODR, ASR and ANONDSR.

A node joining the network is preloaded with routing parameters. They include its *ID*, a pool of public key/private key pairs PK_{Tn}/SK_{Tn}, CA's public key PK_{CA}, and the election algorithm with parameters if needed. A node will use more keys and a couple of one-way hash functions H_1 and H_2 in routing. For security, hash functions will be reconfigured after elections or periodically so to control the aftermath of possible node intrusion.

B. Adversary and attack model

Adversaries can be categorized according to their behaviors: passive eavesdroppers and active attackers; or according to their knowledge about the network: external attackers and intruders; or according to their communication ability: individual or collaborative attackers. The HANOR protocol is mainly designed to deal with passive attacks, their goals are to get privacy information without disrupting routing operation. The adversaries could simply eavesdrop, or act protocol-compliantly when they are intruders. But we assume adversary's computational power and capabilities of node intrusion are limited. Multiple attackers can communicate to integrate their knowledge about the network. However, we don't assume a global adversary who is able to monitor all of the wireless transmissions. Such an attack could be either impractical to launch or be very expensive when network is large.

IV. Hierarchical Anonymous Routing Protocol

A. The Scheme Overview

HANOR accomplishes the following anonymous goals:

1) Establishing a path anonymously. This achieves anonymous goal in the route discovery process.
2) Transmitting data anonymously. This accomplishes anonymous goal in data forwarding process.

Anonymous route discovery of HANOR is conducted in a hierarchical way, consisting of intra-group anonymous routing and inter-group anonymous routing. The intra-group anonymous routing includes two phases: (1) route discovery within the source group, where the source node tries to establish an anonymous route towards the group leader, and (2) route discovery within the destination group, where the destination group leader establishes an anonymous route towards the destination. The inter-group anonymous routing phase will establish an anonymous route from the source group leader to the destination group leader. Thus, in a typical scenario where the source and destination reside in different groups, the routing process follows the following three consecutive *phases*: in the source group, between groups and in the destination group.

We adopt ANODR [10] for intra-group anonymous routing. A few modifications to ANODR protocol are needed so it can be integrated with the inter-group protocol. These modifications are presented throughout the following subsections. On the other hand, route discovery in the source group could wait long before completion due to the fact that the RREQ and RREP procedures are separated by the inter-group routing and intra destination group routing. This could result in negative influence on the successfulness of the route discovery. The problem can be solved in several ways. We will discuss these alternatives in the discussion subsection.

In designing the inter-group routing, we intend to treat each intermediate group as a single anonymous routing unit. Such design enables us to retain the cryptographic operation at the group level, which greatly reduces the end-end route acquisition delay. The inter-group routing will establish an one-way relation between groups and keep the cryptographic operation inside the group efficient.

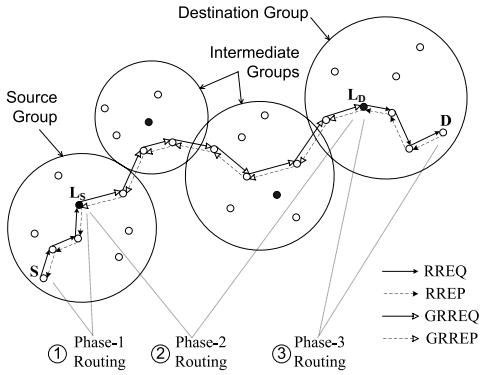

Fig. 1. HANOR Route Discovery

Figure 1 illustrates the process of the route discovery for a cross group path. A route is discovered from the source node *S* to the destination *D*. L_S and L_D are the group leaders of the source and destination groups respectively. The routing process consists of three phases. When *S* wants to discover a route to *D*, it constructs a route request(RREQ) message and sends it to L_S using local in-group anonymous routing algorithm (adapted ANODR is used for this purpose). According to the RREQ, L_S assembles an inter-group route request message(GRREQ) and send it to all other group leaders in the network. Inter-group routing scheme is used in this stage. Each group leader receiving GRREQ messages tries to find whether the destination is one of its members. It again uses flat anonymous routing algorithm(adapted ANODR) to establish a route to the real destination *D*, which sends back a route reply(RREP) message to L_D. L_D continues to reply with a GRREP message to L_S, which after receiving GRREP sends RREP to the original source node *S*. If the original path between *S* and L_S has been broken due to node mobility, L_S can initiate a reverse route request trying to proactively find a route from itself to *S*. After the sub-route between *S* and L_S is discovered, an anonymous route has been established from *S* to *D*.

The rest of the section presents the protocol in detail. *anonymous route request* and *anonymous route reply* subsections describe the above steps of route discovery first, followed by *route maintenance* and *anonymous data forwarding*. Discussions are given when necessary.

B. Anonymous Route Request

1) Anonymous Route Request in the source group: The anonymous route request starts with intra-group routing in the source group. We utilize ANODR to establish an anonymous route from the source node to the source group leader (L_S). The original ANODR RREQ message is modified to include two functions: RREQ flood control and informing L_S the destination trapdoor. In addition, considering the fact that L_S will be used by its group members when they initiate a communication, we avoid any direct use of L_S's trapdoor so to prevent the content correlation attack. The modified RREQ message looks like:

$$< RREQ, seq_1, pk_{one}, TBO, H_1^{n_r}(GID), (Src, tr_{dest}, TK)_{PK_{L_S}} >$$

where RREQ is a routing control message flag identifying route request, seq_1 is the sequence number for this route request session, pk_{one} is a one-time use public key to be used in RREP for ANODR to achieve unlinkability, and TBO (padded to a fixed-length) is the onion structure.

The field $H_1^{n_r}(GID)$ is used to control the RREQ flooding to be within the group (here, the source group). H_1 is a parameterized one-way hash function for each specific group and it is updated after each election process or periodically. Thus, before forwarding a RREQ, each node (including the leader) chooses a random number n_r (bounded by a maximum value) and applies H_1 on its group ID *GID* for n_r times. Upon receiving an unseen RREQ message (a new seq_1), a node applies H_1 a threshold number of times on *GID* and compares the results with the fifth field of the received RREQ, i.e., $H_1^{n_r}(GID)$. If there is a match, the RREQ message is from a node of the same group, and it will be forwarded with an updated n_1. Otherwise, the RREQ is discarded. Clearly, no real group IDs will be revealed in the route request messages and the flooding is controlled. The trade-off is the computation time for one-way hash function, which can be ignored compared to public cryptosystems.

The last field is encrypted by the public key PK_{L_S} of the L_S. It serves as a trapdoor of the L_S, since it is the only node that is going to and is able to decrypt it. And it also prevents correlations among multiple RREQs sending to the same L_S. The encrypted form also protects the source tag *Src*, the trapdoor for the destination tr_{dest}, and an one-time key *TK* to be used in the RREP procedure. After all, the leader of the source group will receive the RREQ message.

2) Inter-Group Anonymous Route Request: The source group leader L_S initiates the inter-group routing phase by sending an inter-group route request message(GRREQ) to all other group leaders in the network. Each group leader receiving the GRREQ message tries to find the destination in its group. Thus, after L_S receives the RREQ message, it stores seq_1, *TK* and *Src*, picks up a new sequence number seq_2, and assembles and floods a new inter-group GRREQ message using the tr_{dest}. The seq_2 will uniquely identify this inter-group route discovery and it is recorded with the tuple *<seq1, seq2, TK, Src>*. The following gives the format of GRREQ.

$$< GRREQ, seq_2, PK_T, H_1^{n_r}(GID_c), (seq_2, tr_{dest})_{SK_{G_S}} >$$

The propagation of GRREQ messages is a controlled flooding by $H_1^{n_r}(GID_c)$, similar to the previous RREQ flooding control, together with the sequence number, i.e., only nodes within the group who receive a GRREQ with a new seq_2 will rebroadcast it. The last field is used for carrying the destination trapdoor tr_{dest} and for authenticating the initiator of the GRREQ. It is encrypted by the source group's private key SK_{G_S} and can only be decrypted by the group leaders, so to verify the validity of tr_{dest}, and then to issue a search for the destination within the group. Since each group's public keys are kept secret from group members, non-leader nodes can not recover tr_{dest}.

The two fields PK_T and $H_1^{n_r}(GID_c)$ are used to implement a hierarchical link security scheme from a per-hop approach on top of the group architecture. PK_T is a one-time use public key replaced by each intermediate node. $H_1^{n_r}(GID_c)$ is used for intermediate nodes to judge whether the received GRREQ message is from a node in the same group. Where, GID_c is the group id of the current node forwarding the GRREQ, and H_1 is the one-way hash function for the current group. GID_c is hashed by H_1 for a random n_r times (bounded by a maximum value).

In this implementation, each intermediate node needs to observe and distinguish the following two situations upon receiving a GRREQ message:

- The previous hop is a node from the same group as itself.
- The previous hop is a node from a different group.

As analyzed before, the field of $H_1^{n_r}(GID_c)$ in GRREQ enables this identification. Accordingly, the intermediate node records (seq_2, n_r) in a table called *S-Table*, if the GRREQ message is from a node in the same group; otherwise, it records (seq_2, PK_T) in a table named *P-Table*. For both cases, the node then generates new n_r and PK_T to replace previous PK_T and $H_1^{n_r}(GID_c)$ in the GRREQ, and rebroadcasts the message. The *S-Table* and *P-Table* tables are used when/if the GRREP is returned. The advantages of using the two fields and the two tables will be discussed in the *discussion* paragraph later.

In all, when processing a GRREQ, an ordinary node computes only efficient hash operations while a group leader performs additional cryptographic operations to decode the destination. This results in significant computation overhead reduction. On the other hand, every node in the network receives and forwards a copy of each GRREQ. To reduce this routing overhead, many flooding suppression schemes [19] [20] [21] can be used. And if the network has high-bandwidth links supporting interconnection among groups, the overhead of propagating GRREQ messages in a flooding manner can be removed by taking advantage of the physical capability in that broadcasting GRREQ is only over the high-bandwidth links.

3) Anonymous Route Request for the destination: The encrypted form of destination trapdoor tr_{dest} in a GRREQ prevents the destination from knowing that it is being searched. Thus the destination group leader has to conduct another intra-group route discovery. In fact, since the group leaders do not know whether or not the destination is in its group, all the group leaders will initiate a route discovery within the group. This feature increases

the routing overhead. But on the other hand, it strengthens the anonymity protection.

The group leaders use ANODR to look for the destination, and if found, to establish an anonymous route to it. According to the modified RREQ message format, a leader constructs the following message and initiates a search within the group.

$$< RREQ, seq_3, pk_{one}, TBO, H_1^{n_r}(GID_c), (tr_{dest}, PAD)_{SK_{L_D}} >$$

The RREQ message uses a new sequence number seq_3 for this routing phase. The $H_1^{n_r}(GID_c)$ is used for RREQ flood control as before. The destination trapdoor tr_{dest} is signed by the private key of the leader SK_{L_D}. PAD is a random string for making this phase-3 RREQ message the same length with that of phase-1 RREQ. Thus, by simply eavesdropping, an attacker is not able to distinguish RREQs in different phases, nor is an legitimate node. But being legitimate, an ordinary node will decrypt the last field of a new received RREQ using its leader's public key to check if it is the intended destination. If yes, the node initiates the route reply procedure as described in the next subsection. Otherwise, it does nothing. A group leader receiving a RREQ that is not initiated by itself will decrypt the last field using its private key, for the message can be a phase-1 RREQ. In addition, all the nodes participate in the control flooding of RREQ within the group.

C. Anonymous Route Reply

1) Anonymous Route Reply in the destination group: After the destination successfully verifies the trapdoor, it initiates route reply with a proof pr_{dest} for the successful opening on the destination trapdoor. Since the destination node does not know in which group the source node resides, not to mention the source node's identity information, the first step of RREP is targeted at the destination's group leader L_D. ANODR's RREP message is modified to carry the necessary information for L_D (so is encrypted by L_D's public key PK_{L_D}) to further forward the reply. The RREP procedure of ANODR completes the establishing of an anonymous route between the destination and its leader L_D. As in standard ANODR, the symmetric encryption by a randomly chosen symmetric key K_{seed} and the public key encryption of K_{seed} by pk_{one} ensures untraceability. The added information by HANOR does not weaken the protocol.

$$< RREP, (K_{seed})_{pk_{one}}, ((pr_{dest}, seq_3, K_1)_{PK_{L_D}}, TBO)_{K_{seed}} >$$

2) Inter-group Anonymous Route Reply: After receiving the RREP, the destination group leader L_D recovers the sequence number seq_3 and pr_{dest}. It remembers the key K_1 for later data transfer within the group. K_1 is used as session key to encrypt the data payload. L_D then sends the second phase GRREP toward the source group:

$$< GRREP, (((pr_{dest}, seq_2, K_2)_{SK_{G_D}})_{PK_{G_S}}, K_n, KEY)_{E_{KEY}} >$$

For security purpose, pr_{dest}, seq_2, along with a session key K_2 (used for ene-to-end encryption during data transfer between L_S and L_D) are encrypted by the destination group's secret key SK_{G_D} and the source group's public key PK_{G_S}. These fields are only understandable to the source leader. The message also builds a per hop symmetric link key K_n for data transmission.

All these information are encrypted by the key *KEY* using the encryption method E_{KEY}. *KEY* and E_{KEY} are interpreted differently depending on the next hop R on the GRREP path. Specifically, (1) if node R is in the same group, $KEY = KEY_s = H_2^{n_r}(GID_c)$, and E_{KEY} refers to a symmetric encryption using KEY. Here GID_c is the current group ID, H_2 is new group specific hash function, and n_r is retrieved from table *S-Table*. The result of hashing n_r times with function H_2 is used as a symmetric key in the most outer encryption of GRREP. (2) If R is in a different group, $KEY = KEY_p = PK_T$ and E_{KEY} refers to an asymmetric encryption using public key KEY. Here PK_T is retrieved from the *P-Table*. Then the most outer encryption in GRREP is a public-key encryption. Both encryption methods can only be decrypted correctly by the next hop node R. R records K_n as a *VCI*(virtual circuit identifier) for data transmission. The advantages of the mechanism are that the relation between the upstream and downstream nodes is not revealed to any nodes, and only a few nodes along group borders need to perform asymmetric cryptographical operations.

In order to understand a received GRREP correctly, an intermediate node will first try to decode it using KEY_s. If failed, i.e, it can not match the KEY_s from the decrypted text, it tries to decrypt using the private key SK_T that matches PK_T. If again failed, the node is not on the path and the GRREP is dropped. If one of the decoding is successful, the intermediate node replaces a new K_n, encrypts the whole message using a appropriate KEY according to the aforementioned rules, and broadcasts it locally.

This process repeats until the source group leader receives the GRREP. The route established is anonymous and untraceable with reduced computation overhead.

3) Anonymous Route Reply in the source group: At the source group leader L_S, after recovering $< pr_{dest}, seq_2, K_2 >$ from GRREP, it finishes anonymous route discovery in the source group by initiating a RREP. Recall that L_S has stored the tuple $< seq1, seq2, TK, Src >$, it is able to generate a RREP in the following format. It generates a new session key K_3 for data transmission between the source and itself. It then encrypts the needed information using the key TK to authenticate itself to the source, and further encrypts the information using its private key SK_{L_S} so to authenticate itself to all the intermediate nodes. At the meantime, it remembers the inter-group session key K_2. The propagation of the RREP follows ANODR protocol. The ANODR protocol ensures that the route established is anonymous and untraceable. The added information by HANOR does not weaken the original protection.

$$< RREP, (K_{seed})_{pk_{one}}, (((pr_{dest}, Src, K_3)_{TK})_{SK_{L_S}}, TBO)_{K_{seed}} >$$

D. Discussions

1) Route Discovery in the Source Group: In a large-scale network, a HANOR route may be very long. The route discovery process thus may experience prolonged acquisition time. It is

possible that when GRREP message arrives at L_S, some of the nodes which are originally on the path from the source to L_S have moved away. Sending a RREP following such a reverse path is doomed to fail, resulting in all the previous steps wasted. A possible solution works as follows. Upon receiving a RREQ from one of its member, the leader L_S immediately respond a RREP as an acknowledge of the request. Then after receiving the GRREP, L_S will initiate a separate ANODR route discovery to establish a route to the previous source. When the previous source replies to L_S's RREQ, it can start sending the first data packet with the RREP.

2) Overhead Trade-off: During the route discovery, each of the nodes in the network receives and forwards a copy of each GRREQ and one copy of RREQ for the destination. This routing overhead is twice the routing overhead generated by a flat anonymous routing protocol like ANODR. When broadcast suppression schemes like passive clustering, or dominant set are used, the overhead remains twice by HANOR. However, understanding that the flooding of GRREQ is merely for establishing anonymous virtual links among the group leaders, we could exploit the possible existence of a physical higher tier network in significantly reducing routing overhead. In many envisioned applications, such physical supports are feasible. The overhead of HANOR, then, will reduce to half since the GRREQ messages can be propagated in the intergroup backbone. This makes the routing overhead of both flat and hierarchical schemes at the same level.

On the other hand, HANOR greatly reduces the sizes of routing packets. Most existing flat schemes use *onion*. In order to hide the path length, it must be padded to a maximum size. This results in large control packets. Broadcasting a large control packet increases channel contention, which could result in long queueing delay or packet loss. In HANOR, the *onion* is padded only up to the size of a group - a much smaller size than it is in a flat scheme.

In addition, HANOR reduces overall computational overhead. In HANOR expensive public key cryptography is only needed at the border nodes of the groups, rather than at each node if using a flat scheme like SDAR or ANONDSR. An great advantage of HANOR then is the reduced end-to-end route acquisition latency, and this leads to improved data delivery ratio.

E. Anonymous Data Forwarding

The design of HANOR is to achieve both data confidentiality and data privacy in data transmission. The former requires an end-to-end encryption while the latter needs per hop treatment to prevent content correlation. Given the design, HANOR is able to prevent a set of colluding attackers from tracing a data forwarding path.

Like the route discovery, the data transmission of HANOR is a three-phase process. Along the forwarding path, HANOR employs a two-tier data variation procedure. The first tier is a three-phase content variation and a three-phase end-to-end encryption. After route discovery, secret sharing is established between the source and L_S, between L_S and L_D, and between L_D and destination node. They are symmetric keys of K_3, K_2 and K_1 for these phases respectively. By re-encrypting data per phase using the phase-specific secret keys, content correlation during data forwarding is prevented on the level of phases, that is, without compromising source/destination or their group leader nodes, attackers are unable to correlate data traffic across phases. Compared with brute force solutions which use a direct symmetric key between the destination and the source (can be established during the route discovery), this three-phase encryption reduces the risk of being traced through content correlation in the presence of intruded attackers.

The phase-wise content correlation protection has to be further protected inside each phase. In HANOR, we use a virtual circuit behavior for per hop data forwarding, which is widely used in protocols such as ANODR, ASR and MASK. For the intra-group phases, such behavior is guaranteed automatically through ANODR. For the inter-group sub-route, per hop symmetric link keys (K_n) have been setup during the GRREP propagation for this purpose. Thus at each hop, the phase-session key protected data payload will be encrypted again using the per hop key. Content correlation is impossible throughout the forwarding path.

F. Route Maintenance

An established route breaks due to many reasons. For our scenario, a new reason could be the change of a group leader or a group membership. However, in HANOR, when a route breaks, it is not always necessary to re-initiate the route discovery from scratch. Given the separated three routing phases, the rebuild of a broken route can be limited only within the associated phase. The routing initiator of each phase, i.e., the source, the L_S or the L_D can choose to immediately re-initiate a route request for its sub-route when a route breakage in its phase is detected. This is another advantage of HANOR compared to a flat scheme.

V. Anonymity Analysis

Our analysis first concerns the two aspects of the anonymity concept, namely, the identity anonymity, and the unlinkability of the senders and the receivers. The protocol doesn't reveal nodes' identities (including leaders), nor pseudonyms, nor temporal group IDs in route discovery and data transmission. Individual intruders don't obtain additional information about the network except that pertained to the node itself. Cooperative intruders have their best chance if they happen to be on the same routing segment so to break the phase session key and correlate the data packets. In order to trace to the source or the destination, they have to compromise every consecutive nodes which is very difficult to do when the network is large.

But there are two major concerns regrading to the hierarchy of the network: (1) whether the protocol reveals group structures if they do bear logical organization information that need to be hidden? and (2) whether the hierarchical routing increases the chance for the adversary to trace the routing path? We discuss them below.

A. Group Anonymity

Protecting group structures faces a dilemma: on one hand, in order to preserve advantages of a hierarchical scheme, the intra-group routing should be confined within the area of the group; on the other hand, such action would reveal the group structure. Using HANOR, while each node performs hash function secretly to control RREQ flooding, the sequence number seq reveals the boundary of a group if the adversary can monitor an extended area. Node mobility and group re-election can alleviate this problem since unstable group structure is less meaningful. However, as long as the newly elected group consists of most of the members from the old one, the attackers could still trace the group in a probability based on group venue correlation. Note that the adversary has to be densely distributed and collaborative, which makes the cost of accomplishing such an attack very high. Having some nodes compromised and some GIDs revealed does not result in immediate threat to a group structure if such compromise does not become an extended monitoring.

B. Group Unlinkability

During route discovery, the unlinkability problem has three sub-problems: unlinkability of route request messages at different routing phases, unlinkability of route reply messages at different phases, and unlinkability of route request messages with route reply messages. Recall we assume adversary has no global traffic monitoring ability and the timing analysis can be treated using well adopted MIXing technology such as [22].

When only external attackers exist, group correlation through route request messages is impossible since the only information indicating a connection is $trDest$, which is encrypted. The route reply messages are encrypted with different keys at each hop, leaving no clue to an eavesdropper to correlate two replies or to a previous seen request.

With the existence of compromised nodes, if a normal node is compromised, it can only get *trDest*. Collaborating with other intruders will not generate more useful information. Compromised group leaders cause more problems as they know each other's group public keys. If a source group leader is compromised, it knows the relation between Src and *trDest* immediately. When collaborating with a compromised node in the destination group, the attacker reveals the relation between the source and destination groups. If a destination group leader is compromised, the situation is similar. However, in a large scale network, the adversary has to compromise a large portion of nodes in order to get such correlation.

VI. Simulation Results

We evaluate through simulation the advantage of controlled computational overhead achieved by HANOR. The evaluation metrics include: *(i)Number of public key operations en-route*: Only public key operations performed by nodes en-route are counted; *(ii)Number of public key operations network-wide*: public key operations performed by nodes in the entire network, including those performed by nodes not in route but **tried** to decrypt the overheard messages; *(iii)Number of path hops*: the average number of hops of routes discovered.

We use a custom-built simulator to investigate the impact of network size on the aforementioned metrics for two anonymous routing protocols: our proposed HANOR protocol and ANODR, the protocol for flat anonymous routing. The network area in the simulation is square, in which nodes are deployed randomly. The transmitting range of nodes is configured to about 370 meters conforming to the default value of the Qualnet [23] simulator. For different network sizes(i.e., number of nodes in the network), we keep the same node density such that each node has approximately 20 neighbors in its transmitting range. For example, with the network size of 2000, the network area is about 6550x6550. The impact of mobility is not considered in our simulation, for our purpose is primarily focused on showing the impact of network size on the two protocols, and it's expected that with mobility HANOR will perform even better due to its low route discovery latency. For HANOR, we define the groups statically for simplicity by partitioning the network into grids each of which represents a group. The length of the grid edge is 4 times of transmission range. We randomly generate 500 source-destination pairs and average the results for each data point evaluated.

Figure 2 shows the impact of network size on the number of public key operations required for nodes en-route in route discovery. The figure shows that when the network size is less than 3000, the overhead en-route for HANOR is more than that for ANODR. This is because the number of public key operations HANOR performs is determined by the two sub-path length and the number of groups between the source and destination, while for ANODR it is largely determined by the actual distance between the source and destination. When the network size is small, for HANOR the total sub-path length in the two source/destination groups can exceed the path length by ANODR. When network size grows, the path length of ANODR grows accordingly, but for HANOR the total number of hops in source/destination groups remains the same, while at the same time the number of public key operations performed during inter-group routing is greatly reduced compared with ANODR.

Figure 3 reports the impact of network size on the number of public key operations required for nodes in the whole network in route discovery. It can be seen that the number of public key operations ANODR performs is always more that that performed by HANOR. This is because when a RREP message is broadcasted by ANODR, all nodes who overhear it will try to decrypt the message. However, in HANOR, when a GRREP message is broadcasted, only nodes in groups different from that of the local sender will have to try to decrypt the message using public key operations. The nodes in the same group of the local sender only need to perform efficient hash functions. The overhead reduction of HANOR becomes more obvious when the network size increases as the path length of the inter group communication increases.

Figure 4 gives the average number of hops for HANOR and ANODR. It confirms that the number of node-to-node hops of HANOR is only larger than that of ANODR at a constant basis. The additional number of hops are resulted from intra-group routing in source and destination groups which makes the overall

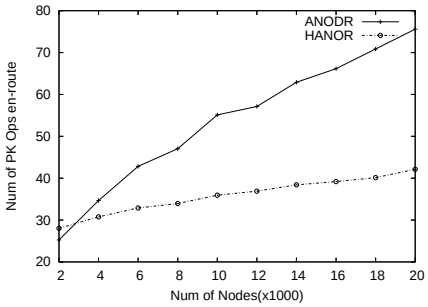

Fig. 2. Pub Key Operations En-route

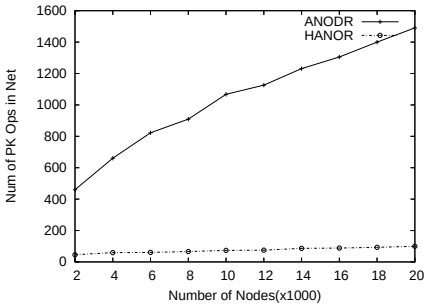

Fig. 3. Pub Key Operations Network-wide

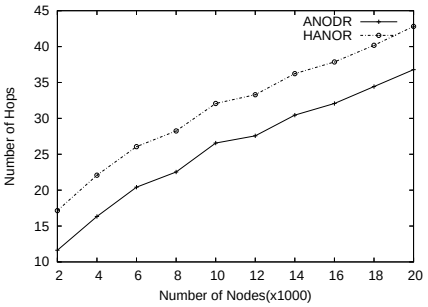

Fig. 4. Number of Hops

path not a shortest-alike path like that of ANODR. When the network size increases, the additional hops by HANOR become less significant compared with ANODR.

In summary, inter-group routing of HANOR increases the routing efficiency by reducing public key cryptography operations. The performance of HANOR is being further investigated as an on-going work.

VII. Conclusion

This paper presents a hierarchical anonymous routing protocol HANOR for mobile ad hoc networks. HANOR uses two levels of anonymous routing: intra-group anonymous routing and inter-group anonymous routing. The main advantage of HANOR is that it effectively controls computational overhead using the hierarchical routing scheme and preserves routing anonymity. Our simulations show a much slower increasing rate of public key cryptograph operations compared to a flat scheme. Our future work includes more theoretical analysis on anonymity and routing overhead, extensive evaluation on communication performance and trade-offs under various network conditions.

References

[1] G. Pei and M. Gerla, "Mobility management for hierarchical wireless networks," *Mob. Netw. Appl.*, vol. 6, no. 4, pp. 331–337, 2001.

[2] T. Ohta, M. Fujimoto, S. Inoue, and Y. Kakuda, "Hi-tora: a hierarchical routing protocol in ad hoc networks," in *p. 143, 7th IEEE International Symposium on High Assurance Systems Engineering (HASE)*, 2002.

[3] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 243–254.

[4] J. Eriksson, M. Faloutsos, and S. Krishnamurthy, "Scalable ad hoc routing: The case for dynamic addressing," in *proceedings of IEEE INFOCOM*, 2004.

[5] G. Pei, M. Gerla, and X. Hong, "Lanmar: landmark routing for large scale wireless ad hoc networks with group mobility," in *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. IEEE Press, 2000, pp. 11–18.

[6] Y. Ge, L. Lamont, and L. Villasenor, "Improving scalability of heterogeneous wireless networks with hierarchical olsr," in *The OLSR Interop & Workshop*, 2004.

[7] Z. J. Hass, "A new routing protocol for the reconfigurable wireless networks," in *Proceedings, IEEE 6th International Conference on Universal Personal Communications*, 1997, pp. 562–566.

[8] N. Nikaein, C. Bonnet, and N. Nikaein, "Harp - hybrid ad hoc routing protocol," in *proceeding of IST '01: International Symposium on Telecommunications*, 2001.

[9] S. Du, A. Khan, S. PalChaudhuri, A. Post, A. K. Saha, P. Druschel, D. B. Johnson, and R. Riedi, "Self-organizing hierarchical routing for scalable ad hoc networking," in *Technical Report TR04-433, Department of Computer Science, Rice University, Houston, TX, USA*, 2004.

[10] J. Kong and X. Hong, "ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks," in *ACM MOBIHOC'03*, 2003, pp. 291–302.

[11] R. Song, L. Korba, and G. Yee, "AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks," in *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, 2005.

[12] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. H. Deng, "Anonymous Secure Routing in Mobile Ad-Hoc Networks," in *29th IEEE International Conference on Local Computer Networks (LCN'04)*, 2004, pp. 102–108.

[13] Y. Zhang, W. Liu, and W. Lou, "Anonymous Communications in Mobile Ad Hoc Networks," in *IEEE INFOCOM*, 2005.

[14] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks," in *29th IEEE International Conference on Local Computer Networks (LCN'04)*, 2004, pp. 618–624.

[15] J. Liu, J. Kong, X. Hong, and M. Gerla, "Performance evaluation of anonymous routing protocols in manets," in *IEEE Wireless Communications and Networking Conference(WCNC)*, 2006.

[16] J. Kong, J. Liu, X. Hong, and M. Gerla, "Toward efficient solutions to resist mobile traffic sensors: How much performance cost is paid by on-demand anonymous routing protocols," in *International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN)*, 2006.

[17] V. Gupta, S. Gupta, S. Chang, and D. Stebila, "Performance analysis of elliptic curve cryptography for SSL," in *Proc. ACM Workshop on Wireless Security 2002*. Atlanta, GA, USA: ACM Press, 2002, pp. 87–94.

[18] Y.-C. Hu and H. J. Wang, "A Framework for Location Privacy in Wireless Networks," in *ACM SIGCOMM Asia Workshop*, 2005.

[19] Y. C. Y. Tseng, S. Ni and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of ACM MOBICOM'99*, 1999.

[20] W. Luo and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," in *IEEE Transactions on Mobile Computing, 1 (2): 111-122*, 2002.

[21] S. P. R. Ganhdi and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proceedings of ACM MobiHoc*, 2003.

[22] S. Jiang, N. H. Vaidya, and W. Zhao, "A mix route algorithm for mix-net in wireless ad hoc networks," in *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[23] Scalable Network Technologies (SNT), "QualNet," http://www.qualnet.com/.