Authenticated Anonymous Secure Routing for MANETs in Adversarial Environments

Madhu T S
M.Tech, CNE student
T.John Institute Of Technology
Banglore, India

Salonee Mishra
Assistant Professor, Dept. of CSE
T.John Institute Of Technology
Banglore, India

***Abstract* -Anonymous communications are important for many applications of the mobile ad hoc networks deployed in adversary environments .The major requirement on the network is to provide unlink ability and unidentifiability for mobile nodes and their traffics. Although a more number of anonymous secure routing protocols have been proposed, those requirement is not fully satisfied. The present protocols are vulnerable to the attacks of fake routing packets, even the node identities are protected by pseudonyms. In this paper, a new routing protocol, i.e. AASR, to satisfy the requirement and defend the attacks. Specifically, the route request packets are authenticated by a group signature, is to defend the potential active attacks without unveiling the node identities. Key-encrypted onion routing with a route secret verification message, is designed to prevent intermediate nodes from inferring a real destination. Simulation results have demonstrated the effectiveness of the proposed AASR protocol with improved performance as compared to the existing protocol.**

Keywords: Anonymous Routing, Authenticated Routing, Onion Routing, Mobile Ad hoc Networks

I. INTRODUCTION

Mobile ad hoc networks (MANETs) are vulnerable to security threats due to the inherent characteristics of such networks, such as the open wireless medium and dynamic topology. It is difficult to provide trusted and secure communications in adversarial environments, such as battlefields. On one hand, the adversaries outside a network may infer the information about the communicating nodes or traffic flows by passive traffic observation, even if the communications are encrypted. On the other hand, the nodes inside the network cannot be always trusted, since a valid node may be captured by enemies and becomes malicious. As a result, anonymous

may be released to the intermediate nodes in backward RREP forwarding. The other protocols rely on the neighborhood detection and authentication, but may partially violate the anonymity requirements for performance considerations. For example, in SDAR, the node and its one hop neighbors are made to know each other's ID during the routing procedures.

These protocols are also vulnerable to the denial-of-service (DoS) attacks, such as RREQ based broadcasting. Due to in adversarial environments, in which the nodes identifications and routes are replaced by random numbers or pseudonyms for protection purpose.

Anonymity is defined as the state of being unidentifiable within a set of subjects. In MANETs, the requirements of anonymous communications can be described as a combination of unindentifiability and unlinkability [1]. Unindentifiability means that the identities of the source and destination nodes cannot be revealed to other nodes. Unlinkability means that the route and traffic flows between the source and destination nodes cannot be recognized or the two nodes cannot be linked. The key to implementing the anonymous communications is to develop appropriate anonymous secure routing protocols. There are many anonymous routing protocols proposed in the past decade. Our focus is the type of topology-based on-demand anonymous routing protocols, which are general for MANETs in adversarial environments. To develop the anonymous protocols, a direct method is to anonymize the commonly used on-demand d hoc routing protocols, such as AODV [2] and DSR [3]. For this purpose, the anonymous security associations have to be established among the source, destination, and every intermediate node along a route. The resulting protocols include ANODR [4], [5], SDAR [6], AnonDSR [7], MASK [8], [9], and Discount-ANODR [10].

After examining these protocols, we find that the objectives of unindentifiability and unlinkability are not fully satisfied. For example, ANODR focuses on protecting the node or route identities during a route discovery process, especially on the routing packets, e.g., Route REQuest (RREQ) and Route REPly (RREP). ANODR adopts a global trapdoor message in RREQ, instead of using the ID of the destination node. However, the route can be identified by a disclosed trapdoor message, which the lack of packet authentication, it is difficult for the protocols to check whether a packet has been modified by a malicious node. Recently, group signature is introduced to anonymous routing. In A3RP [11], the routing and data packets are protected by a group signature. However, the anonymous route is calculated by a secure hash function, which is not as scalable as the encrypted onion mechanism..

II. BACKGROUND AND RELATED WORK

The basic concepts in anonymous routing, and provide a short survey on the existing routing protocols.

A. *Anonymity and Security Primitives*

We introduce some common mechanisms that are widely used in anonymous secure routing.

1) Trapdoor: In cryptographic functions, a trapdoor is a common concept that defines a one-way function between two sets [12]. A global trapdoor is an information collection mechanism in which intermediate nodes may add information elements, such as node IDs, into the trapdoor. Only certain nodes, such as the source and destination nodes can unlock and retrieve the elements using pre-established secret keys. The usage of trapdoor requires an anonymous end-to-end key agreement between the source and destination

2) Onion Routing: It is a mechanism to provide private communications over a public network [13]. The source node sets up the core of an onion with a specific route message. During a route request phase, each forwarding node adds an encrypted layer to the route request message. The source and destination nodes do not necessarily know the ID of a forwarding node. The destination node receives the onion and delivers it along the route back to the source. The intermediate node can verify its role by decrypting and deleting the outer layer of the onion. Eventually an anonymous route can be established.

3) Group Signature: Group signature scheme [14] can provide authentications without disturbing the anonymity. Every member in a group may have a pair of group public and private keys issued by the group trust authority (i.e., group manager). The member can generate its own signature by its own private key, and such signature can be verified by other members in the group without revealing the signer's identity. Only the group trust authority can trace the signer's identity and revoke the group keys.

III. NETWORK SCENARIO

This present the adversaries and attack models as well as the network assumptions and node model.

A. Adversaries and Attack Models

Without loss of generality, we assume that an adversary knows all the network protocols and functions. The attackers outside the network do not know the secret keys, but those inside the network may know the keys. We classify their attacks according to their behaviors (e.g., active or passive) and locations (e.g., inside or outside the network). Passive outside attack: There may be an external global passive adversary, who can observe and record all the wireless communications in the network. It will try to reveal the identities of the source, destination, and en-route nodes of a particular flow, or infer the traffic flows by linking the packets to the source or destination nodes.

Active outside attack: The passive attackers avoid any attack that reveals their actions since they attempt to be invisible, but the active outside attackers do not have such restrictions. They may aim to disrupt the routing or launch a DoS attack. They can move from here to there and launch attacks randomly. Passive inside attack: The attackers are legitimate MANET nodes. Similar to the passive outside attackers, they will try to infer the identities of the source, destination, or enroute nodes without exposing themselves. Since they can read the legitimate packets, the traffic pattern or node mobility information may be learned by them.

Active inside attack: They can modify, inject, and replay genuine messages. They can also masquerade as other nodes and launch the impersonation attacks. They can create one or more phantom nodes by generating valid routing packets

B. Network Assumptions

We denote a MANET by **T** and make the following assumptions.

1) Public Key Infrastructure: Each node **T** initially has a pair of public/private keys issued by a public key infrastructure (PKI) or other certificate authority (CA). For node *A* (*A* ∈ **T**),
its public/private keys are denoted by *KA*+ and *KA*. Similar to the existing secure routing [22], we assume that there exists a dynamic key management scheme in **T**, which enables the network to run without online PKI or CA services.

2) Group Signature: We consider the entire network **T** as a group and each node has a pair of group public/private keys issued by the group manager. The group public key, denoted by *GT*+, is the same for all the nodes in **T**, while the group private key, denoted by *GA* (for *A* ∈ **T**), is different for each node. Node *A* may sign a message with its private key *GA*, and this message can be decrypted via the public key *GT*+ by the other nodes in **T**, which keeps the anonymity of *A* [14]. We also assume that there exists a dynamic key management scheme working together with the admission control function of the network, which enables the group signature mechanism running properly. Such assumptions are also adopted in the existing work of military ad hoc networks [17], [23].

3) Neighborhood Symmetric Key: Any two nodes in a neighborhood can establish a security association and create a symmetric key with their public/private keys. This association can be triggered either by a periodical HELLO messages or by the routing discovery RREQ messages. For two nodes *A* and *B* (*A;B* ∈ **T**), the shared symmetric key is denoted by *KAB* and used for the data transmissions between them. There are some approaches supporting the establishment of one-hop shared key, such as MASK, RAODR, and USOR. In this work, we assume one of the approaches is available in **T**.

C. Node Model

1) Destination Table: We assume that a source node knows all its possible destination nodes. The destination information, including one of destination's pseudonym, public key, and
the pre-determined trapdoor string *dest* will be stored in the destination table. Once a session to the destination is established, the shared symmetric key is required for data encryptions in the session. Such symmetric key is generated by the source node before sending the route requests, and stored in the destination table after receiving the route reply. For example, a sample entry of the destination table is (*Dest Nym*, *Dest String*, *Dest Public Key*, *Session Key*).

2) Neighborhood Table: We assume that every node locally exchanges information with its neighbors. It can generate different pseudonyms to communicate with different neighbors.
The neighbors security associations are established as well as the shared symmetric keys. The information is stored in a neighborhood table. For example, a sample entry of the neighborhood table is (*Neighbor Nym*, *Session Key*).

3) Routing Table: When a node generates or forwards a route request, a new entry will be created in its routing table, which stores the request's pseudonym and the secret verification message in this route discovery. Such an entry will be marked in the status of "pending". If an RREP packet is received and verified, the corresponding entry in the routing table will be updated with the anonymous next hop and the status of "active". Meanwhile, a new entry will be created in the node's forwarding table. For example, a sample entry of the routing table is (*Req Nym*, *Dest Nym*, *V er Msg*, *Next hop Nym*, *Status*). Note that, to simplify the notation, we ignore the timestamp information of the entry in the table.

4) Forwarding Table: The forwarding table records the switching information of an established route. We adopt the per hop pseudonym as the identifier for packet switching, similar to the VCI (virtual channel identifier) in ATM networks. In each entry of the forwarding table, the route pseudonym is generated by the destination node, while the node pseudonyms of the previous and next hop are obtained after processing the related RREQ and RREP packets. For example, a sample entry of the forwarding table is (*Rt Nym*, *Prev hop Nym*, *Next hop Nym*)

III. PROTOCOL DESIGN

In this section, we present the design of AASR protocol. Considering the nodal mobility, we take the on-demand ad hoc routing as the base of our protocol, including the phases of route discovery, data transmission, and route maintenance. In the route discovery phase, the source node broadcasts an RREQ packet to every node in the network. If the destination node receives the RREQ to itself, it will reply an RREP packet back along the incoming path of the RREQ. In order to protect the anonymity when exchanging the route information, we redesign the
formats of the RREQ and RREP, and modify the related processes. As an example, we use a five-node network to illustrate the authenticated anonymous routing processes. The network is shown in Fig.1, in which the source node *S* discovers a route to the destination node *D*

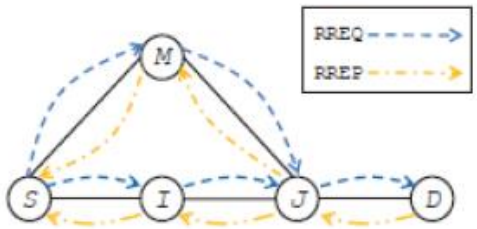

Fig. 1. Network topology

A. Anonymous Route Request

1) Source Node: We assume that *S* initially knows the information about *D*, including its pseudonym, public key, and destination string. The destination string *dest* is a binary string, which means "You are the destination" and can be recognized by *D*. If there is no session key, *S* will generate a new session key *KSD* for the association between *S* and *D*. The following entry will be updated in *S*'s destination table.

Dest.Nym.	Dest.Str	Dest. Pub_Key	Session_Key
N_D	*dest*	K_{D+}	K_{SD}

Then, *S* will assemble and broadcast an RREQ packet in the format of (1). To simplify the notation, we ignore the timestamp information in the RREQ packet.

$$S \rightarrow *: [RREQ, N_{sq}, V_D, V_{SD}, Onion(S)]G_{S-} \quad (1)$$

$$Onion(S) = O_{K_v}(N_S) \quad (5)$$

where *NS* is a one-time nonce generated by *S* to indicate itself. The core is encrypted with the symmetric key of *Kv*,

where *RREQ* is the packet type identifier; *Nsq* is a sequence number randomly generated by *S* for this route request; *VD* is an encrypted message for the request validation at the destination node ; VSD is an encrypted message for the route validation at the intermediate nodes; onion(S) is a key encrypted onion created by *S*. the whole RPEQ packet is finally signed by S with its group private key GS_. The combination of *VD* and *VSD* works similarly to the global trapdoor used in ANODR. We introduce *VSD*:

$$V_{SD} = (N_v)K_v \quad (2)$$

where *Nv* and *Kv* are two parameters created by *S* and sent to *D* for future route verification; *Nv* is a one-time nonce for the route discovery; and *Kv* is a symmetric key.
The secret message *VD* is defined as:

$$V_D = \langle N_v, K_v, dest \rangle K_{SD}, \{K_{SD}\}K_{D+} \quad (3)$$

If *D* is the receiver of the message, *D* can decrypt the second part of *VD* by its private key *KD*, and then decrypt the first part by the obtained *KSD*. Otherwise, the receiver knows that it is not the intended destination. If *S* and *D* have already established *KSD* in a previous communication, the costly public encryption in the second part of *VD* can be eliminated, and then *VD* is defined as:

$$V_D = \langle N_v, K_v, dest \rangle K_{SD}, pad \quad (4)$$

where *pad* is a pre-defined bit-string that pads the message to a constant length. *VSD* and *VD* are separated in the RREQ format (1). For a non-destination node, it can use *VSD* as a unique identity for the route request. Now we describe the encrypted onion *Onion*(*S*). *S* creates the onion core as follow:

and can only be decrypted by *D* via *Kv*. After sending the RREQ, *S* creates a new entry in its routing table, which looks like the following:

Req. Nym.	Dest.Nym.	Ver. Msg.	Next_hop	Status
N_{sq}	N_D	V_{SD}	N/A	Pending

2) Intermediate Node: The RREQ packet from *S* is flooded in **T**. Now we focus on an intermediate node *I*, as shown in Fig. 1. We assume that *I* has already established the
neighbor relationship with *S* and *J*. *I* knows where the RREQ packet comes from. The following entries are stored in *I*'s neighborhood table:

Neigh. Nym.	Session_Key
N_S	K_{SI}
N_J	K_{IJ}

Once *I* receives the RREQ packet, it will verify the packet with its group public key *GT+*. As long as the packet is signed by a valid node, *I* can obtain the packet information. Otherwise, such an RREQ packet will be marked as malicious and dropped. *I* checks the *Nsq* and the timestamp in order to determine whether the packet has been processed before or not.

If the *Nsq* is not known in the routing table, it is a new RREQ request; if the *Nsq* exists in the table but with an old timestamp, it has been processed before and will be ignored; if the *Nsq* exists with a fresh timestamp, then the RREQ is a repeated request and will be recognized as an attack. Then *I* tries to decrypt the part of *VD* with its own private key. In case of decryption failure, *I* understands that it is not the destination of the RREQ. *I* will assemble and broadcast another RREQ packet in the following format:

$$I \rightarrow *: [RREQ, N_{sq}, V_D, V_{SD}, Onion(I)]G_{I-} \qquad (6)$$

where *Nsq* , *VD*, and *VSD* are kept the same as the received RREQ packet; the key-encrypted onion part is updated to *Onion*(*I*). The complete packet is signed by *I* with its group

2) Intermediate Node: We assume that *J* has already established a neighbor relationship with *I*, *D*, and *M*. The following entries are already in *J*'s neighborhood table:

Neigh. Nym.	Session_Key
N_D	K_{JD}
N_I	K_{IJ}
N_M	K_{MJ}

If *J* receives the RREP from *D*, *J* will navigate the shared keys in its neighborhood table, and try to use them to private key *GI*. *I* updates the onion in the following way:

$$Onion(I) = O_{K_{SI}}(N_I, Onion(S)) \qquad (7)$$

where *NI* is a one-time nonce generated by *I* to indicate itself; *Onion*(*S*) is obtained from the received RREQ packet; this layer of onion is encrypted with the symmetric key *KSI* . When *I*'s RREQ reaches the next hop *J*, *J* will perform the same procedures and update the onion in the RREQ with one more layer, which is:

$$Onion(J) = O_{K_{IJ}}(N_J, Onion(I)) \qquad (8)$$

The routing tables of *I* and *J* will also be updated with a new entry as follow:

Req. Nym.	Dest.Nym.	Ver. Msg.	Next_hop	Status
N_{sq}	N/A	V_{SD}	N/A	Pending

3) Destination Node: When the RREQ packet reaches *D*, *D* validates it similarly to the intermediate nodes *I* or *J*. Since *D* can decrypt the part of *VD*, it understands that it is the
destination of the RREQ. *D* can obtain the session key *KSD*, the validation nonce *Nv*, and the validation key *Kv*. Then *D* is ready to assemble an RREP packet to reply the *S*'s route request.

$$D \rightarrow *: (RREP, N_{rt}, \langle K_v, Onion(J)\rangle K_{JD}) \qquad (9)$$

B. Anonymous Route Reply

1) Destination Node: When *D* receives the RREQ from its neighbor *J*, it will assemble an RREP packet and send it back to *J*. The format of the RREP packet is defined as follow:
where RREP is the packet type identifier; *Nrt* is the route pseudonym generated by *D*; *Kv* and *Onion*(*J*) are obtained from the original RREQ and encrypted by the shared key *KJD*. The intended receiver of the RREP is *J*.

decrypt ⟨ *Kv;Onion*(*J*)⟩ *KJD*. In case of a successful decryption, *J* knows the RREP is valid and from *ND*, and *J* also obtains the validation key *Kv*. Then *J* continues to decrypt the onion part. *J* knows the next hop for the RREP is *NI* . Then *J* will verify the linkage of the received RREP with its stored RREQ. It tries to use the obtained *Kv* to decrypt the verification message *VSD* stored in its routing table. Once *J* finds the matched *VSD*, it will update the corresponding routing entry as follows:

Req. Nym.	Dest.Nym.	Ver. Msg.	Next_hop	Status
N_{sq}	N/A	V_{SD}	N_D	Active

Since *Nv* in *VSD* is not issued by *J*, *J* is not the source of the RREQ, then it has to assemble another RREP and forward it. The format of *J*'s RREP towards the previous hop *I* is defined as:

$$J \rightarrow * : (RREP, N_{rt}, \langle K_v, Onion(I) \rangle K_{IJ}) \qquad (10)$$

where *Nrt* and *Kv* are obtained from the received RREP; *Onion*(*I*) is obtained by from the decrypted *Onion*(*J*); the shared key *KIJ* is obtained from *J*'s neighborhood table. The
intended receiver of the RREP is *I*.

When the RREP packet travels according to the layers on the onion, it will start at the destination node and move back to its previous node. Each time the intermediate node can associate a value with the underlying wireless link on which the RREP travels, until the RREP packet reaches the source. In our protocol, every node records the one-time link pseudonyms announced by its neighbor node. Then the intermediate nodes forwarding tables can be established after the RREP's trip. Now we discuss the forwarding table in detail. After *J* updates its routing table, it will also create a new entry in its forwarding table. It may record the multiple paths found in
the route discovery. According to the topology in Fig. 1, *J*'s forwarding table may look like the following, in which *NX;I* stands for the *i*th one-time pseudonyms issued by node *X*: *D* issues different pseudonyms *ND;*1 and *ND;*2 to *J*. There are two forwarding relationships at *J*. *NI;*1 : *ND;*1 and *NM;*1 :

Rt. Nym.	Pre_hop Nym.	Next_hop Nym.
$N_{rt,1}$	$N_{I,1}$	$N_{D,1}$
$N_{rt,2}$	$N_{M,1}$	$N_{D,2}$

*ND;*2 describe the two routes of $I - J - D$ and $M - J - D$, as shown in Fig. 1. It can be seen that the forwarding table is made anonymous to any nodes, except for the switching node that owns the table. At the time of being anonymized, the switching relationship at each node en route can also be guaranteed.

3) Source Node: When the RREP packet reaches *S*, *S* validates the packet in a similar process to the intermediate nodes. If the decrypted onion core *NS* equals to one of *S*'s issued nonce, *S* is the original RREQ source. *S* will update its routing table as follow:

Req. Nym.	Dest.Nym.	Ver. Msg.	Next_hop	Status
N_{sq}	N_D	V_{SD}	N_I	Active

Then the route discovery process ends successfully. *S* is ready to transmit a data along the route indicated by *Nrt*.

$$S \rightarrow D : (DATA, N_{rt}, \langle P_{data} \rangle K_{SD}) \qquad (11)$$

C. Anonymous Data Transmission

Now *S* can transmit the data to *D*. The format of the data packet is defined as follows:
where *DATA* is the packet type; *Nrt* is the route pseudonym that can be recognized by downstream nodes; the data payload is denoted by *Pdata*, which is encrypted by the session key KSD.

Upon receiving a data packet, every node will look into its forwarding table. If *Nrt* in the data packet matches one entry in forwarding table, the node will forward the packet to the anonymous next hop. Otherwise, the data packet will be discarded. Following the similar mechanism as the VCI in ATM network, the data packet can be switched along the route
until it arrives at the destination.

D. Routing Procedure

The routing algorithm can be implemented based on the existing on-demand ad hoc routing protocol like AODV or DSR. The main routing procedures can be summarized as follows:

1) During route discovery, a source node broadcasts an RREQ packet in the format of (1).
2) If an intermediate node receives the RREQ packet, it verifies the RREQ by using its group public key, and adds one layer on top of the key-encrypted onion, as (7). This process is repeated until the RREQ packet reaches the destination or expired.
3) Once the RREQ is received and verified by the destination node, the destination node assembles an RREP packet in the format of (9), and broadcasts it back to the source node
4) On the reverse path back to the source, each intermediate node validates the RREP packet of (2) and updates its routing and forwarding tables. Then it removes one layer on the top of the key-encrypted onion, and continues broadcasting the updated RREP in the format of (10).
5) When the source node receives the RREP packet, it verifies the packet, and updates its routing and forwarding tables. The route discovery phase is completed.
6) The source node starts data transmissions in the established route in the format of (11). Every intermediate node forwards the data packets by using the route pseudonym.

IV. CONCLUSION

In this paper, we design an authenticated and anonymous routing protocol for MANETs in adversarial environments. The route request packets are authenticated by group signatures, which can defend the potential active anonymous attacks without unveiling the node identities. The key-encrypted onion routing with a route secret verification message is designed to not only record the anonymous routes but also prevent the intermediate nodes from inferring the real destination. Compared to ANODR, AASR provides higher throughput and lower packets loss ratio in different mobile scenarios in the presence of adversary attacks. It also provides better support for the secure communications that are sensitive to packet loss ratio.

In our future work, we will improve AASR to reduce the packet delay. A possible method is to combine it with a trustbased routing [24]. With the help of the trust model, the routing protocols will be more active in detecting link failures, caused either by the mobility or adversary attacks.

REFERENCES

[1] D. Kelly, R. Raines, R. Baldwin, B. Mullins, and M. Grimaila, "Towards a taxonomy of wired and wireless anonymous Networks," in *Proc. IEEE WCNC'09*, Apr. 2009.

[2] C. Perkins, E. Belding-Royer, S. Das, *et al.*, "RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing," *Internet RFCs*, 2003.

[3] D. Johnson, Y. Hu, and D. Maltz, "RFC 4728 - The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *Internet RFCs*, 2007.

[4] J. Kong and X. Hong, "ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad hoc networks," in *Proc. ACM MobiHoc'03*, Jun. 2003, pp. 291–302.

[5] J. Kong, X. Hong, and M. Gerla, "ANODR: An identity-free and ondemand routing scheme against anonymity threats in mobile ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 8, pp. 888– 902, Aug. 2007.

[6] 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 *Proc. IEEE Int'l Conf. Local Computer Networks (LCN'04)*, Nov. 2004, pp. 618–624.

[7] R. Song, L. Korba, and G. Yee, "AnonDSR: efficient anonymous dynamic source routing for mobile ad hoc networks," in *Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN'05)*, Nov. 2005.

[8] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," in *Proc. IEEE INFOCOM 2005*, vol. 3, Mar. 2005, pp. 1940–1951.

[9] Y. Zhang, W. Liu, W. Lou, and Y. G. Fang, "MASK: Anonymous On-Demand Routing in Mobile Ad hoc Networks," *IEEE Trans. on Wireless Comms.*, vol. 5, no. 9, pp. 2376–2386, Sept. 2006.

[10] L. Yang, M. Jakobsson, and S. Wetzel, "Discount anonymous on demand routing for mobile ad hoc networks," in *Proc. Int. Conf. on SECURECOMM'06*, Aug. 2006