Strongly Anonymous Communications in Mobile Ad Hoc Networks

Y.Dong[1], V.O.K.Li[1], S.M.Yiu[2] and C.K.Hui[2]

Dept. of Electrical and Electronic Engineering, the University of Hong Kong[1]
Dept. of Computer Science, the University of Hong Kong[2]
Pokfulam Road, Hong Kong, China
e-mail: {ydong, vli}@eee.hku.hk, {smyiu, hui}@cs.hku.hk

Abstract

A mobile ad hoc network consists of mobile nodes that communicate in an open wireless medium. Adversaries can launch analysis against the routing information embedded in the routing message and data packets to detect the traffic pattern of the communications, thereby obtaining sensitive information of the system, such as the identity of a critical node. In order to thwart such attacks, anonymous routing protocols are developed. For the purposes of security and robustness, an ideal anonymous routing protocol should hide the identities of the nodes in the route, in particular, those of the source and the destination. Multiple routes should be established to increase the difficulty of traffic analysis and to avoid broken paths due to node mobility. Existing schemes either make the unrealistic and undesired assumption that certain topological information about the network is known to the nodes, or cannot achieve all the properties described in the above. In this paper, we propose an anonymous routing protocol which can satisfy all the required properties.

Keywords

Anonymous routing, MANETs.

1. Introduction

Mobile ad hoc networks (MANETs) are increasingly adopted in both military and civilian uses due to its self-configuration and self-maintenance capabilities. MANETs are highly vulnerable to security threats due to its inherent characteristics such as wireless transmission, lack of fixed infrastructure, dynamically changing topology, etc. The broadcast nature of the wireless medium makes MANETs susceptible to various malicious attacks. Traffic analysis is one of the most serious security attacks in MANETs. By observing network traffic pattern, adversaries can obtain sensitive information about the applications even without revealing the contents of the messages. For example, an attacker can identify the communicating parties and their positions by tracing and analyzing the network traffic patterns. This may lead to severe threats in security-sensitive applications. For instance, in a battle field the enemy can physically destroy the important mobile nodes if they can identify and locate such nodes by traffic analysis. In order to thwart such attacks, anonymous communication protocols are developed.

To prevent the possible traffic analysis attacks in MANETs, anonymous routing schemes which enable anonymous communications were investigated. In (Kong and Hong, 2003), Kong and Hong proposed to use pseudonyms instead of real identities in the route discovery protocol to hide the identities of the intermediate nodes in the route. The scheme makes use of Onion in the route discovery protocol, as applied in the Internet for anonymous data transmission, to establish an anonymous route. Song et al. (Song et al. 2005) presented another secure anonymous routing protocol (AnonDSR) for MANETs. The protocol employs anonymous onion routing between the source and destination, and each intermediate node owns a shared session key with the source and destination nodes when the protocol is completed. These routing schemes are sensitive to the node mobility because only one route is established in the route discovery. As nodes move, the path may be broken and has to be reestablished. To solve this problem, (Zhang, et al. 2005) proposed an anonymous on-demand routing protocol, called MASK, which can establish multiple routes for data transmission by indicating the real identity of the destination node in the route request packet. With the knowledge of the destination identity in the route request, MASK can obtain multiple routes with the route information cached in other nodes, which cannot be achieved by any other anonymous protocol due to the hiding of the targeted destination node in the route request packet. Although observers cannot correlate a real identity with a particular node, it may detect the traffic pattern of the applications in the system, for example, if most of the data flows are destined to the same identity, attackers can conclude that the node with the identity may be a critical node in the network.

In this work, we propose a new anonymous routing protocol. We employ the Diffie-Hellman key agreement algorithm to design an anonymous route discovery protocol. With the use of the Bloom filter, multiple anonymous routes can be established to achieve the random route transmission in the data packet forwarding phase to prevent adversaries from correlating the captured data transmissions with each other. We also use bloom filter to detect loops in routes.

The rest of the paper is organized as follows. Section 2 provides the background information. Section 3 presents the details of the anonymous route discovery scheme. The discussion and analysis are provided in Section 4. Conclusion is given in Section 5.

2. Preliminaries

2.1. Bloom filter

A Bloom filter (Bloom, 1970) is a method for representing a set of $A = \{a_1, a_2, ..., a_n\}$ of n elements to support membership queries. The idea is to allocate a vector v with m bits, initially all set to 0, and then choose k independent hash functions, $h_1, h_2, ..., h_k$, each with range $\{1, ..., m\}$. For each element $a \in A$, the bits at the positions $h_1(a), h_2(a), ..., h_k(a)$ in v are set to 1. (A particular bit might be set to 1 multiple times.) Given a query for b we check the bits at positions $h_1(b), h_2(b), ..., h_k(b)$. If any of them is 0, then b is definitely not in set A.

Otherwise we conjecture that b is in the set although there is a certain probability that we are wrong. This is called a "false positive".

2.2. Diffie-Hellman Key Agreement Algorithm

The Diffie-Hellman key agreement algorithm (Diffie and Hellman, 1976) allows two parties to exchange a secret key over an insecure medium. Suppose Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol. The procedure is as follows. First, Alice generates a random private value a, while Bob generates a random private value b. Then they derive their public values using parameters p, g and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$. They then exchange their public values. Finally, Alice and Bob now have a shared secret key, $g^{ab} \bmod p$.

3. Multiple Anonymous Routes Discovery

The anonymous route discovery protocol consists of two phases: anonymous route request phase in which the source send anonymous route request to the intended destination, and anonymous route reply phase in which the destination give a reply to the source.

3.1. Anonymous Route Request

The anonymous route request phase allows a source node *S* to discover and establish a routing path to a destination node *D* through a number of intermediate nodes. To keep communication anonymity, none of the intermediate nodes participating in this phase should discover the identities of *S* and *D*. The source node *S* triggers an anonymous route request (ARREQ) by broadcasting an ARREQ packet to its neighboring nodes. The format of the ARREQ packet is as follows,

$$< ARREQ, H(N), y_s, E_{PK_d}(ID_d, ID_s, K_{sd}), E_{K_{sd}}(N) > \qquad (1)$$

$H(N)$ is the hash value of a randomly generated integer N, and serves as the unique identifier of the request. It is also used by the intermediate nodes to validate whether an anonymous route reply is generated by the real destination in the route reply phase. The source node *S* generates a random private value x_S to establish a secure link with its forward node by applying the Diffie-Hellman key agreement algorithm. Forward node is the next node in the route request phase and data forwarding phase. y_S is the corresponding public value of secret x_S, and also serves as the pseudonym of the source node, which is the temporary identity of node *S*. The encrypted data block, $E_{PK_d}(ID_d, ID_s, K_{sd})$, contains the identities of the source and the destination node, and the symmetric key K_{sd} generated by the source node, all encrypted by the public key PK_d of the destination, thus only the intended

destination node can decrypt the information with its private key to obtain the symmetric key K_{sd} and the value of N.

When a node i receives an ARREQ packet with the following format,

$$< ARREQ, H(N), y_{i-1}, E_{PK_d}(ID_d, ID_s, K_{sd}), E_{K_{sd}}(N) > \qquad (2)$$

it processes the packet according to the following steps.

1. Check if the packet has already been received, using $H(N)$ as the unique identifier for the packet.
2. If the message has not been received, then
    a. Check if it is the sender's intended destination: decrypts $E_{PK_d}(ID_d, ID_s, K_{sd})$ with its private key, and compares the ID_d with the node's id.
    b. If node i is not the intended destination, it records $H(N)$ and y_{i-1} into its routing table, then generates a random number x_i and computes the corresponding public value y_i. Finally node i replaces y_{i-1} with y_i in the received ARREQ packet and rebroadcasts the request to its neighbors.
    c. If node i is the destined destination, it generates an anonymous route reply packet and reverses it to the source.
3. If the message has been received, check if the pseudonym y_{i-1} has been recorded in the routing table associated with $H(N)$.
    a. If y_{i-1} is one of the nodes in the routing table, drop the packet and stop.
    b. Otherwise node i records the pseudonym y_{i-1} into its routing table as one of the reverse node. The reverse node is the next node in the reverse path towards the source in the route reply phase.

In the anonymous route request phase, each intermediate node maintains all the pseudonyms from which it receives the ARREQ packet. The structure of each entry maintained in the routing table is $< H(N), (x_i, y_i), y_{i-1,1}, ..., y_{i-1,k} >$, which includes the identifier of the request, $H(N)$, the temporary key pair (x_i, y_i), and the list of reverse node pseudonyms.

3.2. Anonymous Route Reply

The ARREQ packet is forwarded in the network until it reaches the target destination node *D*. Node *D* retrieves K_{sd} with its private key, and obtains the value of N encrypted in $E_{K_{sd}}(N)$. Then it composes an anonymous route reply (ARREP) packet, and sends it back to the source node. The format of the ARREP packet from

an intermediate node i is $< y_{i-1}, y_i, E_{K_{i-1,i}}(N') >$, where y_{i-1} is the pseudonym of one of the reverse nodes recorded in the routing table, and y_i is the pseudonym of node i generated in the anonymous route request phase.

An intermediate node $i-1$ receiving an ARREP packet first checks the y_{i-1} field in the packet. If the value of y_{i-1} does not match its own pseudonym, node $i-1$ will discard the packet. Otherwise, as the intended reverse node, it computes the secret key $K_{i-1,i}$ with y_i and x_{i-1} using the Diffie-Hellman key agreement algorithm, and obtains the value of N' by decrypting $E_{K_{i-1,i}}(N')$ with the secret key $K_{i-1,i}$. Node $i-1$ computes $H(N')$, and compares $H(N')$ with $H(N)$ which is recorded in the routing table in the route request phase. If they are not equal, node $i-1$ discards the ARREP packet. If they are equal, node $i-1$ believes that the received ARREP is originated from the intended receiver, since only the intended receiver can compute the value of N. Then node $i-1$ records y_i as one of its forward nodes in the routing table together with the corresponding secret key $K_{i-1,i}$. Finally, node $i-1$ constructs and broadcasts an ARREP packet containing its reverse node pseudonym, its own pseudonym, and encrypted value of N with the secret key shared between the reverse node and itself. The same procedure is repeated until the ARREP packet reaches the source node *S*.

An intermediate node may have to send ARREP packets to multiple nodes if it maintains several reverse nodes. Here we use an efficient method to send the multiple replies in one packet. All the reverse node pseudonyms are listed in order in the ARREP packet together with the same number of data blocks encrypted with the corresponding secret keys. The format of the ARREP packet is as follows,

$$\begin{aligned} &< y_{i,1}, \ldots, y_{i,n}, \\ &\quad y_{i+1}, \\ &\quad E_{K_{i+1,i,1}}(N'), \ldots, E_{K_{i+1,i,n}}(N') > \end{aligned} \tag{3}$$

The order of the encrypted data blocks is consistent with that of the pseudonyms listed in the packet. Whenever a node $y_{i,j}$ receives an ARREP packet and finds its pseudonym in the list, it checks the encrypted data block in the corresponding order of the encrypted data blocks. If N' decrypted from the data block is correct, it records y_{i+1} as one of the forward nodes in its routing table.

At the end of the anonymous route reply phase, each intermediate node maintains the following information in its routing table, and multiple anonymous routes would be established hop by hop.

$$< H(N),(x_i, y_i),$$
$$reverse\_node\_list, reverse\_node\_key\_list,$$
$$forward\_node\_list, forward\_node\_key\_list >$$

3.3. Establishment of Multiple Loop-free Anonymous Route

In the above anonymous route discovery scheme, when an ARREQ packet arrives at a node, the pseudonym in the packet will be accepted as one of the reverse nodes if it is not in the list of the reverse nodes in the routing table, which may lead to routing loops. To see how loops can occur, consider a simple example. Source S initiates a route request by flooding ARREQ to all its neighboring nodes. An intermediate node A broadcasts the ARREQ packet. One of its neighbors B rebroadcasts it, which in turn is heard by node A. If A accepts this ARREQ copy to form a reverse path, a loop will be formed. On the other hand, loops cannot be formed and an alternate route can be constructed if A accepts a duplicate copy of the ARREQ arriving via a trajectory that does not already include A.

In order to eliminate the possibility of loops, most of the existing protocols based on the on-demand distance vector routing scheme (Marina and Das, 2001) (Higaki and Umeshima, 2004) for MANETs make use of cached routes information to achieve multiple loop-free routes. However, in an anonymous communication system, the identity of a destination node in an ARREQ packet can not be revealed by all the nodes except the intended receiver, therefore there is no route information cached in any intermediate node. In order to obtain the multiple loop-free routes in the anonymous communication system, we make use of Bloom filter to detect the routing loop. The Bloom filter is embedded in the packet header, and is used to track the set of nodes it visited. To avoid loops, each node detecting its existence in the Bloom filter will not forward the packet. We make use of this application to avoid loops in the anonymous route request phase.

In route request phase, a source node *S* embeds a Bloom filter in the ARREQ packet. When an intermediate node i receives an ARREQ packet it has never seen before, it applies the k hash functions to its pseudonym y_i and sets the corresponding positions of the Bloom filter to one, and then rebroadcasts the ARREQ packet. If it has seen an ARREQ packet with the same identifier $H(N)$ but coming from a node with different pseudonym, it first checks whether its pseudonym y_i exists in the Bloom filter embedded in the received request. If y_i exists in the Bloom filter, node i discards the ARREQ packet since the packet has already been received before. Otherwise, node i records the pseudonym as one of the reverse nodes in its routing table. The format of the ARREQ packet is extended as

$$< ARREQ, H(N), y_i, BF_i, E_{PK_d}(ID_d, ID_s, K_{sd}), E_{K_{sd}}(N) > .$$

Combined with the procedure of the anonymous route request described in Section 3.2, the complete procedure is shown in Figure 1.

The destination node may receive multiple ARREQ packets that traverse joint or disjoint routes before reaching it. The destination node can deduce the route relationship information, completely disjoint or partially joint, by comparing each pair of Bloom filters in different ARREQ packets, since the Bloom filter records the pseudonyms of intermediate nodes on the route. For two Bloom filters, the more the number of different positions set to 1, the higher the probability that the Bloom Filters contains different intermediate nodes. In the case that two Bloom filters are completely different, that is all positions of 1 are different in the two Bloom filters, they must represent two disjoint routes. The destination can reply all ARREQ packets, or only reply those ARREQ packets with more different positions in the Bloom filters if disjoint routes are preferred.

An intermediate node may find that its pseudonym is contained in the Bloom filter of an ARREQ packet, and discard the packet accordingly, however in fact it has not received the copy of ARREQ packet from the node. This happens when the node's bit positions in the Bloom filter happen to be a combination of other nodes' bit positions, which is called the false positive of Bloom filter. The false positive probability of a Bloom filter is related with the configuration of the Bloom filter, $(1-e^{km/b})^k$, and could be reduced by increasing m, the Bloom filter size, or increasing k, the number of hash function. Once false positive of the Bloom filter occurs, the links established from the node will be missed, thus some available routes are not discovered, which would not defeat our scheme.

Normally, the initial Bloom filter is set to be empty in all applications. However, considering the anonymity feature, the initial value of the Bloom filter should not be zero in our scheme. Otherwise, a malicious observer may be aware of the location of the source node once it captures the packet with an empty Bloom filter. To avoid such an attack, the initial value of the Bloom filter should be set with a certain value by the source node. The source node can generate n random integers, insert them into the Bloom filter by applying k hash functions to each of them, and embed the resultant Bloom filter into the ARREQ packet. The number of random integer is chosen randomly from a range $[1, R]$, where R is the system parameter. By this means, when an attacker captures an ARREQ packet, it is hard for him to guess the distance or the location of the source node.

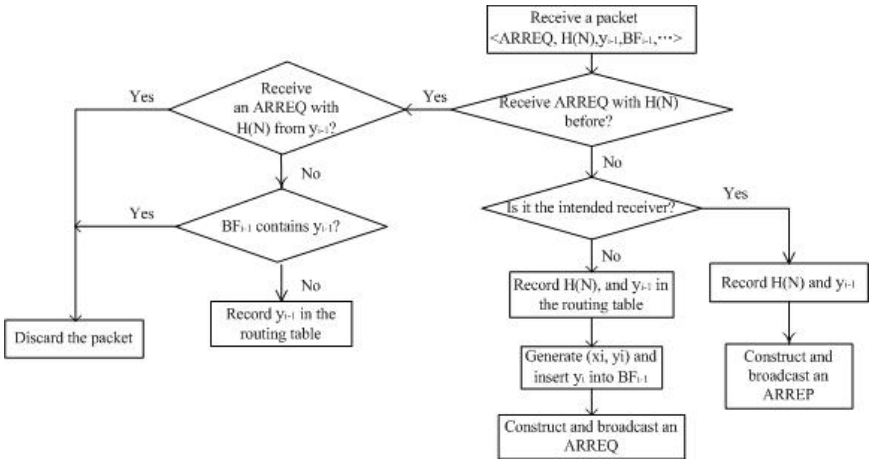

Figure 1: The procedure of loop-free anonymous route request

4. Discussion and Analysis

4.1. Anonymity Analysis

In this work, we aim at designing an anonymous routing scheme to prevent the passive adversaries from identifying the critical node or detecting the traffic pattern of the applications by observing the traffic and mounting traffic analysis attack to the captured traffic. An adversary may have local or global view of the network traffic according to its strength, for example, computational power. The adversary with the global view of the traffic could detect the traffic flow by observing the common contents in packets at different locations. To prevent such an attack, we make the common contents appear different at each hop with different link keys. Overall, our scheme presents the following anonymity features.

1. Identity anonymity: In our design, each node uses its random pseudonym instead of real identity to identify itself, and only the end nodes in the communication can identify each other, and any other node, including the intermediate nodes on the route, cannot match a pseudonym to a particular node. Morever, since the pseduonym of a node is changed with each session, it is hard for an adversary to trace a particular node.

2. Location anonymity. In (Song et al. 2005), the pseudonyms of all the intermediate nodes are exposed to the communication nodes, while in another anonymous routing scheme (Zhu et al. 2004) the hops on an anonymous route (the distance between the source and the destination) is revealed to the source and the destination nodes without the pseudonyms of the nodes on the route. However, these anonymous routing scheme may reveal the topology information of the network to a malicious node. For example, an adversary can intentionally generate route requests to all other nodes in the network, and get to know the location information, the distance of other nodes, from the route reply message. In our scheme, the source

node, the destination node, and the intermediate nodes are unaware of the identities information on the route, and each node only knows the pseudonyms of its one-hop forward node and reverse node. No informaiton about the locations is exposed to any node.

3. Route anonymity: An adversary with the global view of the network traffic may deduce the route by detecting common information among sniffed packets with the assumption that two packets are transferred along the same route with higher probability if they have some information in common. In our protocol, the same content in the route reply packet, N', appears different at each node by encrypted with different link key, which makes it extremely hard for an adversary to detect the traffic flow by finding the common information in the packets. To make it further hard for the adversaries to correlate the observed transmmision with each other and acquire the actual network pattern, we use randomly selected route from multiple available routes for data transmission.

In summary, we compared our routing scheme with other existing anonymity solutions for MANETs in the aspects of identity anonymity, location anonymity, and route anonymity in 0

4.2. Security Analysis

The main security attacks include passive attacks and active attack. In the above analysis, we have shown the means we take to encounter with the passive attacks in which an adversary observes and analyze the network traffic to obtain the sensitive information of the application or the underlying system. In this section, we discuss the active attacks an adversary could launch to our scheme in route discovery and data transmission.

In the anonymous route discovery, an adversary may mount the following active attacks to the protocol. 1) A malicious node may modify the pseudonym in a route request packet, and replace it with its own one, so that it could be on the route, and then mount the severe attacks to the communications, such as intentionally dropping, delaying or altering the data traffic passing through it. Actually, the anonymous route request is flood to the network when a route is required. Without any information of the destination node, it is hard for an adversary to locate the position of inserting a route request packet in order to be on the route. If it is lucky to be on the route, the random selection of routes for data transmission would help to reduce the damage the malicious node could perform. In the route reply phase, an adversary cannot modify the content of the route reply packet or generating a route reply packet without the corresponding secure link key established in the route request phase. 2) An adversary may modify the Bloom filter in a route request, for instance, trivially set all bits of the Bloom filter to 1, or change some positions of the Bloom filter. The Bloom filter is used for loop detection in our protocol. When a node receives a route request packet with the request identifier that have been cached in the routing table, it accepts and processes the request only when the embedded Bloom filter does not include its pseudonym, otherwise it discards the request for loop detection. Therefore, the modification of the embedded Bloom filter will affect the multiple

routes establishment. In the case of trivially set all bits of the Bloom filter to 1, the request packet will be blocked in the vicinity of the attacking point, thus some potential routes between the source and destination nodes could not be established. In the case that the Bloom filter is modified in some positions, the consequence is less severe, and only some of the routes would be missed. The problem is alleviated by establishing multiple routes for data transmission. 3) Denial-of-Service (DoS) attack could be a very dangerous attack. An adversary can simply flood the anonymous route request packets to exhaust the computation resources of the nodes in the route, since they try to decrypt the encrypted data in the request packets. Currently all the anonymous routing protocols for MANET (Song et al. 2005) (Kong and Hong, 2003) (Zhang et al. 2005) (Zhu et al. 2004) are vulnerable to the DoS attack. The effective method to solve the problem is to provide authentication for each request message. However, it is non-trivial to provide authentication in anonymous communication system since the real identities of nodes are hidden from others. Symmetric key based authentication can be provided by sharing a system authentication key to all the nodes, while the public key based authentication can be provided by anonymous authentication algorithm (Schechter et al. 1999) in the anonymous environment.

In anonymous data transmission protocol, the source node and the destination node secure their communications with the end-to-end session key. Also, any two adjacent nodes on the route make use of the temporary link key established in the route discovery protocol to protect their communications. Malicious nodes without the corresponding keys could not interfere with the data traffic.

Comparison of anonymity features

Categories/protocol	Our scheme	AnonDSR	MASK
Identity anonymity	Strong	Strong	Weak
Location anonymity	Strong	Weak	Strong
Route anonymity (multiple routes)	Strong	Weak	Strong

Table 1: Comparison of anonymity features

4.3. Communication and Computation Overhead

In our scheme, we make use of the Diffie-Hellman key agreement algorithm, and the Bloom filter to establish multiple loop-free anonymous routes, which introduce the computational overhead. In this section, we analyze the overhead of our anonymous routing protocol compared with normal routing protocol for MANETs. Whenever a route is required, an ARREQ packet will be flooded in the network. The node will generate a secret integer and compute its corresponding public value when it receives a copy of the request from one of its neighbors. This temporary public and private key pair is prepared to be used to generate link key by the Diffie-Hellman algorithm if the node is one the route. Otherwise, the temporary key pair can be maintained in its memory for future uses. The Diffie-Hellman key agreement algorithm is performed only by the nodes on the established route. Bloom filter is used to record the list of nodes that have received the request. To insert its pseudonym in the Bloom

filter, a node has to perform k hashing functions, and set the corresponding positions to 1, which is very fast. Therefore, the Bloom filter only introduces a slight workload. On the other hand, the established multiple routes can reduce the overhead caused by re-establishing an anonymous route when a link in the route is broken due to the node mobility.

5. Conclusion

In this paper we design an anonymous routing protocol in which multiple routes a are established. The data packets can be transmitted on selected routes to make it hard for adversaries to trace data flow and correlate the data packets with the source and destination nodes. The multiple anonymous route discovery protocol provides strongly anonymous communications in MANETs. In the future, we would investigate the efficiency of our schemes with experimental data.

6. References

Boukerche, A., EI-Khatib, K., Xu, L. and Korba, L. (2004), "A novel solution for achieving anonymity in wireless ad hoc networks," *the 1st CM PE-WASUN,* pp. 30-38.

Bloom, B. (1970), "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, 1970, 13 (7): 422-426.

Diffie, W. and Hellman, M. E. (1976), "New directions in cryptography," *IEEE Transactions on Information Theory* 22, 644-654.

Higaki, H. and Umeshima, S. (2004), "Multiple-route ad hoc on-demand distance vector (MRAODV) routing protocol," *the 18th International Parallel and Distributed Processing Symposium conference (IPDPS'04)*, pp. 234-244.

Kong, J. and Hong, X. (2003), "ANODR: Anonymous on demand routing with untraceable routes for mobile ad hoc networks," *the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing conference (MobiHoc' 03)*, pp. 291-302.

Marina, M. K. and Das, S. R. (2001), "On-demand Multipath distance vector routing in ad hoc networks," *the IEEE International Conference for Network Protocols (ICNP'01)*, pp. 14-23.

Song, R., Korba, L. and Yee, G. (2005), "AnonDSR: Efficient anonymous dynamic source routing for mobile ad-hoc networks," *ACM Workshop on Security of Ad Hoc and Sensor Networks conference (SASN'05), pp. 33-42*.

Schechter, S. E., Parnell T. C. and Hartemink, A. J. (1999), "Anonymous authentication of membership in dynamic groups," *Financial Cryptography, Third International Conference*, FC'99, volume 1648 of Lecture Notes in Computer Science, pages 184-195.

Zhang, Y., Liu, W. and Lou, W. (2005), "Anonymous communications in mobile ad hoc networks," *the IEEE INFOCOM conference 05*, pp.1940-1951.

Zhu, B., Wan, Z. G. and Kankanhalli, M. S., (2005), "Anonymous secure routing in mobile ad-hoc networks," *the 29th Annual IEEE International conference on Local Computer Networks (LCN'04)*, pp102-108.