

A Secure Location Service for Ad Hoc Position-Based Routing Using Self-signed Locations^{*}

Jihwan Lim¹, Sangjin Kim², and Heekuck Oh¹

¹ Hanyang University, Department of Computer Science and Engineering,
Republic of Korea

jhl@hanyang.ac.kr, hkoh@hanyang.ac.kr

² Korea University of Technology and Education,
School of Information and Media Engineering, Republic of Korea
sangjin@kut.ac.kr

Abstract. Location service, which provides current geographic positions of nodes, is one of the key elements of position-based routing schemes for ad hoc networks. In this paper, we define security threats of location service and propose a new secure location service protocol that uses self-signed locations. In our proposed protocol, nodes register their public keys in other nodes during the initialization phase and these registered keys are used to verify the locations of other nodes and to generate their self-signed locations. In this paper, we show that our protocol is robust against traditional attacks and new attacks that may occur in position-based routings. We also analyze the efficiency of our protocol using various simulations.

Keywords: ad hoc network, position-based routing, secure location service.

1 Introduction

An ad hoc network is a network that does not use any existing infrastructure and is formed autonomously by mobile nodes. Participating nodes communicate with other nodes that are outside their transmission range by using multi-hop routing. In other words, a node plays the role of a router as well as a host. These nodes can also move freely causing the network topology to change dynamically. These characteristics make designing a scalable and robust routing protocols a real challenge.

Earlier researches on routing for ad hoc networks are based on table-driven or on-demand methods [1]. Recently, position-based routing methods are attracting many researches since these types of methods use geographical coordinates of nodes to effectively route messages [2,3]. In position-based routing, participating nodes can recognize their own geographic locations using equipments such as GPS (Global Positioning System). However, to route messages using the destination node's location, one must obtain

^{*} This research was supported by the MIC (Ministry of Information and Communication), Korea, under the HNRC (Home Network Research Center) - ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment). This work was also supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (No. R01-2006-000-10957-0).

such information. Therefore, in position-based routings, there must be a way for nodes to obtain locations of other nodes.

Location service of position-based routing provides such mechanism. Location service may also refer to locating a data item, but in this paper, this service refers to locating the position of the destination node. Generally, a node queries another node to retrieve such information. A node who maintains locations of other nodes are referred to as a location server. We can use a single centralized server that maintains locations of all the participating nodes. However, since true ad hoc networks do not have any central administration, distributed approach is more suitable. In such approach, each participating nodes normally manage locations of some other nodes.

The location service is composed of three components: location update, location request, and location response. Location servers must maintain the latest location of nodes it manages. In other words, location service is sensitive to nodes' mobility. To accomplish this task, a node reports its new location to the server as it moves around the network. This process is referred to as a location update. This update can occur periodically or when a node moves a certain distance from the previous reported location. A node requests the location of the destination node if it does not have that information in its cache. The location server responds by sending the latest location information it maintains to the requesting node.

Most of the current researches on position-based routings for ad hoc network do not deal with new security threats caused by location service. For example, messages can be routed to wrong location if the location information obtained from a server is false. In this paper, we propose a new secure location service for position-based routing that uses self-signed locations. However, we do not deal with location privacy of nodes. The reason is that making location information private conflicts with the inherent nature of position-based routing. In other words, the location servers must know the location information of other nodes and these servers cannot be regarded as trusted entities in ad hoc networks.

The remainder of this paper is organized as follows. In section 2, we briefly introduce related work. Our proposed scheme will be present in section 3. In section 4, we analyze the security and the efficiency of our scheme using various simulations. Finally, we conclude this paper in section 5.

2 Related Work

2.1 Location Services for Position-Based Routing

At one extreme, we can think of a scheme where each node maintains the locations of all the nodes in the system [4]. In such schemes, it is inevitable that a node must flood its location to other nodes periodically. Therefore, efficiency of a location update and storage burden on each node may be too heavy. To reduce the cost of a location update, most of the schemes send an update message to only a subset of nodes. At another extreme, a single location server can be assigned to each node [2]. These approaches, compared to flooding-based, are sometimes referred to as rendezvous-based protocol, since location servers serve as rendezvous point for updates and lookups. Rendezvous-based approach can be further divided into hash-based or quorum-based [2,3]. In this paper, we divide

rendezvous-based approach into static-based and dynamic-based depending on whether the rendezvous point is fixed or not.

Location services that use a HR (Home Region) [5,6] are typical examples of static-based approach. In these approaches, a universal hash function maps each node's identifier to a HR and nodes residing in that region serve as location servers of that node. When a node in a region receives an update request, it shares this information with other nodes in the region through local flooding. A node that wants to acquire the location information of a node sends a query to the HR of that node.

XYLS (column-row quorum-based Location Service) protocol [7,8] is a typical example of dynamic-based approach. In this protocol, a node reports its position to the nodes currently residing in the north-south direction of its current position. A node requests other nodes' location by sending a request message in the east-west direction. Therefore, there is always an intersection between an update message and a request message which guarantees that lookups will always be satisfied by some node.

2.2 Security of Ad Hoc Routing

Since participating nodes act as routers in ad hoc network, there are many security threats such as black hole, replay, worm hole, blackmail, and routing table poisoning [9]. However, current proposals to defend these attacks use unrealistic assumptions. For example, some assume that each pair of nodes shares a common secret with each other when nodes can freely join and leave the network [10,11]. Others assume that all nodes have a certificate issued by a common CA (Certification Authority) when it is difficult to predetermine which kind of nodes will participate in the network [12,13]. Therefore, we suggest a protocol that uses only self-signed certificates. In our scheme, nodes pre-register their public keys in other nodes and use an assumption that vast majority of nodes are honest.

2.3 Security of the Location Service

Positioning-based routing brings about new threats that do not exist in ad hoc networks based on other routing methods. The most obvious attacks on location service are as follows.

- False location update attack: An attacker may try to update the location of another node causing the server to maintain false location.
- False location response attack: An attacker may try to alter the response from a server causing a node to receive wrong location of another node.

In order to defend against these attacks, authentication of the sender and authentication of update and response messages must be provided. However, since nodes in ad hoc network cannot be regarded as trusted entities, authenticating the sender of a message is not sufficient to provide a secure location service. To this end, we use self-signed locations. More precisely, nodes sign their current location using their own private key and these signatures are maintained in location servers. Location servers also use these signatures to respond to location requests. Therefore, nodes receiving a location response do not have to trust the location servers.

Our idea requires some sort of PKI (Public Key Infrastructure). However, as stated earlier, using existing PKI is not a feasible solution due to the fact that there is no prior knowledge of the participants involved. Furthermore, true ad hoc networks may not have any access points. In other words, participating nodes may not have any certificates or even if nodes possess certificates it may not be possible to verify other parties' certificates while participating in the ad hoc network.

3 Proposed Protocol

3.1 Overview

Our proposal is not affected by the location service mechanism used by the network. In other words, our scheme can also operate in ad hoc networks that use static-based location service. However, in this paper, we will explain our protocol using the XYLS scheme proposed by Stojmenovic and Pena [7]. This scheme is based on the fact that a vertical and a horizontal line in a square always intersect with each other. In this scheme, as shown in Fig 1, a node updates by broadcasting its new position to the north and south of the current location and requests locations of other nodes by broadcasting the message to the east and west. As a result, there is always an intersecting point between location update and request messages. This enables nodes to receive the latest location information of other nodes. As with most of position-based routing protocols, a node updates when it moves a certain distance from the previous position or periodically. However, in our scheme, a node must send an update message in both cases. This is required since location servers remove old information from its table as time elapses.

To provide a secure location service, a node generates a public key pair and registers its public key in other nodes when it joins the network. If the majority of the nodes are honest, then this process will be sufficient to provide a safe public key environment. A node updates its position by sending its location digitally signed and a node receives this digitally signed location when requesting other node's location. If a node cannot obtain other node's private key, it will be infeasible for nodes to alter or generate a valid update or response message. The security of this mechanism will be discussed in more detail in section 4.1.

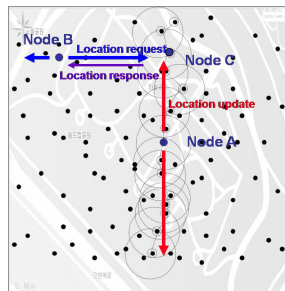


Fig. 1. Overview of Our Proposed Location Service

3.2 Assumption and Notation

We assume the followings about the ad hoc network environment and the nodes participating in our protocol.

- Nodes are assumed to be located uniformly in the given network. We also assume that they move at arbitrary speed and direction.
- All nodes are equipped with the same transmission radius and calculation capacity. That is, links between nodes are symmetric.
- Participating nodes can obtain their own geographical location through GPS and can accurately synchronize time through GPS.
- Participating nodes know in advance the protocols and algorithms used in the network to communicate with each other.
- Participating nodes know the unique ID of the respective node that they want to communicate with.
- Participating nodes can generate a public key pair by themselves and can generate and verify digital signatures.

Throughout this paper, we will use the notations given in Table 1.

Table 1. Notation

A	the identifier of node A .
A_{PK}, A_{PR}	the public and private key of node A .
$Sig_A(M)$	A 's signature on message M using A_{PR} .
$Cert_A$	the self-signed certificate of A 's public key.
T_{A_I}	the timestamp representing the time A_{PR} was first generated.
T_{A_C}	the timestamp representing the current time generated by A .
Pos_A	A 's geographical location (coordinate) obtained from GPS.
Loc_A	the self-signed geographical location of node A .
	$Loc_A = Sig_A(A Pos_A T_{A_I} T_{A_C})$

3.3 Registration and Initialization

We use self-signed locations to provide a secure location service. To use our idea, we require a public key system that can be used in true ad hoc networks. As stated earlier, using an existing PKI in ad hoc network may not be plausible. Therefore, we propose the following public announcement method instead of traditional PKI. However, if there is a more suitable way to provide a PKI for ad hoc network exists, such mechanism can be used instead.

A node who wants to join the network follows the following steps to announce and register its public key in other nodes.

- **Step 1.** A node A generates a public key pair (A_{PK}, A_{PR}) . It then creates a simple self-signed certificate of A_{PK} as follows:

$$Cert_A = Sig_A(A || A_{PK} || T_{A_I}).$$

- **Step 2.** The node A creates and vertically broadcast the following public key registration message (PK_init):

$$PK_init = [Type, Seq, width, Cert_A, Loc_A],$$

where $Type$ refers to the type of the message such as such as ‘public key registration’, ‘location update’, and ‘location request’, Seq indicates a sequence number used to prevent loops and duplicate messages, and $width$ is a system parameter denoting the transmission width in hop distances. As one can see, this message also includes the initial location update.

In ad hoc network, a broadcasted message is received by all the nodes within one hop distance of the sender and one of the receiver will forward the message to the next hop. The width of a message is sometimes referred to as the thickness of reporting. For example, if the width is 1, a node located one hop east and a node located one hop west also forward the message vertically.

We do not flood the PK_init message to reduce the cost of registration and storage requirement. Instead, a node broadcasts its PK_init message vertically. When a message is broadcasted vertically, there are several ways to process the message. We use the following method 1 for normal location updates and method 2 for PK_init messages.

- Method 1. A node receiving the message unconditionally stores the certificate. If there are total n nodes and they are uniformly distributed, about $2r/l \times n$ nodes will store the certificate, where r is the transmission range of a node and l is the width of the terrain of the network assuming that the terrain is a square.
- Method 2. A node receiving the message determines the geographic location of the original source of the message and stores the certificate if the location is within certain boundary. If the boundary is divided into s disjoint columns, then n/s nodes will store the certificate.

When a node receives a message, depending on the policy used, the node may be responsible for determining the validity of the message. Obviously, due to the efficiency consideration, all nodes receiving a message do not have to verify it. Most of the messages in our protocol include a digital signature such as self-signed locations. To verify these signatures, nodes use self-signed certificates maintained in their storages. If a message is invalid, the node broadcasts an error alarm message in the reverse direction. However, nodes may not be able to verify a message because it does not have the required certificate. Since nodes moves around the network, the large number of nodes that receives this message will have the required certificate. However, when a node receives an error alarm message or when it is the target node it must always verify the received message. In this case, if a node does not have the necessary certificate, it requests the certificate from other nodes. If node B needs node A ’s certificate, the node B broadcasts the the following message horizontally:

$$PK_request = [Type, Seq, A, Loc_B].$$

If a node has the requested certificate, it sends the following message to B :

$$PK_response = [Type, Seq, Cert_A].$$

3.4 Location Update

When a node moves a certain distance from the previous location or if a certain time has elapsed from the last update, the node broadcasts the following message vertically:

$$Loc_update = [Type, Seq, width, Loc_A].$$

Unlike PK_init messages, method 2 is always used. In other words, all nodes receiving this message will update the given location. If the width value is large, more nodes will preserve the location information which results in more nodes that can respond to location request messages. On the other hand, the cost of location update increases as the width value increases.

A node receiving an update message stores the message in its location table. An entry in a location table is maintained as follows:

$$[A, Cert_A, Loc_A, \Gamma_A],$$

where Γ_A denotes reliability of node A . When a node's reliability level falls below a certain level, the node is excluded from the network. The reliability threshold and adjustment of a node's reliability will be determined by the policy established prior to network deployment. All nodes receiving the location update message do not have to verify the validity of the message. We can use policies such as the followings.

- Policy 1. A message is verified every h hops.
- Policy 2. Every node randomly determines by itself whether it will verify the message or not.
- Policy 3. A node verifies a message only if it has the required certificate.

3.5 Location Request and Response

A node B broadcasts the following message horizontally to request the location of node A :

$$Loc_request = [Type, Seq, A, Loc_B].$$

When a node receives this message, it first looks for node A 's location information in its table. If the node has the requested information, it unicasts a location response message. A location response message from C in response to node B 's query about node A 's location is formed as follows:

$$Loc_response = [Type, Seq, Loc_B, Loc_A, Loc_C].$$

3.6 Error Alarm

When a node A receives a message, depending on the policy, it verifies the signature included in the message. If the message is invalid, it broadcasts the following message in the reverse direction:

$$Err_alarm = [msg, Sig_A(msg)],$$

where $msg = (Type, Seq, err_type)$. Nodes receiving this message must verify the validness of this message and perform necessary actions such as removing the previous location update and changing the reliability of a node.

4 Analysis

4.1 Security Analysis

Analysis of Public Key Registration. Our scheme uses a public key system that uses only self-signed certificates to provide secure location service. Obviously, in a normal environment, this kind of public key system cannot provide a safe environment, since it is difficult to prevent false registrations. However, as stated earlier, using an existing PKI in an ad hoc environment may not be a plausible solution. Therefore, it is inevitable that systems such as ours must be used in such environment. In our scheme, nodes register their self-signed certificates when they join the network. Since nodes cannot determine whether the received certificate is valid or not, they accept the registration unconditionally. In this case, we have to consider the outcome of the following attacks. In this discussion, we assume that the current request is a legitimate one.

- Attack 1. Someone else has already register a public key using the same ID as the current one.
- Attack 2. Someone else may later try to register a public key using the same ID as the current one.
- Attack 3. Someone may simultaneously send a registration message using the same ID as the current one in a different location.
- Attack 4. Someone swaps the public key in the current message with another one and forwards the altered message.

We assume that it is difficult for nodes to know in advance the IDs of participants that will join the network. If this assumption holds, then attack of type 1 and type 3 cannot occur. In our scheme, nodes reject duplicate registrations using the timestamp included in the certificate. As a result, nodes that have already accepted a registration for that ID in the past will reject this attack. However, there may be nodes that are receiving such registration for the first time. In this case, these nodes will accept this fraud registration. However, due to our grouping policy, there will be nodes in the current column who have accepted the legitimate registration in the past. These nodes will send an error alarm message which will cause nodes to reject the fraud registration. Our scheme also assume that each node monitors neighboring nodes' behavior by using techniques suggest in [14]. Therefore, attack of type 4 can also be detected with high probability.

Security against Attack Threat. If the PKI used in our protocol is secure, our new location service is robust against various attacks.

- False location update attack/False response attack: In our protocol, we use self-signed locations. Therefore, without acquiring the private key of a certain node, one cannot generate a false but valid self-signed location. As a result, these kinds of attacks cannot succeed.
- Replay: In our protocol, old replayed messages will be discarded using the timestamp included in that message.

- Blackmail attack: This kind of attack is related to false error alarm messages. In our protocol, nodes receiving an error alarm message will verify both the current message and the previous message that is reported to be invalid. Therefore, assuming that the PKI used in our protocol is secure, nodes can detect a false error alarm message.
- Blackhole/Wormhole attack: These kinds of attacks are not applicable to position-based routing protocols.

4.2 Efficiency Analysis

In our scheme, additional measures are used to provide a secure location service. Compared to the basic XYLS scheme, our scheme requires the following additional costs:

- public key registration cost,
- signature generation cost needed when constructing a self-signed location,
- signature verification cost, and
- public key query costs for signature verification.

The costs for signature creation and verification can be regarded as basic costs for secure communication when using a public key system. In other words, these costs are inevitable. Therefore, in this paper, we analyze the number of additional messages exchanged instead of analyzing the number of public key operations performed by a node. Compared to the basic XYLS scheme, additional messages used in our scheme are related to public key queries and error alarm messages. However, public key registration messages should not be regarded as additional messages. This is because nodes also report their locations during this registration. That is, public key registration is a special case of location update message. However, there is an obvious increase in the size of messages we use. If we assume majority of nodes are honest, the frequency of error alarm messages will be low. Therefore, in this analysis, we will focus on public key queries only.

In our scheme, a node maintains only a subset of self-signed certificates of other nodes. Therefore, when a message arrives, nodes may not be able to verify the validity of the received message. A node can always use a public key query to obtain the required certificate to verify the message. If a node maintains $x\%$ of entire nodes' certificate, then this node will request on the average $(1 - x)\%$ of messages it must verify. However, if nodes cache previous obtain certificates and they tend to only communicate with a subset of nodes, then this percentage will be lower than $(1 - x)\%$. Moreover, even if a public key query is required, this query will only require a single hop once the nodes are uniformly distributed from their initial locations. Therefore, the cost of public key queries will not effect the network performance.

We will show that this argument holds using a hypothetical ad hoc network of 200 nodes. The terrain of this network is assumed to be a square of 1km^2 and the communication radius of a node is assumed to be 200m. If a *PK_init* message is sent vertically using the *width* = 0 and method 1 is used, about $80(= 200\text{nodes} \times 400\text{m}/1\text{km})$ nodes will store the certificate. This is because approximately nodes residing in a column of 400m will receive this message. Let's assume the policy 3 given in section 3.4 is used and a certain amount of time has elapsed since the network was initially formed. This

means that nodes that maintain the same certificate are uniformly distributed throughout the network. In this case, the number of neighbors of a node is as follows:

$$\text{The number of neighbor nodes} = d \times r^2 \times \pi = (200/km^2) \times (0.2km)^2 \times \pi \approx 25,$$

where d is the node density of the network. Since 40% of nodes maintains the same certificate, the same percentage of neighbors, which is about $10(25 \times 0.4)$, will have the required certificate. Even if a node started at the edge of the terrain, about 40 nodes will have that node's certificate. In this case, 20% of the neighbors, which is about 5, will have the required certificate. If we assume that 50% of the nodes are honest, some node will always detect and report invalid messages. If method 2 is used and 6 segments are used, about $33(= 200/6)$ nodes will store the certificate. In this case, 17% of the neighbors, which is about 4, will have the required certificate.

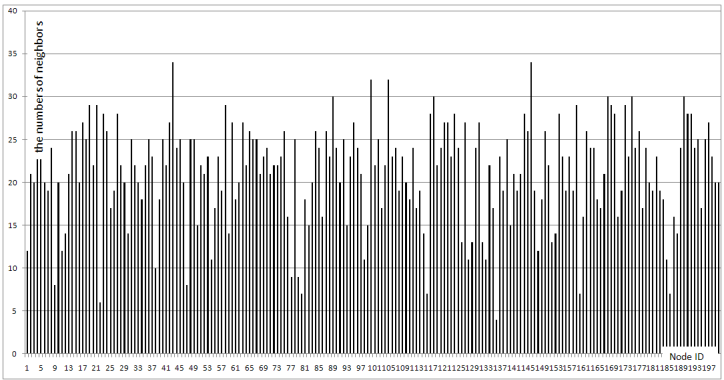


Fig. 2. The Number of Neighbor Nodes of Participating Nodes at Time $t = 100$

We ran a simulation to verify our above analysis. In this simulation, we assumed the same environment as used in above analysis. Moreover, we assumed that all 200 nodes participate from the start and we set the average movement speed of each node at 2.5m/sec and the maximum movement speed at 5m/sec. We ran the simulation for 180 seconds. We used method 2 with 6 segments which resulted in six distinct group and the size of each group was $A = 35, B = 30, C = 45, D = 29, E = 30,$ and $F = 31$. Fig 2, shows the number of neighbor nodes of each nodes at time $t = 100$ sec. The number of neighbors of a node ranged from 4 to 34 and the average was 22. We also observed the number of neighbors of a certain node numbered 0 which is given in Fig 3. We also observed the changes in neighbors of that node. This is also illustrated in Fig 3. As can be seen in the figure, during the simulation, except for group C at time 100sec to 120sec, there always exists a member from each group as the neighbor of node 0. If we think of the hops a message travels, it is reasonable to argue that there will always be an honest node that receives the message who has the required certificate.

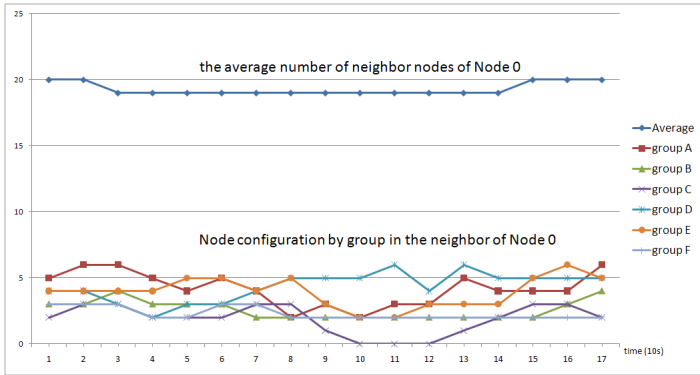


Fig. 3. Change of the Average Number of Neighbor Nodes and Change of Their Composition

5 Conclusion

In this paper, we proposed a new secure location service for ad hoc position-based routing. In our scheme, a node updates its location by sending its location digitally signed which we call a self-signed location. The use of this mechanism allows nodes to authenticate locations of others without relying on any trust on location servers. Our mechanism can be used in any position-based routing. However, the security of our mechanism depends on the PKI used in ad hoc network. Although, we have introduced an idea of using a public key announcement method, the security of our protocol can be enhanced further if a more efficient and secure way of deploying a PKI in ad hoc network can be devised.

References

1. Royer, E.M., Toh, C.: A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks. *IEEE Personal Communications* 2(6), 46–55 (1999)
2. Das, S.M., Pucha, H., Hu, Y.C.: Performance Comparison of Scalable Location Services for Geographic Ad hoc Routing. In: *Proc. of the IEEE INFOCOM 2005*, vol. 2, pp. 1228–1239. IEEE, Los Alamitos (2005)
3. Friedman, R., Kliot, G.: Location Services in Wireless Ad hoc and Hybrid Networks: A Survey. Tech. Rep. CS-2006-10. Haifa Univ. (2006)
4. Camp, T., Boleng, J., Wilcox, L.: Location Information Services in Mobile Ad hoc Networks. In: *Proc. of the IEEE Int. Conf. on Communications*, vol. 5, pp. 3318–3324. IEEE, Los Alamitos (2005)
5. Woo, S.C., Singh, S.: Scalable Routing in Ad hoc Networks. *Wireless Networks* 7(5), 513–529 (2001)
6. Cheng, C.T., Lemberg, H.L., Philip, S.J., van den Berg, E., Zhang, T.: SLALoM: A Scalable Location Management Scheme for Large Mobile Ad-hoc Networks. In: *Proc. of the IEEE Wireless Communications and Networking Conf.*, vol. 2, pp. 574–578. IEEE, Los Alamitos (2002)

7. Stojmenović, I., Peña, P.: A Scalable Quorum based Location Update Scheme for Routing in Ad hoc Wireless Networks. Tech. Rep. TR-99-09. Ottawa Univ. (1999)
8. Melamed, R., Keidar, I., Barel, Y.: Octopus: A Fault-Tolerant and Efficient Ad-hoc Routing Protocol. In: Proc. of the 24th IEEE Symp. on Reliable Distributed Systems, pp. 39–49. IEEE, Los Alamitos (2005)
9. Argyroudis, P.G., Mahony, D.O.: Secure Routing for Mobile Ad hoc Networks. In: IEEE Communications Surveys & Tutorials, vol. 73, pp. 2–27. IEEE, Los Alamitos (2005)
10. Hu, Y.C., Johnson, D.B., Perrig, A.: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks. In: Proc. of the IEEE Workshop on Mobile Computing Systems and Applications, pp. 3–13. IEEE, Los Alamitos (2002)
11. Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks. In: Proc. of the 8th ACM Int. Conf. on MobiCom, pp. 12–23. ACM, New York (2002)
12. Zapata, M.G.: Secure Ad hoc On-demand Distance Vector routing. In: ACM Mobile Computing and Communications Review, vol. 6(3), pp. 106–107. ACM, New York (2002)
13. Sanzgiri, K., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: A Secure Routing Protocol for Ad hoc Networks. In: Proc. of the 10th IEEE Int. Conf. on Network Protocols, pp. 78–87. IEEE, Los Alamitos (2002)
14. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: Proc. of the 6th ACM Int. Conf. on Mobile Computing and Networking, pp. 255–265. ACM, New York (2000)