SECURITY ENHANCEMENT OVER AD-HOC AODV ROUTING PROTOCOL

Zongwei Zhou
Department of Computer Science and Technology, Tsinghua University, Beijing, China
zhou-zw02@mails.tsinghua.edu.cn

ABSTRACT
For most existing routing protocols of mobile ad hoc network (MANET), more efficient security mechanisms against the attacks from malicious, compromised and selfish nodes are highly demanded. This paper proposes a series of security mechanisms for the Ad-hoc On-demand Distance Vector (AODV) Routing protocol. Three techniques, including digital signature, one-way hash function and double one-way hash verification are introduced to ensure the authentication, nonrepudiation and integrity of the important routing information in AODV protocol. The comparison with some existing secure AODV protocols demonstrates that our solution expands the security scope and guards against several attacks out of their range.

KEY WORDS
AODV Routing Protocol, Digital Signature, One-way Hash Function, Double One-way Hash Verification

1. Introduction

Originated from the DARPA PRNet [1] and SURAN program[2], mobile ad hoc networks (MANET) now becomes a hot topic in wireless network research. MANET is an infrastructureless, multi-hop network with dynamic topology. Nodes in ad hoc networks are constrained by power supply, computation ability and storage competence. Considering all these special features, designing an efficient and reliable routing protocol strategy for MANET is a big challenge.

Nowadays, numerous ad hoc routing protocols have been proposed and developed such as DSDV[3], OLSR[4], TBRPF[5], AODV[6], DSR[7] and ZRP[8]. Among them, Ad-hoc On-demand Distance Vector (AODV) is already identified as one of the major IETF standards for MANET routing[15]. However, AODV focuses on enhancing routing performance, but pays little attention to routing security, which means that it is vulnerable to numerous attacks from malicious, compromised and selfish nodes.

Researchers now have proposed some security mechanisms, such as ARAN[10], SAODV[11] and SRAODV[12], to build the protection, detection and reaction system for AODV against normal attacks and misbehaviours[16]. But, most of these mechanisms still cannot fully protect important routing information in AODV protocol. Many attacks, such as destination sequence number flood[11] and selfish increment of hop count[16] are still threatening AODV protocol. Facing such problems, this paper use digital signature, one-way hash function and a novel mechanism called double one-way Hash verification (DOHV) to enhance the security of AODV protocol. These mechanisms ensure the authentication, nonrepudiation and integrity of the important routing information in AODV protocol, preventing them from being forged or tampered.

The remainder of this paper is organized as follows: Section 2.1 reviews the AODV routing protocol and Section 2.2 analyses the treats towards AODV and its related security requirement. Some current and effective secure AODV protocols are introduced in Section 2.3. Section 3 details the security enhancement over AODV routing protocol. After that, Section 4 gives the treat analysis and security comparison between our solution and the secure AODV mechanisms described in Section 2.3. Conclusions and future works are in the last section.

2. Related Works

2.1 Brief Introduction of AODV Protocol

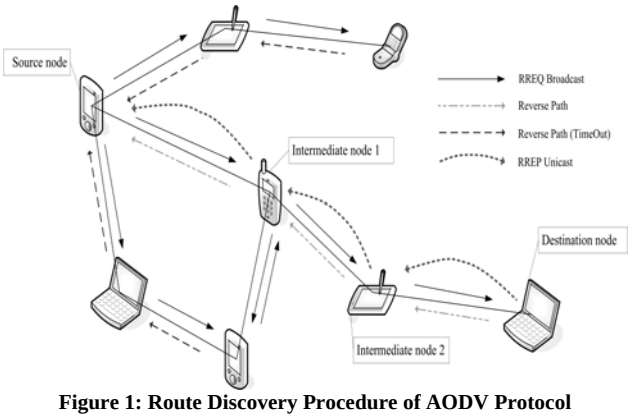

Figure 1: Route Discovery Procedure of AODV Protocol

AODV can be called as a pure on-demand routing protocol: routes are not built until certain nodes intend to communicate or transmit data with each other. And relevant routing information stores only in source node, destination node and intermediate nodes along the active route which deals with data transmission. AODV consists of two important stages: Route Discovery procedure and Route Maintenance procedure. Figure 1 shows all related operations in the Route Discovery procedure.

In order to initiate a Route Discovery procedure, a source node broadcast Route Request broadcast packets (RREQ) to all its accessible neighbours, similar with Dynamic Source Routing (DSR)[8]. The RREQ packet is of the following format:

<s_addr, id, s_seq, d_addr, d_seq, hop_count>

s_addr and *d_addr* denotes the IP address of source node and that of destination node, *id* is the broadcast ID, *s_seq* and *d_seq* represent the sequence number of source and destination node, *hop_count* is the number of nodes this message have passed.

Receiving RREQ, the intermediate nodes which have no route to the destination node would add *hop_count* by 1, rebroadcast this RREQ to its neighbours and set up a Reverse Path Pointer for the node from which it receives the RREQ. When the destination node receives RREQ, the active route is found. Then it would unicast a Route Reply packet (RREP) along the reverse path back to the source node. The RREP contains the following items:

<s_addr, d_addr, d_seq, hop_count, lifetime>

s_addr, *d_addr* and *d_seq* are directly copied from RREQ. *hop_count* is reset to zero and counted again. Every intermediate node will increase the *hop_count* by 1 and relay it according to its Reverse Path Pointer. As soon as the source node receives the correct RREP, the data transmission begins. Moreover, in order to speed up the Route Discovery procedure, AODV also allows the intermediate nodes which have the route to the targeted destination node to generate a RREP and send it back to source node. Note that only the nodes along the active route or Reverse Path store necessary information in their route tables and the other intermediate nodes will eliminate the routing information like Reverse Path pointer.

In Route Maintenance procedure, nodes keep an entry for each active route in their route table and periodically broadcast *hello* message to its neighbors in order to detect possible link failure. If a node detects a link failure, it would know that all active routes via this link would fail, so a Route Error message (RERR) is send to announce all relative source nodes. The source nodes then will decide whether to refresh the route or not. The RERR message contains the following items:

< d_addr, new d_seq, hop_count=∞ >

new d_seq is bigger than the maximum *d_seq* of all the RREQ or RREP this node have received. *hop_count* is set to an infinite number which means the destination node is now unreachable.

2.2 Treats Analysis and Security Requirement

From 2.1, we know that all the important information like *d_seq*, *hop_count*, *s_addr* and *d_addr* are not protected in original AODV protocol, so it is vulnerable to numerous attacks from malicious, compromised and selfish nodes. Therefore, AODV demands for special mechanisms to enhance its security.

As summary in[13], there are about six categories of important security services that should be provided in protocol for communication and data transmission: Authentication, Access Control, Data Confidentiality, Data Integrity, Nonrepudiation and Availability. We now analyze the security requirements for AODV under such definition.

Authentication is concerning with the authentic assurance between communication entities. There are three types of entities in AODV active route, including the source node, the destination node and intermediate nodes. In original AODV, A malicious or compromised node can easily impersonate the source node by forging a RREQ packet with its address, or pretend to be the destination node or route-aware intermediate node to relay RREP. Therefore, source and destination node authentication should be included in RREQ and RREP.

Instead of full data integrity, this paper will limit its scope on the integrity of the important routing information in AODV. Nodes use the *d_seq* to identify the most current information, suppress redundant routing packets and ensure loop-free routing. Meanwhile, the *hop_count* is responsible for critical routing selection and update. But attackers could simply reduce the *hop_count* to increase the chances of being in a certain active route, while a selfish node would try to increase it in the purpose of eliminating itself from certain route to save recourses. The attacker could also make a *big sequence number flood* attack by deliberately initiating a packet with a much bigger *d_seq*. This attack will wrongly update the sequence number of other nodes and finally prevent their responses to normal RREQ and RREP.

Moreover, nonrepudiation service is also useful in ad hoc network, as argued by Zhou and Haas in [14]. Once “erroneous message” is detected, this service would help tracing correctly and undeniably back to its originator and convincing other normal nodes that this originator is misbehaving. However, in AODV, any malicious node could forge a RERR message to tell other nodes that a certain node in the network is unreachable. This would greatly influence the flows via this “unreachable” node.

In addition, access control is not necessarily critical service, because host systems and applications are rare in the context of routing problem of ad hoc network. And, although availability is quite desirable and crucial, it does not seem to be feasible to prevent denial-of-service attacks in wireless network where the attacker can focus on the physical layer without bothering routing protocol. Even in the routing layer or upper levels, such mechanisms call for audition and detection system to take responsibilities [11]. So this paper does not count on solving such problems

2.3 Related Works on Securing AODV

Researchers now have proposed several protocols to secure the AODV protocol. K. Sanzgiri and B. Dahill have developed authenticated routing for ad hoc networks on AODV (ARAN)[10]. In ARAN, very node has its digital certificate signed by a trusted authority. ARAN uses a digital signature to provide authentication of all unaltered information in Route Request and Reply packet. Each node along the route should check the signature of its upstream node and replace it with its own signature. However, ARAN is not complete since it does not provide enough protection to *hop_count* information.

M. Zapata and N. Asokan also proposed a secure AODV protocol (SAODV)[11]. Similar digital signature protection as ARAN is used in SAODV and it further use one-way hash chain to secure the *hop_count* information from being decreased. This idea is borrowed from SEAD[9]. However, the one-way hash chain can not stop any attackers or selfish nodes from increasing the *hop_count* or just keeping it unchanged, as is described in Section 2.2.

Secure Routing with AODV (SRAODV), a series of security mechanisms, including Key Exchange, Secure Routing, Data Protection, are proposed by A. Pirzada and C. McDonald[12]. Considering about secure routing mechanism, the author recommended peer-to-peer symmetric encryption to all routing information in RREQ, RREP and RERR, using a group session key negotiated by neighbour nodes. However, this design requires each node to maintain a table along with associated group members and session keys. It would become less efficient as the number of nodes in ad hoc network increases. And moreover, a compromised node could still juggle *hop_count* or *d_seq* to interrupt the normal routing procedure.

Explicitly, abnormal modification of important routing information like *hop_count* and *d_seq* in RREQ, RREP and RERR messages of AODV can not be fully prevented by the above mechanisms we found. Some attacks and misbehaviours of network nodes are still threatening AODV. For example, both the big sequence number flood and the selfish increment of *hop_count* mentioned in Section 2.2 are not well handled by all the above mechanisms. This is important motivation of our security enhancement in this paper.

3. Security Solutions

The basic assumption in this solution is that there is a trusted certificate authorization and key distribution system in the MANET and every node in the network has a unique and safe public key pair and can acquire other nodes' public keys if needed. However, similar with all the security mechanisms in Section 2.3, the public key infrastructure in MANET is beyond our scope.

Three major mechanisms are introduced to secure the important routing information in RREQ, RREP and RERR packet. First, *digital signature* is used to authenticate some of the un-mutable fields of the above four messages, such as *s_addr, s_seq, lifetime* and so on. Secondly, *one-way hash chain* is applied to secure important routing information which should be updated in the packet transmission procedure, like *d_seq* and *hop_count*. Thirdly, Double One-way Hash Verification (DOHV) would ensure that intermediate nodes along the route could only follow AODV standard operation of *hop_count*. Both *hop_count* abnormal decrease and increment are not allowed.

Exception would appear when there are two or more than two malicious nodes performing a collusive attack to forge invalid packets. However, such collusive attack is difficult to withstand unless positive detection system or third-party authority joins in. So, similar with other secure AODV protocol in Section 2.3, this would be beyond the scope of this paper.

3.1 RREQ Protection

The RREQ is protected by *digital signature* and *one-way hash function*, which is similar with that of SAODV. And in our solution, we also extend *one-way hash function* to construct a new mechanism, Double One-way Hash Verification (DOHV), in order to prevent the abnormal *hop_count* operation of malicious node and selfish node.

The detail operations of one-way hash chains are listed as follows: The owner or initiator of *one-way hash chain* first chooses a finite positive integer *N* as the total length and a random number as the initial seed of this chain. Then, it selects an efficient and secure hash function *H*, such as MD5 and SHA-1. Let $H^{\sim i}(x)$ to be the result of applying the hash function H to the number x for i times, then the complete hash chain can be calculated as:

$$h_0 = a, h_1 = H(a), h_2 = H(h_1) = H^{\sim 2}(a), \cdots\cdots h_N = H^{\sim N}(a) \quad (1)$$

Following this procedure, when an intermediate node attains an element h_j and has no idea about the initial number a, it can only compute h_k ($N \geq k \geq j$) in ascendant sequence and is impossible to get h_k ($j > k \geq 0$), at least computationally.

Then, the protection of RREQ could be divided into three main steps:

Firstly, our solution uses one-way hash chain h^{seq} to protect *d_seq* field in RREQ. h^{seq} is generated by destination node following equation (1). And the hash value h^{seq}_{N-i} represents *d_seq* = *i*. N is the predefined maximum sequence number. This could be estimated by

common maximum number of nodes in ad hoc network. If the hash chain is out of use, N could be updated to a bigger number. Thus, when an intermediate node get h_{N-i}^{seq}, it can verify whether $H^{\sim i}(h_{N-i}^{seq})$ equals to h_N^{seq} or not. An attacker could only forge a smaller sequence number than *i*, but this would not do any help because packet with smaller sequence number would be discarded by downstream nodes.

Secondly, a digital signature is introduced to protect all the constant information in RREQ during transmission, including *s_addr, id, s_seq, d_addr, d_seq*. They are signed by source node's private key and the intermediate node can validate its authentication and integrity by public key. Thus no intermediate node is possible to juggle the protected information except that it already acquires the source node's private key.

Thirdly, double one-way hash verification (DOHV) mechanism is proposed to protect *hop_count.* Different from h^{seq} in the first step, h^{hop} is generated by the source node when a new RREQ is initiated. *N* would be an estimated number of maximum *hop_count* and h^{hop} is computed in descendant sequence as follows:

$$h_N = a, h_{N-1} = H(a), h_{N-2} = H(h_1) = H^{\sim 2}(a), \cdots\cdots h_0 = H^{\sim N}(a) \quad (2)$$

And h_{N-i}^{hop} represents *hop_count = i*.

DOHV means that a RREQ carries two hash values, one for the *hop_count* of the node from which the RREQ is received, the other for the *hop_count* of its upstream node. Assumes that node p receives a RREQ from node q and q receives it from r. *hop_count* in node r is *i*. So when p receives this RREQ, the DOHV item would be

$$(i, h_{N-i}^{hop}, timestamp_r)_{k_r^-}\ (i+1, h_{N-i-1}^{hop}, timestamp_q)_{k_q^-} \quad (3)$$

Two timestamps represent the time when node r and node q signed the above item. We can identify whether node r is actually the upstream node of q in this turn of RREQ propagation or not through time sequence and interval of two timestamps and the time when p receives the RREQ.

Therefore, RREQ in our solution will contain the following items:

$$<(s\_addr, id, s\_seq, d\_seq = j, h_{N_{seq}-j}^{seq}, h_0^{hop}, N_{hop})_{k_s^-}, (3)> \quad (4)$$

3.2 RREP Protection

In AODV, both the destination node and the intermediate nodes that has "fresh enough" route to destination node can generate RREP back to the source node. So, we design two types of securing RREP packet in the solution: dRREP for RREP generated by destination node and iRREP for intermediate node.

3.2.1 dRREP

By receiving RREQ, the destination node verifies the *hop_count* in the above equation (3) and use the source node's public key to check the authentication and integrity of unaltered information. If the whole RREQ is validated, the destination node then generates dRREP as follows:

$$<(s\_addr, d\_addr, d\_seq = j, h_{N_{seq}-j}^{seq}, i, h_{N_{hop}-i}^{hop}, lifetime)_{k_d^-}> \quad (5)$$

Unlike the original AODV, we do not set *hop_count*=0 and require every intermediate node along the route to increase this *hop_count* again, because in this secure solution, increasing *hop_count* by intermediate node again needs DOHV information. We believe that this would be unnecessary for common symmetric links in MANET. Along the reverse path, every intermediate node validates dRREP and makes sure d_seq is "fresh enough" and relays it without any change, which is more efficient than SAODV.

3.2.2 iRREP

Figure 2 show the *hop_count* verification procedure concerned with iRREP. The cache route is the route from which the intermediate node p knows how to get to the destination. In order to verify that the cache route in intermediate node p is correct and secure, p must add $(j, h_{N-j}^{hop}, timestamp_r)_{k_r^-}$ —part of the RREQ DOHV item in this cache route—in iRREP. Node p should also provide the total *hop_count i* from cache source node to destination node, which is included in the previous dRREP of this route. The *hop_count* between node p and destination node then can be calculated by $i - j - 1$.

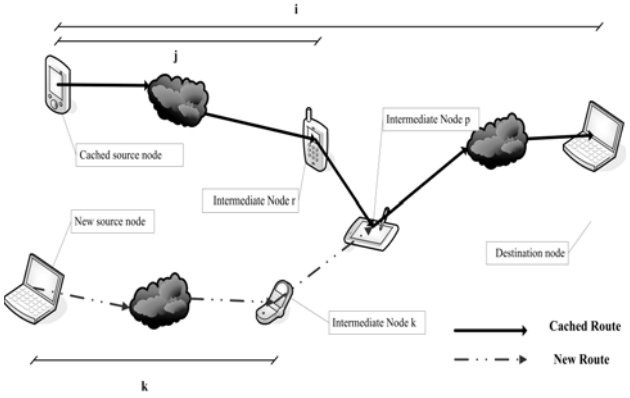

Figure 2: iRREP *hop_count* computation

Note that the *lifetime* field in cached dRREP would be stale and $(h_0^{hop}, N_{hop}^{'})$ for *hop_count* verification is different between cached source node and new source node, the iRREP contains a new lifetime field and $(h_0^{hop}, N_{hop}^{'})$ of cached source node, which is then signed by node p' private key.

For verification of the *hop_count* between new source node and node p, this iRREP should also include the new

RREQ information it received from node k, like the DOHV item $(k, h^{hop}_{N_{hop}-k}, timestamp_k)_{k_k^-}$. Therefore, the complete iRREP packet returned by node p would be as follows:

$$< (s\_addr, newlifetime, h_0^{hop}, N_{hop}^{'}),$$
$$(k, h^{hop}_{N_{hop}-k}, timestamp_k)_{k_k^-}, (j, h^{hop}_{N_{hop}-j}, timestamp_r)_{k_r^-}, \quad (6)$$
$$(s^{'}\_addr, d\_addr, d\_seq = x, h^{seq}_{N_{seq}-x}, lifetime, i, h^{hop}_{N_{hop}^{'}-i})_{k_d^-} >$$

As is shown in Figure 2, when the new source node receives the iRREP from intermediate node p, it can compute the *hop_count* of the new route by $k + (i - j)$.

3.3 RERR Protection

Protection for RERR message is comparatively simple. In our security solution, the intermediate node could not increase the d_seq, so we just require the reporters to use the most current d_seq they get. In order to authenticate the sender of RERR and help trace back to a possible malicious node who sends fabricated RERR to announce a normal route to be failed, the initiator is also required to sign the RRER with its own private key. Therefore, the secure RERR message contains such items as below:

$$< (\text{d\_addr, j, } h^{seq}_{N\text{-}j}, \text{hop\_count}=\infty)_{k_p^-} > \quad (7)$$

4. Security Analysis and Comparison

Our solution can protect AODV protocol form being compromised by the treats analyzed in Section 2.2. With the digital signature, no node can easily forge a RREQ to impersonate a source node. And any intermediate node can not edit the protected information without break up the integrity of RREQ and RREP. The destination node use one-way hash chain to generate *d_seq*, so big sequence number flood attack is prevented. Moreover, DOHV item in RREQ ensures that an intermediate node should follow the normal AODV procedure to increase the *hop_count* by 1. In addition, our security solution also prevents the attacks that a malicious node forge a RERR to announce a non existed link failure, or malicious intermediate node juggle the RERR to change the information of link failure. Owing to the digital signature, RERR initiator masquerade is impossible except for private key leakage.

Comparing with some existing secure protocol mentioned in Section 3, our solution provides more security services than those two current protocols. Details are given in Table 1, according to the analysis of important routing attack in [10] and [11]. Our solution can defend more routing information attack than ARAN[10], SAODV[11] and SRAODV[12], such as big sequence number flood, keeping the *hop_count* unchanged and selfish *hop_count* increment.

The main contribution of this paper is to introduce double one-way hash verification (DOHV) to prevent the malicious or selfish behavior on *hop_count*, such as keeping the *hop_count* unchanged and selfish *hop_count* increment. Every RREQ message should carry two hash values for *hop_count* (*hc*1 for the *hop_count* of the node from which the RREQ is received, *hc*2 for the *hop_count* of its upstream node). Every node that receives a RREQ could verify whether *hc*1equals to *hc*2+1. If not, the node would know that *hop_count* is not properly updated by previous nodes. A malicious node could forge its own *hop_count*, but it can not forge the *hop_count* of its upstream node, because of this *hop_count* needs to be signed by the upstream node. Moreover, the node could directly keep still *hc*1, sign *hc*2, and rebroadcast the RREQ, which is improper but invisible by the downstream node in original AODV protocol. However, in our solution, two timestamps are introduced to represent the time of the *hop_count* signature. If the malicious node keeps still *hc*1, the downstream node could identify that *hc*1 is faked because of the timestamp of *hc*1. The interval between this timestamp and RREQ receive time is too long. And we know that the timestamp of *hc*1 is digitally signed. It could not be forged or edited.

Another problem is why we do not eliminate the one-way hash protection of *hop_count* in RREQ to further reduce the security overhead, when DOHV already ensure that the *hop_count* cannot be increased by intermediate node.

Table 1: Routing attacks protection comparison between ARAN, SAODV, SRAODV and our solution

	ARAN	SAODV	SRAODV	Our Solution
Source node Impersonation	Yes	Yes	Yes	Yes
Destination node Impersonation	Yes	Yes	Yes	Yes
Big Sequence Number Flood	No	No	No	Yes
Reduce the hop count	No	Yes	No	Yes
Keep the hop count unchanged	No	No	No	Yes
Selfish hop count increment	No	No	No	Yes
RERR fabrication	Yes	Yes	Yes	Yes
Not forward routing packet	No	No	No	No
Collude attack	No	No	No	No

Because the one-way hash chains assure that *hop_count* is unique in every route discovery procedure. Without doing that, a malicious node can store previous DOHV items and reply it in new route discovery procedure or even the discovery procedure raised by the other source node, which may result in extremely bad route interruption.

However, the mechanism of one-way hash function and Double One-way Hash Verification (DOHV) could bring in extra overhead. To some extent, such overhead introduced by enhancing security is ineluctable owing to the tradeoff between performance and security. Under high security demand, the extra message length and processing time is necessary for protecting AODV from dangerous attacks.

5. Conclusions and Future Works

In this paper, a set of novel security mechanisms based on the Ad-hoc On-Demand Distance Vector Routing (AODV) are proposed. Three techniques, including digital signature, one-way hash function and double one-way hash verification ensure the authentication, nonrepudiation and integrity of important routing information in RREQ, RREP and RERR packet of AODV. When comparing to some current secure AODV protocols like ARAN, SAODV and SRAODV, our solution expands the security scope of them and provides more security service, such as protection towards big sequence number flood attack and selfish increment of *hop_count*.

However, there are still plenty of works for future research: First, the expansion of security service and competence of our solution does bring in more route update and maintenance overhead into the protocol. We need to simplify the DOHV structure in RREQ in order to reduce the overhead. Secondly, we could implement our solution onto computer simulation software platform or real MANET testing bed, and test the related packet delivery ratio, packet overhead, link throughput and average latency of our secure protocol.

Acknowledgements

The author would like to thanks Prof. Jun Li and Prof. Yibo Xue for their encouragement and advices. The author would also like to greatly thank Mr. Zhiming Zhang and Mr. Weirong Jiang for numerous enlightenment and helpful discussions.

References

[1] J. Jubin, J. Tarnow, The DARPA packet radio network protocols, *Proceedings of the IEEE, 75*(1): 21-32, January 1987.

[2] G. Lauer, Packet-radio routing. *In Routing in Communications Networks, edited by Martha E. SteenStrup* (Englewood Cliffs, NJ: Prentice-Hall, 1995).

[3] C. Perkins and P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, *ACM SIGCOMM Computer Communication Review*, *24*(4):234-244 Oct. 1994.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum & L. Viennot, Optimized link state routing protocol for ad hoc networks, *IEEE INMIC*, Pakistan, 2001.

[5] B. Bellur, R. Ogier, F. Templin, A reliable, efficient topology broadcast protocol for dynamic networks, *In Proc of INFOCOM '99*, New York, USA, 1999:178-186

[6] C. Perkins, E. Royer, Ad-hoc on-demand distance vector routing. *Proc of. 2nd IEEE Workshop. Mobile Computing. Systems and Applications (WMCSA 00)*, 1999, 90–100.

[7] D. Johnson, D. Maltz, Dynamic source routing in ad-hoc wireless networks, *Mobile Computing*, 1996, 153–81.

[8] Z. Haas, M. Pearlman, The Zone routing protocol (ZRP) for ad hoc networks. *IETF Internet draft*, 1998.

[9] Y. Hu, D. Johnson, & A. Perrig, SEAD: Secure efficient distance vector routing in mobile wireless ad hoc networks, *Proc. 4rth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 02)*, 2002, 3–13.

[10] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, & E. Belding-Royer, A secure routing protocol for ad hoc networks, *Proc. 10th IEEE International Conference of. Network Protocols (ICNP '02)*, 2002, 78–87.

[11] M. Zapata, N. Asokan, Securing ad hoc routing protocols, *Proc. ACM Workshop on Wireless Security (WiSe)*, 2002, 1–10.

[12] A. Pirzada, C. McDonald, Secure routing with the AODV protocol, P*roc. the Asia-Pacific Conference on Communications*, 2005, 57-61.

[13] W. Stallings, *Network security essentials: applications and standards, second edition* (Prentice Hall, 2002).

[14] L. Zhou, Z. Haas. Securing ad hoc networks. *IEEE Network Magazine, 13*(6):24–30, November, 1999.

[15] C. Perkins, E. Belding-Royer, & S. Das. Ad hoc on-demand distance vector (AODV) routing. *RFC3561*, 2003.

[16] S. Buchegger, J. Le Boudec, Cooperative routing in mobile ad-hoc networks: current efforts against malice and selfishness, *Lecture Notes on Informatics, Mobile Internet Workshop, Informatics 2002*, Dortmund, Germany, 2002.