SCIS 2010 The 2010 Symposium on Cryptography and Information Security
Takamatsu, Japan, Jan. 19-22, 2010
The Institute of Electronics, Information and Communication Engineers

Authenticated and DoS-resilient channel assignment mechanism for wireless mesh networks

Sungmok Shin * Junhyun Yim * Kwangjo Kim *

Abstract— *Hyancinth* is one of the popular node architectures proposed by *Ashish et al.* for efficient channel assignment in multi-radio multi-channel(MRMC) wireless mesh networks(WMNs). However, we found that *Hyacinth* was designed without considering the security mechanism. Fake channel control packet from external attacker can degrade the network goodput. Also, internal attacker *e.g.,* compromised node, can launch denial of service(DoS) attacks by two methods: malicious channel switching(MCS), and consecutive channel switching(CCS) attack. In this paper, we propose an authenticated and DoS-resilient channel assignment mechanism for MRMC WMNs, and provide secure channel control, and message verification module to prevent these attacks. Our experiments show that the proposed scheme can effectively prevents both internal and external attacks, thus maintaining stable channel assignment process.

Keywords: MRMC, WMN, Channel assignment

1 Introduction

Based on concept of MANETs, wireless mesh networks (WMNs) were invented[8]. WMNs are very similar to MANET in some points. However, WMNs have relatively static architecture compared to MANET. Also, minimum *partial Mesh* topology exists in WMNs to maintain constant bandwidth and throughput. Mobility of relaying nodes are also lower than MANETs. These characteristics of multi-hop relaying makes bandwidth usage of WMNs worse as the traffic increases[1]. This bandwidth problem is the serious difficulty of utilizing WMNs in metropolitan area.

However, IEEE 802.11 a/g wireless LAN interface is capable of utilizing wide area of spectrum[2]. IEEE 802.11 specification *a* and *g* provides 3 and 12 non-overlapped frequency channels, respectively. Therefore, bandwidth problem can be solved by using multiple channels[4]. On the other hand, assigning channel to each interface is not a simple problem. Let us think about mesh topology of multiple nodes that equips with three Network Interface Cards (NICs). Sender and receiver should use same channel with corresponding interface. Due to this characteristic, channel dependency problem happens during the channel assignment process. To solve this problem, *Hyacinth* is proposed to manage safe and stable channel assignment[3].

However, *Hyacinth* suffers from security vulnerabilities by malicious attacker. Two known attacks are explained in Section 2. In MCS (Malicious Channel Switching), a node which is compromised by malicious attacker falsely change its channel usage regardless of its optimal candidate channel. CCS (Consecutive Channel Switching) falsely broadcasts wrong CHNL_USAGE message which leads to WMNs into quasi-stable state. All of these attacks severely degrade the performance of WMNs.

In this paper, we propose secure channel assignment architecture that solves known problems. We make use of two feature of Hyancinth: First, *Hyacinth* nodes' upper interface is dependent on its parent nodes' down interface. Second, node that is close to each other used to have a similar interference range, which means that their channel usage would be similar. Using these two features, we provide Message verification module that is capable of checking the consistency of channel assignment. In channel usage gathering phase, each node maintains candidate preferred channel based on the collected CHNL_USAGE message from neighboring nodes in its interference range. Then in *discrepancy checking* phase, node checks its interface and channel usage with received CHNL_USAGE message. Finally in *message verification* phase, consistency of CHNL_CHANGE message is checked by comparing new channel in the CHNL_USAGE message with candidate preferred channel. If candidate preferred channel and received channel matches, POS_ACK is sent to PARENT node to allow channel switching. If it does not match, NEG_ACK is send and channel switching is denied. In this way, we can prevent malicious node from propagating fabricated CHNL_USAGE and CHNL_CHANGE message.

2 Background and related work

Here, we explain the channel assignment mechanism of *Hyacinth* and observe possible attacks that disrupt

* KAIST-ICC, Munji Campus 103-6 Munji-dong Yusong-ku, Daejeon, 305-714, KOREA, (cabin15, junhyunv, kkj@kaist.ac.kr)

the channel assignment process in WMNs.

2.1 *Hyacinth*

2.1.1 Distributed channel assignment algorithm

Hyacinth proposes distributed channel assignment algorithm that takes advantage of load information based on local traffic. Local traffic load information is propagated among nodes within its interference range using CHNL_USAGE message. Based on the information from received CHNL_USAGE message, each node makes their decision whether to change their interface to less loaded channel or fix it to old channel as it was. If node finds relatively less loaded channel in its interference range, it changes one of its interface to that less loaded channel, and sends CHNL_CHANGE message to its child node for changing its channel as well. Fig.1 describes the interface structure of *Hyacinth* node. *Hyacinth* separates each nodes' interface into two NICSs, UP-NICs and DOWN-NICs.

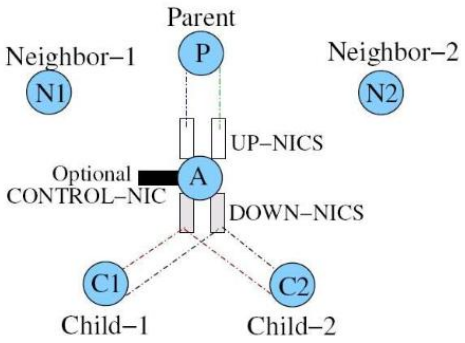


Figure 1: Interface structure of *Hyacinth* node

UP-NICs of each node are filled with channel used by its parent nodes' DOWN-NICs. Thus, child node does not have to consider the channel assignment of its UP-NICs. Other than UP-NICs, each node only has to consider about channel assignment of its DOWN-NICs. In CHNL_CHANGE message, interface ID, current channel and new channel is specified. Also in CHNL_US AGE message, node ID, Interface ID, current channel, hop count and its bandwidth usage is specified.

2.2 Denial of Service(DoS) vulnerabilities of channel assignment algorithm in *Hyacinth*

2.2.1 Attacker model

We consider an omnipresent but computationally bounded adversary. She controls the communication channel in the sense that she is able to eavesdrop, insert, modify, and block arbitrary messages by adding her own signal to the channel (*e.g.*, in order to jam the signal). We distinguish two attacker models: internal and external. In the external attacker model, we assume that none of the nodes involved in the protocol are compromised. Thus, an external attacker cannot authenticate herself as an honest network node to other network nodes or to the central authority. An internal attacker, however, controls one or more network nodes. We assume that when a node is compromised, its secret keys are known to the attacker. Subsequently, compromised nodes can authenticate themselves as legitimate nodes to the authority and to other network nodes. A non-compromised node can also misbehave because of non-malicious corruptive processes such as software, hardware, or system faults. We classify these nodes likewise as internal attackers. As we will show, our protocols are indifferent to the cause of misbehavior.

2.2.2 Flooding fake channel control message

A channel assignment algorithm in *Hyacinth* does not provide any means of secure transmission of channel control message, which means this algorithm is no designed with security in mind. In this way, if any external attacker broadcasts false channel control message into WMNs, any node accept, and change their channel status as received message. If any external attacker *e.g.,* laptop class attacker, continuously broadcast same CHNL_CHANGE message to neighboring nodes, making every nodes use same channel, entire WMNs fall into denial of service status.

2.2.3 Malicious channel switching (MCS)

The compromised node is able to change its down interface. Each nodes' upper interface is conformed to its PARENT nodes' down interface. Therefore, if compromised node deliberately change its down interface with heavily loaded channel. This results bandwidth problem of entire WMNs. Node M is the node compromised by a malicious attacker. Attack sequence of MCS is as follows :

1. Node M receives CHNL_USAGE message from neighboring nodes in its interference range.
2. Node M finds out that channel k and l is relatively much loaded than other channel.
3. Node M switches its communication channel to channel k, l, respectively.
4. Node M then transmits CHNL_CHANGE message to neighboring nodes.
5. All link that locates comes before node M suffers from performance degradation due to heavily loaded channel

2.2.4 Consecutive channel switching (CCS)

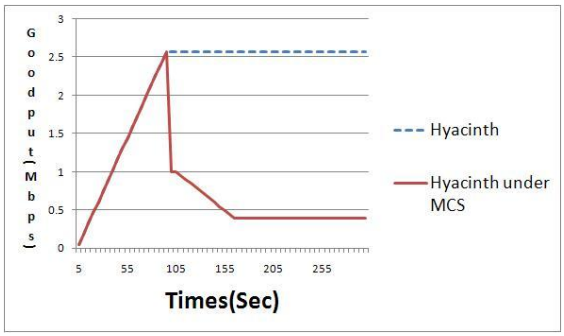
The purpose of CCS attack is to put WMNs into a quasi-stable state. Quasi-stable state means that each node are forced to change their channel so frequently that networks can not support stable bandwidth to user. Attack sequence of CCS is as follows:

1. Node M receives CHNL_USAGE message from neighboring nodes in its interference range.

2. Node M randomly selects one of the channel in the channel list that are normally loaded.

3. Node M then constructs CHNL_USAGE message with selected channel assignments and broadcasts it the neighboring nodes.

Unlike MCS attack, CCS does not deliberately change any channel assignment of its DOWN-NICs. Instead, it just selects middle priority channels, and propagates CHNL_USAGE which results in propagation of change upwards in the routing tree. Heavily loaded channels are not selected because such selection will affect the links closer to gateway resulting in quick adjustment to the change and hence no ripple effect will be created. We first measure bandwidth usage under normal multi-radio multi-channel WMNs. Attack simulation is performed using NS2 simulator. First, we measure the bandwidth usage under attacks MCS and CCS, respectively. Under CCS, bandwidth shows unstable state due to frequent channel switching. Fig.2 shows simulation results of these attacks.



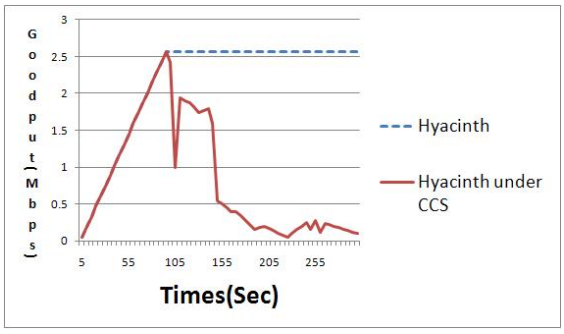


Figure 2: Simulation result of *Hyacinth* under MCS and CCS attack

3 Our proposed scheme

In this Section, we propose secure channel control and message verification module to prevent attacks mentioned in Section 2. First, secure channel control provides authenticated channel assignment process between each node, thus preventing external attacker from propagating fabricated channel control message. Second, message verification module prevents internal attacker from disabling the stable channel assignment process.

3.1 Network model

A wireless mesh network can be modeled as a connected graph $G = (V,E)$, where V is the set of N mesh nodes and $E \subset V \times V$ is the set of wireless links. We assume that each node uses omni-directional antennas and all wireless links are bi-directional[7]. A wireless link exists between nodes i and j if the distance between the two nodes, $d_{i,j}$, is smaller than R_t, where R_t is a fixed transmission range. For simplicity, we assume that a transceiver has the same receiving and transmission range. Thus, in our context, each edge $(i, j) \in E$ represents an undirected edge of the graph G. Let the set of channels supported by the 802.11 spectrum be denoted as K, where $K = 1, 2, \ldots, k$, and the number of radios on each node as $M_i \leq |K|$, $\forall i \in V$. We assume that all channels are orthogonal, so the interference exists between two links if they are within interference range and are assigned the same channel. We believe that our model can be easily extended to account for non-orthogonal channels. To model the interference we consider a conflict graph $G_c = (V_c, I)$, where $V_c = E$ and $I \subset E \times E$. Two links (i, j) and (u, v) interfere with each other if they operate on the same channel and any of the quantities d_{ui}, $d_{v,i}$, $d_{u,j}$, $d_{v,j}$ is smaller than sR_i, where R_i denotes the fixed interference range. Let $I_{i,j} \subset I$, $\forall (i, j) \subset E$, denote the set of all links in the network within the interference range of link (i, j). Let L be the load matrix of the network. Thus, $L_{i,j}$ is the expected traffic on link (i, j). This flow estimate of network traffic can be obtained using tools like the CoMo project. We also make the following assumptions while modeling the channel assignment problem in wireless mesh networks.

- The traffic flow on the network is relatively stable over a period of time and is easy to predict. This is a fairly reasonable assumption for enterprise networks which are designed for balanced network flows.

- Nodes are generally static. This ensures no major topology changes during the course of channel assignment.

3.2 Assumption

Our proposed scheme modifies *Hyacinth* to protect channel assignment procedure from internal and external attacker. We assume each node has a capability of using multiple interface and radio. Our network consists of a set of mesh nodes, which communicate using both radio and wire transmissions. We assume that the radio link between neighboring devices is bidirectional. The network is operated by an authority. The authority controls the network membership and assigns a uniue identity to each node. Each pair of nodes holds a shared secret key that can either be manually preloaded into the nodes during the deployment phase or can be generated during the network setup phase using key establishment protocols(*e.g.,* Perrig *et al*[9];Eschenauer and Gligor[10]).

3.3 Secure channel control

We initially propose secure channel control scheme to prevent false channel control message of external attacker. We use cryptographic measure to authenticate the received control message. This scheme is designed to be run by two nodes that reside within each others' communication ranges. Table 1 described each steps of secure channel control.

Table 1: Secure channel control

Secure channel control(SCS)
1. A → B : $ENC_{K_{AB}}$(CHNL_USAGE, $MAC_{K_{AB}}(CHNL\_USAGE)$)
2. B : Decrypt received message using K_{AB}
3. B : Verify $MAC_{K_{AB}}(CHNL\_USAGE)$
4. B → A : If verified, ACK
or, NEG_ACK

In this protocol, integrity and authenticity of channel control message are ensured through use of Message Authentication Codes(MAC) and of a key K_{AB} shared between node *A, B*. This prevents external attackers from modifying values in the channel control message or in the acknowledgement packet, without being detected. Furthermore, the attacker cannot impersonate node B as she does not know the secret key K_{AB}.

3.4 Message verification module

We modified *Hyacinth* to prevent known attacks explained in the previous Section. To prevent those attacks, we need additional module to verify two information message, CHNL_USAGE message and CHNL_CH ANGE message. In our scheme, we call this module as, message verification module. This module consists of three phase. Followings are the key idea of message verification module :

A. Neighboring nodes retains similar interference range : In *Hyacinth*, each node shares channel usage status with its (k+1) neighboring nodes. In other words, node that is close to each other used to have similar interference range. Therefore, each node can maintain similar channel usage information. This can help each node to independently judge whether the received channel usage information is correct or not based on other CHNL_USAGE information from neighboring nodes. In our proposed scheme, each node always maintain 'candidate channel' to compare with CHNL_CHANGE message from PARENT node.

B. CHILD node's UP-NIC is dependent on the PARENT node's DOWN-NICs : To prevent the channel dependency problem of channel assignment, each node's UP-NIC is restricted to its PARENT node's DOWN-NIC. In this way, each node only concentrates on assigning channel to its DOWN-NICs. Therefore, channel dependency problem can be prevented. This feature can be used efficiently to judge CHNL_USAGE information from PARENT node. For example, if one PARENT node is compromised and propagates falsely modified CHNL_USAGE message to its CHILD node, CHILD node can compare its own CHNL_USAGE with received message. Therefore, verification mechanism can be constructed.

Based on above concept, channel control message can be filtered through our scheme. Fig.3 describes structure of message verification module.

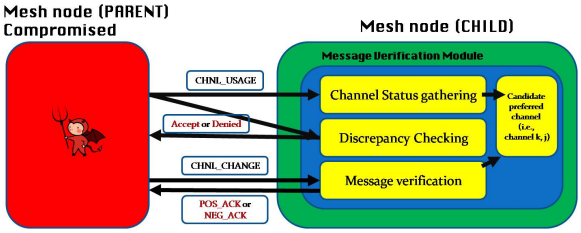


Figure 3: Message verification module

Channel status gathering phase : In *channel status gathering phase*, we decide candidate preferred channel by using CHNL_USAGE message received from neighboring in the interference domains of each nodes. Since CHNL_USAGE message is propagated to each node's (k+1) hop range, single node receives multiple CHNL_USAGE message from neighboring nodes. Then, it sorts each message's interface list in increasing order of bandwidth usage. Top list of each message indicates the least loaded channel. Checking consistency of above value, we can select which channel is the least loaded channel in its interference range even if there are malicious nodes broadcasting false CHNL_USAGE message. Pseudo code is described in Table 2.

Table 2: *Channel status gathering* phase

Algorithm 1 : Channel status gathering
collecting n CHNL_USAGE from neighboring nodes for every CHNL_USAGE message from 1th to nth
for every channel list in CHNL_USAGE
sort the list in increasing order depending on bandwidth usage
select two topmost least loaded channel
end for
mark the frequency of two topmost crowded channel
end for
select the two most frequently marked channel as a candidate preferred channel

Discrepancy checking : As described above, each node always maintain a n number of candidate preferred channel (n is a number of NICs). In this phase,

we check the discrepancy between each DOWN-NICs and UP-NICs. Discrepancy means that the CHILD node's channel which is assigned to one interface doesn't corresponds to the PARENT node's channel and interface. We covered the concept that CHILD node's UP-NICs are restricted to PARENT node's DOWN-NICs. This feature can used to prevent the CCS attack that forced ripple effect to neighboring node. Let's say that one node receives fabricated message that triggers CCS attack as shown below. Pseudo code is described in Table 3.

Table 3: *Discrepancy checking* phase

Algorithm 2 : Discrepancy checking
Receiving CHNL_USAGE message from its PARENT node
compare channel usage of node's UP-NICs with the one in
CHNL_USAGE message from PARENT node
if the channel of PARENT node's interface matches with channel of
CHILD node's corresponding interface
CHNL_USAGE message is accepted
else
CHNL_USAGE message is discarded
end for

Message verification : False CHNL_USAGE message can be filtered using second phase by checking consistency of interface and channel between PARENT node and CHILD node. However, channel change process is different from CHNL_USAGE message propagation. In other words, channel change happens before the PARENT node sends CHNL_USAGE message to its CHILD node. Therefore, it is not possible for current *Hyacinth* architecture to prevent MCS attack, in that compromised node deliberately changes its DOWN-NICs to heavily loaded channel. Thus, we modified current *Hyacinth* to send CHNL_USAGE message along with CHNL_CHANGE message. Previous channel switching is performed without the agreement of CHILD node. In our model, we transmit the CHNL_USAGE and CHNL_CHANGE simultaneously to compare the consistency of channel switching. If CHILD node's candidate preferred channel is consistent with PARENT node's switching channel, channel switching is performed. However, if it is not consistent, channel switching is denied. Pseudo code is described in Table 4.

Table 4: *Message verification* phase

Algorithm 3 : Message verification
Receiving CHNL_CHANGE message from its PARENT node
compare requested channel with node's candidate preferred channel
if
requested channel matches candidate preferred channel
channel switching is performed
else
channel switching is denied
end if

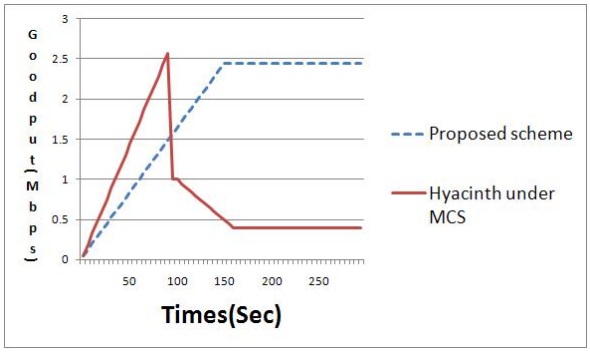
4 Security and performance analysis

We conducted performance evaluation using NS-2 simulator[5]. We implemented *Hyacinth* architecture and add security mechanism at the network layer protocol stack.

4.1 Performance analysis

We first measure the goodput of network when there is no attack. Then, *Hyacinth* under each attack is simulated. Finally, *Hyacinth* with security mechanism under each attack is simulated to measure the network goodput. Fig.4 shows the simulation result under three circumstances. Result shows that goodput under each attack severely degrades the performance of WMNs. However, our security mechanism can protect those attack and preserve the stable goodput of WMNs. Simulation parameter is as follows:

- Node / topology : 15 node / grid layout
- Interference range : Two hop range
- Flow type : Two Constant Bit Rate (CBR)
- Flow rate : 5Mbps



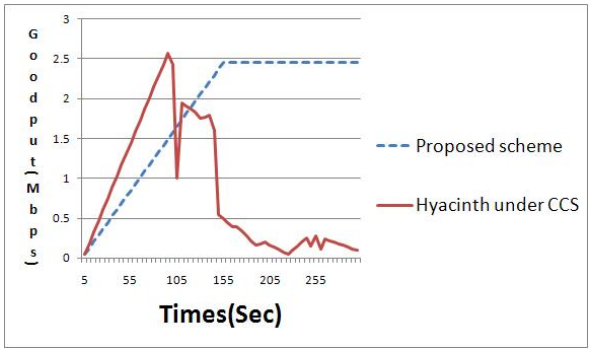


Figure 4: Simulation result of our proposed scheme

4.2 Security analysis

Confidentiality The CHNL_USAGE and CHNL_CHANGE messages are valuable to external attacker, investigating which channel the target node is using. If any specific nodes takes charge of significant role in WMNs, disabling only those nodes would put entire WMNs into poor performance. Thus, protecting these channel related information is important task. Proposed scheme encrypts CHNL_USAGE and CHNL_CHANGE message transmitting between each nodes, thus preventing external attacker from collecting these messages.

Availability The attacks mentioned in Section 2 significantly degrades the performance of WMNs. As the attacks continuously happen, whole WMNs goes into DoS status. Those attacks is launched by compromised nodes, thus making it difficult to protect by cryptographic measure. Using message verification module, each nodes filter the receiving control message from the neighboring nodes. Based on the inherent character of *Hyacinth*, receiver can judge if the CHNL_USAGE message originates from the legitimate nodes or not. Moreover, *candidate preferred channel* is periodically maintained on each nodes to judge if the demanding channel in the received CHNL_CHANGE message is proper or not. Even if the compromised nodes exist among the neighboring nodes, false channel control message is filtered through proposed filtering mechanism.

5 Conclusion

In this paper, we explain the security vulnerabilities of channel assignment architecture, *Hyacinth*. If malicious attacker compromises any node in WMNs, he or she is able to control node's function. This attacker can transmit fabricated CHNL_USAGE or CHNL_CHANGE message to neighboring nodes. Compromised node then can deliberately change its DOWN-NICs with heavily loaded channel, so that bandwidth usage over entire WMNs are degraded. Moreover, propagating wrong CHNL_USAGE message triggers further channel switching to reverse direction of spanning tree of nodes. We define these attacks as MCS and CCS.

Thus, we modified current *Hyacinth* architecture to prevent malicious attacks. We add additional module called, Message verification module. This module consists of three phase, *Channel status gathering*, *Discrepancy checking* and *Message verification*. First, in *channel status gathering*, node collects CHNL_USAGE message from neighboring nodes and selects the candidate preferred channel. Second, in *discrepancy checking* phase, node check the discrepancy of its channel and interface from requested channel and interface. Third, in *message verification* phase, node compares requested switching channel from PARENT node with its candidate preferred channel to verify the integrity of CHNL_CHANGE message. Our architecture can protect channel assignment algorithm from MCS and CCS attacks.

References

[1] Krishna N. Ramachandran, Elizabeth M. Belding, Kevin C. Almeroth, Milind M. Buddhikot, "Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks," *In Proceedings of IEEE Infocom'06*, 2006.

[2] Ashish Raniwala, Kartik Gopalan, Tzicker Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networs," *In ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 2004.

[3] Ashish Raniwala, Tzi-cker Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network," *In proceedings of IEEE InfoCom'05*, 2005.

[4] Murali Kodialam, Thyaga Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," *In proceedings of Mobile Computing and Networking*, 2005.

[5] S. Kurkowski, T. Camp and M. Colagrosso, "MANET Simulation Studies: The Incredibles," *ACM SIGMOBILE Mobile Computing and Communication Review, Vol. 9, Issue 4*, 2005.

[6] Corson, S. and J. Macker, *Book Mobile ad hoc networking(MANET)*, IETFRFC 2501, 1999.

[7] S. Bellofiore, J. Foutz, R. Govindaradjula, I. Bahceci, C.A. Balanis, A.S. Spanias, J.M. Capone, T.M. Duman, "Smart antenna system analysis, integration and performance for mobile ad hoc networks (MANETs)" *IEEE Transactions on Antennas and Propagation*, 2002.

[8] Ian F. Akyildiz , Xudong Wang , Weilin Wang, "Wireless Mesh Networks: a Survey," *Computer Networks and ISDN Systems*, 2005.

[9] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, David E. Culler, "SPINS: Security Protocols for Sensor Networks," *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, 2001.

[10] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks," *In 9th ACM Conference on Computer and Communication Security*, 2002.