ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

Mitigation of Vampire Attacks in Wireless Sensor Networks

M.Nageswara Prasadhu[1], D.V.Bharath[2] and G.Lakshmikanth[3]

[1]PG Student, Dept. of CSE, Sree Rama Engineering College, Tirupathi, Andhra Pradesh, India

[2]Assistant Professor, Dept. of CSE, Sree Rama Engineering College, Tirupathi, Andhra Pradesh, India

[3]Associate Professor & HOD, Dept. of CSE, Sree Rama Engineering College, Tirupathi, Andhra Pradesh, India

ABSTRACT: Ad-hoc sensor network and routing data in them is a most significant research area. There are lots of protocols established to protect from DOS attack, but it is not perfectly possible. One such DOS attack is Vampire attack. This vampire attack is a resource depletion attacks at the routing protocol layer, which permanently disconnect the networks by quickly draining nodes' battery power. These "Vampire" attacks are not specific to any specific protocol, but rather depend on the characteristics of many popular classes of routing protocols. This project illustrates a technique to tolerate the attack by employing the Cluster Head. In case of each Vampire attack, the Cluster Head employs in this situation and distributes the packet to destination without dropping the packet. Thus give a successful and reliable message delivery even in case of Vampire attack. In the worst case, a single Vampire can increase network wide energy usage by a factor of O(N), where N is the number of network nodes.

KEYWORDS: PGLP**,** GPSR, Carousel attack.

I. INTRODUCTION

Wireless sensor Network (WSN) consists of mostly tiny, resource-constraint, simple sensor nodes, which communicate wirelessly and form ad hoc networks in order to perform some specific operation. Due to distributed nature of these networks and their deployment in remote areas, these networks are vulnerable to numerous security threats that can adversely affect their proper functioning. Simplicity in WSN with resource constrained nodes makes them very much vulnerable to variety of attacks. The attackers can eavesdrop on its communication channel, inject bits in the channel, replay previously stored packets and much more. The extreme resource limitations of sensor devices pose considerable challenges to resource-hungry security mechanisms. The hardware constraints necessitate extremely efficient security algorithms in terms of bandwidth, computational complexity, and memory. This is no trivial task. Energy is the most precious resource for sensor networks. Communication is especially expensive in terms of power. The communications links available for coordinating their attack. The sensor nodes may not be tamper resistant and if an adversary compromises a node, it can extract all key material, data, and code stored on that node. As a result, WSN has to face multiple threats that may easily hinder its functionality and nullify the benefits of using its services.

An adversary can easily retrieve valuable data from the transmitted packet that are sent (Eavesdropping). That adversary can also simply intercept and modify the packets content meant for the base station or intermediate nodes (Message Modification), or retransmit the contents of those packets at a later time (Message Replay). Finally, the attacker can send out false data into the network, maybe masquerading as one of the sensors, with the objectives of corrupting the collected sensors reading or disrupting the internal control data (Message Injection). Securing the WSN needs to make the network support all security properties: confidentiality, integrity, authenticity and availability.

Attackers may deploy a few malicious nodes with similar or more hardware capabilities as the legitimate nodes that might collude to attack the system cooperatively. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. Also, in some cases colluding nodes might have high-quality communications links available for coordinating their attack. The sensor nodes may not be tamper resistant and if an adversary compromises a node, it can extract all key material, data, and code stored on that node. As a result, WSN has to face multiple threats that may easily hinder its functionality and nullify the benefits of using its services.

Routing and data forwarding is a crucial service for enabling communication in sensor networks. Unfortunately, current routing protocols suffer from many security vulnerabilities. For example, an attacker might launch denial of-service attacks on the routing protocol, preventing communication. The simplest attacks involve injecting malicious routing information into the network, resulting in routing inconsistencies. Simple authentication might guard against injection attacks, but some routing protocols are susceptible to replay by the attacker of legitimate routing messages. The wireless medium is inherently less secure because its broadcast nature makes eavesdropping simple. Any transmission can easily be intercepted, altered, or replayed by an adversary. The wireless medium allows an attacker to easily intercept valid packets and easily inject malicious ones. Although this problem is not unique to sensor networks, traditional solutions must be adapted to efficiently execute on sensor networks.

II. OVERVIEW

The extreme resource limitations of sensor devices pose considerable challenges to resource-hungry security mechanisms. The hardware constraints necessitate extremely efficient security algorithms in terms of bandwidth, computational complexity, and memory. This is no trivial task. Energy is the most precious resource for sensor networks. Communication is especially expensive in terms of power. Clearly, security mechanisms must give special effort to be communication efficient in order to be energy efficient. The proposed scale of sensor networks poses a significant challenge for security mechanisms. Simply networking tens to hundreds of thousands of nodes has proven to be a substantial task. Providing security over such a network is equally challenging. Security mechanisms must be scalable to very large networks while maintaining high computation and communication efficiency.

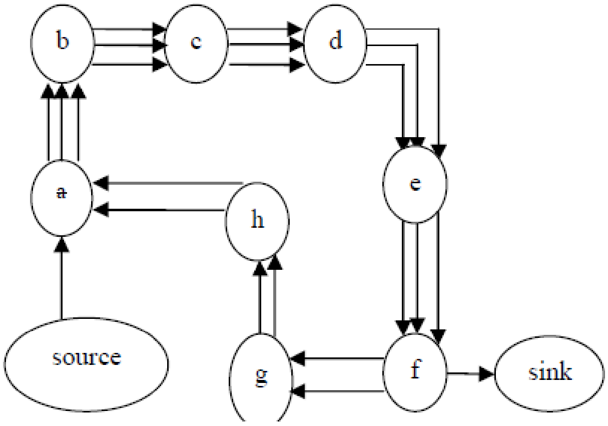

Fig. 1: An honest node would exit the loop immediately from node, but a malicious packet makes its way around the loop twice more before exiting.

Depending on the function of the particular sensor network, the sensor nodes may be left unattended for long periods of time. In our first attack, an adversary composes packets with purposely introduced routing loops. We call it the carousel attack, since it sends packets in circles as shown in Fig. 1. It targets source routing protocols by exploiting the limited verification of message headers at forwarding nodes, allowing a single packet to repeatedly traverse the same set of nodes. In our second attack, also targeting source routing, an adversary constructs artificially long routes, potentially traversing every node in the network. We call this the stretch attack, since it increases packet path lengths, causing packets to be processed by a number of nodes that is independent of hop count along the shortest path between the adversary and packet destination.

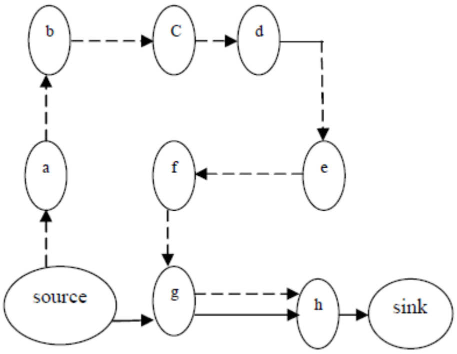

Fig. 2: Honest route is dotted while malicious route is dashed. The last link to the sink is shared.

An example is illustrated in Fig.2. Results show that in a randomly generated topology, a single attacker can use a carousel attack to increase energy consumption by as much as a factor of 4, while stretch attacks increase energy usage by up to an order of magnitude, depending on the position of the malicious node. The impact of these attacks can be further increased by combining them, increasing the number of adversarial nodes in the network, or simply sending more packets. Although in networks that do not employ authentication or only use end-to-end authentication, adversaries are free to replace routes in any overhead packets, we assume that only messages originated by adversaries may have maliciously composed routes.

III. ENERGY DRAINIING ATTACKS ON STATELESS AND STATEFUL PROTOCOL

In the DSR[9] source node specifies the entire route in the packet header to a destination, so intermediate node's do not make independent forwarding decisions, instead of a route specified by the source. To forward a message, the intermediate node finds itself in the route and transmits the message to the next hop. The fardel is on the source to ensure that the route is valid at the time of sending, and that every node in the route is a physical neighbour of the previous route hop. Both the carousel and stretch attacks are evaluated in a randomly generated 30- node topology. It causes delay as well as increase communication overhead and energy consumption in resource limited networks. The effect of denial or degradation of service on battery life and other finite node resources has not generally been a considered securely.

1) Carousel attack: In this attack, a malicious node forward a packet with a route included a chain of loops, such that the packets traverse several times in the same route. This strategy can be used to increase the route length beyond the number of nodes in the network An example of this type of route is in Fig.3the thick path shows the honest path and thin shows the malicious path.

2) Stretch attack: Another attack in the same layer is the stretch attack, where a malicious node constructs falsely long source routes, causing packets to traverse a longer than optimal number of nodes. In this example given below honest path shown with thick lines and adversary or malicious path with thin lines. The honest path is very less distant but the malicious path is very long to make more energy consumption. Per-node energy usage under both attacks is shown in Fig.3.2. As expected, the carousel attack causes excessive energy usage for a few nodes, since only nodes along a shorter path are affected.

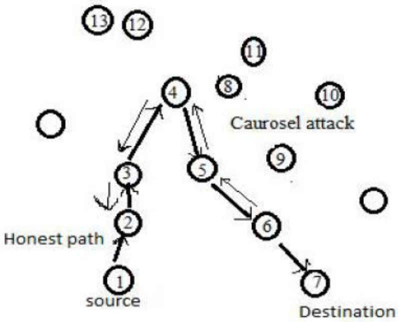

Fig. 3.1 shows the caurosel attack same node appears in the route many times.

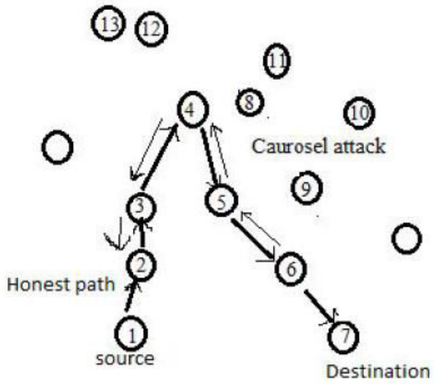

Fig. 3.2 Shows Stretch attack with two different paths from source to destination.(4-9-10-11-12-8-9—long route)

In contrast, the stretch attack shows more uniform energy consumption for all nodes in the network, since it lengthens the route, causing more nodes to process the packet. While both attacks drastically network-wide energy usage, individual nodes are also noticeably affected, with some losing almost 10 percent of their total energy reserve per message.

Two important classes of stateful protocols are link state and distance-vector. In link-state protocols, such as OLSR [2], nodes keep a record of the up-or-down state of links in the network, and flood routing updates every time a link goes down or a new link is enabled. Distance vector protocols like DSDV [11] keep track of the next hop to every destination, indexed by a route cost metric, e.g., the number of hops. In this scheme, only routing updates that change the cost of a given route need to be propagated. Routes in link-state and distance-vector networks are built dynamically from many independent forwarding decisions, so adversaries have limited power to affect packet forwarding, making these protocols immune to carousel and stretch attacks.

In GPSR, a packet may encounter a dead end, which is a localized space of minimal physical distance to the target, but without the target actually being reachable. The packet must then be diverted until a path to the target is available. In BVR, packets are routed toward the beacon closest to the target node, and then move away from the beacon to reach the target. Each node makes independent forwarding decisions, and thus a Vampire is limited in the distance it can divert the packet. These protocols also fall victim to directional antenna attacks in the same way as link-state and distance-vector protocols above, leading to energy usage increase factor of O(d) per message, where d is the network diameter. Moreover, GPSR does not take path length into account when routing around local obstructions, and so malicious misrouting may cause up to a factor of O(c) energy loss, where c is the circumference of the obstruction, in hops.

IV. CLEAN STATE SECURE ROUTING PROTOCOL (PLGP)

The PLGP protocol is modified as clean state secure routing protocol such that they can resist vampire attacks during the forwarding. PLGP was vulnerable to vampire attacks even though they were said to be secured. When the route discovery begins each node has a limited view about the network. As already said nodes discover the other nodes in a group by broadcasting a certificate id, signed by the public key of the online authority, thus forming a single group and a tree structure that will be used for addressing and routing. All nodes compute the same address as the other odes they also learn each other's virtual address as well as their cryptographic keys. The final address is verifiable after the network convergence and all forwarding decisions can be independently verified. PLGP consists of a topology discovery phase, followed by a packet forwarding phase, with the former optionally repeated on a fixed schedule to ensure that topology information stays current. (There is no on- demand discovery.) Discovery deterministically organizes nodes into a tree that will later be used as an addressing scheme. When discovery begins, each node has a limited view of the network—the node knows only itself. Nodes discover their neighbors using local broadcast, and form ever expanding "neighborhoods," stopping when the entire network is a single group. Throughout this process, nodes build a tree of neighbor relationships and group member- ship that will later be used for addressing and routing.

At the end of discovery, each node should compute the same address tree as other nodes. All leaf nodes in the tree are physical nodes in the network, and their virtual addresses correspond to their position in the tree (see Fig. 6).

All nodes learn each others' virtual addresses and cryptographic keys. The final address tree is verifiable after network convergence, and all forwarding decisions can be independently verified. Furthermore, assuming each legitimate network node has a unique certificate of membership (assigned before network deployment), nodes who attempt to join multiple groups, produce clones of themselves in multiple locations, or otherwise cheat during discovery can be identified and evicted.

A. Provable Security Against Vampire Attacks

Here, we modify the forwarding phase of PLGP to provably avoid the above-mentioned attacks. First we introduce the no backtracking property, satisfied for a given packet if and only if it consistently makes progress toward its destination in the logical network address space. More formally:

Definition 1. No-backtracking is satisfied if every packet p traverses the same number of hops whether or not an adversary is present in the network. (Maliciously induced route stretch is bounded to a factor of 1.)

This does not imply that every packet in the network must travel the same number of hops regardless of source or destination, but rather that a packet sent to node D by a malicious node at location L will traverse the same number of hops as a packet sent to D by a node at location L that is honest. If we think of this in terms of protocol execution traces, no-backtracking implies that for each packet in the trace, the number of intermediate honest nodes traversed by the packet between source and destination is independent of the actions of malicious nodes. Equivalently, traces that include malicious nodes should show the same network- wide energy utilization by honest nodes as traces of a network with no malicious actors. The only notable exceptions are when adversaries drop or mangle packets en route, but since we are only concerned with packets initiated by adversaries, we can safely ignore this situation: "premangled" packets achieve the same result—they will be dropped by an honest intermediary or destination.

No-backtracking implies Vampire resistance. It is not immediately obvious why no-backtracking prevents Vampire attacks in the forwarding phase. Recall the reason for the success of the stretch attack: intermediate nodes in a source route cannot check whether the source-defined route is optimal, or even that it makes progress toward the destination. When nodes make independent routing decisions such as in link-state, distance-vector, coordinate-based, or beacon-based protocols, packets cannot contain maliciously composed routes. This already means the adversary cannot perform carousel or stretch attacks— no node may unilaterally specify a suboptimal path through the network. However, a sufficiently clever adversary may still influence packet progress. We can prevent this interference by independently checking on packet progress: if nodes keep track of route "cost" or metric and, when forwarding a packet, communicate the local cost to the next hop, that next hop can verify that the remaining route cost is lower than before, and therefore the packet is making progress toward its destination. (Otherwise we suspect malicious intervention and drop the packet.) If we can guarantee that a packet is closer to its destination with every hop, we can bound the potential damage from an attacker as a function of network size. (A more desirable property is to guarantee good progress, such as logarithmic path length, but both allow us to obtain an upper bound on attack success.)

Definition 2. The hop count of packet p received or forwarded by an honest node, is no greater than the number of entries in p's route attestation field, plus 1. When any node receives a message, it checks that every node in the path attestation 1) has a corresponding entry in the signature chain, and 2) is logically closer to the destination than the previous hop in the chain (see Function secure_forward_packet). This way, forward- ing nodes can enforce the forward progress of a message, preserving no-backtracking. If no attestation is present, the node checks to see if the originator of the message is a physical neighbor. Since messages are signed with the originator's key, malicious nodes cannot falsely claim to be the origin of a message, and therefore do not benefit by removing attestations.

V. MODULES

A. Topology Discovery and Cluster Head Selection

The topology we are going to use here is a mesh topology. In this case each node sends a message to the other nodes which it detects. Once a node detects the message it maintains a record to store information about the neighbor. Using multicast socket all nodes are used to detect their neighbor node. Cluster Head is elected based on range, mobility and battery power.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

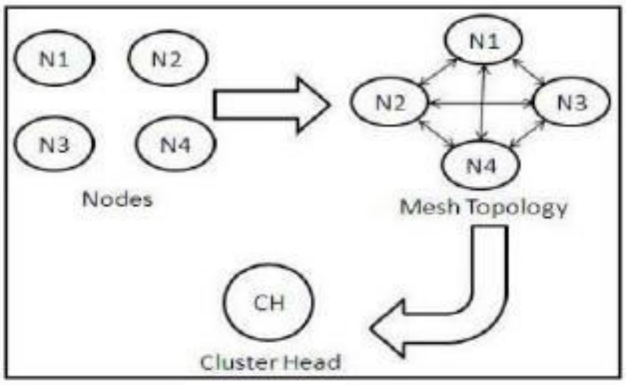

Fig: 6. Topology Discovery and Cluster Head Selection

B. Tree Formation and Route Discovery

Trees are formed as nodes form group. Each node starts with group size 1 and virtual address 0 so that one group is formed. Similarly other groups are also formed. When two nodes form a group their group size becomes 2 with one node taking a virtual address 0 and other taking the address 1.Each group can have their own group address. Example: node 0 in one group0 becomes 0.0 and node 0 in group 1 becomes 1.0. Each time a group is added or merged the address of each node is lengthened by one bit . Thus a tree structure is formed with address in the network and node address as leaves. Generally small groups form with 1 node later they merge to form large groups.

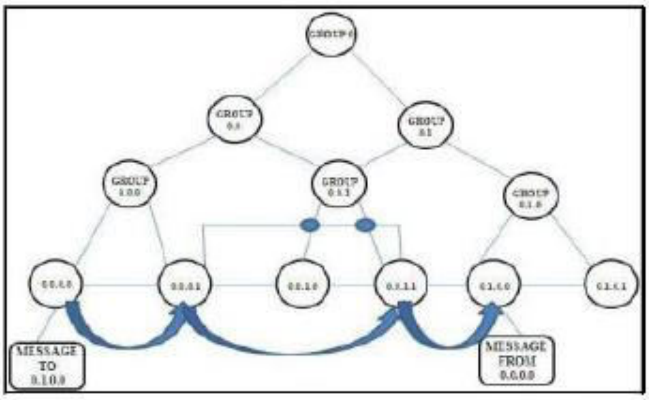

Fig: 7.Group identification

For example when two groups merge to form a large group they broadcast their group id to each other and precede with the merge protocols. Each node stores the id of one or more nodes such that they can know that the other group exists such that every node within a group will end up with the next-hop path to every other group as in distance vector. Thus a tree is formed and the route is chosen in this manner until all network nodes are members of single group.

C. Forwarding the Packets

During this phase each node is independent of other node and hence the decision made by them is also independent. When a node receives a packet it determines the next hop by finding the most significant bit(MSB) address as it differs from the message originators address as it differs from the originators address.

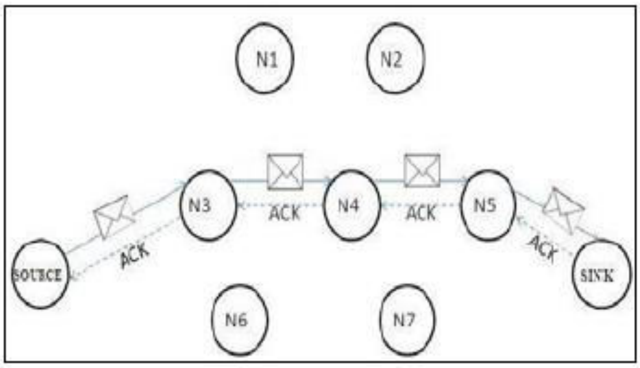

Fig: 8.Message traversal in normal situation.

When a packet is moving within a group and when they want to move to the next group they shortens the logical distance to destination since their address must be close to the destination.

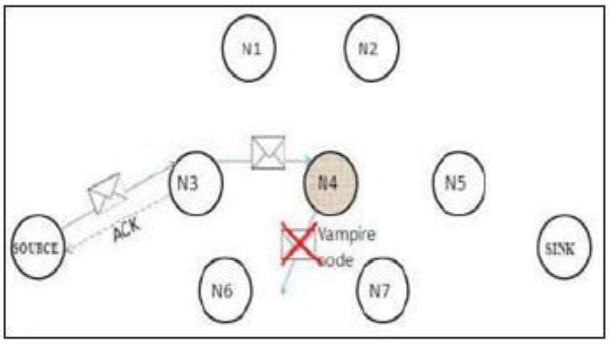

Fig: 9. Vampire Attack leading to message drop.

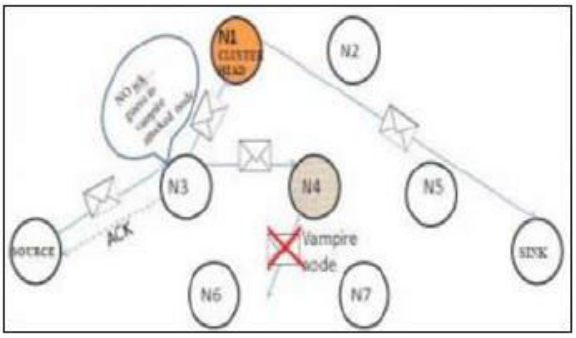

Fig: 10. After N trials node N3 sends data to cluster head (chosen based on highest coverage range, battery power) which sends data to sink.

VI. CONCLUSION

In this paper we defined Vampire attacks, a new class of resource consumption attacks that use routing protocols to permanently disable ad-hoc wireless sensor networks by depleting nodes' battery power. These attacks do not depend on particular protocols or implementations, but rather expose vulnerabilities in a number of popular protocol classes. We showed a number of proof-ofconcept attacks against representative examples of existing routing protocols using a small number of weak adversaries, and measured their attack success on a randomly-generated topology of 30 nodes.

Simulation results show that depending on the location of the adversary, network energy expenditure during the forwarding phase increases from between 50 to 1,000 percent. We proposed defenses against some of the forwarding-phase attacks and described PLGPa, the first sensor network routing protocol that provably bounds damage from Vampire attacks by verifying that packets consistently make progress toward their destinations. We have not offered a fully satisfactory solution for Vampire attacks during the topology discovery phase, but suggested some intuition about damage limitations possible with further modifications to PLGP. Derivation of damage bounds and defenses for topology discovery, as well as handling mobile networks, is left for future work.

REFERENCES

[1]. “The Network Simulator - ns-2,” http://www.isi.edu/nsnam/ns,2012.
[2]. A.D. Wood and J.A. Stankovic, “Denial of Service in Sensor Networks,” Computer, vol. 35, no. 10, pp. 54-62, Oct. 2002.
[3]. I. Aad, J.-P. Hubaux, and E.W. Knightly, “Denial of Service Resilience in Ad Hoc Networks,” Proc. ACM MobiCom, 2004.
[4]. J. Bellardo and S. Savage, “802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions,” Proc. 12th Conf. USENIX Security, 2003.
[5]. J. Deng, R. Han, and S. Mishra, “Defending against Path-Based DoS Attacks in Wireless Sensor Networks,” Proc. ACM Workshop Security of Ad Hoc and Sensor Networks, 2005.
[6]. J. Deng, R. Han, and S. Mishra, “INSENS: Intrusion-Tolerant Routing for Wireless Sensor Networks,” Computer Comm., vol. 29,no. 2, pp. 216-230, 2006.
[7]. A. Nasipuri and S.R. Das, “On-Demand Multipath Routing for Mobile Ad Hoc Networks,” Proc. Int’l Conf. Computer Comm. And Networks,1999.
[8]. M.G. Zapata and N. Asokan, “Securing Ad Hoc Routing Protocols,” Proc. First ACM Workshop Wireless Security (WiSE),2002.
[9]. Y.-C. Hu, D.B. Johnson, and A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks,” Proc. IEEE Workshop Mobile Computing Systems and Applications, 2002.
[10]. Y.-C. Hu, D.B. Johnson, and A. Perrig, “Ariadne: A Secure On- Demand Routing Protocol for Ad Hoc Networks,” Proc. MobiCom, 2002.
[11]. B. Parno, M. Luk, E. Gaustad, and A. Perrig, “Secure Sensor Network Routing: A Clean-Slate Approach,” CoNEXT: Proc. ACM CoNEXT Conf., 2006.
[12]. Y.-C. Hu, D.B. Johnson, and A. Perrig, “Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks,” Proc.IEEE INFOCOM, 2003.
[13]. H. Chan and A. Perrig, “Security and Privacy in Sensor Networks,” Computer, vol. 36, no. 10, pp. 103-105, Oct. 2003.
[14]. D.R. Raymond, R.C. Marchany, M.I. Brownfield, and S.F. Midkiff, “Effects of Denial-of-Sleep Attacks on Wireless Sensor Network MAC Protocols,” IEEE Trans. Vehicular Technology, vol. 58, no. 1, pp. 367-380, Jan. 2009.
[15]. C. Karlof and D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures,” Proc. IEEE Int’l Workshop Sensor Network Protocols and Applications, 2003.
[16]. J.-H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks,” IEEE/ACM Trans. Networking, vol. 12,no. 4, pp. 609-619, Aug. 2004.
[17]. S. Doshi, S. Bhandare, and T.X. Brown, “An On-Demand minimum Energy Routing Protocol for a Wireless Ad Hoc Network,” ACM SIGMOBILE Mobile Computing and Comm. Rev.,vol. 6, no. 3, pp. 50-66, 2002.
[18]. L.M. Feeney, “An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,”Mobile Networks and Applications, vol. 6, no. 3, pp. 239-249, 2001.