Wireless Sensor Networks of Battlefields Hotspot:

Challenges and Solutions

Naif Alsharabi, Li Ren Fa, Fan Zing
College of Computer and Communication Engineering
Hunan University
Hunan, Changsha, 410082, CHINA
Sharabi28@hotmail.com, lirenfa@vip.sina.com

Mossa Ghurab
College of Computer Science
Zhejiang University
Zhejiang Province, Hangzhou, CHINA
mossaghurab@hotmail.com

***Abstract*— in recent years, a lot of research is focused on wireless sensor network applications, which is focused on field of performance, security, and energy. This paper addressed the difficulties and challenges facing the wireless sensor networks on the battlefield. Which is often vulnerable to attacker's networks either in the data or corrupting control devices and attempt to consume a lot of energy by sending a large quantity of useless packets, which contributes to excessive consumption of energy and leads to exit nodes from work. Since technology has become widespread on battlefields at the present time, then the sensor nodes are vulnerable to attackers from both sides. This research discussed many challenges and gave appropriate solutions. The simulations showed that these solutions can help secure data and saved 40% of energy consumed.**

Index Terms - sensor, attacker, security, energy, battlefield

I. Introduction

In sensor networks, clustering is used to organize sensor nodes into groups based in part on their physical proximity [1]. In the clustering algorithm proposed in [2], clusters are formed by having each sensor node wait a random amount of time. If a node has not had the opportunity to join a cluster after this random amount of time, then it can declare itself to be a cluster head and subsequently start soliciting neighboring nodes to join its cluster. To maintain the cluster, the cluster head will select its own successor. We foresee two vulnerabilities with this approach. First, during cluster formation an adversary could ensure its selection as cluster head by immediately soliciting other nodes to join its cluster. Second, once an adversary node has been selected as cluster head it can remain cluster head indefinitely, by never selecting a successor. Consequently, this approach readily allows an adversary to launch a sleep deprivation attack.

There are many other distributed clustering algorithms and sensor network applications, which rely upon clustering [3-10], each of which assumes that participating nodes will act honestly. Thus, an adversary can exploit each of these algorithms to ensure its selection as cluster head. Given that clustering is a widely used algorithm, it is crucial to make it secure.

II. Attacks Model[①]

A. *Black Hole Attack*

In multi-hop WSNs, the sensor nodes act as routers to relay messages from their children to their parents and eventually to the base. In a black hole attack (ex. table (1)), an attacker drops the incoming packets from its children nodes. In order to remain unnoticed, the adversary keeps sending self-generated packets only; thus, the malicious node may appear normal to its parent, which makes it hard for the administrator to figure out the cause of disconnection from a certain group of nodes to the base. In dense networks, it is even harder to detect and locate which cluster infected of the attack and localize the malicious node because the aftermath of this attack is more severe in terms of the total number of affected nodes.

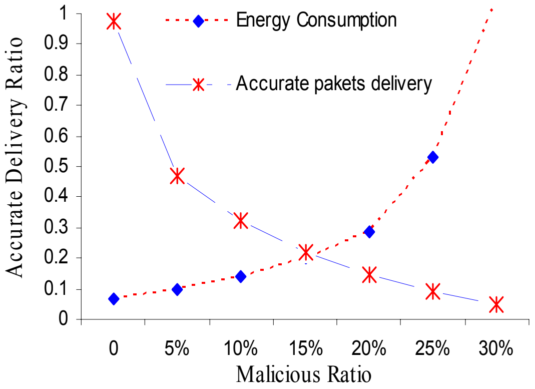

Figure 1Black Hole attack

Seven scenarios implemented according to the number of malicious nodes on the network, the energy consumption of each sensor node is as follows: Ea =100 pJ/bit/m2, Ee = 50 nJ/bit and Ec = 5 nJ/bit where consumed for transmitting, receiving and listening respectively. Each sensor needs to send a packet of length R = 400 bits to the cluster head on random time. Cluster head period T is set as 2000s and the execution time of task is set as = 0.005 s. The data packet size is 2 KB and the parameter r =105 and the sensing range to 64 meters.

[①] Example drawing reproduced from [11-17]

This research supported by
National Natural Science Foundation of China (Grant NOs.60673061)
Hunan Natural Science Foundation of China (Grant NOs.063350111,063350113)
National Research Foundation for the Doctor Program of Higher Education of China (Grant NOs. 2006 05 32024)

WIOPT 2008, 1st－3rd Apr 2008, Berlin, Germany.
ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.2983

Thus, network administrators may correlate the nodes' disconnections with other known factors, such as throughputs, no path to the sink, energy consumption, etc. As shown in Figure 1, the inverse relationship between the increase in the number of infected node and the wireless sensor networks performance rate. The figure shows that the network would fail whenever increased the number of malicious nodes in the network. These indicators could certainly show the administrator the affected area and correct the imbalance

B. *Sink Hole Attack*

The number of children nodes, using a malicious node to relay to the sink, limits the effect of a black hole attack. Therefore, a sink hole attack is an advanced version of the black hole attack. In this attack, an attacker tries to attract more neighbors by advertising wrong routing information, often in shorter hops. This makes the attacker capable of affecting a larger number of nodes.

C. *Selective Forwarding Attack*

The selective forwarding attack (ex. table (2)) is a smarter attack than the previous two. In this attack, the attacker selectively drops packets. The selection of packets is based on some predefined criteria, which makes it even harder to detect. The attacker selects either on the basis of the packet's contents or the packet's source/origin address(s). Even though there can be many different versions of this attack, in our implementation, we focus on an address based selective forwarding attack.

Figure 2 shows that the delivery packets normal when the malicious node zero and decrease when the malicious node increase. This evidence can guide the administrator to detect the affected areas by this attack and correct the imbalance.

D. *Flooding Attacks*

This attack leads to DOS by over-consuming the resources of the network nodes.

An attacker tries to flood the network so that either the nodes' battery depletes at a faster rate, or the memory is exhausted. Thus, the affected nodes die or crash much earlier than their expected lifetime. This attack has many versions; a few of them are listed below:

1) *Simple Broadcast Flooding Attack*

An attacker simply floods the network with broadcast messages. As a result, every node, which hears these messages, gets affected, since each of them has to waste energy in processing the received messages. This causes the affected nodes to die earlier than their normal lifetime.

2) *Simple Target Flooding Attack*

An attacker targets a particular node, or a group of nodes, by changing the destination address in the packet header of the outgoing message. Messages will then be routed to the parent of the targeted node and eventually to the sink; thus all the nodes in the path of the targeted nodes are affected. This is an advanced version of the previous attack.

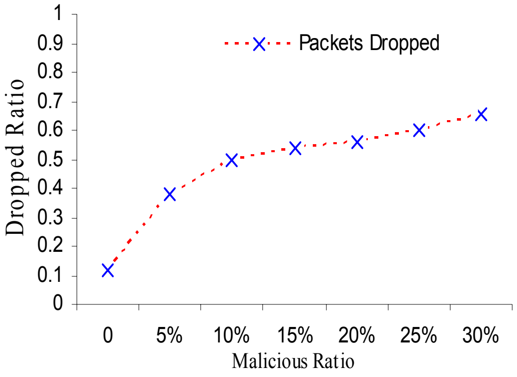

Figure 2 Selective Forwarding Attack

Seven scenarios implemented according to the number of malicious nodes on the network, the energy consumption of each sensor node is as follows: Ea=100 pJ/bit/m2, Ee = 50 nJ/bit and Ec = 5 nJ/bit where consumed for transmitting , receiving and listening respectively. Each sensor needs to send a packet of length R = 400 bits to the cluster head on random time. Cluster head period T is set as 2000s and the execution time of task is set as = 0.005 s. The data packet size is 2 KB and the parameter r =105 and the sensing range to 64 meters.

3) *False Identity Broadcast Flooding Attack*

This is similar to the simple broadcast flooding attack, with one main addition: the attacker advertises a wrong origin/source address(s) in the header of the flood messages. This makes it harder for the network administrator to identify the malicious nodes; in addition, the base station receives packets containing wrong source/origin address(s).

4) *False Identity Target Flooding Attack*

This is a combination of the previous two attacks. In this attack, the adversary not only hides its original identity, but also targets a node, or a group of nodes, with a flood of messages. Since it is one of the worst types of flooding attacks, we use this attack to demonstrate the detection capability of SNAIDS. However, SNAIDS fails to locate an attacking node when an attacker falsifies its identity, because like all other nodes, the base station also receives incorrect information in the packets' header.

III. Proposed Solutions and Simulation

In this section, we will implement our algorithm [18], which published earlier as an optimum and suitable solution for most of the assailants, which attack sensor networks on the battlefield. We are not going to repeat the SASO algorithm details here, So a brief information of SASO algorithm will setup in this section.

A. *Assumptions*

The algorithm assumed that the sensor nodes should have the following keys:

1) *Master key*: each sensor node in manufacture time is imprinted with Master key and Local Administrative Function.

2) *Local control key:* Before node deployment, each node is injected with initial Local Control Key (LC), which is the basic parameter for the re-keying function of our proposal.

3) Session key: a session key randomly is generated to ensure the security of a communications session between nodes. A session key is derived from master key and LC key using session-key derivation scheme. Session keys are changed frequently.

B. Algorithm Description

The algorithm was proposed that the Local Administrative Function imprinted with sensor node to achieve a high-level security of node-to-node communication. The Local Administrative Function is the core task of SASO algorithm, which is the HMAC is the base of Local Administrative Function. To clarify the significance of our proposal, we introduce the basic components of our function MAC and HMAC.

MAC function stands for Message Authentication Code. In general, a MAC can be thought of as a checksum for data passed through an unreliable (or more importantly, unsecured) pipeline. A sender will typically generate a MAC code by first passing their message into some MAC algorithm. The sender will then send their message M with the MAC (M). The receiver can then generate their own MAC (M) and verify that MAC (M) sent by the receiver matches the MAC (M) they themselves generated.

A MAC algorithm can be generated using multiple different techniques; however, sender and receiver generally need to have a shared secret key K. A MAC algorithm could be done out of a common symmetric cipher such as DES2 or AES3. A sender wanting to send a secure message can send M encrypted, e(M), with a symmetric cipher and then resend M||K (M concatenated with K) encrypted, e(M||K). The receiver first decrypts M, d(e(M)), to generate M'. Then we encrypt M'||K, e(M'||K) and compares with the e(M||K) originally sent. If the two match, mean that the data did not corrupt.

HMAC is merely a specific type of MAC function. It works by using an underlying hash function over a message and a key. Any hashing function could be used with HMAC, although more secure hashing functions are preferable. The following flowchart shows how Local Administrative Function works.

The derivation function $f(\ .\ )$ is used to generate new key values based on the old key values, our goal of using a re-keying process function is to achieve two properties. First is , given k is easy to compute f(k), but given f(k), it is computationally infeasible to compute k. The second is, given $k_0, k_1, k_2,\ldots k_n$, it is computationally infeasible to compute f(k), if it is computationally infeasible to compute k. Proposed f(.) function to a void the producing repetitive key values with the same input key value k, in some occasions, a non-zero salt value LC key is used in $f_i(k \oplus LC_i)$[2]

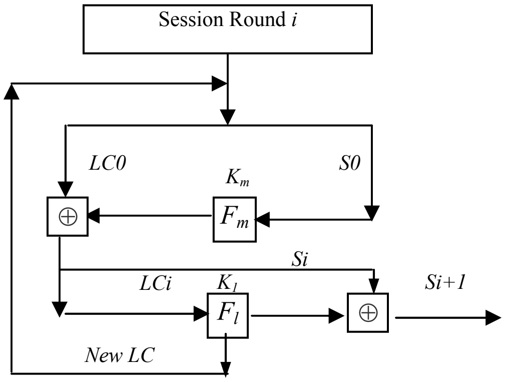

Figure 3 Derivation Function
$k_0, k_1 \ldots k_n$ are a keys driving from master key K of the cluster, S_0 is the first session key generated according to k_0 and Lc_0 , *LC* is the local control key and F_m, F_l is a Master function and Re-king local control function respectively.

This is able to produce different values of session keys even with the same k when LC is varied. After session key is performed, the re-keying function will assign a new value to LC. More details about SAO algorithm refer to ref. [18].

IV. Simulations and Discussion

Figure 4 showed the platform of the battlefield in our proposed simulation which is the role-based hierarchical of SASO algorithm and Administrative Function [18] was simulated using Omnet++ [19]. The simulator can also be used to view the topology generated by the initial self-organization algorithm. A comparison between Leach showed in figures 1, 2 and Leach with added our approach showed in figures 5, and 6 have been done using the same number of clusters and sensing zones. To achieve this, the simulator assumed that no packet collisions occurred. It also assumed that there were no packet errors during transmission and reception.

In other words, we assumed a perfect wireless channel. Figures 5 and 6 show the results of an example simulation (16 rounds[3]) run with the following simulation parameters:

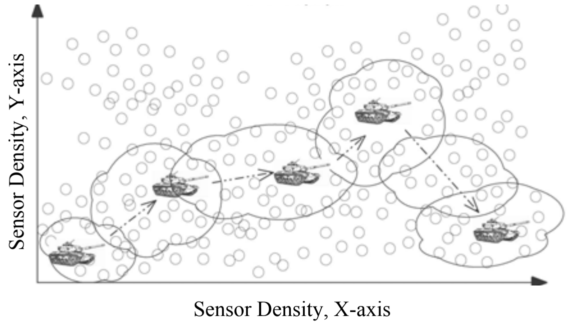

Figure 4 Battlefield Platform
Five tanks moving in square meters 300x300, with 250-sensor nodes density.

[2] i^{th} which is mean each session it has a new session key according to derivation function and the i^{th} incremental is administrative by cluster head.

[3] very simple part choose from round 0 because the simulation results gave much data for each round, which is difficult presented here were represented in graph.

- 100 nodes in an area of 300 × 300 meters;
- 200 nodes in an area of 500 × 500 meters, (round 0 showed in table [3] [4]);
- 600 nodes in an area of 1400 × 1400 meters;
- 1000 nodes in an area of 2200 × 2200 meters.

For all the topologies, we set the radio range and the sensing range to 64 meters. The minimum and maximum sensing zone (or cluster) membership size was set to 5% and 10%, respectively. Finally, through the study of the characteristics of some attackers, some of nodes were selected and have been added to them special characteristics of the attackers to play the attack jobs on the networks.

The simulation results of data delivery are only for the normal data delivered, provided that the network is working normally. Simulations take into consideration only special types of attacks, like Selective Forwarding and Black Hole attacks. The simulation result compared the network with SASO algorithm (Figure 5 and 6) and with out SASO algorithm (Figure 1 and 2).

Figure 5 and 6, illustrate the effects observed. It allows us to measure the correct packet accurately delivered and the remaining transmission power for different scenarios of malicious nodes. The curve in the figure 5 illustrates that when the network is free from malicious nodes, 95% of accurate data reach safely and is real without falsification. The percentage of delivering accurate data reduces as the number of malicious nodes increases. Keeping the same conditions and simulation environments, when the malicious nodes are 30% the data delivered ratio more than 60% in figure 5 compared with less than 20% of data delivery in figure 1 with same rate of malicious nodes. In addition, figure 1 shows less gradually with increasing the proportion of malicious nodes until reach to specific rate; the network stopped when the malicious nodes ratio reached more than 30% because of very low accurate information reached to the sink, this is the structure of the network setup.

In the same vein, we find that when the network is free from malicious nodes, which is the first purpose, energy drain and re-submission of counterfeit data leads to a much loss of energy. Figure 6 indicates that more than 60% of energy would be lost when the malicious nodes rate is more than 30%, On the contrary when the simulation ran with out SASO algorithm, compared to figure 1 shows that when the malicious nodes are 30% in the network almost 90% of the power is consumed. This will lead to stopping the network, as half of the nodes within network would have died.

Figure 5 and 6, showed that the use of high efficient algorithm structure like SASO can protect the network from attackers and non-response data counterfeit return it to the network to improve network performance and maintain the accurate data exchanged between the nodes and sink. Moreover limiting the excessive consumption of energy that consumed by attackers.

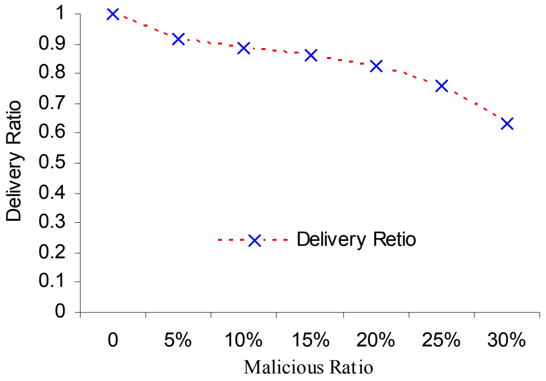

Figure 5 SASO: Black Hole and Selective attacks
Seven scenarios implemented according to the number of malicious nodes on the network, the energy consumption of each sensor node is as follows: Ea=100 pJ/bit/m2, Ee = 50 nJ/bit and Ec = 5 nJ/bit where consumed for transmitting , receiving and listening respectively. Each sensor needs to send a packet of length R = 400 bits to the cluster head on random time. Cluster head period T is set as 2000s and the execution time of task is set as = 0.005 s. The data packet size is 2 KB and the parameter r =105. and the sensing range to 64 meters.

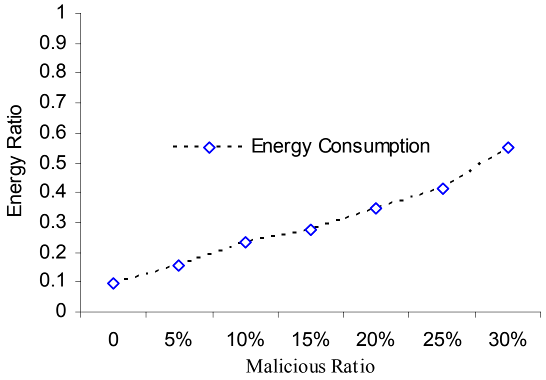

Figure 6 SASO: Energy Consumption, Black Hole and Selective attacks
Seven scenarios implemented according to the number of malicious nodes on the network, the energy consumption of each sensor node is as follows: Ea=100 pJ/bit/m2, Ee = 50 nJ/bit and Ec = 5 nJ/bit where consumed for transmitting , receiving and listening respectively. Each sensor needs to send a packet of length R = 400 bits to the cluster head on random time. Cluster head period T is set as 2000s and the execution time of task is set as = 0.005 s. The data packet size is 2 KB and the parameter r =105. and the sensing range to 64 meters.

V. CONCLUSIONS AND FUTURE WORKS

We have applied and simulated a SASO algorithm that was proposed in our former paper [18] to hotspot battlefield to achieve a secure group communication, reduce the challenges of link layer communication and energy consumption of the wireless sensor networks. This paper simulated a Local Administrative Function algorithm using Omnet++, which the simulation results showed that is efficient to establish a secure link-layer communication, keeping high rate of accurate date transferred, and the conservation of energy consumed by attackers. For future work, we plan to focus on the problems facing the energy efficiency of tiny-sensor in another area field.

Finally, we are planning to improve our simulations by adding the notions of program execution speed of the simulated software components and correct modeling of energy consumption.

TABLE I. BLACK HOLE ATTACKS

Round			Received					Error packets	
ID	*HOPS*	*SEND*	*N1*	*N2*	*N3*	*N4*	*N5*	*Forward*	*Dropped*
2	1	234	232	13	10	10	10	210	2
3	1	105	48	18	29	29	29	60	45
5	1	117	43	70	20	20	20	25	80
10	1	161	141	15	18	9	5	64	105
11	1	98	14	55	25	25	25	20	60
12	2	87	58	25	25	25	25	40	20
13	1	96	76	15	19	19	19	60	12
16	2	111	49	19	26	26	26	12	88
18	2	134	112	9	9	9	9	80	33
21	1	115	32	77	14	14	14	33	70

Simulation round 0: 2nodes with 5% malicious nodes attached

TABLE II. MALICIOUS NODE DURING SELECTIVE FORWARDING

Node ID	Seq. No	Received	Forward	Last Update
2	17	2016	1880	Unique S. K
3	41	1219	1116	Unique S. K
5	42	1000	911	Unique S. K
6	40	1813	1514	Unique S. K
7	42	391	216	Unique S. K
8	44	1625	1013	Unique S. K
9	44	594	410	Unique S. K
10	43	1828	1315	Unique S. K
11	44	1422	1015	Unique S. K
12	44	1000	911	Unique S. K
13	0	1203	1115	Unique S. K
14	50	1000	910	Unique S. K
15	49	1625	1310	Unique S. K
17	38	22610	1892	Unique S. K
18	49	2016	1808	Unique S. K
20	38	21813	1995	Unique S. K
21	45	3032	2915	Unique S. K

Simulation round 0: 2nodes with 5% malicious nodes attached

TABLE III. SASO ALGORITHM DURING SELECTIVE FORWARDING AND BLACK HOLE ATTACKS[④]

ID	Session Key		Hops	SEND	Successes Received				Error Packets	
	M	Re-k			*N1*	*N2*	*N3*	*N4*	*forward*	*Failed*
2	-1	0	1	234	232	13	10	10	8	0
3	-1	0	1	105	48	18	29	29	7	0
5	-1	10	1	117	43	70	20	20	0	0
6	-1	4	1	161	141	15	18	9	0	0
7	-1	8	1	98	14	55	25	25	0	0
8	-1	0	2	87	58	25	25	25	0	0
10	-1	10	1	96	76	15	19	19	0	0
11	-1	6	2	111	49	19	26	26	0	0
12	-1	6	2	134	112	9	9	9	0	0
13	-1	0	1	115	32	77	14	14	0	0
14	-1	0	1	234	232	13	10	10	0	0
21	-1	2	2	105	48	18	29	29	0	0

TABLE IV. SASO:RE-KEYING ALGORITHM DURING SELECTIVE FORWARDING AND BLACK HOLE ATTACKS

Node ID	received	forwarding	Pinging	Last Update
2	71	53	1406	Keying
3	66	50	1406	Keying
6	65	50	1812	Keying
7	67	53	406	Keying
9	64	47	1000	Keying
10	64	52	203	Keying
12	39	27	51359	Keying
19	61	46	2218	Keying
21	30	29	2015	Keying

Simulation round 0: current time 1161078286046, 200 nodes with 5% malicious nodes attached

REFERENCES

[1] D. W. Carman, P. S. Kruus, and B. J. Matt, “Constraints and approachesfor distributed sensor network security,” NAI Labs #00-010, Tech. Rep.,**2000**

[2] Wendi Rabiner Heinzelman, Anantha Chandrakasan and Hari Balakrishnan Hawaaian“Energy-Efficient Communication Protocols for Wireless Microsensor Networks (LEACH)”. Int'l Conf. on Systems Science, January **2000**.

[3] S. Bandyopadhyay and E. J. Coyle, “An energy-efficient hierarchical clustering algorithm for wireless sensor networks,” in *INFOCOM*, vol. 3, **2003**, pp. 1713–1723.

[4] E. J. Duarte-Melo and M. Liu, “Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks,” in *Globecom*, **2002**, pp. 21–25.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energyefficient communication protocol for wireless microsensor networks,” in *HICSS*, **2000**.

[6] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu, “Discrete mobile centers,” in *Computational Geometry*, **2001**, pp. 188–196.

[7] S. Basagni, “Distributed clustering for ad hoc networks,” in *ISPAN*, **1999**, pp. 310–315.

[8] M. Gerla and J. T.-C. Tsai, “Multicluster, mobile, multimedia radio network,” *Wireless Networks*, **1995,** vol. 1, no. 3, pp. 255–265,.

[9] C. Chiang, H. Wu, W. Liu, and M. Gerla, “Routing in clustered multihop, mobile wireless networks with fading chanel,” in *SICON*, **1997**, pp. 197–211.

[10] Jing Deng, Richard Han, and hivakant Mishra. Defending Against Pathbased DoS Attacks in ireless Sensor Networks. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2005), November **2005**. p 1 – 2.

[11] James Newsome, Elaine Shi, Dawn Song, Adrian Perrig. The Sybil Attack in Sensor Networks: Analysis & Defenses. New York, NY, USA: ACM Press, **2004**. p 3 – 8.

[12] Michael Brownfield, Yatharth Gupta, Nathaniel Davis IV. Wireless Sensor Network Denial of Sleep Attack. **2005**. p 356 – 364.

[13] Thomas Martin, Michael Hsiao, Dong Ha, Jayan Krishnaswami. Denial-of- Service Attacks on Battery-Powered Mobile Computers. Washington, DC, USA: IEEE Computer Society, **2004**.

[14] Chris Karlof, David Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. Anchorage, AK, USA: 1st IEEE International Workshop on Sensor Network Protocols and Applications, May 11, **2003**. p 293 – 312.1

[15] Adrian Perrig, John Stankovic, David Wagner. Security in Wireless Sensor Network. New York, NY, USA: ACM Press, **2004**. p 53 – 57.

[16] Anthony D.Wood, John A.Stankovic. Denial of Service in Sensor Networks. Los Alamitos, CA, USA: IEEE Computer Society Press, October **2002**. p 54 – 62.

[17] John R. Douceur. The Sybil Attack. 2002. p 1 – 6.

[18] Naif Alsharabi, Li RenFa“ Security Adaptive Self-Organization for WSNs” IEEE Computer Society Press, Wireless Communications Networking and Mobile Computing, 2007,P 2503 -2506.

[19] http://www.omnetpp.org/filemgmt/viewcat.php?cid=2

[④] The tables’ values are very simple values taken from whole simulation data to explain how the data transfer and the re-king algorithm is working.