Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 6, June 2016, pg.476 – 486

Insider Threats in Wireless Sensor Networks and Their Countermeasures

Shivani Garg, Mukul Varshney, Aparajita Nailwal

Department of CSE, Sharda University

shivani.garg@sharda.ac.in, mukul.varshney@sharda.ac.in, aparajita.naiwal@sharda.ac.in

Abstract- Wireless sensor networks tend to have a wide range of applications in our day to day life. In future, they can be used to survey our health, our home, the roads we follow, the office or the industry we work in or even the aircrafts we use, in an attempt to enhance our safety. But, these networks themselves are prone to security attacks. The list of security attacks is already very large and keeps on increasing with the expansion of these networks. A powerful tool for the detection of faulty or malicious nodes is the trust management schemes. Having detected the misbehaving nodes, their neighbours can use this information to avoid relying on them, either for data forwarding, data aggregation or any other cooperative function. There are a variety of trust models and most of them focus on defending against certain insider attacks. This paper discusses several security vulnerabilities that the trust mechanisms have. We also examine how inside attackers can exploit these security holes, and propose approaches that can mitigate the weaknesses of trust mechanisms.

Keywords: wireless sensor networks, security attacks, malicious nodes, insider attacks, trust mechanisms

I. INTRODUCTION

A vital security concern in wireless sensor network (WSN) is the insider threat. This is because inside attackers cannot be caught using traditional security mechanisms, like authentication an authorization, as they are the legal members of the network. These inside attackers can cause harm to the normal network functionalities by dropping, modifying or misrouting data packets. Thus, insider attacks are a serious threat for critical activities like military surveillance system and other critical infrastructures

To defend against the insider attacks [11,12,15,24],Trust mechanisms have been developed with the notion of trust in human society. As the WSNs consist of hundreds or thousands of tiny sensor nodes, the trust mechanism is usually realized as a distributed system where each sensor is equipped with the capability to evaluate, update, and store the trustworthiness of other nodes based on the trust model.

In general, working of trust mechanism consists of the following three stages 1) node behavior monitoring, 2) trust measurement, and 3) insider attack detection. A popular monitoring mechanism for the first stage is Watchdog [8]. A trust model such as beta trust model [4] and entropy trust model [10] processes the other two stages using the data collected by the watchdogs. These trust mechanisms work by continuously monitoring the behavior of the nodes and updating the trust value of the nodes. Each node has a watchdog which monitors the behavior of the neighboring nodes and lowers the trust value of the misbehaving neighbor. When the trust value of a certain node goes below a trust threshold, this node is termed untrustworthy and removed from the neighbor list.

Though it seems to be a sound mechanism, but, there are several weaknesses in it. First, due to inherent weaknesses of WSNs, watchdog has some security vulnerabilities such as distributed sensors, limited transceiver range, and multi-hop routing [5, 8]. Second, inside attackers cannot be prevented completely from dropping packets by any trust model. This is because packet can be dropped not only by an attacker but also due to contention or noise. Thus, an inside attacker can disguise its malicious behavior taking advantage of network traffic or noise. Third, we cannot ignore the fact that as insiders have internal knowledge about our network and security mechanisms against attacks, they can launch their attacks intelligently by exploiting such knowledge and avoid being detected.

Many existing trust models with watchdog as their monitoring mechanism do not explicitly address these weaknesses. Our goal in this paper is to demonstrate how these insider attacks can pose threats to WSNs even after having a trust mechanism and watchdog, and to introduce defending approaches to improve the trust mechanism.

II. INSIDER ATTACKS IN WSNs

To safeguard our network, our WSN is assumed to be equipped with cryptography-based authentication and authorization to withstand outside attackers launching eavesdropping or packet modification [5]. In this WSN, outside attacks may be limited to directly damaging sensors by physical strike or jamming. Meanwhile, inside attackers have some advantages compared with outside attackers. First, as inside attackers can avoid our authentication and authorization, they can secretly cause damage to our network and it is difficult to foresee their attack patterns. Second, inside attackers can cause damage to the sensors and also disturb our network by dropping critical packets or by maliciously modifying packet information. Inside attackers can launch various types of active (modification, packet drop, or misrouting) as well as passive (eavesdropping) attacks. While modification, misrouting, and eavesdropping can be prevented to some extent by the authentication and authorization, it is quirky to counter packet drop attacks because for a particular packet drop, it is difficult to conclude that it is the result of an act by attacker or it is due to a collision or noise. Moreover, inside attackers situated at a critical place in the network (e.g., near BS) can significantly deteriorate network performance such as packet delivery rate as a result of their repeated packet drops.

There are several types of packet drop attacks such as balckhole attack, grayhole attack, and on-off attack [10, 15] as described in Table 1. Compared to blackhole attack, it is difficult to detect grayhole attack and on-off attack because of their complex attack patterns. Moreover, packet drop attacks have evolved to drop packets intelligently by exploiting inside knowledge about our network and security mechanism to avoid being detected [15]. Hence, in this paper, main focus is on inside attackers' packet drop attacks.

Attack	Description
No Attack	Forward all packets
Blackhole Attack	Drop all packets
Grayhole Attack	Drop (specific) packets randomly
On-off Attack	Drop all or some portion of packets periodically

Table 1: Insider Packet Drop Attacks and Description

III. TRUST MECHANISM

In general, the working of a trust mechanism constitutes following stages.

1) *Node behavior monitoring*: Each sensor node is required to monitor and record its neighbors' behaviors such as packet forwarding. In the next stage, this collected data will be used for evaluating trustworthiness. A popular monitoring mechanism used in this stage is watchdog. The confidence of the trustworthiness evaluation depends on the amount of data collected by a sensor and reliability of the collected data.
2) *Trust measurement*: Trust model defines how to measure the trustworthiness of a sensor node.Several representative approaches to build the trust models have been introduced by Yu et al [15], which include Bayesian approach, Entropy approach, Game-theoretic approach, and Fuzzy approach. The trust value of a node may be different when we use different trust models. For example, when a node is observed to forward the packet *s* times and drops the packet *f* times, the beta trust model [4] will assign trust value T ($0 \le T \le 1$) to this node using the following formula

$$T= (s+1)/(s+f+2)$$

Meanwhile, entropy trust model [10] uses entropy function $H(p)$, whose input p is the trust value that can be obtained from beta trust model, to determine the trust value T. The entropy function $H(p) = -p \log_2 p -(1-p) \log_2 (1-p)$ and the trust value T ($-1 \le T \le 1$) is defined by

$$T = \begin{cases} 1 - H(p), \mathit{for}\ 0.5 \le p \le 1; \\ H(p) - 1, \mathit{for}\ 0 \le p < 0.5. \end{cases}$$

3) *Inside attack detection*: Based on the trust value, a sensor node determines whether its neighbor is trustworthy for collaboration (such as packet forwarding). If a neighbor's trust value is less than a certain threshold θ_T, it will be considered as an untrusted or malicious node. Depending on the WSN's trust mechanism, the detection of such insider attacker may or may not be broadcast to the rest of the nodes in the WSN.

IV. VULNERABILITIES IN TRUST MECHANISM

In this section, we examine the security vulnerabilities in each of the above mentioned three stages of a trust mechanism.

1) *Vulnerabilities in the node behavior monitoring stage:* This stage collects data for the evaluation of trust value in the later stages. It is necessary to collect reliable and trusted data. The entire trust mechanism will be ruined if this stage is infected by inside attackers. As we focus on trust mechanisms that rely on watchdog for data collection, this stage will have the same vulnerabilities as those for watchdog: ambiguous collision, receiver collision, and limited transmission power.

2) *Vulnerabilities in the trust measurement stage:* The principal threat in this stage is that an inside attacker may figure out the trust mechanism and the associated parameters, such as the trust (or trustworthiness) threshold θ_T, being used. This can be easily accomplished owing to the simplicity of the data collected in the previous stage and the limited available trust model. Also, the fact that an inside attacker need not know the exact information of the trust model to launch insider attacks without being detected, makes it even worse. For example, an inside attacker may be able to drop packets as long as its trust value is well above the trust threshold.
3) *Vulnerabilities in the inside attacker detection stage:* This stage classifies a node to be either trustful or distrustful. The single most important parameter for this classification is the value of trust threshold (θ_T). This threshold must be chosen carefully as a low θ_T will misclassify attackers as trustful nodes and a high θ_T will cause unnecessary false alarm. However, if an attacker gets a reasonably good estimation on the value of θ_T, insider attacks can be launched without being detected. As shown below in Table 2, if the attacker assumes $\theta_T = 0.7$, after certain number of initial successful forwarding (to build a high trust value), the attacker can drop a considerable number of packets consecutively without bringing its trustworthiness to 0.7 or below. For example, with s = 1000 previous successful forwarding, the next 428 packets can be dropped without being detected by the beta trust model, and 170 packets can be dropped if the entropy model is used.

Trust Model	Number of previous successful forwarding(s)					
	10	100	200	300	500	1000
Beta	3	42	85	128	213	428
Entropy	0	16	33	50	84	170

Table 2: Blackhole Attacker's Maximum Packet Drops Without Being Detected By Trust Models

V. DEFENDING APPROACHES

In the previous sections, we have shown that even a single security vulnerability in trust mechanism can result in a huge damage on our network. Therefore, there is a need to eliminate the identified vulnerabilities and have countermeasures that provide a shield against inside attackers exploiting the security holes. In Table 3, the working stages of a general trust mechanism are listed at the first column and related security vulnerabilities/attacks are listed at the second column. In this section, we present approaches to defend against the security vulnerabilities in each step and some existing works with their advantages and disadvantages (at the third column).

	Working Stages	**Security Vulnerabilities/Attacks**	**Defending Approaches**
Direct	1. Behavior monitoring based on watchdog	Limited overhearing/ Intentional collision, false behavior, collusion, partial dropping	Neighbor-based monitoring, acknowledgement-based monitoring, indirect observation
	2. Direct trust evaluation	Limited information/ Reverse engineering	Anomaly detection (e.g., consecutive failures), hiding trust evaluation mechanism (e.g., software/data obfuscation)

	3. Detection based on direct trust	Incomplete trust threshold, miss detection and false alarm/ Reverse engineering	Optimized trust threshold, dynamic trust threshold, avoidance (e.g., multipath routing), redundancy
Indirect (extended)	4. Collecting indirect information from neighbors	Unreliable information/ False behavior, bad mouthing, conflict behavior attack	Anomaly detection (eliminating erroneous measurement), redundancy (k fault tolerance)
	5. Reputation evaluation based on both direct and indirect information	Unreliable information/ Reverse engineering	Anomaly detection, hiding trust evaluation mechanism (e.g., software/data obfuscation)
	6. Detection based on reputation	Incomplete trust threshold, miss detection and false alarm/ Reverse engineering	Optimized trust threshold, dynamic trust threshold, avoidance (e.g., multipath routing), redundancy

Table 3. Security Vulnerabilities in Trust Mechanism and Defending Approaches

A. *Improving stage 1: behavior monitoring*

1) *Neighbor-based monitoring* As a consequence of limited overhearing distance, a sender S cannot completely monitor misbehaviors of a receiver or multiple colluding attackers. This limitation can be improved upon by increasing S's monitoring coverage with the help of other neighbors who can also contribute in monitoring all forwarding participants' behaviors in a routing path. This approach can reduce several types of colluding attacks. For example, two colluding attackers M1 and M2 located in a routing path S→M1→M2→BS, M2 can drop all packets without being detected by S due to the S's limited overhearing distance. On the other hand, in this approach, M2's misbehaviors can be detected by common good neighbors (called *guard nodes* in [21]) of M1 and M2. Moreover, the guard nodes will easily detect M1's misbehaviors, since they observe that M1 violates trust mechanism because M1 keeps forwarding packets to M2 although M2 keeps dropping all packets received from M1. For another example, guard nodes can detect whether an attacker sent a packet to a non-existing node by trying to contact the non-existing node. In addition, power-adjusting attack can be detected by guard nodes examining whether or not the strength of transmission power is enough to reach to the receiving node.

Several works [2, 14, 21] that used neighbor-based approach have been introduced in order to mitigate selective forwarding attacks. In [14], when an inside attacker drops a packet, a monitoring neighbor (called *monitor node*) alarms it to S and BS and also sends a copied packet to BS along a new routing path that is disjoint with the original routing path.

However, there are some limitations in these approaches. First, they do not address how their approaches can counter M2's selective packet drops against S. If M2 stores enough packets received from multiple nodes in its forwarding buffer, M2 can safely pinpoint S's packets by using a simple scheduling method so as not to trigger neighbor nodes' alert mechanism. To defend against M2's selective forwarding attack, neighbor nodes must be able to figure out which source node is under selective forwarding attack. In addition, a serious problem happens when neighbor nodes falsely accuse good nodes of attackers. In this case, we must have a countermeasure that not only detects selective forwarding attackers but also locates the misbehaving guard nodes.

2) *Acknowledgement-based monitoring:* A multi-hop acknowledgement scheme to detect selective forwarding attacks was proposed by Xio et al [13]. In this approach, This scheme chooses some random nodes (called *checkpoints*) in a routing path in order to report ACKs back to source node S (hop by hop). These randomly chosen nodes will use the same but reversed routing path when they receive a packet. In case a previous checkpoint does not receive ACK from a next checkpoint, it reports an alert ACK to S or BS hop by hop along the same path. Then, S figures out which nodes are malicious or suspicious based on collected ACKs from checkpoints, and then discards them. This approach, however, has some weaknesses. First, while an ACK traverses back to S, inside attackers in the routing path can drop it as they dropped packets. Second, it is unclear how to accurately locate inside attackers. Third, it fails to handle when this checkpoints nodes falsely prosecute good nodes.

B. *Improving stage 2: direct trust evaluation*

1) *Anomaly detection:* To counter intelligent behavior attack, first, we must inhibit inside attackers from obtaining trust related critical information such as trust value and trust evaluation procedure. However, achieving this completely may be difficult because inside attackers may be able to steal that information through reverse engineering. Second, inside attackers must be detected by accurate measurement of trust values of nodes and then classification of the nodes into two groups (*bad* and *good*) based on the trust threshold. There must be certain unique characteristics of inside attackers since their goal must be different to that of normal nodes. Thus, these unique aspects of inside attackers must be captured by the trust model and then they must be considered for trust evaluation.

Consecutive failures: One abnormal characteristic of packet drop attackers, is *consecutive failures* (or *consecutive drops*). If consecutive failures are handled appropriately, it will improve the early detection ability of a trust model for two reasons. First, a certain degree of consecutive failures are generated by most packet drop attacks such as blackhole, grayhole, and on-off attack. Second, the belief that the node generating the *n* consecutive failures is not a attacker/faulty node will grow as the size of consecutive failures n grows based on the following probabilistic reasoning. Assuming that $P[f]$ is the probability that a normal node generates a failure, the probability that the *n* consecutive failures happens ($P[f]n$) decreases exponentially as n grows.

Meanwhile, it is observed that two trust models (beta trust model and entropy trust model) do not address consecutive failures as (1) and (2). Consider the two observations that contain 10 successes and 10 failures: fsfsfsfsfsfsfsfsfsfs and ssssssssssffffffffff. Both models will equally treat them although the latter looks more suspicious due to the recent 10 consecutive failures according to the above reasoning. Moreover, it is often assumed that inside attackers launch attacks after they develop high trust to avoid being easily detected [24].

It can be shown, through a simple analysis, how the two trust models fail to quickly detect a naive inside packet drop attacker. Suppose that a node's trust value is approximately 1 (the node is very trustful) after it successfully forwarded 1000 packets (that is, $s = 1000$), the node starts dropping packets. As the number of consecutive failures *n* goes from 1 to 20, the upper two curves in Fig. 1 show how their trust values *T* drops; Trust values in beta trust model and entropy trust model are calculated by (1) and (2), respectively. Surprisingly, after 20 consecutive failures, the trust values in beta trust model and entropy trust model are 0.979 and 0.927, respectively. Even for a very noisy channel with $P[f] = 0.5$, the event of 20 consecutive drops happens with probability 0.5^{20} ($\approx 10^{-8}$). Therefore, we need to build a new trust model that considers consecutive failures. In the event of consecutive failures, such model will give significant penalty on a node's trust value as shown in the bottom curve in Fig 1.

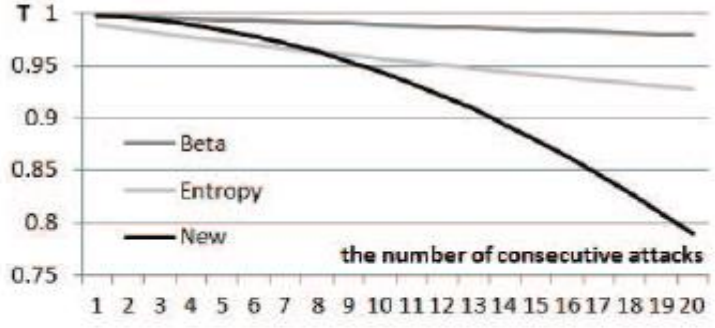

Figure 1 Trust evaluation under consecutive attacks

In this way, we may find *abnormal behavioral characteristics of* inside attackers *that make them distinguishable from normal nodes*. To detect inside attackers in WSNs, various anomaly detection techniques [18, 19, 20] have been used. We can use direct and indirect information together to detect abnormal behaving nodes which are statistically deviated from normal nodes. However, most of these anomaly detection techniques require nontrivial computation cost and message exchanges which result in high power consumption. Therefore, it is crucial to make them very suitable for WSNs.

2) *Hiding trust evaluation function from inside attackers:* If an inside attacker figures out the working of trust evaluation function, it becomes easy for the attacker to estimate its trust values at its neighbors based on its packet drop attack rate. Once the attacker knows its estimated trust values at others, it can intelligently adjust its attack rate never to be detected by its neighbors. Therefore, all critical functions (including source codes) must be hidden appropriately from even the owner (sensor). In fact, a sensor node may not need to know the trust evaluation function or exact trust values to do certain trust-related operations. For example, for trust-based packet forwarding, a sender only needs to pick up a trustful next hop to send its packet to BS via the next hop. A sender does not need to know the exact trust value of the next hop or how the next hop is chosen. That is, an authorized node should be *allowed* to access only necessary information. This can be achieved by using cryptography, authentication, and authorization.

However, there remains a risk that inside attackers may reverse engineer trust evaluation function to figure out how it works and estimate its own trust value at its neighbors in order to avoid being detected. Obfuscation [17, 27] can defend against inside attackers' reverse engineering by making internal software (layout, design, and control) and data ambiguous and hard to interpret by the attackers. In addition, various software protection techniques such as watermarking, application performance degradation, and anti-debugging can be used in order to detect unauthorized access to the software, alter the software when it is accessed in unauthorized ways, and prevent attackers from using a debugger that tracks the execution of software by detecting the use of the debugger, respectively [16, 28].

C. *Improving stage 3: detection based on direct trust*

1) *Optimized trust threshold:* In stage 3, one problem is that of determinining a trust threshold. In Fig. 1, assuming that we use beta trust model, if we simply set the threshold to 0.5, an inside attacker who forwarded 100 packets previously will not be detected even after 100 consecutive

packet drops. Meanwhile, if we set the threshold to 0.99, there will be a high false alarm. Determining the value of threshold may differ depending on applications that we use. For example, we may have a high threshold if the cost introduced from a high false alarm is very low in the application. We should find a trust threshold that maximizes detection rate and minimizes false alarm rate since there is a trade-off between detection and false alarm. A reasonable trust threshold can be determined theoretically or by well-designed simulation by considering our network environment and applications.

2) *Static trust threshold vs. dynamic trust threshold:* A trust threshold can be designed in static manner or dynamic manner. Static trust threshold might be optimal only for limited cases that we consider in the simulation. As a result, it may not be good for unconsidered situations. Meanwhile, dynamic trust threshold that adaptively changes according to situations in our network may have reasonably good results, although it may not be optimal for all situations. However, since dynamic trust threshold will be frequently computed, it must be designed in an energy-efficient way.

D. Improving stage 4 and 5:collecting indirect information from neighbors and reputation evaluation

In stage 4 and 5, for reputation evaluation, a node additionally utilizes indirect information (trust values or observations) from its neighbors. To defend against bad mouthing attack and conflict behavior attack, a couple of methods can be considered as follows.

First, we should consider only trustful indirect information provided by trustful neighbors. This is obvious because the information from distrustful neighbors will corrupt reputation evaluation. Second, indirect information can be weighted according to the trust level of the information provider [4]. Third, redundancy and statistical methods can be used for detecting those attacks. Reputation, which is obtained by both direct and indirect trust, can defend against bad mouthing attack because the attacker's misbehaviors will be different with what other neighbors observed [7]. In addition, if the number of good neighbors is larger than that of bad attackers, the bad mouthing attack can be mitigated and detected by majority voting or some statistical methods. Fourth, using multiple trust values on multiple types of behaviors is recommended in practice since a node might be distrustful for one behavior while it is trustful for another behavior [24]. For example, Sun et al [10] considered a special type of direct trust (recommendation trust) which is evaluated by nodes' past recommendation behaviors. It is calculated as $(sr+1)/(sr+fr+2)$ where sr and fr are the number of good and bad recommendations received from the evaluated node. They showed that considering two types of trust values together better mitigates inside attacks. Finally, we introduce a general principle on how many redundancies we must have in order to defend against k colluding inside attackers disrupting our decision system. In Lamport's Byzantine agreement problem [6], $3k+1$ nodes (redundancies) are required to achieve a reliable agreement by beating k misbehaving faulty nodes by using $2k+1$ correctly behaving nodes. Thus, critical decision functions in trust mechanism must be designed based on this principle.

E. Improving stage 6: detection based on reputation

Note that this part also can be applied to the stage 3.

1) *Avoidance:* Regardless of how elegant detection techniques we have, inside attackers with high trust value can drop a certain portion of packets because of the weaknesses of trust mechanism that we have explained. Therefore, we must have avoidance techniques to ensure that packets eventually reach to BS. Karlof and Wagner [5] mentioned k disjoint multipath routing can completely defend against selective forwarding attacks involving at most k compromised nodes

and still offer some probabilistic protection when there are more than *k* compromised nodes. Several works showing that multipath routing defends against inside attackers' packet dropping can be found in [22, 26]. Similarly, Sun et al [9] introduced multiple data flow scheme using multiple disjoint topologies. In this scheme, a sending node sends its packet through one or more randomly chosen topologies among the pre-established multiple topologies to mitigate selective forwarding attacks.

2) *Trade-off between redundancy and energy:* It is apparent that the more redundancies we have, the more reliable our network is. However, we must keep in mind the redundancies are the cost we must pay. For example, in *n* multipath routing, a sending node first determines *n* disjoint multiple paths from itself to BS and then sends *n* identical packets along the *n* disjoint paths. Consequently, this may introduce at least *n* times of computation complexity and power that a single path routing requires. In addition, the newly introduced workloads such as message exchanges that are required to manage disjoint paths may significantly degrade our network functions [25]. Therefore, we must utilize redundancy energy-efficiently.

CONCLUSION AND FUTURE WORK

In this paper, we discussed the serious threats that insider attacks pose to WSNs even with the presence of trust mechanism and watchdog. We also discussed defending approaches for improving trust mechanism to counter these insider attacks. In the near future, we can design a reliable, energy-efficient trust mechanism for WSNs by considering the identified vulnerabilities and defending approaches in Table 3.

REFERENCES

[1] Azahdeh Faridi et al, "Comprehensive Evaluation of the IEEE 802.15.4 MAC Layer Performance With Retransmissions," IEEE Transactions on Vehicular Technology, Vol.59, No.8, October 2010, pp. 3917-3932.

[2] Tran Hoang Hai and Eui-Nam Huh, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks Using Two-hops Neighbor Knowledge," Seventh International Symposium on Network Computing and Applications (NCA '08), July 2008, pp. 325-331.

[3] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of Attack and Defense Techniques for Reputation Systems," ACM Computing Surveys, Vol 41, Issue 4, 2009.

[4] A. Josang and R. Ismail, "The Beta Reputation System," In Proc. Of the 15th Bled Electronic Commerce Conference, June 2002.

[5] Chris Karlof and David Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," Ad Hoc Networks Journal, Vol.1, Issue 2-3, 2003, pp. 293-315.

[6] Leslie Lamport, Robert Shostak, and Marshall Pease, "The Byzantine Generals Problem," ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, pp. 382-401.

[7] Jie Li, Ruidong Li, and Jien Kato, "Future Trust Management Framework for Mobile Ad Hoc Networks," IEEE Communication Magazine, Vol 46, Issue 4, 2008, pp.108-114.

[8] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker, "Mitigating Routing Misbehavior in Mobile and Ad Hoc Networks," In Proc. Of International Conference on Mobile Computing and Networking (Mobicom), 2000, pp. 255-265.

[9] Hung-Min Sun, Chien-Ming Che, and Ying-Chu Hsiao, "An efficient countermeasure to the selective forwarding attack in wireless sensor network," IEEE Region 10 Conference (TENCON), 2007, pp.1-4.

[10] Yan (Lindsay) Sun, Zhu Han, and K. J. Ray Liu, "Defense of Trust Management Vulnerabilities in Distributed Networks," IEEE Communications Magazine, Vol 46, Issue 2, 2008, pp.112-119.

[11] Denis Trcek, "Trust management in the pervasive computing era," IEEE Security & Privacy, Vol. 9, No. 4, July 2011, pp. 52-55.

[12] Vijay Varadharajan, "A Note on Trust-Enhanced Security", IEEE Security & Privacy, Vol. 7, Issue 3, May/June 2009, pp. 57-59.

[13] Bin Xiao, Bo Yu, Chuanshan Gao, "CHEMAS: Identify suspect nodes in selective forwarding attacks," Journal of Parallel and Distributed Computing, 67, 2007, pp. 1218-1230.

[14] Wang Xin-sheng, Zhan Yong-zhao, Xiong Shu-ming, and Wang Liangmin," Lightweight defense scheme against selective forwarding attacks in wireless sensor networks," Intl. Conf. on Cyber-Enabled Distributed and Knowledge Discovery (CyberC), 2009, pp. 226-232.

[15] Yanli Yu, Keqiu Li, Wanlei Zhou, and Ping Li, "Trust mechanisms in wireless sensor networks: attack analysis and countermeasures," Journal of Network and Computer Applications, Elsevier, 2011, in press.

[16] Martin R. Stytz and James A. Whitaker, "Software Protection: Security's Last Stand?" IEEE Security & Privacy, Vol 1, Issue 1, 2003, pp. 95-98.

[17] David E. Bakken, Rupa Parameswaran, Douglas M. Blough, Ty J. Palmer, and Andy A. Franz, "Data Obfuscation: Anonymity and Desensitization of Usable Data Sets," IEEE Security & Privacy, Vol 2, Issue 6, 2004, pp. 34-41.

[18] Fang Liu, Xiuzhen Cheng, and Dechang Chen, "Insider Attacker Detection in Wireless Sensor Networks," IEEE International Conf. on Computer Communications (INFOCOM), May 2007, pp. 1937-1945.

[19] Miao Xie, Song Han, Biming Tian, and Sazia Parvin, "Anomaly detection in wireless sensor networks: A survey," Journal of Network and Computer Applications 34, 2011, pp. 1302-1325.

[20] Sutharshan Rajasegarar, Christopher Leckie, and Marimuhu Palaniswami, "Anomaly Detection In Wireless Sensor Networks," IEEE Wireless Communications, 2008, pp. 34-40.

[21] Issa Khalil, Saurabh Bagchi, Cristina N. Rotaru, Ness B. Shroff, "UnMask: Utilizing neighbor monitoring for attack mitigation in multihop wireless sensor networks," Ad Hoc Networks, in press, 2009.

[22] Y. Challal, A. Ouadjaout, N. Lasla, M. Bagaa, A. Hadjidj, "Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks," Journal of Network and Computer Applications 34, 2011, pp. 1380-1397.

[23] Brad Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," In Proc. of International Conference on Mobile Computing and Networking (Mobicom), 2000, pp. 243-254.

[24] Javier Lopez, Rodrigo Roman, Isaac Agudo, and Carmen Fernandez- Gago, "Trust management systems for wireless sensor networks: Best practices," Computer Communications, Vol 33, 2010, pp.1086 – 1093.

[25] David R. Raymond and Scott F. Midkiff, "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses," Pervasive computing, 2008, pp. 74-80.

[26] Suk-Bok Lee and Yoon-Hwa Choi, "A Resilient Packet-Forwarding Scheme against Maliciously Packet-Dropping Nodes in Sensor Networks," In Proc. of the fourth ACM workshop on Security of ad hoc and sensor networks (SANS), 2006, pp. 59-69.

[27] Vivek Balachandran and Sabu Emmanuel, "Software Code Obfuscation by Hiding Control Flow Information in Stack," IEEE Int. Workshop on Information Forensics and Security (WIFS), 2011, pp.1-6.

[28] Tang Jiutao and Lin Guoyuan, "Research of Software Protection," Intl. Conf. on Educational and network Technology (ICENT), 2010, pp. 410-413.

[29] Youngho Cho, Gang Qu, Yuanming Wu, "Insider Threats against Trust Mechanism with Watchdog and Defending Approaches in Wireless Sensor Networks", IEEE CS Security and Privacy Workshops 2012, pp. 134-141.