



COPY RIGHT

2017 IJIEMR. Personal use of this material is permitted. Permission from IJIEMR must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. No Reprint should be done to this paper, all copy right is authenticated to Paper Authors

IJIEMR Transactions, online available on 20th July 2017. Link :

<http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-5>

Title: Trust Aware Routing Framework For WSN.

Volume 06, Issue 05, Page No: 1951 – 1959.

Paper Authors

*** P.SEENAMMA, J.SARITHA.**

* Dept of CSE, CVRT.



USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per **UGC Guidelines** We Are Providing A Electronic Bar Code

TRUST AWARE ROUTING FRAMEWORK FOR WSN

*P.SEENAMMA, **J.SARITHA

*PG Scholar, Dept of CSE, CVRT, AP, India.

**Assistant Professor, Dept of CSE, CVRT, AP, India.

ABSTRACT

The multi-hop routing in wireless sensor networks (WSNs) offers little protection against identity deception through replaying routing information. An adversary can exploit this defect to launch various harmful or even devastating attacks against the routing protocols, including *sinkhole* attacks, wormhole attacks and *Sybil* attacks. The situation is further aggravated by mobile and harsh network conditions. Traditional cryptographic techniques or efforts at developing trust-aware routing protocols do not effectively address this severe problem. To secure the WSNs against adversaries misdirecting the multi-hop routing, we have designed and implemented TARF, a robust trust-aware routing framework for dynamic WSNs. Without tight time synchronization or known geographic information, TARF provides trustworthy and energy-efficient route. Most importantly, TARF proves effective against those harmful attacks developed out of identity deception; the resilience of TARF is verified through extensive evaluation with both simulation and empirical experiments on large-scale WSNs under various scenarios including mobile and RF-shielding network conditions. Further, we have implemented a low-overhead TARF module in TinyOS; as demonstrated, this implementation can be incorporated into existing routing protocols with the least effort. Based on TARF, we also demonstrated a proof-of-concept mobile target detection application that functions well against an anti-detection mechanism.

1 INTRODUCTION

Wireless sensor networks (WSNs) [2] are ideal candidates for applications to report detected events of interest, such as military surveillance and forest fire monitoring. A WSN comprises battery-powered sensor nodes with extremely limited processing capabilities. With a narrow radio communication range, a sensor node wirelessly sends messages to a base station via a multi-hop path. However, the multi-hop routing of WSNs often becomes the target of malicious attacks. An attacker may tamper nodes physically, create traffic collision with seemingly valid transmission, drop or misdirect messages in routes, or jam the communication channel by creating radio interference [3]. This paper focuses on the kind of attacks in which adversaries misdirect network traffic by identity

deception through replaying routing information. Based on identity deception, the adversary is capable of launching harmful and hard-to-detect attacks against routing, such as *selective forwarding*, *wormhole* attacks, *sinkhole* attacks and *Sybil* attacks [4]. As a harmful and easy-to-implement type of attack, a malicious node simply replays all the outgoing routing packets from a valid node to forge the latter node's identity; the malicious node then uses this forged identity to participate in the network routing, thus disrupting the network traffic. Those routing packets, including their original headers, are replayed without any modification. Even if this malicious node cannot directly overhear the valid node's wireless transmission, it can

collude with other malicious nodes to receive those routing packets and replay them somewhere far away from the original valid node, which is known as a *wormhole* attack [5]. Since a node in a WSN usually relies solely on the packets received to know about the sender's identity, replaying routing packets allows the malicious node to forge the identity of this valid node. After "stealing" that valid identity, this malicious node is able to misdirect the network traffic. For instance, it may drop packets received, forward packets to another node not supposed to be in the routing path, or even form a transmission loop through which packets are passed among a few malicious nodes infinitely. It is often difficult to know whether a node forwards received packets correctly even with overhearing techniques [4]. *Sinkhole* attacks are another kind of attacks that can be launched after stealing a valid identity. In a *sinkhole* attack, a malicious node may claim itself to be a base station through replaying all the packets from a real base station [6]. Such a fake base station could lure more than half the traffic, creating a "black hole". This same technique can be employed to conduct another strong form of attack - *Sybil* attack [7]: through replaying the routing information of multiple legitimate nodes, an attacker may present multiple identities to the network. A valid node, if compromised, can also launch all these attacks.

The harm of such malicious attacks based on the technique of replaying routing information is further aggravated by the introduction of mobility into WSNs and the hostile network condition. Though mobility is introduced into WSNs for efficient data collection and various applications [8], [9], [10], [11], it greatly increases the chance of interaction between the honest nodes and the attackers. Additionally, a poor network connection causes much difficulty in distinguishing between an attacker and a honest node with

transient failure. Without proper protection, WSNs with existing routing protocols can be completely devastated under certain circumstances. In an emergent sensing application through WSNs, saving the network from being devastated becomes crucial to the success of the application.

Unfortunately, most existing routing protocols for WSNs either assume the honesty of nodes and focus on energy efficiency [12], or attempt to exclude unauthorized participation by encrypting data and authenticating packets. Examples of these encryption and authentication schemes for WSNs include TinySec [13], Spins [14], TinyPK [15], and TinyECC [16]. Admittedly, it is important to consider efficient energy use for battery-powered sensor nodes and the robustness of routing under topological changes as well as common faults in a wild environment. However, it is also critical to incorporate security as one of the most important goals; meanwhile, even with perfect encryption and authentication, by replaying routing information, a malicious node can still participate in the network using another valid node's identity. The gossiping-based routing protocols offer certain protection against attackers by selecting random neighbors to forward packets [17], but at a price of considerable overhead in propagation time and energy use.

In addition to the cryptographic methods, trust and reputation management has been employed in generic ad hoc networks and WSNs to secure routing protocols. Basically, a system of trust and reputation management assigns each node a trust value according to its past performance in routing. Then such trust values are used to help decide a secure and efficient route. However, the proposed trust and reputation management systems for generic ad hoc networks target only relatively

powerful hardware platforms such as laptops and smart phones [18], [19], [20], [21]. Those systems can not be applied to WSNs due to the excessive overhead for resource-constrained sensor nodes powered by batteries. As far as WSNs are concerned, secure routing solutions based on trust and reputation management rarely address the identity deception through replaying routing information [22], [23]. The countermeasures proposed so far strongly depends on either tight time synchronization or known geographic information while their effectiveness against attacks exploiting the replay of routing information has not been examined yet [4].

At this point, to protect WSNs from the harmful attacks exploiting the replay of routing information, we have designed and implemented a robust trust-aware routing framework, TARF, to secure routing solutions in wireless sensor networks. Based on the unique characteristics of resource-constrained WSNs, the design of TARF centers on *trustworthiness* and *energy efficiency*. Though

TARF can be developed into a complete and independent routing protocol, the purpose is to allow existing routing protocols to incorporate our implementation of TARF with the least effort and thus producing a secure and efficient fully-functional protocol. Unlike other security measures, TARF requires neither tight time synchronization nor known geographic information. Most importantly, TARF proves resilient under various attacks exploiting the replay of routing information, which is not achieved by previous security protocols. Even under strong attacks such as *sinkhole* attacks, *wormhole* attacks as well as *Sybil* attacks, and hostile mobile network condition, TARF demonstrates steady improvement in network performance. The effectiveness of TARF is verified through extensive evaluation with simulation and

empirical experiments on large-scale WSNs. Finally, we have implemented a ready-to-use TARF module with low overhead, which as demonstrated can be integrated into existing routing protocols with ease; the demonstration of a proof-of-concept mobile target detection program indicates the potential of TARF in WSN applications.

We start by stating the design considerations of TARF in Section 2. Then we elaborate the design of TARF in Section 3, including the routing procedure as well as the *EnergyWatcher* and *TrustManager* components. In Section 4, we present the simulation results of TARF against various attacks through replaying routing information in static, mobile and RF-shielding conditions. Section 5 further presents the implementation of TARF, empirical evaluation at a large sensor network and a resilient proof-of-concept mobile target detection application based on TARF. Finally, we discuss the related work in Section 6 and conclude this paper in Section 7.

2 DESIGN CONSIDERATIONS

Before elaborating the detailed design of TARF, we would like to clarify a few design considerations first, including certain assumptions in Section 2.1 and the goals in Section 2.3.

2.1 Assumptions

We target secure routing for data collection tasks, which are one of the most fundamental functions of WSNs. In a data collection task, a sensor node sends its sampled data to a remote base station with the aid of other intermediate nodes, as shown in Figure 1. Though there could be more than one base station, our routing approach is not affected by the number of base stations; to simplify our discussion, we assume that there is only one base station. An

adversary may forge the identity of any legal node through replaying that node's outgoing routing packets and spoofing the acknowledgement packets, even remotely through a *wormhole*.

Additionally, to merely simplify the introduction of TARF, we assume no data aggregation is involved.

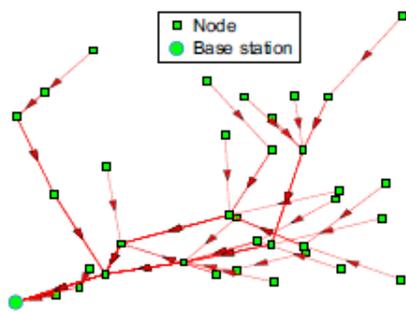


Fig. 1. Multi-hop routing for data collection of a WSN.

Nonetheless, our approach can still be applied to cluster-based WSNs with static clusters, where data are aggregated by clusters before being relayed [24]. Cluster-based WSNs allows for the great savings of energy and bandwidth through aggregating data from children nodes and performing routing and transmission for children nodes. In a cluster-based WSN, the cluster headers themselves form a sub-network; after certain data reach a cluster header, the aggregated data will be routed to a base station only through such a sub-network consisting of the cluster headers. Our framework can then be applied to this sub-network to achieve secure routing for cluster-based WSNs. TARF may run on cluster headers only and the cluster headers communicate with their children nodes directly since a static cluster has known relationship between a cluster header and its children nodes, though any link-level security features may be further employed.

Finally, we assume a data packet has at least the following fields: the sender id, the sender sequence number, the next-hop node id (the receiver in this one-hop transmission), the source id (the node that initiates the data), and the source's sequence number. We insist that the source node's information should be included for the following reasons because that allows the base station to track whether a data packet is delivered. It would cause too much overhead to transmit all the one-hop information to the base station. Also, we assume the routing packet is sequenced.

2.2 Authentication Requirements

Though a specific application may determine whether data encryption is needed, TARF requires that the packets are properly authenticated, especially the broadcast packets from the base station. The broadcast from the base station is asymmetrically authenticated so as to guarantee that an adversary is not able to manipulate or forge a broadcast message from the base station at will. Importantly, with authenticated broadcast, even with the existence of attackers, TARF may use TrustManager (Section 3.4) and the received broadcast packets about delivery information (Section 3.2.1) to choose trustworthy path by circumventing compromised nodes. Without being able to physically capturing the base station, it is generally very difficult for the adversary to manipulate the base station broadcast packets which are asymmetrically authenticated. The asymmetric authentication of those broadcast packets from the base station is crucial to any successful secure routing protocol. It can be achieved through existing asymmetrically authenticated broadcast schemes that may require loose time synchronization. As an example, μ TESLA [14] achieves asymmetric authenticated broadcast through a symmetric cryptographic algorithm and a loose delay schedule to disclose the

keys from a key chain. Other examples of asymmetric authenticated broadcast schemes requiring either loose or no time synchronization are found in [25], [26].

Considering the great computation cost incurred by a strong asymmetric authentication scheme and the difficulty in key management, a regular packet other than a base station broadcast packet may only be moderately authenticated through existing symmetric schemes with a limited set of keys, such as the message authentication code provided by TinySec [13]. It is possible that an adversary physically captures a non-base legal node and reveals its key for the symmetric authentication [27]. With that key, the adversary can forge the identity of that non-base legal node and joins the network “legally”. However, when the adversary uses its fake identity to falsely attract a great amount of traffic, after receiving broadcast packets about delivery information, other legal nodes that directly or indirectly forwards packets through it will start to select a more trustworthy path through TrustManager (Section 3.4).

2.3 Goals

TARF mainly guards a WSN against the attacks mis-directing the multi-hop routing, especially those based on identity theft through replaying the routing information. This paper does not address the denial-of-service (DoS) [3] attacks, where an attacker intends to damage the network by exhausting its resource. For instance, we do not address the DoS attack of congesting the network by replaying numerous packets or physically jamming the network. TARF aims to achieve the following desirable properties:

High Throughput *Throughput* is defined as the ratio of the number of all data packets delivered to the base station to the number

of all sampled data packets. In our evaluation, *throughput* at a moment is computed over the period from the beginning time (0) until that particular moment. Note that single-hop re-transmission may happen, and that duplicate packets are considered as one packet as far as *throughput* is concerned. *Throughput* reflects how efficiently the network is collecting and delivering data. Here we regard high *throughput* as one of our most important goals.

Energy Efficiency Data transmission accounts for a major portion of the energy consumption. We evaluate energy efficiency by the average energy cost to successfully deliver a unit-sized data packet from a source node to the base station. Note that link-level re-transmission should be given enough attention when considering energy cost since each re-transmission causes a noticeable increase in energy consumption. If every node in a WSN consumes approximately the same energy to transmit a unit-sized data packet, we can use another metric *hop-per-delivery* to evaluate energy efficiency. Under that assumption, the energy consumption depends on the number of hops, i.e. the number of one-hop transmissions occurring. To evaluate how efficiently energy is used, we can measure the average hops that each delivery of a data packet takes, abbreviated as *hop-per-delivery*.

Scalability & Adaptability TARF should work well with WSNs of large magnitude under highly dynamic contexts. We will evaluate the scalability and adaptability of TARF through experiments with large-scale WSNs and under mobile and hash network conditions. Here we do not include other aspects such as latency, load balance, or fairness. Low latency, balanced network load, and good fairness requirements can be enforced in specific routing protocols incorporating TARF.

3 DESIGN OF TARF

TARF secures the multi-hop routing in WSNs against intruders misdirecting the multi-hop routing by evaluating the trustworthiness of neighboring nodes. It identifies such intruders by their low trustworthiness and routes data through paths circumventing those intruders to achieve satisfactory *throughput*. TARF is also energy-efficient, highly scalable, and well adaptable. Before introducing the detailed design, we first introduce several necessary notions here.

Neighbor For a node N , a neighbor (neighboring node) of N is a node that is reachable from N with one-hop wireless transmission.

Trust level For a node N , the trust level of a neighbor is a decimal number in $[0, 1]$, representing N 's opinion of that neighbor's level of trustworthiness. Specifically, the trust level of the neighbor is N 's estimation of the probability that this neighbor correctly delivers data received to the base station. That trust level is denoted as T in this paper. **Energy cost** For a node N , the energy cost of a neighbor is the average energy cost to successfully deliver a unit-sized data packet with this neighbor as its next-hop node, from N to the base station. That energy cost is denoted as E in this paper

3.1 Overview

For a TARF-enabled node N to route a data packet to the base station, N only needs to decide to which neighboring node it should forward the data packet considering both the trustworthiness and the energy efficiency. Once the data packet is forwarded to that next-hop node, the remaining task to deliver the data to the base station is fully delegated to it, and N is totally unaware of what routing decision its next-hop node makes. N maintains a

neighborhood table with trust level values and energy cost values for certain known neighbors. It is sometimes necessary to delete some neighbors' entries to keep the table size acceptable. The technique of maintaining a neighborhood table of a moderate size is demonstrated by Woo, Tong and Culler [28]; TARF may employ the same technique.

In TARF, in addition to data packet transmission, there are two types of routing information that need to be exchanged: broadcast messages from the base station about data delivery and energy cost report messages from each node. Neither message needs acknowledgement. A broadcast message from the base station is flooded to the whole network. The freshness of a broadcast message is checked through its field of source sequence number. The other type of exchanged routing information is the energy cost report message from each node, which is broadcast to only its neighbors once. Any node receiving such an energy cost report message will not forward it. For each node N in a WSN, to maintain such a neighborhood table with trust level values and energy cost values for certain known neighbors, two components, *EnergyWatcher* and *TrustManager*, run on the node (Figure 2). *EnergyWatcher* is responsible for recording the energy cost for each known neighbor, based on N 's observation of one-hop transmission to reach its neighbors and the energy cost report from those neighbors. A compromised node may falsely report an extremely low energy cost to lure its neighbors into selecting this compromised node as their next-hop node; however, these TARF-enabled neighbors eventually abandon that compromised next-hop node based on its low trustworthiness as tracked by *TrustManager*. *TrustManager* is responsible for tracking trust level values of neighbors based on network loop discovery and broadcast messages from the base station about data delivery. Once N is able to decide its next-

hop neighbor according to its neighborhood table, it sends out its energy report message: it broadcasts to all its neighbors its energy cost to deliver a packet from the node to the base station. The energy cost is computed as in Section 3.3 by *EnergyWatcher*. Such an energy cost report also serves as the input of its receivers' *EnergyWatcher*.

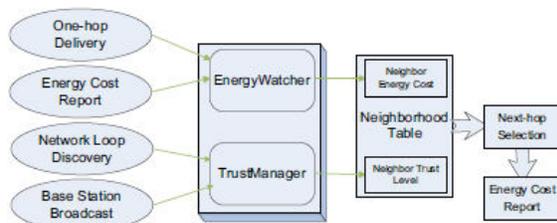


Fig. 2. Each node selects a next-hop node based on its neighborhood table, and broadcast its energy cost within its neighborhood. To maintain this neighborhood table, *EnergyWatcher* and *TrustManager* on the node keep track of related events (on the left) to record the energy cost and the trust level values of its neighbors.

3.2 Routing Procedure

TARF, as with many other routing protocols, runs as a periodic service. The length of that period determines how frequently routing information is exchanged and updated. At the beginning of each period, the base station broadcasts a message about data delivery during last period to the whole network consisting of a few contiguous packets (one packet may not hold all the information). Each such packet has a field to indicate how many packets are remaining to complete the broadcast of the current message. The completion of the base station broadcast triggers the exchange of energy report in this new period. Whenever a node receives such a broadcast message from the base station, it knows that the most recent period has ended

and a new period has just started. No tight time synchronization is required for a node to keep track of the beginning or ending of a period. During each period, the *EnergyWatcher* on a node monitors energy consumption of one-hop transmission to its neighbors and processes energy cost reports from those neighbors to maintain energy cost entries in its neighborhood table; its *TrustManager* also keeps track of network loops and processes broadcast messages from the base station about data delivery to maintain trust level entries in its neighborhood table.

To maintain the stability of its routing path, a node may retain the same next-hop node until the next fresh broadcast message from the base station occurs. Meanwhile, to reduce traffic, its energy cost report could be configured to not occur again until the next fresh broadcast message from the base station. If a node does not change its next-hop node selection until the next broadcast message from the base station, that guarantees all paths to be loop-free, as can be deducted from the procedure of next-hop node selection. However, as noted in our experiments, that would lead to slow improvement in routing paths. Therefore, we allow a node to change its next-hop selection in a period when its current next-hop node performs the task of receiving and delivering data poorly.

Next, we introduce the structure and exchange of routing information as well as how nodes make routing decisions in TARF.

3.2.1 Structure and Exchange of Routing Information

A broadcast message from the base station fits into at most a fixed small number of packets. Such a message consists of some pairs of <node id of a source node, an undelivered

sequence interval $[a, b]$ with a significant length, \langle node id of a source node, minimal sequence number received in last period, maximum sequence number received in last period \rangle , as well as several node id intervals of those without any delivery record in last period. To reduce overhead to an acceptable amount, our implementation selects only a limited number of such pairs to broadcast (Section 5.1) and proved effective (Section 5.3, 5.4). Roughly, the effectiveness can be explained as follows: the fact that an attacker attracts a great deal of traffic from many nodes often gets revealed by at least several of those nodes being deceived with a high likelihood. The undelivered sequence interval $[a, b]$ is explained as follows: the base station searches the source sequence numbers received in last period, identifies which source sequence numbers for the source node with this id are missing, and chooses certain significant interval $[a, b]$ of missing source sequence numbers as an undelivered sequence interval. For example, the base station may have all the source sequence numbers for the source node 2 as $\{109, 110, 111, 150, 151\}$ in last period. Then $[112, 149]$ is an undelivered sequence interval; $[109, 151]$ is also recorded as the sequence boundary of delivered packets. Since the base station is usually connected to a powerful platform such as a desktop, a program can be developed on that powerful platform to assist in recording all the source sequence numbers and finding undelivered sequence intervals.

Accordingly, each node in the network stores a table of \langle node id of a source node, a forwarded sequence interval $[a, b]$ with a significant length \rangle about last period. The data packets with the source node and the sequence numbers falling in this forwarded sequence interval $[a, b]$ have already been forwarded by this node. When the node receives a broadcast message about data

delivery, its *TrustManager* will be able to identify which data packets forwarded by this node are not delivered to the base station. Considering the overhead to store such a table, old entries will be deleted once the table is full. Once a fresh broadcast message from the base station is received, a node immediately invalidates all the existing energy cost entries: it is ready to receive a new energy report from its neighbors and choose its new next-hop node afterwards. Also, it is going to select a node either after a timeout is reached or after it has received an energy cost report from some highly trusted candidates with acceptable energy cost. A node immediately broadcasts its energy cost to its neighbors only after it has selected a new next-hop node. That energy cost is computed by its *EnergyWatcher* (see Section 3.3).

A natural question is which node starts reporting its energy cost first. For that, note that when the base station is sending a broadcast message, a side effect is that its neighbors receiving that message will also regard this as an energy report: the base station needs 0 amount of energy to reach itself. As long as the original base station is faithful, it will be viewed as a trustworthy candidate by *TrustManager* on the neighbors of the base station. Therefore, those neighbors will be the first nodes to decide their next-hop node, which is the base station; they will start reporting their energy cost once that decision is made.

7 CONCLUSIONS

We have designed and implemented TARF, a robust trust-aware routing framework for WSNs, to secure multi-hop routing in dynamic WSNs against harmful attackers exploiting the replay of routing information. TARF focuses on trustworthiness and energy efficiency, which are vital to the survival of a WSN in a hostile environment. With the idea

of trust management, TARF enables a node to keep track of the trustworthiness of its neighbors and thus to select a reliable route. Our main contributions are listed as follows. (1) Unlike previous efforts at secure routing for WSNs, TARF effectively protects WSNs from severe attacks through replaying routing information; it requires neither tight time synchronization nor known geographic information. (2) The resilience and scalability of TARF is proved through both extensive simulation and empirical evaluation large-scale WSNs; the evaluation involves both static and mobile settings, hostile network conditions, as well as strong attacks such as *wormhole* attacks and *Sybil* attacks. (3) We have implemented a ready-to-use TinyOS module of TARF with low overhead; as demonstrated in the paper, this TARF module can be integrated into existing routing protocols with the least effort, thus producing secure and efficient fully-functional protocols. (4) Finally, we demonstrate a proof-of-concept mobile target detection application that is built on top of TARF and is resilient in the presence of an anti-detection mechanism; that indicates the potential of TARF in WSN applications.

REFERENCES

- [1] G. Zhan, W. Shi, and J. Deng, "Tarf: A trust-aware routing framework for wireless sensor networks," in *Proceeding of the 7th European Conference on Wireless Sensor Networks (EWSN'10)*, 2010.
- [2] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann Publishers, 2004.
- [3] A. Wood and J. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, Oct 2002.
- [4] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.
- [5] M. Jain and H. Kandwal, "A survey on complex wormhole attack in wireless ad hoc networks," in *Proceedings of International Conference on Advances in Computing, Control, and Telecommunication Technologies (ACT '09)*, 28-29 2009, pp. 555 –558.
- [6] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Launching a sink-hole attack in wireless sensor networks; the intruder side," in *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB '08)*, 12-14 2008, pp. 526 –531.
- [7] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *Proc. of the 3rd International Conference on Information Processing in Sensor Networks (IPSN'04)*, Apr. 2004.
- [8] L. Bai, F. Ferrese, K. Ploskina, and S. Biswas, "Performance analysis of mobile agent-based wireless sensor network," in *Proceedings of the 8th International Conference on Reliability, Maintainability and Safety (ICRMS 2009)*, 20-24 2009, pp. 16 –19.
- [9] L. Zhang, Q. Wang, and X. Shu, "A mobile-agent-based middleware for wireless sensor networks data fusion," in *Proceedings of Instrumentation and Measurement Technology Conference (I2MTC '09)*, 5-7 2009, pp. 378 –383.
- [10] W. Xue, J. Aiguo, and W. Sheng, "Mobile agent based moving target methods in wireless sensor networks," in *IEEE International Symposium on Communications and Information Technology (ISCIT 2005)*, vol. 1, 12-14 2005, pp. 22 – 26.