Malicious Node Detection in Wireless Sensor Networks

Waldir Ribeiro Pires Júnior Thiago H. de Paula Figueiredo Hao Chi Wong
Antonio A.F. Loureiro
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, MG, Brazil
{wpjr,thiagohp,hcwong,loureiro}@dcc.ufmg.br

Abstract

This work provides a solution to identify malicious nodes in wireless sensor networks through detection of malicious message transmissions in a network. A message transmission is considered suspicious if its signal strength is incompatible with its originator's geographical position. We provide protocols for detecting suspicious transmissions – and the consequent identification of malicious nodes – and for disseminating this information in the network. We evaluate the detection rate and the efficiency of our solution along a number of parameters.

1. Introduction

A wireless sensor network (WSN) consists of a set of compact and automated devices called sensing nodes. A sensing node is a computational device that has memory, battery, processor, transceiver, and a sensing device. The Berkeley MICA Mote [4, 1], SmartDust [7, 8, 12], and CotsDust [5] are examples of such nodes. These nodes are distributed across an area and communicate among themselves, forming an ad hoc network. Sensor networks contain special nodes that process and store the information collected by the network; they are called sink nodes. Communication between two nodes is performed in multiple hops if they are not within each other's transmission range.

Wireless sensor networks can collect data from the environment where they are embedded. The data are often first processed by the sensor nodes and then sent over non-secure channels to the sink node for further processing. Some of the applications envisioned for sensor networks are environmental monitoring, infrastructure management, public safety, medical, home and office security, transportation, and battlefield surveillance. Given their criticality, these applications are likely to be attacked.

There are a number of ways one can attack a WSN. For example, one can spoof the various fields of a message while it is in transit, in such a way that what the recipient receives is an altered copy of the original message. One can also tamper with a node (its hardware and/or software), so as to alter its behavior. Different types of attacks will require different types of countermeasures.

In this work, we focus on two types of attacks: HELLO flood attacks [9] and wormhole attacks [6]. HELLO messages are used in many protocols by nodes that want to announce their presence and proximity to their neighbors. Most of these protocols rely on the assumption that a node A is within the radio transmission range of another node B if A is able to receive messages from B. In a HELLO flood attack, a malicious node may try to transmit a message with an abnormally high power so as to make all nodes believe that it is their neighbor.

Wormhole attacks can be described in the following steps. An adversary A tunnels a message received to a second adversary B in a distant part of the network using a low-latency out-of-band channel. B then retransmits the message exactly as received to the nodes in its neighborhood. An immediate result of a wormhole attack is that nodes that hear the transmission from B are tricked into thinking that they are neighbors of whichever node originated the message (this node is most likely located in a distant part of the network).

Both the HELLO flood attack and the wormhole attack are typically carried out to compromise route establishment in a network. For example, a malicious node that broadcasts a routing beacon with an extra high power could lead a large number of nodes to attempt to use it as their next hop in their route to the sink. But those sufficiently far away would be simply sending their messages into the oblivion. A similar scenario results from a wormhole attack. A malicious node could convince nodes that are normally multiple hops from the sink node that they are just one hop away. These nodes would try to send their packets directly to the sink node,

which would not be able to hear them.

Hu, Perrig, and Johnson have proposed a countermeasure for wormhole attacks in ad hoc networks [6]. They introduced the concept of a packet leash, which is a piece of additional information added to standard packets to restrict its maximum allowed travel distance. Two types of packet leashes were proposed: geographical and temporal. The former ensures that the recipient of a packet is within a certain distance from the sender. The latter limits the lifetime of a packet. Both rely on some kind of clock synchronization between nodes. Because clock synchronization is resource demanding, and, thus, packet leashes have limited applicability in wireless sensor networks.

In this work, we propose a mechanism based on signal strength and geographical information for detecting malicious nodes staging HELLO flood and wormhole attacks. The idea is to compare the signal strength of a reception with its expected value, calculated using geographical information and the pre-defined transceiver specification. A protocol for disseminating information about detection of malicious nodes is also proposed. The detection rate of our solution depends on a number of parameters. We evaluate their correlation through simulation.

The rest of this paper is organized as follows. In Section 2, we present our model and our detection scheme. In Section 3, we present the various parameters which our detection scheme depends on, and whose impact on the detection rate we investigate in this work. We describe the simulation we used to evaluate our work and discuss the results in Section 4. Finally, in Section 5 we present our concluding remarks and future work.

2. Suspicious node detection by signal strength

In this section, we describe our approach for detecting suspicious messages and suspicious nodes based on signal strength.

2.1. The model

In this work, we assume WSNs are homogeneous (all network nodes contain the same hardware and software configuration), symmetric (node A can only communicate with node B if and only if B can communicate with A), and static (network nodes do not move after deployment). In particular, the radio transceivers of all members of the network operate under the same configuration throughout the lifetime of the network (e.g., transmission power, antenna height, and antenna gain).

All nodes are uniquely identified, and know their own geographical position, which can be obtained using a positioning system such as the GPS. The value of a node's geographical position as well as its identifier are included in each of the messages it sends. We assume that message exchanges in the network are protected against tampering (using some cryptographic mechanism, for example) [11].

We further assume that radio propagation follows a well defined model, such as the Free Space Model and the Two-Ray Ground Model [10], which specify how the values of transmission power, received signal strength and distance between the transmitter and the receiver relate to each other. As an example, the Two-Ray Ground propagation model (Equation 1) makes the assumption that a signal sent from one node does not arrive at another node through a unique path (a straight line), but eventually also through a reflection in the ground.

$$P_r = \frac{P_t \times G_t \times G_r \times h_t^2 \times h_r^2}{d^4 \times L} \quad (1)$$

In Equation 1, P_r is the received signal power in Watts, P_t is the transmission power also in Watts, G_t is the transmitter antenna gain, G_r is the receiver antenna gain, h_t is the transmitter antenna height in meters, h_r is the receiver antenna height in meters, d is the distance between the receiver and transmitter in meters and L is the system loss (a constant). A signal is only detected by a receiving node if the received signal power P_r is equal or greater than the received signal power threshold P_m.

We also assume that the signal strength of a received signal can be easily acquired from a transceiver. The Chipcon SmartRF CC1000 [3] transceiver, used in the latest MICA Motes series [2], for example, has a built-in RSSI (Received Signal Strength Indicator) giving an analogue output signal at its RSSI pin. When the RSSI function is enabled, the output current of this pin is inversely proportional to the input signal level. The voltage at the pin can be measured by a analog/digital converter, which is the only additional hardware required. Formulas 2 and 3 specify the received signal strengths P_r, in dBm, when the transceiver is operating at 433 MHz and 868 MHz, respectively. V_{RSSI} is the voltage measured at the RSSI pin.

$$P_r = -51.3 \times V_{RSSI} - 49.2 \quad (2)$$

$$P_r = -50.0 \times V_{RSSI} - 45.5 \quad (3)$$

Finally, we assume that malicious nodes are capable of HELLO flood attacks and wormhole attacks only.

In what follows, a transmission is malicious if the geographical position included in the corresponding message is made up or was transmitted with a power that differs from the one agreed upon by all the other nodes in the system. A node is malicious if it broadcast a malicious transmission.

Upon receiving a message, a node can classify it as suspicious or unsuspicious, depending on whether the node thinks the transmission is malicious. Given that this classification (suspicious vs. unsuspicious) is done locally at

the receiving node, malicious transmissions may not always be classified as suspicious (false negatives), and and non-malicious transmissions may be classified as suspicious (false positives).

2.2. Suspicious message detection by signal strength (SMDSS)

Under the model described above, any node can obtain two values on any transmission it hears. The first value is the expected signal strength of the received signal, which can be computed using the transmission power that was agreed upon for message transmissions in the system and the distance between the node that hears the transmission and the source of the transmission itself. The second value is the actual signal strength detected at the listener's transceiver.

In a system where all is well, the two values should match. The same would not be true in most cases, however, if the system is under a HELLO flood attack or a wormhole attack. We make use of this fact to identify suspicious messages in the system.

In our proposed scheme, all transmissions in the network are subject to scrutiny: all nodes monitor all transmissions they hear. Concretely, the following protocol is run locally in each sensor node. For each transmission a node hears, it compares the expected and the actual signal strengths of the received signal, independently of whether it is the intended recipient of the transmission. When the difference between both is greater than a given threshold, the message is regarded as suspicious.

Each node also keeps a local table containing the "reputation" of other nodes in the system. Each entry contains the node id, the number of suspicious votes, and the number of unsuspicious votes.

After checking the suspiciousness of a received message, the node updates its table accordingly: if the message is suspicious, it increases the message originator's suspicious count by one; otherwise, the unsuspicious count is increased. Note that the message's originator can be determined, given that its id is included in the message.

If the message is suspicious, the node takes a further action: it disseminates this information among its neighbors. We describe the dissemination protocol in Section 2.3. A suspicious message is discarded by its intended recipient and not acted upon.

2.3. Suspicious node information dissemination protocol (SNIDP)

Upon detecting a suspicious message, a node A broadcasts the identity of the sender S to its neighbors, informing them that S is suspicious. This broadcast also works as an inquiry: those that hear this broadcast (e.g., B) should reply with their opinion (suspicious or unsuspicious) of S. B determines its reply the following way:

- If B is not S's neighbor, i.e., B does not hear transmissions from S, it does not respond;
- If B is a neighbor of S, i.e., B hears transmissions from S, then it responds with "suspicious" if the suspicious count for S in its table is greater than its unsuspicious count; otherwise it responds with "unsuspicious".

A collects all the replies and updates its table accordingly: for each "suspicious" vote it receives, it increases its own suspicious count for S by one; and similarly with "unsuspicious" votes.

Note that B's reply is a broadcast. This means that it could be heard by all its neighbors, including those that are neither S's neighbor nor A's neighbor (i.e., nodes that did not hear either S's malicious transmission or A's broadcast to disseminate the fact that S is suspicious). They all update their tables accordingly.

We clarify a number of our design decisions. First, the SNIDP protocol is executed only when a suspicious message is detected. The assumption here is that, in normal circumstances (we are being optimistic here), all transmissions will be unsuspicious, and the network do not need to incur the overhead of the SNIDP protocol. Second, one might argue that a single "suspicious" vote would be more than enough to render a node suspicious. This reasoning would make our scheme vulnerable to malicious nodes that can also lie. A malicious node could disseminate the false information that a regular node C is suspicious and, with this single vote, effectively eliminate C from the network, potentially causing a massive denial-of-service in significant parts of the network. Our requirement that a node be considered suspicious only if its suspicious count outnumbers its unsuspicious count makes our scheme more robust against this type of attack.

One might argue that this last design decision could potentially increase the number of false negatives (malicious nodes that are not detected as suspicious) in our system. We believe that this will not be the case, because in most cases, a malicious transmission can fool only a single target among all the neighbors of the malicious node. This means that there will be many more nodes broadcasting warnings saying the malicious node is suspicious than those saying it is not.

3. Evaluation

Our scheme depends on a number of parameters. In this section we investigate how variations of these various pa-

rameters impact the detection rates of our scheme. Four parameters are evaluated in this work: network density, transmission power multiplier of the malicious node, message check probability, and maximum ratio difference. We discuss each in turn.

Network density determines directly the number of neighbors a node will have. Given that our suspicious node detection depends on exchange of information among neighbors, this parameter should have a role to play in the detection rate of suspicious nodes.

Sensor nodes are resource-constrained, and a cost is incurred by each message that a node receives and checks. Decreasing the number of messages checked would decrease the overall resource consumption. Message check probability C determines the probability that a transmission will be checked by a node that hears it. For each transmission, a number c between 0 and 1, given by a uniform random variable, will determine whether a node will check the message. The message is checked only if $c < C$.

Maximum ratio difference R determines how much the received signal strength P_r can differ from the expected received signal strength P_e without the transmission being classified as suspicious. Given a signal, its ratio difference r is defined by Equation 4:

$$r = 1 - \left(\frac{min(P_r, P_e)}{max(P_r, P_e)} \right) \quad (4)$$

where $min(a, b) = a$ if $b > a$, and $min(a, b) = b$ otherwise; $max(a, b) = b$ if $b > a$, and $max(a, b) = b$ otherwise. For each message that a node receives, the message is classified as suspicious if $r > R$. This parameter is used to deal with precision issues concerning the measurement of signal strengths and the determination of a node's geographical position.

We use the Two-Ray Ground propagation model (described in Section 2.1) in this study. Table 1 shows the values we assume for the transceivers of non-malicious nodes [3]. Transceivers of malicious nodes have the same characteristics except for the transmission power, which we specify as the regular transmission power multiplied by the transmission power multiplier. Depending on the value of its multiplier, a malicious transmission may or may not be detected as suspicious.

We consider two types of scenarios in our evaluations: unfocused and focused. In an unfocused scenario, all nodes, including the malicious one, are turned on at the same time, and broadcast one HELLO message each. Malicious nodes send their messages at a power other than the one used by the members of the network, but do not fine tune it in any other way (this is why the scenario is called unfocused). We use this scenario to model cases where malicious nodes manage to be present right at the deployment and initialization of the network.

Transmission power P_t	-5 dBm
Transmitter antenna gain G_t	1.0
Receiver antenna gain G_r	1.0
Transmitter antenna height h_t	0.05 m
Receiver antenna height h_r	0.05 m
Received signal power threshold P_m	-104 dBm
System loss L	1.0

Table 1. Reliable node parameters

In a focused scenario, malicious nodes start their activities after the network has been deployed and initialized. That is, a malicious node starts transmitting only after all members have been turned on and sent a HELLO message to each other. In this case, the malicious node chooses a target victim (that is why this case is called focused), makes up a position that is within the victim's radio range, and fine tunes its transmission power accordingly, so as to avoid raising suspicion. The right transmission power to use can be obtained straightforwardly using expression 1.

We evaluate our scheme through simulation.

4. Simulation and results

In this section, we present the simulation model and results of our work.

4.1. Simulation model

We developed a wireless sensor network simulator to create an environment to evaluate our work. It is a discrete event simulator written in Java. A network generator was built, which generates networks comprised of n nodes plus one malicious node, all located in an $S \times S$ square field. Each node has randomized x and y coordinates. No two different nodes share the same coordinates. Networks with $50, 100, 150, \ldots, 500$ nodes in 179 $\times$ 179 m^2 fields were generated and used as input to the simulator. For each network with a given number of nodes, 200 network topologies were created. As all networks we consider are in a 179 $\times$ 179 m^2 field, the density is measured in number of nodes. Fig. 1 shows the average number of neighbors of a node for each network density.

We ran the simulation on each generated input file for both scenarios (focused and unfocused) with different combinations of values for each of the parameters described in Section 3. Each parameter has a range of values it can assume (see Table 3), one of which is the default value (Table 2). In each run of the simulation, we set three parameters to their default values while varying the value of the fourth.

In the unfocused scenario, we ran 200 different simulations for each combination of parameters, one for each net-

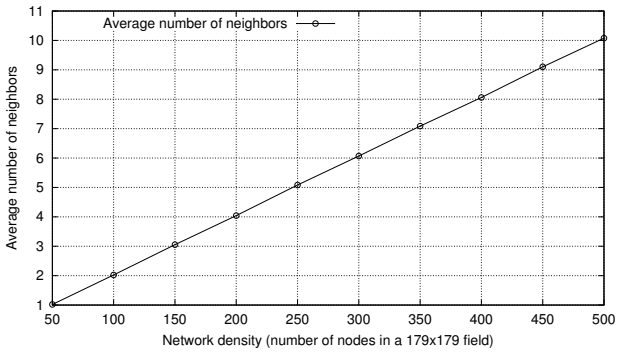

Figure 1. Average number of neighbors vs. network density

Parameter	Default value
Network density	200 nodes in a 179×179 m^2 field
Malicious node multiplier	2
Maximum ratio difference	0.3
Message check probability	1.0

Table 2. Default parameters

work topology generated. In the focused scenario, we ran five simulations for each network topology. In each simulation, the malicious node targets its transmission to a different node.

Right before a malicious transmission starts, all statistics are reset in order to evaluate only the number of message transmissions triggered by the detection of a suspicious message.

4.2. Results

In the following, we present and discuss the simulation results. All values in the graphs are the average value of multiple runs for a given set of parameter values. In each case, we investigate the malicious message detection rate (the inverse of the ratio between the total number of receptions of malicious messages and the number of those considered suspicious) as well as malicious node detection rate (the inverse of the ratio between the total number of nodes that has heard of the malicious node and the number of those that consider it as suspicious).

Parameter	Values
Network density	50, 100, . . ., 500 nodes
Malicious node multiplier	1.1, 1.2, . . ., 2.0
Maximum ratio difference	0.1, 0.2, . . ., 0.9
Message check probability	0.1, 0.2, . . ., 1.0

Table 3. Parameter values evaluated

As expected, no one single non-malicious transmission was classified as suspicious.

4.2.1 Detection rate vs. network density

In Figure 2, we show how the malicious message detection rate correlates with network density. Our simulation shows that the results are quite similar for the two scenarios. In the unfocused case, all malicious messages are classified as suspicious. In the focused case, a small fraction of malicious message receptions are not classified as suspicious. This happens because for each malicious transmission the target victim is actually fooled.

To be sure, rarely some other nodes are also fooled. But this happens only when the target node and the one that is also fooled have almost the same distance to the made up position of the malicious node as well as to its actual position.

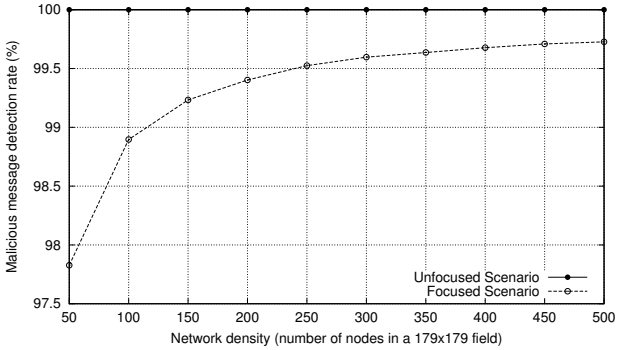

Figure 2. Percentage of malicious message receptions detected vs. network density

In both scenarios, all nodes that are within the radio range of the malicious node heard its transmissions and, using the SNIDP protocol, are able to conclude that it is suspicious. This result holds for all the network densities we considered. Given the uniformity of the result, we omit the corresponding graph here.

4.2.2 Detection rate vs. transmission power multiplier

Malicious message detection rate and transmission power multiplier used by the malicious node (in the unfocused case) have a fairly simple correlation. If the multiplier is above approximately 1.43, all malicious transmissions are regarded as suspicious by everyone that hears the transmissions; otherwise the result is diametrically opposite (none

of the malicious transmissions are regarded as suspicious by anyone).

4.2.3 Detection rate vs. maximum ratio difference

The results concerning the maximum ratio difference are summarized in Figures 3 and 4.

The impact of this parameter in the two scenarios is almost identical. In the unfocused case, the malicious message detection rate is 1, when the maximum ratio difference is less than or equal to 0.4; while, for the focused case, the detection rate is slightly under 1 for the same values of maximum ratio difference.

The results for malicious node detection (Fig. 4) are similar.

Note that the scheme works well even when the maximum ratio difference is high (0.4). This means that neither the value of a reception's signal strength nor the value of a node's geographical position need to have high precision.

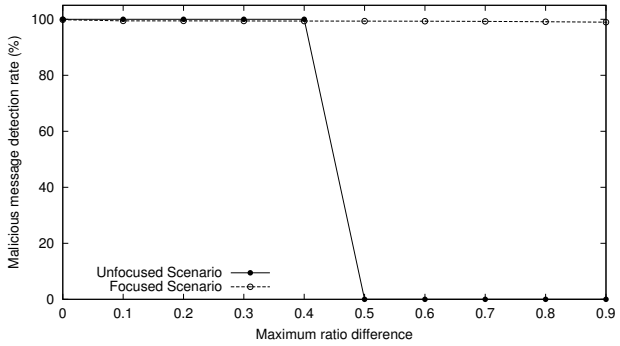

Figure 3. Malicious message detection percentage vs. maximum ratio difference

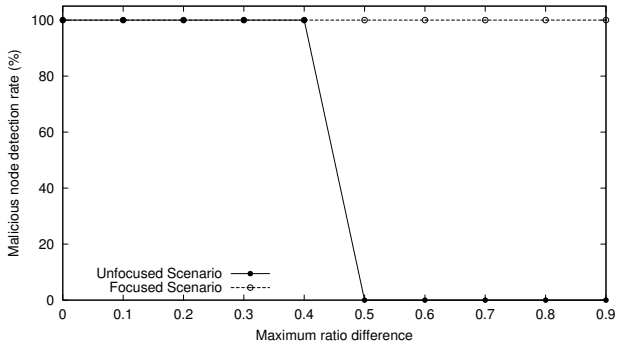

Figure 4. Malicious node detection percentage vs. maximum ratio difference

4.2.4 Detection rate vs. Message check probability

The results concerning the message check probability are shown in Figure 5. In these experiments, nodes of the network do not check all transmissions they hear. Instead, they do so with a certain probability, given by the check probability. Our results show that, in the unfocused scenario, a check probability of 70% will yield malicious node detection rates that are near 90%. This shows that it is not necessary to check all transmissions one hears in order to have a very good malicious node detection rate. On the other hand, in the focused case, the detection rate is 1 for any check probability.

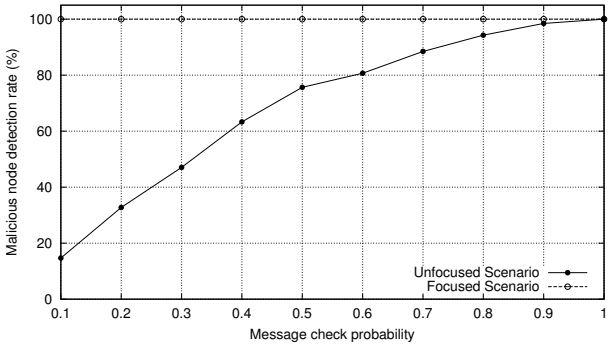

Figure 5. Malicious node detection percentage vs. message check probability

4.2.5 SNIDP overhead

Figure 6 shows the average number of transmissions and receptions in the focused scenario. These values are restricted to SNIDP protocol message exchanges triggered by malicious transmissions. These values are quite high considering sensor nodes' resources limitations, specially in terms of energy. Efficiency was not in the foreground of our considerations when we designed the SNIDP protocol (we were interested in finding firstly a simple version that would work correctly), but we expect that it will be possible to greatly optimize it. For example, a possible improvement would be for a node N to respond to inquiries regarding some suspicious node S only if N has not answered any inquiries about S for a predefined period of time. In fact, optimization of SNIDP is one of our most pressing future work.

5. Conclusion and future work

Our detection scheme detects HELLO flooding attacks because a malicious node S can only trick one node N at a time by advertising a made up position and adjusting its transmission power accordingly. However, most neighbors

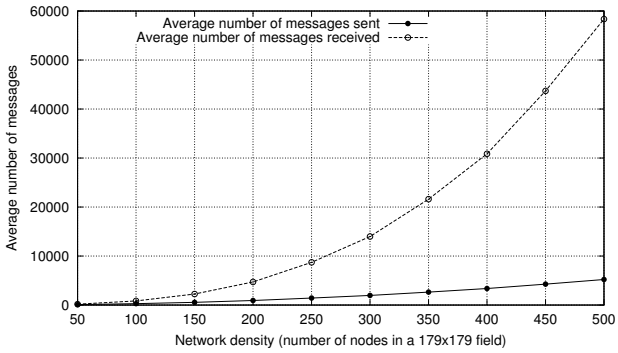

Figure 6. Average number of messages sent and received vs. network density

of N will detect this malicious transmission, and will disseminate this information among their neighbors. Through this mechanism, N will also learn that S is suspicious. Our scheme also detects wormhole attacks because messages that travel beyond their originator's transmission range are discarded.

Our proposed scheme can be easily integrated into other protocols. It would interface with the rest of the system through an API that provides information about whether a node or a message is regarded as suspicious. The MNDSS protocol does not have heavy requirements in terms of the underlying hardware; low-precision devices can be used, as the scheme works well even for relatively high values of maximum ratio difference.

Energy consumption is directly correlated to the number of message checkings and the number of message transmissions and receptions incurred by the execution of the SNIDP protocol. Regarding the number of message checkings, our experimental results show that it is not necessary that nodes check all transmissions they hear to obtain a good malicious node detection rate. Regarding the number of message transmissions and receptions, our SNIDP protocol has not been optimized yet, and the number of messages exchanged between nodes can definitely be decreased without compromising the current detection rates. A possible improvement would be for a node N to respond to inquiries regarding some suspicious node S only if N has not answered any inquiries about S for a predefined period of time.

As future work, we intend to explore other physical models. The Two-Ray Ground model alone does not model signal power loss due to obstacles, weather conditions, interference, etc.

Determination of a malicious node's location is another interesting question. It is, however, not a straightforward task. It would require some degree of cooperation between nodes, so that they may together pinpoint the approximate location of the adversary. When a node receives a message and considers it as suspicious, it may coordinate with its neighbors to try to locate the origin of the message by using the signal strength detected by its own receiver and the signal strength received by other nodes. A minimum of three nodes should be sufficient in finding the approximate region where the adversary is located. We envision a scheme similar to what is deployed in GPS Radio systems, where a receiver needs to obtain a position from at least three orbiting satellites.

References

[1] Berkeley MICA mote. http://webs.cs.berkeley.edu/tos/hardware/hardware.html, 2003.

[2] MICA2 radio stack for TinyOS. http://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/mica2radio/CC1000.html, 2003.

[3] Chipcon. SmartRF CC1000 single chip very low power RF transceiver. http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf, 2003.

[4] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization. Technical report, University of California, Berkeley, 2001.

[5] S. Hollar. COTS Dust. Master's thesis, University of California, Berkeley, December 2000.

[6] Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, April 2003.

[7] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *International Conference on Mobile Computing and Networking (MOBICOM)*, pages 271–278, 1999.

[8] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Emerging challenges: Mobile networking for "smart dust". *Journal of Communications and Networks*, 2(3):188–196, September 2000.

[9] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[10] T. S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall, 2nd edition, 2002.

[11] W. Stallings. *Cryptography and Network Security – Principles and Practice*. Prentice Hall, 3rd edition, 2003.

[12] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart dust: Communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, 2001.