Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks

Jing Deng Richard Han Shivakant Mishra
Computer Science Department
University of Colorado at Boulder
Boulder, Colorado, USA
{jing,rhan,mishras}@cs.colorado.edu

Abstract

Wireless sensor networks are highly vulnerable to the failure of base stations. An adversary can render a wireless sensor network useless by launching remote, software-based attacks or physical attacks on the base stations. This paper addresses the problem of defending a base station against physical attacks by concealing the geographic location of a base station. Typical packet traffic in a sensor network reveals pronounced patterns that allow an adversary analyzing packet traffic to deduce the location of a base station. The paper investigates several countermeasures against traffic analysis techniques aimed at disguising the location of a base station. First, a degree of randomness is introduced in the multi-hop path a packet takes from a sensor node to a base station. Second, random fake paths are introduced to confuse an adversary from tracking a packet as it moves towards a base station. Finally, multiple, random areas of high communication activity are created to deceive an adversary as to the true location of the base station. The paper evaluates these techniques analytically and via simulation using three evaluation criteria: total entropy of the network, total energy consumed, and the ability to guard against heuristic-based techniques to locate a base station.

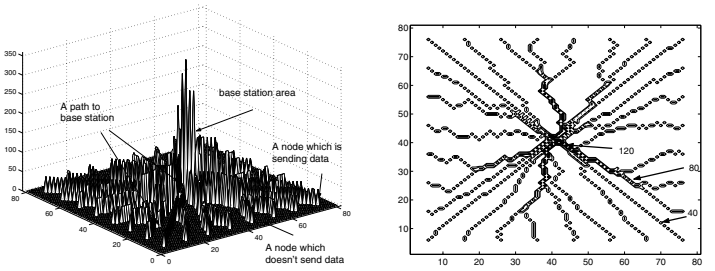
1 Introduction

A wireless sensor network (WSN) consists of a large number of small, resource-constrained sensor nodes, e.g. Berkeley MICA2 motes [11], and a small number of relatively powerful base stations, e.g. PC-caliber gateways. Each sensor node acts as an information source, sensing and collecting data samples from its environment. Each sensor node communicates this data to a base station via a multi-hop network in which each node performs routing functions.

Base stations are a critical part of a WSN. In fact, an entire WSN can be rendered useless by taking down its base stations. Hence, it is vital to protect a base station against both software-based and physical attacks. Several secure and intrusion tolerant techniques have been developed in recent years that protect a base station against remote attacks that exploit some software vulnerability in the base station. However, these techniques cannot protect the base station against a physical attack. In this paper, we address the problem of protecting a base station against physical attacks by concealing its geographic location in the network.

Sensor data is typically routed along relatively fixed paths from sensor nodes towards the base station. This produces quite pronounced traffic patterns that reveal the direction towards and hence the location of the base station. Figure 1 illustrates the packet traffic volume forwarded by each node in the network with the shortest path routing scheme (we call it the SP scheme). The nodes near the base station clearly forward a significantly greater volume of packets than nodes further away from the base station, in the same manner that a river grows wider as it collects more water from its tributaries. Aggregator nodes that compress the data from multiple child nodes before forwarding upstream towards the base station can mitigate the pronounced increase in traffic volume towards the base station. However, the data traffic still accumulates towards the base station, if the aggregators send their data through multiple hops.

An adversary can analyze the traffic patterns revealed in Figure 1 to deduce the location of the base station within the WSN's topology. Since the base station is a central point of failure, once the location of the base station is discovered, an adversary can disable or destroy the base station, thereby rendering ineffective the data-gathering duties of the entire sensor network. Targetting the base station is also the most efficient use of an attacker's resources, since energy, time, and effort need be expended to destroy only a small number of base stations rather than to destroy every sensor node in the network. Given that the number of base stations in a WSN is relatively small, the pronounced data traffic pattern shown in Figure 1 is not likely to be mitigated in any sig-



(a) 3-D graph of data traffic. (b) Contour map of data traffic.

Figure 1. Pronounced data traffic patterns in a WSN using SP routing scheme reveal the location of the base station.

nificant way by introducing multiple base stations. So, even if there are multiple base stations, an adversary can employ the same traffic analysis techniques to locate and destroy each base station one by one.

Traffic analysis is a very effective way to determine the geographic location of a base station. For example, if a base station is well-concealed visually, an adversary cannot determine its location by visually scanning the area where the WSN is deployed. He needs to analyze the network traffic to determine the location of a base station in such cases. Furthermore, if the WSN covers a large area, such as a WSN deployed over several square miles of a battlefield, it is very difficult for the adversary to scan every location to find a base station. However, by analyzing network traffic, he can quickly track its location. In some other cases, it is impractical for the adversary to freely move from place to place to visually search a base station. For example, an adversary monitoring a sensor network needs to hide himself from sensor nodes. Traffic analysis provides him an efficient way to find the location of the base station.

Even if the data packets are encrypted, e.g. by pairwise key schemes [8, 3, 7, 15, 26], an adversary can deduce significant information by monitoring traffic volume and traffic path information in a sensor network. We identify two classes of traffic analysis attacks in wireless sensor networks, a *rate monitoring* attack and a *time correlation* attack. In a *rate monitoring* attack, an adversary monitors the packet sending rate of nodes near the adversary, and moves closer to the nodes that have a higher packet sending rate. In a *time correlation* attack, an adversary observes the correlation in sending time between a node and its neighbor node that is assumed to be forwarding the same packet, and deduces the path by following the "sound" of each forwarding operation as the packet propagates towards the base station. One way to defend against a time correlation attack is to buffer incoming packets in the nodes for some random period before forwarding them. However, an adversary can pro-actively trigger the forwarding of packets by generating abnormal sensory events, e.g. abnormal temperature, that need to be forwarded as quickly as possible.

Deducing the location of a base station is further simplified if two or more adversaries cooperate with one another. For example, two cooperating adversaries on different sides of a WSN can respectively determine the direction (a vector) where a base station is possibly located from their current location by analyzing packets over just a few hops. They can then get a pretty accurate estimate of the base station location by intersecting the two vectors.

In this paper, we focus on developing countermeasures against traffic analysis attacks that seek to locate the base station, particularly the *rate monitoring* attack and the *time correlation* attack. Without loss of generality, we consider sensor networks with a single base station. The anti-traffic analysis techniques proposed in this paper introduce randomized traffic volumes throughout the sensor network away from the base station, in order to deceive and misdirect an adversary so that the true path towards the base station cannot be easily found. Four anti-traffic analysis techniques are proposed to generate randomness. First, a multiple parent routing scheme is introduced that allows a sensor node to forward a packet to one of multiple parents. This makes the patterns less pronounced in terms of routing packets towards the base station. Second, a controlled random walk is introduced into the multi-hop path traversed by a packet through the WSN towards the base station. This distributes packet traffic, thereby rendering less effective rate monitoring attacks. Third, random fake paths are introduced to confuse an adversary from tracking a packet as it moves towards a base station. This mitigates the effectiveness of time correlation attacks. Finally, multiple, random areas of high communication activity are created to deceive an adversary as to the true location of the base station, which further increases the difficulty of rate monitoring attacks. We have analyzed our anti-traffic

analysis techniques against rate monitoring and time correlation attacks. However, we believe that they can withstand other unforeseen traffic analysis attacks as well by virtues of providing increased randomness in communication patterns and increased deceptive mechanisms to confuse an adversary.

These anti-traffic analysis techniques are specially suited to the characteristics of wireless sensor networks, and exhibit several advantages. First, all four techniques are distributed in nature. There is no single initialization or coordination point involved to setup these mechanisms. Second, memory and computation requirements in each sensor node are relatively low, and can easily be met by modern sensors such as MICA2. Third, any compromise of one or a small number of sensor nodes by an adversary is easily tolerated. If an adversary compromises some nodes, the damage it can inflict upon the WSN is limited. Fourth, our techniques don't require a node to delay sending packets, as would be the case in standard de-correlation approaches. A node can send/forward its packet as soon as it is ready. This aids in reducing the time delay introduced by anti-traffic analysis techniques. Finally, the cost of these techniques is moderate and the techniques are applicable to large sensor networks. This is confirmed by the simulation results presented in the evaluation section.

While our techniques significantly delay an adversary, they also introduce overhead that reduces the energy lifetime of the network. Our experiments show that the number of messages increases by about 2 to 3 times, while the mechanisms delay the time of finding a base station by about 19 times. While energy is certainly critical in sensor networks, *our claim is that the benefits of employing decorrelation far outweigh the reduction in energy lifetime*. Consider first one extreme in which no attempt is made to disguise the traffic patterns. As shown later, the number of search steps to find the base station given no decorrelation techniques and shortest path routing is on the order of tens of steps. If each step takes say ten minutes to physically move from one hop to the next, then we see that the base station will be found and destroyed in under two hours. Since the energy lifetime of the WSN is typically far longer, e.g. 2-3 months assuming two AA batteries and a low duty cycle, then the lifetime of the unprotected network will be dominated by its vulnerability to traffic analysis attacks. Therefore, it is well worth the investment to employ energy-intensive decorrelation techniques to prolong the lifetime of the WSN. On the other extreme, consider that maximal decorrelation is applied. This effect is achieved when every packet is broadcast flooded to the network, resulting in a uniform traffic rate at every node that completely hides the base station. This extreme also incurs the maximum overhead and energy costs. *The strategy of this paper is to employ countermeasures against traffic analysis, e.g. DEFP defined later, that achieve close to broadcast flooding's maximal decorrelation at a fraction of the cost.* As shown later, our techniques are able to force an adversary to search close to the maximum number of steps required given flooding (within two-thirds), at about two orders of magnitude less overhead than flooding.

The paper is organized as follows. In Section 2, the network model, threat model, and capabilities of sensor nodes are described. In Section 3, the probabilistic countermeasures embedded into routing algorithms are described, and an analysis of their security under node-compromise is presented. In Section 4, algorithms are simulated and quantitatively measured in terms of their effectiveness and cost. Section 5 discusses related work, and finally, Section 7 concludes the paper.

2 Network Traffic and Threat Model

We assume the sensor network has a base station and a number of aggregator nodes. Each aggregator node processes data that it received from its group of local sensor nodes and sends that processed data to the base station through multiple hops.

For the capabilities of an adversary, we assume that an adversary can monitor network traffic, and launch a *rate monitoring attack* and a *time correlation attack*. An adversary can capture sensor nodes, compromise them and obtain all information, e.g. encryption keys and routing tables, inside a node. He can reprogram a node and convert it into a malicious node. However, we assume that the adversary requires some non-trivial amount of time to compromise a node. So, he can compromise only a small number of nodes in any reasonable period of time. In particular, we assume that the time an adversary takes to compromise all nodes along a path to a base station is much longer than the interval in which a base station changes its location (T_b). We also assume that an adversary can physically move from one location to another in the network. However, he doesn't have global information about the whole network, and cannot jam the entire network. We think that this assumption is reasonable for a large scale sensor network, e.g, a sensor network deployed in a battle field. Our solutions are designed for a large sensor network, in which an adversary cannot see the base station visually, although if he is close to the base station, he can identify it immediately. We call the area within which an adversary can immediately identify a base station as the *base station area*.

We assume that sensor nodes use the key framework proposed in LEAP [26] to protect hop-by-hop communication. Nodes can set up pair-wise keys using existing protocols [8, 3, 7, 15, 26]. Every node can also set up a single cluster key [26] with all of its neighbor nodes. As described in [5], when a node sends a packet, it protects and encrypts the

packet with its cluster key. An adversary cannot decrypt the contents of a packet. At the same time, other nodes in the cluster can easily understand the type of packet and process it accordingly.

In this paper, we focus on protecting the data traffic from aggregator nodes to base station through multiple hop routing. The local data traffic between sensor nodes and aggregator node can be protected by simple anti-traffic analysis schemes such as those proposed in [5].

3 Anti-traffic analysis strategies

3.1 Multi-parent routing scheme

To reduce the starkness of pronounced paths, we modify the shortest path (SP) routing scheme shown in Figure 1 by having each node randomly select one of multiple parent nodes to route data to the base station. When a node needs to forward a packet, the node randomly selects one of its parent nodes to forward the packet. We call this scheme multi-parent routing (MPR). We propose two methods for setting up multiple parents for each node. In the first method (See Figure 2), the beacon message sent by the base station contains a *level* field. The base station sets the value of *level* to 0. When a node forwards a beacon message, it increments it by 1. So the value of *level* represents the number of hops that a node is from the base station along a particular path. A sensor node s selects all neighbor nodes whose *level* value is less than s's *level* value as its parent nodes. In the second method, a node monitors all beacon messages it receives before forwarding the first beacon message. Since a node s has to wait for some amount of time before forwarding a beacon message (waiting time in MAC layer), it selects all nodes from whom it receives a beacon message while waiting to forward the first received beacon message as its parent nodes.

An adversary has several ways to attack these multi-parent routing setup schemes. For example, a malicious node can claim a low *level* value to attract traffic from other nodes, or it can use unfair media access control mechanisms to occupy the wireless channel. Protecting routing schemes from such attacks is beyond the scope of this paper. Here we assume that the routing set up scheme is relatively fast, so an adversary doesn't have enough time to attack routing set up process. Several mechanisms [14, 5] have already been proposed to protect against attacks during the routing setup.

3.2 Random Walk

To further diversify routing paths and mitigate rate monitoring attacks, we propose a random walk (RW) routing scheme. In RW, when a node receives a packet, it forwards

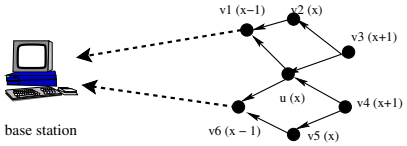


Figure 2. Neighbors and parents of node u. Figure shows node ID and its $level$ value. In SP, node u has one parent node v_1. In MPR, node u has two parent nodes, v_1 and v_6. In RW, u forwards packets to v_1 with probability p_r or v_6 with probability $1 - p_r$.

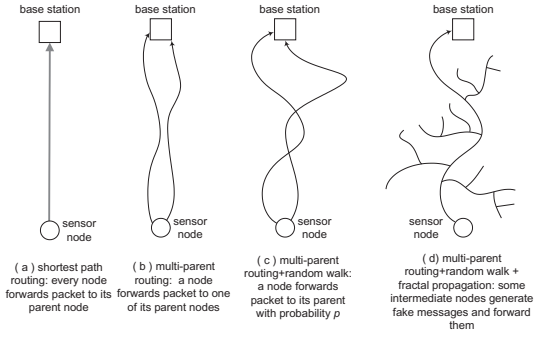


Figure 3. Techniques to counter traffic analysis.

the packet to one of its parent nodes with probability p_r. However, it uses a random forwarding algorithm with probability $1-p_r$. In the random forwarding algorithm, the node forwards the packet to one of its neighbor nodes with equal probability. Like [13] and [24], MPR and RW use probabilistic routing. However, [13] and [24] use probabilistic routing for reliable data transmission in sensor networks, while we use probabilistic routing to defend against the *rate monitoring attack*.

The RW technique results in some packets traversing a longer path to reach the base station than the shortest available path, as shown in Figure 3(c). This implies that RW will consume more energy per node on an average. To estimate how much extra energy is consumed by RW, we calculate the cost C of RW, where cost is defined as [5]: $C = \frac{M'}{M}$. Here, M' is the average number of hops a packet takes to reach the base station from an aggregator node in RW, and M is the number of hops a packet takes to reach the base station from the same aggregator node in SP. Clearly, M' depends on the several factors related to network topology, e.g. how many neighbors a sensor node has, how far the base station is from a sensor node or from one of its neighbor nodes, and so on. We calculate the value of C by mak-

ing the following simplifying assumption. Suppose a node u randomly selects a neighbor node v to forward a packet, and the distance (number of hops along the shortest path) between v and the base station is d, while the distance between u and base station is d'. We assume that the probability that $d > d'$ is same as the probability that $d < d'$. So on average, when u forwards a packet to v, the distance from the base station doesn't change. Only when u forwards the packet to its parent node, the distance is reduced by 1. We denote n as the number of hops from the aggregator to the base station in SP, and n' as the number of average hops in RW. We have $n' \times p_r = n$. This implies $C = \frac{M'}{M} = \frac{1}{p_r}$.

In addition, a packet will take a longer time to reach the base station in RW. In fact, the extra time delay is directly proportional (linear) to the extra hops used for forwarding the packet. So, the time cost for each packet to reach the base station is roughly $\frac{1}{p_r}$ in RW.

3.3 Fractal Propagation

MPR and RW spread out data traffic and make it difficult to use a rate monitoring attack. However, RW is still vulnerable to the *time correlation* attack. Usually, for a node s, the number of parent nodes is less than half of s's neighbor nodes, and for energy and efficiency considerations, $p_r > 0.5$ typically. As a result, the possibility that a node forwards a packet to its parent node is higher than the possibility it forwards the packet to any one of its other neighbors. An adversary can exploit this to launch a time correlation attack, either by injecting abnormal report data or monitoring over a long period of time.

To address the shortcomings of MPR and RW, we propose a new technique called *fractal propagation*. In this technique, several *fake* packets are created and propagated in the network to introduce more randomness in the communication pattern. When a node hears that its neighbor node is forwarding a packet to the base station, the node generates a fake packet with probability p_c, and forwards it to one of its neighbor nodes. To control the propagation range of the fake packet, each newly generated fake packet contains a *TTL* parameter with value K. K is a constant that is known to all nodes, so an adversary cannot flood the whole network by sending fake packets with *TTL* parameter higher than K. When a node receives a fake packet, it decrements *TTL* by 1. If the value of *TTL* is greater than zero, the node forwards the fake packet to one of its neighbor nodes (not necessarily in the direction of the base station). If the value of *TTL* is zero, a node stops forwarding the fake packet. In addition, when a node hears that its neighbor node is forwarding a fake packet to someone else with *TTL* value k ($k < K$), it generates and forwards another fake packet with probability p_c and *TTL* value $k - 1$.

These fake packets spread out in the network and their transmission paths form a tree (see Figure 3(d)). In particular, the communication traffic is much more spread out than RW. So even if an adversary can track a packet using time-correlation, she cannot track where the real (as opposed to fake) packet is going. This is because she cannot differentiate between a real and a fake packet without knowing the encryption key.

Suppose a node has x neighbor nodes on average. Let $p_f = p_c \times x$ and $f(K)$ represents the total length of a fake tree that originated with *TTL* value K. We have

$$f(K) = p_f \times f(K-1) + f(K-1) + 1$$

Solving this recursive equation, we get

$$f(K) = \sum_{i=0}^{K-1} (p_f + 1)^i = \begin{cases} \frac{(p_f+1)^K - 1}{p_f} & \text{if } p_f > 0 \\ K & \text{otherwise} \end{cases}$$

Suppose the length of real path from the aggregator node to the base station is n. The cost is

$$C = \frac{n + n \times p_f \times \frac{(p_f+1)^K - 1}{p_f}}{n} = (p_f + 1)^K$$

If we combine RW and the fractal idea, the total cost is

$$C = \frac{(p_f + 1)^K}{p_r}$$

If we use fixed values of p_r, p_f and K, the average cost is a fixed value that is independent of the size of the network.

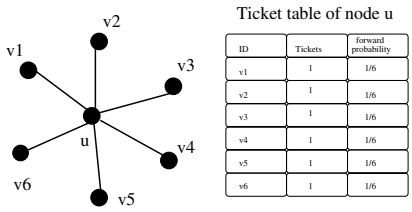
3.3.1 Fractal propagation with different forking probabilities

One problem with simple fractal propagation is that it generates a large amount of traffic near the base station. This will potentially increase packet collision rate and packet loss rate.

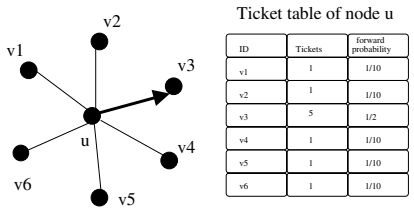
To address this problem, nodes can use different probabilities to generate fake packets. When a node forwards packets more frequently, it sets a lower probability for creating new fake packets. This technique is called Differential Fractal Propagation (DFP). The algorithm for setting this probability is as follows. When the packet forwarding rate r at a node is lower than a threshold h, the node generates new fake packets with probability p. When the packet forwarding rate is higher than h, the node generates new fake packets with probability $p' = p \times (h/r)^2$; h can be chosen as the packet sending rate of the aggregator node.

3.3.2 Enforced fractal propagation

The idea of fractal propagation aids significantly in spreading out the communication traffic evenly over the network



(a) before node u forwards any fake packet



(b) after u forwards a fake packet to node v_3

Figure 4. Ticket table of node u

and obfuscating any paths to the base station. To make matters worse for an adversary, we generate local high data sending rate areas, called *hot spots*, in the network. An adversary may be trapped in those areas and not be able to determine the correct path to the base station. This routing technique is called Differential Enforced Fractal Propagation (DEFP). The challenge here is how to create hot spots that are evenly spread out in the network, such that only a minimum (preferably zero) amount of extra communication/coordination among the sensor nodes is needed.

DEFP is a simple distributed algorithm based on DFP. The key idea is to let the nodes that forwarded fake packets earlier have a higher chance to forward fake packets in the future. In particular, if a node u forwarded a fake packet to another node v in the past, then it forwards the next fake packet received to v with a higher probability. The node uses a *lottery scheduling* algorithm [23] to choose the next node to forward the fake packet to. In this algorithm (see Figure 4), a node assigns tickets to each of its neighbor nodes. It chooses the next node to forward a fake packet to based on the number of tickets assigned to the neighbor nodes. A neighbor node with more tickets assigned has the higher probability of being chosen. In the beginning, all neighbor nodes are assigned one ticket. When the node chooses a neighbor node as the next node for forwarding a fake packet, it increments that node's tickets by k. This way, after a node has forwarded a fake packet to one of its neighbor nodes, it will continue to forward other fake packets to the same neighbor node with higher and higher probability. If an area of nodes receive fake packets, they are more likely to process more and more fake packets in the future. This will turn that area into a hot spot. It is also very easy to destroy current hot spots and reconstruct new hot spots at different places. For example, sensor nodes just reset the value of tickets to 1 when they receive a broadcast message from base station, and then start to build hot spots from the beginning. A patient attacker can wait at a hot spot until the communication pattern changes. While this will allow the attacker to determine that he was at a fake hot spot, it does not provide any other information about the possible location of the base station. Furthermore, waiting for long time at a fake hot spot will add more delay to finding the location of the base station.

3.4 Node Compromises

If an adversary compromises a node, she can find out the identity of its parent nodes, and read the contents of all packets passing through this node. In addition, by monitoring the traffic for some sufficiently long period of time, she can obtain distribution information of all the ancestor nodes within her activity range. However, with this knowledge, she cannot determine the location of the base station, and cannot block communication between an aggregator node and the base station. To determine the location of the base station, the adversary will have to compromise a large number of nodes along the path to the base station.

In fractal propagation, if an adversary compromises a node, she can find out whether a packet is a fake or real. However, she cannot obtain any information other than the ones discussed above. The adversary can attempt to launch a DoS attack by generating several fake packets and forwarding them to flood the network. However, the propagation area of a fake packet is limited by the value of the *TTL* parameter. A fake packet can propagate and generate new fake packets only within a small part of the network, so the damage due to such DoS attacks is limited to a small part of the network.

Cooperating adversaries can launch a cooperative attack, such as the one described in Section 1 by compromising sensor nodes. However, such an attack requires that the direction in which a parent node is located is precisely the direction towards the base station. This is quite unlikely in a randomly distributed sensor network. In addition, MPR increases the difficulty in determining the precise geographic direction towards the base station, forcing the adversary to compromise a large number of nodes.

Finally, an adversary can also generate several forged data packets and forward them to the base station in an attempt to flood the base station. However, mechanisms exist currently that allow intermediate nodes to filter out forged data packets, e.g. see [25, 27]. In these mechanisms, intermediate nodes use randomly pre-distributed pair-wise keys to verify the authenticity of the data sent by the aggregator

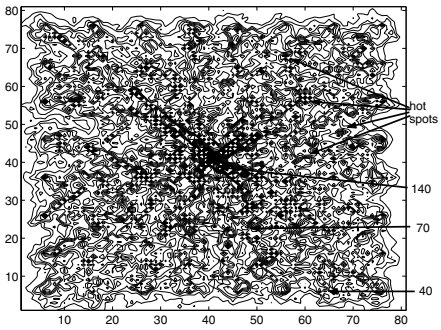


Figure 5. Number of packets sent/forwarded by each node in DEFP.

node. Forged packets are filtered out by each intermediate node with certain probability and thus prevented from propagating over a long path.

3.4.1 Simulation Results

We simulated our anti-traffic analysis techniques in our simulator, which is based on a standard discrete event generator. Simulation results show that RW creates a more diffuse routing pattern than SP, while both fractal propagation techniques, DFP and DEFP considerably obfuscate the location of the base station. Figure 5 shows the cumulative routes taken by packets through a sensor network employing DEFP. The network configuration for these simulations is a grid network described in Section 4.

4 Evaluation

4.1 Evaluation Criteria

The main goal of anti-traffic analysis techniques is to prevent an adversary from tracking the location of a base station by analyzing communication patterns of a WSN over some reasonable period of time. Our goal is to make communication patterns as random as possible while minimizing costs, so that an adversary does not have sufficient information to deduce the location of the base station in a reasonable amount of time. Our evaluation focuses on how random the network traffic is, and the cost of our anti-traffic analysis schemes. We haven't simulated the effectiveness of defending against *time-correlation* attacks. A higher forking probability (p_f) and a larger TTL of fake path will make it more difficult to launch a *time correlation* attack. We evaluate the randomness of network traffic and effectiveness against *rate monitoring* attacks through two metrics —*entropy* of the network traffic and the *GSAT* test. To estimate the cost of our techniques, we count the number of messages exchanged in our techniques and compare them with the number of messages exchanged in SP. Since our techniques incur very little memory cost on each sensor node, e.g. a few encryption keys and tickets of neighbor nodes, we have not bothered to measure memory consumption overhead in our simulation.

Entropy We use entropy to measure the randomness of network traffic. Entropy is a mathematical measure of information uncertainty, and it has been widely used as a metric to measure randomness in many applications, e.g. data communication, data compression, random number generators, and security of cryptographic algorithms. Entropy of a random variable X with a probability function $p(x)$ is defined as $H(X) = -\sum p(x)\log_2 p(x)$. Suppose that during a time period T, a sensor node a sent/forwarded p_a packets, and a total of M packets were sent/forwarded in the WSN N. We use the following formula to measure the entropy of N during the time period T: $H(N) = -\sum_{a \in N} \frac{p_a}{M} \log_2 \frac{p_a}{M}$. In general, a higher value of $H(N)$ implies that the communication traffic pattern of N is more random.

GSAT Test The GSAT test is intended to measure the ability of a routing technique to guard against heuristic-based algorithms that an adversary may use to locate a base station. The GSAT algorithm [20] was proposed for solving NP-hard satisfiability problems, such as the 3SAT problem [4]. In contrast to the traditional deterministic solutions, GSAT is a probabilistic algorithm that combines a hill-climbing search algorithm with a random restart mechanism. GSAT can solve most of the large 3SAT instances in a short time.

We use the idea of the GSAT algorithm to design a heuristic-based algorithm that an adversary uses to track the location of the base station. In this algorithm, an adversary starts at some location in the sensor network N. She monitors network traffic around her within her activity range. If she finds that a different node s within her activity range has the highest traffic, she moves to s, and continues to monitor traffic from s. Using this mechanism, she can move towards the locations that have higher and higher traffic volume. However, if she reaches a location that has the highest traffic with in the neighborhood (local maxima), she selects a direction at random, moves in that direction for some time, and then repeats the above algorithm. She continues to do this until she finds the base station.

The GSAT test measures the average number of hops an adversary takes to finally reach the base station using this heuristic algorithm. A large value of GSAT test implies that the routing technique has better potential to guard against heuristic-based algorithms that an adversary may use to locate a base station.

In addition to randomness, the exact values of entropy and the GSAT test depend on several other network characteristics, e.g. network structure, network size, number and

	Size	Average # Neighbors	Number of Aggregators	Sending Rate
Grid	81×81	8	28	4/minute
Random	4500	20	28	4/minute

Table 1. Network configuration Parameters

location of aggregator nodes. To evaluate our techniques, we have focused on differences in entropy and GSAT test values measured under the cases when one of the proposed anti-traffic analysis techniques is used and the case when no anti-traffic analysis technique is used. We also experimented with different values of P_r in RW and P_f in DEFP, to understand the effects of these parameters. We simulated two network structures in our experiments: a grid topology and a random topology. Table 1 shows the parameters used in our simulation.

4.2 Effectiveness and Cost of Anti-Traffic Analysis Techniques

To evaluate the effectiveness of our anti-traffic analysis techniques, we simulated them over a grid network (see Table 1) and measured the values of entropy, GSAT test, and energy cost (number of messages exchanged). We simulated the following techniques: MPR, MPR+RW, MPR+RW+DFP, and MPR+RW+DEFP. For simplicity, we use MPR, RW, DFP, and DEFP respectively to refer to these techniques in the rest of the paper. In these simulations, we set p_r to 0.6, p_f to 0.2, and K to 6.

To obtain an estimate of an upper bound of entropy and GSAT values, and a lower bound on the cost, we simulated two routing mechanisms. The first routing mechanism is SP, which selects the shortest path to the base station from each sensor node. SP provides a measure of lower bound on the cost of routing, but results in very pronounced communication patters as shown in Figure 1. The second routing mechanism is called the broadcast scheme (BR scheme). In this mechanism, every message sent by an aggregator node is flooded to the entire network. Since BR generates uniform network traffic, it provides a measure of an upper bound of entropy and GSAT values. Table 2 shows the entropy values and number of messages exchanged in SP and BR.

Figure 6 (a) shows the entropy measured for various routing techniques. All data reported here are an average over 20 runs. As expected, entropy is lowest for SP and highest for broadcast. Entropy for MPR and RW is higher than SP, but lower than DFP and DEFP. This shows that the idea of generating fake packets in a controlled manner does aid in making the network traffic pattern more random. This is in addition to the original goal of defending against time-correlation analysis.

To determine resiliency against a GSAT search, we simulated the data traffic and recorded the number of packets sent/forwarded by each node in a log file. We initialized a starting point for the adversary in the network and used the GSAT algorithm to discover the base station area. We recorded the number of steps the adversary takes to get into the base station area. For each log file, we set 81 different initial locations. For each initial location, we ran GSAT to search for the base station area for 100 times, and recorded the number of hops the adversary takes to get into the base station area. Finally, we computed the average number of hops the adversary takes to get into the base station area for each technique. In addition, we experimented with three different activity ranges of the adversary: adversary could monitor data traffic over 3×3, 5×5, and 9×9 areas around her respectively.

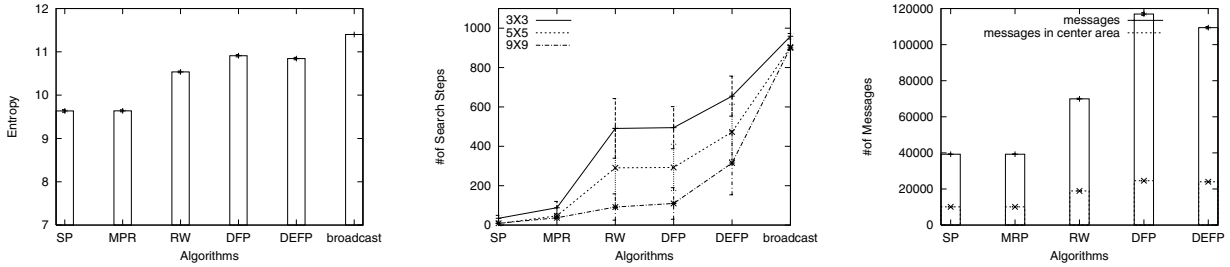
Figure 6 (b) shows the results of the GSAT test. First, we see that the GSAT values correlate with the entropy values shown in Figure 6 (a) (except DEFP). Higher entropy corresponds to a larger value of GSAT. This implies that both entropy and GSAT are useful metrics to measure the randomness in network traffic. The only exception is DEFP. Since DEFP converges some traffic together to form *hot spots*, it results in less entropy compared to DFP. However, those *hot spots* make it more difficult for an adversary to locate the base station using a GSAT search algorithm. This is evident from the higher values of GSAT in DEFP.

The activity range of an adversary also impacts the GSAT value. If the activity range is larger, the corresponding GSAT value is smaller. This implies that the adversary can find the base station in less number of hops. Also, we observe that anti-traffic analysis techniques significantly increase the number of steps an adversary has to take to locate the base station. For example, she can discover the base station area in 34 steps in SP (activity range 3×3), and 653 steps in DEFP, which is about 19 times more. *Notice that the number of search steps required when DEFP is applied as a countermeasure (about 600 in 3x3 monitoring) is within about two-thirds of the number of search steps required when broadcast flooding is used as a defense (about 900).* Even when the activity range of the adversary is large (9×9), our anti-traffic analysis techniques significantly increase the number of hops an adversary has to take to locate the base station area.

Figures 6 (c) shows the energy overhead of our techniques. We are interested in the overall energy overhead of the network, and also the energy overhead of nodes in the vicinity of the base station. The energy overhead is critical, because it affects the lifetime of a sensor node, as well as the packet loss rate caused by packet collisions. We are particularly interested in energy overhead in the nodes near the base station, because these nodes typically carry larger amounts of traffic, and any problem with these nodes may

	Entropy		Traffic		Center Traffic	
	(SP)	(BR)	(SP)	(BR)	(SP)	(BR)
Grid	9.64	11.40	39000	7×10^6	10080	4×10^5
Random	8.20	12.08	21000	5×10^6	2792	1.8×10^5

Table 2. Entropy and Number of messages exchanged in SP and BR. (Traffic means the total messages exchanged in the network, and Center Traffic means the number of messages exchanged in the close vicinity of the base station.)



(a) entropy (b) GSAT test: number of search steps (c) message cost

Figure 6. Effectiveness and cost of anti-traffic analysis mechanisms.

cause serious communication problems in the WSN.

Figure 6 (c) shows the total number of messages sent/forwarded by all nodes in the network, and the number of messages sent/forwarded by nodes near the base station (which is an area of 20×20 nodes with base station at center). The traffic in RW is about 1.8 times larger than the traffic in SP for the whole network and the area near the base station. The message cost of DFP and DEFP is about 2.8 times the message cost of SP in the whole network, and 2.4 times near the base station. *When DEFP is compared with broadcast in terms of total overhead cost, i.e. DEFP incurs* 10^5 *messages while broadcast incurs 7x*10^6 *from Table 2, then broadcast costs about 70 times more than DEFP. Thus, DEFP requires close to the same number of search steps as flooding (within two-thirds), at a fraction of the overhead cost (about two orders of magnitude less overhead).*

In our simulation, when aggregators send four packets per minute, the nodes directly connected to the base station forward about 14 packets per minutes in SP, and about 34 packets per minute in DEFP. This is easily feasible in the current sensor network technology. An important point to note is that the message cost of these algorithms is constant. It doesn't increase with increase in network size.

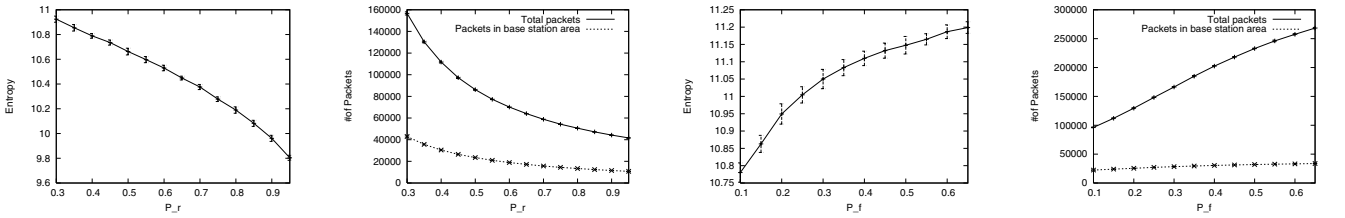
4.3 Effectiveness of p_r and p_f

To understand the effect of different values of p_r and p_f, we varied parameters for RW and DEFP. We simulated them on both a grid network and a random network (Table 1. In RW, we varied p_r from 0.3 to 0.95. In DEFP, we fixed p_r at 0.6 (since it generates enough randomness in RW, and makes sure that a packet can be sent to base station with little extra delay), and varied p_f from 0.1 to 0.65. The results are shown in Figures 7 and 8. We notice that the variation in the values of entropy and message cost based on p_r and P_f is similar in both grid and random networks. In RW, the entropy sub-linearly decreases and the number of messages decreases with increasing p_r. In DEFP, entropy sub-linearly increases and the number of messages dramatically increases with increasing p_f.

These results imply that we should chose p_f as small as possible, as long as it satisfies our requirements. In Section 3, we analyzed the relation between message cost, and p_r and p_f. The results from these experiments imply that there is a relation between the entropy of network traffic, and p_r and p_f, which is independent of the size of the network. Another observation is that although the total number of messages exchanged is quite large for very large values of p_f, the number of messages exchanged near the base station doesn't change a lot. This shows that the traffic control mechanism proposed in DFP and DEFP works quite well.

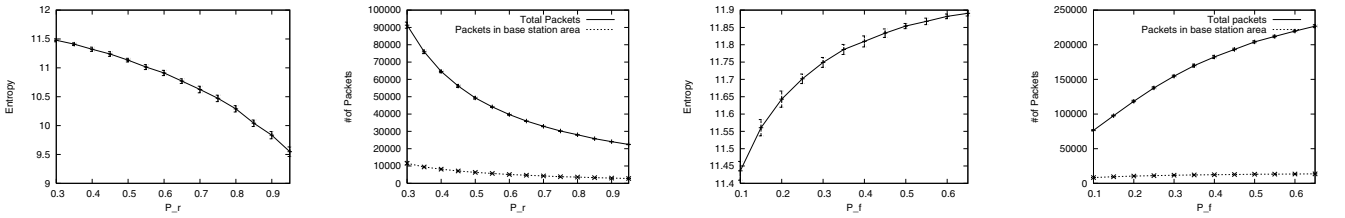
5 Related Work

Research in security issues in sensor network research has received much attention recently, e.g. secure data communication [17], secure routing [14, 12, 5], secure data aggregation [19], and pairwise key setup [8, 3, 7, 15, 26]. In the area of privacy in E-commerce, many techniques



(a) Entropy (RW) (b) Message overhead (RW) (c) Entropy (DEFP) (d) Message overhead (DEFP)

Figure 7. Effectiveness of p_r and p_f (Grid network).



(a) Entropy (RW) (b) Message overhead (RW) (c) Entropy (DEFP) (d) Message overhead (DEFP)

Figure 8. Effectiveness of p_r and p_f (Random network).

have been developed to protect the anonymity of message senders and receivers. Our countermeasures against traffic analysis are similar to the methods used in traditional privacy and anonymity research, but we have three unique properties. First, we focus on hiding the physical location of a base station, instead of hiding the identity of a message sender or receiver. Second, the communication pattern in sensor networks is highly asymmetric and converges on a base station. This make it more difficult to protect the base station against traffic analysis attacks. Third, traditional networks are resource-rich compared to a WSN, and so the techniques developed for traditional networks are unlikely to be directly applied in sensor networks.

In traditional privacy research, mist routing requires predeployed, hierarchical and trusted routers [2]. [10] requires that every node can talk to every other node. The Onion routing protocol [9] disguises who talks to whom on the Internet by layered encryption and by forwarding received messages in a random order. In addition, a large number of messages are stored before forwarding them in a different order. A sensor node doesn't have enough memory to store many packets. The k-anonymous message transmission protocol proposed in [1] protects anonymity for both sender and receiver with low data transmission latency. Unfortunately, its high communication and computational requirements prevent it from being used in sensor networks. The techniques to disguise a receiver by routing each message to multiple receivers using a multicast mechanism are proposed in [18, 21]. Tor [6] is the second-generation onion router, which is a circuit-based low-latency anonymous communication service on the Internet. However, it needs to set up a large number of directory servers, which is difficult to envision in sensor networks.

Recently, techniques to randomize communications during the network setup phase to protect the anonymity of the sensor network infrastructure were proposed in [22]. On the other hand, we focus on defending against traffic analysis during the data sending phase. In addition, we assume that an adversary can launch active attacks such as injecting traffic in the network, and compromising sensor nodes. Preserving source-location privacy in WSNs was discussed in [16]. This work proposes randomization techniques such as fake packets, persistent fake sources, and a random walk to hide the location of the source of data packet from discovery. Unlike our approach, fake packets are always flooded, which incurs a high overhead cost. The key advantage of our approach is that it achieves much of the decorrelative effects of flooding at a fraction of the cost. Also, our focus is on the arguably more difficult task of hiding the *destination* of a data packet, i.e. base station, from discovery, since the patterns produced by the tree-structured routing are quite pronounced and difficult to hide.

6 Future Work

In the future, we plan to perform a more formal analysis to evaluate our schemes. We also plan to improve upon the GSAT search strategy and employ more advanced search strategies for an attacker, including genetic algorithms. Several aspects of the algorithm could be made more flexible, including the use of variable K hops instead of a fixed number of K hops for the length of fractal propagation.

7 Conclusion

This paper proposes countermeasures that make it difficult for an adversary to track the geographic location of a base station using traffic analysis. The paper presents four anti-traffic analysis techniques, MPR, RW, DFP and DEFP. In MPR and RW, random walks and randomness are introduced in the multi-hop path a packet takes from a sensor node to a base station. In DFP, fractal propagation and random fake paths are introduced to confuse an adversary from tracking a packet as it moves towards a base station. Finally, in DEFP, multiple, random areas of high communication activity are created to confuse an adversary into searching in a wrong area. The paper evaluates these techniques analytically and via a simulation using three evaluation criteria: total entropy of the network, total energy consumed, and the ability to guard against heuristic-based techniques to locate a base station. The combination of random walks, fractal propagation, and hot spots are shown to force an adversary to search nearly as many steps as if the network were uniformly flooded. Thus, the proposed countermeasures achieve most of the benefits of flooding, i.e. maximally confusing the attacker, at a fraction of the cost in overhead and energy.

References

[1] L. V. Ahn, A. Bortz, and N. J. Hopper. k-anonymous message transmission. In *10th ACM Conference on Computer and Communications Security*, pages 112–130, Washington D.C, USA, October 2003.

[2] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi. Routing through the mist: Privacy preserving communication in ubiquitous computing environments. In *International Conference of Distributed Computing Systems (ICDCS 2002)*, Vienna, Austria, July 2002.

[3] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, May 2003.

[4] S. A. Cook. The complexity of theorem-proving procedures. In *3rd Annual ACM Symposium on Theory of Computing (STOC'71)*, pages 151–158, Shaker Heights, Ohio, USA, March 1971.

[5] J. Deng, R. Han, and S. Mishra. Inrusion tolerance and anti-traffic analysis strategies in wireless sensor networks. In *IEEE 2004 International Conference on Dependable Systems and Networks (DSN'04)*, Florence, Italy, June 2004.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *13th USENIX Security Symposium*, Dan Diego, CA, USA, August 2004.

[7] W. Du, J. Deng, Y. Han, and P. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *10th ACM Conference on Computer and Communications Security (CCS'03)*, Washington D.C, USA, October 2003.

[8] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Conference on Computer and Communications Security, (CCS'02)*, Washington DC, USA, November 2002.

[9] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications of ACM*, 42(2), February 1999.

[10] Y. Guan, C. Li, D. Xuan, R. Bettati, and W. Zhao. Preventing traffic analysis for real-time communication networks. In *1999 IEEE MIlitary Communications Conference*, October 1999.

[11] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Cullar, and K. Pister. System architecture directions for network sensors. In *Nineth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'00)*, Cambridge, MA, USA, November 2000.

[12] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *11th Annual Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2004.

[13] C. Karlof, Y. Li, and J. Polastre. Arrive: Algorithm for robust routing in volatile environments. Technical Report Technical Report UCBCSD-02-1233, Computer Science Department, University of California at Berkeley, May 2002.

[14] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Networks*, 1(2-3), September 2003.

[15] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *CCS'03*, Washingon D.C, USA, October 2003.

[16] C. Ozturk, Y. Zhang, and W. Trappe. Source-location privacy in energy-constrained sensor network routing. In *2004 ACM Workshop on Security of Ad Hoc and Sensor Networks*, October 2004.

[17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. Spins: Security protocols for sensor networks. *Wireless Networks Journal(WINET)*, 8(5):521–534, September 2002.

[18] A. Pfitzmann and M. Waidner. Networks without user observability: Design options. In *Advances in Cryptology - EUROCRYPT'85*, Linz, Austria, April 1985.

[19] B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks. In *ACM SenSys'03*, Los Angeles, CA, USA, November 2003.

[20] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *10th National Conference on Artificial Intelligence (AAAI'92)*, pages 440–446, San Jose, CA, USA, July 1992.

[21] C. Shields and B. Levine. A protocol for anonymous communications over the internet. In *7th ACM Conference on Computer and Communications Security*, Athens, Greece, November 2000.

[22] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones. On providing anonymity in wireless sensor networks. In *10th International Conference on Parallel and Distributed Systems*, Newport Beach, CA, USA, July 2004.

[23] C. A. Waldspurger and W. E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *1st Symposium on Operating Systems Design and Implementation(OSDI'94)*, Monterey, CA, USA, November 1994.

[24] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges fo reliable multihop routing in sensor networks. In *First ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Log Angeles, CA, USA, November 2003.

[25] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route detection and filtering of injected false data in sensor networks. to appear in IEEE INFOCOM 2004.

[26] S. Zhu, S. Setia, and S. Jajodia. Leap: Efficient security mechanisms for large-scale distributed sensor networks. In *10th ACM Conference on Computer and Communications Security*, Washington D.C, USA, October 2003.

[27] S. Zhu, S. Setia, S. Jajodia, and P. Ning. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In *2004 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2004.