On the Use of Fake Sources for Source Location Privacy

Arshad Jhumka, Matthew Leeke and Sambid Shrestha

Department of Computer Science, University of Warwick
Coventry, CV4 7AL, United Kingdom

1 Introduction

Over the last decade the advent of wireless sensor networks has enabled several novel classes of application, including monitoring and tracking. In the case of monitoring applications, the deployment of sensor networks will vary from mission critical applications, such as military, health and asset monitoring, to non-critical applications, such as temperature and humidity control. For some critical applications, such as military surveillance or asset monitoring, security is an important requirement. One important aspect of security is privacy. Privacy is often important in critical applications, as sensitive information may need to be kept private. However, owing to the fact that wireless sensor networks operate in a broadcast medium, attackers can easily intercept messages and attempt to subvert a system.

The privacy threats that exist for sensor networks can be broadly classified along two dimensions, namely (i) *content-based* privacy threats and (ii) *context-based* privacy threats. Content-based privacy threats relate to threats that are based on the contents of messages, i.e., the threats are against the values generated by the various network layers, either at the application level, e.g., sensed values, or lower-layer levels, e.g., time-stamps. Context-based privacy threats are based on the context associated with the measurement and transmission of sensed data. Context is a multi-attribute concept that captures several environmental aspects associated with sensed data, including aspects such as location and time. While content-based threats are well-understood [16], context-based threats are becoming increasingly important. For content-based threats, nodes launching attacks are often modelled as Byzantine nodes [9], with cryptographic techniques often being used to address these problems [9] [2]. However, cryptographic techniques do not address context-based threats.

One aspect of context that is important in several applications is *location*. For example, in a military situation, location can be that of a soldier. In asset monitoring, the location can be that of a very expensive painting in a museum. Location information can be embedded in a message, but still remain inacces-

sible to an attacker due to the encryption of the message. Thus, as location information cannot be directly obtained, an attacker may attempt to deduce it using a variety of techniques, e.g., traffic analysis [4]. One important problem for monitoring applications is the problem of *source location* privacy. In this problem a wireless sensor network is monitoring an asset. The nodes detecting the presence of the asset, hence known as *source nodes*, will periodically send messages to a dedicated node, known as the *sink node*, for data collection. If the location(s) of the source node(s) is compromised, then an attacker can capture the asset. In the seminal work on source location privacy, the asset being monitored was taken to be a panda, which is an endangered animal [7].

It may be possible to infer location information through various techniques, depending on the power of the attacker. For example, Mehta et al. [13] assume that an attacker has a small wireless network of his own that can capture messages and shows how the location of source nodes can be inferred once messages have been intercepted. In another work, Kamath et al. [8] presented an approach that used triangulation to locate targets. In contrast, in a seminal paper on source location privacy, Kamat et al. [7] assume a single attacker, who can use the adopted routing protocol to infer the location of a source node. For example, in a military environment, soldiers on surveillance may relay information to a sink. An attacker can intercept these messages and trace them back to their source, thereby compromising the safety of the soldiers.

Several techniques to handle the source location privacy problem have been proposed, e.g., [7] [13] [1] [10] [20]. In this paper, we focus on the *fake source* technique. The fake source technique works as follows: whenever a real source node sends a message to the sink, another node, known as the fake source, will similarly send a message to the sink so as to confuse an attacker as to the location of the asset. However, current algorithms that are based on the fake source technique are of limited relevance, since they make a number of assumptions which limit their practicality. For example, (i) fake sources are often known *a priori* or arbitrarily chosen [13], (ii) fake sources are selected based on a prohibitively expensive pre-configuration phase or the need to have network-wide knowledge [7], (iii) no common attacker model is adopted, making it is difficult to understand the efficiency of proposed algorithms [7] [13] [10], and (iv) there is a lack of results demonstrating the level of privacy that can be afforded by the fake source technique.

To address the described limitations we investigate the efficiency of the fake source technique with respect to possible implementations, configurations and extensions. More specifically, (i) we detail two possible implementations of the fake source technique, both of which circumvent the need for fake sources to be known *a priori*, (ii) we make no undue assumption regarding the capabilities of sensor nodes, (iii) we investigate the efficiency of fake sources in presence of a distributed eavesdropper, which can have multiple implementations, e.g., using single or multiple attackers, and (iv) we develop a hybrid technique that accounts for multiple attackers.

Our main contributions are as follows: (i) we show that the fake source technique is efficient in enhancing the source location privacy of wireless sensor

networks, (ii) we detail an implementation, FS2, which achieves near perfect privacy level when enhanced with two protocol extensions proposed in this paper, namely (a) unique messages and (b) increased rates, (iii) we show that multiple attackers, with a very simple logic, can cause a drastic reduction in the level of privacy achieved, and (iv) we propose a novel hybrid technique, combining fake sources with phantom routing, and show that this hybrid technique provides a near perfect privacy level when at least one fake source is present.

One limitation of our approach is that, in trading off network energy consumption against source location privacy, there are times when there are no fake sources that are selected, thereby reducing the level of privacy. On the other hand, our protocols are adaptive, in that the privacy levels they impart can be fine-tuned through the setting of appropriate parameter values.

1.1 Paper Structure

The remainder of this paper is structured as follows: In Section 2 we provide a survey of related work. In Section 3 we define the network and attacker models used in the paper. In Section 4 we describe the protocols investigated, outlining their key characteristics and the reasoning on which they are based. We outline our experimental setup in Section 5. The results generated are presented and discussed in Section 6. Finally, Section 7 concludes this paper with a summary of what has been achieved and a discussion of future work.

2 Related Work

The problem of source location privacy first appeared around 2004 [7] [15]. Since then, the problem has been addressed using a variety of attacker models and assumptions. These varied attacker models and assumptions have led to the development of many solutions and techniques for enhancing source location privacy. Ozturk et al. [15] investigated the privacy imparted by flooding, and several other techniques such as fake source and phantom routing. A similar investigation was performed by Kamat et al. [7]. Subsequently, a new attack was shown to subvert the technique proposed by Kamat et al. [7], based on the assumption that nodes have access to their location using GPS devices. The most commonly adopted attacker model is a local attacker model, where nodes have only local knowledge. Other approaches have begun to investigate the impact of a global eavesdropper. Under these circumstances it has been shown that, for a fake source protocol where every node in the network acts as a fake source, maximal privacy can be ensured.

Despite the body of work relating to source location privacy in wireless sensor networks, little work has investigated the impact of possible fake source implementations and configuration parameters on source location privacy. This problem is directly addressed in this paper.

3 Models

In this section, we detail the network and attacker models adopted in this paper.

3.1 System Model

We define a wireless sensor node as a computing device equipped with a wireless interface and associated with a unique identifier. Communication in wireless sensor networks is typically modelled with a circular communication range centred on the node. We assume that all nodes have the same communication range, implying that nodes have omni-directional antennas. This assumption contrasts with work that assumes directional antennas [19]. Under this model a node is thought of as able to exchange data with all devices within its communication range. A wireless sensor network is a collection of wireless sensor nodes and is modelled as an undirected graph $G = (V, E)$, where V is a set of N wireless sensor nodes and E is a set of edges or links, each link being a pair of distinct nodes. A link exists between two nodes if they are in each other's communication range. Two nodes $m \in V$ and $n \in V$ are said to be 1-hop neighbours (or neighbours) iff $\{m, n\} \in E$, i.e., m and n are in each other's communication range. We denote by M the set of m's neighbours. The graph $G = (V, E)$ defines the topology of the network. In this paper, we focus on grid-like network topology, i.e., network of size $n * n = N$.

There exists a distinguished node S in the network called a sink, which is responsible for collecting data. Other nodes $v \in V \setminus \{S\}$ can sense data and then route the data to the sink for collection. In general, any node can be a source of sensed data. In this paper, however, we assume that only a subset of nodes can be a source of sensed data. Specifically, since we assume a grid-like network, we further assume nodes on the perimeter of the network to be sources of sensed data.

Sensor nodes route messages to the sink, generally using data aggregation convergecast protocols [5]. It has been previously shown that an attacker can use routing information to launch source-location privacy attacks. It has been also shown that a basic routing strategy like flooding does not have good source location privacy properties [7]. Variants have been proposed to improve the privacy provided by a given routing strategy, e.g., the use of fake source on top of flooding and directed random walk followed by flooding. We assume message content to be encrypted, thus it can only be read by the correct node and not by an attacker.

3.2 Attacker Model

In this paper an attacker is considered to be a set of sensor nodes. It has been proposed in [3] that the strength of an attacker for a wireless sensor network can be captured along two dimensions; (i) presence and (ii) actions. For example, presence can be local or global, while examples of actions includes eavesdropping and reprogramming. Using these two dimensions, a lattice of attacker strengths

was developed. Based on this lattice, we consider one type of attacker, namely a *distributed eavesdropping* attacker. There can be different implementations of this type of attacker. For example, such an attacker can be a single mobile person with a sensor node trying to eavesdrop. Another implementation can be multiple persons, each with a sensor node, eavesdropping on the network. In this paper, we consider these two possible implementations of a distributed eavesdropper and analyse the performance of a given routing strategy with respect to the attacker implementation. We also assume that the attacker does not have any knowledge of the network, i.e., the attacker does not know the network topology or the adopted routing algorithm. The only knowledge a distributed eavesdropper has is that which is deduced based on eavesdropping on the network. For example, when a message from a legitimate node within its neighbourhood is received, the sender of that message can be located but the source of the message is not known. The assumption of limited network knowledge is due to the fact that a typical sensor network is expected to contain tens of thousands of nodes, hence making it difficult for an attacker, even with a large amount of resources, to keep track of all sensor nodes.

4 Fake Sources: Problem Statement and Protocols

In this section, we first present a formalisation and description of the fake source technique for providing source location privacy. In our description, we abstract away from implementation details in order to identify important parameters for any implementation of the technique. We then present two possible distributed protocols that provide source location privacy, as well as presenting the implementations of the protocols used in this paper.

4.1 Problem Statement

Given a network $G = (V, E)$, a sink $S \in V$, an attacker $A \in V$ initially located at S, a safety period τ, and a frequency f_n for the rate of message generation by a node $n \in V \setminus \{S, A\}$. We assume A to be mobile, but only able to move one hop at a time.

The definitions and assumptions given above allow the fake source technique to be formalised as follows:

Fake Source: Given a real source ($s \in V \setminus \{S, A\}$), choose a set $F_s \subseteq G \setminus \{S, A\}$ and f_n such that A does not reach s within τ.

As its name suggests, the fake source technique involves selecting a subset of nodes to act as fake sources, i.e., to simulate a real source, by transmitting messages to surrounding nodes at a particular frequency. The current state-of-the-art techniques either (i) assume *a priori* knowledge of these fake sources, or (ii) choose fake sources randomly or (iii) require network-wide knowledge. However,

in practice fake sources have to be chosen at runtime. Further, network-wide knowledge is seldom available, or even desirable, due to the levels of energy consumption required to maintain it. Much previous work on the fake source technique distinguished between temporary fake sources and permanent fake sources [7]. Work by Kamat et al. [7] concluded that permanent fake sources outperform temporary fake sources in terms of privacy. Thus, in this paper, we focus on *permanent fake* sources. A permanent fake source is a node that continuously sends network messages to simulate a real source for at least the duration of message transmission from the real source, i.e., the fake source simulates the real source until the real source's data stream is over.

A real source is characterised by its location and message transmission rate. Any implementation of the fake source technique must investigate the impact of at least these two parameters. It has been argued that better privacy is achieved by having a fake source be approximately the same distance away from the sink as the real sources [7]. Based on this observation, we develop two algorithms, namely *Fake Source 1* (FS1) and *Fake Source 2* (FS2), that generate fake sources. We detail the FS1 and FS2 algorithms in Section 4.2 and Section 4.3 respectively.

4.2 Fake Source 1 (FS1) Implementation

We assume that real sources use a flooding protocol to send messages to a sink. This assumption is more general than assuming a single shortest path routing algorithm from source to sink. The flooding protocol is implemented as follows. The source generates a message and then broadcasts it to every node in its neighbourhood. The general structure of a message is:

$$< text, count, hash, destination, origin, hops >$$

For example, a transmitted message might be:

$< message, 1, H(message), -, 121, 0 >$, where H is some hash function.

In this example, the unique identifier of the source node is 121 and the *destination* field is empty. When a node receives a message, it checks the *count* value and determines whether it is a new message. If this is a new message, the node records the *count* value, increments the hop count and forwards the message. The node will drop any message that it has already been forwarded. Thus, using the flooding protocol, messages generated by the real source are forwarded by each node to the sink. When the sink receives the first such message it broadcasts an *away* message to each of its neighbours. For example, an *away* message might be:

$$< away, 1, H(away), -, 99, 5 >$$

Here, as a matter of example, the sink id is 99 and the hop distance between the real source and sink is 5. Observe that an *away* message is a specific instance of

the more general message structure. The nodes that receive the *away* message check if they have received a message from the source with the count value of 1. If they have received such a message, the node reduces the hop count by one and generates a *choose* message and broadcasts this message to its neighbours. The purpose of this *away* message is to ensure that only nodes that are further away from the real source forward the *choose* message.

Each intermediate node that receives the *choose* message will generate a random number R. If R is greater than a given threshold τ, the node will decrement the hop count and forward the message to its neighbours. Further, when a node receives a *choose* message that has a hop-count value of 0, the node generates its random number R. If R is greater than τ, then the node becomes a fake source and starts generating messages, which we call fake messages. The structure of the fake messages generated by a fake source is:

$$< fakemessage, 1, H(fakemessage), -, 19, 0 >$$

Again, observe that a *fake* message is a specific instance of the more general message structure.

4.3 Fake Source 2 (FS2) Implementation

The FS2 protocol is similar to FS1. As described previously, the real source floods network messages to be delivered to the sink. FS1 and FS2 differ in the way that intermediate nodes communicate messages. In FS2, intermediate nodes forward each *choose* message to all of its neighbours. In FS2, when a node receives a *choose* message that has a hop-count value of 0, the node generates a random number R. If R is greater than a given threshold then the node becomes a fake source and begins generating fake messages. The message structure of an *away* message is as in FS1, whilst *choose* messages have the same purpose as in FS1.

The key difference between the FS1 and FS2 techniques is that, in FS1, all intermediate and final nodes generate a random number to determine whether to forward a message or to become a fake source. On the other hand, in FS2, only the last nodes that receive a *choose* message generate the random number to decide whether to become a fake source.

4.4 Design Decisions

By selecting a set of fake sources during operation, our protocol becomes adaptive, in the sense that for every real source, there potentially exists a set of fake sources. This is in contrast with earlier work, where a set of fake sources is chosen at deployment time, is known *a priori* or requires network-wide knowledge to generate [7]. Note that, ideally, for maximum privacy, every node apart from the real source needs to be a fake source [13]. However, such an approach is not energy-efficient. Hence, there is a need to achieve a trade-off between privacy and energy consumption. *This is the thrust of this paper.* Note that

our algorithms are able to obtain fake sources which are a similar distance away from both the sink and real source by incrementing and decrementing the hop count.

5 Experimental Setup

In this section we outline the simulation environment and protocol configurations that were used to generate the results presented in this paper.

5.1 Simulation Environment

The simulation environment was based on the JProwler simulator [6]. JProwler is a discrete event simulator that can accurately model sensor nodes and the communications between them. JProwler provides two radio models, Gaussian and Rayleigh, which determine the signal level of transmissions and the communication range of nodes. The Rayleigh model was selected for use in all experiments because it models the situation where sensor nodes have high mobility, which is consistent with the assumption that an attacker will have high mobility within a sensor network. An experiment constituted a single execution of the simulation environment using a specified protocol configuration, network size and safety period. An experiment is terminated when the source node has been captured or the safety period has expired.

The JProwler simulator was extended to allow the safety period, capture ratio and total energy consumption to be monitored during simulation. Energy consumption was measured independently for each node in the network. The adopted energy model was consistent with [17]; thus values for node voltage (V_{node}), current at idle (I_i), current at send (I_s) and current at receive (I_r) were required. As in [17], $V_{node} = 3V$, $I_i = I_r = 7mA$ and $I_s = 21.5mA$. All nodes were assumed to operate at maximum power, thus the transmission strength of each node was also maximal.

5.2 Network Configuration

A square grid network layout was used in all experiments so the network is symmetric. Experiments were performed for network sizes of 11, 15, 21 and 25, i.e., networks of 121, 225, 441 and 625 nodes respectively. However, our proposed algorithms are not based on the network topology, hence will work with any type of network. A single source node generated messages at a given rate and a single sink node collected the messages. The source and sink nodes were distinct. Messages from the real source were generated at a constant rate of 1 message per second. The sets of experiments for each network size were performed for five different source node locations; (iv) the four corners of the grid and (ii) a random location at the perimeter of the grid. To ensure the validity of the results presented, 100 repeats were performed for each source location. The sink node was located at the centre of the grid to achieve symmetry in the

network. Nodes were located 28 meters apart. The node separation distance was determined analytically, based upon the static fading values calculated by the adopted radio model. This separation distance ensured that messages (i) pass through multiple nodes from source to sink, (ii) can move only one hop at a time and (iii) can only be passed to horizontally or vertically adjacent nodes.

5.3 Protocol Configuration

All protocols were implemented according to the descriptions given in Section 4. The flooding protocol was used as a baseline against which other protocols were measured. It was not technically possible to compare our algorithms against other known fake source implementations due to the various differences in underlying assumptions, e.g., network-wide knowledge as opposed to local knowledge. Experiments involving FS1 and FS2 were conducted with threshold values of 0.5, 0.6, 0.7, 0.8 and 0.9.

5.4 Protocol Extension

In Section 4 we outlined the algorithms that we develop in this paper. In this section we give an overview of the various extension and parameters which impact the privacy afforded by these algorithms.

Unique Messages (UM): In FS1 and FS2 fake sources generate fake messages which are identical to those generated by a real source. In networks with more than one fake source node, this results in a recipient, i.e., an attacker or intermediate node, dropping messages from two different fake source nodes on the basis that the messages were identical. If fake source nodes generated unique messages, duplicates would never be encountered and this message dropping could not occur. The unique messages extension is intended to ensure that an attacker will be forced to relocate more frequently. However, it should be remembered that the energy consumption of intermediate nodes is likely to increase due to increased network traffic.

Increased Rates (IR): To this point it has been assumed that real sources and fake sources broadcast messages at a rate of 1 message per second. The increased rates extension observes protocol performance when the broadcast rates of fake sources is increased to 2 and 4 messages per second. The premise behind this decision is to enable fake source messages to catch up with an attacker, thus being able to divert the attacker from the real source towards the fake source.

Multiple Attackers (MA): The possibility of multiple attackers is explored by having four attackers co-ordinate their actions whenever a new message is received. The network grid is divided into quadrants, where each attacker was assigned a quadrant within which to operate. When an attacker receives a new message, they move to the sender of the message and instruct all other attackers to drop messages which are identical to the received message. The extension was run in conjunction with the unique messages extension, as it seeks to ensure that a received message provokes a response from exactly one attacker.

6 Results: Single Attacker

In this section, we present the results generated by the described experiments. We first implement the distributed eavesdropper attacker model as a single mobile attacker. We then focus on another implementation of the distributed eavesdropper model, namely a multiple attacker variant, investigating the impact of these two different implementations on the fake source technique.

6.1 Single Attacker on Flooding

The flooding protocol is used as a baseline routing technique, against which we will compare all privacy-aware routing protocols proposed here. The reasons for using flooding as a baseline are (i) it has a high message delivery ratio, hence an attacker can capture a high proportion of messages, (ii) a subset of messages will be delivered along the shortest path from source to sink, thus the attacker can try to reach the real source as quickly as possible, and (iii) it has been shown to offer poor levels of privacy [7].

A concept known as *safety period* was introduced in [7] to capture the number of messages that have to be sent by the real source before it is detected. For example, a low safety period will indicate low levels of privacy. On the other hand, maximum privacy can be characterised by an infinite safety period, i.e., the real source is never detected. In general, for high levels of privacy, the safety period should be high. However, this definition implicitly assumes that an asset will not change its position over time, which is only applicable in certain situations.

In this paper, to circumvent this limitation, we use an alternative, but similar, definition for safety period. For each network size, using flooding, we calculate the average time it takes to detect the real source, i.e., the average time to capture the asset. Then, when running simulations for privacy-aware protocols, we allow for a higher safety period, since the premise is that the proposed routing techniques will provide a higher source location privacy and may require more messages, hence more time. The reason for this new definition of safety period is two-fold: (i) it bounds simulation time, and (ii) it allows us to define a time period within which an asset needs to be captured before that asset will move on from its current location. The second observation is particularly well-suited to animal/habitat monitoring. The safety period, for each network size, for flooding is shown in Table 1, Column 2. Observe that the safety period for privacy-aware network routing protocols is twice the average time taken for source detection/asset capture for the baseline flooding algorithm (Table 1, Column 3).

6.2 Single Attacker on FS1

We observe from Figure 2b that algorithm FS1 offers a similar level of privacy as baseline routing, i.e., FS1 provides poor privacy as its capture ratio is comparable to that of baseline routing, especially when the threshold is high. This

Table 1: Safety period for different network sizes: (i) Column 2 for baseline flooding, (ii) Column 3 for privacy-aware protocols.

Network Size	Average Time Taken (secs)	Safety Period (secs)
11×11	16.00	32.01
15×15	23.41	46.83
21×21	34.74	69.50
25×25	42.89	85.77

is due to the low number of fake sources selected, which is due to the very low probability of a node being selected as a fake source. This low number of fake sources is reflected in the fact that the network energy consumed is comparable with the network energy consumed in baseline routing.
To understand the reason for the low number of fake sources, we must consider the FS1 algorithm. As detailed in Section 4.2, the first phase of FS1 is a flooding phase, during which the real source floods the network with messages. On receiving the first of these messages, the sink sends a *choose* message to nodes h hops away, where h is the distance between the source and sink. Thus, for a threshold value τ, a node h hops away will decide to become a fake source with probability based on Equation 1.

$$(1 - \tau)^{h-1} \tag{1}$$

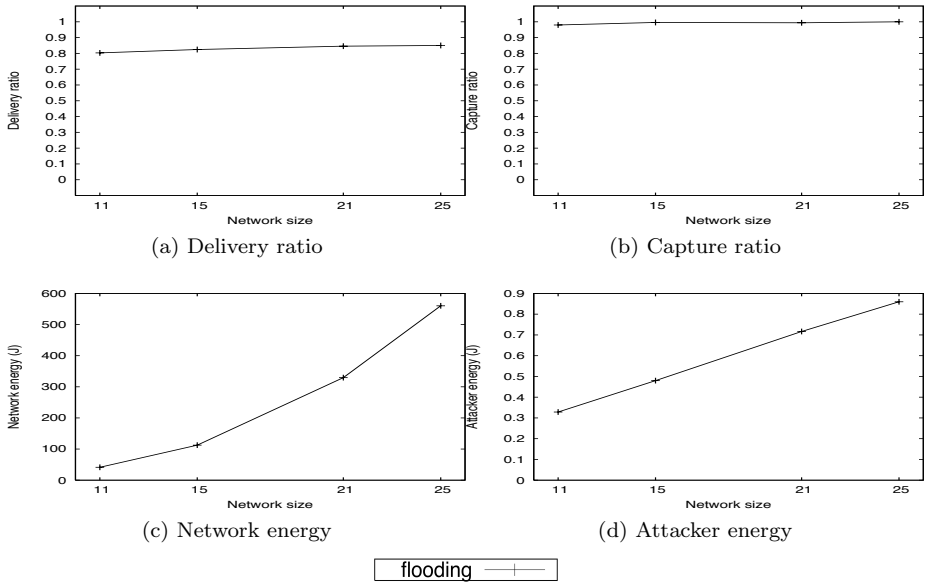
For example, with a threshold value of 0.9 and a distance of 5 from source to sink, the probability of a given node 5 hops away becoming a fake source node is approximately 0.00001.

Even when using various thresholds $(0.5, 0.6, \ldots, 0.9)$, the capture ratio remains very high. On the other hand, Figure 2e shows that, whenever there is a fake source in the network, the capture ratio becomes very low. This provides evidence that, as a technique, the use of fake sources does provide a very good level of privacy.

6.3 Single Attacker on FS2

To address the limitation of FS1 in selecting appropriate fake sources, we propose algorithm FS2, which is detailed in Section 4.3.

The first phase of FS2 is a flooding phase, as in FS1, during which the real source floods the network with messages. Upon receiving the first of these messages, the sink sends a *choose* message and each intermediate node becomes a forwarder of the message. Only potential fake sources, which are h hops away, generate a random number to decide whether to become a fake source. So, in this case, the probability of a potential fake source node of actually becoming a



(a) Delivery ratio (b) Capture ratio

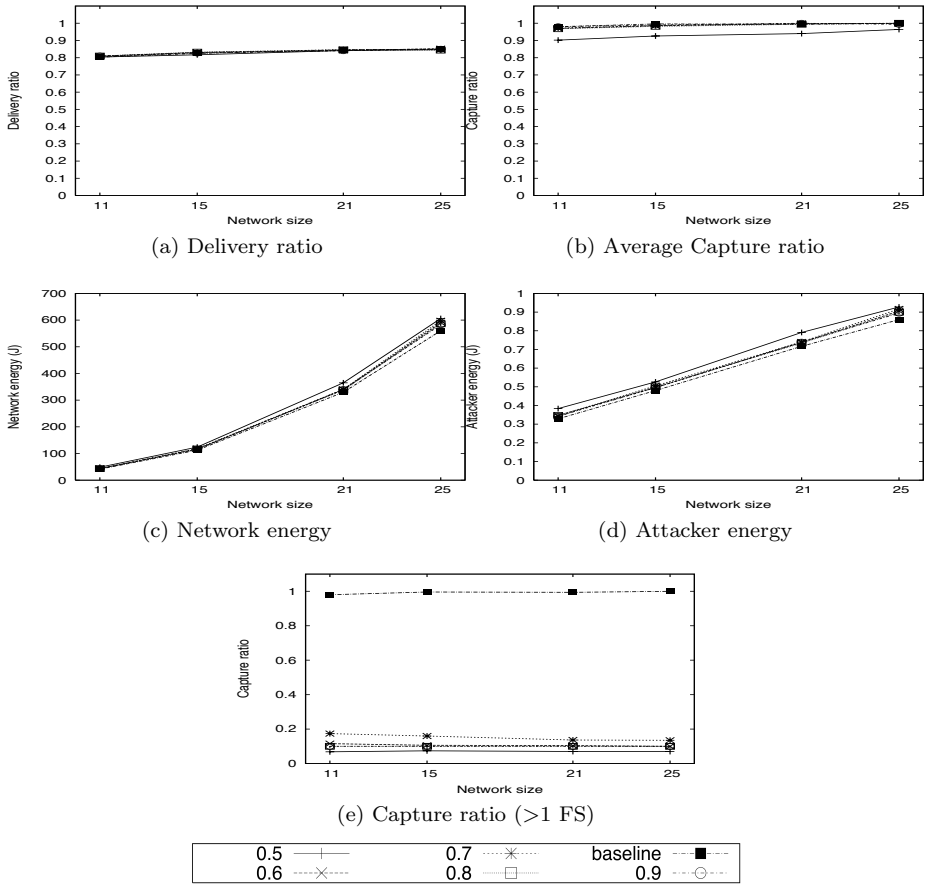
(c) Network energy (d) Attacker energy

Figure 1: Flooding (Baseline)

fake source is given by $(1-\tau)$, where τ is a given threshold, in contrast to FS1 where the probability is $(1-\tau)^{h-1}$.

As expected, we observe that this FS2 significantly improves upon the FS1 algorithm. To determine the level of privacy imparted by the FS2 algorithm, a closer look at the capture ratio from Figure 3b shows that the technique provides a significant improvement on privacy by reducing the capture ratio by as much as 50%. On the other hand, the delivery ratio is comparable to that of baseline routing, i.e., flooding, since the technique does not generate significantly more messages than in baseline routing. The network (resp. attacker) energy spent to provide privacy to (resp. capture) the asset increases. The reason the network energy increases faster than the attacker energy is because FS2 may select more than one fake source, whereas the number of attackers remains the same.

On the other hand, it has been observed that, sometimes, there are no fake source selected by FS2. By decreasing the threshold $(0.9 \ldots 0.5)$, one can increase the probability of at least one node being selected, while also increasing the probability of multiple fake sources being selected, hence providing higher levels of privacy (Figure 3b) at the expense of higher network energy consumption. This is corroborated by the fact that the network energy consumption (Figure 3c) is higher than the network energy consumed in FS1 (Figure 2c). We also observe that attacker energy is higher in FS2 (Figure 3d) than in FS1 (Figure 2d), since an attacker is receiving more messages in FS2 than in FS1.



(a) Delivery ratio

(b) Average Capture ratio

(c) Network energy

(d) Attacker energy

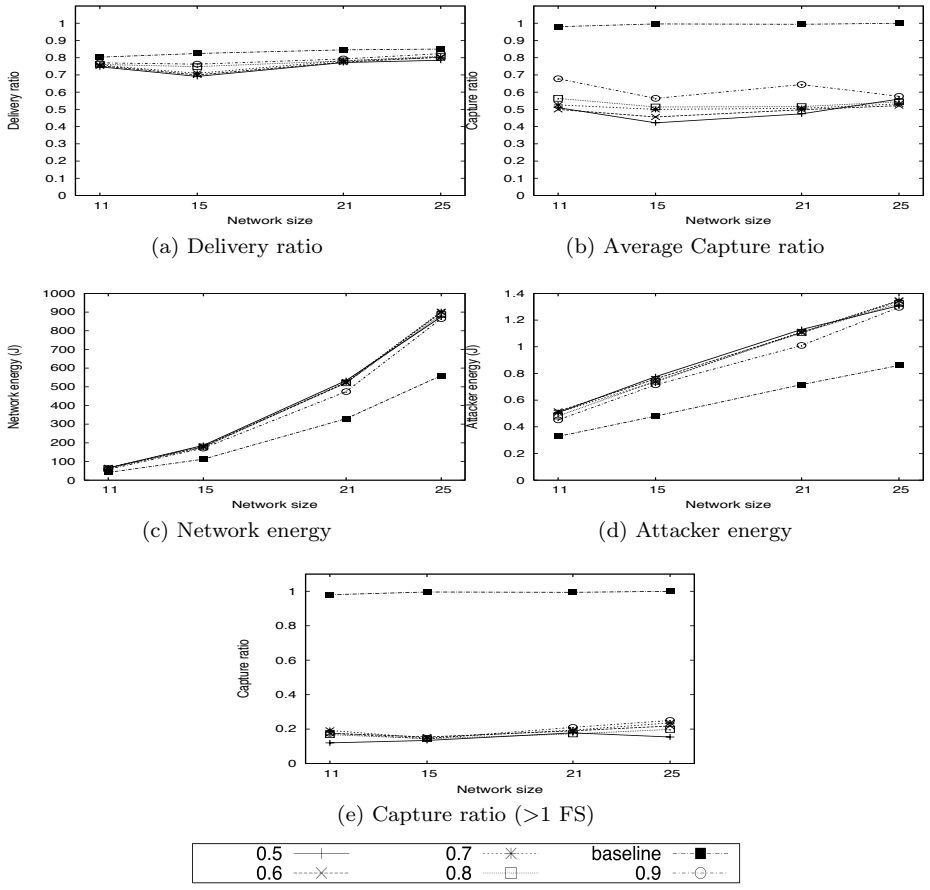
(e) Capture ratio (>1 FS)

Figure 2: Performance of FS1

6.4 Single Attacker on FS2 with UM

One of the limitations that can be observed with FS2 is that nodes drop fake messages whenever they were deemed identical. As fake sources were sending identical messages, not all of them were being forwarded towards the attacker, thereby reducing the efficiency of FS2. Thus, an implementation change that was undertaken was for each fake source to generate unique messages, as detailed in Section 5.4.

Figure 4c shows how incorporating the unique message extension increases network energy consumption. This increase, which is attributable to the increased volume of traffic on the network, can be as high as 75.9% with respect to baseline routing, whilst the corresponding increase using only FS2 was 60.4%. Broadcasting unique messages has a negative impact on an attacker. The larger volume of network traffic means that an attacker receives more messages and



(a) Delivery ratio

(b) Average Capture ratio

(c) Network energy

(d) Attacker energy

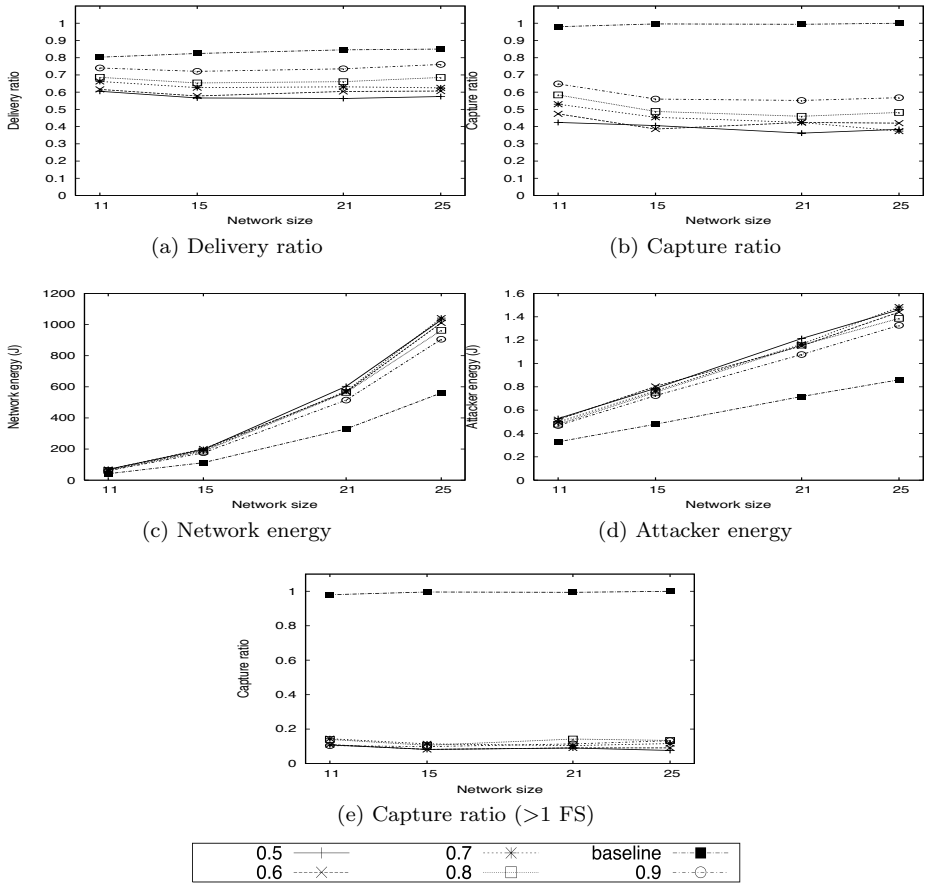
(e) Capture ratio (>1 FS)

Figure 3: Performance of FS2

consumes more power. As messages are unique, an attacker must also relocate more frequently and will therefore take longer to find a real source. The consequences of this can be seen in Figure 4b, which shows that the capture ratio can be reduced by more than 60%, compared to flooding. In fact, the performance of FS2 with unique messages is better than with FS2 alone. This level of privacy is also an improvement over FS2, which reduced the capture ratio by up to 50% compared to the same baseline. Also, in runs where fake sources were selected, the capture ratio was better for FS2 with unique messages, with a capture ratio of as low as 10%.

6.5 Single Attacker on FS2 with UM & IR

The next improvement, namely increased rates, as detailed in Section 5.4, investigates the impact when the fake sources generate messages faster than the



(a) Delivery ratio

(b) Capture ratio

(c) Network energy

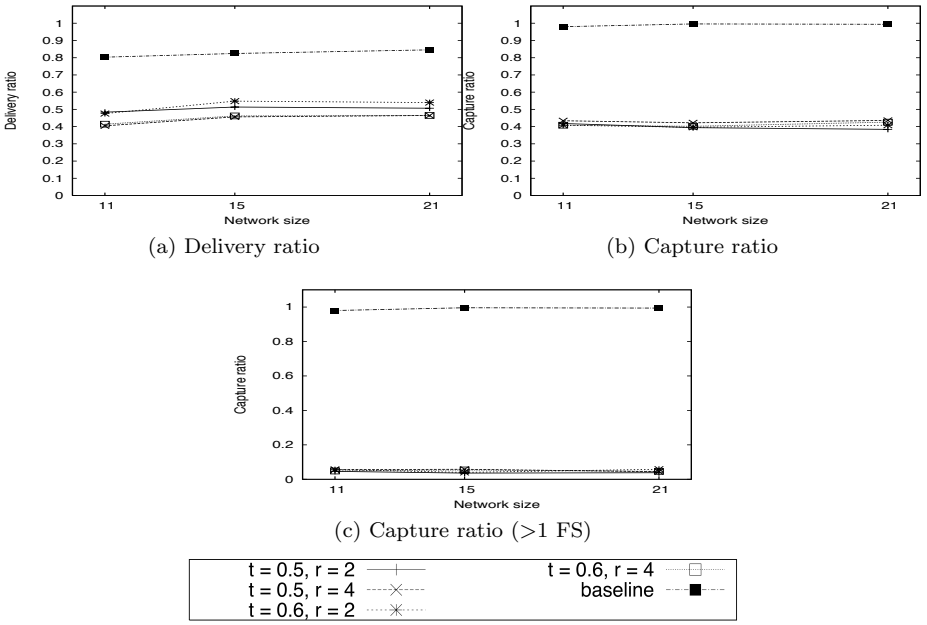
(d) Attacker energy

(e) Capture ratio (>1 FS)

Figure 4: Performance of FS2 with UM

real source. The rationale behind the decision is that, when broadcast at an increased rate, fake messages can catch up with an attacker earlier, thereby drawing them towards a fake sources more quickly, which will in-turn cause a reduction in the observed capture ratio.

The graphs in Figure 6 show a 2-fold and a 4-fold increase in message rates, with thresholds of 0.5 and 0.6. Figures 5b and 5c respectively show the capture ratio in a general situation and in a situation when a fake source is definitely present. We observe that the capture ratio is nearly 0 when there is at least one fake source in the network (Figure 5c). Further, the capture ratio is lower than when only unique messages are used (Figures 4b and 4e), thereby confirming the intuition that increased rate, together with unique messages, is indeed effective in reducing the capture ratio.



(a) Delivery ratio

(b) Capture ratio

(c) Capture ratio (>1 FS)

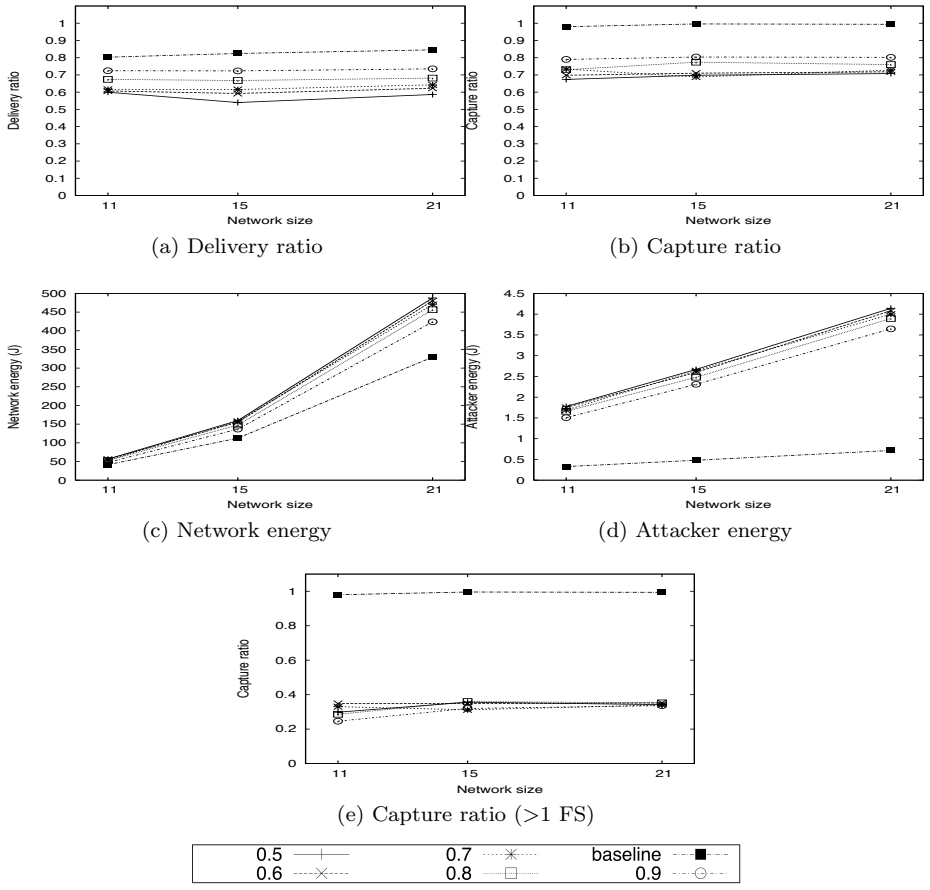
Figure 5: Performance of FS2 with UM and IR

6.6 Multiple Attackers on FS2 with UM

FS2 with UM has been shown to provide the very good source location privacy (Figures 4b and 4e). We now investigate the impact that incorporating multiple attackers has on the performance of the FS2 algorithm with unique messages.

Figure 6b shows an increase in the capture ratio of approximately 100%, to an overall of around 80%, over FS2 with unique messages. This increase can be explained by the simple coordination between attackers. If an attacker has a fake source within its operating quadrant, it will receive corresponding fake messages and prevent the other attackers from reacting to the associated fake source, even when they receive the same fake messages. An attacker A will be located in a quadrant that contains the real source. As other attackers will potentially receive fake messages before they are received by A, A can move towards the real source whilst dropping messages from fake sources, i.e., A will not be perturbed by the fake messages. The energy consumption associated with the attacker is expected to increase in the context of multiple, coordinating attackers, and this is corroborated by Figure 6d, which shows an increase in attacker energy as compared to Figure 4d.

When operating in isolation, an attacker only needs to receive messages. However, when multiple attackers are present, additional messages must be sent and received in order to facilitate cooperation among attackers. Since the network deployment is essentially the same as in the case of FS2 with unique



(a) Delivery ratio

(b) Capture ratio

(c) Network energy

(d) Attacker energy

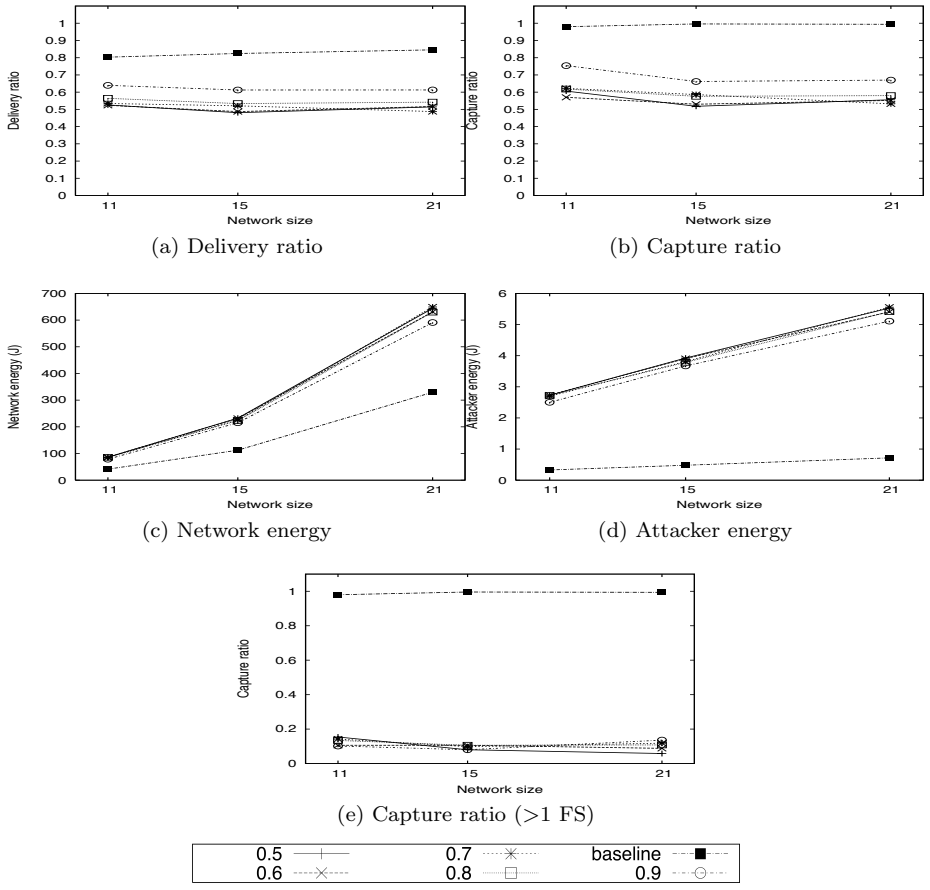
(e) Capture ratio (>1 FS)

Figure 6: Multiple attackers on FS2 with UM

messages, we do not expect any deviation in terms of network energy and delivery ratio, and this is corroborated by Figures 6a and 6c. This suggests that these metrics are invariant to the presence of multiple attackers. At this point, an important point to observe is that, though a single mobile eavesdropper attacker and multiple eavesdropper attackers are viable implementations of a distributed eavesdropper, the fact that they have such different impacts suggests the existence of dimensions to an attacker model that are not captured by the attacker taxonomy for wireless sensor networks, as proposed by Benenson *et al.* [3].

6.7 Multiple Attackers on FS2 with UM & PR

Since multiple attackers are able to cause an increase in the capture ratio when using FS2 with unique messages, we propose a simple technique to circumvent the coordination logic of the attackers. The premise of our circumvention is



(a) Delivery ratio

(b) Capture ratio

(c) Network energy

(d) Attacker energy

(e) Capture ratio (>1 FS)

Figure 7: Multiple attackers on FS2 with UM & PR

that only one attacker should follow any given fake message. To achieve this we adapt the Phantom Routing (PR) protocol proposed in [7]. In Phantom Routing a real source initially sends a message on a directed random walk of length λ. After λ hops, the node that has the message behaves as the "real" source of the message and proceeds to flood the network with messages. The idea here is that, since different "real" nodes will initiate the flooding, the attackers will not receive successive messages, thus making it difficult to capture the source within the safety period.

Results for the adapted Phantom Routing approach are presented in Figure 7, which shows for FS2 with UM and PR in presence of multiple attackers, the capture ratio falls to around 50%, which offers a 30% improvement on using only FS2 with UM. In runs when at least one fake source was selected (Figure 7e), the capture ratio falls to below 10%, showing a dramatic increase in privacy. These results suggest that a hybrid technique of fake sources combined

with phantom routing is a promising approach to delivering high levels of source location privacy.

7 Conclusion

In this section, we summarise the achievements of this paper and provide a discussion of future work.

7.1 Summary

In this paper, we have investigated the problem of providing source location privacy in wireless sensor networks in the presence of a distributed eavesdropper attacker. We have investigated the efficiency of different variants of the fake source technique in presence of two different implementations of the distributed eavesdropper attacker model. We have detailed the variants of the fake source technique which avoid a prohibitively expensive pre-configuration phase and the requirement that fake sources to be known *a priori*. One fake source implementation, FS1, showed poor level of privacy, owing to the very low probability of a node being chosen as a fake source in the network. On the other hand, the FS2 protocol achieved, on average, a good level of privacy. In the case where at least one fake source was selected, the FS2 protocol achieves near perfect privacy. Our fake source implementations achieve a trade-off between privacy and network energy consumption. Additionally, the privacy achieved by FS2 is improved by protocol extensions such as unique messages and increased rates. In the development of the proposed implementations, no assumption was made regarding the capabilities of sensor nodes, except that they are capable of sensing and relaying data.

We have also investigated the impact of multiple attackers, which is an alternative implementation of the distributed eavesdropper model, on the fake source protocol FS2 with unique messages. In this context, there was a decrease in the level of privacy provided, resulting in a 100% increase in capture ratio. To circumvent this increase in capture ratio, hence drop in privacy, we have proposed a hybrid technique, namely combining the FS2 with unique messages protocol with phantom routing. This hybrid protocol efficiently handled the privacy problem, especially when at least one fake source was selected in the network.

7.2 Future Work

In future work, we plan to undertake a wider exploration of the applicability of the fake source technique. In particular, we will investigate the impact of different network topologies on the efficiency of the fake source technique. Given our observation that the presence of at least a single fake source yields significant reduction in capture ratio, we will develop algorithms that can provide such a

guarantee. We will also investigate routing algorithms that are able to handle multiple attackers.

References

[1] S Armenia, G Morabito, and S Palazzo. Analysis of location privacy/energy efficiency tradeoffs in wireless sensor networks. In *Proceedings of th 6th International IFIP-TC6 Networking Conference on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pages 215–226, November 2007.

[2] I C Avramopoulos, H Kobayashi, R Wang, and A Krishnamurthy. Highly secure and efficient routing. In *Proceedings of the 23rd Conference on Computer Communications*, pages 197–208, March 2004.

[3] Z Benenson, P M Cholewinski, and F C Freiling. *Wireless Sensor Network Security*, volume 1, chapter Vulnerabilities and Attacks in Wireless Sensor Networks. IOS Press, April 2008.

[4] J Deng, R Han, and S Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 113–126, 2005.

[5] Arshad Jhumka. Crash-tolerant collision-free data aggregation scheduling in wireless sensor networks. In *Proceedings International Symposium on Reliable Distributed Systems*, 2010.

[6] JProwler. http://w3.isis.vanderbilt.edu/project-s/nest/jprowler/, 2010.

[7] U Kamat, Y Zhang, and C Ozturk. Enhancing source-location privacy in sensor network routing. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 599–608, November 2005.

[8] S Kamath, E Meisner, and V Isler. Triangulation based multi target tracking with mobile sensor networks. In *Proceedings IEEE International Conference on Robotics and Automation:*, pages 3283–3288, 2007.

[9] L Lamport, R E Shostak, and M C Pease. The byzantine generals problem. *ACM Transactions on Lanaguages and Systems*, 4(3):382–401, July 1982.

[10] S-W Lee, Y-H Park, J-H Son, S-W Seo, U Kang, H-K Moon, and M-S Lee. Source-location privacy in wireless sensor networks. *Korea Institute of Information Security and Cryptology Journal*, 17(2):125–137, April 2007.

[11] Y Li and J Ren. Preserving source-location privacy in energy-constrained wireless sensor networks. In *Proceedings of the 6th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 493–501, June 2008.

[12] Y Li and J Ren. Providing source-location privacy in wireless sensor networks. In *Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications*, pages 338–347, August 2009.

[13] K Mehta, D Liu, and M Wright. Location privacy in sensor networks against a global eavesdropper. In *Proceedings of the IEEE International Conference on Network Protocols*, pages 314–323, October 2007.

[14] Y Ouyang, Z Le, D Liu, J Ford, and F Makedon. Source location privacy against laptop-class attacks in sensor networks. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, pages 22–25, September 2008.

[15] C Ozturk, Y Zhang, and W Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 88–93, October 2004.

[16] A Perrig, J Stankovic, and D Wagner. Security in wireless sensor networks. *Communications of the ACM - Special Issue on Wireless Sensor Networks*, 47(6):53–57, June 2004.

[17] V Shnayder, M Hempstead, B Chen, G W Allen, and M Welsh. Simulating the power consumption of large scale sensor network applications. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 188–200, May 2004.

[18] Y Tscha. Routing for enhancing source-location privacy in wireless sensor. *Journal of Communication and Networks*, 11(6):589–598, December 2009.

[19] Y Xi, L Schwiebert, and W Shi. Preserving source location privacy in monitoring-based wireless sensor networks. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium*, 2006.

[20] J Yao and G Wen. Preserving source-location privacy in energy-constrained wireless sensor networks. In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*, pages 412–416, June 2008.